



UNIVERSIDAD NACIONAL AUTONOMA DE
MEXICO

FACULTAD DE CONTADURÍA Y
ADMINISTRACION

CONFIGURACIÓN Y PUESTA A PUNTO DE UN CLUSTER DEL
DEPARTAMENTO DE SUPERCÓMPUTO DE LA DGSCA-UNAM
PARA EL PROYECTO PRAGMA

DISEÑO DE UN SISTEMA O PROYECTO PARA UNA
ORGANIZACIÓN
QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTAN:

PAULINA MARTÍNEZ PORTILLO
PERLA CRISTAL PALACIOS MORALES

ASESOR:
L.I. LUZ MARÍA RAMÍREZ ROMERO



MÉXICO, D.F.

2009



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

agradecimientos de Paulina

A Díos

Por llenar de bendiciones mi vida, iluminar mi camino y ser mi fortaleza.

A mis padres, Margarita y Manuel

Agradezco infinitamente a la vida el haberme concedido nacer en nuestro maravilloso hogar.

Agradeceré y valoraré por siempre su inmenso amor, educación, apoyo y por dejarme ser.

Este trabajo, no es mío, es de ustedes.

A mis hermanas, Carolina, Olivia y Susana

Por su compañía, regaños, atenciones, y sobre todo, gran cariño. Tengo la certeza de que siempre

contaré con ustedes.

A mis sobrinos, Lilián, César, Ángel, Brisa, Marlene y Melany

Por inyectarme siempre de energía positiva e inigualable compañía.

A mi Tía Juanita Portillo

Por sus sabios consejos e incondicional atención.

Por encomendarme con Díos, yo sé que sus oraciones fueron escuchadas.

A Perla

Por compartir esfuerzos, comidas, dulces, clases, lecturas, risas, llantos, fotos, música, sorpresas, frío, calor, mañanas, tardes y noches consecutivas durante nuestro trayecto universitario y aún después. Gran amiga y compañera de ruta, mil gracias por todo.

PSP.

A mis amigos y más que amigos

A los que están cerca y los que no tanto, que aparecieron en mi vida y la han cambiado.

Por crecer, soñar, compartir y madurar juntos, gracias por todo y por todo lo que falta.

A Patricia Adriana Aguilar Salazar

Por demostrarme y enseñarme de tu fortaleza, que todo se aprende, que todo se puede. Gracias

por siempre estar ahí. Gracias por tu amistad.

A Lourdes Yolanda Flores Salgado (Super Yolí)

Por su invaluable y generoso apoyo para el inicio, desarrollo y culminación del presente.

Por recibírnos con una sonrisa, con disposición y sobre todo, con la calidez de una amistad.

Escucharnos, corregírnos, ordenarnos las ideas, dedicarnos considerables lapsos de su valiosísimo tiempo, siempre paciente y amable...

Por sus clases de excelencia, siempre serás fuente de inspiración y admiración.

Mil Gracias Yolí. Que Díos te bendiga.

A Eduardo Murrieta León

*Por dedicarnos gran parte de su valioso tiempo.
Por compartir y amenizar las tardes sabatinas y a veces dominicales con sus conocimientos,
una taza de café sin azúcar y su buen sentido del humor.
Siempre atento, paciente y sencillo.
Han sido pocas las personas que he considerado como excelentes en todo sentido,
y sin duda, eres una de ellas.*

A Luz María Ramírez Romero

*Por su valiosa asesoría y gran apoyo para la realización del presente.
Por dedicarnos una parte considerable de su tiempo y ser paciente.
Muchas gracias Luz*

Al Departamento de Supercómputo de la DGSCA

*Por todas las facilidades otorgadas y concedernos la oportunidad
de colaborar con este proyecto.*

A la UNAM

*Por sus formidables maestros, instalaciones y privilegios, es un honor ser parte de esta
comunidad.*

A los demás

*Que han quedado en los espacios más recónditos de mi memoria, pero que estuvieron implicados
en esculpir a esta Paulina*

A todos, mi más sincero reconocimiento y gratitud...

agradecimientos de Perla

A mi mamá

Gracias por estar siempre ahí, quererme incondicionalmente, cuidarme y preocuparte por mí, enseñarme grandes valores que ahora son parte de mi personalidad. Gracias por ser un ejemplo de fuerza, un ejemplo de que puedes hacer las cosas siempre y cuando así lo desees. Muchas gracias por compartir mis logros y por hacerlos tuyos también, por estresarte conmigo cada que tenía un examen o tenía que entregar un proyecto. Gracias por creer en mí y por no dejarme caer.

A mi papá

Gracias por enseñarme el grandísimo valor de la responsabilidad, haberme dado una excelente educación con la cual he llegado hasta donde estoy ahora, inculcarme que no hay nada mejor que brindar una sonrisa, un apretón fuerte de manos y un "buenos días" para abrir miles de puertas. Gracias por haberme dado una infancia con recuerdos y colores muy gratos. Gracias por enseñarme que una canción puede llevarte muy lejos de tus problemas y lo que es más, puede hacer que los olvides.

A ambos

Gracias por haberme dado la mejor herencia que pude haber recibido: mi carrera profesional. Por tolerar mis arranques de perfeccionismo y haberme tenido tanta paciencia y tolerancia para despertarme cada mañana sin falta. Gracias por haberme enseñado a dar las gracias. Y sobretodo gracias por dejarme ser Libre.

A mis hermanas

Las dos mejores hermanas que un ser humano pueda tener. Gracias hermanas mías por estar siempre ahí, en las buenas y en las malas, por haber sido mis maestras a domicilio, mi ejemplo a seguir, por enseñarme el significado de apoyo, perseverancia, unión y amistad. Por compartir y disfrutar conmigo mis logros y por ser a la vez, mis mejores amigas. Gracias por todo el soporte emocional y respaldo económico que me brindaron y me siguen brindando cuando lo necesito. Gracias por todo lo que han hecho por mí. Dios se los regresará multiplicado.

A Paulína

Por tantos momentos compartidos, por la ñoñez conjunta, por mantenerme siempre con mil cosas por hacer, natación, inglés, dulces, servicio social, clases en la mañana y en la noche, graduación, plan. Por todas esas desveladas en las que se nos iba el tiempo platicando y reflexionando. Por estar conmigo en este gran reto y haberme permitido conocerme a través de las vivencias que pasamos juntas. Por tu paciencia. Gracias por ser y estar.

PSP

A mis amigos

Gracias por haber sido la materia prima de mis años maravillosos como estudiante, la mejor época de mi vida. Cada etapa, cada ciclo, se conformo de hermosas luces que alumbraron mi camino. Gracias a todas esas luciérnagas que me acompañaron en algún tramo de ruta, siempre estando ahí para brindarme la mejor de las compañías, dejándome tantos momentos gratos, tantas risas involuntarias, enseñanzas y lecciones de vida. Gracias por hacer que valiera la pena.

A todos ustedes gracias por su infinita paciencia durante la realización de este gran proyecto, su comprensión fue mi principal motivación...

A Luz

Por tu apoyo, paciencia y tiempo para que este proyecto saliera adelante.

A Eduardo Murrieta

Gracias por todos los sábados que compartiste tu valioso tiempo y grandes conocimientos con nosotras, apoyándonos con temas muy avanzados del cluster, sin no omitir los divertidos momentos geek que nos hiciste pasar, gracias de verdad, sin tí nuestra tesis no hubiera llegado tan lejos.

Un agradecimiento especial a Yolí

Sin tu ayuda, paciencia, asesoría, tiempo, amabilidad, motivación, y un largo etcétera, difícilmente estaríamos presentando un trabajo como éste. Gracias por haberme dado la mejor de las formaciones académicas y presentarme la licenciatura como una de las mejores decisiones que he tomado hasta ahora. Tus clases fueron una de las razones por las que hoy amo mi carrera. Gracias por haberme dado tantas oportunidades profesionales, desde estar de "esclavo", el servicio social, iiel plan de becarios!!, hasta el término de mi tesis. Me enseñaste el verdadero valor de la docencia. Aprendí mucho caminando a tu lado Yolí. Tu apoyo ha sido para mí un valor invaluable que solo Dios y la vida sabrán pagarte. Espero de verdad que Dios te cuide y bendiga. Gracias por todo Yolí.

Al Departamento de Supercómputo

Por permitirnos ser parte de su equipo y brindarnos toda la colaboración, herramientas y conocimientos compartidos, que necesitábamos.

A la UNAM, mi segunda casa

Por acogerme en sus aulas, abrirme los brazos al conocimiento, ceñirme en sus hermosas, instalaciones y darme la oportunidad de formarme académicamente en ella. Es realmente un orgullo haber sido alumna de esta magnífica institución.

A Dios

Por darme la vida y permitirme estar en los lugares y tiempos correctos para encontrarme con todas estas maravillosas personas que han hecho que mi estancia en este mundo tenga sentido.

ÍNDICE GENERAL

Página	
1	Introducción
5	Convenciones Tipográficas
8	Capítulo 1. Marco Teórico y Antecedentes
8	1.1 Marco Teórico
8	1.1.1 Administración de Sistemas
29	1.1.2 Arquitectura de Computadoras
35	1.1.3 Cómputo Paralelo
36	1.1.4 Cómputo Distribuido
38	1.1.5 Supercómputo
40	1.1.6 Cluster
51	1.1.7 Grid
54	1.1.8 Mecanismos de Seguridad
56	1.2 Antecedentes
57	1.2.1 Grid PRAGMA
62	1.2.2 Colaboración de la UNAM con PRAGMA
63	1.2.3 Problemática
63	1.2.4 Justificación
66	1.2.5 Objetivo del Proyecto
66	1.2.6 Alcances
69	Capítulo 2. Instalación y Configuración del Cluster
69	2.1 Requerimientos y Procedimiento para unirse a PRAGMA
73	2.2 Instalación del Sistema Operativo: Linux Rocks
73	2.2.1 Distribución Rocks
84	2.2.2 Pre-Instalación
86	2.2.3 Instalación
106	2.3 Configuración del Middleware: Globus
107	2.3.1 Globus Toolkit
109	2.3.2 Configuración
121	Capítulo 3. Puesta a punto del Cluster para el Proyecto PRAGMA
121	3.1 Creación de Cuentas de Usuarios en Rocks
124	3.1.1 Comprobación de las Cuentas Creadas
125	3.1.2 Creación de Cuentas para Usuarios PRAGMA
134	3.2 Instalación y Configuración de Aplicaciones
134	3.2.1 Química Computacional
135	3.2.2 Openbabel
144	3.2.3 Gamess
151	3.3 Actualización Automática de Certificados
152	3.3.1 Instalación de VDT
154	3.3.2 Configuración para la Actualización Automática de Certificados
159	Conclusiones
162	Glosario

ÍNDICE DE FIGURAS

Página	
	Capítulo 1
25	Figura 1. Sesión del superusuario
30	Figura 2. Modelo de arquitectura SISD
31	Figura 3. Modelo de arquitectura SIMD
32	Figura 4. Modelo de arquitectura MISD
32	Figura 5. Modelo de arquitectura MIMD
33	Figura 6. Modelo de arquitectura de una computadora paralela con memoria compartida
35	Figura 7. Modelo de arquitectura de una computadora paralela con memoria distribuida
38	Figura 8. Aplicaciones del Supercómputo
39	Figura 9. Kan Balam. Vista frontal de la segunda línea de racks
40	Figura 10. Mixbaal. Vista posterior de los módulos
41	Figura 11. Esquema general de un cluster
42	Figura 12. Componentes de un cluster
53	Figura 13. Ejemplos de grids alrededor del mundo
59	Figura 14. Instituciones que forman parte de la grid de PRAGMA
60	Figura 15. Grupos de trabajo de PRAGMA
	Capítulo 2
71	Figura 1a. Procedimiento para la configuración y puesta a punto del Cluster Mixbaal
72	Figura 1b. Procedimiento para la configuración y puesta a punto del Cluster Mixbaal
85	Figura 2. Esquema de conectividad del cluster
87	Figura 3. Instalación Rocks: Pantalla 1
88	Figura 4. Instalación Rocks: Pantalla 2
89	Figura 5. Instalación Rocks: Pantalla 3
89	Figura 6. Instalación Rocks: Pantalla 4
90	Figura 7. Instalación Rocks: Pantalla 5
90	Figura 8. Instalación Rocks: Pantalla 6
92	Figura 9. Instalación Rocks: Pantalla 7
92	Figura 10. Instalación Rocks: Pantalla 8
93	Figura 11. Instalación Rocks: Pantalla 9
94	Figura 12. Instalación Rocks: Pantalla 10
94	Figura 13. Instalación Rocks: Pantalla 11
95	Figura 14. Instalación Rocks: Pantalla 12
96	Figura 15. Instalación Rocks: Pantalla 13
97	Figura 16. Instalación Rocks: Pantalla 14
98	Figura 17. Instalación Rocks: Pantalla 15
99	Figura 18. Instalación Rocks: Pantalla 16
99	Figura 19. Instalación Rocks: Pantalla 17
100	Figura 20. Instalación Rocks: Pantalla 18
101	Figura 21. Instalación Rocks: Pantalla 19
101	Figura 22. Instalación Rocks: Pantalla 20
108	Figura 23. Grid Security Infraestructure
	Capítulo 3
151	Figura 1. Visualización de resultados de Gamess, con el programa MacMolPlt

ÍNDICE DE TABLAS

Página	
	Capítulo 1
15	Tabla 1. El largo de la contraseña contra el número de contraseñas potenciales
53	Tabla 2. Uso de las grids
65	Tabla 3. Comparativo de equipos de supercómputo en México
	Capítulo 2
84	Tabla 1. Características técnicas de Mixbaal
86	Tabla 2. Rolls disponibles para descargar
95	Tabla 3. Particionado por default Rocks
96	Tabla 4. Particionado del frontend en Mixbaal
	Capítulo 3
142	Tabla 1. Estructura de un archivo en formato pdb

INTRODUCCIÓN



La Universidad Nacional Autónoma de México (UNAM) es una institución que siempre se ha interesado en fomentar la investigación científica en nuestro país mediante el uso de tecnologías de cómputo vanguardistas, apoyándose en la Dirección General de Servicios de Cómputo Académico (DGSCA).

La DGSCA es la dependencia universitaria encargada de proveer, administrar y mantener los recursos de informática, cómputo, Internet y telecomunicaciones que posee la Universidad.

Como herramientas para la investigación científica y tecnológica, la DGSCA proporciona una infraestructura de supercómputo compuesta por equipos de cómputo numérico intensivo, dispositivos de almacenamiento masivo y redes de alta velocidad.

Esta infraestructura es operada por el Departamento de Supercómputo, cuya función principal es proporcionar a la comunidad el servicio de supercómputo, a través de mantener en funcionamiento los equipos y de proporcionar a los usuarios los mecanismos necesarios para su utilización, así como las asesorías y ayuda que requieran.

Como parte de las actividades que se llevan a cabo a fin de promover la investigación científica, el Departamento de Supercómputo participa en diversos proyectos de prospección tecnológica, tales como proyectos de grid.

Grid es una tecnología que permite compartir a gran escala recursos que se encuentran distribuidos geográficamente, en su mayoría equipos de cómputo de alto rendimiento, como clusters y supercomputadoras, haciendo uso de la Internet. Los recursos que se comparten no sólo incluyen datos, sino también capacidad de procesamiento y de almacenamiento. Asimismo, ofrece la funcionalidad para que de forma transparente y segura, sea posible compartir y explotar simultáneamente los recursos pertenecientes a diferentes organizaciones, siempre respetando sus propias políticas, procedimientos de seguridad y gestión de recursos.

El término cluster hace referencia a un conjunto de computadoras que se encuentran interconectadas y configuradas de modo que en conjunto, puedan ser utilizadas como un solo recurso, y el término supercomputadora se define como una computadora con la máxima capacidad de procesamiento, almacenamiento, memoria, velocidad de comunicación y software que existe en la actualidad.

Las asociaciones científicas son las principales entidades que demandan soluciones de este tipo, que les permitan llevar a cabo una amplia y variada gama de investigaciones en todas las áreas, como pueden ser: Astronomía, Física, Química, Ciencias Genéticas, Ciencias del Medio Ambiente, entre otras. Tal es el caso de la Asamblea de la Cuenca del Pacífico para Aplicaciones de Grid y Middleware (PRAGMA).

PRAGMA es una comunidad que surge en 2002 con la filosofía de establecer colaboraciones sólidas entre grupos de científicos ubicados geográficamente a lo largo de toda la Cuenca del Pacífico y promover el uso de aplicaciones basadas en tecnologías grid, proporcionando a través del PRAGMA Grid Testbed, en lo sucesivo “grid de PRAGMA”, una infraestructura y un ambiente de colaboración que permita llevar a cabo pruebas y experimentos relacionados con el cómputo científico, conformada por múltiples clusters que son contribución de sus integrantes. La UNAM pertenece a esta comunidad desde agosto del 2004.

A través del Plan de Becarios 2006, programa de cursos impartidos por la DGSCA, dirigido a los estudiantes de licenciatura de las áreas de ciencias, ingenierías e informática y cuyo objetivo es la formación de recursos humanos altamente especializados en diversas áreas de cómputo científico como es Administración de Sistemas Unix, Optimización, Realidad Virtual y Visualización, fuimos asignadas para proponer una solución que permitiera al Departamento de Supercómputo cumplir con el compromiso adquirido a través de los convenios de colaboración celebrados con PRAGMA de una manera efectiva.

Este compromiso consiste en integrar a la UNAM en la grid de PRAGMA a través de la aportación de un cluster configurado y puesto a punto, con el fin de participar activamente con esta comunidad en la ejecución de sus experimentos.

Para su cumplimiento, se llevó a cabo un análisis en conjunto con el Departamento de Supercómputo de los requerimientos mínimos que establece el Centro de Operación Grid (GOC) de PRAGMA y de los equipos que tiene a su cargo el Departamento, derivado del cual, se resolvió que el cluster denominado Mixbaal sería el destinado para dicho fin, ya que se acopla a los requisitos del GOC y es un equipo que fue adquirido con fines de experimentación.

Por lo anterior, nuestra propuesta consiste en la configuración y puesta a punto del cluster Mixbaal, para ser empleado y dedicado totalmente a pruebas experimentales de cómputo científico, a la vez que sirva como herramienta para integrar al Departamento de Supercómputo y a la Universidad a la grid de PRAGMA.

Mixbaal es un Cluster Intel Pentium III Beowulf, el cual obedece a la siguiente descripción:

Arquitectura:

- Un nodo maestro (frontend)
- 9 nodos esclavos
- Conectados en una topología de estrella

Características de los nodos:

- Dos procesadores Pentium III a 1130.497 MHz
- Memoria Caché de 512 KB

- 1 GB de memoria RAM
- Discos duros SCSI de 18 GB
- Cada nodo posee 1 tarjeta de red Gigabit Ethernet (1000mbps) y 2 tarjetas de red Fast Ethernet (100mbps).
- Poder de cómputo de 1 Gflops

Red:

- Conexión entre los nodos tipo Fast Ethernet a 100mbps, por medio de un switch Extreme Networks Summit.

Ahora bien, el presente trabajo tiene la finalidad de mostrar el conjunto de actividades que se realizaron durante el proceso de instalación, configuración y puesta a punto, mismas que incluyen, a *grosso modo*:

- Instalación del sistema operativo Rocks
- Instalación de middleware Globus Toolkit
- Instalación de aplicaciones de Cómputo Científico
- Creación de cuentas de usuario.

Por otro lado, este documento está conformado por las siguientes secciones:

- Índice general
- Índice de figuras
- Índice de tablas
- Introducción
- Convenciones tipográficas
- Capítulo 1. Marco Teórico y Antecedentes
- Capítulo 2. Instalación y Configuración del Cluster
- Capítulo 3. Puesta a punto del Cluster para el Proyecto PRAGMA
- Conclusiones
- Glosario

La sección “Introducción”, explica de forma general el contenido de este trabajo, asimismo describe cómo está estructurado.

La sección “Convenciones Tipográficas” nos explica los diversos estilos tipográficos empleados para hacer referencia a comandos, salidas de comandos, notas informativas, de advertencia, entre otros.

Por otro lado, en el primer capítulo se exponen los conceptos fundamentales, mismos que sustentan el desarrollo de este proyecto. Estos conceptos tienen la finalidad de iniciar el proceso de inmersión del lector, familiarizándolo con la terminología empleada. Asimismo se enuncia la problemática, justificación, objetivos y alcances de este trabajo.

El orden y los temas que se abordan son:

- Nociones de la Administración de Sistemas, lo cual permite comprender el quehacer diario de un administrador de sistemas, las actividades que realiza, que van desde la puesta en marcha del sistema, su configuración y mantenimiento, así como la definición y aplicación de políticas para el control de acceso y el control de los recursos, hasta la supervisión y monitoreo del sistema. Estas actividades en su conjunto, están enfocadas a mantener la continuidad operacional del sistema que se gestiona.
- Conceptos propios del cómputo intensivo, como son cómputo paralelo y distribuido, supercómputo, cluster, grid y mecanismos de seguridad.
- Planteamiento de la problemática, justificación, objetivos y alcances de este proyecto, el cual surge con motivo de la colaboración de la UNAM con la comunidad de PRAGMA.

El capítulo segundo presenta las actividades iniciales de instalación y configuración del cluster que se llevaron a cabo para cumplir parte de los requisitos establecidos por el Centro de Operación Grid (GOC) de PRAGMA, que incluyen:

- Proveer un sistema con al menos dos computadoras, una utilizada como frontend y otra como nodo de cómputo. Para nuestro caso, el Departamento de Supercómputo facilitó diez nodos, de los cuales uno funge como frontend y los nueve restantes como nodos de cómputo.
- Instalar como sistema operativo una distribución de UNIX, se sugiere utilizar la distribución Rocks, mismo que fue instalado.
- Instalación y configuración del middleware Globus.
- Instalación y configuración de una Autoridad Certificadora para pruebas.

En el tercer y último capítulo se detalla sobre las tareas que se ejecutaron para concluir con la puesta a punto del cluster, a saber:

- Creación de cuentas de usuario.
- Instalación y configuración de las aplicaciones requeridas por el GOC.
- Verificación del correcto funcionamiento de las aplicaciones.
- Configuración de actualizaciones automáticas de certificados.

Posterior a los capítulos, se encuentra el apartado “Conclusiones” donde se presentan los resultados obtenidos tras la culminación de este proyecto.

La última sección es el “Glosario”, la cual nos provee de una lista de palabras de carácter técnico con su respectiva definición, ordenadas alfabéticamente, facilitando al lector la comprensión del contenido del texto.

CONVENCIONES TIPOGRÁFICAS



Esta sección nos explica el conjunto de convenciones tipográficas empleadas en el presente trabajo, que nos sirven para hacer referencia a comandos y sus salidas en pantalla, notas informativas, notas de advertencia, entre otros, las cuales se identifican de acuerdo a lo siguiente:

Convención 1: Ejecución y salida de comandos

```
#comando1  
Salida del comando1
```

Explicación: En un recuadro con borde negro, fondo gris y texto con fuente Courier, se indica el nombre del comando seguido del símbolo del sistema, cuando es # el comando debe ser ejecutado por el usuario administrador (root) y cuando corresponde a \$, el comando debe ser ejecutado por un usuario distinto del administrador. En las subsiguientes líneas se desplegará la salida de dicho comando.

Ejemplo:

```
[root@mixbaal ~]# ls /export  
apps bio home lost+found root site-roll
```

Convención 2: Nota de Advertencia



Información de la nota de advertencia.

Explicación: Se presenta esta imagen, seguida de un texto explicativo, cuando se indica una advertencia sobre el tema tratado, a fin de informar al lector que lo expuesto puede representar un riesgo.

Ejemplo:



Se debe prestar atención, ya que si no se ingresa rápidamente la palabra `frontend`, el programa de instalación asumirá que se trata de un nodo cualquiera, fallando la instalación del frontend, obligando a reiniciar el nodo.

Convención 3: Nota Informativa



Información de la nota.

Explicación: Se presenta esta imagen, seguida de un texto explicativo, cuando se indica con énfasis algún dato adicional sobre el tema tratado.

Ejemplo:



Ningún nodo cuenta con unidad de CD/DVD-ROM, de modo que se consiguió una unidad de CD/DVD-ROM extraíble USB.

Convención 4: Nota de Problema



Información de la nota de problema.

Explicación: Se presenta esta imagen alineada a la izquierda, seguida de un texto explicativo, cuando se indica que se ha presentado algún problema sobre el tema tratado.

Ejemplo:



No se obtuvo una dirección IP de la red a la que está conectada la interfaz eth1, presentándose el escenario 1, mencionado en la sección 2.2.2.

Convención 5: Nota de Solución



Información de la nota de solución.

Explicación: Se presenta esta imagen, seguida de un texto explicativo, cuando derivado de algún problema presentado sobre el tema tratado, se ha encontrado una solución, misma que es expuesta.

Ejemplo:



Se ingresó al BIOS del frontend para modificar la configuración de arranque, indicándole que iniciará con la unidad de CD-ROM.

Convención 6: Palabras referidas en el Glosario

Palabra*

Explicación: Dentro del texto, las palabras que se encuentran en la sección del Glosario, están indicadas con un asterisco (*) al final de dicha palabra.

Ejemplo:

[...] tales como: evolución de galaxias, modelos magnetohidrodinámicos* [...]

Convención 7: Información del sistema

Texto... Información del sistema... Texto

Explicación: Dentro del contenido del documento, el texto que aparece en fuente Courier, hace referencia a nombres de archivos y directorios, nombres de usuario, comandos y sus argumentos, nombres de botones de las pantallas, nombres de equipos, nombres de procesos e información propia del sistema.

Ejemplo:

4. Una vez verificado el script, se adicionó la ubicación del aplicativo en el PATH, editando el archivo `/etc/profile`, adicionando la línea:
`PATH=$PATH:/share/apps/games_2008`

Convención 8: Información de URL y direcciones de correo electrónico

protocolo://maquina/ruta_del_contenido
usuario@dominio

Explicación: Dentro del texto, las direcciones de páginas web o direcciones de correo electrónico están indicadas en fuente Courier y subrayadas.

Ejemplo:

- E-mail de contacto. Preferentemente el del administrador del cluster.
Contact: yoli@super.unam.mx
- URL de la página a la que se podrá acceder para monitorear el cluster vía WEB. Misma que una vez que se ponga a punto el cluster debe enviarse a la coordinadora de PRAGMA.
URL: http://mixbaal.supercomputo.unam.mx



CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

8	1.1 Marco Teórico
56	1.2 Antecedentes

1.1 Marco Teórico

El presente capítulo expone los conceptos fundamentales aplicados en el desarrollo de este proyecto, a la vez que proporciona una visión general sobre el uso de tecnologías de cómputo intensivo. Estos conceptos permitirán al lector familiarizarse con la terminología empleada en éste y en los subsiguientes capítulos, asimismo la sección “Glosario” aporta una serie adicional de conceptos que pretenden complementar el seguimiento de los temas tratados.

Este proceso de inmersión inicia con nociones de la administración de sistemas, fundamento principal de este trabajo, a fin de proporcionar una idea clara del proceso de gestión de recursos que se lleva a cabo por parte del administrador, tareas que incluyen la puesta en marcha del sistema, su configuración y mantenimiento, la definición de políticas para el control de acceso y la asignación de recursos, las cuales van de la mano con actividades de supervisión y monitoreo permanente, todas ellas enfocadas en mantener la continuidad operativa del sistema en cuestión.

Los recursos gestionados están en función del tipo de sistema implementado, para nuestro caso de estudio, se trata de un cluster tipo Beowolf, el cual se atañe a conceptos propios del cómputo intensivo, por lo cual se abordan definiciones esenciales de este tema, tales como cómputo paralelo y distribuido, supercómputo, grid y los mecanismos de seguridad que le son inherentes dada su naturaleza, ya que es utilizado en ambientes con múltiples usuarios y que generalmente se encuentran distribuidos geográficamente, lo que hace imprescindible la implantación de mecanismos que garanticen el resguardo de información, sin que ello implique degradación en el rendimiento del sistema.

La conjugación de estos elementos, sustentan la construcción de soluciones basadas en el cómputo intensivo o de alto rendimiento, que ofrecen un mayor número de posibilidades a los usuarios que demandan grandes cantidades de recursos, tanto de cálculo, como de almacenamiento, un ejemplo de este tipo de soluciones, es el cluster Mixbaal, configurado para integrarse a la grid de PRAGMA.

1.1.1 Administración de Sistemas

Es de alto grado de interés hablar de la Administración de Sistemas porque de ésta surge la respuesta al por qué un Licenciado en Informática participa en un proyecto de clusters y grids computacionales.

Dicha respuesta la encontramos en la definición de Administración de Sistemas, la cual nos dice que: “Es la rama de la ingeniería que se encarga de la administración operacional de un sistema de cómputo, involucrando tanto a la tecnología como a los usuarios de la misma.”¹

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Como se puede observar parte de la administración de sistemas consiste en lograr que un sistema computacional se pueda utilizar de manera efectiva. Para nuestro caso el Cluster Mixbaal es el sistema computacional que dejamos listo para su uso.

Ahora bien, el personaje principal en la administración de sistemas es el Administrador, quién es la persona responsable de configurar, mantener y actualizar el sistema o conjunto de sistemas que forman una red, cuidando el funcionamiento del software, hardware y periféricos de forma que estén disponibles para ser utilizados por los usuarios².

Las características que debe cubrir un Administrador de Sistemas son:

- Personalidad:
 - Paciencia
 - Responsable
 - Trabajar bajo presión
 - Aceptar nuevos retos
 - Innovador
 - Espíritu de colaboración
 - Honesto, íntegro, confiable
 - Respetuoso, imparcial, tolerante

- Conocimientos:
 - Técnicas de programación
 - Dominio de al menos un lenguaje de programación
 - Funcionamiento del sistema operativo
 - Técnicas de administración del sistema operativo
 - Conocimientos básicos de hardware y mantenimiento de dispositivos

- Actitud para:
 - Planear actividades (organizado)
 - Respalda (precauido)

- Aptitudes:
 - Analista
 - Tenaz
 - Ingenioso
 - Capaz de tomar decisiones y afrontar consecuencias
 - Intuitivo

Adicionalmente, un Administrador de Sistemas debe establecer políticas de uso y administración de los equipos. Éstas involucran un esquema de respuesta ante posibles eventos y determinan un punto de equilibrio en el sistema.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Algunos tipos de políticas son:

- Apertura de cuentas
- Asignación de nombre de usuario
- Manejo de contraseñas
- Control de acceso: seguridad del sistema, horas de mantenimiento, conexiones remotas, etc.
- Asignación de recursos: cuotas de disco duro
- Responsabilidad de los respaldos
- Borrado de archivos temporales

Administración de Usuarios

La administración de usuarios es una actividad realizada por el Administrador de Sistemas, en la que se define el acceso y los privilegios a los recursos que serán asignados a cada usuario del sistema³.

Una cuenta de usuario permite identificar a una entidad que accede al sistema, asimismo verificar los permisos que tiene asignados.

Al decir recursos nos referimos a archivos, directorios y dispositivos. El control de acceso a éstos, forma en gran medida la rutina diaria de un administrador de sistemas, controlado generalmente por "grupos". Los grupos son construcciones lógicas que permiten relacionar a usuarios para un propósito en común. Por ejemplo, si una organización tiene varios administradores de sistemas, todos ellos se pueden colocar en un grupo administrador de sistema. Posteriormente se le pueden otorgar permisos al grupo para acceder a recursos claves del sistema. De esta forma, los grupos pueden ser una herramienta eficiente para la administración de recursos y su acceso.

Las cuentas de usuario tienen diferentes componentes:

- Nombre de usuario: El nombre de usuario es la respuesta a la pregunta "¿quién eres?", por tanto, deben cumplir con un requisito en particular: deben ser únicos. Por ello es importante que se defina una política para la asignación de nombres de usuario.
- Contraseña: Una contraseña es una forma de autenticación que utiliza información secreta para controlar el acceso a un recurso. Si el nombre de usuario responde a la pregunta "¿quién eres?", la contraseña es la respuesta a otra, consecuente de la primera: "¡ demuéstalo!" Las contraseñas proporcionan una forma de probar la autenticidad de la persona que dice ser el usuario con el cual intenta ingresar al sistema. Este esquema, ha resultado muy efectivo, por más de una razón:

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

confidencialidad de la contraseña, resistencia de la contraseña a ser adivinada, la resistencia ante un ataque de fuerza bruta, entre otras.

- Información de control de acceso: Además del nombre de usuario y contraseña asignados, las cuentas de usuario también contienen información sobre el acceso. Esta información es variable en función del sistema operativo utilizado, misma que generalmente incluye:
 - Información específica del usuario en el sistema
 - Información específica del grupo en el sistema
 - Información sobre los privilegios adicionales de cada usuario (grupo en los cuales el usuario es miembro)
 - Información del acceso por defecto de cada usuario
 - Información sobre atributos a aplicar por defecto en todos los archivos y recursos creados por el usuario

Por lo anterior, en la administración del sistema, resulta indispensable definir políticas que establezcan el acceso y la asignación de los recursos.

Políticas para la asignación de nombres de usuario

Precisar un convenio de nombres para los usuarios, es una forma de facilitar las actividades del administrador del sistema a la hora de que se requiere dar de alta un nuevo usuario. Para definir esta política deben tomarse en cuenta diversos factores, como son: el tamaño, la estructura y naturaleza de la organización⁴.

El tamaño de la organización es importante, porque nos indica cuántos usuarios puede soportar la convención que se haya definido para nombres de usuario. Por ejemplo, una organización muy pequeña quizás pueda permitir que todo el mundo utilice su primer nombre. Para un organismo mucho más grande, este convenio no funciona.

La estructura de la entidad también puede tener influencia sobre el convenio de nombres más apropiado. Para las que cuenten con una estructura bien definida puede ser adecuado incluir elementos de ésta en la convención de nombres. Por ejemplo, puede incluir los códigos de los departamentos o área de adscripción, como parte del nombre de usuario.

La naturaleza de la organización también puede significar que algunas convenciones sean más apropiadas que otras. Supongamos una organización que maneja datos confidenciales, en este caso, puede decidirse por una convención que no indique ningún tipo de información personal que proporcione elementos para vincular al individuo con su nombre. Por ejemplo, el nombre del usuario Juan Pérez podría ser ZAPYP2K9.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Ejemplo de una política para la asignación de nombres en una organización es:

- Primeras dos letras del primer apellido, para Martínez sería MA
- Primera letra del segundo apellido, para Palacios sería P
- Primera inicial del nombre, para Patricia sería P
- Año de nacimiento a 2 dígitos, para 1984 sería 84
- Mes de nacimiento a 1 dígito o letra, para Enero sería 1, para Octubre, Noviembre y Diciembre serían A, B y C respectivamente.
- Día de nacimiento, donde cada día hasta el 09 es ese mismo, pero a partir del 10, le corresponde una letra, empezando por la A:
01 = 1
02 = 2
...
10 = A
11 = B... etc., hasta 31 = V

De acuerdo con esta política, el nombre de usuario para Paulina Martínez Portillo, cuya fecha de nacimiento es 31 de enero de 1984 sería: MAPP841V y para el usuario Perla Palacios Morales, con fecha de nacimiento del 03 de junio de 1984 sería PAMP8463.



Es importante tener en cuenta que si la convención de nombres incluye añadir datos diferentes para formar un nombre de usuario, existe el potencial de que el resultado sea gracioso u ofensivo. Por lo tanto, aún si crea los nombres de usuario de forma automática, es recomendable que lleve a cabo alguna forma de revisión.



Es importante considerar que con la convención de nombres descrita aquí existe la posibilidad de que dos individuos, obtengan el mismo nombre de usuario. Esto se conoce como una colisión.

Manejo de colisiones

Las colisiones generalmente resultan inevitables, por lo que no se deben omitir y minimizar al momento de definir una política para la asignación de nombres de usuario, por lo que pueden precisarse varias formas para contrarrestarlas, por ejemplo:

- Añadiendo números en secuencia al final del nombre de usuario en colisión (MAPP841V1, MAPP841V2, etc.).
- Añadiendo información adicional de la organización al nombre de usuario (PAMP8463DEP1, PAMP8463DEP2, etc.)

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Manejo de cambios de nombres de usuario

Cuando una organización utiliza una convención de nombres basada en el nombre real de cada usuario, es de esperarse que de vez en cuando tenga que enfrentarse a cambios de nombres. Las razones pueden variar desde un usuario que no le gusta su nombre de usuario hasta un empleado con más jerarquía en la organización que prefiere un nombre de usuario "más acorde".

No importa cuál sea la razón, hay muchos aspectos a tener en mente cuando se cambie un nombre de usuario:

- Efectuar el cambio en todos los sistemas afectados
- Mantener constante toda la información relacionada con el usuario
- Cambiar la propiedad de todos los archivos y otros recursos específicos del usuario (en caso de ser necesario)
- Manejar los problemas relacionados con el correo electrónico

Antes que todo, es importante asegurarse de que el nuevo nombre de usuario es propagado a todos los sistemas donde se utilizaba el nombre original. De lo contrario, cualquier función del sistema operativo que esté vinculado con el nombre de usuario, funcionará en algunos sistemas y en otros no.

Por ejemplo, los problemas derivados de cambios de nombres de usuario en el correo electrónico tienen múltiples dimensiones. En su forma más básica, un cambio de nombre de usuario implica que la gente ya no conoce el nombre de usuario correcto de esa persona. La tarea de notificar a la organización del cambio del nombre, generalmente no representa un conflicto, pero sí lo es cuando se tiene que hacer del conocimiento de externos. Quizá la mejor alternativa es la de crear un alias para el correo electrónico de manera que todo el correo enviado al viejo nombre de usuario sea redirigido al nuevo nombre. Se le puede indicar al usuario que debe alertar a todas las personas que su nombre de usuario ha sido modificado. A medida que el tiempo pasa, menos mensajes serán entregados usando el alias y llegará el momento en el que podrá ser eliminado.

Políticas para el manejo de contraseñas

Los administradores de sistemas, al crear las contraseñas, deben tomar en cuenta que éstas son confidenciales y deben ser difíciles de adivinar.

Las contraseñas que cumplen con estos requisitos se les conocen como contraseñas robustas, mientras que aquellas que no, se les llama débiles⁵. Es importante para la seguridad de la organización crear contraseñas robustas, mientras más robustas sean las contraseñas, existe menos posibilidad de que éstas sean comprometidas. Los administradores de sistemas pueden escoger entre dos opciones:

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Crear contraseñas para todos los usuarios.
- Permitir que los usuarios creen sus propias contraseñas.

Al crear contraseñas para todos los usuarios asegura que éstas sean robustas, pero se vuelve una tarea pesada a medida que crece la organización. También incrementa el riesgo de que los usuarios escriban sus contraseñas. Por estas razones, la mayoría de los administradores de sistemas prefieren dejar que los usuarios mismos creen sus contraseñas. Sin embargo, un buen administrador se encargará de verificar que las contraseñas creadas sean lo suficientemente robustas.

La necesidad de mantener secretas las contraseñas debe ser una parte arraigada en la mente de un administrador de sistemas. Sin embargo, este punto se pierde con frecuencia en la mayoría de los usuarios. Dado este hecho, es de vital importancia proporcionar educación a los usuarios en este sentido, para que así estos puedan entender que sus contraseñas deben mantenerse en secreto. Las contraseñas deben ser tan difíciles de adivinar como sea posible. Una contraseña robusta es aquella que un atacante no podrá adivinar, aún si el atacante conoce bien o posee información del usuario.

Un ataque de fuerza bruta sobre una contraseña implica el intento metódico (usualmente a través de un programa conocido como password-cracker) de cada combinación de caracteres posible con la esperanza de que se encontrará la contraseña correcta. Una contraseña robusta se debería construir de manera tal que el número de contraseñas potenciales a probar sea muy grande, forzando al atacante a tomarse un largo tiempo buscando la contraseña.

Contraseñas débiles

Una contraseña débil⁶ es la que no supera alguna de estas tres pruebas:

- Es secreta
- Es resistente a ser adivinada
- Es resistente a un ataque de fuerza bruta

Las contraseñas pueden ser débiles, si presentan alguna de las siguientes características:

- Contraseñas cortas.

Una contraseña corta es débil porque es mucho más susceptible a un ataque de fuerza bruta. Para ilustrar esto, consideremos la tabla siguiente, en el que se muestran el número de contraseñas potenciales que deben ser evaluadas ante un ataque de fuerza bruta. (Se asume que las contraseñas consisten solamente de letras en minúsculas.)

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Largo de la contraseña	Contraseñas potenciales
1	26
2	676
3	17,576
4	456,976
5	11,881,376
6	308,915,776

Tabla 1. El largo de la contraseña contra el número de contraseñas potenciales

Podemos observar en la Tabla 1, que el número de contraseñas posibles incrementa dramáticamente a medida que se incrementa el largo.



Nota: Aún cuando esta tabla termina en seis caracteres, no significa que las contraseñas de seis caracteres de longitud son suficientemente largas para una buena seguridad. En general, mientras más larga la contraseña mejor.

- Conjunto de caracteres limitado.

El número de caracteres diferentes que comprenden una contraseña tiene un gran impacto en la habilidad de un atacante de conducir un ataque de fuerza bruta. Por ejemplo, en vez de 26 caracteres diferentes que se pueden utilizar en una contraseña de solamente minúsculas, se complicaría si se agregan números. Esto significa que cada caracter en una contraseña es uno de 36 en vez de uno de 26. En el caso de contraseñas de seis caracteres de largo, esto representa un incremento de contraseñas posibles de 308,915,776 a 2,176,782,336.

Si se utilizan contraseñas alfanuméricas con mayúsculas y minúsculas (para aquellos sistemas operativos que lo soporten), el número posible de contraseñas de seis caracteres se incrementa a 56,800,235,584. Y si se añaden otros caracteres (tales como símbolos de puntuación) aumenta aún más los números posibles, haciendo un ataque de fuerza bruta mucho más difícil.

Sin embargo, un punto a tener en consideración es que no todos los ataques contra una contraseña son de fuerza bruta. A continuación se describen otros atributos que pueden hacer una contraseña débil.

- Palabras reconocibles

Muchos ataques contra contraseñas están basados en el hecho de que la gente generalmente se siente más cómoda con contraseñas que pueden recordar. Y

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

para la mayoría de la gente, las contraseñas más fáciles de recordar son las que contienen palabras de nuestro lenguaje. Por lo tanto, muchos ataques a contraseñas están basados en el diccionario. Es decir, el atacante utiliza diccionarios de palabras en un intento de encontrar la palabra o palabras que forman la contraseña.



Muchos programas de ataque a contraseñas basados en diccionarios, utilizan léxico en diferentes idiomas. Por lo tanto, no es correcto suponer que la contraseña es más robusta simplemente porque está utilizando una palabra que no es en español.

- Información personal

Las contraseñas que contienen información personal, como el nombre o fecha de nacimiento de un ser querido, una mascota o un número de identificación personal, puede que no sean encontradas a través de un ataque basado en contraseñas de diccionario. Sin embargo, si el atacante conoce al usuario personalmente o está lo suficientemente motivado para investigar su vida personal, puede ser capaz de adivinar su contraseña con poca o ninguna dificultad.

- Trucos simples de palabras

Usando cualquiera de la información discutida anteriormente como la base para una contraseña, pero invirtiendo el orden de las letras, tampoco hace una contraseña débil en una robusta. La mayoría de los descifradores de contraseñas hacen estos trucos en todas las contraseñas posibles. Esto incluye sustituir ciertas letras por números en palabras comunes. He aquí algunos ejemplos:

- 53r610
- 54lt4m0nt35

- La misma contraseña para todos los sistemas

Aún si una contraseña es robusta, no es una recomendable utilizar la misma contraseña en más de un sistema. Obviamente se puede hacer muy poco si los sistemas son configurados para utilizar un servidor central de algún tipo, pero en cualquier otro caso, se sugiere utilizar contraseñas diferentes para sistemas diferentes.

- Contraseñas en papel

Una contraseña que es considerada como robusta puede arruinarse y convertirse en una débil en el momento en el que se escribe. Al colocar las contraseñas en

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

papel, el problema ahora es de seguridad física, ya que la preocupación se centra en mantener seguro ese pedazo de papel.

Sin embargo, algunas organizaciones tienen la necesidad legítima de escribir las contraseñas, incluso puede ser parte de un procedimiento para recuperarse de la pérdida de un empleado clave, como puede ser un administrador de sistemas. En estos casos, el papel conteniendo las contraseñas es almacenado en una ubicación físicamente segura que requiere la autorización de varias personas para tener acceso al papel. Usualmente se utilizan cajas de seguridad o un sitio restringido.

Lamentablemente las contraseñas escritas que no forman parte de un plan de recuperación y no son almacenadas en una bóveda o algo similar, son las contraseñas de los usuarios comunes quienes las almacenan en sitios como:

- Cajones del escritorio
- Bajo el teclado
- La cartera
- Pegada al lado del monitor

Ninguna de estas ubicaciones son lugares apropiados para una contraseña escrita.

Contraseñas robustas

A continuación se describirán las contraseñas que pueden caracterizarse como robustas:

- Contraseñas largas

Mientras más larga es una contraseña, menor es la probabilidad de que tenga éxito un ataque de fuerza bruta. Por tanto, si el sistema operativo lo soporta, se recomienda establecer un largo mínimo para las contraseñas de los usuarios.

- Conjunto de caracteres expandido

Es altamente recomendable que en la definición de la política de manejo de contraseñas, se puntualice que combinen el uso de mayúsculas y minúsculas así como la adición de un carácter no-alfanumérico, por ejemplo:

- n3N1t15%
- k.0n53nt1D1

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Memorizables

Una contraseña es robusta solamente si se puede recordar. Sin embargo, usualmente el ser fácil de memorizar y fácil de adivinar van de la mano. Un consejo sobre la creación de contraseñas fáciles de recordar pero que no sean fáciles de adivinar es, por ejemplo, tomar una frase o dicho favorito y utilizar las primeras letras de cada palabra como el punto de comienzo para la creación de la nueva contraseña. El resultado es fácil de memorizar (pues la frase en la cual está basado es, en sí misma, recordable), sin embargo el resultado no corresponde a ninguna palabra, como ejemplo tomaremos la siguiente frase célebre de Aristóteles:

- 1) Frase fácil de recordar: *“El sabio no dice todo lo que piensa, pero siempre piensa todo lo que dice”*
- 2) Tomar la primera letra de cada palabra: **Esndtlqppsptlqd**, ésta es nuestra contraseña.

La contraseña que se obtiene como resultado, es larga, no es fácil de adivinar y no es una palabra de diccionario, lo que proporciona un alto nivel de robustez.

- Caducidad de las contraseñas

Se recomienda ampliamente que sean establecidos períodos de vigencia para las contraseñas. La caducidad de las contraseñas es una funcionalidad, disponible en muchos sistemas operativos, que coloca límites en el tiempo para que una contraseña dada sea considerada válida. Al final del tiempo de vida de la contraseña, se le pide al usuario que introduzca una nueva contraseña, que se puede utilizar hasta que, igualmente, expire.

Hay dos problemas totalmente opuestos con respecto al tiempo de vida de las contraseñas:

- Conveniencia del usuario
- Seguridad

Por un lado, un tiempo de vida de una contraseña de 50 años presentará muy pocos problemas (si es que llega a presentar alguno), sin embargo, proporcionará muy poco en términos de mejorar la seguridad.

En el otro extremo, un tiempo de vida de una contraseña de 50 minutos será un gran inconveniente para los usuarios, pero la seguridad mejorará en extremo.

La idea es encontrar un balance entre la conveniencia para sus usuarios y la necesidad de seguridad de cada organización. Las organizaciones, comúnmente

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

establecen tiempos de vida para las contraseñas dentro del rango de semanas o un par de meses.

Políticas del control de acceso

Las cuentas de usuario, además del nombre de usuario y contraseña, contienen la información sobre el acceso a los recursos y aplicaciones, controlados a través del manejo de perfiles de usuarios en los cuales se establecen los privilegios de cada uno.⁷

Dependiendo del tipo de organización es la frecuencia con la que se requiere modificar esta información, por ejemplo, en estaciones de trabajo personales y sistemas independientes, puede que no sea tan necesario, en cambio, cuando se trata de organizaciones que hacen uso extensivo de los recursos compartidos a lo largo de la red entre diferentes grupos de usuarios, demandan que la información de control de acceso se modifique con frecuencia.

Mantener apropiadamente la información de control de acceso de los usuarios, depende en gran medida de los diferentes perfiles que la organización requiera, lo que a su vez, impacta en la carga de trabajo del administrador.

Es aconsejable documentar los pasos exactos involucrados para la creación y configuración correcta de una cuenta de usuario y con mayor razón si los perfiles son variados: administrador de sistemas, usuario local, usuario de contabilidad, usuario de comercio exterior, etc.

Cambios en la organización

Una vez más, citando a Aristóteles, con otra de sus frases célebres, “lo único constante es el cambio”, los administradores de sistemas deben tener muy en cuenta que todo cambia, todo el tiempo.

En una organización los cambios que se pueden presentar son: la llegada de nuevos usuarios, usuarios actuales que finalizan sus funciones, o bien, que cambian de responsabilidades, despido de usuarios, etc. De modo que el administrador debe ser capaz de responder a dichos cambios, identificando los recursos que se verán afectados así como llevar a cabo las modificaciones correspondientes, logrando que la operación no pierda su continuidad.

El alta y baja de cuentas de usuario, son procesos que se ejecutan en directo. Por otro lado cuando el usuario cambia de responsabilidades dentro de la organización es probable que solo se deba actualizar su cuenta o bien mantenerla sin variación.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Se pueden identificar al menos tres personas relacionadas en asegurarse de que la cuenta del usuario sea configurada adecuadamente para coincidir con las nuevas responsabilidades:

- El administrador de sistemas
- El responsable original del usuario
- El nuevo responsable del usuario

Entre los tres debe ser posible determinar qué se debe hacer para concluir limpiamente las responsabilidades anteriores del usuario y qué se debe hacer para preparar la cuenta para las nuevas. En muchos casos, este proceso se puede pensar como el equivalente de dar de baja a un usuario existente y crearle una nueva cuenta.

Sin embargo, es más probable que se mantenga la cuenta del usuario y que se modifique para adaptarse a las nuevas actividades. Este enfoque implica que se debe revisar cuidadosamente la cuenta para asegurarse de que se tiene solamente los recursos y privilegios adecuados para las nuevas responsabilidades de la persona.

La situación se complica aún más, cuando existe un período de transición donde la persona realiza tareas relacionadas a ambos grupos. Es aquí donde los responsables, el original y el nuevo, pueden apoyar al administrador de sistemas, otorgando una ventana de tiempo para este período de transición

Nuevos usuarios

Cuando una persona ingresa a una organización, normalmente se les da acceso a varios recursos, dependiendo de sus funciones. El administrador del sistema, como parte de sus responsabilidades, debe verificar que esto se haga rápidamente y de la forma adecuada.

El primer paso es confirmar la llegada de la nueva persona. Esto se maneja de diferentes formas en la organización. He aquí algunas posibilidades:

- Crear un procedimiento donde el departamento de personal de la organización notifique cuando llega una nueva persona.
- Crear un mecanismo en el que el responsable de la nueva persona autorice la creación de una cuenta de usuario.

Las organizaciones tienen enfoques diferentes. Es vital que sea establecido un procedimiento confiable que pueda alertar de cualquier trabajo relacionado a las cuentas que se necesite crear o modificar.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Bajas de usuario

Inevitablemente habrá personas que dejen la organización. Algunas veces esto puede ser bajo circunstancias apropiadas y otras quizás no. En cualquier caso, es necesario que se le informe de la situación para que así pueda tomar las acciones adecuadas.

Como mínimo, las acciones apropiadas deben incluir:

- Inhabilitar el acceso del usuario a todos los sistemas y recursos relacionados, para ello, usualmente mediante el cambio o bloqueo de la contraseña.
- Hacer una copia de seguridad de los archivos del usuario, en caso de que contengan algo que se pueda necesitar en un futuro.
- Coordinar el acceso a los archivos del usuario para el responsable de ese usuario.

La principal prioridad es asegurar sus sistemas contra un usuario que ha dejado de trabajar con la organización recientemente. Esto adquiere mayor relevancia cuando el usuario finalizó sus funciones bajo circunstancias que pueden haberlo dejado con un poco de desagrado hacia la organización. Sin embargo, aún si las circunstancias no son graves, se sugiere desactivar rápidamente el acceso a la persona que ya no formará parte de la organización.

De lo anterior, surge la necesidad de un proceso que lo alerte sobre las terminaciones, preferentemente antes de que estas sean efectivas.



Cuando se manejan los "bloqueos" en respuesta a un término de funciones, es de suma importancia que las cosas se hagan en el momento correcto. Si el bloqueo ocurre después de completarse el proceso de liquidación, existe un gran riesgo de un acceso no autorizado de la persona. Por otro lado, si el bloqueo ocurre antes de que se inicie el proceso de término de funciones, esto puede alertar a la persona sobre el despido inminente y hacer el proceso más difícil para ambas partes.

Una vez desactivado el acceso, es el momento para hacer una copia de seguridad de los archivos de esta persona. Este respaldo puede ser parte de los respaldos estándares de su organización, o puede ser un procedimiento dedicado a hacer copias de seguridad de viejas cuentas de usuario. Aspectos tales como regulaciones sobre la retención de datos, guardar evidencia en casos de demandas por liquidaciones erróneas y otras parecidas, todas forman parte para la determinación de la forma más apropiada de manejar las copias de seguridad.

Asimismo, debe definirse el tipo de acceso que requiere el responsable de la persona que recientemente finalizó sus funciones, a los archivos de ésta. Dependiendo de la organización y la naturaleza de las responsabilidades de la persona, es posible que no se requiera ningún acceso, o bien, que se necesite acceso completo.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Administración de Recursos

Así como es de vital importancia definir las políticas para la creación y mantenimiento de las cuentas de usuario, también lo son las políticas para la asignación de los recursos⁸.

A continuación se describen tres políticas a considerar:

- Política para el acceso de los usuarios a los recursos compartidos
- Política para el acceso desde las diferentes ubicaciones de los usuarios
- Política para establecer barreras contra el abuso de los recursos

Política para el acceso de los usuarios a los recursos compartidos

El acceso de un usuario a una aplicación, archivo o directorio es determinado por los permisos asignados a éstos. Es altamente recomendable aplicar diferentes permisos para los diferentes tipos de usuarios.

Por ejemplo, en ambientes en los que el almacenamiento es compartido, el acceso debe estar controlado para que un usuario solo pueda agregar/modificar/eliminar los archivos de los cuales es propietario. Y así evitar las acciones maliciosas de usuarios malintencionados.

Otro ejemplo es el acceso asignado al directorio principal de un usuario. Solamente el propietario del directorio principal debería poder crear y ver los archivos que se encuentran allí. Se debería negar el acceso a todos los otros usuarios (a menos que el propietario desee lo contrario). Esto incrementa la privacidad del usuario y previene de la posible apropiación incorrecta de archivos personales.

Sin embargo, existen numerosas situaciones en las que múltiples usuarios pueden requerir acceso al mismo conjunto de recursos en un mismo sistema. En este caso, puede ser necesaria la creación de grupos compartidos.

Grupos y datos compartidos

Como se mencionó en la sección 1.1.1, apartado *Administración de usuarios*, los grupos son construcciones lógicas que permiten vincular cuentas de usuario para un propósito específico. Cuando se administran grupos dentro de una organización, es prudente identificar los datos a los que ciertos grupos deben tener acceso, así como los datos que deben ser negados a otros y qué datos deben ser compartidos por todos.

Una vez identificado esto, es posible la creación de una estructura de grupos adecuada, junto con los permisos apropiados para los datos compartidos.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Las interrogantes básicas a considerar al momento de definir los grupos que serán creados, son:

- ¿Qué grupos crear?
- ¿A quién colocar en un grupo determinado?
- ¿Qué tipo de permisos deberían tener estos recursos compartidos?

Para estas preguntas se necesita un enfoque con sentido común. Una posibilidad es reflejar la estructura de la organización cuando se crean grupos.

Por ejemplo, si hay un área de mercadotecnia, sería conveniente crear un grupo llamado "mercadotecnia" y todo el personal de esa área podría incluirse en el grupo, y si la información es requerida por otra área, por ejemplo, "diseño web" a fin de que la publicidad en la web cumpla con las políticas de imagen, logos, etc., entonces se pueden otorgar a éstos, permisos a nivel de grupo para el acceso a los directorios y los datos utilizados por el departamento de mercadotecnia añadiéndolos al grupo.

Sin embargo, se sugiere ser precavido cuando se otorguen permisos para los usuarios, quizá sólo algunos directorios son los que deben ser compartidos y no todos.

De esta forma, existen menos probabilidades de que los datos confidenciales caigan en las manos incorrectas.

Política para el acceso desde las diferentes ubicaciones de los usuarios

Es importante considerar que los usuarios pueden acceder desde diversas ubicaciones a los datos. Cuando se comparten datos entre usuarios, es una práctica común tener un servidor central (o grupo de servidores) que hacen ciertos directorios disponibles a otras máquinas en la red. De esta forma los datos son almacenados en un lugar; no es necesario sincronizar los datos entre múltiples máquinas. Antes de asumir este enfoque, primero debe determinarse cuáles son los sistemas a los que se requiere acceder centralmente.

Al hacer esto, los diferentes sistemas operativos utilizados son clave, ya que el servidor de almacenamiento debe ser capaz de proporcionar sus datos a cada uno de los sistemas operativos utilizados en la organización.

Lamentablemente, una vez que los datos son compartidos entre múltiples sistemas en una red, surge la posibilidad de que se creen conflictos por la propiedad de un archivo.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Problemas globales de propiedad

El tener los datos almacenados centralmente y accesibles por múltiples sistemas sobre la red tiene sus ventajas. No obstante, debe asumirse que cada una de esas computadoras tiene una lista mantenida localmente de las cuentas de usuarios. Si las listas de usuarios en cada uno de estos sistemas no son consistentes con la lista de usuarios en el servidor central, o peor aún, si las listas de usuarios en cada uno de esos sistemas no son ni siquiera consistentes unas con otras, pueden desencadenarse múltiples problemas derivado de ello.

Esto depende del cómo se implementen los usuarios y los permisos de acceso en cada sistema, pero en algunos casos es posible que el usuario A en un sistema pueda ser conocido como B en otro. Esto se vuelve un verdadero problema cuando los datos son compartidos entre sistemas, pues los datos a los que el usuario A tiene permitido acceder desde un sistema también pueden ser leídos por el usuario B desde otro sistema.

Por esta razón, se aconseja utilizar algún tipo de base de datos central de usuarios. Esto asegura que no haya confusión entre las listas de usuarios en sistemas diferentes.

Directorios principales

Existe la posibilidad de que los usuarios tengan directorios principales centralizados en un servidor conectado a la red, cuya ventaja principal es que si un usuario se conecta a cualquier máquina en la red, podrá acceder a sus archivos desde su directorio principal.

La desventaja es que, si la red no se encuentra disponible, los usuarios de la organización no podrán hacer uso de sus archivos.

En algunas situaciones, tales como organizaciones que hacen gran uso de equipos portátiles, el tener directorios principales centralizados no es recomendable, debido a la movilidad de éstos existe la posibilidad de que el equipo no se encuentre dentro de la red de la entidad, dejando inaccesibles sus archivos.

Política para establecer barreras contra el abuso de los recursos

Unas de las tareas más importantes que un administrador de sistemas debe realizar, es la organización cuidadosa y la asignación de permisos para los recursos compartidos, a fin de prevenir el abuso de éstos entre usuarios dentro de la entidad.

De esta forma, aquellos que no deben tener acceso a recursos confidenciales, simplemente se les niega el acceso.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Algo que no debe perderse de vista es que, independientemente de cómo sean establecidas las políticas y que se tenga la certeza de que se ha implementado la más completa,, la mejor guardia contra el abuso de recursos siempre es el monitoreo permanente por parte del administrador del sistema.

Usuarios y Grupos en Sistemas Linux

Los conceptos y políticas detallados en la sección anterior son aplicados de diversas maneras, las cuales varían dependiendo del tipo de sistema utilizado. En los sistemas Linux, existe un usuario que es el administrador del sistema, conocido como *superusuario*, es generalmente, como se vio en la sección 1.1.1, un usuario individual y responsable de mantener el sistema operando de acuerdo con el uso para el que fue destinado y cumpliendo con las políticas establecidas. En Linux, se le denomina usuario *root*⁹.

El administrador del sistema se convierte en el punto de contacto para las peticiones de otros usuarios y tiene la responsabilidad de reflejar esas peticiones en el sistema.

Cuando se ingresa al sistema con la cuenta del superusuario, el intérprete de comandos utilizado (shell*), proporciona un introductor (prompt) especial que indica que el usuario en activo es root, éste símbolo es #. Ver Figura 1.

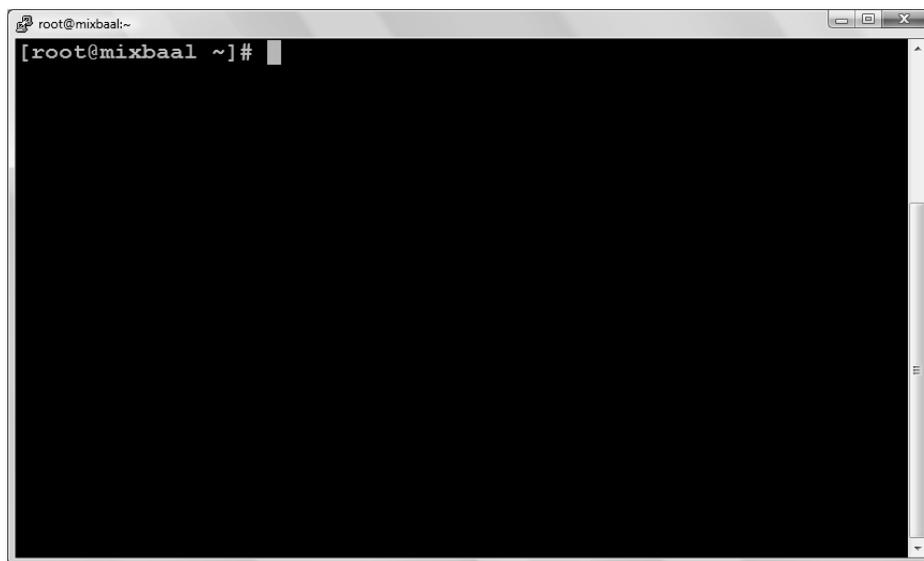


Figura 1. Sesión del superusuario

Es el usuario administrador del sistema quien puede agregar, modificar o bien, eliminar información de las cuentas de usuario y grupos. En los sistemas Linux, la información referente a los usuarios y grupos está almacenada en los siguientes archivos:

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

`/etc/passwd` → Información de los usuarios del sistema
`/etc/group` → Información de los grupos del sistema
`/etc/shadow` → Información de las contraseñas de cada usuario
`/etc/gshadow` → Información de las contraseñas de cada grupo.

Cada línea de estos archivos es separada por el caracter “:”. A continuación se muestra un fragmento del archivo `/etc/passwd`:

```
[root@mixbaal ~]# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
tomcat:x:412:412:Tomcat:/opt/tomcat:/bin/bash
globus:x:401:401:Globus:/opt/globus:/bin/bash
sge:x:400:400:GridEngine:/opt/gridengine:/sbin/nologin
perla:x:500:500:~/home/perla:/bin/bash
zhengc:x:501:501:Cindy Zheng:/export/home/zhengc:/bin/bash
eml:x:502:502:~/export/home/eml:/bin/bash
eccf:x:503:503:~/export/home/eccf:/bin/bash
gfarmfs:x:506:506:Gfarm gfsd:/export/home/ gfarmfs:/bin/bash
```

Cada línea representa un usuario del sistema. Este archivo se conforma de siete campos, separados por : y cada uno contiene la siguiente información:

- Login: cadena para identificar un usuario. Depende de la política para la asignación de nombres de usuario.
- Password: contiene la contraseña cifrada. Actualmente, en la mayoría de los sistemas operativos se encuentra la contraseña en el archivo `/etc/shadow` y en este campo se limita a simbolizarse con el caracter `x`.
- UID: user id, número de identificación del usuario, es un entero mayor a cero, único en el sistema. El user id del usuario `root` es cero. Por convención, del 0 al 100 son para cuentas del sistema, y del 1000 en adelante para cuentas de usuarios.
- GID: group id, número de identificación del grupo principal del usuario, es un entero mayor a cero, único en el sistema.
- GECOS: este campo significa General Electric Comprehensive Operating Supervisor, fue llamado así por los Laboratorios Bell en la implementación original de Unix. Es opcional y se utiliza para ingresar información adicional, generalmente el nombre del usuario.
- Home: ruta absoluta* del directorio al que el usuario entra por omisión y puede escribir archivos debajo de éste, habitualmente es el directorio principal del usuario.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Shell: Es el programa que se ejecuta cada vez que el usuario se conecta al sistema. Usualmente es un intérprete de comandos. Si se escribe un archivo inexistente, el usuario no podrá conectarse.

Por otro lado el archivo `/etc/group` tiene una estructura similar como se muestra en el siguiente fragmento:

```
[root@mixbaal ~]# more /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
nobody:x:99:
users:x:100:
sge:x:400:
perla:x:500:
zhengc:x:501:
eml:x:502:
```

Cada línea representa un grupo del sistema. Este archivo se conforma de cuatro campos, separados por `:` y cada uno contiene la siguiente información:

- Nombre del grupo: cadena para identificar un grupo. Utilizado por varios programas de utilidades como un identificador legible para el grupo.
- Password: contiene la contraseña del grupo cifrada, actualmente, en la mayoría de los sistemas operativos se encuentra la contraseña en el archivo `/etc/gshadow` y en este campo se limita a simbolizarse con el carácter `x`.
- GID: group id, número de identificación del grupo, es un entero mayor a cero, único en el sistema. Lo utilizan el sistema operativo y las aplicaciones para determinar los privilegios de acceso.
- Lista de usuarios: lista delimitada por comas de los usuarios que pertenecen al grupo.

Otro archivo importante es `/etc/shadow` está conformado por nueve campos, igualmente delimitados por `:` como se muestra en el siguiente fragmento:

```
[root@mixbaal ~]# more /etc/group
root:7fWMeWCdbax0Y:13982:0:99999:7:::
bin:*:13982:0:99999:7:::
shd!!!:13982:0:99999:7:::
nfsnobody!!!:13982:0:99999:7:::
globus!!!:13982:0:99999:7:::
sge!!!:13982:::
perla:$1$8.eNkmXM$0f433SCvaT4PW.cRvAgiO1:14088:0:99999:7:::
zhengc!!!:14088:0:99999:7:::
eml:$1$FII5he5XShTPp.KEn/Cr9KUznqBvFU.:14117:0:99999:7:::
```

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Este archivo es accesible únicamente por el usuario `root` y contiene las contraseñas de los usuarios e información adicional sobre las caducidades de éstas. A continuación se describe su estructura:

- Login: cadena para identificar un usuario cuando se conecta al sistema. Esto permite que el sistema, al iniciar la sesión recupere la contraseña del usuario (y la información relacionada).
- Contraseña cifrada: Este campo contiene la contraseña cifrada usando la biblioteca de funciones `crypt()` o el algoritmo de hash* `md5*`. En este campo, los valores diferentes a una contraseña cifrada o en hash*, se utilizan para controlar la conexión del usuario y para mostrar el status de la contraseña. Por ejemplo, si el valor es `!` o `*`, la cuenta es bloqueada y al usuario no se le permite conectarse. Si el valor es `!!`, nunca se ha establecido una contraseña, y el usuario no podrá ingresar al sistema.
- Última fecha en que se cambió la contraseña: Número de días desde el 1 de enero, 1970 (también conocido como epoch), en que la fecha fue modificada. Esta información es utilizada en conjunto con los campos de caducidad de la contraseña.
- Número de días antes de que la contraseña deba ser modificada: Número mínimo de días que deben pasar antes de que la contraseña se pueda cambiar.
- Número de días antes de que se requiera un cambio de contraseña: Cantidad de días que deben transcurrir antes de que el sistema solicite cambiar la contraseña.
- Número de días de advertencia antes de cambiar la contraseña: Cantidad de días antes de que el usuario sea notificado de que la contraseña va a expirar.
- Número de días antes de desactivar la contraseña: Días que transcurren antes de desactivar la cuenta, una vez que la contraseña expiró.
- Fecha desde que la cuenta fue desactivada: La fecha, almacenada como el número de días desde epoch, a partir que la cuenta del usuario ha sido desactivada.
- Reservado: Campo apartado.

Asimismo, el archivo `/etc/gshadow`, es legible solamente por el usuario `root` y contiene las contraseñas cifradas para cada grupo, así como información de sus miembros, a continuación, un ejemplo del archivo:

```
[root@mixbaal ~]# more /etc/gshadow
root:::root
bin:::root,bin,daemon
daemon:::root,bin,daemon
nobody:::
users:::
mysql:x::
haldaemon:x::
```

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

La estructura de este archivo es la siguiente:

- Nombre del grupo: Utilizado por varias herramientas como un identificador legible para el grupo.
- Contraseña cifrada: Si está configurada, los que no sean miembros del grupo pueden unirse a éste escribiendo la contraseña para ese grupo usando el comando `newgrp`. Si el valor de este campo es `!`, entonces ningún usuario tiene acceso al grupo usando el comando `newgrp`. Un valor de `!!` se trata de la misma forma que un valor de `!` sin embargo, también indica que nunca se ha establecido una contraseña para el grupo. Si el valor es nulo, solamente sus miembros pueden acceder a él.
- Administradores del grupo: Los integrantes del grupo listados en este campo, en una lista delimitada por comas, pueden añadir o eliminar miembros usando el comando `gpasswd`.
- Miembros del grupo: Esta lista delimitada por comas, contiene aquellos miembros normales, no administrativos, del grupo.

1.1.2 Arquitecturas de computadoras

En la sección anterior se describieron las diversas actividades que conforman el quehacer del administrador de sistemas. Se mencionó que se encarga de la puesta en marcha del sistema, de su configuración y mantenimiento, así como de la aplicación de políticas para el control de acceso y de recursos, y que todas, en conjunto, permiten llevar a cabo la gestión del sistema. Sin embargo, no se ha profundizado en qué recursos son los que se administran.

Esta sección pretende mostrar un panorama general sobre los diferentes tipos de sistemas de cómputo, iniciando con la definición de arquitecturas de computadoras la cual es un modelo general que describe la forma en cómo trabaja un sistema en particular, los elementos que lo componen, su organización, los recursos que utiliza y cómo hace uso de ellos.

Posteriormente se detalla en los que le confieren al cómputo intensivo: computadoras paralelas, supercomputadoras, clusters y grids, siendo éstos dos últimos el sustento conceptual de este proyecto. Asimismo se describen las ventajas y desventajas que conlleva cada uno de ellos y los mecanismos de seguridad que emplean.

Definición

La arquitectura de computadoras es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para los diferentes componentes de una computadora, con especial interés en la forma en que la unidad central de proceso (CPU) trabaja internamente y accede a las direcciones de memoria.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

También suele definirse como la forma de seleccionar e interconectar componentes de hardware para crear computadoras según los requerimientos de funcionalidad, rendimiento y costo.¹⁰

A continuación se describen dos formas de clasificar los diferentes tipos de arquitecturas de computadoras usadas comúnmente.

Clasificación de Flynn

De acuerdo con Flynn y su clasificación clásica de arquitecturas de computadoras, la cual hace alusión a sistemas con uno o varios procesadores, se basa en el flujo que siguen los datos dentro de la máquina y de las instrucciones sobre estos datos.

Se define como flujo de instrucciones al conjunto de instrucciones secuenciales que son ejecutadas por un único procesador y como flujo de datos al flujo secuencial de datos requeridos por el flujo de instrucciones. Con base en estas consideraciones, Flynn clasifica los sistemas en cuatro categorías:¹¹

SISD (Single Instruction stream, Single Data stream – Una Instrucción, Un dato)

Un procesador. Los sistemas de este tipo se caracterizan por tener un único flujo de instrucciones sobre un único flujo de datos, es decir, se ejecuta una instrucción detrás de otra. Este es el concepto de arquitectura serie de Von Neumann donde, en cualquier momento, sólo se ejecuta una única instrucción. Ver Figura 2.

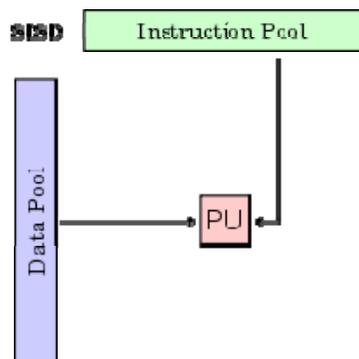


Figura 2. Modelo de arquitectura SISD

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

SIMD (Single Instruction stream, Multiple Instruction stream – Una Instrucción, Múltiples Datos)

Procesadores Vectoriales. Estos sistemas tienen un único flujo de instrucciones que operan sobre múltiples flujos de datos. Ejemplos de estos sistemas los tenemos en las máquinas vectoriales con hardware escalar y vectorial.

El procesamiento es síncrono, la ejecución de las instrucciones sigue siendo secuencial como en el caso anterior, todos los elementos realizan una misma instrucción pero sobre una gran cantidad de datos. Por este motivo existirá concurrencia de operación, es decir, esta clasificación es el origen de la máquina paralela. Ver Figura 3.

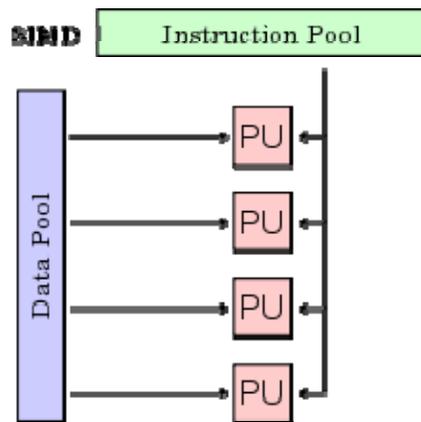


Figura 3. Modelo de arquitectura SIMD

MISD (Multiple Instruction stream, Single Data stream – Múltiples Instrucciones, Un Dato)

Poca implementación. Sistemas con múltiples instrucciones que operan sobre un único flujo de datos. Este tipo de sistemas no ha tenido implementación de forma masiva hasta hace poco tiempo, pero solo con fines experimentales. Los sistemas MISD se contemplan de dos maneras distintas:

- Varias instrucciones operando simultáneamente sobre un único dato.
- Varias instrucciones operando sobre un dato que se va convirtiendo en un resultado que será la entrada para la siguiente etapa. Se trabaja de forma segmentada, todas las unidades de proceso pueden trabajar de forma concurrente. Ver Figura 4.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

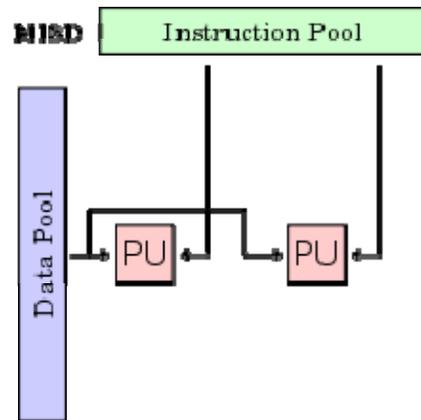


Figura 4. Modelo de arquitectura MISD

MIMD (Multiple Instruction stream, Multiple Data stream – Múltiples Instrucciones, Múltiples Datos)

Cómputo distribuido. Sistemas con un flujo de múltiples instrucciones que operan sobre múltiples datos. Estos sistemas empezaron a utilizarse antes de la década de los 80s. Son sistemas con memoria compartida que permiten ejecutar varios procesos simultáneamente (sistema multiprocesador). Cuando las unidades de proceso reciben datos de una memoria no compartida estos sistemas reciben el nombre de MULTIPLE SISD (MSISD). En arquitecturas con varias unidades de control (MISD Y MIMD), existe otro nivel superior con una unidad de control que se encarga de controlar todas las unidades de control del sistema, ejemplo de estos sistemas son las máquinas paralelas actuales. Ver Figura 5.

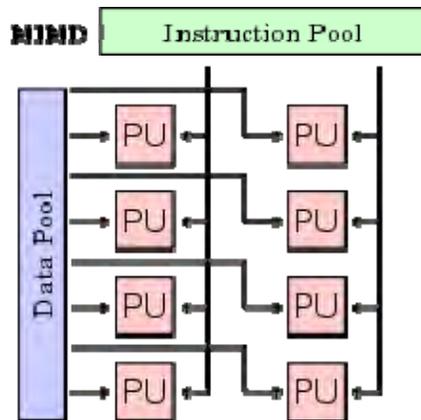


Figura 5. Modelo de arquitectura MIMD

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Categorías de Computadoras Paralelas

Una clasificación moderna, diferente a la de Flynn, hace referencia a los sistemas que tienen más de un procesador, teniendo en cuenta su acoplamiento.

Los Sistemas Fuertemente Acoplados (Sistemas Multiprocesador o Equipo Paralelo de Memoria Compartida)

Son aquellos en los que los procesadores dependen unos de otros. Un sistema multiprocesador puede verse como una computadora paralela conformada por varios procesadores interconectados que comparten un único espacio de direcciones para todos los procesadores y los mecanismos de comunicación se basan en el paso de mensajes.

Dado que los sistemas multiprocesadores comparten los diferentes módulos de memoria, se da el caso de que varios procesadores soliciten el mismo módulo, por lo que se requiere hacer uso de técnicas de semáforos que permitan mantener la integridad de la memoria. Una desventaja de esta arquitectura es que no puede crecer mucho en el número de procesadores interconectados debido a la saturación rápida del bus o del medio de interconexión.

Existe una clasificación de acuerdo a la forma en que los procesadores comparten la memoria:

UMA (Uniform Memory Access)

Sistema multiprocesador con acceso uniforme a memoria. La memoria física es uniformemente compartida por todos los procesadores, esto quiere decir que todos los procesadores tienen el mismo tiempo de acceso a todas las palabras de la memoria*. Cada procesador tiene su propia caché privada y también se comparten los periféricos. Ver Figura 6.

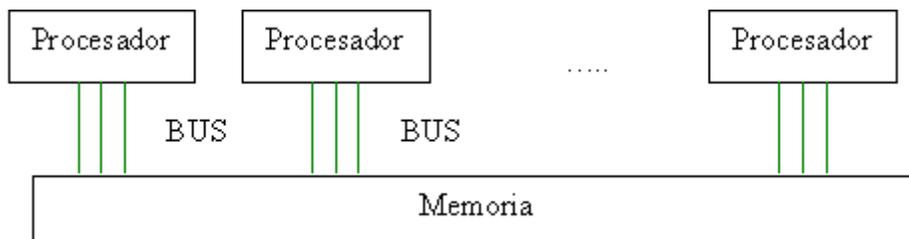


Figura 6. Modelo de arquitectura de una computadora paralela con memoria compartida

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Pipeline

Procesador Vectorial Pipeline. En la actualidad es común encontrar en un solo procesador los denominados Pipeline o Procesador Vectorial Pipeline del tipo MISD, descrita anteriormente. En estos procesadores los vectores fluyen a través de las unidades aritméticas Pipeline. Las unidades constan de una cascada de etapas de procesamiento compuestas de circuitos que efectúan operaciones aritméticas o lógicas sobre el flujo de datos que pasan a través de ellas, las etapas están separadas por registros de alta velocidad usados para guardar resultados intermedios. Así la información que fluye entre las etapas adyacentes está bajo el control de un reloj que se aplica a todos los registros simultáneamente.

Los Sistemas Débilmente Acoplados (Sistemas Multicomputadoras)

Son aquellos en los que existe poca interacción entre los diferentes procesadores que forman el sistema.

Existe una clasificación de acuerdo a la forma en que los procesadores comparten la memoria:

NUMA (Non Uniform Memory Access)

Sistema multiprocesador con acceso no uniforme a memoria. Es un sistema de memoria compartida donde el tiempo de acceso varía según donde se encuentre localizado el dato. Existen memorias locales asociadas a cada procesador y estos tienen prioridad de utilizarla antes que los demás procesadores, lo cual incrementa la velocidad de acceso. Todas las memorias locales conforman la memoria global compartida y físicamente distribuida y accesible por todos los procesadores.

Sistemas Multicomputadoras o Equipo Paralelo de Memoria Distribuida

Los sistemas multicomputadoras se pueden ver como una computadora paralela en el cual cada procesador tiene su propia memoria local. En estos sistemas la memoria se encuentra distribuida y no compartida como en los sistemas multiprocesador. Los procesadores se comunican a través de paso de mensajes, ya que éstos sólo tienen acceso directo a su memoria local y no a las memorias del resto de los procesadores. Ver Figura 7.

La transferencia de los datos se realiza a través de la red de interconexión que conecta un procesador o conjunto de procesadores con otros. Dado que la memoria está distribuida entre los diferentes elementos de proceso, estos sistemas reciben el nombre de distribuidos. Por otra parte, estos sistemas son débilmente acoplados, ya que los

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

módulos funcionan de forma casi independiente unos de otros. Este tipo de memoria distribuida es de acceso lento por ser peticiones a través de la red, pero es una forma muy efectiva de tener acceso a un gran volumen de memoria.

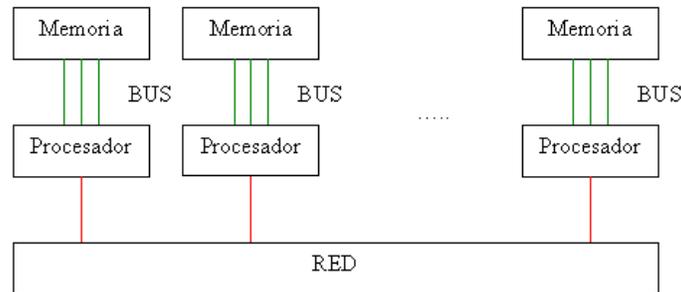


Figura 7. Modelo de arquitectura de una computadora paralela con memoria distribuida

Equipo Paralelo de Memoria Compartida-Distribuida

La tendencia actual en las máquinas paralelas es de aprovechar las facilidades de programación que ofrecen los ambientes de memoria compartida y la escalabilidad de los ambientes de memoria distribuida. En este modelo se conectan entre sí módulos de multiprocesadores, pero se mantiene la visión global de la memoria a pesar de que es distribuida.¹²

Una vez expuestas las diferentes clasificaciones de arquitecturas de computadoras, en las siguientes secciones se explicará qué es el cómputo paralelo y el cómputo distribuido, en qué consisten y su aplicación en el supercómputo y el clustering.

1.1.3 Cómputo Paralelo

El paralelismo es la realización de varias actividades al mismo tiempo, el cual tiene aplicación en diversas áreas, como en el cómputo paralelo. Éste surge de la necesidad de resolver problemas complejos en un tiempo razonable, utilizando de manera simultánea más de un procesador en una sola computadora, situación que un equipo serial (no paralelo) no satisfaría las demandas de tales problemas.

Las arquitecturas de procesamiento paralelo han sido ampliamente aprovechadas en numerosas aplicaciones, ya que proveen de un alto rendimiento, cuya meta es la reducción del tiempo de cómputo a través de la distribución de la carga de trabajo entre los procesadores existentes.

De esta manera, un programa diseñado para ejecutarse en un equipo paralelo puede aprovechar la disponibilidad de varios procesadores para aumentar la capacidad de procesamiento y minimizar los tiempos de ejecución. Existen varias razones por las que se puede determinar que un programa sea factible a ser implementado en paralelo, ya

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

sea que se requiera de una respuesta muy pronta o inmediata de los resultados, por el enfrentamiento a un problema de gran magnitud o complejidad, o bien, por gusto del programador.¹³

Sin embargo, cabe hacer mención que no cualquier programa es susceptible de ejecutarse en este tipo de plataformas. Dos puntos importantes a considerar son la independencia de datos y la identificación de secciones en el código dónde exista una mayor carga de trabajo, esto además de que se tienen que conocer ciertos datos de hardware y software al momento de decidir paralelizar, como puede ser: número de procesadores, cantidad de memoria, espacio en disco, el medio de interconexión, sistema operativo, versiones de aplicaciones, soportabilidad de los compiladores para paralelismo, entre otros, y de esta forma evaluar si realmente vale la pena implementar el paralelismo en un código o si con una arquitectura que no opere de esta forma pueden obtenerse los resultados esperados.

El cómputo paralelo ha evolucionado en todos los sentidos, desde el hardware, software, costos, hasta los algoritmos y bibliotecas especializados utilizados, puesto que muchos factores están involucrados: diseño, especialistas en la materia, tecnología empleada para su construcción, instalación, mantenimiento físico y lógico, operación, entre otros.

1.1.4 Cómputo Distribuido

Es un conjunto de elementos de procesamiento autónomos, no necesariamente homogéneos, no comparten memoria, se encuentran interconectados por una red de comunicaciones, con la posibilidad de estar distribuidos geográficamente a través del mundo y cooperan entre ellos para realizar las tareas que les han sido divididas y asignadas.¹⁴

Se diseñó para resolver problemas demasiado grandes y/o de gran complejidad, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños, es decir, se basa en la descentralización del trabajo. Esta forma de trabajar es una consecuencia natural de la evolución en el uso de las computadoras y las redes.

Existen diversos modelos de operación asociados al cómputo distribuido como:

- P2P (peer to peer): los procesos son totalmente descentralizados y cada punto de la red funciona como cliente y servidor a la vez.
- Clustering (cómputo basado en clusters): Agrupación de computadoras interconectadas a través de un nodo central que trabajan como un solo recurso.
- Grid Computing: Guarda similitud con las dos anteriores, existe una asociación de computadoras, pero sin un nodo central.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Entre la amplia gama de tecnologías aplicadas a este paradigma encontramos:

- Sockets (API Berkeley, WinSocks) C, Java.
- RPC (Remote Procedure Call) C
- RMI (Remote Method Invocation) Java
- DCOM (Distributed Component Object Model) propietario de Microsoft
- CORBA (Common Object Request Broker Architecture) C, C++, Java.
- NET Remoting C#, C++, J#.
- Servicios Web (SOA), XML (SOAP, WSDL, UDDI)

Un escenario al que se enfrenta esta tecnología es la heterogeneidad de sistemas involucrados, lo que implica que se realice un enorme esfuerzo para lograr la transparencia ante el usuario.

Entre sus características encontramos:

- **Transparencia:** Es una de sus principales características, la cual se refiere a la percepción que tiene el usuario respecto al sistema de cómputo utilizado: él ejecuta sus procesos, almacena su información, la extrae, la mueve dentro del sistema, etc., y no sabe en qué equipo específico se encuentra ésta o qué procesadores está usando, para él, es un solo sistema.
- **Escalabilidad:** Se refiere a la posibilidad de incrementar o disminuir, a conveniencia de la situación, el número de equipos interconectados al sistema distribuido existente sin que se vea afectada la calidad en los servicios que proporciona.
- **Descentralización de la carga de trabajo:** Basado en el paradigma: “Divide y vencerás”, la carga de trabajo se distribuye a través de todo el sistema, lo que se traduce en un rendimiento superior y una mayor tolerancia a fallos.
- **Cliente/Servidor:** Generalmente asociados al modelo Cliente/Servidor, donde el cliente realiza peticiones de procesamiento al Servidor y éste asigna los recursos.

Asimismo el cómputo distribuido permite:

- **Obtener gran poder de cómputo:** Los recursos son utilizados en conjunto, por lo que la potencia total equivale a la suma de todos los recursos involucrados.
- **Ganar tiempo:** Los trabajos o investigaciones que requieren de grandes cantidades de recursos de cómputo, se aceleran cuando hacen uso de computación distribuida.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Almacenamiento: El volumen de datos generados por los procesos puede ser distribuido entre los dispositivos de almacenamiento que se encuentren disponibles.
- Disminuir costos: Al utilizar recursos de diversas máquinas, mismas que tienen relativamente un costo más bajo en comparación con los grandes equipos de cómputo intensivo (por ejemplo: supercomputadoras).
- Colaboración: Entre países, organismos o grupos de investigadores situados en distintos puntos del planeta para trabajar en un proyecto común.

1.1.5 Supercómputo

La aplicación de los conceptos expuestos en las secciones anteriores: arquitectura de computadoras, cómputo paralelo y cómputo distribuido, dan lugar al Supercómputo o Cómputo de Alto Rendimiento, éste se refiere a la utilización de computadoras con grandes capacidades de cálculo, memoria, almacenamiento y comunicaciones, compiladores altamente desarrollados, bibliotecas numéricas, herramientas auxiliares para optimizar y paralelizar códigos, y sistemas para balanceo y calendarización de los trabajos enviados para su procesamiento.¹⁵

La finalidad de esta tecnología es servir como una herramienta de propósito general para apoyar la investigación científica y aplicada, lo que se conoce como hacer experimentación numérica, que permite resolver problemas más complejos, o con escalas de tiempo mayores que, equipos convencionales de cómputo, simplemente no podrían solucionar. La Figura 8 ilustra ejemplos de aplicaciones del supercómputo.

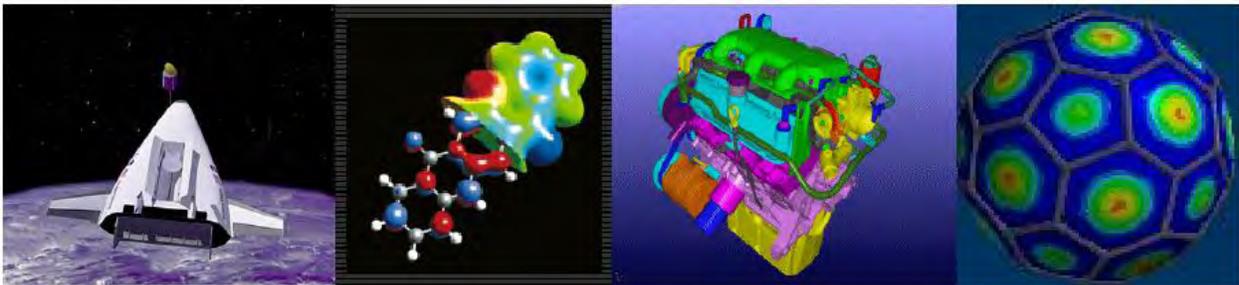


Figura 8. Aplicaciones del Supercómputo

Supercomputadora

Una de las herramientas vitales para llevar a cabo el Supercómputo o Cómputo de Alto Rendimiento, son las supercomputadoras, las cuales permiten resolver problemas de gran complejidad o tamaño que demandan proporciones de cálculo que exceden las capacidades de cómputo convencionales tanto de estaciones de trabajo como de PCs.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

En términos generales, "una supercomputadora es aquella computadora con la máxima capacidad de procesamiento, almacenamiento, memoria, velocidad de comunicación y software que existe en la actualidad."¹⁶ Puede atender más de un trabajo, a la vez que demanda el uso de todos los recursos, sin degradación para el usuario, es decir, trabaja bajo el paradigma del cómputo paralelo, explicado en la sección 1.1.3. Ver Figura 9.



Figura 9. Kan Balam. Vista frontal de la segunda línea de racks

Una supercomputadora se caracteriza por gran demanda de:

- Procesamiento de datos, memoria y otros recursos de hardware de baja latencia*.
- Millones de operaciones aritméticas por segundo.
- Números de gran precisión*.
- Gran capacidad de almacenamiento.

Es el recurso más poderoso del cómputo científico, pues es una de las principales herramientas auxiliares para el desarrollo científico y tecnológico, ya que permite al investigador llevar a cabo, con certeza y velocidad, billones de cálculos matemáticos para estudiar problemas de gran magnitud, realizar complejas investigaciones en todos los ámbitos del conocimiento humano, abriendo en todo el mundo nuevas líneas de investigación científica en áreas como ingeniería, medicina, geofísica, geografía, astronomía, química, ciencias de la atmósfera, ciencias nucleares, entre otras.¹⁷

Así pues, los productos derivados del uso de las supercomputadoras, están presentes a nuestro alrededor y forman parte integral de nuestra cultura. Algunos ejemplos de éstos son los que Oscar García y Enrique Cruz explican en su documento *Supercómputo*, los cuales son:

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

[...] el diseño automotriz, desde la aerodinámica, diseño de dispositivos hasta análisis de choques; la localización de yacimientos petroleros mediante técnicas de reconstrucción de capas del subsuelo; diseño de aeronaves de las nuevas generaciones y análisis de resistencia de materiales; estudio de fenómenos naturales como ciclones, lluvias, temblores, erupciones entre otros para la prevención de desastres; estudio de la contaminación atmosférica en grandes urbes y de crecimiento del hoyo en la capa de ozono; estudio de moléculas biológicas y diseño de nuevos fármacos; predicción del clima local y global, por citar sólo algunos.¹⁸

Como vemos, el supercómputo forma parte de nuestra cotidianeidad.

Por otro lado, existen muchas otras áreas de la investigación que no están enfocadas en aspectos cotidianos, en donde el supercómputo es una herramienta esencial, tales como: evolución de galaxias, modelos magnetohidrodinámicos*, relatividad numérica, evolución de rayos cósmicos, física atómica, mecánica estadística, sistemas complejos y dinámicos en general.

1.1.6 Cluster

Otra herramienta del cómputo de alto rendimiento son los clusters, que al igual que las supercomputadoras, están basados en el cómputo paralelo y en el cómputo distribuido.

El cómputo con clusters surgió como resultado de la convergencia de diversas tendencias, que incluyen la disponibilidad de microprocesadores económicos y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

En la actualidad, los clusters han solucionado problemas en diversos campos de aplicación que exigen un intenso poder de procesamiento, debido a la facilidad de interconectar múltiples procesadores y coordinarlos, dando como resultado computadoras paralelas.

Por lo anterior, el término cluster se aplica a un sistema de procesamiento paralelo o distribuido, formado por conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes, interconectados de tal modo que funcionan como un solo recurso de cómputo, siendo transparente para el usuario.¹⁹



Figura 10. Mixbaal. Vista posterior de los módulos

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Aunque no existe un acuerdo general en cuanto a la definición exacta de lo que significa un cluster, y muchas computadoras diferentes pueden llegar a ser clasificadas como tales, existen dos características esenciales que permiten identificarlos, y que, probablemente, son las que mayormente han influido en su rápido desarrollo, éstas son, que se construyen a partir de componentes que pueden encontrarse en el mercado habitual de cómputo, lo que en EU se conoce como “commodity of the shelf” (COTS), y que se desarrollan bajo la filosofía de “hágalo usted mismo”.²⁰

Para que un cluster funcione como tal, no basta sólo con conectar entre sí las computadoras, sino que es necesario contar con un sistema operativo que permita el manejo del mismo, interactuar con el usuario y los procesos que corren en él para optimizar su funcionamiento.

Sus características son:

- Alto desempeño.
- Gran capacidad de procesamiento y almacenamiento.
- Alta disponibilidad.
- Escalabilidad.
- Bajo costo.

Así, la tecnología de clusters ha evolucionado en apoyo de actividades como son: aplicaciones de supercómputo, software de misiones críticas, servidores Web, comercio electrónico, bases de datos de alto rendimiento, por mencionar sólo algunas.

La Figura 11 muestra el esquema general de un cluster:

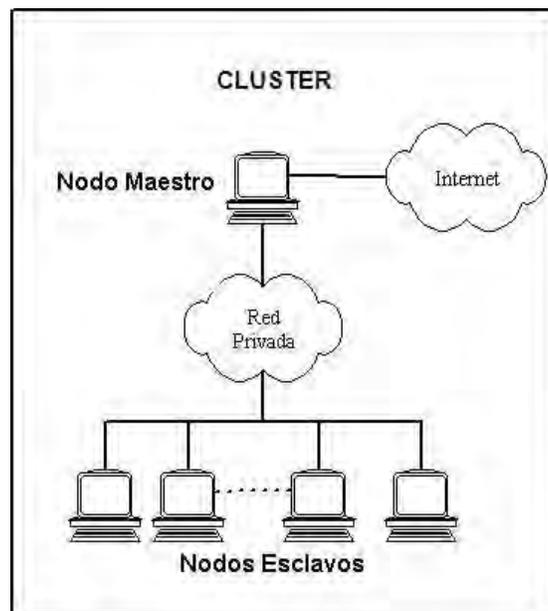


Figura 11. Esquema general de un cluster

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Componentes de un cluster

En general, un cluster requiere de diversos componentes de software y hardware, a saber:

- Nodos
- Sistemas Operativos
- Conexiones de Red
- Middleware
- Ambientes de programación paralela
- Protocolos de comunicación y servicios
- Aplicaciones (pueden ser paralelas o no)

La Figura 12 muestra dichos componentes:

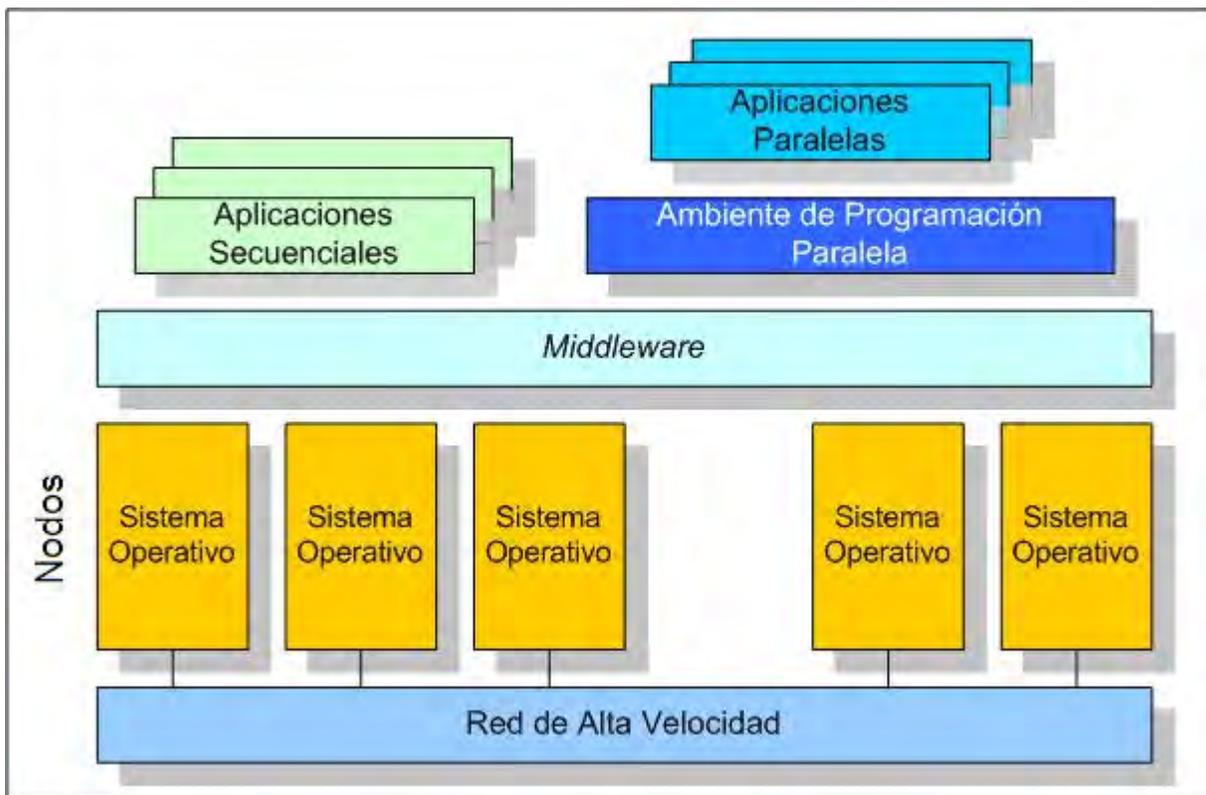


Figura 12. Componentes de un cluster

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

A continuación se describen cada uno de estos componentes:

Nodos

Se refiere a una computadora sola que contiene recursos específicos tales como memoria, interfaces de red, uno o más CPUs.

Como se puede observar en la Figura 11 en un cluster es necesario un nodo maestro y varios nodos esclavos. El nodo maestro es el servidor del cluster, es el responsable de decirle a los nodos esclavos qué deben hacer. Las funciones principales del nodo maestro son:

- Ser la interfaz entre el cluster y la red externa.
- Controlar el acceso y validación de los usuarios.
- Llevar a cabo las funciones de monitoreo.
- Exportar algunas aplicaciones del sistema a los nodos esclavos.
- Implementar los mecanismos de seguridad en el cluster.

Mientras que los nodos esclavos escuchan las instrucciones (ya sea vía ssh* o rsh*) desde el nodo maestro. Éstos deben dedicarse a proporcionar resultados y, por tanto, no ejecutar ningún servicio innecesario. Las funciones principales de los nodos esclavos son:

- Ser los encargados de hacer el cálculo numérico.
- Almacenar y compartir archivos.²¹

Sistemas Operativos

Un sistema operativo es un programa o conjunto de programas de computadora destinados a gestionar eficazmente el uso de sus recursos. Su ejecución inicia cuando se enciende la computadora. Administra el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario.²²

Los sistemas operativos utilizados en clusters deben ser multiproceso y multiusuario, así como permitir el uso y acceso fáciles para el usuario.

Algunos ejemplos de sistemas operativos para clusters, son los siguientes:

- GNU/Linux: OpenMosix, Rocks, Kerrighed
- Unix: Solaris, HP-Ux, Aix.
- Windows: NT, 2000 Server, 2003 Server
- Mac OS X

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Conexiones de Red

Los nodos de un cluster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Gigabit Ethernet, Myrinet, Infiniband y SCI.

Middleware

El middleware es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un cluster lo siguiente:

- Una interfaz única de acceso al sistema, denominada SSI (Single System Image), la cual hace que el conjunto de maquinas sea visto por el usuario como uno solo.
- Herramientas para la optimización y mantenimiento del sistema: migración de procesos, checkpoint-restart (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de cargas* y tolerancia a fallos.
- Escalabilidad: debe poder detectar automáticamente nuevos nodos conectados al cluster para proceder a su utilización²³.

Existen diversos tipos de middleware, por ejemplo: Globus, MOSIX, OpenMOSIX, Cándor, entre otros.

El middleware recibe los trabajos entrantes al cluster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un nodo. Esto se realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican dónde y cómo debe distribuir los procesos, por un sistema de monitoreo, el cual controla la carga de cada CPU y la cantidad de procesos en él.

El middleware permite migrar procesos entre nodos con distintas finalidades:

- Balanceo de carga*: si un nodo está muy cargado de procesos y otro está ocioso, pueden transferirse procesos a este último para liberar de carga al primero y optimizar el funcionamiento.
- Mantenimiento de nodos: si hay procesos corriendo en un nodo que necesita mantenimiento o una actualización, es posible migrar los procesos a otro nodo y proceder a desconectar del cluster al primero.
- Prioridad de trabajos: en caso de tener varios procesos corriendo en el cluster, pero uno de ellos de mayor importancia que los demás, puede migrarse este proceso a los nodos que posean más o mejores recursos para acelerar su procesamiento.²⁴

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Ambientes de programación paralela

Los ambientes de programación paralela permiten implementar algoritmos que hacen uso de recursos compartidos: CPU, memoria, datos y servicios.

Herramientas y protocolos de comunicación y servicios

Un protocolo de comunicación es un conjunto de reglas que indican cómo se debe llevar a cabo un intercambio de datos o información. Para que dos o más nodos en una red puedan intercambiar información es necesario que manejen el mismo conjunto de reglas, es decir, un mismo protocolo de comunicaciones.

Además permiten realizar tareas de negociación segura, inicialización, monitoreo y control de uso de los distintos recursos compartidos.

Bibliotecas para programación paralela

- MPI (Message Passing Interface) es un estándar para la comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida. Las implementaciones en MPI consisten en un conjunto de bibliotecas de rutinas que pueden ser utilizadas en programas escritos en los lenguajes de programación C, C++, Fortran, Ada y otros. MPI ha sido utilizada para casi toda arquitectura de memoria distribuida y cada biblioteca implementada ha sido optimizada para el hardware en la cual se ejecuta.²⁵
- PVM (Parallel Virtual Machine) es una biblioteca para el cómputo paralelo en un sistema distribuido de computadoras. Está diseñado para permitir que una red de computadoras heterogénea comparta sus recursos de cómputo (como el procesador y la memoria RAM) con el fin de aprovechar dichos recursos para disminuir el tiempo de ejecución de un programa al distribuir la carga de trabajo en varias computadoras.²⁶

Herramientas para la administración de tareas

Los sistemas de colas* también forman parte fundamental para las comunicaciones y servicios, que igualmente se encargan de la administración de tareas y recursos. Los siguientes son ejemplos de sistemas de colas*:

- LSF (Load Sharing Facility): Sistema de administración de recursos que funciona en un ambiente de cómputo distribuido. Proporciona al usuario una imagen unificada del sistema ya que permite clasificar los trabajos que envían los usuarios según la cantidad, el tipo de recursos que solicitan y las políticas de uso

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

de los mismos, equilibrando la carga de trabajo. Administra trabajos secuenciales y paralelos, tanto interactivos como en batch.²⁷

- OpenPBS (Portable Batch System): Versión original de PBS*, el cual es un sistema de administración de tareas flexible desarrollado para la NASA a mediados de los años 90. Opera en ambientes UNIX multiplataforma, clusters y supercomputadoras. Permite definir colas, controlar nodos y recursos que serán utilizados para la ejecución de los trabajos.²⁸

Aplicaciones (pueden ser paralelas o no)

Una aplicación es el nombre que reciben los programas especializados en tareas concretas. Para el caso de las aplicaciones propias de un cluster, las tareas realizadas son muy diversas, así como la complejidad de las mismas. A continuación se listan algunas aplicaciones comunes de un cluster dedicado al cómputo científico:

- Gaussian: Paquete de software para llevar a cabo una variedad de cálculos moleculares semi-empíricos, capaz de predecir muchas propiedades y reacciones de las moléculas, estados de transición, energías y estructuras moleculares, afinidades electrónicas y magnéticas, frecuencias vibracionales, en estado fundamental o excitado y en fase gas o disolución, haciendo uso del método "ab initio*".
- Gopenmol: Paquete de software para la visualización, análisis y la modificación de estructuras y trayectorias moleculares que pueden ser producto del cálculo de programas externos.
- Grass: Sistema de información geográfica (GIS) que ofrece rastreo, topología, análisis y gestión de datos geospaciales, procesamiento de imágenes, producción de mapas/gráficos, modelado espacial y visualización.
- OpenBabel: Es un conjunto de herramientas químicas diseñadas para convertir formatos de diferentes programas químicos. Permite convertir, analizar, capturar, tabular, etc. datos para el estudio de la estructura molecular, química, bioquímica o áreas afines.
- Gamess: Es un paquete de software de química cuántica ab initio*, sus siglas en inglés significan: General Atomic and Molecular Electronic Structure System (Sistema General de Estructura Electrónica, Atómica y Molecular). Permite realizar una gran diversidad de cálculos mecanocuánticos* de sistemas moleculares: cálculos *ab initio**, cálculos semiempíricos de orbitales moleculares, cálculos de propiedades, entre otros.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Clasificación de los clusters

Una vez descritos los diferentes componentes de un cluster, se hablará sobre su clasificación. Existen diferentes tipos de clusters, los cuales son:

Alta-disponibilidad (Fail-over o High-Availability)

El artículo *Clusters: Qué son y para qué sirven?* nos explica que este tipo de cluster “esta diseñado para mantener uno o varios servicios disponibles incluso a costa de rendimiento, ya que su función principal es que el servicio jamás tenga interrupciones como por ejemplo un servicio de bases de datos.”²⁹

Algunos ejemplos de aplicaciones para este tipo de clusters son:

- Oracle Real Application Clusters
- IBM DB2 Integrated Cluster Environment
- Linux HA
- FreeNAS
- HA-OSCAR: The High Availability Linux Project
- Kimberlite
- Lifekeeper
- Linux FailSafe
- Linux Replicated High Availability Manager
- MC/Serviceguard³⁰

Alto-rendimiento (HPC o High Performance Computing)

Este tipo de cluster está diseñado para obtener el máximo rendimiento de la aplicación utilizada incluso a costa de la disponibilidad del sistema. Este tipo de configuración está orientada a procesos que requieran mucha capacidad de cálculo. Algunos ejemplos de aplicaciones son:

- PVM
- pvmsync
- MPI
- LAM/MPI
- PLUS: MPI&PVM integration
- Parallel Knoppix

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Y algunos ejemplos de clusters de este tipo son:

- Beowulf (for Linux)
- Berkeley NOW (for Solaris)³¹

Balaneo de Carga (Load-balancing)

Este tipo de cluster esta diseñado para balancear la carga de trabajo entre varios servidores, lo que permite tener, por ejemplo, un servicio de cálculo intensivo multiusuarios que detecte tiempos muertos del proceso de un usuario para ejecutar, en dichos tiempos, procesos de otros usuarios. Algunos ejemplos de aplicaciones son:

- Alinka Oranges
- Keepalived
- Linux Virtual Server
- LVSM
- Red Hat Cluster Suite
- Turbolinux Cluster Server
- Ultra Monkey
- Condor
- openMosix
- Cluster Knoppix³²

Es importante mencionar que los nodos de un cluster pueden tener, todos, la misma configuración de hardware y sistema operativo (cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (cluster semi-homogéneo), o tener diferente hardware y diferente sistema operativo (cluster heterogéneo), lo que hace más fácil y económica su construcción.

Ventajas de usar clusters

Los clusters permiten a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo, o bien usar el que ya se tiene. Es importante resaltar esta característica de los clusters, pues los costos de implementación son muy bajos confrontados con el gran rendimiento, ganancias y utilidad que éste le reporta al usuario.

Por otro lado, las aplicaciones paralelas requieren: buen rendimiento, baja latencia*, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos, requerimientos que un cluster puede satisfacer usando los recursos que tiene asociados a él.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Con lo anterior podemos enumerar las siguientes ventajas:

- Flexibilidad.
- Disponibilidad.
- Escalabilidad.
- Incorporación de tecnología de punta.
- Heterogeneidad.
- Costos bajos.

Adicional a estas ventajas, alrededor del 80% de las 500 máquinas más potentes del mundo, son clusters.³³

Cabe señalar que para la construcción de un cluster se requieren conocimientos medios/avanzados en administración de sistemas, construcción e implementación de clusters, sistemas operativos y conocimientos básicos de redes y seguridad en informática.

Desventajas de usar clusters

Algunas de las desventajas del uso de clusters de computadoras para procesamiento paralelo son las siguientes:

- Existe poco soporte de software para manejar un cluster como un sistema integrado.
- Muchas aplicaciones importantes disponibles en multiprocesadores y optimizadas para ciertas arquitecturas, no lo están en clusters.

A pesar de sus desventajas, podemos observar que, los clusters presentan una alternativa importante para varios problemas particulares, no sólo por su economía, si no también porque pueden ser diseñados y ajustados para diversas aplicaciones. Las aplicaciones que pueden obtener provecho de clusters son aquellas en donde el grado de comunicación entre procesos sea de bajo a medio.

Cluster Tipo Beowulf

Dejando atrás los aspectos generales de los clusters, es importante hablar sobre los clusters tipo Beowulf, debido a que el desarrollo de este proyecto se basa en la implementación y configuración de un cluster de este tipo. Por esta razón, en esta sección se hablará sobre la historia, características y ventajas de los mismos.

Respecto a su historia, en 1994, bajo el patrocinio del proyecto Earth and Space Sciences (ESS), Donald Becker y Thomas Sterling, un grupo de investigadores del Center of Excellence in Space Data and Information Sciences (CESDIS), que desarrolla

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

proyectos para la NASA, construyeron un cluster que consistía en 16 equipos con procesadores Intel DX4, interconectados por una red tipo Ethernet de canal múltiple. Thomas Sterling nombró Beowulf a dicho proyecto en honor al héroe de un cuento inglés, quien liberó a “Danes of Heorot” del monstruo opresivo Grendel. A manera de metáfora, se nombró Beowulf a esta nueva estrategia en cómputo de alto rendimiento que hace uso de tecnología bien difundida en el mercado, para derrotar a los costos opresores en tiempo y dinero que requiere el supercómputo.

Nikos Drakos y Ross More, hablan más al respecto:

La motivación del proyecto que creó el primer Beowulf era el explorar la posibilidad de construir una plataforma de cómputo paralelo de alto rendimiento basándose en componentes comerciales comunes. Esto se hizo pensando en limitantes que existen en sistemas MPP -Procesamiento masivamente paralelo, por sus siglas en inglés- comerciales. Estos sistemas, [...], dependen en gran medida del bus de comunicaciones, que requiere microelectrónica especializada, y otros elementos que en ocasiones se construyen a propósito para cada equipo. Un equipo con construcción muy especializada tiene un tiempo de vida limitado. Esto afecta al personal que trabaja en el mantenimiento de dicho equipo, así como a los usuarios del mismo, implicando un alto costo cuando el equipo, ya obsoleto, es reemplazado por uno más reciente.³⁴

Es por ello que la utilización de tecnología más común, en la construcción de equipos de alto rendimiento, buscó obtener una plataforma más genérica, que requiriera menos cambios en el modelo y técnicas de programación, permitiendo una mayor continuidad en los proyectos que utilizarían esta plataforma.

De esta manera Beowulf fue un gran éxito, y los clusters de este tipo pronto se popularizaron dentro de la NASA, y más allá, convirtiéndose en una técnica extremadamente popular para obtener cómputo de alto rendimiento a bajo costo.

Pero, a todo esto, ¿qué es un cluster tipo Beowulf? Pues bien, existen tantas definiciones de Beowulf así como personas que han construido o usado clusters de este tipo. Algunos autores afirman que se debe llamar Beowulf sólo aquellos construidos de la misma forma que el cluster original de la NASA. Otros autores, un tanto extremistas, llaman Beowulf a cualquier sistema de computadoras que ejecute código paralelo.

Nikos Drakos y Ross More proporcionan una definición acertada: “Es una clase de computadora masivamente paralela de alto rendimiento construida primordialmente con componentes comerciales ampliamente disponibles”.³⁵

Profundizando un poco más, un sistema Beowulf usualmente consiste de un nodo servidor (maestro) y uno o más nodos clientes (esclavos), interconectados a través de alguna de las redes tipo Ethernet, en sus tres variantes (Ethernet, Fast Ethernet y

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Gigabit Ethernet) o a través de conexiones, como HIPPI* y Myrinet, esta última de alta velocidad (2 Gbps), con conmutación y bajo tiempo de latencia*.

Una característica importante de un Beowulf es que las actualizaciones de hardware más comunes, como son actualización de procesador, incremento de memoria, o mejora de velocidad de transferencia en la red, no cambian el modelo de programación utilizado. Por lo tanto, los usuarios de estos sistemas pueden contar con mejor compatibilidad con equipos futuros.

Respecto a los nodos que conforman el cluster, éstos se encuentran dedicados únicamente a tareas del cluster. En cuanto a los programas, éstos están escritos, en su mayoría, en C o en Fortran, utilizando bibliotecas de paso de mensajes para realizar operaciones paralelas.

Los cluster tipo Beowulf se construyen con una infraestructura de software de código abierto* (Linux). Existen diversos factores que determinan que Linux sea el sistema operativo más utilizado en un Beowulf, tales como:

- Código fuente, disponible y manipulable, que permite ajustarse a las necesidades del usuario, consiguiendo que éste alcance sus metas de rendimiento, mismo que se encuentra en constante desarrollo, logrando así, que el sistema operativo se mantenga a la vanguardia con los avances tecnológicos.
- Multiusuario.
- Multitarea.
- Portable.

Por otro lado, existen algunas herramientas que permiten instalar, configurar, mantener y actualizar clusters Beowulf de forma automática, como es el caso de OSCAR y Rocks. Las dos herramientas incluyen las mejores prácticas de computación para clusters de este tipo.

Actualmente existen infinidad de Beowulfs en toda clase de ambientes, tales como: instituciones de investigación, universidades, dependencias gubernamentales y empresas privadas, que sitúan el cómputo de alto rendimiento al alcance de las masas.

1.1.7 Grid

El término "Grid" proviene de una analogía con la "red eléctrica" (power grid), para describir una "fuente universal de poder de cómputo", extendida y fácil de usar.³⁶

Grid es una tecnología que permite ubicar y acceder a recursos compartidos en forma segura vía redes de área extendida o WANs. Es un conjunto de equipos de alto rendimiento (clusters o supercomputadoras), con arquitecturas en paralelo,

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

interconectados por red, los cuales distribuyen tareas entre los equipos que lo forman, éstos pueden ser homogéneos o heterogéneos.

Este tipo de arquitecturas trata de distribuir cargas de trabajo acorde a las características internas de cada recurso y las necesidades propias de cada problema, esto se hace a dos niveles, uno en la parte de programación conjuntamente con el balance de cargas* y otro en la parte de hardware según las características de cada arquitectura que conforman a la grid.

El cómputo en grid es naturalmente un entorno multiusuario, por ello, las técnicas de autorización segura son esenciales antes de permitir que los recursos informáticos sean controlados por usuarios remotos.

Enrique Cruz y José Luis Gordillo dan una definición más de grid:

Una grid es la integración de equipos de trabajo ubicados físicamente en forma remota que comparten recursos, instrumentos, dispositivos y tecnologías como las asociadas a redes de alta velocidad o dedicadas, equipos disponibles para el cálculo numérico como computadoras vectoriales*, paralelas, clusters, redes de estaciones de trabajo, dispositivos de almacenamiento e instrumentos como telescopios, microscopios o controladores, y sistemas de software que interaccionan entre el usuario final y los dispositivos para recibir y ejecutar sus solicitudes.³⁷

Entonces, las tecnologías de grid buscan cambiar la forma en que se usan los recursos computacionales, de manera que los investigadores tengan la oportunidad de usar cierta cantidad de recursos, sin importar su ubicación geográfica.

Los requerimientos básicos para lograr una grid son:

- Integración de recursos individuales de software y hardware en un recurso combinado vía red.
- La implementación de un middleware para proveer una vista transparente de los recursos disponibles.

Una grid es más que una colección de recursos, también es un conjunto de servicios, tales como:

- Obtener información acerca de sus componentes.
- Localizar y calendarizar recursos.
- Ofrecer comunicación.
- Acceder a código y datos.
- Incluir medidas de desempeño.
- Realizar la autenticación de usuarios y recursos.
- Garantizar la privacidad de las comunicaciones.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Algunos ejemplos del uso de las grids se muestran en la siguiente tabla:

Categoría	Características	Ejemplos
Supercómputo Distribuido	Problemas muy grandes que requieren enormes cantidades de memoria y tiempo de CPU.	Dinámica estelar. Química computacional
Hight Throughput	Utilizar recursos en desuso para incrementar la cantidad de trabajo realizado.	Diseño de chips, estudio de parámetros, criptografía.
Sobre demanda	Integración de recursos remotos y locales por un período limitado.	Instrumentación médica, sistemas distribuidos para la solución de ecuaciones.
Datos intensivo	Síntesis de nueva información de grandes cantidades de datos.	Exploración espacial, datos físicos, minería de datos.
Colaborativo	Apoyar la comunicación o trabajo colaborativo entre múltiples participantes.	Diseño colaborativo, exploración de datos, educación.

Tabla 2. Uso de las grids

La Figura 13 ilustra algunas grids alrededor del mundo:



Figura 13. Ejemplos de grids alrededor del mundo

En México, las grids son una consecuencia positiva de cohesionar a las diversas comunidades de cómputo de alto rendimiento, lo que se traduce en colaboraciones más estrechas entre éstas, permitiendo el fomento de la investigación científica en el país.

1.1.8 Mecanismos de Seguridad

Ahora bien, como se mencionó en la sección anterior, en una grid se permite acceder a recursos compartidos en forma segura, por ello es importante abordar algunos conceptos de Seguridad en Informática, tales como, Criptografía, Certificados Digitales y Autoridades Certificadoras (CA, por sus siglas en inglés).

Cabe señalar que para fines de este trabajo, no se profundizará en dichos temas, de modo que si el lector desea conocer más sobre éstos puede acercarse a las fuentes disponibles que considere necesarias.

Criptografía

La criptografía, del griego krypto, (oculto) y graphos (escribir), significa literalmente "escritura oculta".³⁸ Es el arte/ciencia de cifrar y descifrar información mediante técnicas especiales y se emplea frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos.

La criptografía se conforma de los siguientes elementos:

- Cifrar. Transforma la información en una forma no legible asegurando la privacidad.
- Descifrar. Es el inverso de cifrar; es decir, transforma la información cifrada a su forma original legible.
- Autenticación. Identifica a una entidad como un individuo, una máquina en la red o una organización.
- Firmas digitales. La relación de un documento con el dueño de una "llave" particular siendo el equivalente a la firma de un documento.
- Verificación de firmas. Verifica que una firma en particular sea válida.

De estos elementos se deriva un mecanismo que permite identificar a todos los clientes que intenten conectarse a un servidor (DNS* dinámico) así como verificar que los clientes sean realmente los que dicen ser y no sean impostores.

Para llevar a cabo éste mecanismo se han implementado dos métodos:

Llave simétrica o secreta

Utiliza una misma llave para cifrar y descifrar la información enviada a través de la red; pero el problema que se presenta es que tanto quien envía como quien recibe la información deben tener la misma llave asegurándose que nadie más pueda obtenerla porque si intercepta la información pudiera descifrarla y leerla fácilmente.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Llave asimétrica o pública

Ésta surgió para resolver el problema presentado por la llave simétrica. Es un método de transmisión de información en donde el que recibe la información puede estar seguro de la identidad de quien la envió. La idea básica de este método es el uso de un par de llaves:

- Llave privada. Solamente su dueño la conoce y se usa para descifrar la información enviada por otras personas.
- Llave pública. Esta se publica y se usa por cualquier persona para cifrar la información antes de enviarla a su destino (dueño).

El par de llaves se genera simultáneamente, usando algoritmos especiales en donde los mensajes que se cifran con la llave pública de una persona puedan ser descifrados solamente con la llave privada de esa misma persona y viceversa. Por lo tanto, para establecer una comunicación segura ya no es necesario compartir primeramente una llave privada. Por ejemplo, si un cliente deseara enviar información segura a un servidor, el servidor daría su llave pública (por correo electrónico) y el cliente haría lo siguiente:

- Cifrar la información usando la llave pública del servidor y luego se la envía.
- El servidor recibiría la información y la descifraría usando su llave privada.

Esta transmisión es segura en el sentido de que nadie más que reciba la información podrá leerla porque no sabe el valor de la llave privada.

Sin embargo, existe un problema con éste método, pues si bien se tiene la llave pública del servidor, ¿cómo sé que la llave pública realmente es suya y no una llave pública generada por algún impostor que desea interceptar mis mensajes? Este problema es más serio cuando es usado para verificar automáticamente la comunicación entre dos hosts, tales como un cliente (browser) y un servidor (DNS* dinámico). Aquí es donde intervienen los certificados.³⁹

Certificados digitales

Los certificados digitales, también llamados identificadores digitales, pasaportes digitales o certificados de llaves públicas, son documentos electrónicos que verifican que una llave pública corresponda a un individuo o entidad determinados. De este modo se evita que intrusos utilicen una combinación de claves asegurando ser otra persona. El certificado permite a un individuo demostrar que es quien dice ser. Es el equivalente digital a un pasaporte, gafete de empleado o licencia de manejo, por poner algunos ejemplos.

En la red de redes, un certificado establece nuestra identidad, ante servidores que requieran verificar que somos realmente clientes autorizados para acceder a él.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Un certificado debe contener, por lo menos, lo siguiente:

- Nombre Distintivo de la entidad. Incluye la información de identificación (el nombre distintivo) y la llave pública.
- Nombre Distintivo de la Autoridad Certificadora. Identificación y firma de la Autoridad Certificadora (CA) que firmó el certificado.
- Período de Validez. El período de tiempo durante el cual el certificado es válido.
- Información adicional. Puede contener información administrativa de la CA como un número de serie o versión.

Los certificados digitales sólo son útiles si existe alguna Autoridad Certificadora (Certification Authority o CA) que los valide, ya que si alguien se certifica a sí mismo no existe ninguna garantía de que su identidad sea la que anuncia, y por lo tanto, no debe ser aceptada por un tercero que no lo conozca.⁴⁰

Autoridad Certificadora (CA)

Una Autoridad Certificadora (CA, por sus siglas en inglés) es la encargada de confirmar que el dueño de un certificado es realmente la persona que dice ser.

Las Autoridades Certificadoras (o notarios electrónicos) son entes fiables y ampliamente reconocidos que firman las claves públicas de las personas, certificando con su propia firma la identidad del usuario. Por lo tanto, si se desea establecer una Autoridad Certificadora, éstas deben tomar extremadas precauciones para evitar que sus claves caigan en manos de intrusos, lo cual comprometería todo el sistema.

Una CA provee los servicios de verificación, procesamiento, firma, asignación y manejo de Certificados. Dicho manejo consiste en determinar cuánto tiempo van a ser válidos y mantienen listas de certificados que ya no son válidos (Listas de Revocación de Certificados o CRLs).⁴¹

Así, si dos partes poseen certificados, y cada una confía en la Autoridad Certificadora que generó el certificado de su contraparte, entonces, las dos partes pueden probarse una a la otra que ellos son quienes dicen ser dándose así la *autenticación mutua*. Y para que ésta se logre cada parte debe poseer el certificado de la CA y estar seguro de que efectivamente le pertenece a la CA.⁴²

1.2 Antecedentes

En la sección que antecede se presentó un panorama general sobre los diferentes sistemas de cómputo, detallando sobre arquitecturas de computadoras, computadoras paralelas, supercomputadoras, clusters, grids y mecanismos de seguridad.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Por lo que esta sección pretende aterrizar estos conceptos, exponiendo una problemática real y aplicando una solución que involucra estos elementos.

De modo que a continuación se enunciará qué es PRAGMA, su historia, conformación, actividades que realiza y en qué consiste la participación de la UNAM con PRAGMA. Así mismo se expondrá la problemática, justificación, objetivos y alcance de este proyecto.

1.2.1 Grid PRAGMA

PRAGMA (Asamblea de la Cuenca del Pacífico para Aplicaciones de Grid y Middleware - Pacific Rim Applications and Grid Middleware Assembly) se formó en 2002 por PRAGMA Resources Working Group con el objetivo de establecer grupos de apoyo permanentes y promover el uso de tecnologías de grid entre una comunidad de investigadores que trabajan con las principales instituciones de la Cuenca del Pacífico.

PRAGMA ofrece la infraestructura y el ambiente necesario, a través de su grid, para que los investigadores puedan innovar y experimentar con nuevos enfoques y nuevas soluciones para hacer más fácil el uso de la Grid. Un ejemplo de esto fue la creación de una nueva distribución de Linux llamada *Rocks*, preparada especialmente para uso en clusters y grid computing, pues cuenta con *grid-enabled**, lo que facilita las configuraciones e integraciones de software en una infraestructura grid, logrando así que los grupos de trabajo se dediquen a resolver más rápidamente sus problemas en la grid y no que la grid sea un problema.

Actualmente existen 35 instituciones, en 16 países, agrupando 27 centros de cómputo, que forman parte de PRAGMA, las cuales se reúnen dos veces al año durante Talleres PRAGMA, mismas que se listan a continuación:

- Australia:
 - APAC (Australia Partnership for Advanced Computing)
 - MU (Monash University)
- China:
 - CNIC (Computer Network Information Center)
 - GUCAS (Graduate University of Chinese Academy of Sciences)
 - JLU (Jilin University)
 - LZU (LanZhou University)
- Chile:
 - UChile (University of Chile)
- Hong Kong:
 - CUHK (Chinese University of Hong Kong)
- India:
 - UoHyd (University of Hyderabad)

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Japón:
 - AIST (National Institute of Advanced Industrial Science and Technology)
 - OSAKAU (Osaka University)
 - TITECH (Tokyo Institute of Technology)
 - UTsukuba (University of Tsukuba)
- Korea:
 - KISTI (Korea Institute of Science and Technology Information)
- Malasia:
 - MIMOS (MIMOS Berhad)
 - USM-CS (University Sains Malaysia)
 - USM-Ph (University Sains Malaysia – Pharmaceutical)
- México:
 - UNAM (Universidad Nacional Autónoma de México)
 - CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada)
- Puerto Rico:
 - UPRM (Universidad de Puerto Rico en Mayaguez)
- Singapur:
 - BII (Bioinformatics Institute, Singapore)
 - IHPC (Institute of High Performance Computing)
 - NGO (National Grid Office), NTU (Nanyang Technological University)
- Suiza:
 - UZH (University of Zurich)
- Taiwan:
 - ASGC (Academia Sinica Grid Computing Center)
 - NCHC (National Center for High-performance Computing)
- Tailandia:
 - NECTEC (National Electronics and Computer Technology Center)
 - ThaiGrid
- Estados Unidos:
 - Binghamton University
 - NCSA (National Center for Supercomputing Applications)
 - SDSC (San Diego Supercomputer Center)
 - UUtah (University of Utah)
- Vietnam:
 - HCMUT (Ho Chi Minh City University of Technique)
 - HUT (Hanoi University of Technique)
 - IOIT-HCM (Institute of Information Technology, HCMC)

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

La Figura 14 ilustra la ubicación de estas instituciones a lo largo de la Cuenca del Pacífico:

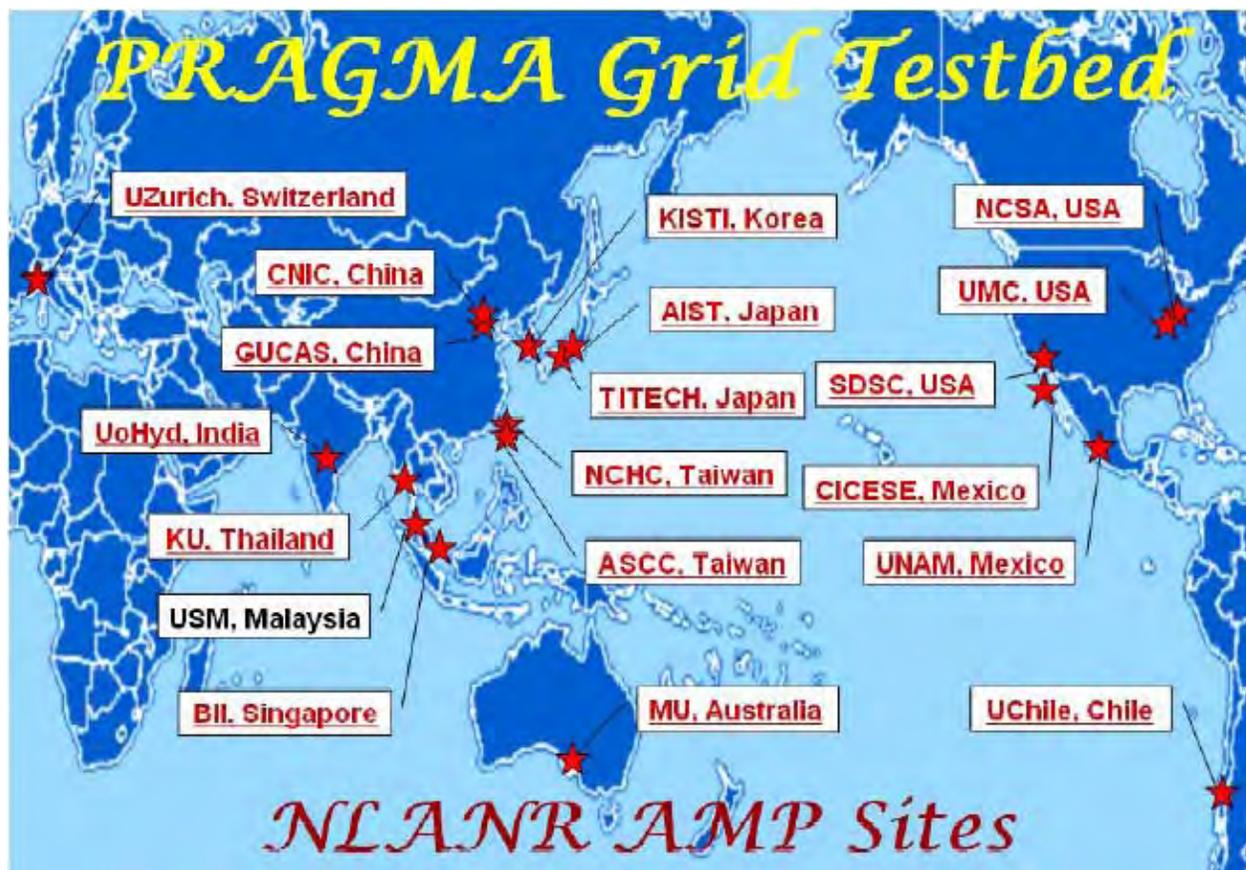


Figura 14. Instituciones que forman parte de la grid de PRAGMA

Hasta ahora los resultados obtenidos por parte de PRAGMA han sido el desarrollo de aplicaciones científicas, la creación de un lecho de pruebas para proyectos regionales e-science*, la participación en las investigaciones desarrolladas en la grid a nivel internacional, así como también ha incrementado la interoperabilidad de la grid en la Cuenca del Pacífico y en todo el mundo.

Grupos de trabajo

PRAGMA está conformado por 3 grupos de trabajo:

- **Biociencia:** Este grupo ha establecido varios proyectos que son de interés para los miembros participantes. Para alcanzar la meta común de conducir investigaciones científicas de rutina se utiliza la evolución de infraestructura de grid, mientras se desarrollan los componentes faltantes necesarios en colaboración con otros grupos de trabajo, y difundiendo las lecciones aprendidas.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- **Teleciencia:** Tiene como propósito crear y desarrollar nuevas tecnologías de información que permita a los científicos usar de manera remota dispositivos científicos avanzados y mostrar el futuro de la ciencia sobre una infraestructura avanzada.
- **Recursos y datos:** Su meta es mejorar la interoperabilidad del middleware alrededor del mundo y hacer más fácil el uso de grids para los científicos.



Figura 15. Grupos de trabajo de PRAGMA

Principales actividades

Las principales actividades que se llevan a cabo en PRAGMA se centran en fomentar y dirigir proyectos conjuntos (multilaterales) que promuevan el desarrollo de instalaciones y tecnologías de grid, compartir recursos para asegurar el éxito de los proyectos y efectuar intercambio de investigaciones.

Respecto a sus actividades internas relacionadas con el crecimiento y progreso de la institución, llevan a cabo reuniones regulares, eventos culturales y conferencias internacionales para una retroalimentación entre sus miembros.

Casos de éxito

- **GLEON:** La Global Lake Ecological Observatory Network es una red con base en limnólogos*, expertos en tecnología de la información, e ingenieros que tienen la meta común de construir una red escalable y persistente de observatorios de ecología de lagos. Los datos de estos observatorios permiten comprender mejor los procesos claves como los efectos del clima y cambios en el uso de la tierra en función de los lagos, el papel de eventos episódicos como tifones en el reajuste de la dinámica de los lagos, y los ciclos de carbono dentro de éstos.
- **GEOGrid:** Destinado a ofrecer una infraestructura e-science* a la comunidad mundial dedicada a las Ciencias de la Tierra. En esta rama se generan grandes

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

variedades de conjuntos de datos incluyendo imágenes de satélite, datos geológicos, entre otros. De manera que GEOGrid integra todos estos datos de manera virtual, gracias a la tecnología grid.

- Enciclopedia of Life: es un proyecto de gran escala computacional con el objetivo de contar con un catálogo del proteoma* de cada especie viviente. Los tres objetivos específicos que persigue este proyecto son: 1) registrar todas las secuencias de proteínas de todos los organismos, 2) integrar dichos registros con los de una amplia variedad de recursos públicos de datos y 3) distribuir esta información a la comunidad, a través de interfaces innovadoras y herramientas de software para compartir la información. Dichos objetivos se cumplen a través de tecnologías grid ofrecidas por PRAGMA y otras instituciones.

Hardware y Software

Los equipos con los que se trabaja son sistemas Linux y actualmente PRAGMA está constituido aproximadamente de más de 240 nodos, distribuidos en más de 16 lugares. Trabaja con ApGrid (<http://www.apgrid.org>) y puede compararse con un TeraGrid, los cuales tienen un poder de cómputo de 20 Tflops en promedio.

Las aplicaciones que se pueden correr en la Grid son:

- Aplicaciones Real Science y Real Grid
- Aplicaciones que puedan enviar trabajos a través del Globus Gatekeeper a los despachadores de trabajos locales, tales como SGE*, PBS*, LSF*, lo que permite compartir recursos en la grid.

Aspirantes a unirse PRAGMA

Cualquier institución que haya trabajado con aplicaciones de grid y middleware, que esté interesada en construir una larga relación de colaboración con PRAGMA.

Los investigadores interesados en la grid de PRAGMA pueden trabajar a través de sus instituciones de origen o a través de las instituciones que ya son miembros de la grid de PRAGMA.

No es necesario que cada institución miembro contribuya con recursos de hardware, sin embargo para propósitos prácticos, es mucho más fácil para los manejadores de aplicaciones acceder a la grid desde un nodo local que sea parte de la grid.

Por otro lado si algún miembro de PRAGMA desea correr una aplicación en la grid, debe enviar un mensaje de correo electrónico a la Coordinadora de PRAGMA (Cindy Zheng), indicando su petición correspondiente, incluyendo datos como descripción de la

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

aplicación, qué problema científico trata de resolver, la persona que administrará la aplicación, solicitar una cuenta de acceso y realizar pruebas de acceso.

Beneficios al formar parte de PRAGMA

- Apoyar la investigación en México.
- Ser parte de grupos de investigación a nivel internacional.
- Compartir recursos, independientemente de su localización.
- Una buena forma de hacer e-science*
 - Capacidad para resolver problemas de mayor complejidad.
- Menor gasto que el de una supercomputadora.
- Se puede crecer de manera incremental.
 - Se evita caer en la obsolescencia.
 - Se actualiza más fácil y económicamente.
- Innovación
 - Se pueden desarrollar novedosas aplicaciones impactando a nivel nacional y regional.
- Ciencia sin fronteras

Desventajas al formar parte de PRAGMA

- La comunicación entre los miembros de la Grid puede llegar a ser lenta y confusa.
- La instalación del middleware, como Globus, no es sencilla.

1.2.2 Colaboración de la UNAM con PRAGMA

En la UNAM, el supercómputo se aplica en diferentes áreas de investigación, como son las ciencias de materiales, ciencias de la atmósfera, ciencias de la tierra, ciencias biológicas, ciencias químicas, astronomía, astrofísica, física de altas energías, física atómica y molecular, química cuántica, bioquímica, dinámica de fluidos, entre otras.

El 25 de agosto de 2004 la UNAM se unió al proyecto PRAGMA, con el objetivo de participar en las diferentes pruebas experimentales de cómputo científico propuestas por PRAGMA. La colaboración se inició con lo siguiente:

- Infraestructura
 - Cluster Linux 10 cpus heterogéneos, denominado Malicia
 - PBS*
 - MPI/MPICH-G
 - Globus 2.4

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- Aplicaciones
 - SCMSweb*: Monitoreo
 - MOGAS: Contabilidad
 - Ninf-G: Grid-RPC
 - Nimrod: Experimentos Paramétricos
 - Amber

Así la UNAM ha participado en diferentes experimentos de aplicaciones de grid, usando equipos locales, tales como aulas de la DGSCA, y a partir de la prueba de diferentes arquitecturas y procesadores.

Posteriormente se adquirió el compromiso de integrar a la UNAM en la grid de PRAGMA a través de la aportación de un cluster configurado y puesto a punto, con el fin de participar activamente con esta comunidad en la ejecución de sus experimentos, compromiso que se cumplió con la realización de este proyecto.

1.2.3 Problemática

En el Departamento de Supercómputo de la DGSCA, UNAM, es en donde recae directamente la responsabilidad de cumplir con el compromiso adquirido a través de convenios de colaboración establecidos con PRAGMA.

El Departamento de Supercómputo cuenta con varios equipos de cómputo numérico intensivo destinados a diversos proyectos, algunos de ellos en producción y otros en pruebas experimentales. De éstos últimos ninguno cumplía con los requisitos mínimos de configuración requeridos por PRAGMA para poder formar parte de la grid, es por ello que surge la propuesta de configurar y poner a punto alguno de los equipos disponibles en el Departamento, para que así la UNAM a través de la DGSCA y el Departamento de Supercómputo, pueda participar de manera activa en este proyecto de grid.

1.2.4 Justificación

Situación Actual

La DGSCA de la UNAM, es la dependencia universitaria encargada de proveer, administrar y mantener los recursos de informática, cómputo, Internet y telecomunicaciones que posee la Universidad.

Así como también posee una infraestructura tecnológica que la ubica en los mejores niveles nacionales e internacionales. Es responsable y representante de la actuación de la Universidad y de México, en el espacio cibernético, ante instancias internacionales

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

que reconocen el trabajo, profesionalismo y calidad de la institución y de sus integrantes.

La DGSCA proporciona una infraestructura de supercómputo compuesta por equipos de cómputo numérico intensivo, dispositivos de almacenamiento masivo y redes de alta velocidad, como herramientas para la investigación científica y tecnológica.

Esta infraestructura es operada por el Departamento de Supercómputo, cuya función principal es proporcionar a la comunidad el servicio de supercómputo, a través de mantener en funcionamiento los equipos y de proporcionar a los usuarios los mecanismos necesarios para su utilización, así como las asesorías y ayuda que requieran.

Ahora bien, actualmente el Departamento de Supercómputo cuenta con 3 equipos principales de supercómputo: Kanbalam (HP CP 4000), Alebrije (SGI Altix 350) y Bakliz (HP AlphaServer SC 45) y con 4 equipos destinados a proyectos específicos. Y como ya se mencionó, ninguno estaba destinado a satisfacer los compromisos contraídos con el proyecto PRAGMA.

Por lo anterior se propone la configuración y puesta a punto de uno de esos 4 equipos destinados a proyectos específicos, a saber, el Cluster Intel Pentium III Beowulf de 10 nodos, denominado Mixbaal, el cual cumplía con los requisitos mínimos para unirse a la grid de PRAGMA. Además de documentar el procedimiento a seguir para su futura consulta y referencia.

Es importante mencionar el porqué la implementación de este cluster va a resultar en beneficio de la UNAM, en particular del Departamento de Supercómputo de la DGSCA.

Como ya hemos mencionado en la sección 1.1.6 apartado *Ventajas del Cluster*, uno de los mayores beneficios de usar clusters es el ahorro significativo en cuestión de costos. Y la configuración e implementación del Cluster Mixbaal, no es la excepción.

A continuación se presenta una comparación de tres equipos, que son algunos de los más representativos con los que cuenta el país para realizar cómputo científico. Los cuales son:

- Supercomputadora Cray XD1 (Unpotosi): Instituto Potosino de Investigación Científica y Tecnológica (IPICYT), inaugurada el 4 de agosto de 2006.
- Supercomputadora HP CP 4000 (Kanbalam): Universidad Nacional Autónoma de México (UNAM), inaugurada el 16 de enero de 2007.
- Cluster HP HPC (Aitzaloea): Universidad Autónoma Metropolitana (UAM), inaugurada el 28 de noviembre 2008.

Incluyendo en esta comparativa al cluster en cuestión, Cluster Intel Pentium III (Mixbaal) de la UNAM, inaugurado en 2002.

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

Esto con el objeto de mostrar el ahorro e impacto considerable que tendrá la configuración del Cluster Mixbaal para integrarse a la grid de PRAGMA.

Equipo	Costo Aproximado	Año de Adquisición	Características	Poder de Cómputo
Cluster Intel Pentium III (Mixbaal) ⁴³	500 mil pesos	2002	– 10 Gigabytes de memoria RAM – 400 Gigabytes en disco duro – 20 procesadores a 1130.497 Mhz	20 Gflops
Cray XD1 (Unpotosi) ⁴⁴	10 millones de pesos	2006	– 216 Gigabytes de memoria RAM – 4 Terabytes en disco duro – 432 procesadores a 2.2 Ghz	0.95 Tflops
HP CP 4000 (Kanbalam) ⁴⁵	33 millones de pesos	2007	– 3000 Gigabytes de memoria RAM – 160 Terabytes en disco duro – 1368 procesadores a 2.6 Ghz	7.113 Tflops
Cluster HP HPC (Aitzalao) ⁴⁶	12 millones de pesos	2008	– 4320 Gigabytes de memoria RAM – 150 Terabytes en disco duro – 2160 procesadores a 3.0 Ghz	18.4 Tflops

Tabla 3. Comparativo de equipos de supercómputo en México

Como podemos observar en la tabla anterior, los costos para comprar una supercomputadora son realmente altos.

Ahora bien, la velocidad de cambio en la industria de la computación es tal, que muchos artículos de cómputo se vuelven obsoletos tan solo unos meses después de su compra. Por tanto, muchas veces el valor de los equipos de cómputo puede depreciarse tan rápido que puede quedar registrado como sin valor al paso de dos o cuatro años⁴⁷, por lo que con el dinero invertido la primera vez, se puede comprar un equipo con mayor poder de cómputo, al término de ese periodo, y tomando como base el equipo de supercómputo, adquirido recientemente a nivel nacional por la UAM en el 2008, la UNAM debería erogar como mínimo, 12 millones de pesos, aproximadamente, para adquirir un equipo con capacidades de procesamiento mayores a las de éste último.

Si bien la implementación del Cluster Mixbaal no superará el poder de cómputo de ninguna de las máquinas listadas arriba, mismo con el que el Departamento de Supercómputo contaba desde el 2002 y cuya depreciación en el tiempo deja claro que su valor actual es muy bajo, pertenecerá a la grid de PRAGMA, permitiendo así, compartir recursos de otras instituciones ubicadas a lo largo de la Cuenca del Pacífico, que conjuntamente pueden dar como resultado una capacidad de procesamiento igual o superior a los 20 Tflops.

Además de evadir los costos que implica la actualización tecnológica, reutilizar la tecnología existente tiene la ventaja de que el Departamento de Supercómputo está familiarizado con ella y evita los riesgos de instrumentar una sin probar.⁴⁸

Aunado a lo anterior, la participación de la UNAM en la grid de PRAGMA será para fines de experimentación y pruebas, que servirán como antecedente para que en un futuro, la

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

supercomputadora Kanbalam, o bien otros equipos de producción, se incorporen en algún proyecto de grid.

Todo esto sin necesidad de que la UNAM erogase cantidad alguna para la adquisición de un equipo de supercómputo.

1.2.5 Objetivo del proyecto

Objetivo General

Configurar y poner a punto el Cluster Intel Pentium III Beowulf de 10 nodos, denominado Mixbaal, perteneciente al Departamento de Supercómputo, que permita realizar cómputo de alto rendimiento para que la UNAM se integre a la grid de PRAGMA y éste sea utilizado por las 35 instituciones que forman parte del proyecto. Así como generar la documentación del procedimiento seguido para lograr lo anterior.

Objetivos Específicos

- Instalar los 10 nodos que formarán parte del cluster.
- Configurar la herramienta Globus.
- Crear las cuentas necesarias para el consejo de la asamblea PRAGMA, así como las cuentas de administración para el área de Administración de Sistemas Unix del Departamento de Supercómputo.
- Instalar dos aplicaciones de cómputo científico.
- Configurar los certificados de seguridad correspondientes.
- Realizar pruebas de funcionamiento del cluster.

1.2.6 Alcances

- El cluster estará destinado a proveer un servicio de cómputo de alto rendimiento a las entidades que forman parte del proyecto PRAGMA que son instituciones dedicadas a investigación científica y que por su naturaleza demandan grandes cantidades de recursos de cómputo. En este sentido la participación de la UNAM será limitada a cumplir los requisitos mínimos establecidos por la Asamblea debido a que el cluster Mixbaal es un cluster de experimentación, pero éste servirá como un proyecto de base para futuras implementaciones de producción en proyectos de grid.
- Las cuentas creadas solo serán las establecidas y necesarias para permitir al Departamento de Supercómputo su operación.
- Se instalará OpenBabel y Gamess como parte de las aplicaciones requeridas por PRAGMA. Ambas aplicaciones son de química computacional, ya que la mayoría

CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

de los usuarios que utilizan cómputo científico, trabajan en áreas como dinámica molecular, química cuántica, ciencias de materiales, nanotecnología, bioquímica, entre otras.

Fuentes de Consulta

- ¹ Flores Salgado, Lourdes Yolanda. Apuntes del curso: *Tópicos Selectos de Administración de Supercómputo*, Departamento de Supercómputo, DGSCA, UNAM, México, 2007.
- ² Ídem.
- ³ Red Hat Inc. Red Hat Enterprise Linux 4. *Introducción a la administración de sistemas* [en línea] URL <<http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/ch-acctsgroups.html>>
- ⁴ Linuxtopia. *Introducción a la administración de sistemas* [en línea] URL <http://www.linuxtopia.org/online_books/espaniol/redhat_linux_sysadmin_intro/ch-acctsgroups.html >
- ⁵ Joomla. *Contraseñas robustas* [en línea] URL <<http://comunidadjoomla.org/component/content/article/147-manual-de-instalacion-para-joomla-15x.html?start=6>>
- ⁶ Wikipedia Foundation Inc. *Contraseñas* [en línea] URL <<http://es.wikipedia.org/wiki/Contrase%C3%B1a>>
- ⁷ McAfee Inc. *Control de acceso y monitoreo de usuarios* [en línea] URL <http://www.mcafee.com/mx/enterprise/solutions/system_protection/access_control_and_user_monitoring.html>
- ⁸ Red Hat Inc. Red Hat Enterprise Linux 4. *Administración de recursos de usuarios* [en línea] URL <<http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/s1-acctsgroups-res.html>>
- ⁹ Flores Salgado, Lourdes Yolanda. Apuntes del curso: *Tópicos Selectos de Administración de Supercómputo*, Departamento de Supercómputo, DGSCA, UNAM, México, 2007.
- ¹⁰ Alcántar Aguirre, Aldo Axel, *Programación I, Trabajo de la Primera Unidad*, Instituto Tecnológico de Durango, [en línea] URL <<http://www.scribd.com/doc/249383/Porgramacion-I-Introduccion>>
- ¹¹ Flynn, M., *Some Computer Organizations and Their Effectiveness*, IEEE Trans. Comput., Vol. C-21, pp. 948, 1972, [en línea] URL <http://es.wikipedia.org/wiki/Taxonom%C3%ADa_de_Flynn>
- ¹² Herrera Revilla, Ismael, et al. Apuntes del Taller: *Aplicaciones del Cómputo en Paralelo a Sistemas Continuos*, Semana de Cómputo Científico 2006, DGSCA, UNAM, México, 2006. [PDF] URL <<http://www.mmc.igeofcu.unam.mx/SemanaDeComputoCientifico2006/AplicacionesDelComputoParaleloASistemasConti-nuos.pdf> >
- ¹³ García Regis y Cruz Martínez, *Paralelización en la Supercomputadora "Cray Origin 2000"*, DGSCA, UNAM, 2003, [PDF] URL <<http://www.dynamics.unam.edu/DinamicaNoLineal/Proyectos/Supercomputo/Paralelizacion.pdf>>
- ¹⁴ Anónimo, *Sistemas Distribuidos*, Instituto Nacional de Astrofísica, Óptica y Electrónica, [PDF] URL <http://ccc.inaoep.mx/~lamorales/distribuidos/Sistemas_Distribuidos.pdf >
- ¹⁵ García Regis, Oscar Rafael y Cruz Martínez, Enrique. *Supercómputo*, 2003-I, pág. 5, [PDF] URL <<http://132.248.28.115/DinamicaNoLineal/Proyectos/Supercomputo/Supercomputo.pdf>>
- ¹⁶ Flores Salgado, Lourdes Yolanda. Apuntes del curso: *Tópicos Selectos de Administración de Supercómputo*, Departamento de Supercómputo, DGSCA, UNAM, México, 2007.
- ¹⁷ Cruz Martínez, Enrique y Gordillo Ruiz, José Luís. "Supercómputo en la UNAM", *Entérate en Línea*, México, Noviembre 2002, [en línea] URL <<http://www.enterate.unam.mx/Articulos/2002/noviembre/supercom.htm>>
- ¹⁸ García Regis, Oscar Rafael y Cruz Martínez, Enrique. Op. cit., pág. 6
- ¹⁹ Wikimedia Foundation, Inc., *Cluster*, [en línea] URL <[http://es.wikipedia.org/wiki/Cluster_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Cluster_(inform%C3%A1tica))>
- ²⁰ Gordillo Ruiz, José Luís, "Diez años de computación con clusters", *Entérate en línea*, México, Marzo 2004, [en línea] URL <<http://www.enterate.unam.mx/Articulos/2004/Marzo/clusters.htm>>
- ²¹ Caballero Cruz, Reyna Elizabeth, et al. *Apuntes del Taller Laboratorio de Clusters*, Semana de Cómputo Científico 2006, DGSCA, UNAM, México, 2006.
- ²² Wikimedia Foundation, Inc., op cit.
- ²³ Abarca Castro y Mellado G, *Sistemas Distribuidos. Estructuración de un Cluster*, Temuco, Noviembre 2008, [PDF] URL <<http://xxito.files.wordpress.com/2008/11/trabajo-final-beowulf.pdf> >
- ²⁴ Ídem.
- ²⁵ Wikipedia Foundation, Inc., *Interfaz de Paso de Mensajes*, [en línea] URL <http://es.wikipedia.org/wiki/Interfaz_de_Paso_de_Mensajes>
- ²⁶ Wikipedia Foundation, Inc., *Máquina Virtual Paralela*, [en línea] URL <<http://es.wikipedia.org/wiki/PVM>>

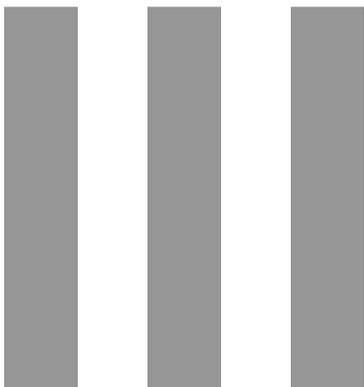
CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES

- ²⁷ Anónimo, *Sistema de colas*, Supercómputo, DGSCA, UNAM, [en línea] URL <http://www.super.unam.mx/index.php?option=com_content&task=view&id=65>
- ²⁸ Ídem.
- ²⁹ Anónimo. *Clusters: Qué son y para qué sirven?*, ChileHardWare [en línea] URL <http://www.chilehardware.com/guias_guia002-20040116.html>
- ³⁰ Adonay Gerónimo, César, et al. *Anteproyecto: Implementación de Clustering de servidores de alta disponibilidad y balanceo de carga*. Facultad de Ingeniería y Arquitectura. Universidad Centroamericana "José Simeón Cañas". 2006.
- ³¹ Ídem.
- ³² Ídem.
- ³³ Architecture share for 06/2008, *Lista del Top500*, [en línea] URL <<http://www.top500.org/stats/list/31/archtype>>
- ³⁴ Nikos Drakos y Ross More, *Construcción y evaluación de desempeño de un cluster tipo Beowulf para cómputo de alto rendimiento*, 1999, [en línea] URL <<http://hdgd.com.mx/~roadmr/tesis/escrito-1-split/node36.html>>
- ³⁵ Ídem.
- ³⁶ Murrieta León, Eduardo, apuntes del Taller: *Grids Computaciones*, Semana de Cómputo Científico 2006, DGSCA, UNAM, México, 2006, [PDF]
- ³⁷ Cruz Martínez, Enrique y Gordillo Ruiz, José Luis. *Supercómputo en la UNAM*. Entérate en línea. Año 7, Número 14, Noviembre del 2002. Dirección General de Servicios de Cómputo Académico, Universidad Nacional Autónoma de México, México, 2002. [en línea] URL <<http://www.enterate.unam.mx/Articulos/2002/noviembre/supercom.htm>>
- ³⁸ Angel, José de Jesús, *Curso gratis de Criptografía*, [en línea] URL <http://www.virusprot.com/cursos/Criptografia-Curso-gratis0.htm#Introduccion_Criptografia>
- ³⁹ Mungía Gutiérrez, Edgar, *Seguridad*, [en línea] URL <<http://www.fismat.umich.mx/~emurguia/mipagina/tesis/node21.html>>
- ⁴⁰ Tales-Oliang, Sergio, *Introducción a los Certificados Digitales*, InfoCentre, 2001 [PDF] URL <http://people.debian.org/~sto/articulos_bei/BEI-2003-11/certificados_digitales.pdf>
- ⁴¹ Mungía Gutiérrez, Edgar, op. cit.
- ⁴² Murrieta León, Eduardo, Apuntes del Taller: *Grids Computaciones*, Semana de Cómputo Científico 2006, DGSCA, UNAM, México, 2006, pág. 41, [PDF]
- ⁴³ Supercómputo, *Cluster Intel Pentium III. Arquitectura*, Supercómputo, DGSCA, UNAM, [en línea] URL <http://www.super.unam.mx/index.php?option=com_content&task=view&id=56&Itemid=90>
- ⁴⁴ CNS, Unpotosi. Supercomputadora Cray XD1, Centro Nacional de Supercómputo, IPICYT [en línea] URL <http://cns.ipicyt.edu.mx/index.php?option=com_content&task=view&id=51&Itemid=79>
- ⁴⁵ Supercómputo, *HP CP 4000 - Kanbalam*, Supercómputo, DGSCA, UNAM, [en línea] URL <http://www.super.unam.mx/index.php?option=com_content&task=view&id=35&Itemid=66>
- ⁴⁶ UAM, *Cluster Aitzaloo*, Laboratorio de Supercómputo y Visualización en Paralelo, UAMI, [en línea] URL <http://supercomputo.izt.uam.mx/index.php?id=Cluster_Aitzaloo>
- ⁴⁷ Enciclopedia ACE, *Políticas de depreciación y amortización*, [en línea] URL <<http://aceproject.org/aces/topics/et/admon-tec/financiamiento-de-la-tecnologia/politicas-de-depreciacion-y-amortizacion>>
- ⁴⁸ Ídem.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER



69	2.1 Requerimientos y Procedimiento para unirse a PRAGMA
73	2.2 Instalación del Sistema Operativo: Linux Rocks
106	2.3 Configuración del Middleware: Globus



CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

El capítulo que antecede nos proporcionó una visión general sobre el uso de tecnologías de cómputo intensivo. Asimismo se explicaron los conceptos fundamentales que sustentan el desarrollo de este proyecto y posteriormente se planteó la problemática que surge con motivo de la colaboración de la UNAM con los experimentos de PRAGMA.

Como ya se enunció en el capítulo anterior, la UNAM adquirió el compromiso de destinar y poner a punto un cluster que permita llevar a cabo pruebas relacionadas con el uso de tecnologías de grid. Con base en lo anterior, este capítulo presenta las actividades iniciales de instalación y configuración del cluster que se llevaron a cabo para cumplir con la primera parte de los requisitos establecidos por el Centro de Operación Grid (GOC) de PRAGMA, que incluyen, a *grosso modo*:

- Proveer un sistema con al menos dos computadoras, una utilizada como frontend y otra como nodo de cómputo. Para nuestro caso, el Departamento de Supercómputo facilitó diez nodos, de los cuales uno funge como frontend y los nueve restantes como nodos de cómputo.
- Instalar como sistema operativo una distribución de UNIX, de preferencia utilizar la distribución Rocks. Por lo que ese fue el sistema instalado.
- Instalación y configuración del middleware Globus.
- Instalación y configuración de una Autoridad Certificadora para pruebas.

Posteriormente la segunda parte de estos requisitos incluyen tareas de creación de cuentas de usuario, instalación de las aplicaciones requeridas, con las respectivas pruebas para verificar su correcto funcionamiento y configuración de actualizaciones automáticas de certificados, actividades que se expondrán en el capítulo 3.

2.1 Requerimientos y Procedimiento para unirse a PRAGMA

Los requerimientos que PRAGMA ha establecido para formar parte de la Grid, son los siguientes¹:

1. Destinar un sistema de cómputo el cual pueda ser usado para el proyecto PRAGMA y satisfacer los siguientes requerimientos mínimos.

Contar con al menos:

- Un nodo frontend (maestro).
- Un nodo de cómputo (La mayoría de las aplicaciones trabajan mejor con 8 o más nodos).
- Una conexión de Internet al nodo maestro (frontend).
- 256mb RAM en cada nodo.
- 20gb de espacio en disco duro en cada nodo.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

El equipo deberá tener instalada y funcionando:

- Cualquier distribución de UNIX, preferentemente Rocks, incluyendo todos sus rolls (serie de herramientas, tales como: area51, grid, java y SCE*).
- Globus 4.x con todo el conjunto SDK (datos, GRAM, info).
- Un despachador de trabajos, ya sea SGE*, PBS* o LSF*.
- OpenBabel 2.x
- Gamess

2. Notificar a la Coordinadora de PRAGMA (Cindy Zheng), el nombre del Administrador y del Soporte Técnico del cluster.

3. Crear una cuenta de usuario para el monitoreo de pruebas PRAGMA. Para esto se requiere:

- Descargar la información de la cuenta de usuario `user account info pak2`.
- Se recomienda usar el script de instalación de la cuenta de usuario PRAGMA, disponible en la página web de PRAGMA³.
- El script se debe ejecutar como root, como se muestra a continuación:
 - `#!/pragma_setuser zhengc.tar.gz`
 - Modificar el script según las características y políticas de sistema del sitio interesado en unirse a la grid.

4. Enviar la siguiente información a la Coordinadora de PRAGMA:

- Hostname* e IP* del cluster.
- Información de la cuenta de usuario que se creó para el monitoreo de pruebas de PRAGMA.
- Especificaciones técnicas del cluster.
- Archivos de la Autoridad Certificadora.
- Dirección de correo electrónico del Administrador del equipo.
- Dirección de correo electrónico del Soporte Técnico.

5. Para formar parte de la grid de PRAGMA, también se requiere:

- Instalar SCMSWeb*.
- Instalar OpenBabel y Gamess.

Con base en lo anterior se propuso la instalación, configuración y puesta a punto del Cluster Mixbaal del Departamento de Supercómputo. Para conseguir esto se siguió el siguiente procedimiento:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

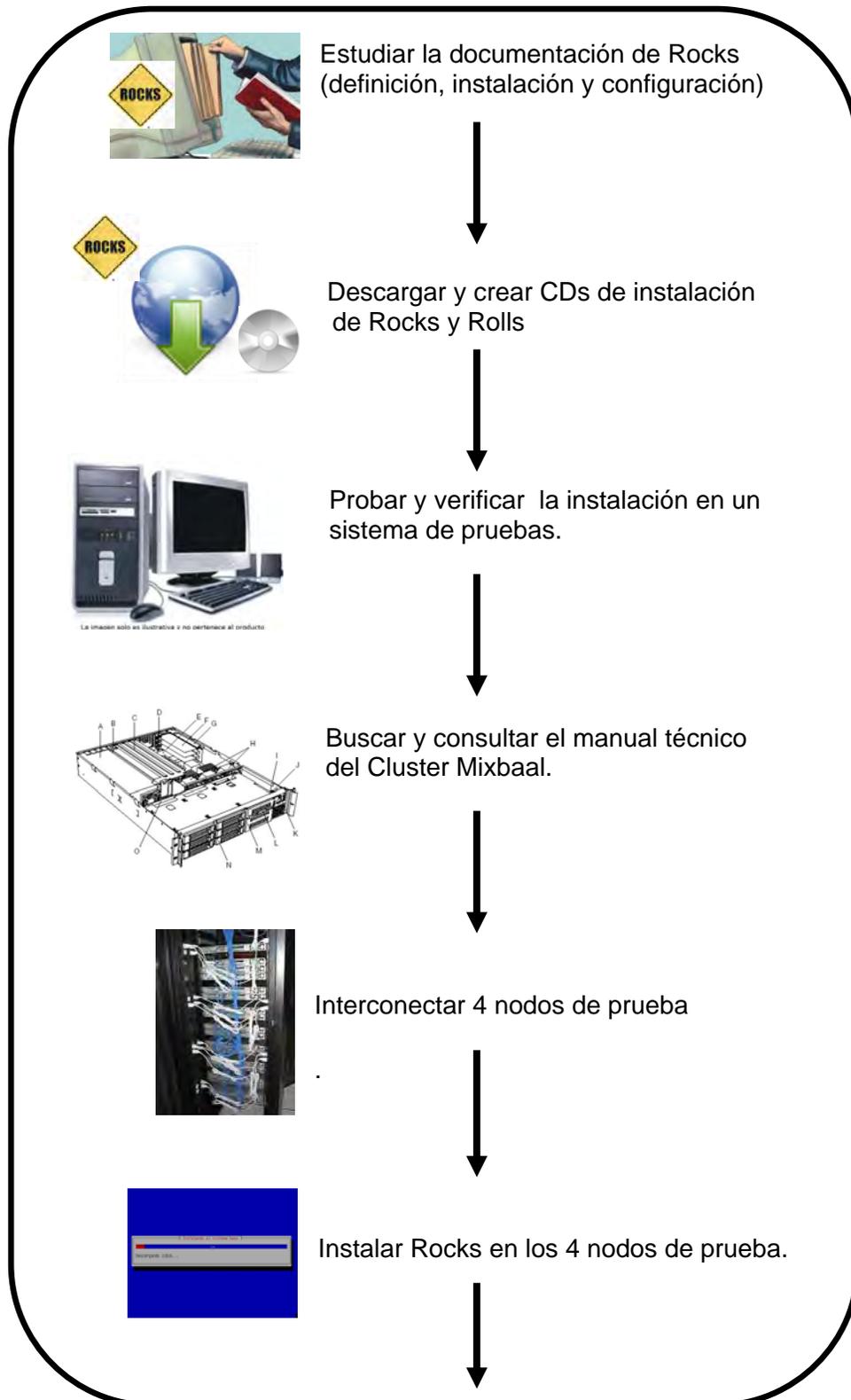


Figura 1a. Procedimiento para la instalación, configuración y puesta a punto del Cluster Mixbaal

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER



Figura 1b. Procedimiento para la instalación, configuración y puesta a punto del Cluster Mixbaal

2.2 Instalación del Sistema Operativo: Linux Rocks

2.2.1 Distribución Rocks

El sistema operativo que se ha instalado es Rocks, el cual es sugerido por el GOC, además de ser el representante de las soluciones mayormente implementadas en ambientes de High Performance Computing (HPC), a pequeña y gran escala.

Rocks and Rolls, es una distribución Linux (Rocks) y un conjunto de herramientas (Rolls), especialmente diseñadas para facilitar la administración de sistemas de cómputo, almacenamiento y visualización distribuida con un mínimo costo administrativo y un alto nivel de flexibilidad y personalización.

A continuación se describen las características particulares de Rocks, su organización y la forma en que hace uso de los recursos, aspectos que serán detallados en las subsecciones Sistemas de Archivos, Sistemas de Archivos por Red y Servicio 411.

Descripción

Jorge Zuluaga nos habla más a profundidad sobre Rocks:

Es una distribución basada en Red Hat Linux que cuenta con una amplia comunidad de desarrolladores y usuarios que garantizan una constante actualización en los paquetes y la disposición de mucho material de ayuda. Asimismo, cuenta con un mecanismo eficiente para realizar instalaciones y configuraciones no supervisadas en una máquina denominado 'Kickstart', mismo que es parte básica para el funcionamiento de los procedimientos de automatización de instalación en Rocks. Con este mecanismo el instalador de cada nodo:

1. Construye, usando las especificaciones publicadas en el frontend, un archivo de Kickstart especialmente diseñado para ese nodo.
2. A partir de ese archivo de Kickstart y conectándose al servidor web del frontend descarga los paquetes que se van a instalar.
3. Los instala.
4. Realiza configuraciones post-instalación.

Una de sus principales características es la forma en la que realiza la sincronización entre nodos, la cual consiste en reinstalar completamente los nodos cuando se produce una actualización en la plataforma. Esta filosofía de Rocks es llamada informalmente 'fire and forget', misma que ha sido ampliamente debatida por la comunidad de HPC. Sin embargo se ha demostrado que si bien es una decisión extrema, es comparativamente, muy sencilla y sobre todo muy efectiva para mantener la sincronización al más alto nivel.⁴ Cabe señalar que Rocks posee un mecanismo con el

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

que al instalar nuevamente cada nodo, evita la pérdida de datos, pues solo modifica la partición que contenga los binarios ya sea a instalar o a modificar. Así la nueva aplicación se agrega a la distribución que será reinstalada en los nodos.

Rocks ofrece un ambiente altamente unificado en cuanto al montaje, instalación, configuración y mantenimiento de un cluster, por lo que el administrador no necesita un conocimiento a profundidad de cada aplicación para poder manejarlas, situación muy diferente a la forma en cómo se efectuaba anteriormente, donde era imprescindible dominar todas las herramientas utilizadas para llevar a cabo todas estas tareas por separado.

Ahora bien, con el objeto de ofrecer una adecuada flexibilidad en la configuración inicial del sistema, Rocks permite configurar desde la instalación del frontend el perfil que tendrá la plataforma. Esto se hace mediante unos paquetes especialmente preparados y configurados, conocidos como Rolls.

Sobre los Rolls, Jorge Zuluaga nos explica lo siguiente:

Rolls es el nombre que reciben las diferentes aplicaciones que forman parte de un paquete de herramientas las cuales han sido diseñadas para la administración de sistemas, el almacenamiento y la visualización distribuida con un costo administrativo reducido y un alto nivel de flexibilidad y configuración personalizada de acuerdo a las necesidades particulares. La instalación de un sistema usando Rocks implica la elección del conjunto de Rolls que deben instalarse de acuerdo a las necesidades. Esto permite desde el principio contar con un frontend que tiene solamente los paquetes que necesita y que no requiere la desinstalación de paquetes que para el propósito con el que es creada la plataforma no son necesarios.⁵

Estos son:

- Base Roll: Este roll es el que permite comenzar con la instalación de Rocks.
- Kernel Roll: Este roll es el que contiene el kernel* del sistema que será instalado.
- HPC Roll: Uno de los rolls más importantes, mismo que es instalado con Rocks, cuenta con herramientas estándar para utilizar la plataforma para cómputo distribuido, además de permitir que la configuración del sistema se prepare en automático para que todos los nodos compartan la información de configuración, tal como información de autenticación de los usuarios, sin que sea necesaria una intervención administrativa o de configuración posterior a la instalación.⁶
- Area 51 Roll: Contiene utilerías y servicios para analizar la integridad de los archivos y el kernel* del cluster. Los paquetes que contiene son: Tripwire* y Chkrootkit*.⁷

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

- Bio Roll: Es una colección de algunas de las más comunes herramientas de bio-informática utilizadas hoy en día. Utiliza las técnicas de matemáticas aplicadas, informática, estadística y ciencias de la computación para la resolución de problemas de Biología. El desarrollo de este roll es un intento por estandarizar y facilitar la instalación de estas herramientas.⁸
- Condor Roll: Este roll utiliza la última versión estable de Condor para proveer un ambiente de Cómputo de alto Rendimiento en Rocks Cluster. La meta del proyecto Condor es desarrollar, implementar, poner en marcha, y evaluar los mecanismos y políticas que asisten a HPC en grandes colecciones de recursos computacionales de propiedad distributiva.
- Ganglia Roll: Sistema de monitoreo distribuido para cómputo de alto rendimiento, de la Universidad de California en Berkeley (UCB)
- Grid Roll: Está basado en Globus Toolkit 4 (GT4). Debido a que esta herramienta funge como middleware del cluster y forma parte de los requerimientos de PRAGMA, más adelante, en la sección 2.3.1, se describirá su funcionamiento y configuración en Mixbaal.
- Java Roll: Permite instalar el Software Development Kit de Java (SDK) y la Máquina Virtual de Java (JVM).
- SGE Roll: Este Roll instala y configura el despachador de tareas SUN Grid Engine. El proyecto Grid Engine es un esfuerzo de la comunidad open source* para facilitar la adopción de soluciones de cómputo distribuido. Es patrocinado por Sun Microsystems y hospedado por CollabNet. Es un proyecto que provee de software para la administración de recursos, desde clusters hasta cómputo en grid.
- Web Server Roll: Instala Apache como servidor web.
- SCE Roll: Es un proyecto open source* para ambientes distribuidos y escalables, permite efectuar el monitoreo de manera rápida y efectiva, reduce la complejidad en el mantenimiento de un cluster tipo Bewoulf, provee de un ambiente amigable para el usuario lo que se traduce en incremento de la productividad. Es una integración de herramientas que interactúan entre si, como SCMS* y SCMSWeb*. Está desarrollado para el monitoreo de sistemas distribuidos a gran escala, como es el caso de las grids. Permite detectar problemas de conectividad, de cargas no equitativas, así como determinar el rendimiento obtenido. El objetivo de este sistema es el desarrollo de tecnologías y herramientas de software que permitan al administrador de sistemas, monitorear el estado de sistemas ampliamente distribuidos y sistemas de cómputo en grid.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Desde el surgimiento mismo de la herramienta en el año 2000 el número de usuarios de Rocks, es decir de clusters instalados y configurados usando la herramienta, crece a un ritmo constante tal y como lo muestra el propio sistema de registro de Rocks al que se puede acceder a través de su sitio web <http://www.rocksclusters.org>.

Actualmente Rocks es utilizado en más de cien mil CPUs alrededor del mundo (115599 registrados para la fecha en la que se redactó este trabajo). Dentro de la lista están registrados clusters que tienen desde dos máquinas hasta grandes plataformas con más de 1200 nodos.

Otro de los escenarios en donde Rocks es utilizado como la distribución favorita, es en el montaje de sistemas grid. Se han desarrollado Rolls particulares para crear instalaciones homogéneas de herramientas y servicios en los clusters que se conectan a la ciberinfraestructura*.

Por otro lado es importante hablar de cómo está organizado y cómo trabaja este sistema operativo, aspectos que se verán en los siguientes tres subtemas.

Sistema de Archivos

Un Sistema de Archivos es una estructura lógica que permite organizar la forma en que se encuentra almacenada la información en una unidad de almacenamiento.⁹

Jorge Zuluaga explica que para Rocks, el disco duro tanto del frontend como de los nodos vienen particionados generalmente en 2 o 3 particiones:

- 1) La partición raíz (montada en el directorio /) contiene los archivos del sistema operativo y el espacio de almacenamiento de archivos temporales, logfiles*, archivos de configuración, etc.
- 2) Una partición especial para almacenamiento masivo en el frontend y en cada nodo que normalmente se monta sobre el directorio `/state/partition1`. En el frontend, esta partición contiene las cuentas de usuario y otros archivos importantes relacionados con la instalación del sistema operativo, mientras que en los nodos esta partición puede usarse libremente para almacenar localmente grandes volúmenes de información. Es importante mencionar que localmente en el frontend el archivo `/state/partition1` tiene asociado el enlace simbólico `/export` como se muestra a continuación:

```
[root@mixbaal ~]# ls -ld /export
lrwxrwxrwx  1 root root 16 Apr 12  2008 /export -> state/partition1
```

De modo que en lo sucesivo hablaremos directamente de `/export` en lugar de `/state/partition1`.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

- 3) En las últimas versiones de Rocks (posteriores a la versión 4.2) se ha incluido en el esquema de particionado por defecto una partición que se monta sobre el directorio `/var` el cual normalmente contiene información generada por los distintos programas y servicios del sistema operativo, incluyendo los logfiles*.¹⁰

Un ejemplo del esquema de particionado del frontend y de los nodos, configurado por defecto, se muestra a continuación:

```
[root@mixbaal ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       7.7G  5.6G  1.8G  76% /
/dev/hda5       25G   4.5G   19G  20% /state/partition1
/dev/hda2       3.9G  1.1G  2.7G  29% /var
```

Cada partición del disco duro, a su vez, cuenta con su propio Sistema de Archivos. De los Sistemas de Archivos locales del frontend vale la pena resaltar los siguientes directorios de gran relevancia para el cluster¹¹:

- `/export/home (/state/partition1/home)`: Este directorio contiene los directorios hogar (home) de los usuarios, el directorio del usuario `condor` donde se depositan importantes archivos del sistema de colas* de Condor y el directorio `install` que se describirá a continuación.
- `/export/home/install`: Este directorio contiene la totalidad de los paquetes de instalación, archivos de configuración, programas y scripts especiales que usa Rocks para realizar la instalación del sistema operativo en los nodos. De los subdirectorios que encontramos en este directorio son de resaltar los siguientes:
 - `rolls`: contiene todos los paquetes instalados con los Rolls agregados al sistema desde la instalación o en un momento posterior a ella.
 - `site-profiles`: aquí se almacenan distintos archivos de configuración usados para personalizar la instalación de los nodos de cómputo en el cluster.
 - `rocks-dist`: este subdirectorio, cuyo contenido es creado con el comando `rocks-dist`, contiene los archivos finales que serán instalados en los nodos del cluster. Al conjunto de estos archivos se lo conoce en Rocks como la 'Distribución'.

Sistemas de Archivos por Red

Además de los Sistemas de Archivos locales (asociados a las particiones en el disco duro del frontend y de cada nodo) Rocks maneja un conjunto de Sistemas de Archivos que se montan por red.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

En un sistema de computación con clusters, es absolutamente necesario que esos archivos puedan ser accedidos cuando se realiza una conexión remota a uno de los nodos de cómputo del cluster. Para esto Rocks emplea dos mecanismos básicos para garantizar que el usuario tenga acceso directo a sus archivos aún si se encuentra conectado a uno de los nodos de cómputo, los cuales Jorge Zuluaga explica a continuación:

- NFS (Network filesystem): monta el sistema de archivos de la cuenta del usuario a través de la red en el nodo al que se conecta. Y los cambios que se hacen sobre el sistema de archivos, ya sea en el frontend o en los nodos, se actualizan automáticamente en el disco duro del frontend donde residen realmente.
- Servicio `autofs` (Automounter): garantiza que el montaje de los sistemas de archivos sea automático y ocurra al momento de acceso del usuario, así como también desmonta el sistema de archivos cuando el usuario deja de utilizarlo.¹²

El montaje de Sistemas de Archivos por red usando NFS tiene un costo en el uso de la red que debe minimizarse en un entorno de computación distribuida. Por esa razón se usa adicionalmente el servicio `autofs`, que permite el montaje automático de solo aquellos directorios de los Sistemas de Archivos por red que son requeridos por el usuario en los nodos.

La lista de los Sistemas de Archivos del frontend que pueden ser montados en los nodos puede consultarse con el comando `exportfs` con la opción `-v`:

```
[root@mixbaal ~]# exportfs -v
/state/partition1
      10.1.1.0/255.255.255.0(rw,async,wdelay,root_squash)
```

Como se puede observar, el único Sistema de Archivos disponible para todos los nodos, es el que se encuentra en la partición `/state/partition1`, cuyo contenido regularmente es:

```
[root@mixbaal ~]# ls -l /export/
total 36
drwxr-xr-x  11 root    root   4096 Aug 22 17:26 apps
drwxr-xr-x  15 root    root   4096 Jul 18 19:33 home
drwxr-xr-x   3 root    root   4096 Apr 12  2008 site-roll
```

Para montar automáticamente los directorios contenidos en `/export` sobre los nodos del cluster se configura el sistema `autofs` a través de los archivos `/etc/auto.master`, `/etc/auto.home`, `/etc/auto.share`. Normalmente estos archivos deben residir en el directorio `/etc` de todas las máquinas del cluster incluyendo el propio frontend.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

El contenido de estos archivos, regularmente, es como sigue:

```
[root@mixbaal ~]# more /etc/auto.master /etc/auto.home
/etc/auto.share

::::::::::::
/etc/auto.master
::::::::::::
/share /etc/auto.share --timeout=1200
/home /etc/auto.home --timeout=1200

::::::::::::
/etc/auto.home
::::::::::::
install      mixbaal.local:/export/home/&
condor      mixbaal.local:/export/home/condor
perla       mixbaal.local:/export/home/perla
paulina     mixbaal.local:/export/home/paulina

::::::::::::
/etc/auto.share
::::::::::::
apps mixbaal.local:/export/&
install mixbaal.local:/export/home/&
```

El archivo `auto.master` contiene la lista de los directorios sobre los que se montan los Sistemas de Archivos por red descritos en detalle en los archivos de configuración respectivos. Así por ejemplo en cada máquina del cluster sobre el directorio `/home` se montan Sistemas de Archivos con las reglas descritas en el archivo de configuración `/etc/auto.home`. La opción `--timeout` permite definir un tiempo de espera después del cual, si el Sistema de Archivos no está siendo utilizado por el usuario en la máquina remota entonces se desmonta automáticamente para liberar la sobrecarga de red que este procedimiento produce.¹³

De estos archivos el `/etc/auto.home` es probablemente el más importante. En él se definen, entre otras, las reglas para montar los directorios hogar (home) de los usuarios en los nodos.

La sintaxis general de las entradas de los archivos del tipo `/etc/auto.*` es:

<directorio-local> <servidor-nfs>:<sistema-de-archivos>/<directorio-remoto>

Donde el `<directorio-local>` es el nombre del directorio sobre el que se deberá montar el `<directorio-remoto>` que en la máquina `<servidor-nfs>` está sobre el Sistema de Archivos `<sistema-de-archivos>`.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Así, por ejemplo, cuando el usuario se conecta a cualquier nodo del cluster su directorio hogar (/home/fulano) es automáticamente montado vía NFS desde el directorio /export/home/fulano que se encuentra en el frontend.

Por otra parte el archivo de configuración /etc/auto.share permite configurar, el montaje automático del directorio /export/apps, donde se pueden instalar o almacenar archivos que requieran ser accedidos desde todos los nodos del cluster.

Según el archivo /etc/auto.master el directorio /export/apps es montado automáticamente en cada nodo en el directorio /share/apps. Este directorio es particularmente útil cuando se desea instalar aplicaciones, bibliotecas, bases de datos, etc. que se quiere acceder desde todos los nodos del cluster.

Cuando se realizan cambios en los archivos de configuración del servicio `autofs`, es decir, los tres mencionados hace un momento, el servicio debe reiniciarse. Esto se realiza usando el comando `service` de Linux como se ilustra a continuación:

```
[root@mixbaal ~]# service autofs reload
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
```

Se puede recargar el servicio también en otros (o todos) los nodos del cluster usando el comando `cluster-fork`:

```
[root@mixbaal ~]# cluster-fork service autofs reload
compute-0-0:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
compute-0-1:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
...
```

`cluster-fork` pertenece a los comandos de la familia `cluster`, al igual que los comandos: `cluster-kill`, `cluster-ps` y `cluster-probe`. Mismos que son una de las herramientas más útiles proveídas por Rocks, ya que permiten ejecutar masivamente comandos o acciones en todos los nodos del cluster.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Cuando se ejecuta un comando de la familia cluster es posible indicarle los nodos sobre los que deberá ejecutarse la acción o el comando seleccionado. Para hacerlo se debe usar:

```
[root@mixbaal ~]# cluster-fork -n "c0-2 c0-3 c0-5" w
```

Donde, después de la opción `-n` y entre comillas, se coloca la lista de los nodos separada por espacios en blanco.

Servicio 411

Esta comunicación entre los nodos del cluster es de suma importancia para el funcionamiento correcto del mismo, situación que Rocks solventa a través del servicio 411.

Jorge Zuluaga nos habla más sobre este servicio:

El sistema de información 411 Secure Information System (Sistema de Información Segura), mantiene automáticamente actualizados los archivos de configuración del sistema y permite también al administrador ejecutar una actualización forzada a través de todo el cluster.

411 garantiza la seguridad de la información que circula a través de la red de la plataforma usando para ello canales seguros con cifrado del tipo SSL*/TLS*.

Además permite que los sistemas de archivos del frontend puedan ser accedidos desde los nodos y que los usuarios puedan conectarse con una sola cuenta de usuario sin necesidad de ingresar alguna contraseña para acceder a los nodos del cluster.

Asimismo, permite que archivos de configuración vitales para los servicios del cluster (listas de usuarios, tabla de contraseñas, grupos, configuración del servicio `autoofs`, entre otras) sean compartidos por todas las máquinas de la plataforma, garantizando además que se mantengan sincronizados en la operación del cluster.¹⁴

Para dar un ejemplo del papel que juega este servicio, cada vez que se crea un usuario en el frontend se agrega una entrada a los archivos `/etc/passwd`, `/etc/shadow`, `/etc/group` y `/etc/auto.home`, 411 se encarga de que esos archivos se propaguen por todos los nodos del cluster de modo que el nuevo usuario pueda conectarse sin problemas a esos nodos. 411 utiliza servicios web para publicar en el frontend, de manera cifrada, los archivos de configuración que serán compartidos y descargados en cada nodo.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Esto implica que para que el servicio funcione debe garantizarse que el servidor web del frontend este funcionando. Para verificarlo se puede usar el comando:

```
[root@mixbaal ~]# service httpd status
```

Si el servicio esta activo la salida sería:

```
httpd (pid 30585 30584 30583 30582 30581 30580 30579 30578 30575) is
running...
```

De lo contrario se obtendría:

```
httpd is stopped
```

En este caso es necesario iniciarlo:

```
[root@mixbaal ~]# service httpd start
Starting httpd: [ OK ]
```

La lista de los archivos que se comparten a través del servicio 411, se puede encontrar en el archivo `/var/411/Files.mk`, cuyo contenido puede ser como sigue:

```
[root@mixbaal ~]# more /var/411/Files.mk
...
AUTOMOUNT = #(wildcard /etc/auto.*)
# These files all take a "#" comment character.
# If you alter this list, you must do a 'make clean; make'.
FILES = #(AUTOMOUNT) \
/etc/passwd \
/etc/shadow \
/etc/group \
/etc/services \
/etc/rpc
# FILES += /my/file
```

El servicio 411 esta programado para realizar en forma automática la sincronización de los archivos de configuración en el cluster, como se mencionó al inicio de esta sección. Sin embargo en algunas situaciones es necesario forzar la sincronización después de que se ha hecho un cambio en los archivos de configuración (creación de un usuario, modificación de la configuración del servicio `autofs`, creación de un nuevo grupo, instalación de aplicaciones, etc.). La sincronización se puede realizar de tres maneras diferentes:

1) Usando el comando `service`:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
[root@mixbaal ~]# service 411 commit
Committing changes to login files using 411.
411 Wrote: /etc/411.d/etc.shadow
Size: 2727/1840 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1
```

Este mecanismo solamente actualiza los archivos de configuración que han cambiado recientemente.

2) Usando el comando make:

```
[root@mixbaal ~]# make -C /var/411
```

Que es equivalente a la forma 1) con la diferencia de que se puede forzar la sincronización de todos los archivos usando la opción `force`:

```
[root@mixbaal ~]# make -C /var/411 force
```

3) Usando el comando `411get` en los nodos:

```
[root@mixbaal ~]# cluster-fork 411get
compute-0-0:
/etc.rpc
/etc.group
/etc.shadow
/etc.auto..misc
/etc.auto..master
/etc.passwd
/etc.auto..home
/etc.auto..net
/etc.services
/etc.auto..share
...
```

Este mecanismo es particularmente útil para detectar problemas en la transferencia de los archivos entre el frontend y los nodos que pueden no detectarse con los mecanismos anteriores. Por ejemplo, si el servidor web se encuentra detenido, el comando anterior tendría como salida lo siguiente:

```
[root@mixbaal ~]# cluster-fork 411get
compute-0-0:
Error:    Could    not    reach    a    master    server.    Masters:
[http://10.1.1.1/411.d/ (-1)]
compute-0-1:
Error:    Could    not    reach    a    master    server.    Masters:
[http://10.1.1.1/411.d/ (-1)]
```

Problema que se solventaría iniciando el servicio `httpd` como se explicó más arriba.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Con esto se concluye la sección dedicada a explicar la manera de trabajar de Rocks y sus principales métodos de intercomunicación en el cluster, los cuales son vitales para el buen funcionamiento del equipo. Expuesto lo anterior pasamos a la parte práctica de la instalación y configuración del cluster.

2.2.2 Pre-Instalación

Los requisitos mínimos de hardware para instalar Rocks, según Jorge Zuluaga, son los siguientes:

- Un conjunto de máquinas de arquitectura similar (compute nodes), cada una con una interfaz de red, disco duro con capacidad para más de 7 GB y memoria RAM superior a 256 MB.
- Un switch (o varios) con un número de puertos mayor al doble del número de máquinas disponibles (para darle escalabilidad).
- Una máquina con 2 interfaces de red, capacidad en disco duro igual o superior a 20 GB, y memoria RAM superior o igual a 512 MB (frontend).
- Cables de red en número y longitud suficiente.¹⁵

La arquitectura de Mixbaal consiste en un nodo maestro y 9 nodos esclavos conectados en una topología de estrella.

Cada nodo esclavo tiene 2 procesadores con 1 Gbyte de memoria RAM y disco duro interno de 40 Gbytes. La comunicación entre los nodos se hace a través de una conexión tipo Gigabit Ethernet a 100 Mbs, que se logra por medio de un switch Catalyst 3550 de 24 puertos. La Tabla 1 muestra las características técnicas de Mixbaal.

Características técnicas	
Número y tipo de CPU's	20 procesadores Intel Pentium III a 1130.497 MHz
Rendimiento de procesador	1 GFLOPS
Rendimiento Total	20 GFLOPS
Organización CPUS	1 nodo maestro con 2 procesadores 9 nodos esclavos con 2 procesadores c/u
CACHÉ por procesador	512 Kb
Memoria por nodo	1 GB
Memoria total	10 GB
Interconexión	Switch Catalyst 3550 de 24 puertos
Unidades de disco internas	Cada nodo con 1 IDE 40 GB

Tabla 1. Características técnicas de Mixbaal

Como podemos observar, el cluster cumple con los requisitos mínimos.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Se realizaron las conexiones de la red externa y la red interna del cluster, tomando en cuenta que Rocks asume que la interfaz identificada como “eth1” por el kernel*, es aquella que está conectada a la red externa y la “eth0” a la red privada del cluster, como se muestra en la figura siguiente:

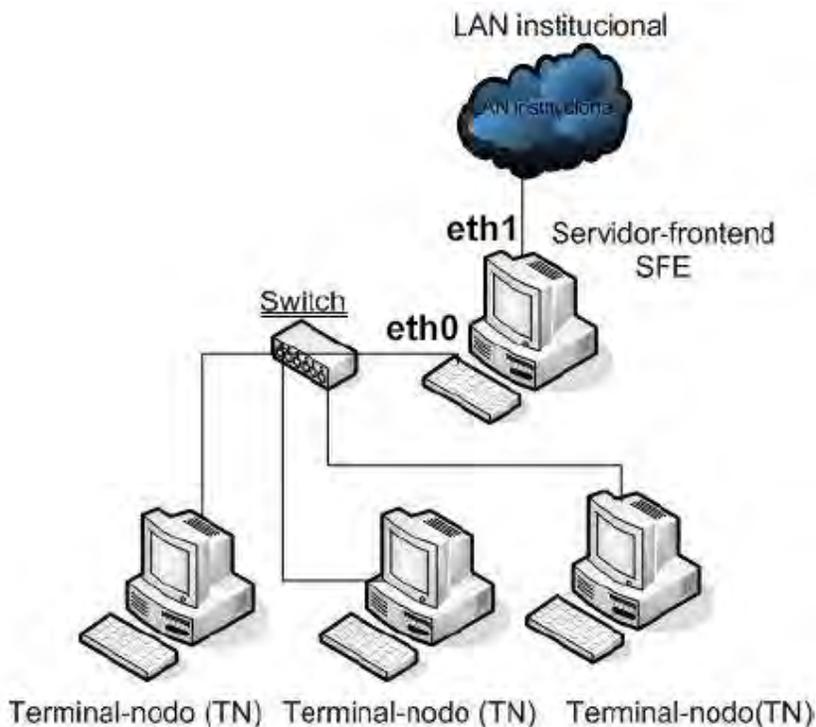


Figura 2. Esquema de conectividad del cluster

Cabe señalar que la manera como el kernel* asigna los identificadores eth0 y eth1 es relativamente compleja y varía con el fabricante y modelo de las tarjetas, la tarjeta madre, los puertos a los que están conectadas y la versión del kernel* de Linux que se esté utilizando. Normalmente en Linux es posible cambiar después de la instalación la manera como el sistema asigna los identificadores a las tarjetas de acuerdo a su MacAddress*. Sin embargo cuando se instala Rocks es importante que el orden de ellas sea el que se espera de acuerdo al esquema de conectividad del frontend (Ver Figura 2).

El síntoma de que existe un problema en el orden de identificación de las tarjetas de red aparece cuando en los primeros momentos de la instalación el proceso no logra obtener una dirección IP* de la red a la que esta conectada la interfaz eth1. En este caso podrían darse 2 escenarios:

1. La red externa tiene servicio de DHCP*. En este escenario el hecho de que al intentar obtener una IP* por parte del servidor no funcione es síntoma de que el orden de identificación no fue el correcto y la interfaz que va a la red externa seguramente esta identificada como eth0, siendo la correcta, eth1.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

2. La red externa no tiene servicio DHCP*. En este escenario no es posible identificar de forma inmediata cuál identificador se asigna a las interfaces. Para resolver el inconveniente podría ser posible usar un disco de arranque de Linux (Linux rescue, Linux Live CD) que use un kernel* de versión no muy distinta a la que usa Rocks (4.2.1 incluye un kernel* 2.6) e investigar el modo en el que los identificadores de las NIC* son asignados.

Los rolls mínimos para la instalación son:

- Kernel/Boot Roll CD
- Base Roll CD
- HPC Roll CD
- Web-Server Roll CD
- OS Roll CDs (disk 1 and disk 2)

Estos rolls se encuentran disponibles en la página web de Rocks¹⁶, de forma individual o bien, reunidos en un solo DVD a modo de combo:

Athlon, Pentium, Xeon i386



CDs	jumbo (DVD)
Kernel/Boot Roll	Boot, Core, & OS Roll dvd only
Core Roll	
OS Roll - disk1	
OS Roll - disk2	

Tabla 2. Rolls disponibles para descargar

Se escogió descargar el *jumbo(DVD)* que trae incluidos todos los rolls solicitados.

Por otro lado, como parte de los requerimientos de PRAGMA, se encuentra la instalación del roll SCE*, mismo que no está disponible directamente en la página de Rocks, sino en la de SCE*. Ahora bien para efectos de este proyecto fue necesario instalar la versión 4.3 de este roll, pero ésta aún no estaba disponible, de modo que se contactó con el programador encargado de este roll (Somsak Sriprayoonsakul), para que nos informará cuando sería liberada la versión 4.3. Una semana después, nos respondió vía correo electrónico, notificándonos que la versión ya estaba disponible, y se procedió con su descarga¹⁷.

2.2.3 Instalación

El proceso que se describirá en esta sección, se refiere a la instalación de la versión 4.3 de Rocks en un Cluster Intel Pentium III Beowulf, inicialmente en cuatro nodos a modo

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

de prueba, para posteriormente, proceder con la instalación en los diez nodos que conforman el cluster.

Instalación del frontend

A continuación se especificará la serie de pasos que se llevaron a cabo durante la instalación del nodo maestro (frontend) del cluster.

Después de elegir el nodo maestro y verificar que las conexiones de la red externa y la red interna del cluster se hicieran a la interfaz de red correcta, como se especificó en la sección 2.2.2 (Ver Figura 2), se inició la instalación, llevando a cabo el siguiente procedimiento:

1. Introducir el jumbo(DVD), descargado previamente, en el frontend del cluster y reiniciar el frontend.



Ningún nodo cuenta con unidad de CD/DVD-ROM, de modo que se consiguió una unidad de CD/DVD-ROM extraíble USB.



El jumbo(DVD) no arrancaba y el frontend no hacía nada.



Se ingresó al BIOS* del frontend para modificar la configuración de arranque, indicándole que iniciará con la unidad de CD-ROM.

2. Si el arranque es exitoso, aparece la siguiente pantalla:

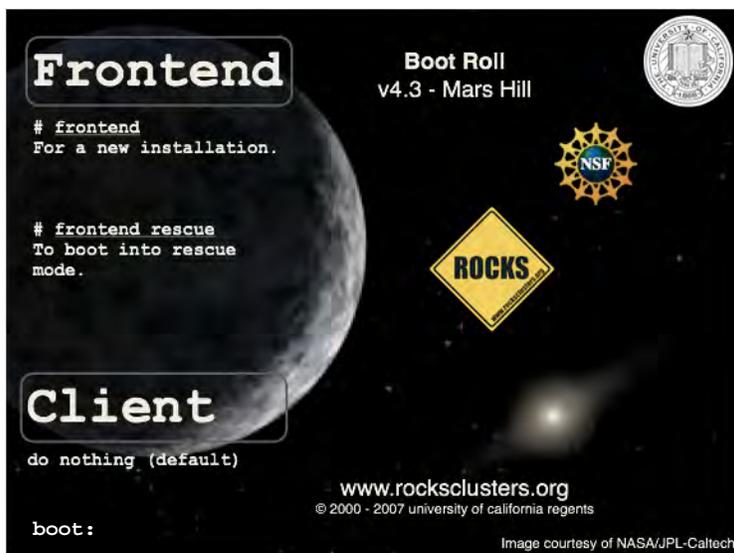


Figura 3. Instalación Rocks: Pantalla 1

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Se ingresa la palabra `frontend` en el prompt `boot`:



Se debe prestar atención, ya que si no se ingresa rápidamente la palabra `frontend`, el programa de instalación asumirá que se trata de un nodo cualquiera, fallando la instalación del frontend, obligando a reiniciar el nodo.

3. Una vez realizado lo anterior, el instalador inicia su ejecución, apareciendo la siguiente pantalla, en la que se permite seleccionar los rolls a instalar y que a su vez se encuentran disponibles en el jumbo(DVD):



Figura 4. Instalación Rocks: Pantalla 2

Los rolls seleccionados fueron: `area51`, `base`, `bio`, `condor`, `ganglia`, `grid`, `hpc`, `java`, `kernel`, `os`, `sgs` y `web-server`.

4. Al dar clic en `Submit` aparece la siguiente pantalla, en la que se muestran los rolls elegidos en el paso anterior:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

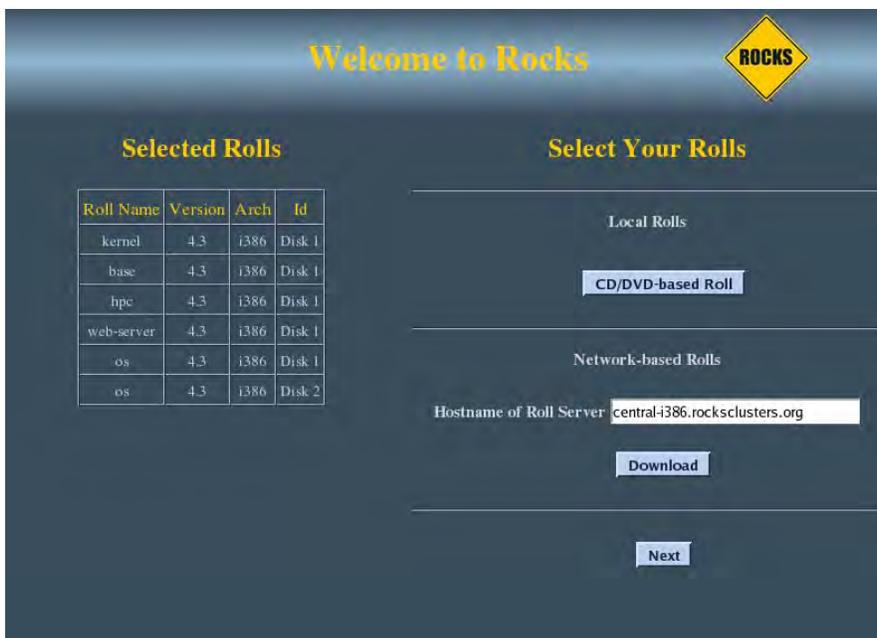


Figura 5. Instalación Rocks: Pantalla 3

5. Oprimir el botón `CD/DVD-based Roll`, para agregar rolls incluidos en otro CD/DVD. Así, el instalador expulsa el CD/DVD actual y se muestra la siguiente pantalla:

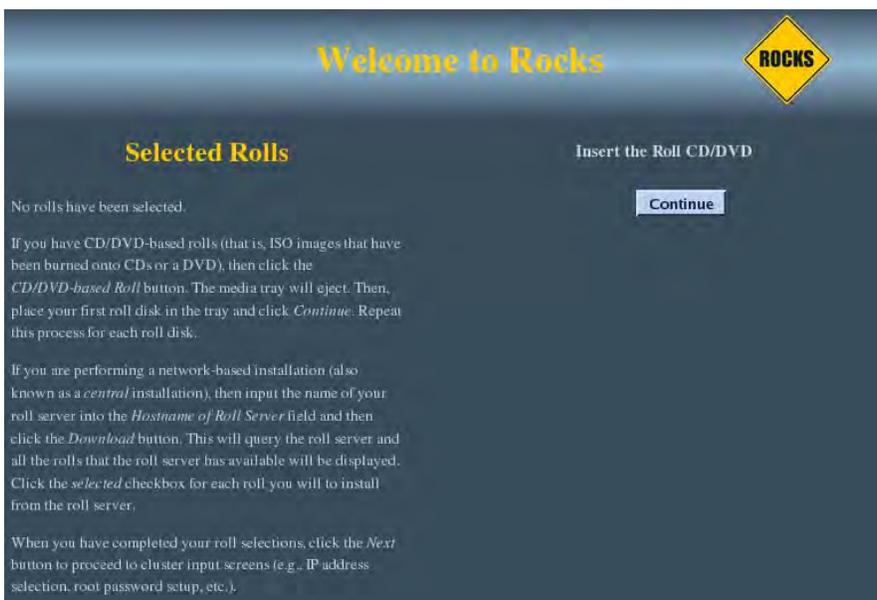


Figura 6. Instalación Rocks: Pantalla 4

Se inserta el CD/DVD con los rolls extra, se presiona `Continue` y aparece la siguiente pantalla mostrando los rolls extra.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

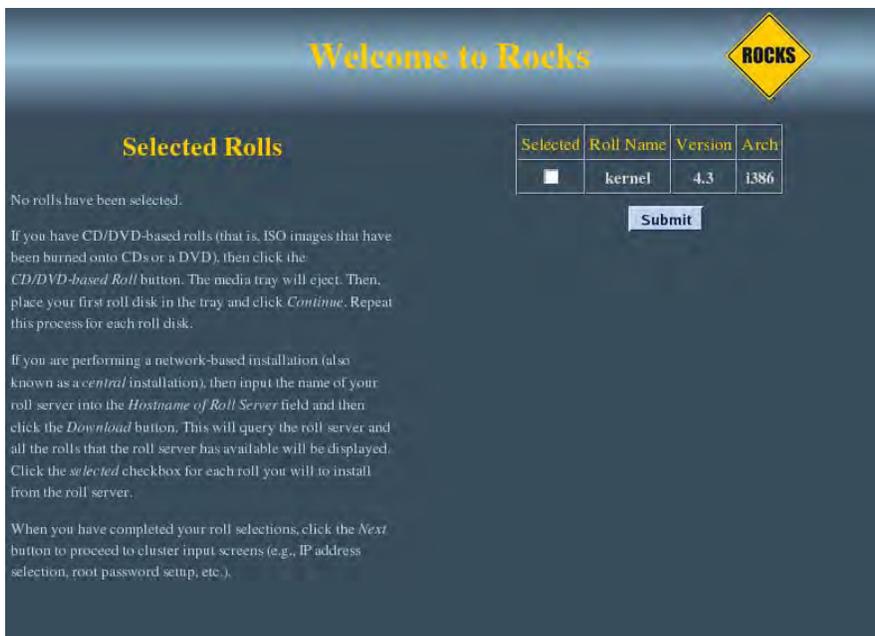


Figura 7. Instalación Rocks: Pantalla 5



Para este caso se insertó el CD que contiene el SCE roll y éste fue el que apareció en la pantalla anterior.

6. Cuando se terminaron de agregar los rolls, se presionó **Next** para continuar con la instalación. La pantalla inmediata corresponde al llenado de los datos generales del cluster:



Figura 8. Instalación Rocks: Pantalla 6

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Los datos ingresados fueron los siguientes:

- Nombre completo del frontend como será reconocido en la red externa. Para esto se consultó con el Administrador de Redes el nombre que se le daría al cluster en la red del Departamento de Supercómputo.

Fully-Qualified Host Name: `mixbaal.supercomputo.unam.mx`



El Hostname* se debe elegir cuidadosamente, ya que si éste se llega a modificar, servicios como Globus, NFS, SGE*, no serían capaces de encontrar al frontend, ocasionando un mal funcionamiento.

- Nombre del cluster. Éste es a elección del usuario y se usa en aplicaciones como Ganglio. Se pueden usar mayúsculas o minúsculas.

Cluster Name: `Mixbaal`

- Organización a la que pertenece el cluster.

Certificate Organization: `UNAM GridCA`

- Ciudad.

Certificate Locality: `México`

- Estado.

Certificate State: `México`

- Abreviatura del país. Para este se consultó la página web de Internacional Country Codes¹⁸.

Certificate Country: `MX`

- E-mail de contacto. Preferentemente el del administrador del cluster.

Contact: yoli@super.unam.mx

- URL de la página a la que se podrá acceder para monitorear el cluster vía web. Misma que una vez que se ponga a punto el cluster debe enviarse a Cindy Zheng.

URL: <http://mixbaal.supercomputo.unam.mx>

- Latitud y longitud geográfica de la locación en la que se encuentra el cluster

Latitude/Longitude: `N19.26 W99.719`

Una vez llenado el formulario, se presionó Next.

7. A continuación se procede a la configuración de la red privada del cluster permitiendo establecer los parámetros de red para la interfaz eth0, que conecta el frontend con los nodos de cómputo.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER



Figura 9. Instalación Rocks: Pantalla 7

Para esta sección se recomienda dejar los valores por defecto:

- IP address: 10.1.1.1
- Netmask: 255.0.0.0

8. Ahora se ingresan los parámetros para la configuración de red para la interfaz eth1 que conecta el frontend con la red externa:

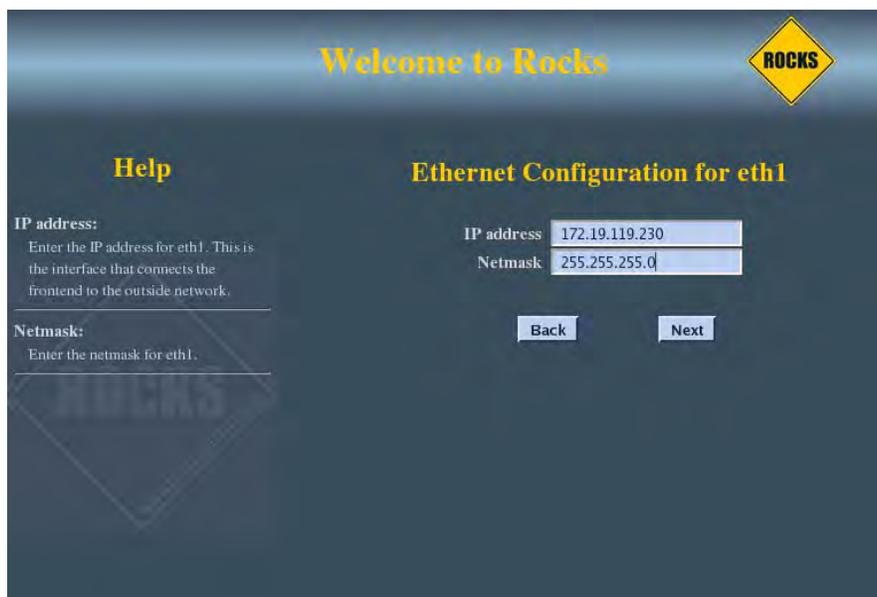


Figura 10. Instalación Rocks: Pantalla 8

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Los datos ingresados fueron:

- La dirección IP* fue la asignada por el Administrador de redes del Departamento de Supercómputo.
Ip address: 132.248.124.81
- Para la Máscara de red se contempló un rango pequeño debido al número de nodos en nuestro cluster.
Netmask: 255.255.255.240



No se obtuvo una dirección IP* de la red a la que está conectada la interfaz eth1, presentándose el escenario 1, mencionado en la sección 2.2.2.



Se abortó la instalación para intercambiar los cables de conexión ya que la interfaz que va a la red externa se identificó como eth0, siendo la correcta, eth1, posteriormente, se reinició el frontend, reiniciándose también la instalación de Rocks, misma que al llegar a este punto nuevamente, se logró obtener una dirección IP* dinámica, con éxito.

9. Configuración del Gateway* y DNS*.

Datos ingresados, proporcionados previamente por el Administrador de Redes del Departamento de Supercómputo:

- IP* del Gateway*
Gateway: 132.248.124.94
- Servicio de DNS*
DNS Servers: 132.248.10.2

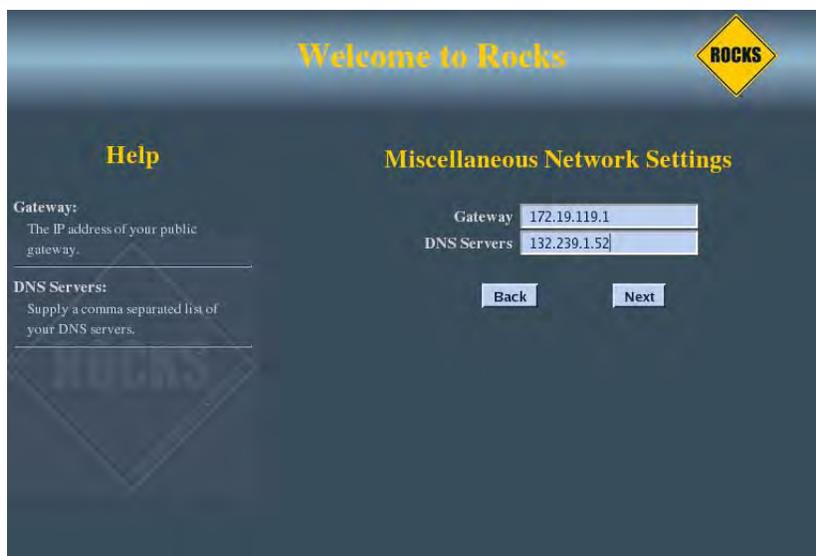


Figura 11. Instalación Rocks: Pantalla 9

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

10. Se estableció el password de root:

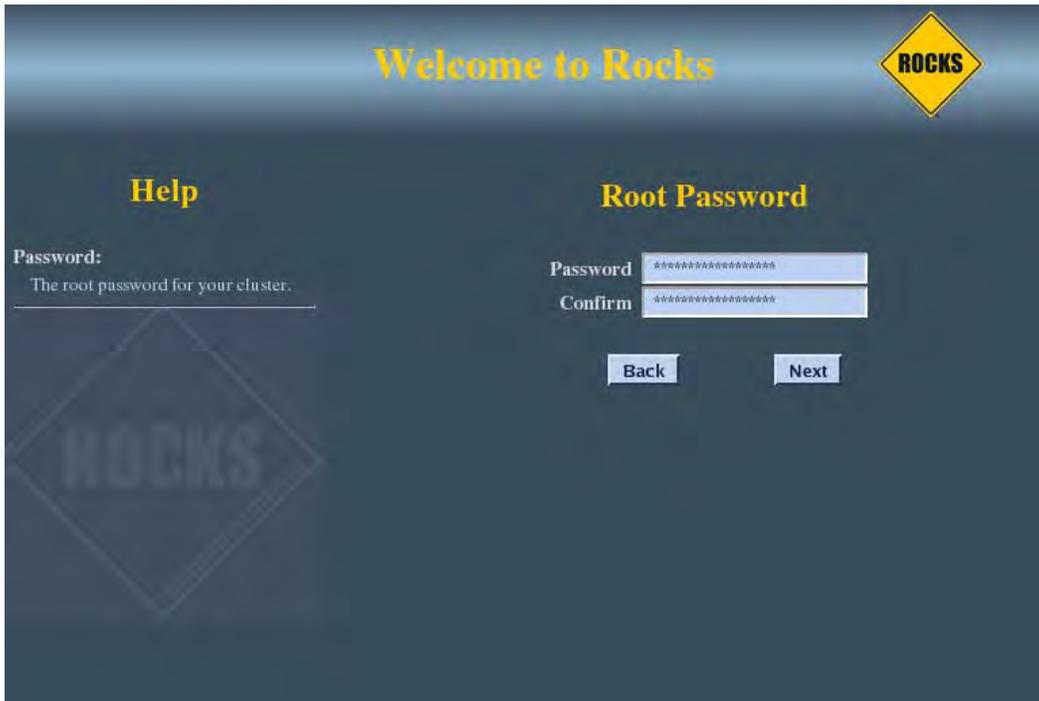


Figura 12. Instalación Rocks: Pantalla 10

11. Se configuró la zona horaria:

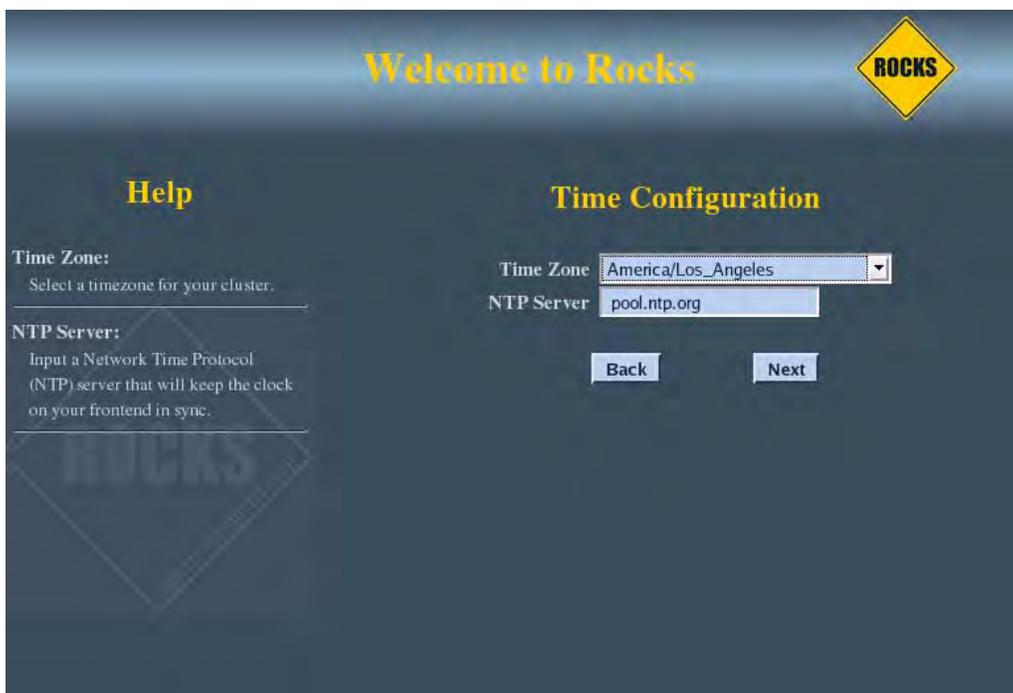


Figura 13. Instalación Rocks: Pantalla 11

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Ingresando:

- Time Zone: México, City
- NTP Server: cronos.cenam.mx

12. Particionado del disco duro. En esta pantalla se permite elegir cómo será el particionado del disco, ya sea automático o manual.

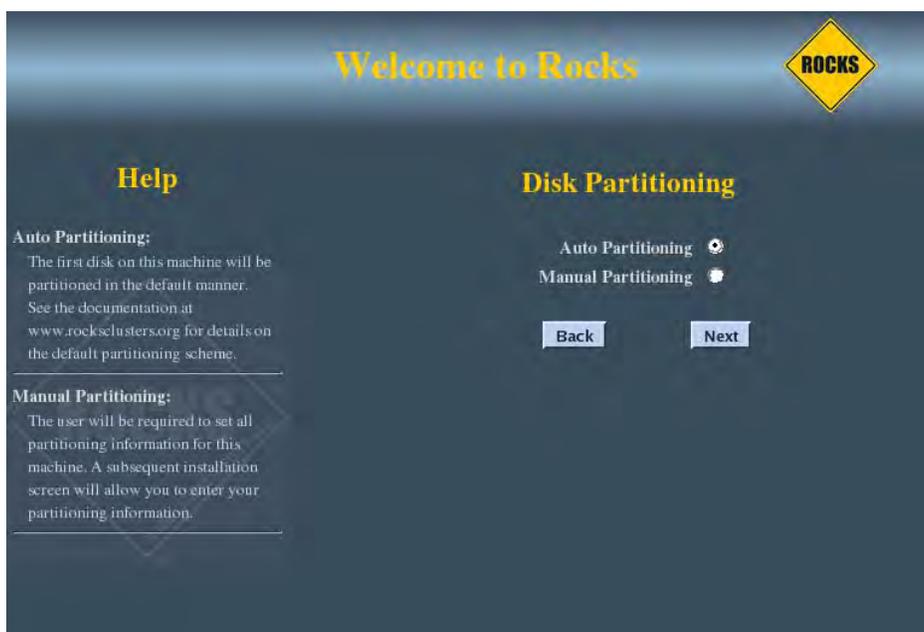


Figura 14. Instalación Rocks: Pantalla 12

El particionado automático, particiona y formatea el primer disco que encuentre conectado al frontend, sin afectar los demás discos duros que se hallen conectados al frontend.

El particionado por defecto, al elegir esta opción, es como sigue:

Partition Name	Size
/	8 GB
/var	4 GB
Swap	1 GB
/export (symbolically linked to /state/partition1)	remainder of root disk

Tabla 3. Particionado por defecto Rocks



Para este proyecto, la opción que se eligió fue el particionado manual, lo que nos permitió establecer nuestros propios parámetros de particionado.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

13. Particionado manual. El disco duro del frontend es de 40GB y para fines de este proyecto se acordó particionar el disco de la siguiente manera:

Partition Name	Size
/	12 GB (12288 MB)
/var	6 GB (6144 MB)
swap	2 GB (2048 MB)
/export (symbolically linked to /state/partition1)	20 GB (20480 MB)

Tabla 4. Particionado del frontend en Mixbaal

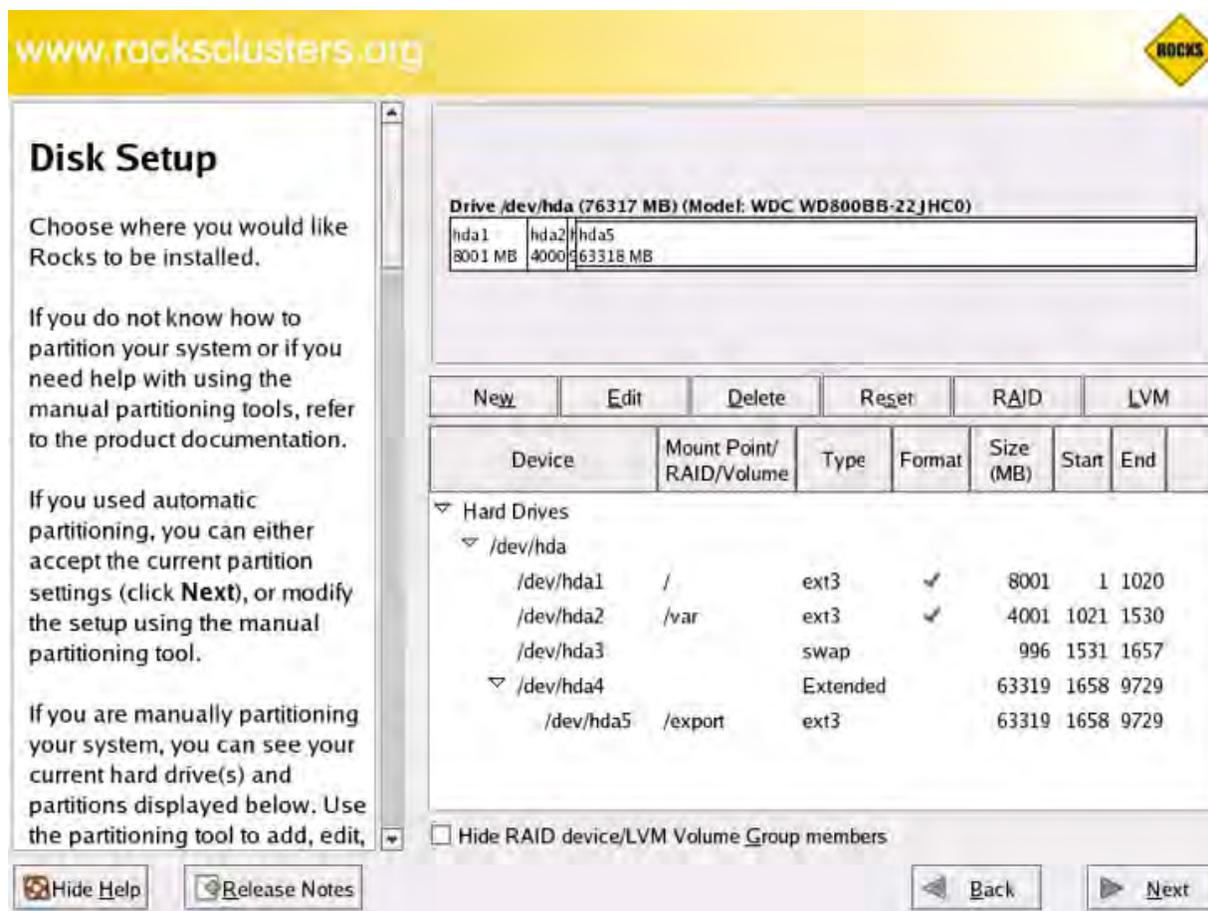


Figura 15. Instalación Rocks: Pantalla 13

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

14. Una vez que se decidieron las particiones, se pulsó el botón `Next` y el frontend fue formateado, borrándose todos los datos contenidos en él. Posteriormente el programa de instalación fue pidiendo los discos con los rolls seleccionados al inicio de la instalación, uno por uno. Para este caso, solo pidió el disco de SCE roll, ya que los demás venían incluidos en el jumbo(DVD)

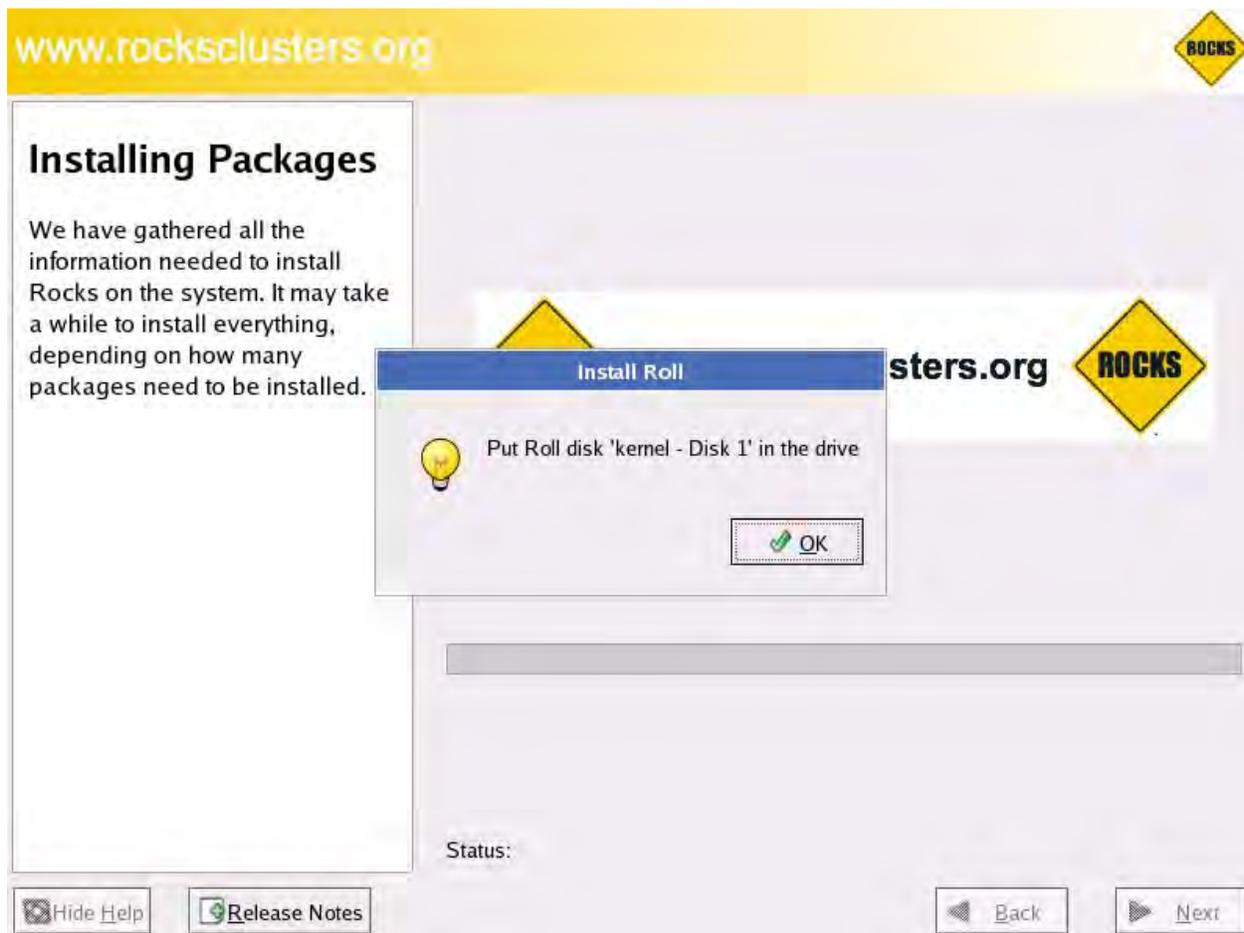


Figura 16. Instalación Rocks: Pantalla 14

Una vez que se insertó el disco con el SCE roll, se presionó `OK` y el contenido de éste se copio en el disco duro del frontend.

15. Ya que se copió toda la información necesaria, se termina la interacción con el usuario e inicia la instalación de todos los rolls seleccionados.

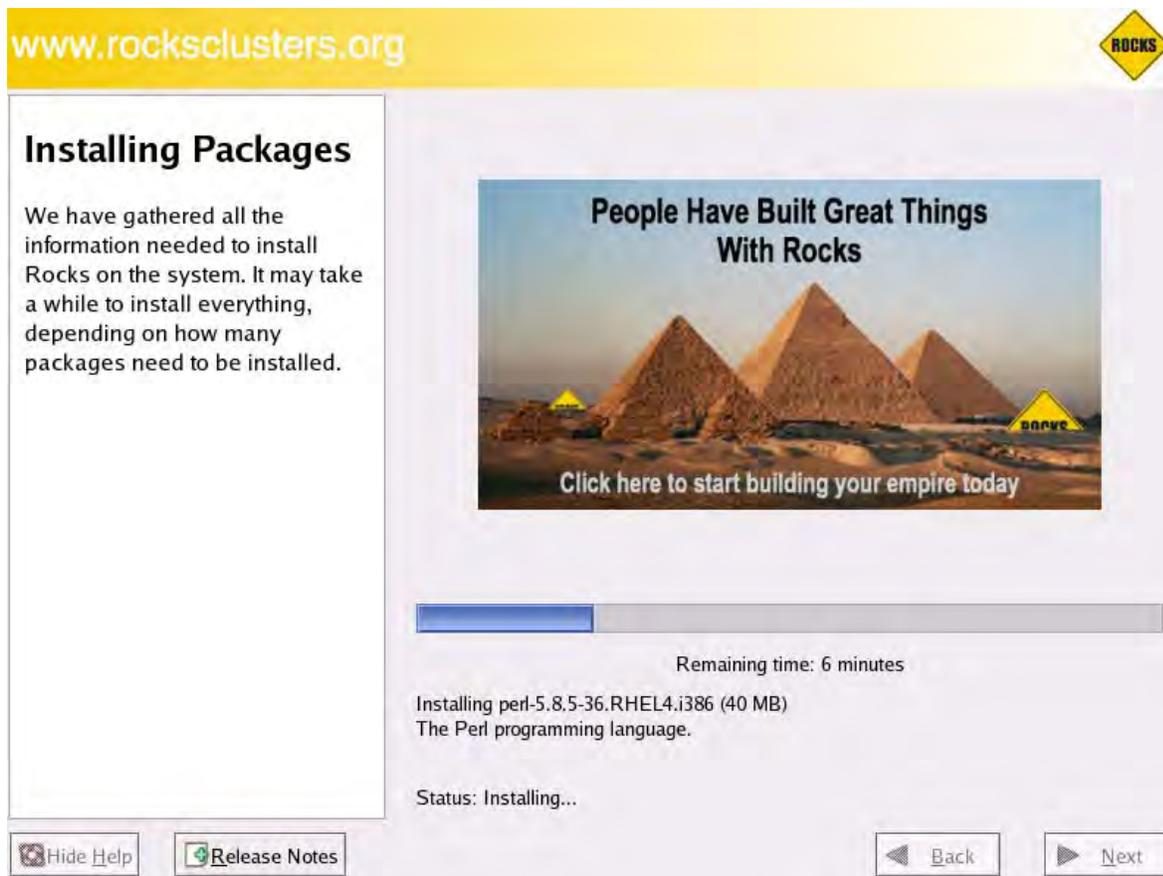


Figura 17. Instalación Rocks: Pantalla 15

16. Finalmente, el programa instala el gestor de arranque y ejecuta en background los programas post-instalación. Cuando esto se completa, el frontend se reinicia ya con Rocks instalado.

Instalación de los nodos

Una vez que se instaló correctamente el frontend, se prosiguió con la instalación de los tres nodos esclavos de prueba, realizando el siguiente procedimiento:

1. Iniciar sesión en el frontend como `root` para ejecutar el programa `insert-ethers` el cual captura las peticiones DHCP* que realizan los nodos esclavos, colocando ésta información en la base de datos MySQL de Rocks de dispositivos conectados al cluster, así como también agrega el registro al archivo `/etc/hosts`. Ésta ejecución se hizo desde línea de comandos:

```
#insert-ethers
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Mostrando la siguiente pantalla:

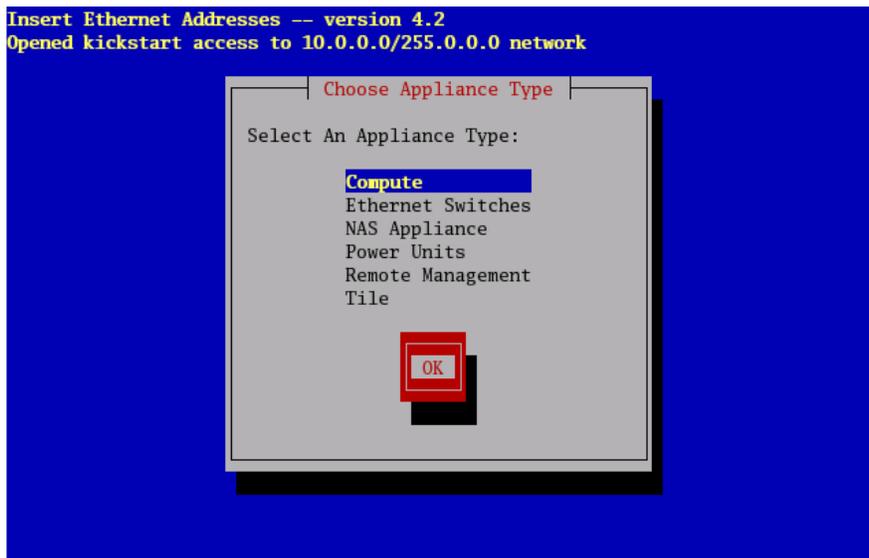


Figura 18. Instalación Rocks: Pantalla 16

Se eligió la opción por defecto `Compute` y se presionó `OK`.

2. La pantalla siguiente, indica que el programa está en espera de reconocer a los nodos esclavos.

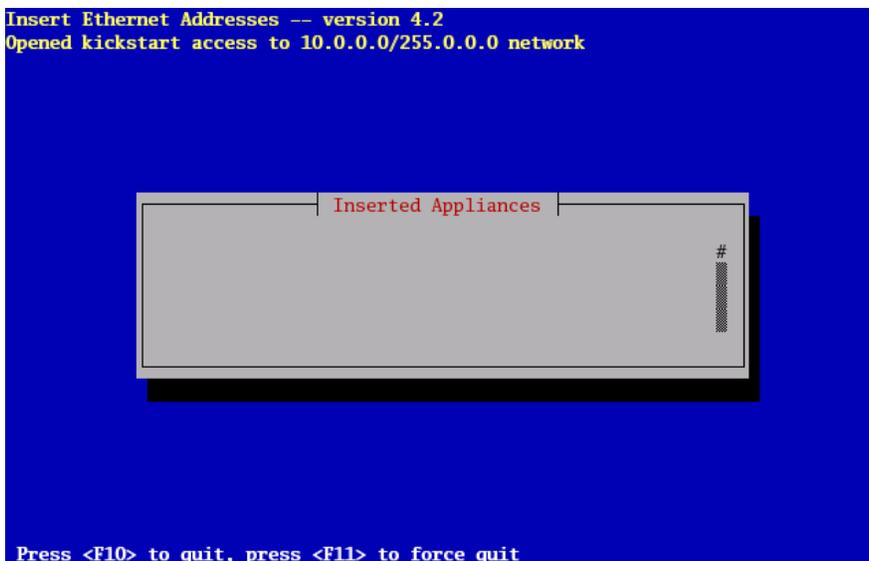


Figura 19. Instalación Rocks: Pantalla 17

3. Insertar el jumbo(DVD) y encender el primer nodo esclavo a instalar.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER



Cabe señalar que en este punto la instalación de los nodos es a ciegas, por lo que se optó por reunir varios monitores para conectarlos a los nodos esclavos y seguir más de cerca su instalación.



Cuando el nodo esclavo arranca con el jumbo(DVD), a diferencia del frontend, en la primera pantalla no se escribe nada, tomando el valor por defecto de `nothing`, que como ya se mencionó, esto le indica al instalador que el nodo a instalar es un nodo esclavo.

4. Cuando el frontend recibe la petición DHCP* del nodo esclavo, en la pantalla del frontend aparece lo siguiente:

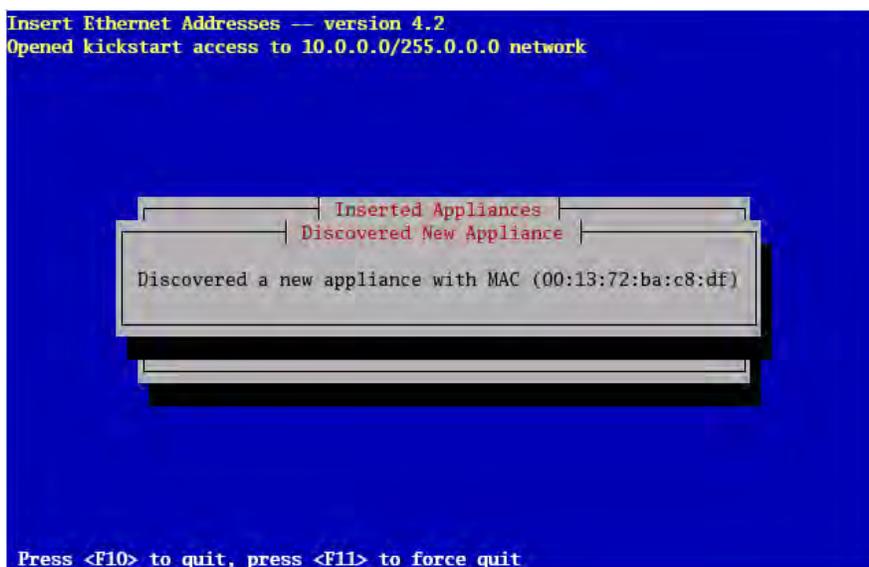


Figura 20. Instalación Rocks: Pantalla 18

Esto indica que el frontend ha reconocido exitosamente el primer nodo esclavo, agregando el registro del dispositivo a la base de datos, así como también se actualizan los archivos de configuración del sistema tales como `/etc/hosts`, `/etc/dhcpd.conf`, entre otros.



Después de una larga espera, el frontend no mostró nada, es decir el nodo esclavo no fue reconocido correctamente por el frontend.



Se cambió el cable de red a otra entrada Ethernet, se reinició el nodo, y esta vez, sí apareció correctamente en el frontend.



CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

5. Después de unos segundos aparece la siguiente pantalla:

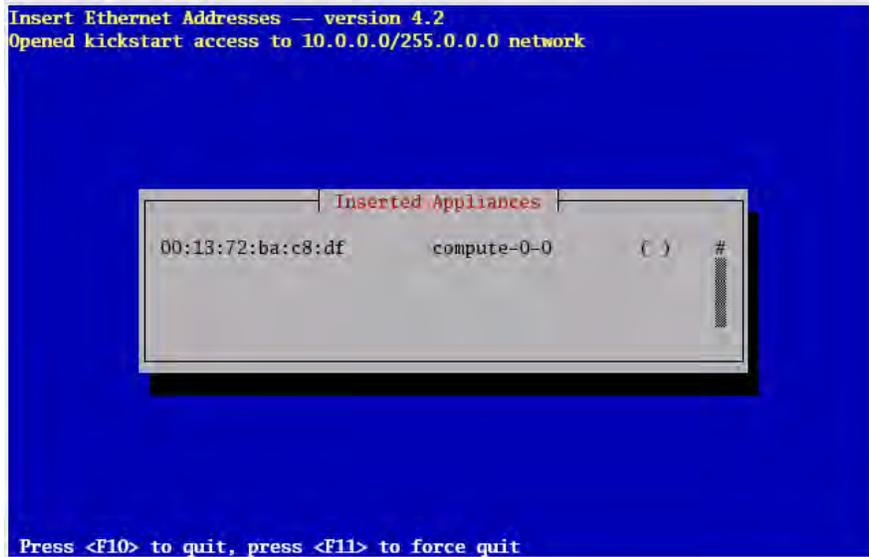


Figura 21. Instalación Rocks: Pantalla 19

Los “()” que aparecen del lado derecho de `compute-0-0` indican que el nodo aún no ha hecho su solicitud para que le sea creado su archivo Kickstart (Véase la sección 2.2.1, apartado *Descripción*)

6. Cuando el nodo ha solicitado su archivo kickstart desde el frontend, aparece la siguiente pantalla:

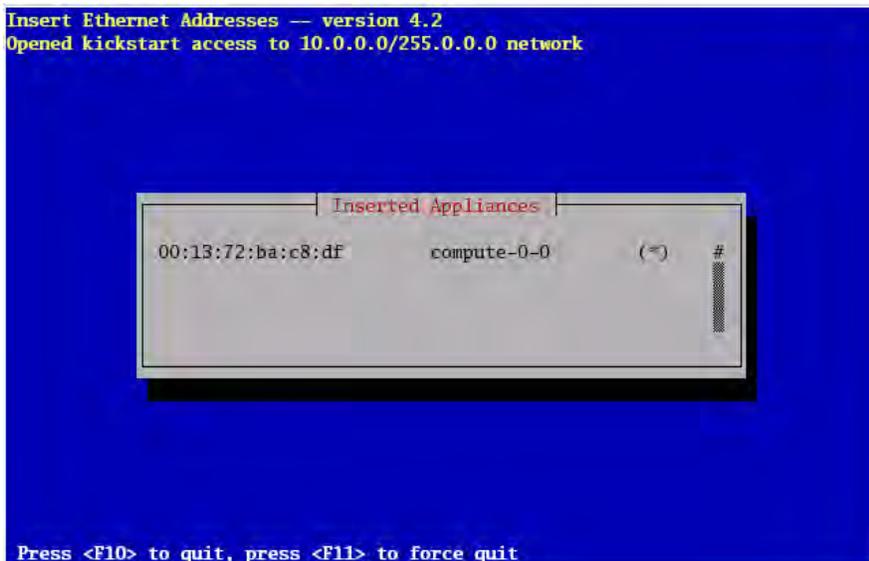


Figura 22. Instalación Rocks: Pantalla 20

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

El “*” indica que la transmisión del archivo ha sido exitosa, de lo contrario aparece un número de error.

En adición a los monitores que agregamos para monitorear la instalación de los nodos, Rocks ofrece `rocks-console`, comando que permite conectarse remotamente a algún nodo en proceso de instalación y ver el progreso de ésta, por ejemplo, si queremos ver como va la instalación del nodo `compute-0-0`:

```
#rocks-console compute-0-0
```



Se debe ejecutar en una sesión con entorno gráfico (X)

7. Cuando se concluye la instalación en un nodo, la unidad de CD/DVD expulsa el disco de instalación, para este caso, jumbo(DVD)

8. Los pasos 3-5 se repiten por cada nodo esclavo que se vaya a añadir, en este caso se llevo a cabo este proceso 2 veces más. Y cada nodo añadido fue apareciendo en la pantalla de `insert-ethers`.



Uno de los nodos esclavos (con `MacAddress* 00:03:47:a5:7c:bd`) se registro en la base y en los archivos de configuración, como `network-0-0`, lo que puede causar algunas complicaciones en su funcionamiento por no llamarse como los demás.



Como no sabíamos qué nodo era el que había sido mal nombrado, nos basamos en el orden de instalación, de modo que el nodo en cuestión resultó ser el 2, a éste le conectamos la unidad de CD/DVD, ratón, teclado y monitor, lo reiniciamos con un LiveCD de Ubuntu y revisamos su `MacAddress*`:

```
#ifconfig -a | grep eth0
```

La `MacAddress*` arrojada por el comando anterior coincidía con el nodo mal nombrado de la pantalla del programa `insert-ethers`, de modo que identificamos correctamente al nodo.

Intentamos reinstalar el nodo, pero el programa `insert-ethers` indicaba que ese nodo ya se había registrado, por lo que en el frontend, eliminamos el registro erróneo del archivo `/etc/hosts`:

```
#vi /etc/hosts
10.1.1.254 network-0-0.local network-0-0 n0-0
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

También lo eliminamos del archivo `/etc/dhcp.conf`:

```
#vi /etc/dhcpd.conf

host network-0-0.local {
    hardware ethernet 00:03:47:a5:7c:bd;
    option host-name "network-0-0.local";
    fixed-address 10.1.1.254;
```



Al hacer lo anterior se perdió la IP* 10.1.1.254.

Se reintentó reinstalarlo, esta vez con éxito y nombrado correctamente.

9. Una vez concluida la instalación de todos los nodos, se recomienda terminar la ejecución de la aplicación `insert-ethers`.

Pruebas Post-Instalación

Con el fin de comprobar que la instalación se haya completado correctamente, se emprendieron las siguientes acciones:

Comando cluster-fork

Como se menciona en la sección 2.2.1, apartado *Sistemas de Archivos por Red*, Rocks proporciona este comando para ejecutar trabajos paralelos de comandos UNIX. Es decir, `cluster-fork`, nos permite ejecutar un mismo comando en varios nodos del cluster.

Si este comando funciona correctamente, significa que la comunicación entre nodos se efectúa de manera adecuada, sin embargo, si al ejecutarlo, los nodos solicitan la contraseña del usuario `root`, es señal de que no se crearon correctamente las llaves de autenticación.



Al ejecutar `cluster-fork` los nodos solicitaron la contraseña del usuario de `root`.



Verificamos el contenido del directorio `.ssh` del frontend:



CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
[root@mixbaal .ssh]# ls -l
total 4
-rw----- 1 root root 2100 Jun 15 2008 authorized_keys
-rw----- 1 root root 883 Apr 12 2008 id_rsa
-rw-r--r-- 1 root root 243 Apr 12 2008 id_rsa.pub
-rw-r--r-- 1 root root 8260 Nov 18 2008 known_hosts
```

Se puede observar, que el directorio no contenía las llaves de los demás nodos, segundo indicio de que las llaves no se habían creado adecuadamente.

Verificamos el contenido del archivo `authorized_keys`:

```
[root@mixbaal .ssh]# more authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAphuEk+ip7W44CHQhOWAQUlmnBkYabJv8wSoZNfM4
IrEvSPVrtbct2NO8cJRxINeKjbr26p1IgpjyGtSKEXZSJppL162nhTs//jFZtsbcfV
cRJjnhwrsKbQsSb04EAG68+miwpjSHprOZfxJpZ8akYUYgIe0iB0pgIF3EmmPm7rc=
root@mixbaal.supercomputo.unam.mx
```

Éste solo contenía su propia llave, siendo lo correcto, contener las llaves de todos los nodos.

Verificamos el directorio `.ssh` de cada nodo hallando a éste vacío, es decir, los nodos no contaban ni con su propia llave.

Por lo anterior se procedió a crear las llaves en cada nodo:

```
[root@compute-0-0 ~]#ssh-keygen -rsa
[root@compute-0-1 ~]#ssh-keygen -rsa
[root@compute-0-2 ~]#ssh-keygen -rsa
```

Se copió el archivo `id_rsa.pub` generado por el comando anterior, renombrado con el número del nodo correspondiente, para fácil identificación, debajo del directorio `.ssh` del nodo maestro.

```
[root@compute-0-0 ~]#scp id_rsa.pub mixbaal:/root/.ssh/id_rsa_c0-0.pub
[root@compute-0-1 ~]#scp id_rsa.pub mixbaal:/root/.ssh/id_rsa_c0-1.pub
[root@compute-0-2 ~]#scp id_rsa.pub mixbaal:/root/.ssh/id_rsa_c0-2.pub
```

Así el nodo maestro tiene todas las llaves:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
[root@mixbaal ~]# ls -l .ssh/
total 7
-rw----- 1 root root 2100 Jun 15 2008 authorized_keys
-rw----- 1 root root 883 Apr 12 2008 id_rsa
-rw-r--r-- 1 root root 232 Jun 15 2008 id_rsa_c0-0.pub
-rw-r--r-- 1 root root 232 Jun 15 2008 id_rsa_c0-1.pub
-rw-r--r-- 1 root root 232 Jun 15 2008 id_rsa_c0-2.pub
-rw-r--r-- 1 root root 243 Apr 12 2008 id_rsa.pub
-rw-r--r-- 1 root root 8260 Nov 18 2008 known_hosts
```

Después se agregó al archivo `authorized_keys` las llaves de los nodos:

```
[root@mixbaal .ssh]#more id_rsa_c0-0.pub >> authorized_keys
[root@mixbaal .ssh]#more id_rsa_c0-1.pub >> authorized_keys
[root@mixbaal .ssh]#more id_rsa_c0-2.pub >> authorized_keys
```

Finalmente el archivo `authorized_keys`, con todas las llaves, se copió a los 3 nodos esclavos:

```
[root@mixbaal .ssh]#scp authorized_keys compute-0-0:/root/.ssh/
[root@mixbaal .ssh]#scp authorized_keys compute-0-1:/root/.ssh/
[root@mixbaal .ssh]#scp authorized_keys compute-0-2:/root/.ssh/
```

Ahora bien, volvimos a ejecutar el comando `cluster-fork` esta vez con éxito, obteniendo la siguiente salida:

```
[root@mixbaal ~]# cluster-fork w
compute-0-0:
 04:47:06 up 442 days, 7:11, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
compute-0-1:
 04:47:08 up 419 days, 8:04, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
compute-0-2:
 04:47:09 up 420 days, 10:07, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
```



Para fines de esta instalación de prueba en los 4 nodos iniciales, ésta fue la manera en que se resolvió el problema, sin embargo si el problema se presenta en una instalación de más de 5 nodos, se recomienda volver a instalar los nodos desde cero, para evitar problemas mayores en el futuro.



Al momento de instalar los 10 nodos finales, éste problema no se volvió a presentar, desde el principio funcionó correctamente el comando `cluster-fork`.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Comando qstat-t

Este comando permite ver el estado de los trabajos así como el estado de las colas.

Se puede ejecutar para ver si los nodos están funcionando correctamente:

```
[root@mixbaal ~]# qstat -f
queuename                qtype used/tot. load_avg arch
states
-----
all.q@compute-0-0.local  BIP   0/2      0.00   1x26-x86
-----
all.q@compute-0-1.local  BIP   0/2     -NA-   1x26-x86
-----
all.q@compute-0-2.local  BIP   0/2      0.00   1x26-x86
```

Lo que nos indica que están funcionando correctamente.

Las pruebas anteriores permiten asegurarse de que el sistema operativo ha sido instalado correctamente, por lo que una vez concluidas, determinamos que el Cluster Mixbaal está listo para el siguiente paso: la instalación del middleware Globus.

2.3 Configuración del Middleware: Globus

Esta sección versará sobre Globus, software que forma parte de la infraestructura que debe cumplir el Cluster Mixbaal, pues sin él difícilmente podría pertenecer a la grid. Así mismo se detallará acerca de su configuración sin mencionar lo que respecta a su instalación, debido a que Globus está incluido en la distribución Rocks por defecto.

Globus es un software de código abierto* desarrollado por la Globus Alliance bajo la dirección de Ian Foster y Carl Kesselman, el cual permite la construcción de infraestructuras y aplicaciones grid²⁰.

Según Isaías Sánchez, Globus funge como el middleware de una grid, es decir una interfaz que se ubica entre los recursos finales y la aplicación de usuario²¹, como se analizó en el Capítulo 1, sección 1.1.6, apartado *Middleware* del presente trabajo.

De esta manera Globus proporciona acceso uniforme y seguro a los recursos de almacenamiento y computación remota, incluyendo software de seguridad, gestión de ejecución de tareas y gestión de datos e información.

Asimismo, Globus se caracteriza por:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

- Su modelo de “paquete de servicios”, el cual permite a las aplicaciones utilizar los servicios de la grid, sin tener que adoptar ningún modelo de programación específico.
- Su conjunto de mecanismos especializados pueden coexistir y, en ocasiones, reemplazar los mecanismos provistos por el cómputo comercial.
- Su soporte para satisfacer los requerimientos de desempeño de una aplicación basada en la información de los recursos disponibles.
- Una interfaz simple y bien definida provee acceso uniforme a diversas implementaciones de servicios locales.
- Para formar parte de una grid, la institución interesada debe proporcionar sólo los servicios de conectividad.²²

2.3.1 Globus Toolkit

El elemento central del proyecto Globus es el Globus Toolkit (GT), el cual está basado en un modelo de paquete de servicios y es considerado como uno de los elementos clave en desarrollo de grids computacionales.

El instituto Tecnológico de Massachusetts (MIT) lo calificó como una de las 10 tecnologías que cambiarían al mundo.

Por otro lado, las grids necesitan soportar una amplia variedad de aplicaciones creadas de acuerdo a diferentes paradigmas de programación. En vez de proveer un modelo uniforme de programación para las aplicaciones grid, Globus Toolkit cuenta con un “paquete de servicios”, como se mencionaba arriba, proveyendo así, una serie de utilidades para que los desarrolladores puedan escoger la que mejor se adecua a sus necesidades. Además, las herramientas pueden introducirse de a una a la vez. Por ejemplo, una aplicación puede usar GRAM o GRIS, de las cuales se hablará más adelante, sin tener que usar necesariamente Globus Security.²³

El elemento básico del Globus Toolkit es la infraestructura GSI, de la cual se hablará a continuación.

GSI (Grid Security Infrastructure)

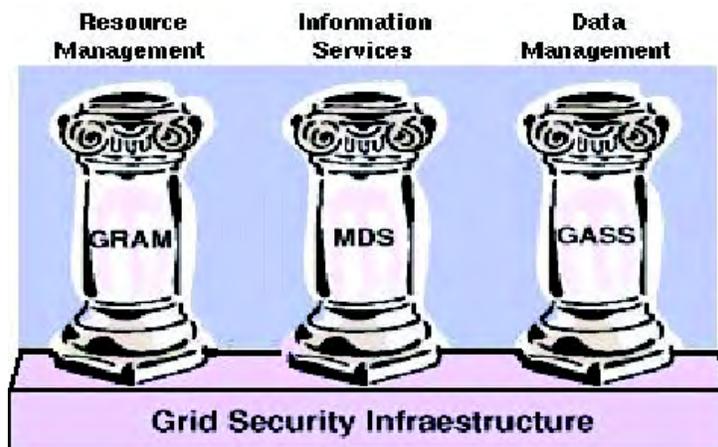
Éste elemento proporciona servicios de autenticación y comunicación segura.

Se conforma por los siguientes elementos:

- GRAM (Globus Resource Allocation Manager): Proporciona los mecanismos necesarios para ejecutar y monitorear trabajos en la Grid.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

- MDS (Monitoring and Discovery System): Está diseñado para proporcionar información sobre los recursos de la Grid, adaptándose a los cambios y estado del sistema.
- GASS (Global Access to Secondary Storage): Permite acceder a las aplicaciones de datos remotos, vía interfaces estándar de entrada y salida.



The Globus Toolkit

Figura 23. Grid Security Infraestructura

El diseño del GSI se basó en las siguientes premisas:

- Comunicación segura (autenticación y confidencialidad) entre los elementos de la grid.
- Mecanismo de seguridad no centralizado.
- Soporte para registro único (single sign-on) y delegación de credenciales para procesos que involucren múltiples recursos.
- Basado en PKI*, certificados X.509*.
- Cada usuario y servicio en la grid es identificado mediante un certificado.
- Utiliza SSL* (TLS*) para su protocolo de autenticación mutua, concepto mencionado en el Capítulo 1, sección 1.1.8, apartado *Autoridad Certificadora (CA)*.

Un certificado GSI incluye:

- Un *subject name* que identifica a la persona u objeto al que pertenece el certificado.
- La *llave pública* del propietario del certificado.
- La *identidad de la Autoridad Certificadora*, CA que firmó el certificado, para garantizar que la llave pública pertenece al propietario del certificado.

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

GSI proporciona la capacidad de delegación, una extensión del protocolo SSL*, que reduce el número de veces que un usuario debe introducir su clave, para tener acceso a otros sistemas. Esto lo logra a través del uso de Proxys.

Un Proxy consiste de un nuevo certificado con:

- Una nueva llave pública.
- Una nueva llave privada (del proxy).
- Firmado con la llave privada del usuario.
- Un tiempo de validez de 12 hrs. típicamente.

El uso de Proxys se emplea cuando un servicio requiere acceder a otros recursos en nombre del usuario. Funcionan de la siguiente manera:

- El servicio remoto crea una solicitud de certificado y lo envía al cliente para que lo firme.
- El cliente firma la solicitud con su llave privada y retorna el nuevo certificado al solicitante.
- Durante el proceso de autenticación mutua, la llave pública del usuario se usa para verificar la validez del certificado del Proxy.
- Se establece una cadena de confianza entre la CA, el usuario y el Proxy.
- Generalmente hay más de un proxy durante una sesión de grid.
- Esta característica permite lo que se conoce como single sign-on.

Con base en lo anterior, los requerimientos para la ejecución de un proceso en una grid basada en Globus, son los siguientes:

- Aceptar los certificados emitidos por alguna CA.
- Cada usuario requiere un certificado emitido por la CA.
- Cada equipo o servicio requiere un certificado.
- Cada Subject o Identidad del certificado de un usuario debe almacenarse en el archivo `grid-mapfile` de cada equipo al que se desee tener acceso. El `grid-mapfile` relaciona a un usuario de la grid con una cuenta en el sistema (función similar al `/etc/passwd`).
- Autenticarse ante la grid.
 - Con el comando `grid-proxy-init`, se genera un certificado temporal (proxy) que permite delegación. Este certificado y el del usuario se emplean para autenticarse con cualquier recurso de la grid.²⁴

2.3.2 Configuración

Ahora bien, como se mencionó en la sección anterior, Globus Toolkit 4 (GT4) se instaló por defecto cuando se instaló Rocks, sin embargo el cluster requiere de un certificado de host único, formado por una llave pública y una privada, que lo identificará de forma

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

segura en la grid de PRAGMA, adicionalmente es necesario realizar una configuración previa de los directorios de seguridad. Esta sección tratará sobre estos aspectos.

Generar una CA

Primeramente se debe elegir la Autoridad Certificadora, quien se encargará de firmar y gestionar los certificados de las máquinas y usuarios, además de confirmar que el dueño de un certificado sea realmente la persona que dice ser, como se analizó en el Capítulo 1 sección 1.1.8, apartado *Autoridad Certificadora (CA)*.

Por defecto, la instalación está preparada para trabajar con la CA gratuita de Globus, así como también se puede crear una CA virtual, ambas de prueba y con poca validez si se requiere confianza con instituciones u organizaciones externas a la nuestra. Para fines de este proyecto se optó por instalar nuestra propia CA en lo que el Departamento de Seguridad en Cómputo de la DGSCA, hace entrega de los certificados generados por la CA de la UNAM, proceso que tarda algunos meses. Una vez se obtengan éstos se procederá a sustituirlos por los que generaremos más adelante.

1. Para generar nuestra CA se deben realizar los siguientes pasos:
 - a. Generar aleatoriamente nuestro propio par de claves.
 - b. Proteger la clave privada.
 - c. Crear nuestro propio certificado.
 - d. Firmar el certificado con nuestra clave privada.

Para llevar a cabo dichos pasos se ejecutó el comando `setup-simple-ca`, provisto por el GT4, en el frontend, quien cumplirá el rol de CA.



Las líneas en negritas corresponden a los datos que introdujimos de manera manual conforme el programa los iba solicitando, tales como `subject`, `e.mail` de la CA, `expiration date` y `pass phrase`.

Las siguientes páginas muestran la salida y procedimiento de este paso:

```
[root@mixbaal ~]# /opt/globus/setup/globus/setup-simple-ca

  C e r t i f i c a t e   A u t h o r i t y   S e t u p

This script will setup a Certificate Authority for signing Globus
users certificates. It will also generate a simple CA package
that can be distributed to the users of the CA.

The CA information about the certificates it distributes will
be kept in:

/root/.globus/simpleCA/
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
/root/.globus/simpleCA/

Enter a unique subject name for this CA:O=DGSCA-UNAM/OU=Supercomputo-
UNAM/CN=Aut Cert Supercomputo

Enter the email of the CA (this is the email where certificate
requests will be sent to be signed by the CA):yoli@super.unam.mx

The CA certificate has an expiration date. Keep in mind that
once the CA certificate has expired, all the certificates
signed by that CA become invalid. A CA should regenerate
the CA certificate and start re-issuing ca-setup packages
before the actual CA certificate expires. This can be done
by re-running this setup script. Enter the number of DAYS
the CA certificate should last before it expires.
[default: 5 years (1825 days)]:

Enter PEM pass phrase: *****
Verifying - Enter PEM pass phrase:

creating CA config package...done.

A self-signed certificate has been generated
for the Certificate Authority with the subject:

/O=DGSCA-UNAM/OU/CN=O=DGSCA-UNAM/OU=Supercomputo-UNAM/CN=Aut Cert
Supercomputo

If this is invalid, rerun this script

/opt/globus/setup/globus/setup-simple-ca

and enter the appropriate fields.

-----
The private key of the CA is stored in
/root/.globus/simpleCA//private/cakey.pem
The public CA certificate is stored in
/root/.globus/simpleCA//cacert.pem

The distribution package built for this CA is stored in

/root/.globus/simpleCA//globus_simple_ca_9e17d234_setup-0.19.tar.gz

This file must be distributed to any host wishing to request
certificates from this CA.

CA setup complete.

The following commands will now be run to setup the security
configuration files for this CA:
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
$GLOBUS_LOCATION/sbin/gpt-build
/root/.globus/simpleCA//globus_simple_ca_9e17d234_setup-0.19.tar.gz

$GLOBUS_LOCATION/sbin/gpt-postinstall
-----

setup-ssl-utils: Configuring ssl-utils package
Running setup-ssl-utils-sh-scripts...

*****

Note: To complete setup of the GSI software you need to run the
following script as root to configure your security configuration
directory:

/opt/globus/setup/globus_simple_ca_9e17d234_setup/setup-gsi

For further information on using the setup-gsi script, use the -help
option. The -default option sets this security configuration to be
the default, and -nonroot can be used on systems where root access is
not available.

*****
setup-ssl-utils: Complete
```



Después de ejecutar el comando anterior, se tendrá la información de la CA generada en `/root/.globus/simpleCA`

En la salida de éste comando (`setup-simple-ca`) prestamos atención a la nota:

```
Note: To complete setup of the GSI software you need to run the
following script as root to configure your security configuration
directory:

/opt/globus/setup/globus_simple_ca_9e17d234_setup/setup-gsi
```

2. Se procede a ejecutar el comando anterior el cual establece a esta Autoridad Certificadora como la CA por defecto de nuestro cluster:

```
[root@mixbaal ~]#
/opt/globus/setup/globus_simple_ca_9e17d234_setup/setup-gsi -default

setup-gsi: Configuring GSI security
Installing /etc/grid-security/certificates//grid-
security.conf.9e17d234...
Running grid-security-config...
sed: -e expression #1, char 67: unknown option to `s'
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
sed: -e expression #1, char 92: unknown option to `s'
sed: -e expression #1, char 57: unknown option to `s'
Installing Globus CA certificate into trusted CA certificate
directory...
Installing Globus CA signing policy into trusted CA certificate
directory...

setup-gsi: Complete
```



El número que aparece en negritas (**9e17d234**) es el número que identifica a la CA.

3. Se confirmó que los certificados se crearan en el directorio correcto:

```
[root@mixbaal ~]# ls /etc/grid-security/certificates/
9e17d234.0                globus-user-ssl.conf.9e17d234
9e17d234.signing_policy  grid-security.conf.9e17d234
globus-host-ssl.conf.9e17d234
```

3. Se verificó el contenido del certificado. Mismo que se muestra en la siguiente página:

```
[root@mixbaal ~]# openssl x509 -in /etc/grid-
security/certificates/9e17d234.0 -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: O=DGSCA-UNAM/OU, CN=O=DGSCA-UNAM, OU=Supercomputo-UNAM,
CN=Aut Cert Supercompuo
        Validity
            Not Before: Jun 24 01:05:23 2009 GMT
            Not After : Jun 23 01:05:23 2014 GMT
        Subject: O=DGSCA-UNAM/OU, CN=O=DGSCA-UNAM, OU=Supercomputo-
UNAM, CN=Aut Cert Supercompto
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c5:2c:2e:81:bd:c7:3a:e3:d4:e8:55:46:b9:6a:
                    9f:8b:be:3c:dd:30:b7:c9:1d:1c:f5:3e:94:a6:98:
                    39:6f:31:34:e9:f0:c0:89:13:82:02:8a:37:02:0f:
                    d8:40:7f:8d:15:45:4c:4a:fe:83:6f:2e:7f:66:81:
                    b3:3f:48:47:ec:ec:89:18:79:16:6f:9f:dd:79:dc:
                    fe:f2:9c:4e:b9:0e:1b:82:f8:a7:06:3d:bf:1d:c8:
                    e9:99:bd:b1:15:82:c8:56:a5:3c:f7:0e:65:b7:06:
                    0c:e6:bb:f0:e8:e4:80:5d:f3:cd:9d:1c:52:25:02:
                    93:54:2b:a3:03:b6:15:89:85
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints: critical
  CA:TRUE
  X509v3 Subject Key Identifier:
A3:16:F6:AF:FB:E0:E8:71:E4:F7:C3:E7:E9:B7:B6:A9:AB:5A:07:0B
  Netscape Cert Type:
    SSL CA, S/MIME CA, Object Signing CA
Signature Algorithm: md5WithRSAEncryption
69:18:7c:c9:08:8e:05:e5:1e:02:91:9f:f7:dc:f6:27:11:e3:
22:1c:78:d7:4f:d2:80:87:b8:71:35:8a:f6:df:c8:bc:f8:7b:
b9:99:78:9c:a5:6e:b7:09:c1:4d:31:52:94:24:66:2f:16:1a:
da:d1:df:00:6a:fc:fe:31:2c:36:57:f4:90:ea:e0:7c:e1:de:
21:30:fe:75:c3:44:ce:9c:41:e8:13:f0:6e:30:13:67:43:4d:
37:b1:24:95:7d:7e:21:97:e7:cc:83:7c:da:44:cf:07:41:1e:
03:32:7c:65:78:12:ca:43:95:b4:a2:2f:22:25:03:5c:8b:5d:
df:fe
```



El comando `openssl`, entre otras cosas, nos permite dar un formato legible a los certificados creados. Con la opción `-text` se le solicitó que nos lo mostrara en texto plano, de lo contrario nos lo mostraría cifrado.

Generar un certificado de host para Mixbaal

En este punto del proceso ya se cuenta con una CA de prueba, lo que procede es generar un certificado para el Cluster, siguiendo los siguientes pasos:

4. Para solicitarle un certificado para Mixbaal, se ejecuta el siguiente comando:

```
[root@mixbaal grid-security]# grid-cert-request -host mixbaal.supercomputo.unam.mx
```



La opción `-host` indica que el certificado será para un recurso físico.

5. Si el comando anterior no arrojó ningún error, significa que todo se ejecutó correctamente. Los archivos que se deben haber creado en `/etc/grid-security` son:

```
[root@mixbaal grid-security]# ls -l host*
-rw-r--r-- 1 root root 0 Jun 23 20:14 hostcert.pem
-rw-r--r-- 1 root root 1285 Jun 23 20:14 hostcert_request.pem
-r----- 1 root root 887 Jun 23 20:14 hostkey.pem
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER



Llave pública: `hostcert.pem`
Solicitud de llave pública: `hostcert_request.pem`
Llave privada: `hostkey.pem`



Si Mixbaal no fuera la propia CA, se procedería a enviar la solicitud de llave pública (`hostcert_request.pem`) a la CA permitida para firmarla. En un futuro la CA de la UNAM.

6. Como el Cluster es nuestra propia CA, omitimos el envío de éste archivo. Así nuestra CA firma dicha solicitud. Para esto se realizaron los siguientes pasos:

- Cambiarse a `/root/.globus/simpleCA`
- Crear el directorio `solicitudes` y `certificados` para un mejor control de las solicitudes y los certificados que generemos.

```
[root@mixbaal grid-security]# cd /root/.globus/simpleCA
[root@mixbaal simpleCA]# mkdir solicitudes
[root@mixbaal simpleCA]# mkdir certificados
```

- Copiar el archivo de `hostcert_request.pem` al directorio `solicitudes` con el nombre del Cluster y con terminación `.req` para identificar que se trata de una solicitud:

```
[root@mixbaal simpleCA]# cp /etc/grid-security/hostcert_request.pem
solicitudes/mixbaal.super.unam.mx.req
```

- Se procede a firmar la solicitud

```
[root@mixbaal simpleCA]# grid-ca-sign -in solicitudes/mixbaal.super.unam.mx.req -
out certificados/mixbaal.super.unam.mx.pem
```



Nótese que el archivo de entrada es la solicitud que nombramos con terminación `.req`, así como también se indicó que el archivo de salida se guardara con el nombre del Cluster y terminación `.pem`, para identificar que se trata de una solicitud firmada, en el directorio `certificados` creado previamente.

Posteriormente el archivo generado se copia a `/etc/grid-security` reemplazando el que existe actualmente, pues éste no está firmado por la CA:

```
[root@mixbaal simpleCA]# cp certificados/mixbaal.super.unam.mx.pem /etc/grid-
security/hostcert.pem
cp: overwrite `/etc/grid-security/hostcert.pem'? y
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

Generar un certificado de usuario

Por el momento, se dispone de una CA y de un certificado firmado (llave pública firmada) para el Cluster Mixbaal, lo último que hace falta es generar un certificado para un usuario de prueba (la creación de cuentas de usuarios se verá con detalle en el siguiente capítulo). Con el siguiente procedimiento:

7. Para éste caso se generará un certificado para `perla`. Nos autenticamos como `perla` y ejecutamos el comando `grid-cert-request`:

```
[root@mixbaal ~]# su - perla
[perla@mixbaal ~]$ grid-cert-request

A private key and a certificate request has been generated with the subject:

/O=DGSCA-UNAM/OU/OU=supercomputo.unam.mx/CN=Perla Cristal Palacios Morales

If the CN=Perla Cristal Palacios Morales is not appropriate, rerun this
script with the -force -cn "Common Name" options.

Your private key is stored in /home/perla/.globus/userkey.pem
Your request is stored in /home/perla/.globus/usercert_request.pem
Please e-mail the request to the
You may use a command similar to the following:

    cat /home/perla/.globus/usercert_request.pem | mail

Only use the above if this machine can send AND receive e-mail. if not, please
mail using some other method.

Your certificate will be mailed to you within two working days.
If you receive no response, contact at
```

Verificamos que se hayan creado los archivos correspondientes:

```
[perla@mixbaal ~]$ ls .globus/
usercert.pem  usercert_request.pem  userkey.pem
```



A diferencia del certificado creado para el host, el de usuario se crea en el home de éste.

8. Se procede a firmar la llave pública, siguiendo el procedimiento del paso 6, del apartado *Generar un certificado de host para Mixbaal*, de esta sección, como `root`.

- Cambiarse a `/root/.globus/simpleCA`
- Copiar el archivo de `usercert_request.pem` al directorio `solicitudes` con el nombre del usuario y con terminación `.req` para identificar que se trata de una solicitud:

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
[root@mixbaal simpleCA]# cp /home/perla/.globus/usercert_request.pem
solicitudes/perla.super.unam.mx.req
```

- Se procede a firmar la solicitud:

```
[root@mixbaal simpleCA]# grid-ca-sign -in solicitudes/perla.super.unam.mx.req -out
certificados/perla.super.unam.mx.pem
```

Posteriormente el archivo generado se copia al home de perla, /home/perla/.globus reemplazando el que existe actualmente, pues éste no está firmado por la CA:

```
[root@mixbaal simpleCA]# cp certificados/perla.super.unam.mx.pem
/home/perla/.globus/usercert.pem
cp: overwrite `/home/perla/.globus/usercert.pem'? y
```

Una vez concluidos los pasos anteriores se inician las pruebas para comprobar que Globus haya quedado correctamente configurado.

Probando Globus Toolkit 4

Para probar que Globus está funcionando correctamente realizamos los siguientes pasos:

Pruebas dentro de Mixbaal

1. Nos autenticamos ante el cluster, con el comando `grid-proxy-init`, como se analizó al final de la sección 2.3.1:

```
[perla@mixbaal ~]$ grid-proxy-init
Your identity: /O=DGSCA-UNAM/OU=OU=supercomputo.unam.mx/CN=Perla Cristal Palacios
Morales
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Wed Jun 24 08:34:26 2009
```

2. Con el comando `grid-proxy-info` verificamos el estado del certificado temporal:

```
[perla@mixbaal ~]$ grid-proxy-info
Subject : /O=DGSCA-UNAM/OU=OU=supercomputo.unam.mx/CN=Perla Cristal Palacios
Morales/CN=623640706
issuer : /O=DGSCA-UNAM/OU=OU=supercomputo.unam.mx/CN=Perla Cristal Palacios
Morales
identity : /O=DGSCA-UNAM/OU=OU=supercomputo.unam.mx/CN=Perla Cristal Palacios
Morales
type : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 512 bits
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

```
path      : /tmp/x509up_u500
timeleft  : 11:59:10
```

3. Con el valor del campo `identity` de la salida anterior, damos de alta al usuario `perla`, agregando dicho valor al archivo `/etc/grid-security/grid-mapfile`. De modo que éste archivo se verá de la siguiente manera:

```
[root@mixbaal ~]# more /etc/grid-security/grid-mapfile
"/O=DGSCA-UNAM/OU/OU=supercomputo.unam.mx/CN=Perla Cristal Palacios Morales" perla
```

4. Probamos que el usuario pueda ejecutar un `job*` una vez que creó su proxy temporal:

```
[perla@mixbaal ~]$ globus-job-run mixbaal.supercomputo.unam.mx /bin/hostname
mixbaal.supercomputo.unam.mx
```

Lo anterior significa que primero se conectó al frontend, hubo un intercambio de información y se validó la CA, hizo un mapeo del archivo `/etc/grid-mapfile` y la información del proxy temporal, comprobó que todo coincidiera, si todo está bien, como este caso, devuelve la salida correcta. Es decir, a nivel cluster, Globus se ha configurado exitosamente.

Pruebas desde otro host

Para realizar pruebas desde un host externo, en donde se asume que también tiene instalado y configurado Globus Toolkit 4, se realizó lo siguiente:

1. Se copiaron los archivos de la CA, al host externo, en el directorio `/home/perla/.globus/certificates`:

```
perla@malicia:~/globus/certificates$ scp
root@mixbaal.supercomputo.unam.mx:/etc/grid-security/certificates/9e17d234.* .
root@mixbaal.supercomputo.unam.mx's password:
9e17d234.0                                100% 912
0.9KB/s 00:00
9e17d234.signing_policy                    100% 1323
1.3KB/s 00:00
```



Estos archivos, proporcionados en un futuro por el Departamento de Seguridad en Cómputo de la DGSCA, serán enviados a la Coordinadora de PRAGMA, para que dé de alta nuestra CA.

2. Generamos el proxy:

```
perla@malicia:~$ grid-proxy-init
Your identity: /O=DGSCA-UNAM/OU/OU=supercomputo.unam.mx/CN=Perla Cristal Palacios
Morales
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Wed Jun 24 08:51:46 2009
```

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

3. Ejecutamos el mismo comando de prueba:

```
perla@malicia:~$ globus-job-run mixbaal.supercomputo.unam.mx /bin/hostname
mixbaal.supercomputo.unam.mx
```

Y vemos que la salida también fue exitosa, lo que nos indicó que la configuración es correcta.

4. Por último se habilitó el sistema de colas* (SGE*) para verificar que se asignaran correctamente, trabajos a los diferentes nodos, ya que, hasta aquí solo se ha probado que el frontend reciba los trabajos.

Probamos el mismo comando del punto 3, esperando como salida, en lugar de mixbaal.supercomputo.unam.mx, compute-0-*n*, esto porque una vez establecido el sistema de colas*, éste revisa qué nodo es idóneo para atender el trabajo y lo envía a éste.

```
perla@malicia:~$ globus-job-run mixbaal.supercomputo.unam.mx
/bin/hostname
-catch_rsh /opt/gridengine/default/spool/compute-0-
9/active_jobs/87.1/pe_hostfile
compute-0-9
/opt/gridengine/bin/lx26-x86/qrsh -V -inherit compute-0-9 /bin/sh
/home/perla/.globus/job/mixbaal.supercomputo.unam.mx/3570.1248561462
/scheduler_sge_cmd_script
compute-0-9.local
```

Como podemos observar ahora la salida es compute-0-9.local lo que indica que el trabajo llegó primero al sistema de colas* y fue despachado en el nodo número 9. Con esta salida podemos dar por concluida la configuración de Globus Toolkit 4.

Fuentes de Consulta

¹ PRAGMA, *Preparation Guide for New PRAGMA Testbed Site*, [en línea] URL <<http://goc.pragma-grid.net/pragmadoc/userguide/join.html>>

² [ibid], [en línea] URL <<http://goc.pragma-grid.net/secure/updaccounts/zhengc.tar.gz>>

³ [ibid], [en línea] URL <http://goc.pragma-grid.net/secure/updaccounts/pragma_setuser>

⁴ Zuluaga, Jorge, *Rocks and Rolls: Sistemas de Computación de Alto Rendimiento de Fácil Despliegue*, Grupo de Física y Astrofísica Computacional, MACOM, Instituto de Física, Universidad de Antioquia, Colombia, pág. 3 [PDF] URL <<http://urania.udea.edu.co/facom/facom-products/documentation.php>>

⁵ Zuluaga, Jorge, op. cit., pág 5.

⁶ University of California, *HPC Roll: Users Guide*, [PDF] URL <<http://www.rocksclusters.org/roll-documentation/hpc/5.2/roll-hpc-usersguide.pdf>>

⁷ University of California, *Area51 Roll: Users Guide*, [PDF] URL <<http://www.rocksclusters.org/roll-documentation/area51/5.2/roll-area51-usersguide.pdf>>

CAPÍTULO 2. INSTALACIÓN Y CONFIGURACIÓN DEL CLUSTER

⁸ University of California, *BioRoll Roll: Users Guide*, pág. 5 [PDF] URL <<http://www.rocksclusters.org/roll-documentation/bio/5.2/roll-bio-usersguide.pdf>>

⁹ Flores Salgado, Lourdes Yolanda. Apuntes del curso: *Introducción a Sistemas Operativos*, Facultad de Contaduría y Administración, UNAM, México, 2004.

¹⁰ Zuluaga, Jorge, *Linux Clustering con Rocks. Una Guía Práctica. Guía 3 :Administración básica de la plataforma*, Grupo de Física y Astrofísica Computacional, MACOM, Instituto de Física, Universidad de Antioquia, Colombia, p. 36 [PDF] URL <<http://urania.udea.edu.co/facom/facom-products/documentation.php>>

¹¹ Íbid, p. 37

¹² Íbid, p. 12

¹³ Íbid, p.39.

¹⁴ Íbid, p. 40.

¹⁵ Zuluaga, Jorge, *Linux Clustering con Rocks. Una Guía Práctica. Apéndice A: Instalación de un Cluster Rocks*, Grupo de Física y Astrofísica Computacional, MACOM, Instituto de Física, Universidad de Antioquia, Colombia, [PDF] URL <<http://urania.udea.edu.co/facom/facom-products/documentation.php>>

¹⁶ Rocks, Downloads, [en línea] URL <http://www.rocksclusters.org/wordpress/?page_id=107>

¹⁷ ThaiGrid Download Server, [en línea] URL <http://download.thaigrid.or.th/pub/rocks/rocks-4.3/sce-4.3-1.7.4_1.i386.disk1.iso>

¹⁸ Internacional Country Codes, [en línea] URL <<http://www.bris.ac.uk/Support/Network/ipcodes1.html>>

¹⁹ Infoplease, *Latitude and Longitude of World Cities*, [en línea] URL <<http://www.infoplease.com/ipa/A0001769.html>>

²⁰ Touriño, Juan, *Globus Toolkit 4*, 32ª Conferencia Latinoamericana de Informática, CLEI'2006, Santiago de Chile, [PDF] URL <<http://www.cytgrid.org/documentos/globus-Juan.pdf>>

²¹ Sánchez Enríquez, Ysaías, *Clustering y Grid Computing*, Escuela de Informática, Universidad Nacional de Trujillo, Perú, 2007, [PDF] URL <<http://www.seccperu.org/files/Clustering%20and%20Grid%20Computing.pdf>>

²² Murrieta León, Eduardo, op. cit., pág. 27 [PDF]

²³ Grid Café, *Construyendo una Grid. Globus Toolkit*, [en línea] URL <http://www.gridcafe.org/es-globus-toolkit_ES.html>

²⁴ Murrieta León, Eduardo, op. cit., págs. 32-36, 39-41, 46, 47 [PDF]

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

- 121 3.1 Creación de Cuentas de Usuario en Rocks
- 134 3.2 Instalación y Configuración de Aplicaciones
- 151 3.3 Actualización Automática de Certificados

PRAGYA

En el capítulo que precede se externó el compromiso que adquirió la UNAM con la comunidad de PRAGMA, el cual consistió en destinar un sistema de cómputo para ser empleado en experimentos basados en tecnologías de grid, por lo que el Departamento de Supercómputo de la DGSCA proveyó un cluster conformado por diez nodos.

Asimismo se describieron las diversas tareas que se llevaron a cabo en el proceso de instalación y configuración, preliminares a la puesta a punto del cluster, mismas que incluyen la instalación del sistema operativo Rocks, la configuración del middleware Globus y de la Autoridad Certificadora.

El presente capítulo detalla las actividades posteriores que se efectuaron en el sistema, a saber, creación de cuentas de usuario, instalación y configuración de las aplicaciones requeridas por el GOC, Openbabel y Gamess, con las correspondientes pruebas para verificar su correcto funcionamiento y configuración de actualizaciones automáticas de certificados. Con ello se concluye la puesta a punto del cluster, listo para integrarse a la grid de PRAGMA.

3.1 Creación de Cuentas de Usuarios en Rocks

La creación de una cuenta de usuario en el cluster se realizó en 4 pasos¹:

1) Creación básica de la cuenta. Para esto, creamos la cuenta para el usuario paulina.

Esto se logra mediante el uso del comando de Linux `useradd`:

```
# useradd paulina
```

Este comando crea un registro nuevo para el usuario en los archivos `/etc/passwd`, `/etc/shadow` y `/etc/group` (por defecto se crea un grupo por usuario, ejemplo: usuario paulina, grupo paulina). Adicionalmente crea el directorio hogar en `/export/home/usuario` y copia allí el contenido del directorio `/etc/skel`, éste contiene archivos de configuración básicos para las cuentas de los usuarios. Una vez creada la cuenta de usuario, el directorio hogar contiene:

```
[root@mixbaal ~]# ls -la /export/home/paulina/
total 28
drwx-----  2 paulina paulina 4096 Jul 12 22:21 .
drwxr-xr-x  13 root    root    4096 Jul 12 22:21 ..
-rw-r--r--   1 paulina paulina   24 Jul 12 22:21 .bash_logout
-rw-r--r--   1 paulina paulina  191 Jul 12 22:21 .bash_profile
-rw-r--r--   1 paulina paulina  124 Jul 12 22:21 .bashrc
-rw-r--r--   1 paulina paulina  383 Jul 12 22:21 .emacs
-rw-r--r--   1 paulina paulina  120 Jul 12 22:21 .gtkrc
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Posteriormente, es necesario que asignemos correctamente la ubicación del directorio hogar. Por omisión, `useradd` fija este directorio en el `/export/home`. Sin embargo para que el usuario encuentre su directorio hogar en los demás nodos éste debe ser cambiado a `/home`.

Para hacerlo se utiliza el comando `usermod`:

```
# usermod -d /home/paulina paulina
```

2) Asignación de una contraseña. La contraseña puede ser asignada con el comando `passwd`:

```
root@mixbaal ~]# passwd paulina
Changing password for user paulina.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

La creación de la cuenta no es garantía de que el usuario pueda conectarse de manera transparente en el cluster, es necesario, hacer una sincronización de los archivos de configuración con 411 a todos los nodos del cluster y configurar el `autofs` para que el directorio hogar sea montado automáticamente en los nodos, como se vio en el capítulo anterior, en la sección 2.2.1, apartado *Sistema de Archivos por Red*. Para nuestro caso fue necesario editar el archivo `/etc/auto.home`, quedando como sigue:

```
[root@mixbaal ~]# more /etc/auto.home
install      mixbaal.local:/export/home/&
condor      mixbaal.local:/export/home/condor
perla       mixbaal.local:/export/home/perla
paulina    mixbaal.local:/export/home/paulina
```

Se añadió la última línea.

3) Sincronización de los archivos de configuración usando 411 (Véase Capítulo 2, sección 2.2.1, apartado *Servicio 411*). Después de que se han completado los pasos anteriores, los archivos de configuración con la información definida en ellos deben sincronizarse en todo el cluster.



Se recomienda usar el método `make` con la opción `force` para garantizar que todos los archivos sean sincronizados.

```
[root@mixbaal ~]# make -C /var/411 force
make: Entering directory `/var/411'
rm -rf /etc/411.d/*
make
make[1]: Entering directory `/var/411'
/opt/rocks/sbin/411put --comment="#" /etc/auto.home

411 Wrote: /etc/411.d/etc.auto..home
Size: 534/217 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

/opt/rocks/sbin/411put --comment="#" /etc/auto.master
411 Wrote: /etc/411.d/etc.auto..master
Size: 417/134 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1
```

4) Una vez sincronizados los archivos de configuración, es necesario reiniciar este servicio en todo el cluster para garantizar que no existan inconvenientes en el montaje de los directorios hogar (Véase Capítulo 2 sección 2.2.1, apartado *Sistema de Archivos por Red*):

- Reinicio en el frontend:

```
[root@mixbaal ~]# service autofs reload
Checking for changes to /etc/auto.master....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
```

- Reinicio en todos los nodos:

```
[root@mixbaal ~]# cluster-fork service autofs reload
compute-0-0:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
compute-0-1: Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
compute-0-2:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
...
```

3.1.1 Comprobación de las Cuentas Creadas

Una vez que se han completado estos 4 pasos, el usuario ahora puede hacer uso de todos los servicios del cluster, no sin antes verificar que todo funcione correctamente. A continuación se enlista una serie de pasos que nos permiten hacer esta inspección:

1. Verificar que el usuario pueda conectarse al frontend, conectándose desde otra ubicación:

```
paulina@malicia:~$ ssh -l paulina mixbaal.supercomputo.unam.mx
paulina@mixbaal.supercomputo.unam.mx's password:
Last login: Sun Jul 12 23:56:33 2009 from malicia.super.unam.mx
Rocks Frontend Node - Mixbaal Cluster
Rocks 4.3 (Mars Hill)
Profile built 00:08 13-Apr-2008

Kickstarted 20:51 12-Apr-2008
[paulina@mixbaal ~]$
```

Nuestra prueba fue exitosa.

Si la respuesta es negativa, se puede deber a que:



- El password es incorrecto. Se recomienda asignarlo nuevamente.
- La cuenta no se creó correctamente. Verificar que ésta aparezca en el archivo `/etc/passwd`.

2. Revisar que el usuario encuentre su directorio hogar con el siguiente comando:

```
[paulina@mixbaal ~]$ pwd
/home/paulina
```



Si la salida anterior no es satisfactoria se sugiere reiniciar el servicio `autofs`.

3. Comprobar que el usuario, una vez que se ha conectado al frontend, puede ingresar a los demás nodos sin necesidad de una contraseña:

```
[paulina@mixbaal ~]$ ssh compute-0-9
Warning: Permanently added 'compute-0-9' (RSA) to the list of known
hosts.
Rocks Compute Node
Rocks 4.3 (Mars Hill)
Profile built 18:47 05-May-2008

Kickstarted 14:02 05-May-2008
[paulina@compute-0-9 ~]$
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Para nuestro caso, fue exitosa la prueba.

En caso de que el nodo solicite al usuario una contraseña, se pueden realizar uno o todos los pasos siguientes:



- Reiniciar el servicio `autofs` en el nodo, con el comando `cluster-fork service autofs reload`
- Revisar que en el archivo `/etc/passwd` el directorio hogar del usuario este como `/home/usuario` y no como `/export/home/usuario`
- Sincronizar todos los archivos de configuración



Es común que en la administración del cluster, se necesiten crear muchas cuentas simultáneamente, para este caso el uso del comando `useradd` no es lo más apropiado. Por lo que se puede recurrir al comando `chpasswd`, el cual permite fijar muchas contraseñas preestablecidas a un conjunto de usuarios con un solo comando.

A continuación se presenta un script de shell que permite la creación múltiple de usuarios a partir de una lista de nombres de usuario almacenados en un archivo de texto plano `usuarios.txt`:

```
[root@mixbaal ~]# more crear_multiples_users.sh
sufijo=pragma
for usuario in $(cat usuarios.txt)
do
echo "Creando $usuario..."
useradd $usuario
echo "$usuario:$usuario-$sufijo" > passwords.txt
done
chpasswd < passwords.txt
```

3.1.2 Creación de Cuentas para Usuarios PRAGMA

Como se mencionó en el capítulo 2, en la sección 2.1, entre los requerimientos que PRAGMA² ha establecido para formar parte de la grid, es la creación de una cuenta para el monitoreo de pruebas PRAGMA.

Debido a que el ambiente y los requerimientos de cada sistema son muy diversos, es responsabilidad del administrador del sistema implementar un proceso de creación y configuración de cuentas de usuario, como se analizó en el Capítulo 1 sección 1.1.1, apartado *Administración de Usuarios*. Sin embargo, es necesario contar con un estándar que facilite esta tarea, por lo que la comunidad PRAGMA ha definido un mecanismo para ello, el cual es aportación de uno de sus miembros.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Para la creación de la cuenta de monitoreo, en la página de PRAGMA está disponible el archivo `user account info pak3` (paquete de información de cuenta de usuario) el cual es un archivo TAR* que ejemplifica cómo debe ordenarse la información relativa a cada usuario, contiene un directorio, para este caso es `cindytest`, mismo que corresponde al nombre de usuario a crear, y éste a su vez, contiene los siguientes archivos:

- `authorized_keys` (llave privada del usuario)
- `b89793e4.0` (certificado)
- `b89793e4.signing_policy` (contiene datos necesarios para reconocer un certificado)
- `user.txt` (información de la cuenta de usuario)

A su vez, el contenido de estos archivos es:

Archivo `authorized_keys`:

```
[root@mixbaal ~]# more authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEArBRU6g1Jxd1QhJv9ZN/NX69Kt38e/WUreF8L1xwK
tG6jeTiibTjQJuZqiYDy8qGAX7NDu3G6twAePbGNuvs+FaWgQhPbyo2YUEXMMFFDlhKL7
eJ+Y/ijAbLgW6xNGGbJtYDn8m6VtxoD8//Mz+86zP9n+YLwKQszGOh1DE0WWSXM=
zhengc@rocks-52.sdsc.edu
```

Archivo `b89793e4.0`:

```
[root@mixbaal ~]# more b89793e4.0
-----BEGIN CERTIFICATE-----
MIIEAjCCAuqgAwIBAgIBADANBgkqhkiG9w0BAQQFADBjMQswCQYDVQQGEwJVUzEO
MAwGA1UEChMFTlBBQ0kxDTALBgNVBAsTBFBNEU0MxHDAaBgNVBAMTE0NlcnRpZmlj
YXRlIEElhbmFnZXIxFzAVBgoJkiaJk/IsZAEBEwdjZXXJ0bWFuMB4XDTAwMTAxMTIw
NTAxNlloXDTIwMTAxMTIwNTAxNlloYzELMAkGA1UEBhMCVVMxZjAMBgNVBAoTBW5Q
QUNJMQ0wCwYDVQQLEwRTRFNDMRwwGgYDVQQDEwNDZXJ0aWZpY2F0ZSBNYW5hZ2V5
MRcwFQYKCCZImiZPyLGQBARMHY2VydG1hbG1hbjCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBAJeeIpMfr3vwdZp1lnMwMgvnf/fwIjl2828EQ0pstUvntMOOru3w
re3at47gPJJ4xZJutRSHcsJZEtPlExsjauHaTZWqn80b/zirIAC/9873hbfsN7sy
r+TTfzpz4AghGe/xyyUtCir4ffQtgSiC6Y/J2z2grSgTFjPtZTYpwIoJOW52Rk2l
2fdVrXIjr4nqDe3ZTnRBwDWomuIelXqL2IxDEEh1LFpjTn+qvZyU+1MDE7r6twqD
2NJ1Sxg8r277XjNHFYC2F0qViT1fmGG18N+7SP9K00GmKDo5wkbKyivVbC2w3jdR
Qbo28VGskfdevMj1qrADi6LQi5jhfc1zDGMCAwEAAoBwDCBvTAdBgNVHQ4EFgQU
ZswI2f1jd8qmJFaG5wvTNnYwtqAwgY0GA1UdIwSBhTCBgoAUZswI2f1jd8qmJFaG
5wvTNnYwtqChZ6RlMGmxCzAJBgNVBAYTAlVTMQ4wDAYDVQQKEwVOUEFDSTENMASG
A1UECxMEU0RTQzEcmBoGA1UEAxMTQ2VydG1maWNhdGUgTWFuYXdlc2V5EXMBUGCgms
JomT8ixkAQETB2NlcnRtYW6CAQAwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0BAQQF
AAOCAQEafzMuN7S7/2y2xxh2f/xCw6wDX7y1ZdT8GNQn1pna jyxADy7m5+j6ez/
Oy9SH56y0Uvm5AjvivoTEeXyn0eHmg==
-----END CERTIFICATE-----
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Archivo `b89793e4.signing_policy`:

```
[root@mixbaal ~]# more b89793e4.signing_policy
# SDSC CA Policy
access_id_CA    X509      '/C=US/O=NPACI/OU=SDSC/CN=Certificate
Manager/USERID=certman'
pos_rights      globus   CA:sign
cond_subjects   globus   '/C=US/O=NPACI/*'
```

Archivo `user.txt`:

```
[root@mixbaal ~]# more user.txt
fullname=Cindy Test
affiliation=PRAGMA
address=UCSD/SDSC La Jolla, CA 92093-0505

country=USA
phone=858-534-5000
fax=858-534-5001
email=cindytest@sdsc.edu
purpose=routine-basis experiments
contribution=work
prefered1=cindytest
prefered2=cindytest1
prefered3=NONE
shell=bash
dn="/C=US/O=NPACI/OU=SDSC/CN=Cindy Zheng/USERID=zhengc"
hostname=rocks-52.sdsc.edu
hostip=198.202.88.52
urlcert=https://pragma-
goc.rocksclusters.org/secure/certificates/b89793e4.0
urlcrl=NONE
```

Ésta es la información que se requiere de cada usuario nuevo en el sistema.

Asimismo, se descargó el script de instalación disponible en la página web de PRAGMA, `pragma_setuser`⁴, el cual está escrito en lenguaje PERL, cuyo contenido se muestra a continuación:

```
#!/usr/bin/perl
#
# Writen by Cindy Zheng, for PRAGMA, 2005
#
print STDOUT ("\nPRAGMA_SET_USER script will perform the following
tasks:\n");
print STDOUT ("\n- Create an UNIX user account:\n");
```

```
print STDOUT ("- Install the user's ssh public key:\n");
print STDOUT ("- Install the user's site CA files:\n");
print STDOUT ("- Install the user's DN:\n");
print STDOUT ("\nThis script requires that you have all the
appropriate privileges to perform the above tasks\n\n");

#
# Check argument
#
if ($#ARGV != 0) {
    print "Usage: pragma_setuser filename\n";
    exit;
}

}

unless (-r $ARGV[0]) {
    die ("Fatal error: Cannot read $ARGV[0]\nAbort...\n");
}

$pakname = substr ($ARGV[0], 0, $len - 7);

#
# Unzip and untar files
#
system ("tar -zxf $ARGV[0]") && die ("Fatal error: Cannot unzip or
untar $ARGV[0]\nAbort...\n");

# Read user.txt file
open (userfile, "$pakname/user.txt") || die ("Fatal error: Cannot
open $pakname/user.txt!\nAbort...\n");
@userinfo = <userfile>;
close (userfile);
chomp (@userinfo);

# Get info from user.txt
foreach $line (@userinfo) {
    if ($line =~ /^dn=/) {
        $len = length ($line);
        $dn = substr ($line, 3, $len - 3);
        last;
    }
}

foreach $line (@userinfo) {
    if ($line =~ /^prefered1=/) {
        $len = length ($line);
        $prefered1 = substr ($line, 10, $len - 10);
        last;
    }
}
}
```

```

foreach $line (@userinfo) {
    if ($line =~ /^prefered2=/) {
        $len = length ($line);
        $prefered2 = substr ($line, 10, $len - 10);
        last;
    }
}
foreach $line (@userinfo) {
    if ($line =~ /^fullname=/) {
        $len = length ($line);
        $fullname = substr ($line, 9, $len - 9);
        last;
    }
}
print "dn=$dn\n";

# Choose a login name
LOGINNAME:
print STDOUT ("The prefered user login name:\n");
print STDOUT ("\t1. $prefered1\n");
print STDOUT ("\t2. $prefered2\n");
print STDOUT ("\t3. $prefered3\n");
print STDOUT ("\t4. or to specify a login name\n");
print STDOUT ("\t0. quit\n");
print STDOUT ("Your choice: ");
$answer = <STDIN>;
if ($answer =~ /^0/) {
    print STDOUT ("\nExit per your request... Good bye!\n\n");
    exit;
} elseif ($answer =~ /^1/) {
    $login = $prefered1;
} elseif ($answer =~ /^2/) {
    $login = $prefered2;
} elseif ($answer =~ /^3/) {
    $login = $prefered3;
} elseif ($answer =~ /^4/) {
    print STDOUT ("Enter the new login name: ");
    $login = <STDIN>;
    chomp ($login);
} else {
    print STDOUT ("\nInvalid choice, please re-enter\n\n");
    goto LOGINNAME;
}
CONFIRM:
print STDOUT ("The new login name will be $login. Continue?
(y\n): ");
$answer = <STDIN>;
if ($answer =~ /^[nN]/) {
    goto LOGINNAME;
}

```

```
# Create a new user account

system ("adduser -c '$fullname' $login") && die ("Fatal error:
Cannot create new user $login\n");
print STDOUT ("\nA new user account $login has been created.\n");

# Install authorized_keys file

$ssshdir = "~$login/.ssh";
if (-e $ssshdir) {
} else {
    system ("mkdir $ssshdir") && print STDOUT ("Error: Cannot
create $ssshdir directory\n");
}
system ("cp $pakname/authorized_keys $ssshdir") && print STDOUT
("Error: Cannont install $ssshdir/authorized_keys file");
system ("chown -R $login $ssshdir") && print STDOUT ("Error: Failed
to change ownership of $ssshdir directory tree\n");
system ("chgrp -R $login $ssshdir") && print STDOUT ("Error: Failed
to change group of $ssshdir directory tree\n");
system ("chmod 700 $ssshdir") && print STDOUT ("Error: Failed to set
permission to $ssshdir directory\n");
system ("chmod 600 $ssshdir/authorized_keys") && print STDOUT
("Error: Failed to set permission to authorized_keys file\n");
print STDOUT ("authorized_keys file has been installed.\n");

# Install site certificates

system ("cp $pakname/*.0 /etc/grid-security/certificates") && print
STDOUT ("Error: Cannont install site certificate file");
system ("cp $pakname/*.signing_policy /etc/grid-
security/certificates") && print STDOUT ("Error: Cannont install
site certificate signing policy file");
system ("chown root /etc/grid-security/certificates/*.0 /etc/grid-
security/certificates/*.signing_policy") && print STDOUT ("Error:
Failed to assign root ownership to certificate files\n");
system ("chgrp root /etc/grid-security/certificates/*.0 /etc/grid-
security/certificates/*.signing_policy") && print STDOUT ("Error:
Failed to assign root group to certificate files\n");
system ("chmod 644 /etc/grid-security/certificates/*") && print
STDOUT ("Error: Failed to set permission to certificate files\n");
print STDOUT ("Site certificate files have been installed.\n");
system ("chmod 644 /etc/grid-security/certificates/*") && print
STDOUT ("Error: Failed to set permission to certificate files\n");
print STDOUT ("Site certificate files have been installed.\n");

# Install user DN
```

```
open (mapfile, ">>/etc/grid-security/grid-mapfile") || die ("Error:
Failed to open /etc/grid-security/grid-mapfile\n");
print mapfile "$dn $login\n";
close(mapfile);
print STDOUT ("User DN has been added in grid-mapfile.\n");

print STDOUT ("All tasks are complete.\n\n");

#
# Review the results
#
REVIEW:
print STDOUT ("Would you like to review the results? \\\n\n): ");
$answer = <STDIN>;
if ($answer =~ /^[nN]/) {
    print STDOUT ("Good Bye!\n");
    exit;
} elsif ($answer =~ /^[yY]/) {
    print "\n< tail /etc/passwd >\n";
    system ("tail /etc/passwd");
    print "\n< ls -ld $ssshdir >\n";
    system ("ls -ld $ssshdir");
    print "\n< ls -l $ssshdir >\n";
    system ("ls -l $ssshdir");
    print "\n< ls -latr /etc/grid-security/certificates | tail
>\n";
    system ("ls -latr /etc/grid-security/certificates | tail");
    print "\n< ls -l /etc/grid-security/grid-mapfile >\n";
    system ("ls -l /etc/grid-security/grid-mapfile");
    print "\n< tail /etc/grid-security/grid-mapfile >\n";
    system ("tail /etc/grid-security/grid-mapfile");
    print "\n\nExit on completion\n\n";
} else {
    print STDOUT ("\nInvalid answer, please re-enter\n\n");
    goto REVIEW;
}

exit;
```

Para ejecutar el script, primero es necesario cambiar los permisos a fin de hacerlo ejecutable y asegurarse de que tiene todos los privilegios necesarios para crear cuentas de usuario y los archivos de certificado.

En nuestro caso se procedió a dar de alta la cuenta del usuario `cindytest`, con la información de los archivos y el script descritos.

```
[root@mixbaal ~]# chmod 7555 pragma_setuser
[root@mixbaal ~]# ./pragma_setuser cindytest.tar.gz
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Una vez creada la cuenta, se procedió a probarla con base en los tres pasos mencionados en la sección 3.1.1 de este capítulo:

1. Verificar que el usuario pueda conectarse al frontend desde otra ubicación:

```
paulina@malicia:~$ ssh -l cindytest mixbaal.supercomputo.unam.mx
cindytest@mixbaal.supercomputo.unam.mx's password:
Last login: Sun Jul 12 23:56:33 2009 from malicia.super.unam.mx
Rocks Frontend Node - Mixbaal Cluster
Rocks 4.3 (Mars Hill)
Profile built 00:08 13-Apr-2008

Kickstarted 20:51 12-Apr-2008
[cindytest@mixbaal ~]$
```

2. Revisar que el usuario encuentre su directorio hogar con el siguiente comando:

```
[cindytest@mixbaal ~]$ pwd
/export/home/paulina
```



La salida anterior indica que el home del usuario `cindytest` no se configuró correctamente



De modo que se siguieron los pasos 1 al 4 de la sección 3.1 de este capítulo. A saber:

- Asignar el home correcto con el comando `usermod`:

```
# usermod -d /home/paulina paulina
```

- Editar el archivo `/etc/auto.home`, quedando como sigue:

```
[root@mixbaal ~]# more /etc/auto.home
install      mixbaal.local:/export/home/&
condor      mixbaal.local:/export/home/condor
perla       mixbaal.local:/export/home/perla
paulina     mixbaal.local:/export/home/paulina
cindytest  mixbaal.local:/export/home/cindytest
```

Se añadió la última línea.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

- Sincronizar los archivos de configuración usando 411:

```
[root@mixbaal ~]# make -C /var/411 force
make: Entering directory `/var/411'
rm -rf /etc/411.d/*
make
make[1]: Entering directory `/var/411'
/opt/rocks/sbin/411put --comment="#" /etc/auto.home

411 Wrote: /etc/411.d/etc.auto..home
Size: 534/217 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

/opt/rocks/sbin/411put --comment="#" /etc/auto.master
411 Wrote: /etc/411.d/etc.auto..master
Size: 417/134 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1
```

- Reiniciar este servicio en todo el cluster

Reinicio en el frontend:

```
[root@mixbaal ~]# service autofs reload
Checking for changes to /etc/auto.master....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
```

Reinicio en todos los nodos:

```
[root@mixbaal ~]# cluster-fork service autofs reload
compute-0-0:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
compute-0-1: Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
/etc/auto.home
compute-0-2:
Checking for changes to /etc/auto.master ....
Reload map /usr/sbin/automount --timeout=1200 /share file
/etc/auto.share
Reload map /usr/sbin/automount --timeout=1200 /home file
...

```

- Comprobar que el usuario, una vez que se ha conectado al frontend, puede ingresar a los demás nodos sin necesidad de una contraseña:

```
[cindytest@mixbaal ~]$ ssh compute-0-9
Warning: Permanently added 'compute-0-9' (RSA) to the list of known
hosts.
Rocks Compute Node
Rocks 4.3 (Mars Hill)
Profile built 18:47 05-May-2008

Kickstarted 14:02 05-May-2008
[cindytest@compute-0-9 ~]$
```

Para nuestro caso, fue exitosa la prueba.

En caso de presentarse algún problema favor de revisar la sección 3.1.1 de este capítulo.

3.2 Instalación y Configuración de Aplicaciones

Con el objeto de cumplir con los requerimientos de PRAGMA expuestos en el Capítulo 2, sección 2.1, se instaló OpenBabel y Gamess. De modo que en esta sección se explicará la definición, instalación y pruebas de estas aplicaciones en el Cluster Mixbaal.

Debido a que dichas aplicaciones serán usadas principalmente para realizar cómputo químico, en la siguiente sección se dará una breve descripción de lo que es la Química Computacional.

3.2.1 Química Computacional

Como se mencionó en el Capítulo 1, sección 1.2.1, PRAGMA está conformado por tres grupos de trabajo, Biociencia, Teleciencia y Recursos y Datos. Particularmente miembros del grupo de Biociencia han participado activamente en proyectos como PRIME (Pacific Rim Undergraduate Experiences), el cual está conformado por estudiantes de la Universidad de California en San Diego, que llevan a cabo investigaciones en diversas ramas de la ciencia, como Química.

Es por ello que parte de los requerimientos de software para integrarse a la grid de PRAGMA son aplicaciones utilizadas en cómputo químico. La Química Computacional es una rama de la química teórica y de la química cuántica. Su objetivo es producir y utilizar programas informáticos para el estudio de las propiedades (como energía, momento dipolar*, frecuencias de vibración) de moléculas y, en menor medida, sólidos extendidos.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

En química teórica, los químicos y los físicos desarrollan algoritmos y teorías que permiten predicciones precisas de propiedades atómicas o moleculares, o caminos para las reacciones químicas. Los químicos computacionales usan los programas y metodologías existentes y los aplican a problemas químicos específicos.

3.2.2 OpenBabel

Es un conjunto de herramientas químicas diseñadas para convertir formatos de diferentes programas químicos. Es un proyecto de código abierto*, el cual permite a cualquier persona buscar, convertir, analizar, capturar, tabular, etc. datos para el estudio de la estructura molecular, química, bioquímica o áreas afines.⁵

Posee las siguientes características:

- Permite leer, escribir y convertir, más de 90 formatos de archivos químicos
- Soporta modelado molecular, bioinformática, cheminformatics*
- Contiene un completo conjunto de herramientas para el programador listas para usarse.

Instalación

La instalación se llevo a cabo en el nodo maestro para después replicarla en el resto de los nodos.

Instalación en el frontend

Para esto se siguió el procedimiento propuesto en la página de PRAGMA⁶:

1. Asegurarse de que las bibliotecas `libxml2` y `libxml2-devel` estén instaladas en el cluster. Esto lo hicimos con el siguiente comando:

```
[root@mixbaal ~]# rpm -q libxml2
libxml2-2.6.16-10
[root@mixbaal ~]# rpm -q libxml2-devel
package libxml2-devel is not installed
```



Con la opción `-q` del comando `rpm` se pueden conocer los paquetes instalados en el sistema.

Como podemos observar en la salida del comando anterior, la biblioteca `libxml2` sí la tenemos instalada, sin embargo la biblioteca `libxml2-devel` no.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

2. Por lo que procedimos a instalarla, buscando primero el paquete de instalación en el repositorio de rpms del cluster ubicado en `/usr/src/redhat/RPMS/i386/`, como no la encontramos ahí, realizamos una búsqueda en todo el sistema:

```
[root@mixbaal ~]# find / -name libxml2-devel*
/state/partition1/home/install/rocks-
dist/lan/i386/RedHat/RPMS/libxml2-devel-2.6.16-10.i386.rpm.torrent
/state/partition1/home/install/rocks-
dist/lan/i386/RedHat/RPMS/libxml2-devel-2.6.16-10.i386.rpm
/state/partition1/home/install/rolls/os/4.3/i386/RedHat/RPMS/libxml2-
-devel-2.6.16-10.i386.rpm
```

Ya que encontramos la biblioteca, procedimos a su instalación:

```
[root@mixbaal ~]# rpm -ivh /state/partition1/home/install/rocks-
dist/lan/i386/RedHat/RPMS/libxml2-devel-2.6.16-10.i386.rpm
Preparing...
##### [100%]
 1:libxml2-devel
##### [100%]
```

Revisamos que aparezca como paquete instalado:

```
[root@mixbaal ~]# rpm -q libxml2-devel
libxml2-devel-2.6.16-10
```

Una vez instaladas las bibliotecas anteriores, se puede iniciar la instalación de OpenBabel.

3. Descargar el paquete:

```
[root@mixbaal apps]# wget
http://downloads.sourceforge.net/project/openbabel/openbabel/2.2.2/o
penbabel-2.2.2.tar.gz
--2009-08-16 23:58:42--
http://downloads.sourceforge.net/project/openbabel/openbabel/2.2.2/o
penbabel-2.2.2.tar.gz
Resolving downloads.sourceforge.net... 216.34.181.59
Connecting to downloads.sourceforge.net|216.34.181.59|:80...
connected.
HTTP request sent, awaiting response... 302 Found
Location:
http://softlayer.dl.sourceforge.net/project/openbabel/openbabel/2.2.
2/openbabel-2.2.2.tar.gz [following]
```

```
--2009-08-16 23:58:43--
http://softlayer.dl.sourceforge.net/project/openbabel/openbabel/2.2.
2/openbabel-2.2.2.tar.gz
Resolving softlayer.dl.sourceforge.net... 74.86.229.28
Connecting to softlayer.dl.sourceforge.net|74.86.229.28|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 6621343 (6.3M) [application/x-gzip]
Saving to: `openbabel-2.2.2.tar.gz'

100%[=====>] 6,621,343  3.00M/s
in 2.1s

2009-08-16 23:58:45 (3.00 MB/s) - `openbabel-2.2.2.tar.gz' saved
[6621343/6621343]
```

4. Desempaquetamos el archivo descargado:

```
[root@mixbaal ]# tar xzvf openbabel-2.2.2.tar.gz
...
openbabel-2.2.2/windows-vc2005/OBPythonOBF/obpython.bat
openbabel-2.2.2/windows-vc2005/OBPythonOBF/OBPythonOBF.vcproj
openbabel-2.2.2/windows-vc2005/OBPythonOBF/Readme.txt
openbabel-2.2.2/windows-vc2005/OBPythonOBF/setup.py
openbabel-2.2.2/windows-vc2005/OBTools/
openbabel-2.2.2/windows-vc2005/OBTools/nbabel.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obchiral.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obconformer.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obenergy.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obfit.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obminimize.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obmm.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obprobe.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obprop.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obrotamer.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obrotate.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obseq.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/obsym.vcproj
openbabel-2.2.2/windows-vc2005/OBTools/Projects in OBTools.txt
openbabel-2.2.2/windows-vc2005/OBTools.sln
openbabel-2.2.2/windows-vc2005/OBUtilFormats/
openbabel-2.2.2/windows-vc2005/OBUtilFormats/OBUtilFormats.vcproj
openbabel-2.2.2/windows-vc2005/OBXMLFormats/
openbabel-2.2.2/windows-vc2005/OBXMLFormats/OBXMLFormats.vcproj
openbabel-2.2.2/windows-vc2005/OpenBabelDLL/OpenBabelDLL.vcproj
openbabel-2.2.2/windows-vc2005/OpenBabelOBF.sln
...
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

5. Cambiarse al directorio creado por el comando anterior:

```
[root@mixbaal ]# cd openbabel-2.2.2
[root@mixbaal openbabel-2.2.2]#
```

6. Crear el directorio `openbabel` en `/share/apps`, que como se mencionó en el Capítulo 2, sección 2.2.1, apartado *Sistema de Archivos por Red*, es una ruta accesible y útil para instalar aplicaciones en todos los nodos del cluster:

```
[root@mixbaal openbabel-2.2.2]# mkdir /share/apps/openbabel
```

7. Ejecutar el archivo `configure` con la opción `--prefix` dónde se le indicará el directorio en el que se instalará OpenBabel:

```
[root@mixbaal openbabel-2.2.2]# ./configure
--prefix=/share/apps/openbabel
```



La mayoría de los paquetes de instalación traen consigo un archivo `configure`, el cual es un script de autoconfiguración del programa a instalar, es decir contiene los valores especiales para la compilación de la aplicación. La ejecución de este comando crea un archivo `Makefile` el cual es la base de la compilación.

8. Ejecutar el comando `make`:

```
[root@mixbaal openbabel-2.2.2]# make
```



El comando `make` se encarga de leer todos los archivos `Makefiles` que el script `configure` haya creado, estos archivos le indican al comando `make` qué archivos debe compilar y en qué orden.

9. Ejecutar el comando `make install`, el cual iniciará la instalación en sistema:

```
[root@mixbaal openbabel-2.2.2]# make install
...
make[1]: Entering directory `/root/openbabel/openbabel-2.2.2'
make[2]: Entering directory `/root/openbabel/openbabel-2.2.2'
make[2]: Nothing to be done for `install-exec-am'.
test -z "/usr/share/apps/openbabel/lib/pkgconfig" || /bin/mkdir -p
"/usr/share/apps/openbabel/lib/pkgconfig"
/usr/bin/install -c -m 644 openbabel-2.0.pc
'/usr/share/apps/openbabel/lib/pkgconfig'
make[2]: Leaving directory `/root/openbabel/openbabel-2.2.2'
make[1]: Leaving directory `/root/openbabel/openbabel-2.2.2'
[root@mixbaal openbabel-2.2.2]#
```

10. Editar la variable \$PATH del sistema:

```
[root@mixbaal openbabel-2.2.2]# vi /etc/profile
...
if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

PATH=$PATH:/share/apps/openbabel/bin
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC
...
```

Se agregó la línea en negritas.

11. Agregar al PATH de PRAGMA el directorio de los binarios de la aplicación:

```
[root@mixbaal ~]# vi /etc/profile.d/pragma.sh
source /opt/vdt/setup.sh
export PATH=/share/apps/openbabel/bin:$PATH
```

Se agregó la línea en negritas.

12. Ejecutar la siguiente línea, para tener disponibles las bibliotecas de OpenBabel en el sistema:

```
[root@mixbaal ]# echo "/share/apps/openbabel/lib" >
/etc/ld.so.conf.d/apps.conf
```

13. Actualizar las bibliotecas en el frontend:

```
[root@mixbaal ]# ldconfig
```



Se recomienda ejecutar este comando después de que se instale una aplicación en el sistema.

Pruebas

14. Probar la aplicación. El comando principal de esta aplicación es `babel`, el cual es el que se encarga de convertir/traducir un archivo de entrada de algún programa químico a otro que se desee. Para esto se realizaron los pasos siguientes:

- Descargar archivos en formato `xyz`. Descargamos el paquete `xtal1.zip`⁷ el cual contiene coordenadas atómicas/moleculares de una colección de cristales en formato `xyz` mismos que vamos a utilizar para convertirlos a formato `pdb`.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

```
[root@mixbaal pruebas]# wget
http://web.uniovi.es/qcg/DetEst/xtall.zip
--2009-08-26 22:02:30-- http://web.uniovi.es/qcg/DetEst/xtall.zip
Resolving web.uniovi.es... 156.35.33.99
Connecting to web.uniovi.es|156.35.33.99|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5300 (5.2K) [application/zip]
Saving to: `xtall.zip'

100%[=====] 5,300      14.0K/s  in
0.4s

2009-08-26 22:02:31 (14.0 KB/s) - `xtall.zip' saved [5300/5300]

[root@mixbaal pruebas]#
```

- Descomprimir el paquete:

```
[root@mixbaal ]# unzip xtall.zip
Archive:  xtall.zip
  inflating: xtA1-111.xyz
  inflating: xtA2-111.xyz
  inflating: xtA3-552.xyz
  inflating: xtA4-111.xyz
  inflating: xtA4-333.xyz
  inflating: xtA9-331.xyz
  inflating: xtB1-111.xyz
[root@mixbaal ]#
```

- Contenido de un archivo xyz:

```
[root@mixbaal ]# more xtA1-111.xyz
14

CU          0.000000      0.000000      0.000000
CU          0.000000      0.000000      6.831290
CU          0.000000      6.831290      0.000000
CU          0.000000      6.831290      6.831290
CU          6.831290      0.000000      0.000000
CU          6.831290      0.000000      6.831290
CU          6.831290      6.831290      0.000000
CU          6.831290      6.831290      6.831290
CU          0.000000      3.415645      3.415645
CU          6.831290      3.415645      3.415645
CU          3.415645      0.000000      3.415645
CU          3.415645      6.831290      3.415645
CU          3.415645      3.415645      0.000000
CU          3.415645      3.415645      6.831290
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

- Conversión a formato `pdb` usando el comando `babel`, con las siguientes opciones:
 - `-ixyz`: indica que el archivo de entrada será un archivo en formato `xyz`
 - `-opdb`: indica que el archivo de salida será un archivo en formato `pdb`

```
[root@mixbaal ]# babel -ixyz xtA1-111.xyz -opdb ej_convertido.pdb
1 molecule converted
11 audit log messages
```

La salida anterior indica que la aplicación funciona correctamente, pues no marcó error y llevó a cabo la conversión sin problemas.

Verificación de resultados

- Verificamos el contenido del archivo de salida:

```
[root@mixbaal ]# more ej_convertido.pdb
COMPND      xtA1-111.xyz
AUTHOR      GENERATED BY OPEN BABEL 2.2.2
HETATM      1 CU    LIG      1      0.000    0.000    0.000    1.00    0.00
Cu
HETATM      2 CU    LIG      2      0.000    0.000    6.831    1.00    0.00
Cu
HETATM      3 CU    LIG      3      0.000    6.831    0.000    1.00    0.00
Cu
HETATM      4 CU    LIG      4      0.000    6.831    6.831    1.00    0.00
Cu
HETATM      5 CU    LIG      5      6.831    0.000    0.000    1.00    0.00
Cu
HETATM      6 CU    LIG      6      6.831    0.000    6.831    1.00    0.00
Cu
HETATM      7 CU    LIG      7      6.831    6.831    0.000    1.00    0.00
Cu
HETATM      8 CU    LIG      8      6.831    6.831    6.831    1.00    0.00
Cu
HETATM      9 CU    LIG      9      0.000    3.416    3.416    1.00    0.00
Cu
HETATM     10 CU    LIG     10      6.831    3.416    3.416    1.00    0.00
Cu
HETATM     11 CU    LIG     11      3.416    0.000    3.416    1.00    0.00
Cu
HETATM     12 CU    LIG     12      3.416    6.831    3.416    1.00    0.00
Cu
HETATM     13 CU    LIG     13      3.416    3.416    0.000    1.00    0.00
Cu
MASTER          0    0    0    0    0    0    0    0    14    0    14
0
END
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER



Los archivos en formato `pdb` son archivos de coordenadas moleculares que contienen información precisa sobre la estructura tridimensional de una molécula. La información de un archivo `pdb` se encuentra distribuida en diferentes líneas ó encabezados que contienen propiedades particulares de la molécula, estas líneas son⁸:

HEADER	Nombre y fecha de creación del archivo PDB.
COMPND	Nombre de la molécula.
SOURCE	Organismo del que se obtuvo la proteína.
AUTHOR	Lista de los Autores que proporcionaron la estructura molecular al PDB
REVDAT	Datos de revisión de la estructura de la proteína.
REMARK	Comentarios en relación a los artículos en donde se publicó la estructura molecular o sobre las características de la molécula.
SPRSDE	Lista de archivos de coordenadas para la misma estructura.
SEQRES	Secuencia de los aminoácidos de la proteína.
FTNOTE	Notas de pie de página. No todos los archivos lo tienen.
HET & FORMUL	Lista de cofactores, grupos prostéticos, inhibidores y otras sustancias no proteicas presentes.
HELIX, SHEET & TURN	Lista de los residuos con estructura secundaria en la proteína.
CRYST1, ORIG & SCALE	Información general sobre los cristales con los que se obtuvo la estructura por Rayos X
ATOM & HETATM	Contiene la información de la posición espacial en los ejes X,Y y Z de cada uno de los átomos, especificando el residuo y la cadena.
CONECT	Enlaces formados entre los átomos no proteicos presentes en el PDB.
MASTER & END	Indican la finalización del archivo.

Tabla 1. Estructura de un archivo en formato `pdb`.

Con esto se concluye que la instalación de la aplicación en el frontend fue exitosa.

Configuración en todos los nodos

Debido a que OpenBabel se instaló en el directorio `/share/apps`, que como ya se mencionó arriba, es un directorio accesible para todos los nodos, basta con replicar el archivo `/etc/profile` y el archivo `/etc/ld.so.conf.d/apps.conf` (pasos 11 y 12 de la sección anterior) en todos los nodos, para que éstos tengan disponible la aplicación:

1. Editar el archivo `/var/411/Files.mk`, (Véase Capítulo 2, sección 2.2.1, apartado *Servicio 411*):

```
[root@mixbaal ]# vi /var/411/Files.mk
...
AUTOMOUNT = $(wildcard /etc/auto.*)

# These files all take a "#" comment character.
# If you alter this list, you must do a 'make clean; make'.
FILES = $(AUTOMOUNT) \
        /etc/passwd \
        /etc/shadow \
        /etc/group \
        /etc/services \
        /etc/rpc \
        /etc/profile \
        /etc/ld.so.conf.d/apps.conf

# FILES += /my/file
FILES += /etc/sce/sce.conf
...
```

Se agregaron las líneas indicadas en negritas.

2. Sincronizar todos los archivos en el frontend:

```
[root@mixbaal ]# make -C /var/411 force
```

3. Replicarlo en todos los nodos:

```
[root@mixbaal openbabel]# cluster-fork 411get
compute-0-0:
/etc.rpc
/etc.group
/etc.shadow
/etc.auto..misc
/etc.sce.sce..conf
/etc.auto..home..respaldo
/etc.auto..master
/etc.passwd
/etc.profile
/etc.auto..home
/etc.auto..net
/etc.services
/etc.auto..share
...
```

4. Probar la aplicación en los nodos. Usamos el comando `cluster-fork` indicándole la ruta absoluta* del comando:

```
[root@mixbaal pruebas]# cluster-fork /share/apps/openbabel/bin/babel
-ixyz xtA1-111.xyz -opdb nodo.pdb
compute-0-0:
1 molecule converted
11 audit log messages
compute-0-1:
1 molecule converted
11 audit log messages
compute-0-2:
1 molecule converted
11 audit log messages
compute-0-3:
1 molecule converted
11 audit log messages
compute-0-4:
1 molecule converted
11 audit log messages
compute-0-5:
1 molecule converted
11 audit log messages
compute-0-6:
1 molecule converted
11 audit log messages
compute-0-7:
1 molecule converted
11 audit log messages
compute-0-8:
1 molecule converted
11 audit log messages
compute-0-9:
1 molecule converted
11 audit log messages
```

Como podemos observar funcionó correctamente, lo que indica que la instalación ha sido replicada y está disponible en todos los nodos del cluster.

3.2.3 Gamess

Gamess es un paquete de software de química cuántica ab initio*, sus siglas en inglés significan: General Atomic and Molecular Electronic Structure System (Sistema General de Estructura Electrónica, Atómica y Molecular). Fue desarrollado en el grupo del profesor M.S. Gordon de la Universidad del Estado de Iowa.

Permite realizar una gran diversidad de cálculos mecanocuánticos* de sistemas moleculares:

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

- cálculos ab initio*
- cálculos semiempíricos de orbitales moleculares
- cálculos de propiedades

Asimismo tiene implementados una gran variedad de métodos para diferentes tipos de cálculos teóricos:

- Métodos de campo autoconsistente: RHF, UHF, ROHF, MCSCF...
- Métodos Post Hartree-Fock (CI, CC, MBPT) y Teoría del Funcional de la Densidad (DFT)
- Métodos semiempíricos: MNDO, AM1 y PM3
- Cálculo de gradientes* y hessianas* para la optimización de puntos estacionarios, cálculo de frecuencias, polarizabilidades*, entre otros
- Métodos de solvatación* implícitos y explícitos
- Cálculo de efectos relativistas
- Cálculo de funciones de onda nucleares
- Cálculo de diferentes propiedades moleculares, cargas atómicas, potencial electrostático molecular
- Entre otros

Una vez obtenidos los cálculos requeridos, está disponible una variedad de programas de visualización de los resultados, por ejemplo MacMolPlt.

Instalación

La instalación se llevo a cabo en el nodo del frontend para después realizar la réplica en el resto de los nodos del cluster.

Instalación en el frontend

GAMESS fue instalado inicialmente en el nodo maestro. Para el proceso de instalación se llevó a cabo el procedimiento descrito en la página de PRAGMA⁹:

1. Se descargó la distribución correspondiente al sistema operativo utilizado:

```
[root@mixbaal ~]# wget http://goc.pragma-grid.net/softdepot/ gamess_2008_i386.tar.gz
```

2. Se seleccionó la ubicación de la instalación. Al igual que OpenBabel, la distribución se instaló en todos los nodos del cluster, incluyendo el frontend, en el directorio /share/apps/gamess_2008. Para esto, se ejecutó el siguiente comando:

```
[root@mixbaal ~]# tar zvxFC gamess_2008_.tar.gz /share/apps/
```

Configuración en el frontend

3. Se revisó el archivo `/share/apps/gamess_2008/rungms` con el objeto de verificar que contenga los valores correctos para las variables `GAMESS_DIR`, `SCR` y `NCPUS` de acuerdo a lo dispuesto en el procedimiento descrito:

```
[root@mixbaal gamess_2008]# vi rungms
#!/bin/csh
#
# This is a C-shell script to execute GAMESS, by typing
# rungms JOB VERNO NCPUS >& JOB.log &
# JOB      is the name of the 'JOB.inp' file to be executed,
# VERNO    is the number of the executable you chose at 'lked' time,
# NCPUS    is the number of processors to be used.
#
# Customizing:
# a) choose a directory GAMESS_DIR where gamess distribution is
installed
# b) choose a directory SCR where large temporary files can
reside.
# This should be the fastest possible disk access, very spacious,
# and almost certainly a local disk.
# Translation: do not put these files on a slow network file
system!
# c) change the NCPUS' default if needed.
#
# ===== start custom =====
set TARGET=sockets
set GAMESS_DIR=/share/apps/gamess_2008           # custom a)
set TEST_DIR=$GAMESS_DIR/tests
set SCR=/state/partition1/$USER                 # custom b)

if (! -d $SCR) then
    mkdir $SCR
endif
#
set JOB=$1      # name of the input file xxx.inp, give only the xxx
part
set VERNO=$2    # revision number of the executable created by
'lked' step

# provide defaults if last two arguments are not given to this
script

if (null$VERNO == null) set VERNO=00
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

```
if (null$NCPUS == null) set NCPUS=1          # custom c)
set NCPUS=$3      # number of compute processes to be run

# provide defaults if last two arguments are not given to this
script

if (null$VERNO == null) set VERNO=00

if (null$NCPUS == null) set NCPUS=1          # custom c)

# ===== end custom =====
#   Customizing
...
```

Para estas variables se deben definir los valores siguientes:

- GAMESS_DIR: ubicación seleccionada para la instalación, por omisión es /share/apps/gamess_2008
- SCR: Directorio para archivos temporales, por omisión es /state/partition1/\$USER
- NCPUS: Número de CPUs a utilizar para cada job run, por omisión es 1

El archivo `runqms` es un script de C-shell* que permite ejecutar Gamess mediante el lanzamiento de trabajos con algún planificador, como puede ser SGE* o PBS*, asimismo incluye diferentes tipos de instrucciones para los diferentes sistemas de colas*.

4. Una vez verificado el script, se adicionó la ubicación del aplicativo en el PATH, editando el archivo `/etc/profile`, adicionando la línea:
`PATH=$PATH:/share/apps/gamess_2008`

```
[root@mixbaal ~]# vi /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

...

PATH=$PATH:/share/apps/openbabel/bin
PATH=$PATH:/share/apps/gamess_2008
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC

...
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

Configuración en los nodos

5. En este punto, Gamess ha sido instalado en el directorio `/share/apps`, accesible para todos los nodos, por lo que el paso siguiente es replicar el archivo `etc/profile` en los nodos restantes para que puedan hacer uso de la aplicación:

- Sincronizar archivos en el frontend

```
[root@mixbaal ~]# make -C /var/411 force
```

- Replicarlos en todo los nodos

```
[root@mixbaal openbabel]# cluster-fork 411get
compute-0-0:
/etc.rpc
/etc.group
/etc.shadow
/etc.auto..misc
/etc.sce.sce..conf
/etc.auto..home..respaldo
/etc.auto..master
/etc.passwd
/etc.profile
/etc.auto..home
/etc.auto..net
/etc.services
/etc.auto..share
...
```

6. A fin de utilizar Gamess de forma interactiva o bien, a través de un planificador de trabajos, antes de utilizarlo por vez primera, es necesario crear un directorio `scratch*` en cada nodo para almacenar de forma temporal los archivos que se generan durante el proceso de cada trabajo, para ello se ejecutó el siguiente comando:

```
[root@mixbaal ~]# cluster-fork mkdir /state/partition1/$USER
```

Pruebas

7. Se procedió a generar una prueba con el usuario `perla`, de modo que se hizo el cambio de cuenta de usuario y se creó el directorio `gamess` para el almacenamiento de los archivos de entrada y salida utilizados por el aplicativo:

```
[root@mixbaal ~]# su - perla
[perla@mixbaal ~]$ mkdir gamess
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

8. Se preparó el archivo `gamessJob.sge` para lanzar los trabajos utilizando el planificador SGE*:

```
[perla@mixbaal gamess]$ vi gamessJob.sge
#!/bin/bash
cd gamess
/share/apps/gamess_2008/rungms exam01 00 1
```



La línea del script de lanzamiento `/share/apps/gamess_2008/rungms exam01 00 1` nos indica:

- `/share/apps/gamess_2008/rungms`: comando utilizado para la ejecución de `gamess`
- `exam01`: archivo de entrada
- `00`: versión utilizada
- `1`: número de cpus utilizados

9. Se lanzó el proceso, ejecutando el archivo creado en el punto anterior, con el comando `qsub`:

```
[perla@mixbaal gamess]$ qsub gamessJob.sge
Your job 94 ("gamessJob.sge") has been submitted
```

Lo cual nos devuelve un número, en este caso el 94, el cual identifica al proceso.



Este número es importante ya que servirá para solicitar información sobre su ejecución.

Para comprobar que realmente el proceso está en la cola de trabajos, ejecutamos el comando `qstat`

```
[perla@mixbaal gamess]$ qstat

job-ID prior  name user state submit/start at queue slots ja-task-ID
-----
 17      0.55500 scheduler_ irving Eqw 06/24/2009 20:40:13 1
 18      0.55500 scheduler_ irving Eqw 06/24/2009 21:30:53 1
 19      0.55500 scheduler_ irving Eqw 06/25/2009 15:24:10 1
 20      0.55500 scheduler_ irving Eqw 06/25/2009 15:29:05 1
 21      0.55500 scheduler_ irving Eqw 06/26/2009 20:41:12 1
 22      0.55500 scheduler_ irving Eqw 06/29/2009 19:03:52 1
 90      0.55500 data      irving Eqw 08/17/2009 21:50:50 1
 94      0.00000 gamessJob. perla  qw 08/25/2009 04:05:19 1
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

De esta forma el trabajo es enviado a la cola de trabajos y el planificador se encarga de asignarlo al nodo menos ocupado para su ejecución.

Verificación de los resultados

10. El archivo de entrada exam01.inp, contiene la siguiente estructura y contenido:

```
[perla@mixbaal gamess]$ more exam01.inp

!      Using the default search METHOD=STANDARD,
!      FINAL E= -37.2322678015, 8 iters, RMS grad= .0264308
!      FINAL E= -37.2351919062, 7 iters, RMS grad= .0202617
!      FINAL E= -37.2380037239, 7 iters, RMS grad= .0013100

$CONTRL SCFTYP=RHF RUNTYP=OPTIMIZE COORD=ZMT NZVAR=0 $END
$SYSTEM TIMLIM=1 $END
$STATPT OPTTOL=1.0E-5 $END
$BASIS  GBASIS=STO NGAUSS=2 $END
$GUESS  GUESS=HUCKEL $END
$DATA
Methylene...1-A-1 state...RHF/STO-2G
Cnv  2

C
H  1 rCH
H  1 rCH  2 aHCH

rCH=1.09
aHCH=110.0

$END
```

Y resultado de la ejecución del job* 94, se generó el archivo de salida exam01.dat, con el siguiente contenido:

```
[perla@mixbaal gamess]$ more exam01.dat

$DATA
Methylene...1-A-1 state...RHF/STO-2G
CNV      2

C          6.0          0.0000000000          0.0000000000          -0.0899124183
  STO      2

H          1.0          0.8928757283          0.0000000000          0.5352858974
  STO      2

$END
```

y el módulo de optimización geométrica. La siguiente figura muestra cómo puede ser visualizado un resultado obtenido de Gamess con el visualizador MacMolPlt.

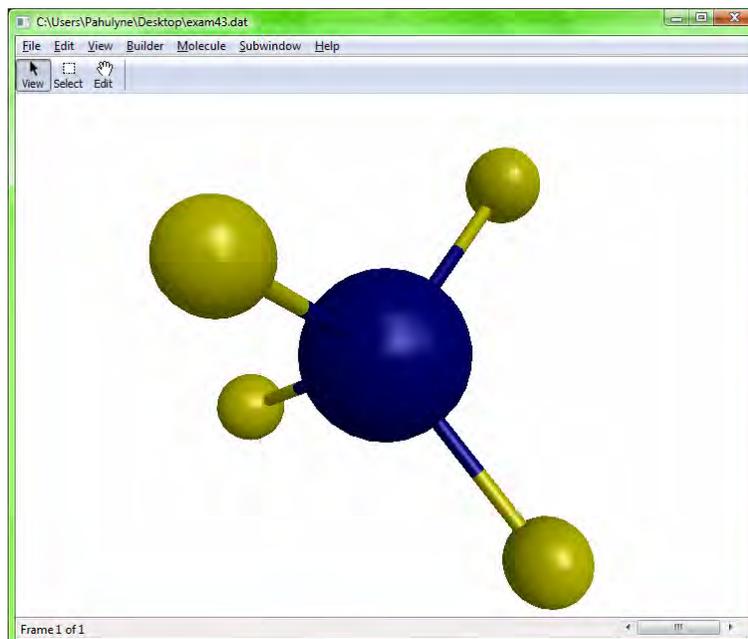


Figura 1. Visualización de resultados de Gamess, con el programa MacMolPlt.

Con lo que se concluye que la instalación ha sido exitosa.

3.3 Actualización Automática de Certificados

Como se ha venido explicando a lo largo del presente trabajo, el compartir recursos en una grid es un labor que se debe ejecutar de forma segura. De aquí surge la necesidad de manejar certificados digitales de los usuarios de la grid. Tarea que se ve facilitada con la actualización de éstos de forma automática a través de diferentes herramientas disponibles para ello.

Para nuestro caso, la herramienta instalada y configurada fue VDT (Virtual Data Toolkit) cuyo objetivo es hacer que los usuarios mantengan y utilicen software grid de la forma más fácil posible.

Entre los componentes que ayudan al funcionamiento de una grid, cuenta con software que permite actualizar las listas de revocación de certificados de CA (fetch-crl), así como software que trabaja en conjunto con GUMS (Grid User Management System).¹⁰

Uno de los comandos importantes de esta herramienta es `vdt-update-certs`, pues es el que ayuda a mantener los certificados de CA actualizados. Se ejecuta desde un

cron, el cual verifica una vez al día, si hay un nuevo paquete de certificados disponibles.¹¹

Por otro lado, es importante señalar que VDT utiliza PACMAN (gestor de paquetes) para su distribución, instalación y configuración.¹²

3.3.1 Instalación de VDT

Para la instalación y configuración de VDT se siguió el procedimiento propuesto por PRAGMA¹³, descrito a continuación:

1. Instalar pacman:

- Descargar el paquete:

```
[root@mixbaal ~]# wget
http://physics.bu.edu/pacman/sample_cache/tarballs/pacman-
latest.tar.gz
```

- Cambiarse al directorio /opt y extraer ahí los archivos del paquete descargado:

```
[root@mixbaal ~]# cd /opt
[root@mixbaal opt]# tar xvzf /root/pacman-latest.tar.gz
```

- Cambiarse al directorio creado:

```
[root@mixbaal opt]# cd pacman-3.26/
```

- Instalar la aplicación:

```
[root@mixbaal pacman-3.26]# source ./setup.sh
...

/root/vdt/vdt/setup/configure_edg_make_gridmap:
add_uninstall_command("$register --name edg-mkgridmap --remove");
/root/vdt/vdt/backup/vdt/fetch-crl/sbin/fetch-crl_001_20090718-
173026:# $Id: edg-fetch-crl.cin,v 1.14 2007/09/14 14:07:34 pmacvsdg

Exp $
/root/vdt/vdt/backup/vdt/fetch-crl/sbin/fetch-crl_001_20090718-
173026:# File:          edg-fetch-crl
#
/root/vdt/vdt/backup/vdt/fetch-crl/sbin/fetch-crl_001_20090718-
173026:# Usage:       edg-fetch-crl [-h|--help]
```

```
#  
/root/vdt/vdt/backup/vdt/fetch-crl/sbin/fetch-crl_001_20090718-  
173026:#          edg-fetch-crl [-l|--loc locationDirectory]  
#  
...
```

2. Instalar vdt:

Crear el directorio `/opt/vdt` y movernos a éste:

```
[root@mixbaal ~]# mkdir /opt/vdt  
[root@mixbaal ~]# cd /opt/vdt
```

Instalar VDT utilizando el comando `pacman`, instalado en el paso anterior:



La instalación realizará una serie de preguntas, mismas que se deben responder con `yes`.

```
[root@mixbaal ~]# pacman -get http://vdt.cs.wisc.edu/vdt_1101_cache:CA-  
Certificates
```

En el archivo `/opt/vdt/vdt/etc/vdt-update-certs.conf` añadir la línea `cacerts_url=http://rocks56.sdsc.edu/certs/igtf-pragma-ca-certs-version`:

```
[root@mixbaal ~]# vi /opt/vdt/vdt/etc/vdt-update-certs.conf  
...  
## statement and cannot use the exclude_ca statement instead,  
## please contact the VDT and let us know.  
cacerts_url=http://rocks56.sdsc.edu/certs/igtf-pragma-ca-certs-  
version
```

Ejecutar los siguientes comandos:

```
[root@mixbaal vdt]# . $VDT_LOCATION/setup.sh  
[root@mixbaal vdt]# . $VDT_LOCATION/vdt-questions.sh  
[root@mixbaal vdt]# $VDT_LOCATION/vdt/sbin/vdt-setup-ca-certificates  
--root
```



Se usó la opción `--root` ya que nuestro directorio de certificados lo tenemos en `/etc/grid-security/certificates`. De lo contrario, se debe usar la opción `--local`.



La instalación realizará una serie de preguntas, mismas que se deben responder con `yes`

Revisar que los servicios VDT se encuentren habilitados:

```
[root@mixbaal vdt]# vdt-control --list
Service          | Type   | Desired State
-----+-----+-----
fetch-crl        | cron   | enable
vdt-rotate-logs  | cron   | enable
vdt-update-certs | cron   | enable
```



Si alguno de los servicios de la salida anterior, se encuentra desactivado, se puede habilitar ejecutando `# vdt-control --enable <service-name>`
Para habilitar todos los servicios VDT: `# vdt-control -on`



Logs disponibles: `$VDT_LOCATION/vdt/var/log/vdt-update-certs.log` y `/etc/grid-security/certificates`.

3.3.2 Configuración para la Actualización Automática de Certificados

Una vez que instalamos VDT, debemos llevar a cabo la siguiente serie de pasos¹⁴ para que la actualización de los certificados de los usuarios PRAGMA con Mixbaal se haga de manera automática:

1. Instalar `edg-mkgridmap` en `/opt/vdt` utilizando el comando `pacman`:



Antes de instalar la aplicación, es importante que se respalde el archivo `/etc/grid-security/grid-mapfile`.



La instalación realizará una serie de preguntas, mismas que se deben responder de acuerdo a las políticas y requerimientos del sitio.

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

```
[root@mixbaal vdt]# pacman -get
http://vdt.cs.wisc.edu/vdt_1100_cache:EDG-Make-Gridmap
Do you want to add [http://vdt.cs.wisc.edu/vdt_1100_cache] to
[trusted.caches]? (y/n/yall): y
Beginning VDT prerequisite checking script vdt-common/vdt-prereq-
check...
All prerequisite checks are satisfied.
VDT 1.10.0 installs a variety of software, each with its own
license.
In order to continue, you must agree to the licenses.
You can view the licenses online at:
    http://vdt.cs.wisc.edu/licenses/1.10.0
After the installation has completed, you will also be able to view
the licenses in the "licenses" directory.
Do you agree to the licenses? [y/n] y
Several services provided by the VDT create unbounded log files.
If you wish, we can rotate those file on a daily basis.
Would you like to setup daily rotation of VDT log files?
Possible answers:
    y: Yes, I want the service to run automatically (once enabled)
    n: No, I do NOT want the service to run automatically
Note: Services are enabled with vdt-control; see 'post-
install/README'.
y
Do you want to run a cron job that will update the CA certificate
revocation lists automatically?
This will use the fetch-crl program that comes with the VDT.
The cron job will run at a random time between midnight and 6:00am.
We select a random time to avoid having all VDT installations
fetching CRLs at the same time.
Do you want to update the CA certification revocation lists (CRLs)
automatically? [y/n] y
The VDT typically installs public certificates and signing policy
files for the well-known public CA's.
This is necessary in order for you to perform GSI authentication
with any remote Grid services
(that have service/host certificates signed by these CA's).
For more information please refer to the VDT documentation:
http://vdt.cs.wisc.edu/releases/1.10.0/setup_ca.html
Where would you like to install CA files?
Choices:
    r (root) - install into /etc/grid-security/certificates (existing
CA files will be preserved)
    l (local) - install into $VDT_LOCATION/globus/share/certificates
    n (no)    - do not install
r
Do you want edg-mkgridmap daemon to be run automatically?
If so, we will run it four times a day via cron.
edg-mkgridmap will update your gridmap file by communicating with VOMS
servers.
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

```
This will only be useful if you are part of a Virtual Organization (VO) or
you allow users from VOs.
Do you want edg-mkgridmap daemon to be run via cron? [y/n] y
Do you want to automatically update your CA certificates?
If so, we will check for updates once a day via cron.
Do you want to automatically update your CA Certificates? [y/n] y
```



El comando `edg-mkgridmap` es una herramienta que sirve para construir el archivo `grid-mapfile` desde servidores VO* tomando en cuenta tanto las políticas VO* como las políticas locales.

2. Configurar `vdt-update-certs`. Para esto descargamos los certificados digitales de PRAGMA¹⁵ en `/etc/grid-security/certificates`.

3. Crear la cuenta de usuario `pragmauser` para los usuarios PRAGMA

```
[root@mixbaal ~]# useradd pragmauser
```

4. Agregar las líneas en negritas, al final del archivo `/opt/vdt/edg/etc/edg-mkgridmap.conf`:

```
[root@mixbaal ~]# vi /opt/vdt/edg/etc/edg-mkgridmap.conf
...
#####
# ALLOW and DENY: deny|allow pattern_to_match
# allow *INFN*
group vomss://vomrs-pragma.sdsc.edu:8443/voms/PRAGMA?/PRAGMA/USERS
pragmauser

#####
# Local grid-mapfile to import and override all the above
information.
# eg, gmf_local /opt/vdt/edg/etc/grid-mapfile-local

gmf_local /opt/vdt/edg/etc/grid-mapfile-local
```

5. Copiar el archivo `/etc/grid-security/grid-mapfile` a `/opt/vdt/edg/etc/grid-mapfile-local`:

```
[root@mixbaal ~]# cp /etc/grid-security/grid-mapfile
/opt/vdt/edg/etc/grid-mapfile-local
```

6. Crear el archivo `/opt/vdt/edg/log/edg-mkgridmap.log` usando el comando `touch`:

```
[root@mixbaal ~]# touch /opt/vdt/edg/log/edg-mkgridmap.log
```

CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

7. Respalidar y borrar el archivo `/etc/grid-security/grid-mapfile`:

```
[root@mixbaal ~]# cp /etc/grid-security/grid-mapfile
[root@mixbaal ~]# rm -f /etc/grid-security/grid-mapfile
```

8. Ejecutar por primera vez el comando `edg-mkgridmap`:

```
[root@mixbaal ~]# touch /opt/vdt/edg/log/edg-mkgridmap.log
[root@mixbaal ~]# /opt/vdt/edg/sbin/edg-mkgridmap --
output=/etc/grid-security/grid-mapfile
```

9. Si todo está correcto, se procede a iniciar el servicio `cron`:

```
[root@mixbaal ~]# source setup.sh
[root@rocks-96 vdt]# vdt-control --on edg-mkgridmap
enabling cron service edg-mkgridmap... no crontab for root
ok
[root@rocks-96 vdt]# crontab -l
18 1,7,13,19 * * * /opt/vdt/edg/sbin/edg-mkgridmap >>
/opt/vdt/edg/log/edg-mkgridmap.log 2>&1
```



El error `no crontab for root` de la salida anterior puede ser ignorado.

10. Instalar `VOMS-Client`:

```
[root@mixbaal ~]# pacman -get
http://vdt.cs.wisc.edu/vdt_1100_cache:VOMS-Client
```

11. Crear el archivo `/opt/vdt/glite/etc/vomses/PRAGMA` con la siguiente línea:

```
[root@mixbaal ~]# vi /opt/vdt/glite/etc/vomses/PRAGMA
"PRAGMA" "vomrs-pragma.sdsc.edu" "15001" "/DC=NET/DC=PRAGMA-
GRID/OU=SDSC/CN=vomrs-pragma.sdsc.edu" "PRAGMA" "https://vomrs-
pragma.sdsc.edu:443/vomrs/PRAGMA/services/VOMRS?WSDL"
```

12. Añadir la siguiente línea al archivo `/etc/profile.d/pragma.sh`:

```
[root@mixbaal ~]# vi /etc/profile.d/pragma.sh
source /opt/vdt/setup.sh
```

13. Probar el `VOMS*` client, autenticándonos con el usuario `pragmauser`, quien es miembro del grupo `PRAGMA/USERS`:



CAPÍTULO 3. PUESTA A PUNTO DEL CLUSTER

```
[pragmauser@rocks-96 ~]$ voms-proxy-init -voms PRAGMA -order
/PRAGMA/USERS
Enter GRID pass phrase:
Your identity: /C=US/O=SDSC/OU=SDSC/CN=Cindy Zheng/USERID=zhengc
Creating temporary proxy
..... Done
Contacting vomrs-pragma.sdsc.edu:15001 [/DC=NET/DC=PRAGMA-
GRID/OU=SDSC/CN=vomrs-pragma.sdsc.edu] "PRAGMA" Done
Creating proxy ..... Done
Your proxy is valid until Fri Aug 22 02:26:38 2008
```

Como podemos observar el acceso al cluster se realizó de forma exitosa.

Con la ejecución de las actividades descritas en los Capítulos 2 y 3 se ha culminado con la configuración y puesta a punto del cluster Mixbaal, de manera que se encuentra listo para ser integrado a la grid de PRAGMA, una vez que el Departamento de Seguridad en Cómputo de la DGSCA otorgue los certificados oficiales, y éstos sean enviados a la Coordinadora de PRAGMA, posibilitando el acceso de esta comunidad al cluster Mixbaal.

Fuentes de Consulta

- ¹ Zuluaga, Jorge, *Linux Clustering con Rocks. Una Guía Práctica. Guía 3 :Administración básica de la plataforma*, Grupo de Física y Astrofísica Computacional, MACOM, Instituto de Física, Universidad de Antioquia, Colombia, p. 42 [PDF] URL <<http://urania.udea.edu.co/facom/facom-products/documentation.php>>
- ² PRAGMA, *New User Setup in PRAGMA testbed* [en línea] URL <<http://pragma-goc.rocksclusters.org/softdepot/setuser.html>>
- ³ *ibid*, [en línea] URL <<http://goc.pragma-grid.net/secure/updaccounts/zhengc.tar.gz>>
- ⁴ *ibid*, [en línea] URL <http://goc.pragma-grid.net/secure/updaccounts/pragma_setuser>
- ⁵ OpenBabel, *OpenBabel: The Open Source Chemistry Toolbox*, [en línea] URL <http://openbabel.org/wiki/Main_Page>
- ⁶ PRAGMA, *PRIME2009*, [en línea] URL <<http://goc.pragma-grid.net/wiki/index.php/PRIME2009>>
- ⁷ Luaña, Victor, *Espectroscopia molecular, curso 2001-2002*, Grupo de Química Cuántica, Departamento de Química Física y Analítica, Universidad de Oviedo, [en línea] URL <<http://web.uniovi.es/qcg/qcg-doc0.html>>
- ⁸ Anónimo, *Propiedades de los Archivos PDB*, Academia Colombiana de Ciencias Exactas, Físicas y Naturales, [en línea] URL <<http://www.accefyn.org.co/rasmol/PDB.htm>>
- ⁹ PRAGMA, *GAMESS US*, [en línea] URL <http://goc.pragma-grid.net/wiki/index.php/GAMESS_US>
- ¹⁰ Stackoverflow, *Virtual Data Toolkit*, [en línea] URL <<http://stackoverflow.com/questions/8626/globus-toolkit-virtual-machine>>
- ¹¹ VDT, *CA-Certificates-Updater*, [en línea] URL <<http://vdt.cs.wisc.edu/releases/2.0.0/notes/CA-Certificates-Updater.html>>
- ¹² The Globus Alliance, *Virtual Data Toolkit (VDT)*, [en línea] URL <http://www.globus.org/grid_software/packaging/vdt.php>
- ¹³ PRAGMA, *Setup Certificates Update*, [en línea] URL <http://goc.pragma-grid.net/wiki/index.php/Setup_Certificates_Update>
- ¹⁴ *Ídem*.
- ¹⁵ PRAGMA, *Setup Certificates Update*, [en línea] URL <<https://goc.pragma-grid.net/secure/certificates/pragma-certs.tar.gz>>

CONCLUSIONES



Con el presente trabajo, se generó la documentación del procedimiento seguido para la configuración y puesta a punto del cluster Mixbaal en la grid de PRAGMA, con la cual el Departamento de Supercómputo cuenta con una referencia documental para su futura consulta y así permitir la continuidad operativa del cluster Mixbaal, en proyectos similares del Departamento, o bien, para el aprovechamiento de quienes requieran iniciarse en la construcción de clusters.

Partiendo de los objetivos iniciales, a saber:

Objetivo General

“Configurar y poner a punto el Cluster Intel Pentium III Beowulf de 10 nodos, denominado Mixbaal, perteneciente al Departamento de Supercómputo, que permita realizar cómputo de alto rendimiento para que la UNAM se integre a la grid de PRAGMA y éste sea utilizado por las 35 instituciones que forman parte del proyecto. Así como generar la documentación del procedimiento seguido para lograr lo anterior.”

Objetivos Específicos

- “Instalar los 10 nodos que formarán parte del cluster.
- Configurar la herramienta Globus.
- Crear las cuentas necesarias para el consejo de la asamblea PRAGMA, así como las cuentas de administración para el área de Administración de Sistemas Unix del Departamento de Supercómputo.
- Instalar dos aplicaciones de cómputo científico
- Configurar los certificados de seguridad correspondientes.
- Realizar pruebas de funcionamiento del cluster.”

Y de forma general, las actividades que se llevaron a cabo durante el proceso de configuración del cluster, fueron:

- Instalación del sistema operativo Rocks
- Instalación de middleware Globus Toolkit
- Instalación de aplicaciones de Cómputo Científico
- Creación de cuentas de usuario

Como resultado del conjunto de las actividades anteriormente expuestas, se logró nuestro objetivo de configurar y poner a punto el cluster de manera que se encuentra listo para ser integrado a la grid de PRAGMA, una vez que el Departamento de Seguridad en Cómputo de la DGSCA otorgue los certificados oficiales, y éstos sean enviados a la Coordinadora de PRAGMA, posibilitando el acceso de esta comunidad al cluster Mixbaal. Así como, permitir que la UNAM cumpliera con el compromiso de aportar un cluster a la grid de PRAGMA.

Adicionalmente, será posible la ejecución de los experimentos de cómputo científico propuestos por el GOC, así como coordinar y compartir recursos, aplicaciones, datos y almacenamiento entre las múltiples organizaciones que conforman la Asamblea, mismas que están distribuidas geográficamente a lo largo de la Cuenca del Pacífico.

Por otro lado, comprobamos que los clusters son una de las herramientas más útiles y más económicas del supercómputo. Pueden implementarse con una serie de computadoras ordinarias conectadas entre sí de tal modo que funcionan como un solo recurso de cómputo, siendo transparente para el usuario.

Aunado al punto anterior, el sistema operativo Rocks, especialmente diseñado para la configuración de clusters, permitió realizar la tarea de forma más sencilla, por lo cual lo recomendamos ampliamente para administradores de sistemas con poca experiencia en la implementación de clusters.

Asimismo, la incorporación de la UNAM a la grid de PRAGMA, nos permitió visualizar que las grids son una consecuencia positiva de cohesionar a diversas comunidades de cómputo de alto rendimiento, lo que se traduce en colaboraciones más estrechas entre éstas, permitiendo el fomento de la investigación científica en el país, a la vez que se abre la posibilidad de colaborar en más proyectos internacionales, empleando recursos de otras instituciones a través de los servicios grid.

Igualmente, podemos concluir que el desarrollo de este proyecto nos posibilitó de un acercamiento significativo a una de las áreas de estudio de la Informática: la Administración de Sistemas, ya que al configurar el cluster, realizamos las tareas propias de un Administrador, tales como la puesta en marcha del sistema, su configuración y mantenimiento, la definición de políticas para el control de acceso, la instalación de aplicaciones y creación de cuentas de usuario, entre otras, mismas que inherentemente conllevan a una serie de problemáticas como versiones incompatibles de las aplicaciones de acuerdo con el sistema operativo utilizado, errores durante la compilación de aplicaciones, escasa documentación en temas particulares, por mencionar algunos, las cuales fueron solventadas, permitiendo aplicar y desarrollar las habilidades adquiridas durante nuestra formación académica y con el apoyo de personal experto en la materia.

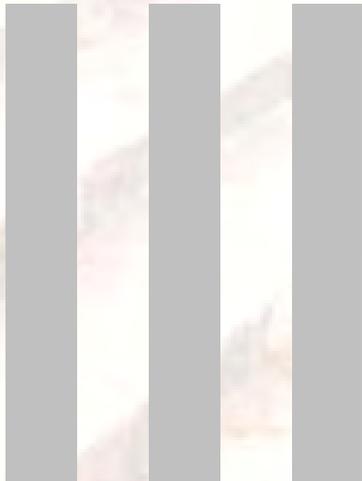
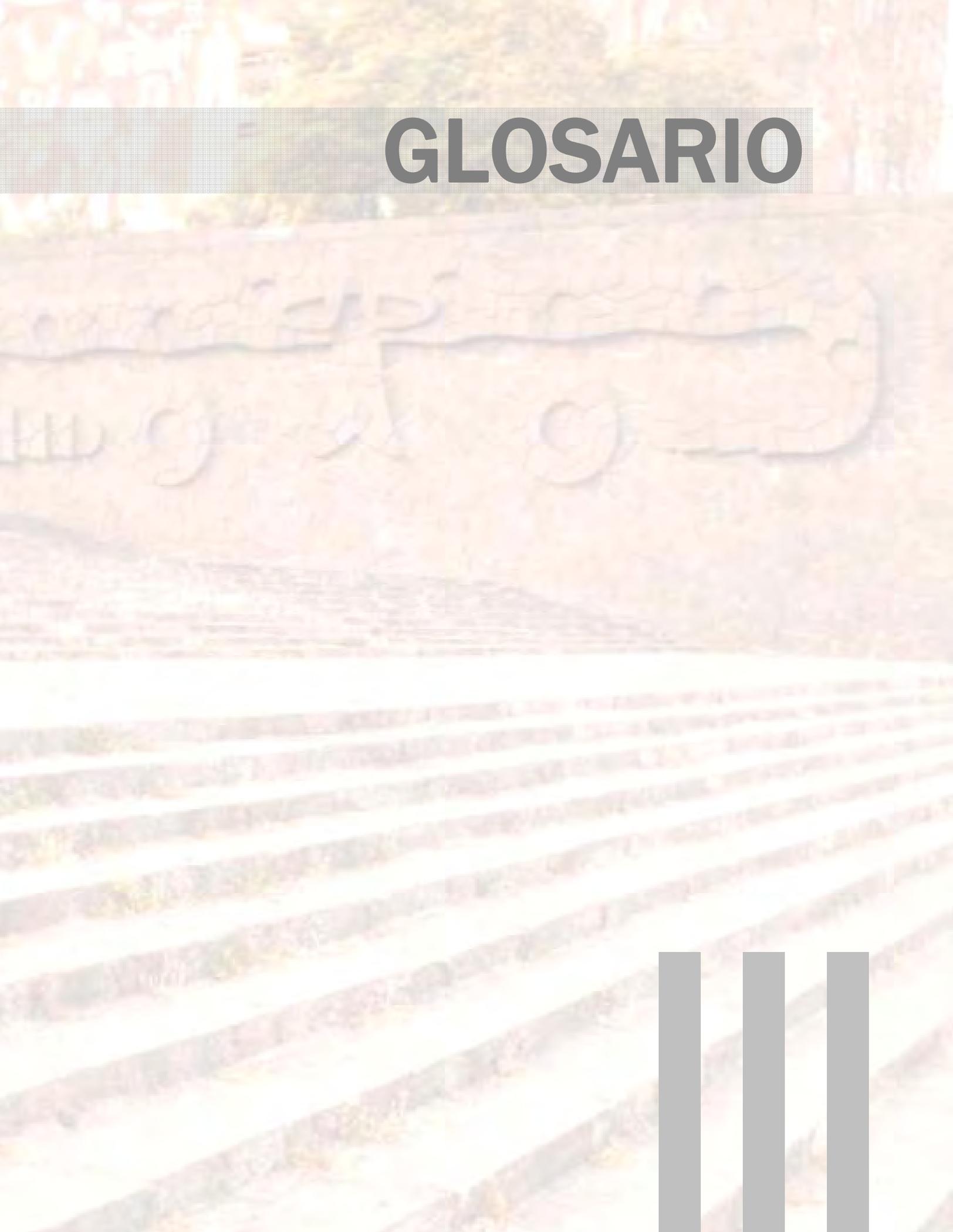
Por otro lado, como parte de la continuidad de este proyecto, se estima un periodo de cuatro meses de estabilización, que incluyen actividades tales como el monitoreo y control de los usuarios, la verificación de la actualización automática de los certificados, la cual es el pilar en el control de acceso al cluster, la instalación de más aplicaciones, la ejecución de pruebas con múltiples usuarios al mismo tiempo, por mencionar algunas. Además de explorar la posibilidad de exportar aplicaciones actuales de los usuarios de supercómputo de la DGSCA a esta infraestructura.

Aunado a lo anterior, la culminación de este proyecto, permitirá al Departamento de Supercómputo contar con un antecedente para llevar a cabo el análisis necesario sobre

la alternativa de migrar al equipo más poderoso en cuanto a cómputo se refiere de la UNAM, Kanbalam, u otros equipos, a la tecnología grid, para ser utilizados en un ambiente de producción, no de pruebas, lo que se traduce en la reducción de costos de la UNAM con motivo de la adquisición de equipos de cómputo.



GLOSARIO



- A**
- Ab initio:** La expresión latina *ab initio* significa *desde el principio*. Dentro del contexto de ciencias, especialmente física y química, se dice que un cálculo es "ab initio" (o "a partir de primeros principios") cuando sólo asume leyes básicas y bien establecidas, excluyendo por ejemplo tablas de parámetros externos o modelos simplificadores.
- Absoluta:** Ver **Ruta absoluta**
- Autoridad Certificadora (CA):** Son entes fiables y ampliamente reconocidos que firman las claves públicas de las personas, certificando con su propia firma la identidad del usuario. Por lo tanto, si se desea establecer una Autoridad Certificadora, éstas deben tomar extremadas precauciones para evitar que sus claves caigan en manos de intrusos, lo cual comprometería todo el sistema. También conocidos como notarios electrónicos.
- Abierto:** Ver **Código Abierto**
- B**
- Balance de cargas:** Es un aspecto central del cómputo paralelo y consiste en, dado un conjunto de tareas que comprenden un algoritmo y un conjunto de procesadores donde ejecutarlas, encontrar el mapeo de tareas a procesadores que resulte en que cada una tenga una cantidad de trabajo que demande aproximadamente el mismo tiempo.
- BIOS:** Por sus siglas en inglés Basic Input-Output System (Sistema de Entrada-Salida Básico) es un código de software que localiza y reconoce todos los dispositivos necesarios para cargar el sistema operativo en la RAM. Proporciona la comunicación de bajo nivel, el funcionamiento y configuración del hardware del sistema que, como mínimo, maneja el teclado y proporciona salida básica (emitiendo pitidos normalizados por el altavoz de la computadora si se producen fallos) durante el arranque.
- C**
- Cargas:** Ver **Balance de cargas**
- Cheminformatics:** Es relativamente un nuevo campo de la tecnología de la información que se centra en la colección, almacenaje, análisis y manipulación de datos químicos. Combina diversas áreas del conocimiento, principalmente la química y las ciencias de la computación.

Chkrootkit: Conjunto de utilerías que permiten verificar las modificaciones que hayan sufrido los binarios del sistema (ejecutables) y el borrado de registros en los archivos de bitácora.

Ciberinfraestructura: Se refiere a las tecnologías distribuidas de informática, información y comunicación combinadas con el personal y los componentes integrantes que proporcionan una plataforma a largo plazo para potenciar el esfuerzo científico de investigación contemporáneo.

Código Abierto: Ver **Open Source**

Colas: Ver **Sistema de colas**

Computadoras vectoriales: Una computadora vectorial es un equipo de cómputo que contiene un conjunto de unidades aritméticas especiales denominadas pipelines. Estas pipelines superponen la ejecución de las diferentes partes de una operación aritmética sobre los elementos del vector, produciendo una ejecución más eficiente de la operación aritmética que se está realizando.

D

DHCP: Por sus siglas en inglés Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host) Protocolo que permite a los administradores de red automatizar y gestionar de manera centralizada la asignación de direcciones del protocolo Internet (IP) en la red de una organización o de un proveedor de servicios de Internet (ISP). Permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Es un protocolo tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas se liberan, manteniendo un registro de la asignación de cada nodo y su tiempo de asignación. Sin DHCP, cada dirección IP debe configurarse manualmente en cada nodo y, si éste se desplaza a otra subred, se debe configurar una dirección IP diferente.

Dipolar: Se define como momento dipolar químico (μ) a la medida de la intensidad de la fuerza de atracción entre dos átomos, es la expresión de la asimetría de la carga eléctrica. Está definido como el producto entre la distancia d que separa a las cargas (longitud del enlace) y el valor de las cargas iguales y opuestas en un enlace químico: $\mu = q \cdot d$

Directorio scratch: Directorio temporal

DNS: Por sus siglas en inglés Domain Name System (Sistema de Nombres de Dominio), es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes. Es utilizada por computadoras, servicios o cualquier recurso conectado al internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir o resolver nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

E **E-science:** Denota el desarrollo de la ciencia que hace uso de recursos de cómputo intensivo, o bien, que utiliza una inmensa cantidad de datos que requieren el procesamiento en un entorno grid. Esta forma de trabajo permite la colaboración de muy diversas entidades en entornos de red altamente distribuida.

F **Frontend:** El concepto de frontend va de la mano con el de backend. Ambos términos se relacionan con el principio y fin de un proceso respectivamente. En lo que respecta al cluster, se refiere al nodo principal del sistema.

G **Gateway:** Dispositivo dedicado a intercomunicar sistemas con protocolos incompatibles. Se trata de un intermediario entre ambos para poder comunicarlos. Su propósito es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino. Es normalmente un equipo informático configurado para dotar a las máquinas de una red local (LAN) conectadas a él de un acceso hacia una red exterior, generalmente realizando para ello operaciones de traducción de direcciones IP (NAT: Network Address Translation). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada IP Masquerading (enmascaramiento de IP), usada muy a menudo para dar acceso a Internet a los equipos de una red de área local compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa.

Gradiente: El gradiente normalmente denota una dirección en el espacio según la cual se aprecia una variación de una determinada propiedad o magnitud física. En otros contextos se usa informalmente gradiente, para indicar la existencia de gradualidad o variación gradual en determinado aspecto, no necesariamente

relacionado con la distribución física de una determinada magnitud o propiedad.

Grid-enabled: Una aplicación grid-enabled tiene un punto de contacto con la grid, normalmente a través de una API que permite al programador enviar fragmentos de código y de datos para que se ejecuten en algún lugar desconocido a priori, gestionado por la grid. Para que esto tenga alguna utilidad, el programador debe seleccionar el código que puede ejecutarse en porciones simultáneamente y promediar o consolidar después los resultados.

H

Hash: Función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, u otro. Una función de hash permite resumir o identificar probabilísticamente un gran conjunto de información, dando como resultado un conjunto imagen finito generalmente menor. Una propiedad fundamental del hashing es que si dos resultados de una misma función son diferentes, entonces las dos entradas que generaron dichos resultados también lo son. Un hash es el resultado de dicha función o algoritmo.

Hessiana: El hessiano, conocido también como discriminante o matriz hessiana, fue introducido en el año de 1844 por Hesse, matemático alemán. La matriz hessiana de una función f de n variables, es la matriz cuadrada de $n \times n$, de las segundas derivadas parciales.

HIPPI: Por sus siglas en inglés HI-performance Parallel Port Interface (Interfaz Paralela de Alto Rendimiento), es una interfaz que se utiliza para conectar en red supercomputadoras, estaciones de trabajo high-end y periféricos que usan conmutadores de circuito de barras cruzadas. Ofrece tasas de transferencia de 800 Mbps sobre 32 cables de cobre de par trenzado (única interfaz HIPPI) y 1.600 Mbps sobre 64 pares (doble HIPPI).

Hostname: Programa que se utiliza para mostrar o establecer el nombre actual del sistema (nombre de equipo). Muchos de los programas de trabajo en red usan este nombre para identificar a la máquina.

I

Initio: Ver **Ab initio**.

IP: Protocolo para la comunicación en una red a través de paquetes conmutados, es principalmente usado en Internet. También puede referirse a un número que identifica de manera lógica y jerárquica a

una interfaz de un dispositivo, que puede ser computadora, una impresora, etc., dentro de una red que utilice el protocolo IP.

J **Job:** Es un proceso listo para su ejecución. Este concepto es utilizado en la ejecución de procesos en batch; cada proceso que se encuentra en la cola de trabajos es un job. Este esquema es utilizado en ambientes de cómputo paralelo, como los clusters, donde cada job se encuentra listo y en espera ya que demanda la utilización de recursos para poder ser ejecutado.

K **Kernel:** Base del sistema operativo que permite a este interactuar el software con el hardware. Sus funciones más importantes son administrar la memoria para todos los programas y procesos en ejecución, administrar el tiempo de procesador que los programas y procesos en ejecución utilizan y permite acceder a los periféricos/elementos del sistema de forma transparente para el usuario.

L **Latencia:** Es el lapso de tiempo que transcurre durante la transferencia de un paquete de información de un lugar a otro. En redes informáticas se denomina latencia a la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

Limnólogos: La limnología es la rama de la ecología que estudia los ecosistemas acuáticos continentales (lagos, lagunas, ríos, etc.), las interacciones entre los organismos acuáticos y su ambiente, asimismo, determina la distribución y abundancia en dichos ecosistemas.

Logfiles: Archivos de bitácora del sistema. Permite mantener un registro de los eventos que ocurren en el sistema, generalmente relacionados con el acceso, la ejecución de procesos, la modificación de archivos de configuración del sistema.

LSF: Por sus siglas en inglés Load Share Facility (Facilitador de compartición de carga) Es un gestor de colas que permite analizar la carga y repartir el trabajo entre los procesadores.

M **Mac Address:** Abreviación del idioma inglés (Media Access Control Address) que en español significa dirección de control de acceso al medio , es un número único asignado a cada tarjeta de red de un dispositivo. En cuanto identifica a dispositivos de red, es también conocida como la dirección física.

Magnetohidrodinámicos: Ciencia que se ocupa del movimiento de los fluidos conductores de electricidad (líquidos y gases).

MD5: Abreviatura de Message-Digest Algorithm 5, que significa Algoritmo de Resumen del Mensaje 5, es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. Surgió como un reemplazo de MD4 cuando fue descubierta su debilidad. La utilidad que tienen los resúmenes MD5 es principalmente en la descarga de archivos. Esto se hace comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, con lo que no detecta solamente los archivos alterados, sino también reconoce una descarga corrupta o incompleta. Para ello puede utilizarse una herramienta MD5 en los sistemas UNIX, el comando de `md5sum` es un ejemplo de tal herramienta.

Mecanocuánticos: El modelo mecano-cuántico es la explicación actual sobre el comportamiento del átomo. Esta se basa en la teoría cuántica, que es la teoría física basada en la utilización del concepto de unidad cuántica para describir las propiedades dinámicas de las partículas subatómicas y las interacciones entre la materia y la radiación.

Memoria: Ver **Palabra de memoria**

N

NIC: Por sus siglas en inglés Network Interface Card (Tarjeta de Interfaz de Red) Se encarga de servir como interfaz de Ethernet entre el medio físico (por ejemplo, un cable coaxial) y el equipo (por ejemplo una computadora o una impresora). Es un chip usado en computadoras o periféricos tales como las tarjetas de red, impresoras de red o sistemas embebidos para conectar dos o más dispositivos entre sí a través de algún medio, ya sea conexión inalámbrica, cable UTP, cable coaxial, fibra óptica, u otro.

Números de gran precisión: Se refiere a la utilización de tipos de datos que permiten el procesamiento de datos numéricos enteros o decimales muy grandes.

O

Open source: En español significa: código abierto. Es el término con el que se conoce al software distribuido y desarrollado libremente.

P

Palabra de memoria: Es una entidad de “n” cantidad de bits que es manejada por los dispositivos de entrada/salida de un sistema y es tratada como una sola unidad.

PBS: Por sus siglas en inglés Portable Batch System (Sistema de Colas Portable) Sistema que proporciona una serie de herramientas para la gestión de trabajos batch, utilizando una unidad de programación de tareas. Además, permite el enrutamiento de estos trabajos a través de diferentes computadores. Cuenta con capacidades para definir e implementar políticas sobre la utilización de los recursos disponibles.

PKI: Por sus siglas en inglés Public Key Infrastructure (Infraestructura de Clave Pública) Forma común de referirse a un sistema complejo necesario para la gestión de certificados digitales y aplicaciones de la Firma Digital.

Polarizabilidad: La polarización electrónica es un desplazamiento de las cargas en presencia de un campo eléctrico externo, es decir en un átomo neutro la nube electrónica se reorienta de tal manera que el átomo se distorsiona ligeramente y pierde su simetría. La dificultad al analizar estos fenómenos varía en el tratamiento de la interacción de muchos cuerpos.

Precisión: Ver **Números de gran precisión**

Proteoma: Conjunto de proteínas que se están expresando en un momento dado, en una célula, tejido u organismo. El proteoma puede considerarse como dinámico si se compara con el genoma, pues varía con el tiempo y/o con diferentes estados fisiológicos o patológicos específicos)

Q

R

Rsh: Es una aplicación utilizada en conexiones remotas. Está basada en el protocolo del rlogin, mismo que interactúa con el demonio rlogind, el cual a su vez es capaz de lanzar un shell para ejecutar los comandos de forma remota. Un inconveniente en su utilización es que la información es transmitida sin cifrar, lo que ha dado lugar al surgimiento de otras aplicaciones que corrigen este problema, como lo es SSH (Secure Shell).

Ruta absoluta: Las rutas absolutas señalan la ubicación de un archivo o directorio desde el directorio raíz del sistema de archivos.

S

SCE: Conjunto de herramientas de software abierto que permiten a los usuarios utilizar de forma efectiva un cluster de tipo Beowulf.

SCMS: Herramienta de administración y monitoreo del cluster o grid.

SCMSWEB: Herramienta que permite monitorear el rendimiento de los recursos en forma gráfica.

Scratch: Ver **Directorio scratch**

SGE: Es un gestor de colas, desarrollado por Sun Microsystems. Entre sus características principales destaca la versatilidad para trabajar con entornos paralelos, así como monitorizar el estado de las tareas que están ejecutándose. Posee una gran cantidad de opciones para facilitar la labor en lanzamiento de trabajos (jobs), observar su estado, eliminarlos, etc. SGE se compone por 3 elementos principales en su infraestructura: Submit Host (máquina desde la cual se envía el trabajo para su ejecución), Master Host (nodo encargado de la gestión de la cola, distribución de los trabajos en ella, recepción de los datos procesados para su envío al usuario) y Execution Host (nodo en el que se ejecuta el trabajo, dependiente del master host).

Shell: Programa informático mediante el cual los usuarios pueden tipear comandos. Ofrece una interfaz a fin de que los usuarios controlen el funcionamiento de la computadora introduciendo la instrucción en forma de texto y el intérprete de líneas de comandos lo ejecute.

Sistema de colas: Aplicación que permite a los usuarios ejecutar, finalizar, modificar y controlar los trabajos en modo no interactivo. Permite ejecutar de manera más eficiente, natural y económica los procesos en el sistema. El usuario tiene la posibilidad de continuar trabajando de forma interactiva en su terminal.

Solvatación: Es el proceso de atracción y asociación de moléculas de un disolvente con moléculas o iones de un soluto. Al disolverse los iones en un solvente, se dispersan y son rodeados por moléculas de solvente. A mayor tamaño del ion, más moléculas de solvente son capaces de rodearlo, y más solvatado se encuentra el ion.

Source: Ver **Open source**

SSH: Nombre de un protocolo y del programa que lo implementa. Permite acceder a máquinas remotas a través de una red y manejarla por completo la computadora mediante un intérprete de

comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si se cuenta con un Servidor de ambiente gráfico. Permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH. Trabaja de forma similar a como se hace con telnet. La diferencia principal es que SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión.

SSL: Por sus siglas en inglés Secure Socket Layer (Capa de Conexión Segura) es un protocolo de red utilizado para establecer comunicaciones seguras entre un Servidor y un Cliente. Proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

T

Tar: Utilidad de UNIX que permite empaquetar un conjunto de archivos en uno solo, la extensión con la que se identifican estos archivos es .tar.

TLS: Por sus siglas en inglés Transport Layer Security (Seguridad de la Capa de Transporte) es un protocolo criptográfico que proporciona comunicaciones seguras por una red, comúnmente Internet. Muy similar a SSL.

Tripwire: Software de Open Source, que permite verificar la integridad de los datos y la seguridad del sistema, utilizada para el monitoreo y alerta sobre los cambios en los archivos.

U**V**

Vectoriales: Ver **Computadoras vectoriales**

VO: Organización Virtual

VOMS: Por sus siglas en inglés, Servicio de Administración de Organizaciones Virtuales (Virtual Organization Management Service), la cual organiza las listas y roles de los miembros de una Organización Virtual (VO). Por ejemplo, un grupo de veinte científicos de tres universidades diferentes pueden colaborar con cada uno para un experimento específico. Con el fin de mantener organizadas sus actividades hacen uso de VOMS.

W

X

X.509: Estándar UIT-T (Unión Internacional de Telecomunicaciones) para infraestructuras de claves públicas. Especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

Y

Z



Orgullosamente UNAM