



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

CONSTRUCCION DE UNA GRID DE CLUSTERS

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

PRESENTA
TANIA CRISTINA ROSAS CASTILLO

DIRECTORA DE TESIS:
ING. LAURA SANDOVAL MONTAÑO



CIUDAD UNIVERSITARIA

MEXICO, D. F., 2006



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

A Dios:

Porque cada mañana me motiva a seguir adelante y me llena de esperanza en cada anochecer.

Porque siempre está junto a mí y con él nada es imposible.

A mi mamá:

Por su amor, consejos, disciplina, apoyo desinteresado e incondicional y porque siempre me guió por un buen sendero.

Por el apoyo recibido durante mi carrera, la confianza brindada aún en momentos difíciles y en especial por su cariño para el cual no existen palabras que expresen lo que ha significado en el transcurso de mi vida.

Pero sobre todo por enseñarme que es admirable luchar hasta el cansancio, pero que es aún más admirable estar cansado y seguir luchando.

Por eso y muchas cosas más le doy las gracias.

A mi papá y hermanas:

Por los gratos momentos de diversión, por su cariño, comprensión y apoyo.

Y porque también de ellos he aprendido.

Gracias por todo.

A mi madrina:

Porque es la muestra viviente de que en el mundo existen personas que brindan su tiempo y esfuerzo para ayudar a otros sin esperar nada a cambio.

Mi más profundo respeto y eterno agradecimiento por todo el cariño y apoyo brindados.

A mi tía Bety:

Por todo el aprecio, cariño y confianza que me ha dado.

Porque siempre ha creído en mí y puedo contar con su apoyo incondicional.

A mi tío Salvador:

Por su valiosa contribución y apoyo para la finalización de la presente tesis.

A mi padrino:

Por sus consejos, cariño, confianza y apoyo incondicional en todo momento.

A Erick:

Por el cariño, apoyo, estímulo y compañerismo que me brindó a lo largo de mi carrera.

A mis familiares y amigos:

Por su amistad, cariño y por permitirme entrar en sus vidas para compartir nuestros éxitos y fracasos.

A la Ing. Laura:

Por haber dirigido la presente tesis, por su ayuda, tiempo, conocimientos y comprensión.

A los profesores:

Por su amplio conocimiento, experiencia y consejos técnicos que me ayudaron en mi formación profesional y que me dieron las herramientas suficientes para desempeñarme bien en cualquier situación.

A la Universidad:

Por su dedicación en la formación de profesionales de gran nivel.

Porque no sólo me enseñó mi profesión, sino que también me dio valiosas lecciones sobre lo que he de encontrarme y superar en la vida.

Y a todas aquellas personas que contribuyeron directa o indirectamente en la realización de la presente tesis.

Una vez más y por siempre GRACIAS a todos.

Tania Cristina Rosas Castillo

Índice General

Índice de figuras	v
Índice de tablas	vi
Agradecimientos	vii
Introducción	1
1. Capítulo I. Antecedentes	3
1.1 Cómputo de alto desempeño	5
1.1.1 Paralelismo	5
1.1.2 FLOPS	7
1.1.3 Programación concurrente	7
1.1.4 Sistemas distribuidos	8
1.2 Equipo de cómputo de alto desempeño	16
1.2.1 Arquitecturas paralelas	16
1.2.2 Clusters	20
1.3 Surgimiento de la GRID	25
2. Capítulo II. Las GRID	29
2.1 Concepto	31
2.2 Computación GRID	33
2.3 GRID y cluster	34
2.4 Fundamentos	34
2.5 Software de soporte	35
2.6 Gestión de recursos	43
2.7 Aplicaciones de la GRID	50
2.7.1 Introducción	50
2.7.2 La evolución de los recursos locales de computación	52
2.7.3 Computación a través de la red	52
2.7.4 Aplicaciones generales de la GRID	53
3. Capítulo III. Proyectos de la GRID	63
3.1 Proyectos GRID en Estados Unidos	65
3.1.1 GADU/GNARE. Usos de TeraGRID para análisis de secuencias de proteínas	65
3.1.2 Condor	67
3.1.3 Entropia	68
3.1.4 La ciencia de SETI@home	73
3.2 Los proyectos GRID europeos	75
3.2.1 GÉANT	76
3.2.2 DataGRID	78
3.2.3 CrossGRID	81
3.2.4 EuroGRID	84
3.2.5. DAMIEN	88
3.3 Construcción de una GRID Interinstitucional en México	90

4. Capítulo IV. Implementación de la GRID	95
4.1 Ciclo 1. Construcción de clusters	97
4.1.1 Diseño	97
4.1.2 Requerimientos mínimos	98
4.1.3 Implementación	99
4.1.4 Configuración del servidor y nodos	99
4.1.5 Configuración inicial	91
4.1.6 Configuración del servidor NFS	101
4.1.7 Configuración del servidor NIS	102
4.1.8 Instalación de PVM (Parallel Virtual Machine)	104
4.1.9 Instalación de MPICH (MPI/Chameleon)	105
4.2 Ciclo 2. Construcción de una GRID usando Globus Toolkit 4.0.1 (GT4)	106
4.2.1 Instalación del Globus Toolkit	107
4.2.2 Configuración e instalación del GT4	111
4.2.3 Implementando la seguridad	112
4.2.4 Configurando GridFTP y Globus Gatekeeper	116
4.2.5 Enviando trabajos a la GRID	119
 Conclusión	 123
 Bibliografía y Mesografía	 125

Índice de figuras:

Figura 1.1 Disposición de los componentes de un sistema distribuido

Figura 1.2 Clasificación de los Sistemas de Cómputo de Flynn

Figura 1.3 Arquitectura SISD

Figura 1.4 Arquitectura SIMD

Figura 1.5 Arquitectura MISD

Figura 1.6 Arquitectura MIMD

Figura 1.7 Cluster

Figura 2.1 Interacción en la GRID

Figura 2.2 Modelo OGSA de la GRID

Figura 2.3 Arquitectura en capas de Globus

Figura 2.4 Globus

Figura 2.5 Modelo en capas

Figura 2.6 Fase 1: Descubrimiento de Recursos

Figura 2.7 Fase 2: Selección del Sistema

Figura 2.8 Fase 3: Ejecución del Trabajo

Figura 3.1 Esquema del proyecto TeraGRID

Figura 3.2 Mapa del proyecto NSC (National Science Foundation) TERAGRID

Figura 3.3.1 SETI@home versión 4.18

Figura 3.3.2 SETI@home de un cliente clásico (versión 3.08)

Figura 3.4. Mapa del *testbed* del proyecto europeo CrossGRID

Figura 3.5. Mapa del proyecto GÉANT

Figura 3.6. Ejemplo de envío de un trabajo, mostrando los elementos básicos de la GRID

Figura 3.7 Organización de los paquetes de trabajo técnicos en el proyecto de DataGrid

Figura 3.8 Tecnología GRID en apoyo a inundaciones

Figura 3.9 Bio GRID

Figura 3.10 Meteo GRID

Figura 3.11 Arquitectura DAMIEN

Figura 3.12 Diagrama de conectividad de GRAMA

Índice de tablas

Tabla 3.1 Proyectos GRID en el mundo

Tabla 4.1 Lista de requerimientos para el GT4

Tabla 4.2. Especificación RLS

Introducción

Una GRID computacional es una infraestructura que suministra, al que la utiliza, acceso seguro, consistente, extensivo y económico, a unas elevadas capacidades de cómputo. Su nombre surgió de *Global Resources Information Database*.

El concepto infraestructura se utiliza porque una GRID es un conjunto de recursos, los cuales necesitan una interconexión de hardware y un control de software para que estén ensambladas en una GRID. Esta infraestructura debe proporcionar a los usuarios un servicio **seguro** a todos los niveles: capacidad de cómputo, de integridad de datos, de seguridad de acceso, etc. El servicio debe ser **consistente**, basado en estándares, y de esta manera el acceso y las operaciones sobre la GRID estarán definidas por dichos estándares evitando la heterogeneidad. La idea de **extensión** se refiere a la posibilidad de que la GRID llegue a cualquier sitio, de esta manera uno se asegura que una vez conectado desde cualquier punto, se puede extraer de la GRID toda la potencia de cómputo que se requiera. Por último el acceso y uso de la GRID debe tener un costo **económico** que lo haga atractivo para que su uso se haga universal.

El concepto de cómputo GRID da la idea de una gran potencia de cálculo y almacenamiento, y parece un gran avance en las ciencias de la computación. El cómputo GRID es actualmente una de las áreas en las que más activamente se está investigando y a la que se han sumado las grandes empresas del desarrollo informático como Microsoft, Sun, Oracle, IBM y HP. Además existen multitud de aplicaciones reales que hacen uso de mini-GRIDs, casi todas ellas están centradas en el campo de la investigación en el terreno de las ciencias físicas, médicas y del tratamiento de la información.

El cómputo GRID es bastante reciente. Sin embargo, parece que no es simplemente una “moda”, ya que está siendo activamente investigada y desarrollada en todo el mundo. Plantea interesantes desafíos en una gran cantidad de áreas de investigación distintas.

La computación GRID es una de las tecnologías más prometedoras actualmente. Ahora, las aplicaciones de la computación GRID están muy orientadas al ámbito científico. Por esto, gracias a la computación GRID ya se está realizando investigación de mayor calidad.

Sin embargo, todavía no existe “*La GRID*”, hay muchas “mini-GRID” a lo largo de todo el mundo (*testbeds*) que se están utilizando para poner a prueba tecnologías GRID.

Lo que se espera es que en un futuro exista una GRID a nivel mundial al que todos podamos tener acceso.

Por ello el objetivo más importante fue el construir una GRID experimental de pequeña escala, y sentar las bases para el desarrollo futuro de una infraestructura de GRID a un nivel mayor, que sirva como un recurso de cómputo de gran escala para la investigación científica.

La construcción de la GRID permite a un mismo tiempo conocer las dificultades técnicas para la implementación de las GRID computacionales; los mecanismos que deben utilizarse para tal fin; algunos aspectos a considerar en términos organizacionales; y el estado actual de estas tecnologías.

En cuanto a los aspectos técnicos y estado actual, se sabe que el software y la documentación necesarios para implementar los servicios básicos de una GRID ya están suficientemente estandarizados, y se encuentran al alcance de cualquier centro de cómputo con recursos humanos experimentados en administración de servicios; sin embargo, los servicios más avanzados, así como las aplicaciones que pueden aprovechar la GRID, aún se encuentran en fase de investigación, experimentación y/o desarrollo.

La dirección a tomar es presentar, en la medida de lo posible, ejemplos de trabajos de investigación que pueden aprovechar la actual infraestructura de GRID, y demostrando que una conectividad más apropiada redundaría en una infraestructura más poderosa y robusta, y por lo tanto en una mayor y mejor utilización.

En cuanto a aspectos de organización, los temas que cubren las tecnologías de GRID son de una gran variedad, por lo que es importante establecer grupos de trabajo independientes, que puedan interactuar con fluidez para resolver los problemas concretos en temas como seguridad, redes, *middleware* y aplicaciones. Es necesario destacar el hecho de que la instalación, configuración y operación de las GRID, la utilización y/o el desarrollo de aplicaciones, etc., requieren de recursos humanos adecuados y suficientes. Debe resaltarse la importancia de interactuar con otros proyectos GRID a nivel internacional, pues esto trae la ventaja de asimilar otras experiencias, lo que acelera el desarrollo de la implementación de estas nuevas tecnologías, además de establecer lazos que puedan resultar en la constitución de proyectos de GRID de mayor envergadura.

El objetivo primordial de esta tesis, es la construcción de una GRID que sirva como infraestructura de pruebas para el desarrollo de tecnologías de GRID.

Una GRID es una infraestructura de hardware, software y comunicaciones que permite la integración de equipos de cómputo localizados en diferentes sitios geográficos, y que pueden ser utilizados de manera compartida y transparente por una comunidad específica. La infraestructura de hardware puede ser constituida a partir de equipos de cómputo existentes; la infraestructura de software se constituye a partir de los estándares actuales de construcción de GRID y de acuerdo a necesidades generales de la comunidad y la infraestructura de comunicaciones puede ser la red local a la que se encuentren conectados los equipos de cómputo.

Esta tesis pretende ser la base para la construcción de una GRID en el Laboratorio de Telemática, la cual posteriormente pueda formar parte de la GRID Interinstitucional de México.

1. Antecedentes

1.1 Cómputo de alto desempeño

El término de supercomputadora fue reemplazado por el término: Computadora de Alto Desempeño o Ambiente de Cómputo de Alto Desempeño, ya que estos últimos reconocen que el alto desempeño tiene que ver con todo el equipo de cómputo integrado y no sólo con una unidad central de proceso.

El cómputo de alto desempeño se forma por dos tipos de sistemas:

Sistemas distribuidos: están diseñados para que muchos usuarios trabajen en forma conjunta.

Sistemas paralelos: están diseñados para lograr la máxima rapidez en un único problema.

1.1.1 Paralelismo

Para entender la computación en paralelo es necesario tener en cuenta el multiproceso en diferentes hilos de ejecución (*multithreading*), la escalabilidad o capacidad de ampliar la potencia de computación y las técnicas de *clustering* o interconexión de varias computadoras formando una única supercomputadora.

➤ Multiproceso

El rendimiento de una aplicación no sólo se basa en la velocidad. En el entorno de un servidor Web un buen rendimiento también implica que el máximo número de usuarios pueda ser servidor concurrentemente.

➤ Escalabilidad

La escalabilidad es la habilidad de utilizar eficientemente la capacidad adicional de proceso de un sistema multiprocesador (2, 4, 8, 32, o más procesadores). Escalabilidad también se refiere a la capacidad de servir a más clientes en un entorno de servidores.

➤ Clustering

Clustering hace referencia al enlace de servidores individuales física y lógicamente y a la coordinación de la comunicación entre ellos, de modo que puedan realizar tareas comunes.

Definición de paralelismo:

El concepto de paralelismo supone la introducción de varios procesadores para resolver un problema. Un procesador diez veces más potente que un procesador de potencia normal es mucho más caro que diez procesadores de potencia normal. Por ello, si se paraleliza un programa, es decir, se divide la carga computacional entre varios procesadores distintos, se obtiene una mejora en la relación entre costo y rendimiento. Con menos inversión en hardware se obtiene mucha más potencia computacional.

Sin embargo, existen varios factores tanto de índole computacional como de índole puramente físico, las comunicaciones entre procesadores son lentas en comparación con la velocidad de transmisión de datos y de cómputo dentro de un procesador, lo que hace que nunca se obtenga un escalado en el rendimiento igual o superior que el escalado en el número de procesadores, es decir, diez procesadores no van a ir diez veces más rápido que un solo procesador. Sin embargo, el incremento de velocidad se aproxima al número de procesadores que se integran en el sistema en determinados algoritmos con una paralelización intrínseca muy fuerte. Además, hay gran cantidad de científicos trabajando para acercar el límite práctico al límite teórico, por lo que, a los avances en hardware, se suman los avances que se producen en algoritmia. A todo esto se añade que el factor costo sigue siendo determinante para optar por soluciones paralelas aunque no sean óptimas, ya que, aunque no se obtenga el rendimiento máximo teórico, un incremento lineal en la potencia del procesador de una máquina implica un crecimiento exponencial del precio de una máquina, mientras que, con el incremento del número de procesadores, el incremento del precio es mucho menor.

Los sistemas paralelos son bastante complejos de programar, y su verificación es más compleja todavía. Por ello, estos sistemas sólo encuentran su justificación en aquellos casos en los que el incremento de costos en la programación realmente está justificado porque esa potencia de cálculo es precisa. Para estas aplicaciones, sin embargo, los sistemas paralelos tienen gran éxito, encontrando como únicas barreras para su imposición final la complejidad del desarrollo (se necesita de programadores realmente preparados, y que conozcan a fondo materias tan complejas como mecanismos de sincronización y compartición de datos) y la escasez de herramientas para hacer más cómoda la verificación de programas paralelos.

Aplicaciones del paralelismo

Algunos ejemplos de aplicaciones en las que el paralelismo es de gran utilidad son:

➤ *Los grandes sistemas gestores de bases de datos*, operaciones de búsqueda y proceso de datos a gran escala, la denominada minería de datos; este problema tiene grandes implicaciones tanto en ciencia, proyecto genoma humano, como en análisis de datos estadísticos para problemas de bolsa y de análisis de mercados.

➤ *Algoritmos de cálculo numérico pesados*. El cálculo se realiza más fácilmente con las mejoras algorítmicas en el proceso de diagonalización. Este es un área fruto de gran investigación, y existen gran cantidad de rutinas ya listas para ser empleadas.

➤ *Los gráficos por computadora a gran escala*. Aunque este problema realmente es una subespecialización de los algoritmos de cálculo numérico pesados, problemas como generación de imágenes realistas o efectos especiales son muy pesados en cálculos.

➤ *Problemas de simulación*. Estos problemas son de gran interés para la industria, ya que ahorran gran cantidad de dinero en experimentos de campo que no necesitan ser realizados por poder ser analizados en etapas de desarrollo del producto mediante complejas simulaciones por computadora.

➤ *Predicción de fenómenos naturales*, como una especialización del campo anterior de gran interés en la actualidad. Tanto fenómenos sísmicos, como grandes fenómenos naturales (inundaciones, tornados, sequías) son centro de estudio en la actualidad, y una parte muy importante de las supercomputadoras con procesamiento paralelo es empleada en la actualidad parcial o totalmente para este tipo de tareas. De hecho, *clusters* de tipo *Beowulf* original fueron desarrolladas para realizar complejas simulaciones atmosféricas.

➤ *Problemas de optimización.* Si bien algunos métodos de optimización son bastante complejos de paralelizar, los problemas de optimización son lo suficientemente pesados computacionalmente como para que sea de interés su paralelización. Algunos de estos problemas son el diseño de estructuras y el de fármacos, problemas de gran interés en la ciencia actual.

1.1.2 Flops

Flop: FLoating-Point Operation, Operación de punto flotante por segundo. Es una unidad de medida utilizada para calcular la potencia y velocidad de las supercomputadoras.

El flop mide la tasa por segundo de las operaciones aritméticas de alta precisión que puede realizar el procesador o el sistema de una computadora. Se utiliza para medir el rendimiento del microprocesador.

1.1.3 Programación concurrente

La programación concurrente es la rama de la informática que trata de las notaciones y técnicas de programación que se usan para expresar el paralelismo potencial entre tareas y para resolver los problemas de comunicación y sincronización entre procesos.

En la programación concurrente se supone que hay un procesador utilizable por cada tarea; no se hace ninguna suposición de si el procesador será una unidad independiente para cada uno de ellos o si será una sola CPU que se comparte en el tiempo entre las tareas.

Independientemente de cómo se vaya a ejecutar realmente el programa, en un sistema uniprocador o multiprocador, el resultado debe ser el correcto. Por ello, se supone que existe un conjunto de instrucciones primitivas que son o bien parte del sistema operativo, o bien parte de un lenguaje de programación, y cuya correcta implementación y corrección está garantizada por el sistema.

Se llama "hilo" a un flujo de programa que se ejecuta a la vez que otros. En un programa "multihilo", es decir que tiene varios hilos, hay varios flujos ejecutándose a la vez.

El concepto de simultaneidad se ha implementado de distintas formas. Hay tres variantes principales. Cuando se habla de "procesos" cada flujo de ejecución tiene una copia particular de la memoria del programa. Si dos procesos tienen que comunicarse, han de usar mecanismos de IPC (Comunicación Entre Procesos = "*Inter Process Communication*") como sockets, tuberías ("pipes") o archivos proyectados en memoria ("*memory mapped files*"). Es el mismo tipo de comunicación que usan dos programas distintos.

Cuando se habla de "hilos" el aislamiento es menor. Hay una sola copia del programa en memoria (aunque cada hilo tiene su pila) y todos los hilos pueden usar la memoria global como propia. En los programas con varios hilos hay que tener cuidado para evitar el acceso simultáneo de dos hilos a la misma zona de memoria, lo que podría provocar corrupción de datos. Se usan "semáforos" o "secciones críticas" para "proteger" un recurso. Esto es asegurar que sólo un hilo a la vez está en posesión del recurso protegido".

Hay una tercera forma llamada "corrutina". Para ser exactos no se puede decir que haya simultaneidad, más bien intercalación. Hay varios flujos de programa, pero sólo uno de ellos

está activo a la vez. El flujo activo completa alguna tarea y él mismo cede el control saltando a otra corrutina que sigue ejecutándose donde lo dejó la última vez. En esto se diferencia de hilos y procesos, en los cuales los flujos de ejecución son detenidos externamente por mecanismos del sistema operativo.

Las corrutinas comparten toda la memoria excepto la pila, pero no son realmente simultáneas. No es necesario usar semáforos ni secciones críticas. Simplemente el código debe ejecutar de un tirón ciertas tareas que no se pueden quedar a medias. En la práctica son mucho más seguras que los hilos.

Cada uno de estos tres tipos de simultaneidad es mejor para un tipo de aplicación. Si el programa no necesita apenas comunicación entre los diversos flujos, lo más sencillo es crear varios procesos, o sea ejecutar varias copias del programa simultáneas e independientes.

La ventaja de los hilos es el acceso más cómodo a la información compartida. Tan cómodo como acceder a una variable global. Pero también es más inseguro. La programación concurrente es difícil. Es importante mantener el esquema sencillo, porque a la mínima complicación se pueden producir bloqueos mutuos y otras sorpresas.

Las corrutinas son malas para programar "demonios" (programas en segundo plano), que pueden tener que atender peticiones de otros programas en paralelo. Por lo demás son preferibles a los procesos por velocidad y a los hilos por seguridad. En realidad hay pocos lenguajes que las implementan. Lenguajes tan populares como C++, Java y Delphi no las tienen.

1.1.4 Sistemas distribuidos

Introducción a los Sistemas Distribuidos

Desde el inicio de la era de la computadora moderna (1945), hasta cerca de 1985, solo se conocía la computación centralizada. A partir de la mitad de la década de los ochentas aparecen dos avances tecnológicos fundamentales:

1. El desarrollo de **microprocesadores** poderosos y económicos con arquitecturas de 8, 16, 32 y 64 bits. Los cuales permitieron reducir en tamaño y costo a las computadoras y aumentar en gran medida las capacidades de los mismos y su acceso a más personas.

2. El desarrollo de las **redes de área local (LAN)** y de las comunicaciones que permitieron conectar cientos de computadoras con posibilidad de transferencia de datos a alta velocidad, de millones de bits por segundo (Mbps).

Aparecen los **sistemas distribuidos**, en contraste con los *sistemas centralizados*.

Los sistemas distribuidos necesitan un software distinto al de los sistemas centralizados.

Los sistemas operativos para sistemas distribuidos han tenido importantes desarrollos pero todavía existe un largo camino por recorrer. Los usuarios pueden acceder a una *gran variedad de recursos computacionales*, de hardware y software, distribuidos entre un gran número de sistemas computacionales conectados.

Definición de Sistemas Distribuidos

Un sistema distribuido es aquél que está compuesto por varias computadoras o procesadores autónomos, que se encuentran físicamente separados, pero conectados mediante una red de comunicación y no comparten la memoria principal. Están equipados con programas que les permiten coordinar sus actividades y compartir recursos, se comunican y coordinan sus acciones entre sí a través del intercambio de mensajes utilizando un medio de comunicación, para el logro de un objetivo. Se establece la comunicación mediante un protocolo prefijado por un esquema cliente-servidor. Los sistemas autónomos pueden tener características no homogéneas.

Arquitectura de software

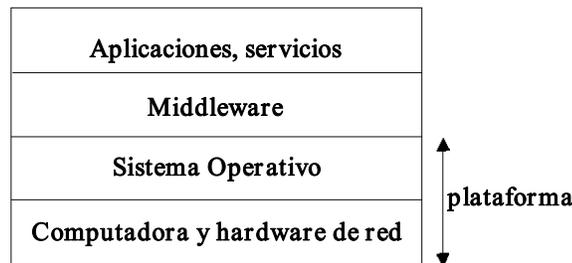


Figura 1.1 Disposición de los componentes de un sistema distribuido

El término arquitectura de software se refería inicialmente a la estructura del software como capas en una única computadora. Recientemente las capas son uno o varios procesos, localizados en la misma o en diferentes computadoras, que ofrecen y solicitan servicios.

En esta arquitectura las capas más bajas proporcionan servicio a las superiores y su implementación es dependiente de cada computadora.

Según estos niveles son como se encuentran los componentes de software, en la capa más baja se encuentra el hardware de red y la computadora, en la siguiente capa se encuentra el sistema operativo, posteriormente se encuentra el *middleware*, y las aplicaciones y servicios.

El *middleware* es el nivel encargado de enmascarar la heterogeneidad del sistema distribuido. Además proporciona un modelo de programación a los programadores de aplicaciones. Y soporta abstracciones como:

- Llamadas a procedimientos remotos o invocación remota
- Comunicación entre grupos de procesos.
- Notificación de eventos.
- Replicación de datos.
- Servicios de nombres.

Propiedades

- **Existencia de varias computadoras**

Los sistemas distribuidos están compuestos por *varios recursos informáticos* de propósito general, tanto físicos como lógicos, que pueden asignarse dinámicamente a tareas concretas. Estos recursos están *distribuidos físicamente*, y funcionan gracias a una red de comunicaciones.

Los sistemas distribuidos están compuestos principalmente por varias computadoras. En general, cada una con su propio procesador, memoria local, subsistema de entrada/salida e incluso memoria persistente.

➤ **Interconexión.**

Existen vías que permiten la comunicación entre las computadoras, a través de las cuales pueden transmitir información.

➤ **Estado compartido.**

Las computadoras cooperan para mantener algún tipo de estado compartido. Dicho de otra forma, puede describirse el funcionamiento correcto del sistema como el mantenimiento de una serie de variables globales que requiere la coordinación de varias computadoras.

Hay un *sistema operativo* de alto nivel, que unifica e integra el control de los componentes. El hecho de la distribución es *transparente*, permitiendo que los servicios puedan ser solicitados especificando simplemente su nombre (no su localización). El funcionamiento de los recursos físicos y lógicos está caracterizado por una *autonomía coordinada*.

Características

➤ **Concurrencia**

Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.

➤ **Carencia de reloj global**

La coordinación para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida a los componentes.

➤ **Fallos independientes de los componentes**

Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

Evolución

➤ **Procesamiento central (Host)**

Uno de los primeros modelos de computadoras interconectadas, llamadas centralizadas, donde todo el procesamiento de la organización se llevaba a cabo en una sola computadora, normalmente un *Mainframe*, y los usuarios empleaban sencillas computadoras personales.

Uno de los problemas que se presentó en este modelo es que cuando la carga de procesamiento aumentaba se tenía que cambiar el hardware del Mainframe, lo cual es más

costoso que añadir más computadoras personales, clientes o servidores que aumenten la capacidad.

El otro problema que surgió fueron las modernas interfases gráficas de usuario, las cuales podían conllevar a un gran aumento de tráfico en los medios de comunicación y por consiguiente podían colapsarse.

➤ Grupo de Servidores

Otro modelo que entró a competir con el anterior, también un tanto centralizado, son un grupo de computadoras actuando como servidores, normalmente de archivos o de impresión, poco inteligentes para un número de minicomputadoras que hacen el procesamiento conectados a una red de área local.

El problema de este modelo es que podría generarse una saturación de los medios de comunicación entre los servidores poco inteligentes y las minicomputadoras, por ejemplo cuando se solicitan archivos grades por varios clientes a la vez, podían disminuir en gran medida la velocidad de transmisión de información.

➤ La Computación Cliente Servidor

Este modelo, que predomina en la actualidad, permite descentralizar el procesamiento y recursos, sobre todo de cada uno de los servicios y de la visualización de la interfaz gráfica de usuario. Esto hace que ciertos servidores estén dedicados solo a una aplicación determinada y por lo tanto ejecutarla en forma eficiente.

Cliente-Servidor

Es el sistema donde el cliente es una máquina que solicita un determinado servicio y se denomina servidor a la máquina que lo proporciona. Los servicios pueden ser:

- Ejecución de un determinado programa.
- Acceso a un determinado banco de información.
- Acceso a un dispositivo de hardware.

Es un elemento primordial, la presencia de un medio físico de comunicación entre las máquinas, y dependerá de la naturaleza de este medio la viabilidad del sistema.

Protocolo

Es un conjunto bien conocido de reglas y formatos que se utilizan para la comunicación entre procesos que realizan una determinada tarea. Se requieren dos partes:

- Especificación de la secuencia de mensajes que se han de intercambiar.
- Especificación del formato de los datos en los mensajes.

Un protocolo permite que componentes heterogéneos de sistemas distribuidos puedan desarrollarse independientemente, y por medio de módulos de software que componen el protocolo, haya una comunicación transparente entre ambos componentes. Es conveniente mencionar que estos componentes del protocolo deben estar tanto en el receptor como en el emisor.

Middleware

El middleware es la capa de software intermedio entre el cliente y el servidor. Esta capa de software permite gestionar los mecanismos de comunicación. Existen dos tipos de software:

- **Software intermedio general.** Servicios generales que requieren todos los clientes y servidores.
- **Software intermedio de servicios.** Software asociado a un servicio en particular,

Características

- Independiza el servicio de su implantación, del sistema operativo y de los protocolos de comunicación.
- Permite la convivencia de distintos servicios en un mismo sistema.
- Permite la transparencia en el sistema.
- Modelo tradicional: Monitor de teleproceso o CICS, Tuxedo, Encina.
- Modelo OO: CORBA.

Ventajas de los sistemas distribuidos

Con respecto a sistemas centralizados:

1. Una de las ventajas de los sistemas distribuidos es la **economía**, pues es mucho más barato, añadir servidores y clientes cuando se requiere aumentar la potencia de procesamiento. Además de que procesadores son más poderosos y menos costos, lo cual permite:

- El desarrollo de estaciones con más capacidades
- Las estaciones satisfacen las necesidades de los usuarios.
- Uso de nuevas interfaces.

2. El **trabajo en conjunto**. Por ejemplo: en una fábrica de ensamblado, los robots tienen sus CPU diferentes y realizan acciones en conjunto, dirigidos por un sistema distribuido.

3. Tienen una **mayor disponibilidad y confiabilidad**. Los sistemas son poco propensos a fallas, ya que al estar distribuida la carga de trabajo en muchas máquinas la falla de una de ellas no afecta a las demás, y el sistema sobrevive como un todo. Existen mayores servicios que elevan la funcionalidad (Monitoreo, Telecontrol, Correo Eléctrico, etc.).

4. Capacidad de **crecimiento incremental**. Se puede añadir procesadores al sistema incrementando su potencia en forma gradual según sus necesidades. Debido a que su crecimiento es modular se pueden incluir rápidamente nuevos recursos, ya que los recursos actuales no afectan.

Con respecto a las PC Independientes:

1. Se pueden **compartir recursos**, como programas (software) y dispositivos (Hardware). Al compartir recursos se satisfacen las necesidades de muchos usuarios a la vez.
2. Se logra una **mejor comunicación** entre las personas, debido a la disponibilidad de elementos de comunicación y al desarrollo de las nuevas técnicas de comunicación.

3. Tienen **mayor eficiencia y flexibilidad**, la carga de trabajo se puede distribuir entre diferentes computadoras, permitiendo así:
 - Una respuesta rápida.
 - Una ejecución concurrente de procesos.
 - El empleo de técnicas de procesamiento distribuido.

Desventajas de los sistemas distribuidos

Cuando se construye cualquier tipo de sistema distribuido hay varios problemas que deben ser resueltos. Estos problemas son fundamentales porque están presentes en cualquier sistema de este tipo, por su propia naturaleza.

- El principal problema es el software, diseño, implementación y uso del software distribuido, pues presenta numerosos inconvenientes. Las *principales interrogantes* son los siguientes:

¿Qué tipo de sistema operativo, lenguaje de programación y aplicaciones son adecuados para estos sistemas?

¿Cuánto deben saber los usuarios de la distribución?

¿Qué tanto debe hacer el sistema y qué tanto deben hacer los usuarios?

La respuesta a estos interrogantes no es uniforme entre los especialistas, pues existe una gran diversidad de criterios y de interpretaciones al respecto.

- **Fallo independiente.** Puede ocurrir que parte de las computadoras que componen el sistema dejen de funcionar, y sin embargo el resto continúe correctamente. En este caso, habitualmente se quiere que el sistema continúe prestando servicio, aunque quizás de forma reducida.
- **Comunicación no fiable.** Las redes que conectan las computadoras no están, en general, exentas de errores. Es muy habitual que los datos transmitidos se pierdan, se corrompan o se repitan por problemas de la red. El sistema distribuido ha de estar diseñado de forma que tolere estas situaciones.
- **Comunicación insegura.** En general, los datos que se transmiten por la red pueden ser observados sin autorización, o incluso modificados intencionadamente.
- **Alto costo de la comunicación.** En la mayor parte de los casos la interconexión entre las computadoras proporciona menos ancho de banda y más latencia que sus buses de comunicación internos.

En general se considera que las ventajas superan a las desventajas, si estas últimas se administran seriamente.

Aplicación

En cuanto a los campos de aplicación, se pueden distinguir por un lado aquellos donde la distribución es fundamentalmente un medio para conseguir un fin y por otro aquellos donde es un problema en sí mismo.

Con respecto a los primeros, el uso de soluciones distribuidas sirve para acercarse a las siguientes metas:

- **Computación masivamente paralela**, de propósito general y de alta velocidad.
- **Tolerancia a fallos** (confianza, disponibilidad).
- Respuesta a demandas con requisitos de **tiempo real**.

Para el segundo caso, son los propios requisitos de la aplicación los que forzan a evolucionar hacia soluciones distribuidas:

- **Bases de datos distribuidas**. Es necesario acceder a los datos desde lugares geográficamente dispersos, y además puede ser conveniente (e incluso imprescindible) almacenarlos también en varios lugares diferentes.
- **Fabricación automatizada**. Es necesaria la colaboración de muchos procesadores para coordinar las tareas en una fábrica.
- **Supervisión remota y control**. Los sensores, los actuadores y los nodos donde se toman las decisiones de control pueden estar en diferentes partes de un sistema distribuido.
- **Toma de decisiones coordinada**. Hay muchas aplicaciones donde es necesario que varios procesadores participen en la toma de decisiones, por ejemplo, porque cada uno de ellos tiene una parte de los datos relevantes.

Desafíos

Heterogeneidad de los componentes. La interconexión, sobre todo cuando se usa Internet, se da sobre una gran variedad de elementos de hardware y software, por lo cual necesitan de ciertos estándares que permitan esta comunicación. Los middleware, son elementos de software que permiten una abstracción de la programación y el enmascaramiento de la heterogeneidad subyacente sobre las redes. También el middleware proporciona un modelo computacional uniforme.

Extensibilidad. Determina si el sistema puede extenderse y reimplementarse en diversos aspectos (añadir y quitar componentes). La integración de componentes escritos por diferentes programadores es un auténtico reto.

Seguridad. Reviste gran importancia por el valor intrínseco para los usuarios. Tiene tres componentes:

- **Confidencialidad**.- Protección contra individuos no autorizados.
- **Integridad**.- Protección contra la alteración o corrupción.
- **Disponibilidad**.- Protección contra la interferencia con los procedimientos de acceso a los recursos.

Escalabilidad. El sistema es escalable si conserva su efectividad al ocurrir un incremento considerable en el número de recursos y en el número de usuarios.

Tratamiento de fallos. La posibilidad que tiene el sistema para seguir funcionando ante fallos de algún componente en forma independiente, pero para esto se tiene que tener alguna alternativa de solución.

Técnicas para tratar fallos:

1. **Detección de fallos.** Algunos fallos son detectables, con comprobaciones
 - a) **Enmascaramiento de fallos.** Algunos fallos detectados pueden ocultarse o atenuarse.
 - b) **Tolerancia de fallos.** Sobre todo en Internet se dan muchos fallos y no es muy conveniente ocultarlos, es mejor tolerarlos y continuar.
 - c) **Recuperación frente a fallos.** Tras un fallo se deberá tener la capacidad de volver a un estado anterior.
2. **Redundancia.** Se puede usar para tolerar ciertos fallos.
3. **Concurrencia.** Compartir recursos por parte de los clientes a la vez.
4. **Transparencia.** Es ocultar al usuario y al programador las aplicaciones de la separación de los componentes en un sistema distribuido. Se identifican 8 formas de transparencia:
 - a) **De Acceso.** Se accede a recursos locales y remotos de forma idéntica.
 - b) **De ubicación.** Permite acceder a los recursos sin conocer su ubicación.
 - c) **De concurrencia.** Usar un recurso compartido sin interferencia.
 - d) **De replica.** Permite utilizar varios ejemplares de cada recurso.
 - e) **Frente a fallos.** Permite ocultar los fallos.
 - f) **De movilidad.** Permite la reubicación de recursos y clientes sin afectar al sistema.
 - g) **De prestaciones.** Permite reconfigurar el sistema para mejorar las prestaciones según su carga.
 - h) **Al escalado.** Permite al sistema y a las aplicaciones expandirse en tamaño sin cambiar la estructura del sistema o los algoritmos de aplicación.

Aplicaciones

Sistemas Comerciales. Inicialmente fueron construidos con hardware dedicado y entornos centralizados, son, por sus características de distribución geográfica y necesidad de acceso a sistemas distintos, ideales para implementarse en sistemas distribuidos. Requieren ciertas características de fiabilidad, seguridad y protección. Algunos ejemplos son sistemas de reservas de líneas aéreas, aplicaciones bancarias, cajas y gestión de grandes almacenes.

Redes WAN. Debido al gran crecimiento de este tipo de redes (Internet), ha tomado gran importancia el intercambio de información a través de la red. Los servicios comunes que brinda Internet son correo electrónico, servicio de noticias, transferencia de archivos y la World Wide Web.

Aplicaciones Multimedia. Son las últimas incorporaciones a los sistemas distribuidos. Estas aplicaciones imponen ciertas necesidades de hardware para poder tener una velocidad y regularidad de transferencia de una gran cantidad de datos. Los ejemplos de estos sistemas son: videoconferencia, televigilancia, juegos multiusuarios, enseñanza asistida por computadora, áreas de la informática aplicada a los sistemas distribuidos (en este punto se tienen en cuenta toda la variedad de aplicaciones de los sistemas distribuidos, pues su diseño involucra a muchas áreas, por ejemplo: Comunicaciones, Sistemas operativos distribuidos, Base de datos distribuidas, servidores distribuidos de archivos, lenguajes de programación distribuidos, sistemas de tolerancia de fallos).

1.2 Equipo de cómputo de alto desempeño

1.2.1 Arquitecturas paralelas

Todos los sistemas distribuidos constan de varias computadoras, organizadas de diversas formas, especialmente respecto de la forma de interconectarlas entre sí y de los *esquemas de comunicación* utilizados.

Existen diversos *esquemas de clasificación* para los sistemas de cómputo con varias computadoras: Uno de los más conocidos es la “*Taxonomía de Flynn*”:

La Taxonomía de Flynn considera como características esenciales el *número de flujo de instrucciones* y el *número de flujos de datos*. La clasificación incluye equipos SISD, SIMD, MISD y MIMD.

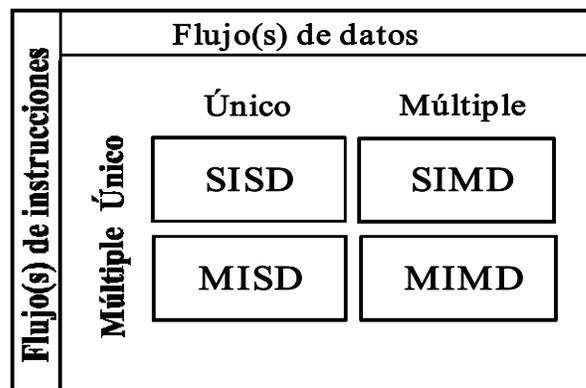


Figura 1.2 Clasificación de los Sistemas de Cómputo de Flynn

SISD (Single Instruction Single Data)

Un flujo de instrucciones y un flujo de datos. Poseen un único procesador.

Las instrucciones se ejecutan secuencialmente, pero pueden estar traslapadas en las etapas de ejecución. Pueden tener más de una unidad funcional bajo supervisión de una sola unidad de control.

Es la organización de la mayoría de las computadoras serie actuales.

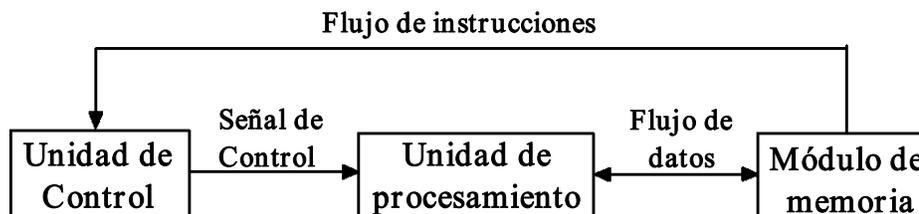


Figura 1.3 Arquitectura SISD

SIMD (Single Instruction Multiple Data)

Un flujo de instrucciones y varios flujos de datos. Se refiere a ordenar procesadores con una unidad de instrucción que: busca una instrucción e instruye a varias unidades de datos para que la lleven a cabo en paralelo, cada una con sus propios datos. Son útiles para el cómputo que repite los mismos cálculos en varios conjuntos de datos.

Las computadoras SIMD poseen una estructura semejante a los anteriores, pero operan sobre vectores de datos y presentan la ventaja de que todas las unidades de ejecución paralela están sincronizadas y todas responden a una sola instrucción que emana de un único contador de programa (PC).

Estos procesadores son muy eficientes a la hora de tratar con vectores en ciclos for y en situaciones de paralelismo de datos (masivos), sin embargo su rendimiento decrece en las sentencias de selección, al tener que ejecutar diferentes caminos para distintas opciones.

Múltiples unidades de ejecución o elementos de proceso (EP) son supervisados por una única unidad de control. Todos los EP reciben la misma instrucción emitida por la unidad de control pero operan sobre diferentes conjuntos de datos procedentes de flujos distintos.

Esta organización corresponde a los procesadores matriciales.

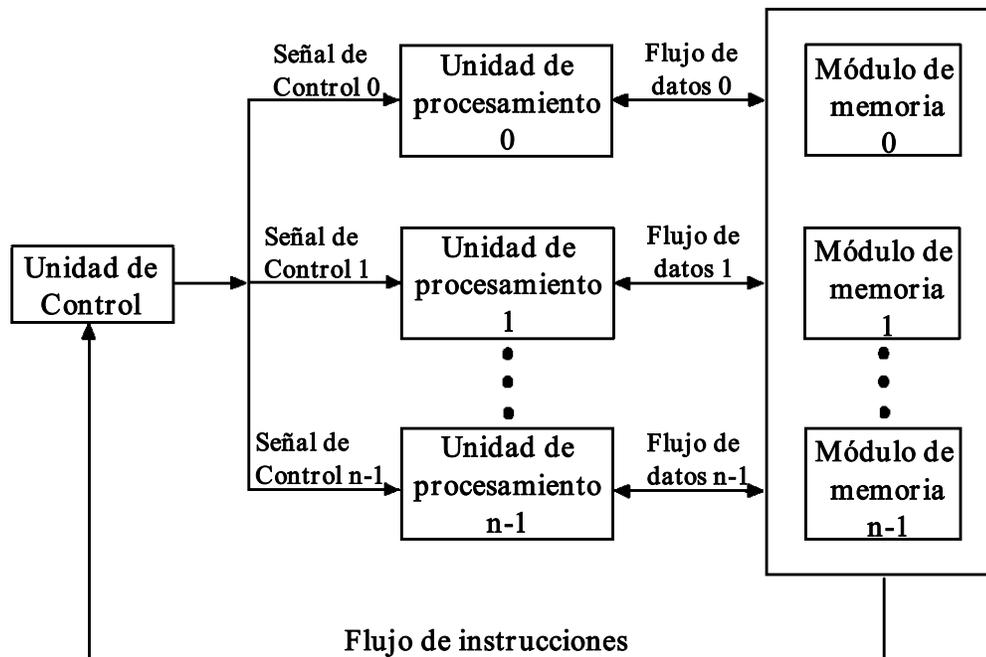


Figura 1.4 Arquitectura SIMD

MISD (Multiple Instruction Single Data)

Múltiple flujo de instrucciones y un solo flujo de datos. No se presenta en la práctica.

Existen n unidades de procesadores.

Cada una recibe distintas instrucciones que operan sobre el mismo flujo de datos y sus derivados. Los resultados (la salida) de un procesador pasan a ser la entrada (los operandos) del siguiente procesador.

Esta clasificación no es un sistema viable para la computación.

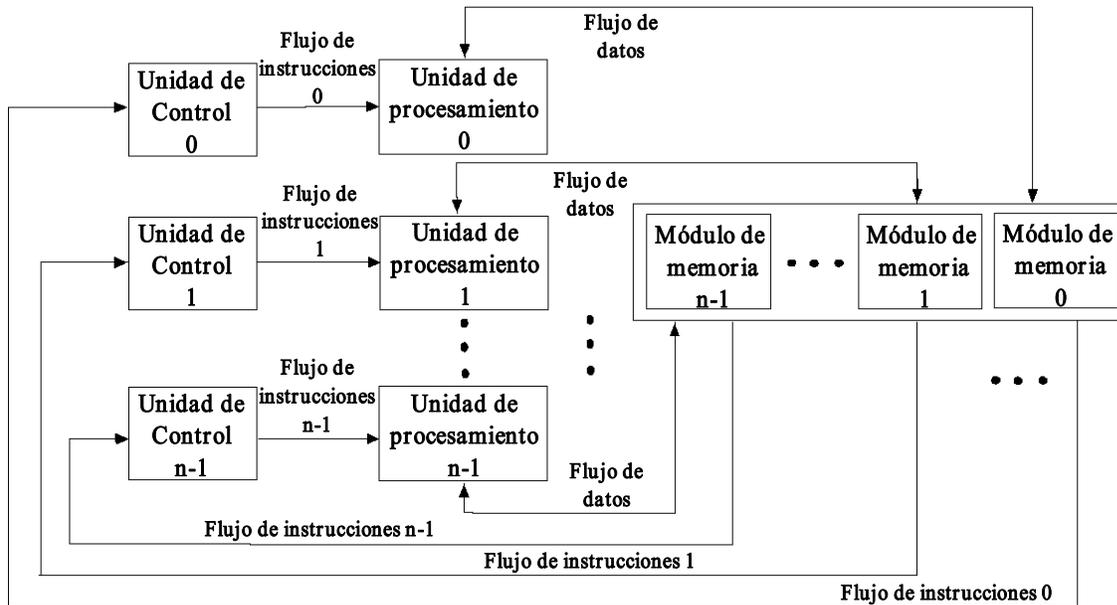


Figura 1.5 Arquitectura MISD

MIMD (Multiple Instruction Multiple Data)

Múltiple flujo de instrucciones- múltiple flujo de datos.

Se basan en la idea de crear potentes computadoras conectando otras más pequeñas que cooperan entre sí frente al costo de crear una supercomputadora SISD. Presentan además de la ventaja de costo un gran soporte a fallos (si disponemos de n procesadores siempre se puede continuar la unidad de procesamiento con $n-1$), lo cual es una garantía en máquinas a las que se exige un servicio continuo.

Un grupo de computadoras independientes, cada una con su propio contador del programa, programa y datos. Todos los sistemas distribuidos son de este tipo.

Esta arquitectura incluye a sistemas con más de un procesador capaz de ejecutar varios programas multiprocesador.

Un avance sobre la clasificación de Flynn incluye la división de las computadoras MIMD en dos grupos:

Multiprocesadores: poseen memoria compartida, los distintos procesadores comparten el mismo espacio de direcciones virtuales.

Multicomputadoras: no poseen memoria compartida, son un grupo de computadoras conectadas mediante una red.

Otro aspecto de la clasificación considera el *acoplamiento entre los equipos*:

Sistemas fuertemente acoplados: El retraso al enviar un mensaje de una computadora a otra es corto y la tasa de transmisión es alta. Generalmente se les utiliza como sistemas paralelos.

Sistemas débilmente acoplados: El retraso de los mensajes entre las máquinas es grande y la tasa de transmisión es baja. Generalmente se le utiliza como sistemas distribuidos.

Generalmente los multiprocesadores están más fuertemente acoplados que las multicomputadoras.

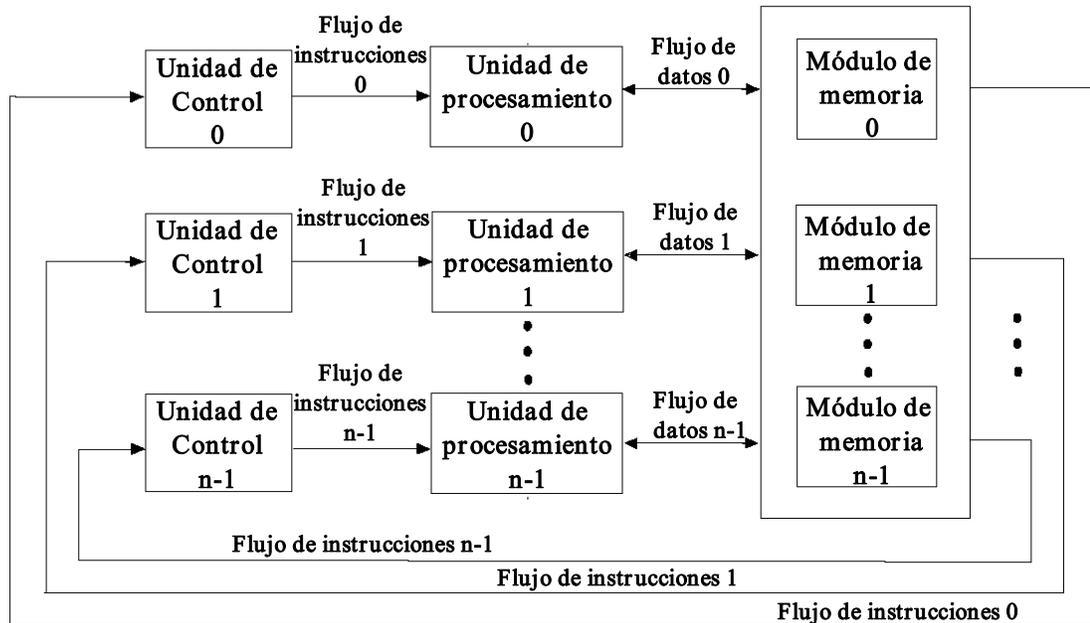


Figura 1.6 Arquitectura MIMD

Algunas de las arquitecturas paralelas que no se encuentran incluidas en la clasificación de Flynn se encuentran agrupadas en las siguientes categorías:

➤ **SMP** (Symmetric Multiprocessors): este término incluye solamente a los procesadores simétricos los cuales se encuentran conectados a la memoria compartida de manera uniforme y tienen igual oportunidad de acceso a los recursos de memoria compartida.

➤ **MPP** (Massively Parallel Processor) es una arquitectura de computadores que provee un rápido desempeño haciendo que múltiples CPU estén disponibles para completar procesos individuales simultáneamente (multiprocesamiento).

➤ La arquitectura **ccNUMA** (Cache Coherent Nonuniform Memory Access) es una extensión de SMP. Diseñada para superar los cuellos de botella inherentes de SMP, la arquitectura ccNUMA deja a los proveedores construir servidores en gran escala. ccNUMA ofrece todos los mejores beneficios de SMP y MPP, sin ninguna de sus desventajas.

Ventajas de ccNUMA

- **Modelo. de memoria compartida.** Como los sistemas SMP, presentan un solo, global y unificado modelo de memoria.
 - **Multi-CPU, Multiprocesamiento.** Como SMP y MPP, soporta multi-CPU y multiprocesamiento en configuraciones de cierto número de CPU.
 - **Distribución de la carga.** Como MPP, distribuye E/S y accesos a memoria por múltiples subsistemas, pero difiere de MPP en que la carga se balancea automáticamente.
 - **Operaciones .de E/S concurrentes.** Como MPP. soporta múltiples operaciones concurrentes al disco usando independientes, pero totalmente conectados, subsistemas de E/S.
- **Sistemas Distribuidos:** se refieren a redes de computadoras independientes, son combinaciones de SMP, MPP, *clusters* y computadoras individuales.
- **Clusters:** son agrupaciones de PCs unidas por una red de alta velocidad. Es un grupo de computadoras las cuales trabajan en conjunto para resolver una tarea. Un *cluster* está conformado por varias computadoras las cuales se comunican por medio de una conexión a red trabajando en un proyecto el cual sería muy largo para una sola computadora, resolviéndolo en un tiempo razonable.

Existen varios tipos de *clusters* uno de los más comunes es el *cluster* computacional: éste generalmente es utilizado para procesar grandes cantidades de datos. Otro tipo de *cluster* es el de base de datos llamado libre de fallos o alta disponibilidad se ocupa para almacenar bases de datos las cuales deben estar siempre en funcionamiento.

1.2.2 Clusters

Antecedentes

Los *clusters* son la principal alternativa para el alto desempeño en computación. *High Performance Computing* HPC, Cómputo de Alto Desempeño.

En muchas ramas de las ciencias la complejidad de los problemas que se estudian, requieren contar con acceso a una supercomputadora, siendo éstas, máquinas poderosas que pueden desarrollar varios miles de millones de operaciones por segundo. Las supercomputadoras tradicionales emplean procesamiento en paralelo; contienen arreglos de microprocesadores ultrarrápidos que trabajan en sincronía para resolver problemas complejos como pronósticos numéricos del estado del tiempo, o modelar estructuras complejas de la materia. Los fabricantes de supercomputadoras como Cray, IBM, Silicon Graphics, entre otros, producen modelos por diseño especial y cuestan decenas de millones de dólares - precios que van más allá de los presupuestos de inversión de los grupos de investigación.

En los últimos años, el personal académico de diversas universidades y centros de investigación se han dado a la tarea de aprender a construir sus propias supercomputadoras conectando computadoras personales y desarrollando software para enfrentar tales problemas extraordinarios.

Definición

Los *clusters* son sistemas que enlazan a varias computadoras llamadas nodos, por medio de una red local de comunicaciones que es controlada a través de una computadora o nodo maestro. Están contruidos utilizando componentes de hardware común y software libre, juegan hoy en día, un papel importante en la solución de problemas de las ciencias, las ingenierías y aplicaciones comerciales.

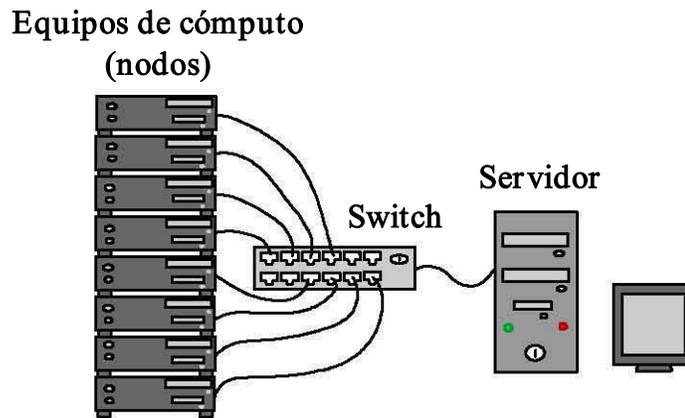


Figura 1.7 Cluster

Los *clusters* han evolucionado para apoyar actividades en aplicaciones como supercómputo y software de misiones críticas, servidores Web y comercio electrónico, bases de datos de alto rendimiento, entre otros.

El cómputo en *clusters* surge como resultado de la convergencia de varias tendencias que incluyen, la disponibilidad de microprocesadores de alto rendimiento más económicos y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, y la creciente necesidad de potencia computacional para aplicaciones en las ciencias computacionales y comerciales. Por otro lado, la evolución y estabilidad que ha alcanzado el sistema operativo GNU/Linux, ha contribuido importantemente al desarrollo de muchas tecnologías nuevas, entre ellas la de *clusters*.

La idea de *cluster de alto rendimiento* es hacer que un gran número de máquinas individuales actúen como una sola máquina muy potente. Este tipo de *clusters* se aplica mejor en problemas grandes y complejos que requieren una cantidad enorme de potencia computacional. Entre las aplicaciones más comunes de *clusters* de alto rendimiento se encuentra el pronóstico numérico del estado del tiempo, astronomía, investigación en criptografía, análisis de imágenes, etc.

El *cluster de servidores virtuales*, permite que un conjunto de servidores de red compartan la carga de trabajo de tráfico de sus clientes. Al balancear la carga de trabajo de tráfico en un arreglo de servidores, mejora el tiempo de acceso y confiabilidad. Además como es un conjunto de servidores el que atiende el trabajo, la falla de uno de ellos no ocasiona una falla catastrófica total. Este tipo de servicio es de gran valor para compañías que experimentan grandes volúmenes de tráfico en sus sitios Web.

Este tipo de *clusters*, involucra el tener servidores que actúan entre ellos como respaldos vivos de la información que sirven. Este tipo de *clusters* se les conoce como *clusters de alta disponibilidad* o *cluster de redundancia*.

Funcionamiento

En su parte central, la tecnología de *clusters* consta de dos partes:

El primer componente, consta de un sistema operativo, un conjunto de compiladores y aplicaciones especiales, que permiten que los programas que se ejecutan sobre esta plataforma tomen las ventajas de la tecnología de *clusters*.

El segundo componente es la interconexión de hardware entre las máquinas (nodos) del *cluster*. Se han desarrollado interfaces de interconexión especiales muy eficientes, pero comúnmente las interconexiones se realizan mediante una red ethernet dedicada de alta velocidad. Es mediante esta interfaz que los nodos del *cluster* intercambian entre sí asignación de tareas, actualizaciones de estado y datos del programa. Existe otra interfaz de red que conecta al *cluster* con el mundo exterior.

Cuando se habla de resolver un problema en paralelo, se refiere a dividir un trabajo en varias tareas que se pueden desarrollar en paralelo. Esto es lo que sucede en un *cluster* de alto rendimiento y la solución de los problemas se termina más rápido.

La tecnología de *clusters* de alto rendimiento para Linux más conocida es el la tecnología *Beowulf*. Esta tecnología se desarrolló en la NASA y puede proporcionar potencial de cómputo del tipo de una supercomputadora utilizando computadoras personales sencillas. Al conectar éstas entre sí mediante una red ethernet de alta velocidad, las computadoras personales se combinan para lograr la potencia de una supercomputadora.

Los servidores de un *cluster* de alta disponibilidad normalmente no comparten la carga de procesamiento que tiene un *cluster* de alto rendimiento. Tampoco comparten la carga de tráfico como lo hacen los *clusters* de balance de carga. Su función es la de estar listos para entrar inmediatamente en funcionamiento en el caso de que falle algún otro servidor. La característica de flexibilidad que proporciona este tipo de tecnología de *cluster*, lo hacen necesario en ambientes de intercambio intensivo de información.

Los *clusters* de alta disponibilidad permiten un fácil mantenimiento de servidores. Una máquina de un *cluster* de servidores se puede sacar de línea, apagarse y actualizarse o repararse sin comprometer los servicios que brinda el *cluster*. Cuando el servidor vuelve a estar listo, se incorpora al *cluster* y se puede dar servicio al siguiente servidor del grupo.

Tecnologías del *cluster*

MOSIX.

Esta tecnología basada en Linux, permite realizar balance de carga para procesos particulares en un *cluster*. Trabaja como un *cluster* de alto rendimiento en el sentido de que está diseñado para tomar ventaja del hardware más rápido disponible en el *cluster* para cualquier tarea que le sea asignada. Pero, lo realiza balanceando la carga de las varias tareas en varias máquinas. Una de las grandes ventajas de MOSIX es que no requiere la confección especial de software como lo requiere los *clusters* tipo *Beowulf*. Los usuarios ejecutan sus programas normalmente y el sistema MOSIX se encarga del resto.

Piranha

Otra tecnología de *clusters* es el paquete Piranha, que permite a los servidores Linux proveer alta disponibilidad sin la necesidad de invertir cantidades mayores en hardware. La tecnología de *cluster* está basada completamente en software, donde los servidores se comunican en una red de alta velocidad. Se puede configurar para trabajar como *cluster* de alta disponibilidad o de balance de carga.

Cluster Beowulf

Antecedentes

El proyecto *Beowulf Cluster* tuvo inicio en (Goddard Space Flight Center GSFC) en el verano de 1994 cuando se ensamblaron 16 nodos, para el desarrollo en ciencias de la tierra y el espacio, el proyecto rápidamente se extendió a la NASA algunos laboratorios, también diversas universidades al rededor del mundo. El primer *cluster* se formó con 16 nodos DX4 conectados en una canal ethernet el máximo poder de procesamiento fue de un GigaFLOP (Floting Point Operation / seg.)

Los actuales sistemas *cluster* cuentan con procesadores y tecnología de red cada vez más eficientes y rápidos, consisten en cientos rara vez en miles de nodos con el poder de procesamiento en varios TeraFLOPS.

En 1994, se integró el primer *cluster* de PCs en el Centro de Vuelos Espaciales Goddard de la NASA, para resolver problemas computacionales que aparecen en las ciencias de la Tierra y el Espacio. Los pioneros de este proyecto fueron Thomas Sterling, Donald Becker y otros científicos de la NASA. El *cluster* de PCs desarrollado tuvo una eficiencia de 70 MegaFLOPS (millones de operaciones de punto flotante por segundo). Los investigadores de la NASA le dieron el nombre de *Beowulf*.

En 1996, hubo también otros dos sucesores del proyecto *Beowulf* de la NASA. Uno de ellos es el proyecto *Hyglac* desarrollado por investigadores del Instituto Tecnológico de California (CalTech) y el Laboratorio de Propulsión Jet (JPL), y el otro, el proyecto *Loki* construido en el Laboratorio Nacional de Los Álamos, en Nuevo México. Cada *cluster* se integró con 16 microprocesadores Intel Pentium Pro y tuvieron un rendimiento sostenido de más de un GigaFLOP con un costo menor a \$50,000 dólares.

En 1996, en el Laboratorio Nacional de Oak Ridge en Tennessee, se enfrentaban al problema de elaboración de un mapa de las condiciones ambientales del territorio de Estados Unidos. El territorio fue dividido en 7.8 millones de celdas de 1Km. Cada celda contenía la información de 25 variables, desde la precipitación promedio mensual hasta el contenido de Nitrógeno del suelo. Ninguna estación de trabajo o PC podría con esta tarea. Se requería una supercomputadora de procesamiento paralelo. En la actualidad cuentan con un *cluster* de 130 PCs para trabajar en la elaboración del mapa de eco regiones.

Ventajas

Los *clusters* de tipo *Beowulf* son los más comunes en el terreno de la Computación Científica y para toda tarea que necesite una gran cantidad de tiempo de procesamiento o de potencia de cálculo. Las ventajas que ofrece un *cluster Beowulf* sobre otras opciones comerciales o de software libre son las siguientes:

Beowulf es un sistema estándar con una serie de características determinadas: Máquinas con hardware estándar, sobre todo PCs o máquinas estándar montadas en Racks, normalmente con los componentes mínimos para proporcionar la potencia de cálculo; las máquinas corren software libre. Los equipos se dedican únicamente a proporcionar potencia de computación al *cluster* y no para otras tareas, lo mismo se aplica para la red, que se dedica únicamente al paso de mensajes. Estas características son las que distinguen a un *cluster Beowulf* de otros tipos de *clusters* basados también en paso de mensajes. El sistema de paso de mensajes del *cluster Beowulf* está basado en PVM o MPI.

Ventajas del *Cluster*

- Incremento de velocidad de procesamiento ofrecido por los *clusters* de alto rendimiento,
- Incremento del número de transacciones o velocidad de respuesta ofrecido por los *cluster* de balance de carga
- Incremento de confiabilidad ofrecido por los *clusters* de alta disponibilidad

Aplicaciones

En investigaciones meteorológicas y pronóstico numérico del estado del tiempo, se requiere el manejo de cantidades masivas de datos y cálculos muy complejos. Al combinar el poder de muchas máquinas del tipo estación de trabajo o servidor, se pueden alcanzar niveles de rendimiento similares a los de las supercomputadoras, pero a menor costo (pues éstas requieren de hardware y software especializado muy caro, así como personal de soporte técnico dedicado).

En un sitio Web de mucho tráfico, si no se cuenta con un plan de alta disponibilidad, cualquier problema menor de una tarjeta de red, puede hacer que un servidor quede completamente inutilizado. Pero al contar con servidores redundantes y servidores de respaldo instantáneos, se puede reparar el problema mientras el sitio funciona con el plan de respaldo, sin suspensión de servicio.

Los sistemas *cluster* han empezado a ser utilizados en donde los sistemas *mainframes* o supercomputadoras eran utilizados en el pasado.

Los *clusters* han formado parte significativa en el avance de las siguientes industrias.

- Investigación científica y defensa
- Procesos de energía sísmica
- Efectos visuales y entretenimiento
- Internet/ISP/ASP

En algunos puntos específicos

- Identificar nuevas partículas sub-atómicas
- Investigación para descubrir un fármaco para el SIDA
- Exploración de recursos como el petróleo y el gas
- Investigación del genoma humano

1.3 Surgimiento de las GRID

Además del concepto de *cluster*, existe otro concepto más amplio y general que es el cómputo GRID (*GRID computing*). Una GRID es un tipo de sistema paralelo y distribuido que permite compartir, seleccionar y añadir recursos que se encuentran distribuidos a lo largo de dominios administrativos múltiples basados en su disponibilidad, capacidad, rendimiento, costo y calidad de servicio que requiere un usuario.

Si los recursos distribuidos se encuentran bajo la administración de un sistema central único de programación de tareas, entonces se refiere a un *cluster*. En un *cluster*, todos los nodos trabajan en cooperación con un objetivo y una meta común pero la asignación de recursos los ejecuta un solo administrador centralizado y global. En una GRID, cada nodo tiene su propio administrador de recursos y política de asignación.

Las aplicaciones presentes y futuras tienen una característica predominante: generan cantidades significativas de datos. Esto implica que para su proceso eficiente requieren de potencia de cálculo. Aplicaciones que generan ambientes gráficos como la realidad virtual, el diseño en ingeniería, la solución a complejos modelos matemáticos que involucran la predicción del clima, las aplicaciones utilizadas en las investigaciones del genoma, etc. Son sólo algunos ejemplos de este tipo de aplicaciones. Una PC de escritorio, no dispone del potencial de cálculo para dar respuesta de manera eficiente a este tipo de aplicaciones. Una solución posible está basada en supercomputadoras de alto rendimiento especializadas en el proceso intensivo de datos. Sin embargo los costos asociados a estos equipos son del orden de los millones de pesos. Obviamente resultan inalcanzables para los presupuestos de las instituciones medianas o pequeñas.

Una solución económica es construir un sistema de cómputo de gran capacidad y de un costo moderado asociando un conjunto de computadoras personales (PC) y comunicándolas por medio de una red de datos de alto desempeño. Este sistema conocido como *cluster* dispone de numerosos recursos distribuidos como procesadores, memorias y discos. Una explotación simple y eficaz de estos recursos requiere de un software que los administre como un todo, ocultando al usuario su carácter distribuido. El objetivo es de poder utilizar un *cluster* como una estación de trabajo de gran capacidad en almacenamiento, procesamiento y comunicación.

Sin embargo existen aplicaciones cuyas necesidades de cálculo son superiores a las proporcionadas por un *cluster* de tamaño mediano o pequeño. Una empresa no puede destinar recursos importantes para implementar un *cluster* con numerosos equipos.

La popularización del Internet, la disponibilidad de redes de comunicación de datos de alta velocidad, la tendencia al bajo costo y sobretodo el incremento en la eficiencia de los componentes de las computadoras personales están modificando la forma en que se utilizan las computadoras. El interés por utilizar y aprovechar al máximo los recursos computacionales distribuidos entre diferentes organizaciones para resolver algoritmos que requieren importantes cantidades de cálculo ha dado origen a lo que se conoce como la computación GRID.

La computación científica ha jugado tradicionalmente un papel muy importante en el avance de la informática. En los inicios de ésta era los problemas eran prácticamente la única fuerza impulsora que motivaron el desarrollo tanto de software como de hardware. Los problemas para los que se construyeron máquinas como el ENIAC procedían de las áreas técnicas y científicas. El primer compilador que se realizó (en 1957) fue el de Fortran I,

acrónimo de FORMula TRANslator, destinado al mercado técnico. Posteriormente, las computadoras se fueron aplicando también en otras áreas. La computación científica acabó convirtiéndose en sinónimo de cálculo intensivo y de grandes máquinas (computadoras vectoriales, paralelismo masivo, etc.) con un número muy limitado de usuarios que se ocupaban de problemas extremadamente técnicos y muy lejos de cualquier aplicación inmediata.

Si bien es cierto que es en las áreas donde las necesidades de velocidad y memoria son más solicitadas, donde la computación científica ha mostrado más su importancia. El mercado técnico y científico es hoy más grande que nunca y, debido a la importancia de la computadora en el desarrollo, diseño y fabricación de nuevos productos, está creciendo muy rápidamente. La última contribución realmente llamativa es el mismo World Wide Web. Este se desarrolló en el CERN (Centre Européenne pour la Recherche Nucleaire) como respuesta a las necesidades de colaboración entre grupos internacionales de investigadores. El posterior desarrollo de un browser gráfico (Mosaic) en el NCSA (National Center for Supercomputing Applications), también dedicado a la computación científica, proporcionó el empuje inicial para el enorme crecimiento que ha experimentado la Internet hasta convertirse en la base de lo que llaman la nueva economía.

Un último desarrollo generado en este campo y con un previsible gran impacto en el resto del mundo informático es la GRID. Este nombre fue puesto por analogía con el nombre inglés para las redes eléctricas. La idea básica es tratar el poder de cálculo o de acceso a datos como si se tratara de un fluido. El objetivo final es que desde cualquier punto de acceso a la red se puedan utilizar todos sus recursos de una manera transparente, independientemente de que aquellas máquinas que hacen los cálculos tengan distintas arquitecturas o que estén geográficamente dispersas o que unas se ocupen de servir o recoger los datos, otras de trabajar con ellos y otras de presentar los resultados. La realización total de este potencial depende de que seamos capaces de sincronizar muchos recursos distintos y dispersos para lo que hay que tener en cuenta, además del hardware, una serie de capas software que van desde las herramientas de programación a los protocolos de comunicaciones y seguridad que permiten el acceso a todos los recursos necesarios requeridos por una aplicación. Realizando autenticaciones entre distintas máquinas, garantizando accesos a ciertos recursos, etc. Y esto realizado de manera transparente para el usuario final, de modo que su uso no sea más complicado que un recurso puramente local.

La computación científica sigue siendo, pues, una fuerza importante en el desarrollo de la informática. Uno de los motivos para el auge que está teniendo esta rama es el abaratamiento de los equipos de cómputo de grandes prestaciones y el aumento de potencia de las computadoras tipo *workstation*. Además, la agrupación de estos en *clusters* y las tecnologías que ya ha generado la GRID han aumentado el tamaño y complejidad de los problemas tratables. Esto ha hecho que el número de usuarios interesados en resolver problemas en ciencia e ingeniería utilizando computadoras se haya incrementado tan notablemente que en algunos países se están ofreciendo especializaciones en una nueva rama, híbrida entre la informática y otra ciencia o ingeniería, bajo el nombre de "Ciencia e Ingeniería Computacional".

Las tecnologías GRID están teniendo un gran auge en los últimos años, debido a que permiten resolver problemas con gran demanda computacional y de datos, y que tienen en su propia definición una característica muy importante dentro de los sistemas distribuidos, como es el compartir recursos.

El desarrollo de las tecnologías de redes de computadoras en las últimas décadas ha hecho posible el desarrollo de sistemas de cómputo de altas prestaciones basados en conjuntos de computadoras independientes e interconectadas entre sí. A mediados de los años 90 aparecieron los primeros *clusters* de tipo *Beowulf*, seguidos de cerca por los *hiperclusters* y por las GRID.

La tecnología basada en *clusters* y GRID ofrece una cantidad y diversidad de recursos de cómputo prácticamente ilimitado, con un bajo costo. Esto permite la coexistencia de arquitecturas y servicios específicos, gran capacidad de cómputo, de almacenamiento, etc. Por otro lado, permite la construcción de sistemas de cómputo de altas prestaciones, con muy poca inversión para las empresas y centros de investigación.

Entre las ventajas fundamentales de la tecnología *cluster* y GRID podemos destacar las siguientes:

- Disponer de una gran potencia de cómputo sin necesidad de adquirir equipo costoso.
- Aumentar el aprovechamiento de los recursos informáticos.
- Mayor colaboración y compartición de recursos entre varios centros, lo que puede dar lugar a la creación de organizaciones virtuales.

Sin embargo, estos grandes sistemas también presentan una serie de problemas, tanto a los usuarios como a sus administradores. Fundamentalmente la heterogeneidad y accesibilidad de los recursos provocan una gran dificultad en su gestión y aprovechamiento. Problemas relacionados con el rendimiento y la disponibilidad real de los nodos son por lo tanto frecuentes en estos sistemas.

Esta transición de los sistemas multiprocesador y centralizados a los sistemas de cómputo de altas prestaciones basados en conjuntos de nodos independientes entre sí (*clusters* y GRID) llevan asociados una serie de cambios importantes en cuanto a la monitorización del sistema. Dada la gran cantidad de nodos de cómputo, dispositivos de I/O, redes de comunicaciones independientes pero relacionados entre sí, es importante tener una visión global del estado del sistema así como del estado particular de cada uno de estos componentes. Esto permitirá detectar fallos en componentes, cuellos de botella, o descensos pronunciados en el rendimiento de alguno de estos componentes. Por otro lado, dada la naturaleza distribuida de estos sistemas las herramientas de monitorización deben ser altamente escalables.

2. Las GRID

2.1 Concepto

Una GRID (*Global Resources Information Database*) es una infraestructura de hardware, software y comunicación que permite la integración de equipos de cómputo localizados en diferentes sitios geográficos y que pueden ser utilizados de manera compartida y transparente por una comunidad específica. La GRID se caracteriza por ser de gran escala, geográficamente distribuida, y compuesta por recursos heterogéneos pertenecientes a múltiples organizaciones administrativas que los comparten con el objetivo de proporcionar soporte computacional a un amplio rango de aplicaciones de forma transparente, de calidad, generalizado y consistente. El soporte proporcionado por la GRID es considerado de gran utilidad para aplicaciones de computación colaborativa dado que la GRID permite y fomenta la interacción entre humanos ya sea de forma síncrona o asíncrona a través de un espacio virtual.

El principio de la computación GRID es el de conjuntar recursos computacionales heterogéneos distribuidos geográficamente entre distintas organizaciones para formar un gran potencial de cálculo, almacenamiento y comunicación. Esta tecnología permite disponer de una infraestructura equipada con suficiente potencial para resolver problemas que involucren el procesamiento intensivo de datos a grande escala. La GRID debe proporcionar servicios accesibles por medio de un conjunto de protocolos e interfaces abiertos. El usuario de la GRID interactúa con un agente de recursos (*resource broker*) que oculta la complejidad y proporciona una vista única de los recursos distribuidos de la GRID. Así mismo el *broker* negocia los recursos requeridos por la aplicación del usuario, proporciona los servicios de monitorización de las actividades implicadas en la ejecución de la aplicación y recolecta los resultados para ponerlos a disposición del usuario. Este proceso se ilustra en la figura 2.1.

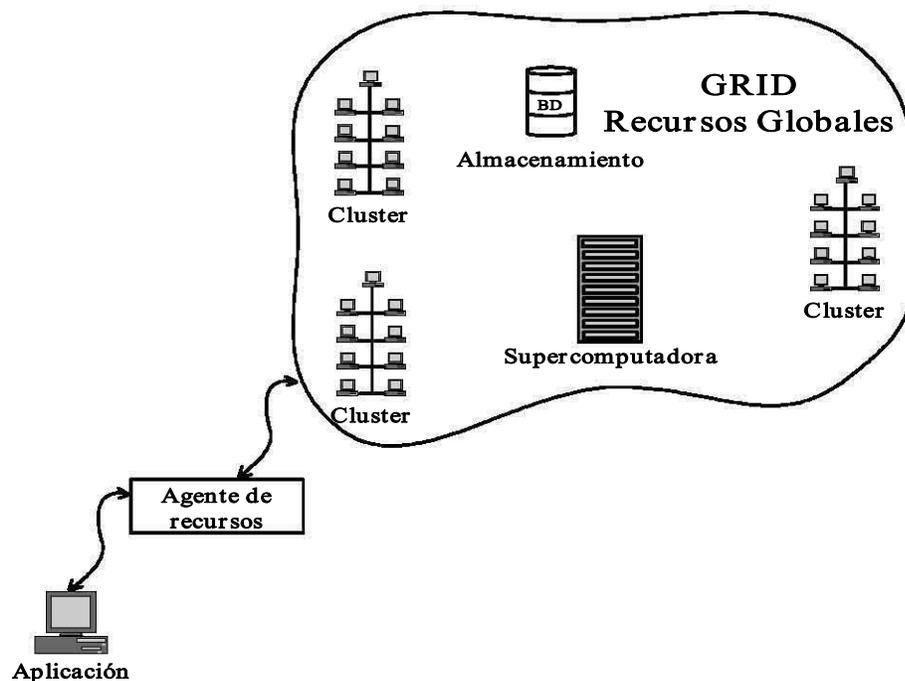


Figura 2.1 Interacción en la GRID

Se pueden distinguir tres categorías de la GRID: GRID de Información, GRID de Recursos y GRID de Servicios.

GRID de Información

Con la invención del World Wide Web (WWW) en los 90 se tomó el paso más importante hacia la estructura global de la GRID. El crecimiento exponencial de la Web creó una infraestructura de red pública y accesible, una GRID de información omnipresente capaz de entregar información de cualquier tipo a cualquier lugar en el mundo. En la actualidad, servicios P2P como Kazaa y Napster explotan las capacidades de este tipo de GRID ya que los datos asociados a la organización están almacenados a lo largo de miles de computadores personales, constituyendo una forma distribuida, escalable y dinámica de compartir música, video, películas o software.

GRID de Recursos

Es el más difícil de implementar porque es donde entran las empresas al ruedo. Es aquí donde se provee el uso de ciclos de procesamiento de PCs, *clusters* y supercomputadoras para usuarios autenticados y por supuesto tiene un componente de inversión que no siempre permite la gratuidad. La mayoría de estas GRID utilizan una herramienta llamada Globus Toolkit y se pueden encontrar en las áreas de la investigación farmacéutica, química, astrofísica, tratamiento y representación de video, post producción, simulación del clima, geología, etc. En esta categoría también están las GRID de datos que proporcionan mecanismos para almacenamiento seguro y redundante, las GRID de aplicaciones paralelas que permiten la donación de ciclos ociosos de una PC para procesar datos captados del espacio en búsqueda de patrones inteligentes y las GRID de acceso que son las que proveen video conferencias interactivas.

GRID de Servicios

Entrega tanto servicios como aplicaciones sin importar la ubicación geográfica. Las máquinas de búsqueda, portales, páginas de servidor activas y de diverso contenido dinámico, Passport, Hotmail y muchos otros son ejemplos de este tipo de GRID. Utilizando la arquitectura OGSA (*Open GRID Services Architecture*) conjuntamente con los *web services*, se puede lograr interoperabilidad entre los servicios sin importar ubicación geográfica o plataforma de ejecución.

Para facilitar la utilización y la implementación de esta infraestructura distribuida es necesario establecer una arquitectura global orientada hacia los servicios que simplifique el desarrollo y la puesta en operación de las aplicaciones que harán uso de estos servicios. Se ha definido el modelo OGSA que está constituido de 4 capas, como se muestra en la figura 2.2.

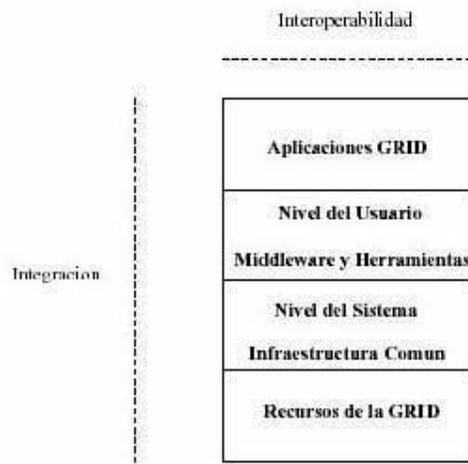


Figura 2.2 Modelo OGSA de la GRID

OGSA *Infrastructure*. Define la infraestructura que conforma la GRID. Incluye computadoras, *clusters*, redes de datos e instrumentos científicos.

OGSA *Services*. Proporciona los servicios necesarios para acceder con seguridad recursos remotos gestionando de manera eficiente la heterogeneidad. Los servicios proporcionados son seguridad, almacenamiento, información sobre el estado de los recursos y envío de trabajos remotos.

OGSA *Schemes*. Proporciona herramientas y ambientes de ejecución en un nivel de usuario que facilitan el acceso a los servicios de la GRID.

El surgimiento de la GRID como una plataforma para el proceso paralelo y la computación distribuida ha despertado el interés de instituciones y universidades en todo el mundo. Actualmente una gran comunidad de investigadores trabajan en diferentes proyectos relacionados con el desarrollo de la Computación GRID. Algunos de estos proyectos son EGGE, *CrossGRID*, *DataGRID*.

La OGSA representa una evolución hacia una arquitectura de sistema de *GRID* basada en conceptos y tecnologías de servicios Web.

OGSA es un producto de la comunidad *GRID*, los miembros de la alianza de Globus han hecho contribuciones significantes al desarrollo de OGSA.

2.2 Computación GRID

Definición

La computación GRID es el resultado de la evolución de la computación distribuida, en ella se trata de solventar el problema de las cargas de trabajo que puede enfrentar una organización en un momento determinado agregando y asignando recursos ya sean de cálculo, de almacenamiento o de red con el fin de lograr mayor eficiencia en los centros de cómputo. Todo esto llevado a cabo de forma coordinada y flexible.

Mientras que las redes convencionales se enfocan en la comunicación entre dispositivos (y personas), la computación *GRID* gestiona los ciclos no usados de todas las computadoras de la red para resolver problemas que exceden la capacidad de una sola máquina.

La computación GRID requiere un *software* especial que es único para cada proyecto.

La computación GRID permite compartir, seleccionar y anexar una gran variedad de recursos computacionales (supercomputadoras, *clusters*, sistemas de almacenamiento, fuentes de datos, instrumentos y personas) distribuidos geográficamente de manera dinámica, en tiempo de ejecución dependiendo de su disponibilidad, capacidad, rendimiento, costo y requerimientos de calidad de servicio (*QoS*) de los usuarios.

Se presenta como un único recurso para resolver grandes problemas de computo: modelado molecular de fármacos, análisis de la actividad cerebral y física de alta energía.

2.3 GRID y *cluster*

El objetivo de las GRID es la explotación de las sinergias resultantes de la cooperación y capacidad de compartir y sumar la potencia computacional distribuida y ofrecerlas como un único servicio.

En el caso de los *clusters* la asignación de recursos se realiza gracias a un recurso administrador centralizado y todos los nodos trabajan cooperativamente juntos como un único recurso. En el caso de las GRID, cada nodo tiene su propio recurso administrador y no es un objetivo el proporcionar una vista de sistema único.

La computación necesaria para abordar los requerimientos de los proyectos científicos está siendo cada vez más elevada, debido a que los problemas son cada vez más complejos. Las aplicaciones en las que se plasman estos problemas son cada vez más complicadas y demandantes de potencia de cálculo, así como grandes consumidoras de datos.

Muchos de estos proyectos, además de requerir una gran capacidad de potencial computacional propiamente dicho, y del almacenamiento de inmensas cantidades de datos, requieren la colaboración de numerosos grupos de científicos. Éstos, así como los recursos de los que disponen, pueden pertenecer a una misma área en la que realizan sus investigaciones, pero encontrarse distribuidos geográficamente.

La utilización eficiente de estos recursos es todo un reto, ya que estos recursos dispersos deben ser operados conjuntamente como sistemas. Además se necesita que estén disponibles la mayor parte del tiempo, y que además den un rendimiento elevado debido a los requerimientos de las aplicaciones.

2.4 Fundamentos

Los fundamentos de la computación GRID son básicamente tres:

1. **Compartición** de recursos a gran escala: la idea fundamental es poder compartir una serie de recursos entre los posibles usuarios, de manera que sea igual de sencillo poder acceder a una infraestructura local que a una localizada en cualquier parte del mundo. Es

más, el uso debe poder llevarse a cabo utilizando muchos recursos distintos, localizándose éstos distribuidos geográficamente. Entre los recursos que podemos compartir se pueden distinguir:

Computadoras: es el recurso básico al que queremos acceder, proporciona potencia de cálculo para realizar las operaciones necesarias.

Redes: de comunicaciones que permitirán interconectar el resto de recursos para que la ejecución de las aplicaciones sea posible.

Instrumentos: generalmente de carácter científico, que son necesarios en algunas aplicaciones. Ejemplos de estos instrumentos pueden ser cabinas de visualización, microscopios electrónicos, radio-telescopios, etc.

Datos: que deban ser compartidos por una comunidad para lograr sus objetivos, como datos para simulaciones nucleares, variables meteorológicas para la obtención de las predicciones, o cadenas de ADN para ser procesadas. Además estos datos pueden tener requerimientos de privacidad, como pueden ser los expedientes médicos y datos asociados a pacientes que pueden ser accedidos por una comunidad médica para analizarlos.

2. **Organización de recursos distribuidos de varias organizaciones:** estos recursos pueden pertenecer a organizaciones distintas que tengan sus propios administradores locales, aplicando las políticas que son necesarias y adecuadas para cada organización. Por lo tanto el control sobre los mismos es limitado y no centralizado.
3. **Recursos heterogéneos:** ya que distintas organizaciones pueden disponer de multitud de elementos para compartir que soporten diferentes protocolos, herramientas, etc. La GRID debe tener en cuenta este punto para proveer una infraestructura común que pueda interoperar estos recursos.

2.5 Software de soporte

GLOBUS

El desarrollo de Internet fue posible gracias a la existencia de un estándar como el protocolo *TCP/IP*; el *WWW* al éxito del lenguaje *HTML* y el protocolo *http*. La tecnología GRID cuenta con un estándar: Globus.

El Globus Toolkit es un producto del Proyecto Globus, una alianza formada por varias instituciones de investigación y desarrollo que concentra su actividad en las tecnologías centrales para GRID computacionales. El proyecto Globus abrió una unidad de consultoría comercial que se llama Univa para desarrollar aplicaciones comerciales basadas en el Globus Toolkit. Adicionalmente, el Dr. Foster se encuentra involucrado en el Consorcio Globus, una iniciativa creada el 24 de enero de 2005 que cuenta con el apoyo de HP, IBM, Intel y SUN. Este consorcio se dedicará a desarrollar el Globus Toolkit desde el punto de vista comercial. Será un bloque de construcción basado en estándares abiertos que generará un camino para implantaciones comerciales de GRID.

En 1995 se celebró el congreso SuperComputing'95, donde se demostró que era posible el ejecutar aplicaciones distribuidas de varias áreas científicas entre 17 centros de Estados

Unidos conectados por una red de alta velocidad de 155 Mbps. Este experimento se denominó I-Way, y fue el punto de partida de varios proyectos en diferentes áreas, con un denominador común que era la compartición de recursos distribuidos de computación. A partir de este momento el libro “*The GRID: Blueprint for a New Computing Instructure*”, editado por Ian Foster y Carl Kesselman supuso el primer paso para establecer las primeras ideas claras sobre cómo debía llevarse a cabo esta nueva tecnología.

A partir de estas ideas se desarrolló el Globus Toolkit, un proyecto *open-source* desarrollado en el Argonne Nacional Laboratory dirigido por Ian Foster en colaboración con el grupo de Carl Kesselman de la Universidad de Southern California. Globus da los medios básicos de la tecnología para construir un GRID computacional, y se ha convertido gracias a su evolución y adopción por la comunidad científica como el estándar en la tecnología GRID.

La arquitectura abierta de Globus se estructura en capas, siguiendo los estándares propuestos por Foster y Kesselman e incluye servicios software para la monitorización de recursos, descubrimiento y gestión, además de servicios de seguridad y de gestión de archivos. Se incluye software organizado en áreas como seguridad, infraestructura de la información, detección de fallos, portabilidad, etc. Está empaquetado como una serie de componentes que pueden usarse ya sea independientemente o conjuntamente para desarrollar aplicaciones.

Las capas que componen la arquitectura de Globus son: la capa básica denominada **Estructura** controla los recursos locales; por encima de ésta, **Conectividad** incluye los aspectos de comunicación y seguridad; la capa **Recursos** el acceso y control de recursos de modo individual, mientras **Colectivo** se encarga de la coordinación de recursos múltiples, y en particular de servicios distribuidos.

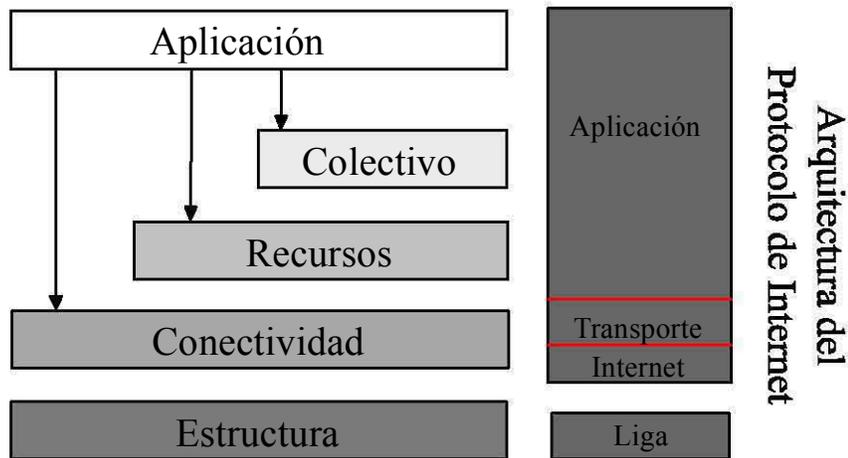


Figura 2.3 Arquitectura en capas de Globus

Uno de los componentes clave de Globus es el protocolo GRID Security Infraestructure (GSI), que permite una autenticación única del usuario para todos los recursos mediante certificación digital basada en PKI y X.509. La capa **Conectividad** incluye además los protocolos habituales de Internet (IP, DNS, etc); los otros tres protocolos clave corresponden a la de **Recursos**: *GRID Resource Allocation Management* (GRAM), *GRID Resource Information Protocol* (GRIS), y el *GRID File Transfer Protocol* (GRIDFTP).

El Toolkit de Globus fue concebido para quitar los obstáculos que impide la colaboración entre diferentes organizaciones o instituciones. Sus servicios centrales (*core services*),

interfaces y protocolos permiten a los usuarios acceder a los recursos remotos como si estuvieran presentes dentro de su propia sala de máquinas, a la vez que preservan el control local sobre quién y cuándo puede usar los recursos.

The Globus Hourglass

Los elementos del Globus Toolkit no asumen que los entornos locales están adaptados para soportarlo. En principio fue diseñado e implementado para adaptar a los muchos y varios entornos locales bajo el cual pueda ejecutarse.

Globus ofrece una serie de servicios básicos para establecer una infraestructura básica. Éstos son luego usados para construir soluciones específicas de cada dominio, de alto nivel. Para ello tres principios de diseño clave que se siguen son mantener los costos de participación bajos, mantener el control local cuando quiera que sea posible, y proveer soporte para la adaptación del Toolkit a las necesidades específicas de cada sitio y proyecto.



Figura 2.4 Globus

La base del Globus *Hourglass* representa los miles de recursos sobre los que los servicios de Globus son construidos. Por ejemplo los sistemas operativos locales de las diversas máquinas sobre las que corre Globus, además de los muchos tipos de redes, los sistemas de planificación, los sistemas de archivos, etc. El medio es comprimido por los servicios básicos (*core services*) que Globus ofrece, y la parte de arriba representa los servicios de alto nivel que Globus ofrece, así como las aplicaciones escritas en Globus, como se muestra en la figura 2.4.

Las implementaciones locales de los servicios de Globus para un sistema operativo en particular liberan tanto a los servicios básicos como a los servicios de alto nivel de conocer cuestiones específicas del sistema operativo. Sólo los servicios locales necesitan conocer cuál es el sistema operativo sobre el que funciona, liberando a los programadores de aplicaciones de conocer estos detalles.

Una visión más detallada de los servicios ofrecidos por Globus puede ser representado en la figura 2.5:

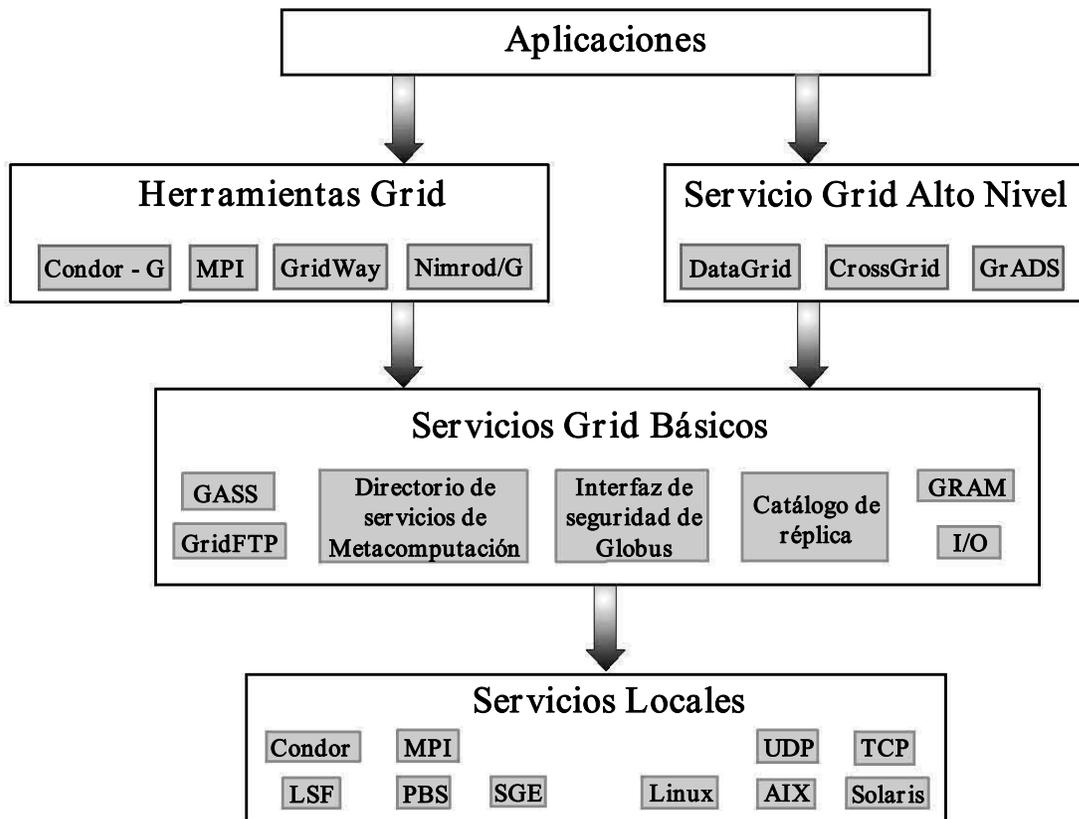


Figura 2.5 Modelo en capas

Los *core services* incluyen cuatro protocolos básicos que se ajustan al diseño de la arquitectura propuesta y proveen la funcionalidad básica necesaria. En la capa de conectividad se ofrece uno de los servicios más necesarios, el de seguridad con GSI (*GRID Security Infrastructure*).

Por encima de éste y en la capa de recursos se ofrecen los servicios de control de recursos: GRAM (*GRID Resource Allocation Management*), los servicios de información: GRIP (*GRID Resource Information Protocol*), y de transferencia de datos: GRIDFTP (*GRID File Transfer Protocol*).

Seguridad en Globus: GSI

La seguridad es uno de los pilares fundamentales sobre los que se tiene que establecer una GRID y todos los servicios superiores, y tenerse muy en cuenta ya que la filosofía de compartición de recursos tiene muchos problemas asociados.

Los recursos pueden ser valiosos y por lo tanto se debe permitir el acceso sólo cuando se desee y a las entidades que se desee. También los problemas a resolver pueden ser sensibles a la privacidad, así como los datos que estos problemas requieren o generan.

Por lo general, los recursos están situados en distintos dominios administrativos, por lo cual cada uno tiene sus propias políticas de acceso, procedimientos, mecanismos de seguridad, etc.

Para tener en cuenta estos puntos, la implementación de los servicios de seguridad tiene que estar públicamente disponible; lo que está relacionado con que los protocolos sean estándar, bien probados, y comprendidos por la comunidad.

El conjunto de recursos puede ser elevado, dinámico e impredecible, por lo que no se está hablando de simple autorización y autenticación en un entorno cliente/servidor, sino que se necesita un método de delegación de identificaciones de servicio a servicio, por el cual sea posible que se pueda autorizar y autenticar en nuestro nombre bajo unas circunstancias determinadas y un entorno controlado.

Los requerimientos en cuanto a seguridad son varios, dependiendo desde el punto de vista que lo tomemos.

Para un usuario las características que se deben satisfacer son las siguientes:

- Que sea fácil de usar, con comandos sencillos y con pocos pasos manuales.
- Que sólo se requiera un único *login*.
- Que se utilice un modelo basado en la credibilidad del usuario.
- Que sea posible la utilización de *proxies* y agentes.

Desde el punto de vista del dueño de los recursos, los puntos más importantes a satisfacer serían:

- Poder especificar políticas de control de acceso local, que sean flexibles y versátiles para poder plasmar las necesidades de cada entorno
- Que sea posible el auditar y controlar, para hacer frente a posibles brechas en la seguridad de sus sistemas y tener un histórico de accesos.
- Integración con los sistemas de seguridad locales como AFS y Kerberos.
- Protección contra otros recursos comprometidos

Desde el punto de vista del desarrollador que necesita utilizar estos servicios:

- Disponibilidad de un API/SDK con métodos para autenticar, hacer flexible la protección de mensajes, proveer métodos para la delegación, etc.
 - A través de llamadas directas (GSS-api)
 - Integrada en los servicios de nivel superior: *GlobusIO*, *Condor-G*, *mpich-g2*.

Para ello en Globus se integran todas estas necesidades en lo que se llama el *Globus Security Infrastructure* (GSI). Es un conjunto de protocolos y APIs que hacen frente a las necesidades planteadas, y que están basados en una serie de protocolos estándar, extendiéndolos. En particular se utilizan protocolos de autenticación de clave pública para la autenticación de usuarios y recursos, y para la protección de mensajes.

Para la identificación se usan certificados X.509, que se basa en una infraestructura de clave pública. La infraestructura de clave pública (PKI) consiste en que permite conocer que una determinada clave pertenece a un determinado usuario (o entidad). Está basada en los principios de cifrado asimétrico, en la que cada entidad posee un par de claves, una pública y otra privada, de manera que los datos cifrados con una clave sólo pueden ser descifrados con la otra. La clave pública es conocida, mientras que la privada es secreta y sólo debe ser conocida por la entidad.

En la PKI, la clave pública es encapsulada en un certificado X.509 que identifica inequívocamente a la entidad para la que se emitió. Además del nombre único del dueño del certificado (Distinguished Names o DN), éstos contienen otra información como la fecha de caducidad, el emisor del certificado, y una firma (hash) emitida por este último. El emisor del certificado será una Autoridad de Certificación (CA), que a través de su clave privada firmará este certificado dando fe de su autenticidad, y que siempre se puede comprobar a través de la clave pública de esta autoridad.

Uno de los requerimientos principales para que sea posible la utilización en sistemas GRID, es que sea posible hacer un único *login* y también la delegación de identificaciones. Para ello es necesaria la utilización de identificaciones *proxy* (*proxy credentials*), que permitan que una entidad A otorgue a otra entidad B, el derecho de ser autorizada con otros como si fuera la primera entidad A. De esta manera un *proxy* podrá ser creado a partir de un certificado normal X.509 o a partir de otro *proxy*, con el propósito de aplicar *proxies* restringidos según las necesidades. Además se podrá aplicar delegación, que consiste en la posibilidad de creación remota de un *proxy* credencial de segundo nivel. Los *proxies* pueden ser restringidos, en el sentido de que se pueden delegar un subconjunto de derechos, lo que es deseable para tener restricciones de más fino grano. Después cada servicio en el que se quiere autenticar puede decidir si permite o no.

GSI además soporta el Standard GSSAPI, implementando interfaces que soportan autenticación, delegación, integridad y confidencialidad de mensajes, lo que permite soportar aplicaciones como SSH, o GRIDFTP.

Una cuestión aparte es la autorización, que no es llevada a cabo directamente por GSI. Es decir una vez que una entidad se ha identificado como tal y sabemos inequívocamente quién es, el problema de autorización se basa en permitir o no el uso de estos recursos a esta entidad autenticada. En la idea más simple, se mantiene en cada recurso que se quiere autenticar un archivo *GRIDmapfile* que contiene los *Distinguished Names* de aquellos certificados que se aceptan, por lo que cada vez que se accede a un servicio de este recurso se comprueba si el *proxy* con el que se está accediendo está incluido, permitiendo el acceso si así es o rechazándolo en caso contrario.

En la práctica el usuario de una GRID, deberá disponer de un certificado de usuario X.509 expedido por una autoridad de certificación aceptada por esta GRID. Con este certificado le será posible contactar con los servicios de recepción de trabajos, pero realmente lo que se realiza en el cliente es obtener un *proxy* a partir del certificado de usuario que será el que realmente se envíe en la petición del servicio.

Básicamente en el servidor se obtiene información del usuario a partir de este *proxy*, su DN que es el que se utiliza para ver si el usuario está finalmente autorizado o no a utilizar el servicio. Si este servicio necesita la comunicación con otros se utilizará la delegación de este *proxy* como método para autenticarse en estos nuevos servicios.

Carencias del Middleware Globus

El middleware, como su nombre en inglés lo indica, es el intermediario que se encarga de gestionar la GRID como tal. Se le puede considerar como el sistema operativo de la GRID. Este software permite compartir recursos localizados en diferentes dominios de administración, con diferentes políticas de seguridad y gestión de recursos.

El *middleware* es una capa de software por encima del sistema operativo de cada nodo y por debajo de las aplicaciones de usuario. Su objetivo es presentar a los usuarios una única imagen de sistema (Single System Image SSI), basada en ocultar la naturaleza distribuida y heterogénea de los recursos disponibles y presentarlos a usuarios y aplicaciones como un sistema de cómputo unificado.

Una de las carencias más importantes de las que adolece el Toolkit es de un servicio de *brokering* automático de recursos dentro de los que sería una entidad como un súper-planificador (superscheduler) que se encargue de tomar las preferencias de los trabajos de los usuarios y buscar los recursos necesarios para satisfacerlas, de manera automática.

Esta entidad está supuestamente prevista en la arquitectura de Globus, de manera que traslada especificaciones generales de recursos en especificaciones más concretas, pero realmente sólo se plantea como una entidad abstracta y no existe implementación real de la misma con lo que la gestión de recursos es incompleta. El usuario debe localizar los recursos y seleccionarlos manualmente, para después lanzar los comandos adecuados sobre éstos. Globus se encarga de enviar el trabajo a estos recursos y ejecutarlos, pero además el usuario es encargado de comprobar que la ejecución es correcta y progresa. En realidad aunque existen los métodos básicos para hacer casi todos los pasos en cuanto a la planificación, las tareas llevadas a cabo automáticamente se reducen al envío del trabajo propiamente dicho.

Si la ejecución no es correcta, el usuario debe realizar las acciones adecuadas para cancelar el comando y volver a reenviar el trabajo, repitiendo varios pasos de localización de recursos compatibles (el filtrado de autorizaciones, la definición de los requerimientos de las aplicaciones, el filtrado de requerimientos mínimos, la recolecta de información dinámica, la selección y planificación, y la reserva). Así mismo no se realizan las tareas de limpieza necesarias, por lo que los datos del trabajo quedan en el lugar de ejecución remoto.

Además aunque existen los métodos básicos para la transmisión de archivos de grandes volúmenes de datos, Globus no contaba al principio con métodos para el tratamiento de réplicas de archivos. Esto significa que el usuario debe actuar como planificador también en los datos dentro de lo que serían las tareas preparatorias. Si un trabajo necesita unos datos específicos se deben poder acceder desde el recurso donde finalmente se ejecute el mismo, lo que puede requerir su copia local. De esta manera es lógicamente más eficiente el acceder a datos que están más cercanos ya que será más eficiente. Esta planificación teniendo en cuenta los datos tampoco existe y el usuario debe realizarla manualmente, cuando realmente debe ser una tarea del *superscheduler* o *broker*.

Arquitectura de la gestión de recursos

Una de las partes más importantes es la gestión de recursos de un sistema GRID, la cual se encarga de asignar los recursos necesarios a los trabajos que son enviados.

La gestión de recursos realizada en Globus está relacionada con los problemas de localizar y asignar recursos computacionales, y con la autenticación, creación de procesos, y otras actividades requeridas para preparar los recursos para su uso. No se ocupa de cuestiones como la planificación (scheduling), descomposición, asignación y ordenación de ejecución de tareas, o la gestión de otros recursos como memoria, discos y redes.

Los sistemas de meta-computación como son las GRID, introducen cinco desafíos de gestión de recursos que son tenidos en cuenta por Globus: autonomía de los recursos, heterogeneidad, extensión de políticas, co-asignación, y control en línea.

El problema de la autonomía de los sitios, hace referencia al hecho de que los recursos son típicamente gestionados por diferentes organizaciones, en diferentes dominios administrativos.

Relacionado con el punto anterior está la heterogeneidad de los sistemas, que está representada por el hecho de que los diferentes sitios pueden usar gestores locales diferentes como Condor, NQW, CODINE, EASY, LSF, PBS.

La política de extensibilidad surge debido a que las aplicaciones sobre GRID pueden venir de muchos recursos administrativos, cada uno con sus propios requerimientos. Una solución correcta debe soportar el desarrollo de nuevas estructuras de gestión específicas de cada dominio, sin requerir cambios en el código instalado en los sitios participantes.

La *co-allocation* (co-asignación) hace referencia al requerimiento de aplicaciones que necesitan recursos simultáneamente en varios sitios. La autonomía de los sitios y la posibilidad de fallo durante la asignación introduce la necesidad de mecanismos especializados para la asignación de múltiples recursos, iniciar la computación en estos recursos, monitorizar y gestionar estas operaciones de cómputo.

El control *online* hace referencia a la negociación requerida para adaptar los requerimientos de la aplicación a la disponibilidad de recursos, particularmente cuando los requerimientos y las características de los recursos cambian durante la ejecución. Por ejemplo, una aplicación que necesita simular una entidad puede preferir renderizar (acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla) a baja resolución, si la alternativa es que no se pueda simular en absoluto. Los mecanismos gestores de recursos deben soportar esta negociación.

No existe ningún gestor de recursos que haga frente a los cinco problemas recién planteados. Algunos sistemas de colas *batch* soportan co-asignación, pero no autonomía, políticas de extensibilidad, o control *online*. Condor soporta la autonomía de los sitios, pero no co-asignación ni control *online*. Legion y Gallop tienen control *online* y extensibilidad de políticas, pero no tienen en cuenta el substrato heterogéneo o los problemas de co-asignación.

En la arquitectura de gestión de recursos de Globus, se tiene en cuenta el problema de la autonomía de los sitios y la heterogeneidad de los diferentes recursos introduciendo las entidades que se denomina *resource managers* (gestores de recursos), para proveer una interfaz bien definida de las diferentes herramientas de gestión de recursos, políticas, y mecanismos de seguridad que cada sitio utilice.

Para el control *online* y la extensibilidad de políticas se define un lenguaje de especificación de recursos, que soporta negociación entre diferentes componentes de la arquitectura de gestión de recursos.

Coordinación, Estándares Internacionales, Relación con la Industria y OGSA

El Global GRID Forum (GGF) es el foro internacional sobre tecnología GRID. Reunido por primera vez en Ámsterdam en marzo de 2001, gracias al esfuerzo de Ian Foster y de Fabrizio Gagliardi) a ambos lados del Atlántico, se realiza una reunión semestral para coordinar los esfuerzos y posibilitar la estandarización. El interés de las compañías comerciales por la tecnología GRID ha ido creciendo, al igual que su presencia como patrocinadores en el GGF: Sun, HP/Compaq, Microsoft, SGI e IBM.

La posible convergencia entre la tecnología de Web Services en el campo de *e-Business*, y la próxima versión de Globus denominada OGSA (Open GRID Services Architecture), ha contado con una fuerte participación de IBM que ha apostado por la tecnología GRID. El Globus Toolkit, apuesta por servicios GRID basados en Web Services potencialmente transitorios, con interfases y comportamientos bien definidos que incluyen creación, asignación de nombres y referenciado, autorización, notificación, tiempo de vida, etc. Esta evolución representa una gran oportunidad para lograr una amplia aceptación y difusión de la tecnología GRID, que puede extenderse, al igual que lo hizo el WWW, desde su ámbito original en el área de la computación científica, al de las aplicaciones comerciales.

2.6 Gestión de recursos

La gestión de recursos GRID (*GRID Resource Management*) se define como el proceso de identificar requerimientos, relacionar a los recursos con las aplicaciones, asignar estos recursos, planificar y monitorizar estos recursos GRID a través del tiempo para correr las aplicaciones sobre la GRID lo más eficientemente posible.

Las aplicaciones GRID compiten por recursos que son muy distintos en su naturaleza, incluyendo procesadores, datos, instrumentos científicos, redes y otros servicios. Los recursos son heterogéneos en sus distintas clases, pero incluso se puede encontrar heterogeneidad en la misma clase y tipo de recurso. Por ejemplo, no hay dos *clusters* de computadoras que tengan el mismo software y configuración exactamente.

El trabajo inicial en la gestión de recursos GRID ha estado orientado a hacer frente a la heterogeneidad, con una serie de protocolos estándar de gestión de recursos y mecanismos estándar para expresar recursos y requerimientos de los trabajos.

Además el hecho de carecer de datos disponibles sobre el sistema actual y las necesidades de los usuarios, así como las de los dueños de los recursos y los administradores del sistema, hacen más complicada esta situación.

En la actualidad las GRID se están convirtiendo en una tecnología muy conocida, en la que se ha invertido una considerable cantidad de tiempo y de dinero, y que se está aprovechando en entornos de producción.

Los desafíos iniciales de la computación GRID (cómo correr un trabajo, cómo transferir grandes archivos, cómo gestionar múltiples cuentas de usuario en sistemas diferentes) se han ido resolviendo en un primer orden, de manera que usuarios e investigadores pueden hacer ahora frente a las cuestiones que permiten hacer más eficiente el uso de los recursos.

El uso de las GRID se ha vuelto cotidiano en muchos ambientes, el uso de buenas herramientas está lejos de ser universal a causa de las muchas cuestiones abiertas en el tema:

- Múltiples capas de *schedulers* (planificadores): La gestión de recursos GRID tiene que contar con la participación de varias capas de planificadores. En el nivel más alto se encuentran los planificadores en el nivel GRID, que pueden tener una visión general de los recursos pero que están muy lejos de los recursos donde la aplicación correrá finalmente. Estos primeros planificadores tienen por objeto selección a grandes rasgos dónde se ejecutará la aplicación, por ejemplo en qué *cluster*, y la reserva de los mismos en el caso de que esté disponible.

En el nivel más bajo se encuentran los gestores de recursos locales que gestionan un recurso o grupo de recursos, como los gestores de colas (PBS, LSF, etc.) que gestionan los *clusters* locales.

Normalmente la interacción entre estas dos capas no es fácil ni directa, por lo que surgen muchas cuestiones que tienen que ser resueltas. Entre estas dos capas además se encuentran algunas más intermedias, por ejemplo planificadores que gestionan una serie de recursos específicos a un proyecto. En cada nivel adicional, más gente y software tiene que ser tenido en cuenta.

- Falta de control sobre los recursos: Los planificadores GRID no son sistemas de gestión locales, por lo que al más alto nivel un planificador GRID puede no tener acceso directo o control sobre los recursos. La mayoría de las veces los trabajos son enviados desde estos gestores de alto nivel a una serie de recursos con los mismos permisos que un usuario de ese sistema tendría. Esta falta de control es uno de los desafíos que se tienen que hacer frente.
- Recursos compartidos y varianza: Relacionado con la falta de control está la falta de accesos dedicados a los recursos. Muchos recursos son compartidos entre varios usuarios y proyectos. Esta compartición resulta en un alto grado de varianza y no es predecible en la capacidad de los recursos disponibles para su uso. La naturaleza heterogénea de los recursos también juega un papel en la variada capacidad. La situación es más complicada por el hecho de que aplicaciones GRID a veces requieren la asignación de múltiples recursos, necesitando una estructura en la cual los recursos deben ser coordinados a través de varios dominios administrativos.
- Conflictos en los objetivos de rendimiento: Cuando se utilizan los recursos GRID para correr las aplicaciones de un usuario, pueden surgir conflictos entre los intereses de éstos y los de los dueños de los recursos. Desde optimizar el rendimiento de una simple aplicación para un costo específico a obtener el mejor rendimiento del sistema (*throughput*) o minimizar el tiempo de respuesta. Además muchos recursos tienen políticas locales que deben ser tomadas en cuenta.

Cuestiones como la de quién debe hacer la planificación exactamente, siguen abiertas: ¿Cuánto de este proceso debe ser hecho por el sistema y cuánto por los usuarios? ¿Cuáles son las reglas para cada uno? Muchas de las investigaciones están orientadas al entendimiento y la gestión de estas diversas políticas desde la perspectiva de ambos, el proveedor de recursos y el consumidor.

El surgimiento de arquitecturas orientadas a servicios, el creciente interés en soportar un amplio espectro de aplicaciones comerciales, y la natural evolución de funcionalidad, están conjuntamente permitiendo avances significativos en las capacidades de la gestión de recursos. Mientras los entornos GRID están primariamente orientados hacia los servicios *best-effort*, se espera que esta situación se torne significativamente diferente en los próximos años, con capacidades de suministro de recursos extremo-a-extremo (*end-to-end*).

Planificación de trabajos

La planificación (*scheduling*) dentro de la gestión de recursos, es definida como el proceso de la toma de decisiones que tienen que ver con recursos sobre múltiples dominios administrativos. Este proceso puede incluir la búsqueda de múltiples dominios

administrativos para usar una única máquina o planificar un único trabajo para usar múltiples recursos en un único sitio, o en varios de ellos.

Dentro de las especificaciones necesarias se define un trabajo (*job*) como la instancia de una aplicación que va a ejecutarse en la GRID y que necesita un recurso.

Un recurso se define como cualquier cosa que puede ser planificada en el tiempo, por ejemplo una máquina, espacio de disco, una red con capacidades de calidad de servicio (QoS), y más.

Las diferencias entre un planificador GRID de alto nivel, y un gestor local de recursos, es que este último hace la gestión de recursos en un único sitio, y representa la capa más baja en lo que se refiere a instancias de planificadores.

La principal característica de los planificadores GRID es que no poseen y por lo tanto no pueden gestionar directamente los recursos de un sitio, por lo que no tienen control sobre éstos. Deben tomar decisiones basadas en un sistema base *best-effort*, y entonces mandar el trabajo a los recursos seleccionados, generalmente como el usuario. Además el planificador GRID no tiene control sobre el conjunto completo de trabajos enviados al recurso, o incluso no conoce sobre los trabajos que están siendo enviados a los recursos que está usando, por lo que las decisiones que compensa los accesos de un trabajo por los de otro pueden no ser realizables en un sentido global. Esta falta de propiedad y control es la causa de muchos de los problemas a resolver en esta área.

Una idea sería utilizar la ejecución especulativa de múltiples instancias, enviando el trabajo a múltiples recursos y cuando uno empieza a ejecutarse, entonces cancelar el resto. Sin embargo no es la aproximación común de los sistemas que existen en la actualidad.

Interacción con los sistemas de Información

Los planificadores GRID tienen que tomar decisiones basándose en la información de la que disponen y la que se les provee. Por una parte pueden tener información del trabajo a ejecutar, sus características y sus requerimientos, así como información que pueda ayudar a una mejor planificación basándose en las preferencias del usuario. Esta información está normalmente y en su mayor parte provista por el usuario mismo.

Por otra parte se necesita información de la GRID misma, sobre su estado actual y sus capacidades, para hacer una planificación eficiente. En general estos planificadores obtienen información de un sistema general de información de la GRID (*GRID Information System* o GIS), que de hecho obtiene la información de los recursos individuales. Muchos planificadores GRID asumen que tienen disponible el 100 % de la información necesaria, y con un nivel de detalle extremadamente fino y que es siempre correcta. En realidad la experiencia con estos sistemas dicta que esta situación ideal está lejos de ser la realidad, y que por lo general sólo se puede disponer de la información de más alto nivel de estos sistemas.

Etapas de la planificación sobre GRID

Existen tres fases dentro del proceso que seguiría un planificador GRID: descubrimiento de recursos (*resource discovery*), que genera una lista de potenciales recursos, recolección de la información (*information gathering*) sobre estos recursos y selección del mejor o los mejores; y la ejecución del trabajo (*job execution*) que incluye el envío de los archivos necesarios y el posterior borrado.

Fase 1: Descubrimiento de recursos

En esta primera fase se determina qué recursos están disponibles a un determinado trabajo de un usuario. La fase de descubrimiento de recursos incluye la selección de un conjunto de recursos que serán investigados en más detalle en la fase 2, de recolección de información sobre los mismos.

Al final de esta primera fase se dispondrá de un conjunto de recursos que habrán pasado unos mínimos requerimientos del trabajo. Esta fase se puede subdividir en tres pasos: filtrado de autorizaciones, definición de los requerimientos del trabajo, y filtrado para satisfacer los requerimientos mínimos del trabajo.

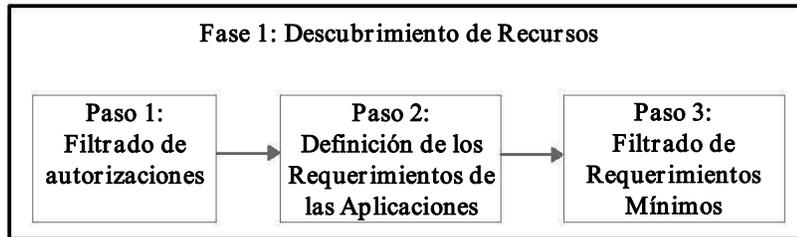


Figura 2.6 Fase 1: Descubrimiento de Recursos

Paso 1: Filtrado de autorizaciones

El primer paso consiste en determinar el grupo de recursos en los cuales el usuario está autorizado. Esto es lógico ya que en un sistema amplio como la GRID es posible que los usuarios, y por lo tanto sus trabajos, estén autorizados únicamente en un subconjunto de recursos cuyos dueños se tiene algún tipo de relación o acuerdo, y no a todos ellos. En este sentido, la computación sobre la GRID no es diferente de enviar un trabajo a un sitio remoto: si no se tiene autorización en ese sitio el trabajo no correrá.

Al final de esta fase se dispondrá de una lista de recursos que el usuario está autorizado a utilizar. Una de las diferencias fundamentales es el número de recursos que se están analizando, ya que debido a la extensión del sistema y del número de recursos, los sistemas de información GIS pueden dar información sobre muchos recursos, incluso aquellos en los que en principio no se está autorizado, por lo que una primera fase de filtrado es necesaria.

Paso 2: Definición de los requerimientos de las aplicaciones

Para proceder en la búsqueda de recursos, el usuario debe ser capaz de especificar un mínimo conjunto de requerimientos para filtrar más convenientemente el conjunto de recursos a los que tiene acceso en el siguiente Paso 3.

El conjunto de posibles requerimientos del trabajo puede ser muy amplio y puede variar significativamente entre trabajos. Puede incluir campos estáticos, en el sentido de que los recursos van a mantenerlos sobre el tiempo, como el sistema operativo o el hardware para el cual ha sido diseñada la aplicación. También puede incluir detalles más dinámicos, como por ejemplo los requerimientos de memoria, la conectividad en cuanto a ancho de banda, o el espacio temporal disponible. Cuantos más detalles son incluidos más eficiente será la planificación y la relación de los recursos, pero también debe tenerse en cuenta que cuantas más restricciones se impongan menor será el conjunto de posibles recursos donde ejecutar, y en un caso extremo este conjunto puede ser igual a vacío.

La información que se usa para corresponder los requerimientos del trabajo con aquellos ofrecidos por los recursos puede hacerse disponible por varios métodos, aunque se sigue trabajando en métodos que automáticamente recojan estos datos. Además puede darse la situación de que muchos de aquellos requerimientos como el tiempo de ejecución mínimo de la tarea para una salida satisfactoria sea manufacturado por el usuario para compensar las decisiones que pudiera tomar cualquier planificador.

Paso 3: Filtrado de requerimientos mínimos

Una vez que se dispone de un conjunto de recursos, y de los requerimientos de los trabajos, el siguiente paso lógico es hacer un filtrado de aquellos recursos que no satisfacen los requerimientos de las aplicaciones. Normalmente en esta fase se deberían filtrar aquellos recursos que no satisfacen los requerimientos estáticos de las aplicaciones, es decir, aquellos valores que no suelen cambiar en un corto espacio de tiempo como el sistema operativo instalado o la configuración hardware. De esta manera se puede hacer un filtrado rápido cuando el número de recursos y trabajos es elevado, para facilitar las siguientes fases.

Fase 2: Selección del sistema

Dado un número determinado de recursos que se han obtenido después de la fase 1, esta siguiente fase decidirá dónde será finalmente planificado el trabajo dependiendo de las políticas asociadas en cada fase.

Para ello se requiere, por una parte información más detallada de los recursos y del estado del sistema, y por otra realizar la decisión final de ejecución del trabajo. Aunque son dos pasos independientes, dependen el uno del otro de modo que la decisión está guiada por la información disponible.

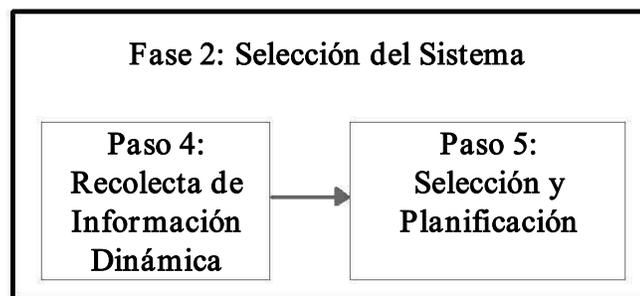


Figura 2.7 Fase 2: Selección del Sistema

Paso 4: Recolecta de Información dinámica

Para hacer más efectiva la selección que se llevará a cabo en el siguiente paso, puede ser necesario la recolecta de información dinámica sobre los recursos. Esta información puede variar con respecto a la aplicación que se está planificando.

Sobre recursos computacionales, la información disponible variará de sitio a sitio. Normalmente la información básica disponible proviene del planificador de recursos local. Además, información proveniente de elementos de monitorización en estos recursos, puede ser disponible para efectuar decisiones más adecuadas. Las políticas de autorización local de cada recurso a través de múltiples dominios de ejecución deben tenerse en cuenta también. Cada vez es más común el que los administradores de los recursos especifiquen el porcentaje

de estos recursos, en términos de capacidad, tiempo o cualquier otra métrica, para ser considerados.

Un punto importante es la escalabilidad de los sistemas usados para la recolección dinámica de la información, ya que el número de recursos puede crecer considerablemente en sistemas GRID. El número de recursos no sólo influye en que se producen más consultas, sino que si algún recurso no está disponible el sistema debe decidir qué hacer con los datos dinámicos que no están disponibles. La aproximación más fácil es desestimar estos recursos, aunque en sistemas más grandes alguna otra aproximación puede ser analizada.

También se puede utilizar sistemas de monitorización y predicción, donde se ha estado realizando mucho trabajo, para utilizar información que sea de utilizad en la toma de decisiones, aunque sin embargo en la actualidad no se están aplicando a sistemas en producción.

Paso 5: Selección y Planificación.

Con la información obtenida en el paso anterior, el siguiente paso es decidir en qué recursos o grupos de recursos se va a seleccionar para la ejecución del trabajo. En esta fase se deben aplicar las políticas de planificación disponibles para seleccionar el que se considere el mejor recurso, dada la información del paso anterior y también un posible conjunto de preferencias establecido por el usuario. La inclusión de este tipo de preferencias por parte del usuario puede influir en el proceso de selección de recursos, dando información extra sobre lo que se considera en cada momento el mejor recurso para el usuario y aquel que él prefiere.

Sin embargo la decisión final estará en manos del planificador, que deberá hacer un balance adecuado de todas las posibilidades y aplicar sus políticas de planificación para ello.

Fase 3: Ejecución del trabajo

La última fase tiene que ver con todos aquellos pasos que hacen posible la ejecución de un trabajo, finalmente en los recursos seleccionados en la fase anterior. Para ello se deben realizar toda una serie de pasos intermedios, muchos de los cuales son casi estándares y describen una vía uniforme entre los recursos.

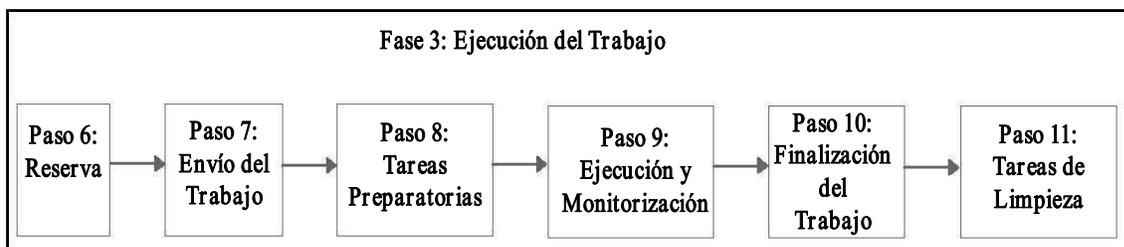


Figura 2.8 Fase 3: Ejecución del Trabajo

Paso 6: Reserva (opcional)

Para llevar a cabo el mejor uso posible de los recursos, a veces es conveniente realizar una reserva de los recursos, principalmente de aquellos que son más difíciles de obtener o que tienen una mayor demanda por parte de los trabajos.

Dependiendo del tipo de recurso, si es computacional, de red, o incluso algún instrumento, puede ser más o menos fácil de realizar. Además el uso de reservas puede estar relacionado con algún tipo de contabilidad de los recursos, para poder hacer cumplir otras cuestiones como son las SLA's entre aquellos que ofrecen recursos y los clientes que los usan.

Para que la reserva sea efectiva, los recursos deben dar el soporte básico a ésta, pero sucede que en la actualidad es bastante difícil encontrar estos servicios de reserva de manera nativa, aunque la necesidad se va haciendo más determinante debido a que los convenios de nivel de servicio son cada vez más comunes.

Paso 7: Envío del trabajo

El envío efectivo del trabajo a los recursos se realiza en este paso a través de protocolos en los que se está trabajando y que pretenden convertirse en estándar de protocolos abiertos.

En esta fase se debe contactar el gestor local de cada recurso para enviar el trabajo o sub-trabajo que le corresponda, coordinando todos aquellos que estén involucrados. Una parte importante es la gestión y la tolerancia a fallos que se den en los recursos en esta fase, que puede requerir el reenvío del trabajo a los mismos u otros recursos.

Paso 8: Tareas preparatorias.

Este paso tiene mucho que ver con el anterior, ya que el envío del trabajo puede requerir de otros pasos para que se realice correctamente. Ejemplos de estos pasos pueden ser el transferir aquellos archivos necesarios para la computación, hacer algún tipo de actualización necesario, requerir la reserva realizada anteriormente, etc.

Esta fase también puede requerir la obtención de identificaciones de usuario específicos para correr un trabajo determinado, ya que aunque estos servicios son ofrecidos a más bajo nivel, puede que no satisfagan automáticamente las necesidades del trabajo. Por ejemplo si un trabajo necesita acceder a otro recurso con nuevas o diferentes identificaciones también se deberían obtener durante esta fase.

Paso 9: Ejecución y monitorización.

Una vez que se han realizado todas las acciones necesarias y el envío del trabajo se ha realizado correctamente, el siguiente paso lógico es comenzar la ejecución del mismo. La ejecución misma está fuera del control de los planificadores de alto nivel, y más relacionada con los planificadores locales, así como la monitorización en primera instancia de lo que está acaeciendo durante la ejecución del trabajo. Sin embargo pueden darse otras posibilidades como que el usuario pueda monitorizar el proceso de ejecución de su aplicación, y posiblemente cambiar su opinión sobre dónde o cómo se está ejecutando.

Además puede darse la posibilidad de detectar automáticamente que el trabajo no se está ejecutando conforme a lo esperado, o que se está produciendo algún tipo de interbloqueo, por lo que el planificador podría decidir re-enviar el trabajo o cancelarlo en última instancia. Esto último es considerablemente más difícil en un sistema GRID que en cualquier máquina paralela, ya que la falta de control sobre los recursos puede que haga fallar los pasos anteriormente descritos para una nueva ejecución. Esto es debido a que nuevos trabajos pueden haberse enviado desde el anterior, ocupando los recursos.

Paso 10: Finalización del trabajo.

Cuando el trabajo ha finalizado el usuario puede requerir el obtener información sobre la ejecución del mismo, y también la obtención de la información y los archivos que el trabajo haya producido. La notificación misma puede ser síncrona o asíncrona, así como la obtención de estos archivos de salida.

Paso 11: Tareas de Limpieza.

Asociado con el paso anterior, las tareas de limpieza se encargarían de obtener los archivos útiles y de limpiar todos aquellos restos de la ejecución del trabajo cuando ya no fueran necesarios.

2.7. Aplicaciones de la GRID

2.7.1 Introducción

Una GRID computacional es una infraestructura de software y hardware capaz de proveer altos niveles de procesamiento de manera omnipresente, confiable, consistente y económica. Este concepto se adoptó rápidamente en el mundo de la ciencia. Las supercomputadoras de los centros espaciales y de los observatorios mundiales tomaban mucho tiempo para procesar los altos volúmenes de datos que se estaban obteniendo y el uso de GRID era la respuesta.

Los proyectos científicos actuales requieren grandes recursos de computación conectados a través de la red, dando origen a lo que se ha denominado e-Ciencia. La tecnología GRID propone los servicios básicos para gestionar el uso compartido, eficiente y seguro de estos recursos. El software Globus es el componente básico de los proyectos multidisciplinarios en marcha en USA y en Europa. Los *testbeds* (mini-GRID) de los proyectos DataGRID y CrossGRID, incluyen un gran número de centros europeos conectados a través de la red Geant de alta velocidad. Esta futura infraestructura GRID común impulsará el desarrollo de los proyectos de e-Ciencia.

Los proyectos científicos abordan objetivos cada vez más ambiciosos que requieren la resolución de problemas computacionales complejos, tanto por el volumen de los cálculos a realizar como por el tamaño y complejidad de las bases de datos utilizadas. Del mismo modo, los equipos científicos son en muchos casos colaboraciones internacionales, con miembros distribuidos por todo el planeta. Áreas científicas como la Física de Altas Energías, Ciencias del Espacio, Genómica y Proteómica, o Meteorología, basan su desarrollo en estos proyectos. El término e-Ciencia se utiliza para denominar la vertiente computacional de estos proyectos. La organización de los correspondientes recursos de computación, es un desafío.

Transformando este desafío en parte de la solución, la tecnología GRID propone agregar y compartir recursos de computación distribuidos entre diferentes organizaciones e instituciones, a través de redes de alta velocidad, de modo que el acceso a los mismos por parte de los científicos para sus necesidades de cálculo sea tan sencillo, flexible y fiable como el uso de la corriente eléctrica para satisfacer sus necesidades de energía.

En la actualidad la GRID ha dejado de estar únicamente asociado a la investigación científica debido a las mejoras en los anchos de banda, las plataformas y las capacidades de las computadoras personales y servidores. Novedosas aplicaciones empresariales están

disponibles en el mercado para aprovechar al máximo las potencialidades de la infraestructura tanto de software como de hardware de la empresa utilizando GRID *computing*. La capacidad de implantación de las GRID es una realidad y existen experiencias exitosas que han logrado reducir los tiempos de procesamiento radicalmente. Sin embargo, GRID está naciendo y falta mucho camino por recorrer y nuevas versiones de aplicaciones empresariales que puedan aprovechar las bondades de este paradigma computacional en su máxima expresión.

La GRID requiere la utilización de un software capaz de dividir y distribuir piezas de un programa entre cientos o miles de computadoras al mismo tiempo. Este software es usualmente utilizado para agilizar el procesamiento de altos volúmenes de datos. Los grandes *clusters* o las supercomputadoras pueden ser parte de una GRID al igual que una PC de escritorio o un sistema heredado de la empresa. La GRID puede ser vista como una forma de procesamiento distribuido utilizando las redes como medio de comunicación. Éste puede implementarse utilizando las bondades de un departamento, una corporación o las del mundo entero.

Actualmente, hay 5 aplicaciones generales para la Computación GRID:

- Supercomputación distribuida
- Sistemas distribuidos en tiempo real
- Servicios puntuales
- Proceso intensivo de datos
- Entornos virtuales de colaboración (Teleinmersión)

Supercomputación distribuida

Aplicaciones cuyas necesidades es *imposible* satisfacer en un único nodo. Estas aplicaciones se distinguen porque satisfacen necesidades *puntuales* e *intensivas* de computación.

Por ejemplo: Simulaciones, cálculos numéricos, minería de datos, análisis de grandes volúmenes de datos.

Sistemas distribuidos en tiempo real

Sistemas que generan un flujo de datos a alta velocidad que debe ser analizado en tiempo real.

Por ejemplo: Medicina electrónica, experimentos de física de alta energía, control remoto de un recurso no-trivial (un microscopio, equipo médico, etc.)

Servicios puntuales

Acceso puntual a un recurso que no vale la pena tener en nuestra organización. Similar a las dos aplicaciones anteriores. La diferencia es que no se refiere a potencia computacional, y no tiene que ser en tiempo real.

Por ejemplo: acceso a hardware específico para ciertos tipos de análisis (químico, biológico, etc)

Proceso intensivo de datos

Aplicaciones que trabajan con grandes volúmenes de datos, y que es imposible *almacenar* en un único nodo. Los datos se distribuyen a lo largo de la GRID.

Teleinmersión

Utilizar la potencia computacional y la naturaleza distribuida de la GRID para crear entornos virtuales 3D distribuidos.

2.7.2 La evolución de los recursos locales de computación

A pesar de que la ley de Moore se cumple año tras año, y una computadora personal, actual tiene una potencia superior a la de un supercomputador *Cray* de hace 10 años, la resolución de muchos problemas científicos computacionales sigue siendo un desafío, tanto en cuanto a técnicas como a recursos. La solución está dada por dos acciones conjuntas: agregar y compartir.

En paralelo al desarrollo de supercomputadores con un número creciente de procesadores, una de las soluciones con más éxito en los últimos años ha sido la construcción de *clusters* de computadoras individuales: “*granjas*” con nodos interconectados mediante una red local de alta velocidad. En particular, para reducir el costo se emplea “*commodity hardware*”: *PCs* con procesadores *Intel Pentium* o similares y conectados por *Fast* o *Gigabit ethernet*.

La computación en Física de Partículas es un ejemplo claro de esta evolución: los experimentos del anterior acelerador LEP del CERN (el Centro Europeo de Física de Partículas, Ginebra) pasaron de usar computadoras *Cray* e *IBM* a granjas con decenas de procesadores *RISC* y sistema operativo *Unix* en la primera parte de la década de los 90, y “*fábricas*” de cientos de *PCs* con sistema operativo *Linux* a finales de la misma. Un ejemplo de las posibilidades de compartir estos recursos lo proporcionó la simulación en un solo fin de semana utilizando los recursos completos del CERN de más de cinco millones de colisiones e^+e^- , para mejorar los resultados de la búsqueda del bosón de Higgs del experimento DELPHI: una tarea de meses en las máquinas *RISC* dedicadas de la colaboración. El próximo acelerador LHC, que entrará en funcionamiento en el año 2007, requerirá el almacenamiento y procesamiento de varios *Petabytes* (millones de *Gigabytes*) de datos cada año. Los recursos necesarios se estiman en el equivalente de una “*fábrica*” de 200,000 *PCs*, un orden de magnitud por encima de las mayores supercomputadoras actuales, y con claras dificultades técnicas, operativas y de financiación. La solución viene de la mano del problema: agregar y compartir los recursos proporcionados por las instituciones participantes en los experimentos del LHC, distribuidas geográficamente por todo el mundo.

2.7.3 Computación a través de la red

La tecnología GRID nace dentro de la comunidad de supercomputación y está basada en las dos acciones, agregar y compartir, junto a la evolución de la red.

La evolución de las redes académicas locales posibilita agregar la capacidad de todos los equipos en funcionamiento de una institución: el sistema Condor desarrollado por Miron Livny para la plataforma Linux nació para aprovechar el tiempo de inactividad de estas

máquinas, permitiendo agregar y compartir estos recursos, transformándose en un sistema completo de gestión de los mismos.

El siguiente paso fue la extensión a recursos conectados a través de Internet: la red Entropía agregó en dos años 30,000 computadoras, logrando por ejemplo calcular el mayor número primo conocido. Del mismo modo, el sistema SETI funciona en más de medio millón de PCs analizando los datos del radio telescopio de Arecibo en la búsqueda de señales de inteligencia extraterrestre.

La computación utilizando recursos distribuidos a través de la red no es una cuestión trivial. Las aplicaciones del denominado entorno de *High Throughput Computing* o HCT (Cómputo de Alto Rendimiento de Procesamiento), son más sencillas de adaptar a una ejecución distribuida ya que permiten dividir un trabajo completo en múltiples partes con un tiempo considerable de ejecución, cada una de las cuales puede procesarse independientemente. Por ejemplo, la simulación de un millón de colisiones en Física de Partículas puede realizarse de modo distribuido en cien máquinas cada una de las cuales realiza de manera independiente la simulación de 10,000 sucesos. Por el contrario, en un entorno de *High Performance Computing HPC* (Cómputo de Alto Desempeño) se requiere una respuesta inmediata global del sistema para que la aplicación progrese y el cálculo no pueda distribuirse de manera independiente. Estas fronteras se difuminan más en cuanto la capacidad de la red en tiempo de respuesta y de transferencia de datos mejora en comparación con el tiempo de ejecución de cada paso en la aplicación, cuando ésta contiene componentes paralelizables. Un caso típico es el entrenamiento de una red neuronal distribuida, donde el error en cada paso sucesivo se puede calcular de modo independiente distribuyendo la muestra de entrenamiento entre los diferentes nodos, y el error global se obtiene al final de cada paso agregando los resultados de cada nodo. Las técnicas de paralelismo aplicables a máquinas multiprocesadoras, como las basadas en el uso de *MPI* pasan a poder aplicarse a nodos conectados en red.

2.7.4 Aplicaciones generales de la GRID

Observatorio

Los datos recogidos en el observatorio de Arecibo (Puerto Rico) son procesados por ordenadores de todo el mundo.

Análisis de Mamografías

Ya existen los análisis de Mamografías en GRID. Con lo cual se reduce el tiempo de 1h a 40 segundos. El Diagnóstico es más completo. Se cuenta con 100 nodos de computación en GRID. Se accede a una BD distribuida con imágenes de mamografías. Esta aplicación se realizó en el Gobierno UK, por IBM, Miranda Solutions.

Sistema de Apoyo a Decisiones para inundaciones.

Se procesa en tiempo real de datos obtenidos en el lugar de la catástrofe, o en lugares susceptibles a grandes inundaciones.

Permite realizar la Monitorización, Predicción y Simulación de las inundaciones. Esta aplicación forma parte del Proyecto Crossgrid (Europa).

Aprendizaje Colaborativo Apoyado por Computadora

El Aprendizaje Colaborativo Apoyado por Computadora (CSCL – *Computer Supported Collaborative Learning*), es un campo de estudio dedicado a la investigación de tecnología educativa que se centra en el uso de las tecnologías de información y comunicaciones (TIC) como herramientas de mediación para la aplicación de métodos colaborativos de aprendizaje.

La GRID es una infraestructura que puede dar un soporte adecuado a las aplicaciones CSCL facilitando su despliegue y potenciando su rendimiento. La gran escala de la GRID podría permitir la participación en la aplicación CSCL de un elevado número de participantes individuales o en grupo. La amplia distribución geográfica de los recursos que forman parte de la GRID facilitaría la colaboración entre usuarios muy distantes entre sí. También la naturaleza heterogénea de los recursos que son compartidos en la GRID podría permitir a los usuarios participar empleando no sólo computadoras, sino también otros dispositivos como PDAs o pizarrones electrónicos además de recursos de software como simuladores. Por otra parte, el acceso de calidad proporcionado por la infraestructura GRID a través de un servicio garantizado o de mayor esfuerzo puede mejorar el rendimiento de la aplicación cuando ésta es desplegada *ad hoc*.

Este último aspecto es especialmente importante para el dominio CSCL dado que el éxito de la colaboración entre usuarios, y por tanto del aprendizaje como objetivo último, sólo es posible si el rendimiento de la aplicación es bueno.

Para lograr que las aplicaciones puedan aprovechar el asignador de recursos, elemento encargado de decidir cómo ha de llevarse a cabo el despliegue de una aplicación de forma que el rendimiento de la misma sea el adecuado de acuerdo con unos criterios predefinidos. Sin embargo, el desarrollo de un asignador de recursos GRID es un problema muy dependiente del dominio de aplicación.

El soporte proporcionado por una infraestructura GRID puede ser de gran utilidad para las aplicaciones CSCL. Sin embargo, para que dicho soporte sea posible, es necesario desarrollar un asignador de recursos.

La tecnología del cómputo GRID en los Negocios

La tecnología de cómputo GRID tiene como antecedente cercano la revolución de Linux y los códigos abiertos en el hermetismo del software, pero que ahora intenta dar un giro radical en el competido universo del hardware. El ejemplo cumbre de este fenómeno lo otorga la iniciativa humanitaria *World Community GRID*, con la participación de 48,734 miembros y 70,214 dispositivos unidos para aprovechar al máximo las computadoras de particulares y empresas, y aplicarlas a la resolución de los problemas de la salud en el planeta.

La potencia computacional subutilizada de las computadoras del mundo se aplicará a la investigación, con el fin de desentrañar los códigos genéticos que subyacen a enfermedades tales como el Sida y el VIH, Alzheimer y el cáncer, y además para mejorar el pronóstico de desastres naturales y respaldar estudios que sirvan para preservar el abastecimiento de alimentos y agua potable para el mundo. El *World Community GRID* se construye a partir del tiempo de computación donado por miles de empleados de esta compañía, así como otros PCs, portátiles y servidores de usuarios de computación de todo el mundo. Por medio de un soporte a millones de consultas por día, se administran los datos de computadoras que trabajan en forma concertada y en aras de un objetivo común.

Durante muchos años las empresas han intentado encontrar fórmulas de competitividad, aunque por lo general se quedan estancadas en los caminos de la negligencia operacional, de los altos costos en plataformas y de los tiempos perdidos en el manejo de procesos.

La GRID *computing*, un tema que comprende un conjunto de tecnologías que permite utilizar al máximo la infraestructura tecnológica, y en particular los recursos ociosos de las computadoras de escritorio y otras fuentes de cómputo subutilizadas (servidores de archivo, impresión, portátiles). En otras palabras, se forma un pool virtual de recursos que se puede comportar como una "gran computadora distribuida".

1. Un computador central distribuye un proceso entre los equipos conectados a una red.
2. El sistema aprovecha la capacidad de todos los computadores conectados. Cuando no están siendo utilizados al 100% por el usuario reciben tareas de la máquina central.
3. Todos los recursos disponibles en la red son aprovechados, independientemente de su arquitectura y sistema operativo.
4. Si el usuario está utilizando sólo una parte de la capacidad del computador, el resto se aprovecha para el cálculo.
5. Si el usuario está utilizando todos los recursos o el equipo se cae, la tarea se reasigna a otra máquina disponible.
6. La aplicación es accesible desde cualquier equipo de la red. Con una interfaz cómoda y sencilla se puede añadir al sistema tantas máquinas como se desee.
7. La conexión directa entre las PCs evita la sobrecarga de la máquina central.
8. La tecnología GRID es fácilmente escalable, creciendo según las necesidades de cada empresa.

Para plantear una infraestructura paralela que integre y unifique las existentes, se aprovecha al máximo las redes de área local (LAN) intra-empresariales, lo que permite agregar recursos para realizar tareas rápidas de cálculo muy grandes como un cierre de nómina o simulaciones de yacimientos de petróleo. Todo, sin la necesidad de una supercomputadora o un gran servidor central.

La única condición para usarla es que los procesos del negocio que emplea esta posibilidad tienen que ser susceptibles de ser "paralelizables", es decir, que se puedan descomponer en muchos procesos pequeños independientes.

Sun Microsystems grafica a GRID *computing* como muchas máquinas que funcionan como si fueran una sola; por lo general pequeñas, de uno o dos procesadores, con una gran capacidad de interconexión en red para que las aplicaciones sean susceptibles de ser "paralelizables" y puedan correr por la red al mismo tiempo en varias máquinas. Además, con una intervención humana mínima que se traduzca simplemente en enviar y recibir una tarea.

Esta premisa califica para todo tipo de empresas, sin importar el tamaño o el sector en que se desenvuelvan, aunque es más susceptible de permearse en las mentes de las gerencias financieras y de Tecnologías de Información (TI) de las pequeñas y medianas empresas. Por su naturaleza, las pequeñas empresas buscan disminuir costos operativos, la adquisición de plataformas sencillas y mejoras en la complejidad de su administración.

Los expertos en la materia reconocen además que cualquier compañía puede atreverse a adoptar alguna de las variantes de GRID *computing*, sin tener que llenar requisitos específicos en materia tecnológica o de presupuesto. Es sólo actitud hacia la tecnología, más que requisitos rígidos.

Oracle se refiere a GRID *computing* como una gran matriz, una gran malla para todo tipo de entidades, en donde cada estación es un recurso que tiene la capacidad necesaria para poderse compartir y comunicar con los demás puntos.

GRID *computing* tiene el potencial de mejorar el tiempo de los resultados para nuevos productos o servicios, aumentando la productividad y la colaboración, logrando mejorar el tiempo para la obtención de resultados en la organización. Además debe ser efectivamente escalable para favorecer distintos tipos de demandas. La idea es que se puedan crear infraestructuras flexibles.

Utilizando este nuevo concepto, las organizaciones pueden unir capacidades tecnológicas dispares para integrar un único sistema. Y esto permite compartir, manejar y acceder virtualmente a dispositivos a lo largo y ancho de cualquier empresa.

De ahí que los objetivos de la computación GRID se enmarquen en varias áreas: primero, en la optimización de los tiempos para la producción de nuevos productos y servicios, porque al incrementar la productividad y la colaboración se acortan los lapsos del resultado. Ya sea que estos logros impliquen poner un nuevo producto en el mercado más rápidamente, como resolver un problema comercial complejo más pronto o trazar una estrategia y completar un análisis de datos.

En segundo lugar, promueve la flexibilidad operacional para que las organizaciones puedan aunar no sólo recursos tecnológicos dispares, sino también gente y aptitudes. De alguna manera se comparte el acceso a la información en sectores como el farmacéutico o bancario.

En tercer término, se puede escalar para satisfacer demandas variables del negocio mediante la creación de infraestructuras operativas flexibles y resistentes. La intención es que las empresas puedan abordar rápidas fluctuaciones en la demanda con acceso instantáneo a recursos de computación y datos para responder a las necesidades del negocio. Al ganar la habilidad de resolver problemas comerciales complejos más rápidamente, las organizaciones pueden moverse con velocidad extrema y obtener ventajas competitivas.

El cuarto objetivo consiste en incrementar la productividad otorgando a los usuarios finales acceso a los recursos de computación, datos y almacenamiento cuando lo requieran. La tecnología GRID puede impulsar a las empresas a capacitar mejor a sus empleados para efectuar sus tareas, resolver problemas comerciales con facilidad y moverse entre etapas de diseño de productos.

Y por último, las firmas comerciales pueden aprovechar inversiones de calidad existentes para maximizar la utilización productiva de los recursos existentes, y así reducir costos operativos. Al asegurar esta utilización óptima de las capacidades de computación, las tecnologías GRID sirven para evitar los errores de costos excesivos en infraestructura.

Los proveedores de GRID están encajando esta nueva propuesta para masificarla en el mercado. Mediante diferentes estrategias buscan demostrar que con la GRID se puede compartir, manejar y acceder virtualmente a todos los dispositivos de la empresa. Y aprovechar al máximo los recursos existentes.

Sun, Oracle e IBM utilizan este recurso tecnológico. En el caso particular de Sun, tiene más de 100 clientes a nivel mundial usando productos de su tecnología "N1 GRID Service Provisioning System" y "N1 Containers". Entre ellos resaltan proyectos del líder automotriz

DaimlerChrysler y de Deutsche Bahn, la más grande ferrocarrilera de Alemania, los cuales han seleccionado las soluciones N1 para ayudar a manejar sus servicios y simplificar las operaciones de sus centros de datos; todo con el objetivo de reducir costos y complejidad en sus ambientes de computación en red.

1. Integrar sistemas y dispositivos heterogéneos. *GRID computing* proporciona un conjunto de capacidades de integración horizontal que dirige de forma efectiva los recursos de toda la empresa, e incluso extiende la solución entre múltiples organizaciones.
2. Mejora el costo efectivo de los entornos operativos. A través de la visualización de la consolidación, reservas, cooperación y gestión de recursos a través de las funciones heterogéneas de tecnologías de la información, la GRID ayuda a simplificar los entornos operativos y su gestión reduciendo la administración de su supervisión. Además, como consecuencia de fomentar la utilización eficiente de los recursos, las empresas pueden construir una infraestructura de tecnologías de la información de costos efectivos que asegure la completa utilización de las inversiones en tecnología existente.
3. Creación de competencias virtuales seguras y flexibles. Las tecnologías GRID adoptan las nociones de flexibilidad, libertad de elección de estándares abiertos. También son capaces de descubrir dinámicamente y ajustarse a los entornos cambiantes y fluctuantes de las tecnologías de la información. Por esta razón facilita el establecimiento, reestablecimiento y cambios de los parámetros que requieren los negocios respecto a la seguridad y asignación de recursos.
4. Incrementa la capacidad de recursos para responder a las fluctuaciones de la demanda. Permite a las organizaciones de tecnologías de la información agregar recursos distribuidos y explotar una capacidad no utilizada. Las GRID incrementan de forma importante la cantidad de recursos computacionales y de datos disponibles. Ayuda a crear infraestructuras de tecnologías de la información que pueden responder rápidamente a oleadas inesperadas en el tráfico y uso de los recursos.
5. Aumenta la fiabilidad de la infraestructura sacando ventaja de los recursos de la GRID como una alternativa ante la recuperación de los desastres tradicionales.

Se pueden encontrar diferentes versiones de GRID en el mercado. Las variantes básicamente se apoyan en varios pilares fundamentales: servicios profesionales de clase mundial para grandes implantaciones; o tecnologías de virtualización dinámica de recursos de infraestructura, que brindan la posibilidad de ver a un conjunto de recursos ubicados en varias computadoras como si estuviera en un solo procesador.

Por otro lado, para algunas tareas se puede necesitar cierta capacidad de cómputo más grande que la que tiene la mayor máquina disponible de una red. Con las tecnologías tradicionales se está supeditado a lo que rinda el equipo más robusto de un centro de cómputo. Pero con las tecnologías GRID se puede realizar una tarea mucho más amplia: por ejemplo, implantar un cierre de una nómina en varias computadoras ociosas de noche, lejos de hacer más lento al proceso por descentralizar, la operación se puede llevar a cabo en forma más expedita que con una sola computadora limitada.

Además, en el universo de GRID aparece otro tipo de virtualización de recursos: el modelo de *Utility computing*. Este nuevo concepto, se define como comprar capacidad de cómputo, tal cual se hace con el servicio de electricidad, agua o gas, y se paga de acuerdo a lo que se consume. Esta iniciativa busca que las empresas paguen por el consumo o lo que se conoce como "computación sobre demanda".

Y por eso existen distintas modalidades como el *outsourcing*, que consiste en entregar toda la operación de activos de TI a un tercero. O el modelo de *hosting*, donde otros ponen y administran los recursos y sólo se obtiene el procesamiento, las entradas y las salidas. Aparece también la posibilidad de *colocation*, que deposita los recursos en un tercero que se ocupa de la seguridad, del aire acondicionado, de la temperatura y de la humedad, por nombrar algunos. Y la alternativa de ASP, que es cuando una compañía tiene una aplicación que interesa y a la cual se le paga por utilizar esa aplicación.

Con GRID *computing* se puede construir una infraestructura de tecnología de información donde todos tienen acceso, no se pierde la continuidad del proceso y los usuarios no se ven afectados. Con todos estos elementos se mejora la competitividad de una empresa pues se ofrecen mejores servicios, con altos estándares de calidad, y se le asegura a los clientes continuidad en el servicio.

Todos estos beneficios se pueden alcanzar porque la GRID logra "democratizar" los servicios de tecnología de la información. Con la GRID lo que se busca es que el usuario pague por usar las cosas o las herramientas y no necesariamente costee unas estructuras que sean ociosas o que puedan tener un lucro cesante.

Beneficios de las empresas

1. **Mejorar el tiempo de resultados para nuevos productos y servicios al aumentar la productividad y la colaboración.** Estos resultados pueden incluir llevar un nuevo producto al mercado más rápidamente, resolver un complejo problema de negocio más pronto, realizar un análisis de datos en profundidad para lanzar un nuevo servicio, la GRID ofrece a las compañías acelerar el tiempo y tomar la delantera al mercado.
2. **Facilitar la colaboración y promover la flexibilidad de las operaciones.** Las ofertas de GRID pueden agrupar no sólo recursos tecnológicos distintos, sino también a la gente, ayudando a que el personal pueda compartir, acceder y gestionar la información, permitiendo que las organizaciones sean capaces de mejorar la colaboración en todas las unidades geográficas y de negocio para dar soporte a estrategias de globalización.
3. **Efectivamente escalable para favorecer distintos tipos de demandas.** Con la GRID los negocios pueden crear infraestructuras flexibles y elásticas que gestionen rápidamente las fluctuaciones de las demandas de los clientes permitiendo el acceso instantáneo a los recursos de computación y datos que respondan a las necesidades del negocio. La capacidad de resolver problemas complejos de negocio más rápido significa que las organizaciones pueden moverse más rápidamente y ganar ventajas competitivas en el mercado.
4. **Incrementar la productividad.** Proporcionar a los usuarios finales un acceso sin restricciones a los recursos informáticos, de datos y de almacenamiento que necesitan. La tecnología GRID puede ayudar a las compañías a mejorar la gestión de recursos humanos.
5. **Mantener las inversiones de capital.** Maximizar la productividad, la utilización eficaz de los recursos existentes es una de las claves para minimizar los costos. Ayudando a asegurar la utilización óptima de las capacidades informáticas, la tecnología GRID puede ayudar a las empresas a evitar las dificultades comunes de sobreprovisionamiento o incurrir en el exceso de costos para infraestructura. Debido a que la GRID *computing* se basa en estándares abiertos para crear una infraestructura única y unificada, la tecnología libera a las organizaciones de TI del peso de administrar sistemas no integrados, reduciendo de esta forma la supervisión.

La GRID son recursos de TI, que pueden ser discos para almacenar información; servidores de base de datos; o servidores de aplicaciones. La GRID está conformada por múltiples y heterogéneos recursos, que pueden ir desde un disco de almacenamiento de datos, hasta servidores sofisticados que hacen funciones muy específicas; y todos hacen parte de la red compartiendo información y comunicándose entre sí.

Por otra parte, en la actualidad la simplificación de los entornos operativos se rige bajo un concepto básico y es que cada aplicación empresarial necesita de un administrador. Cuando están en una GRID sólo se necesita un solo administrador que controla todos los entornos. Se logra centralizar el control de una consola única simplificando la administración, porque hay una persona que tiene la visibilidad de todos los recursos y tiene la capacidad de gobernarlos y de darles seguimiento. Esto facilita la operación, porque hay funciones o servicios automáticos, por ejemplo, el balanceo de carga, que con la GRID se vuelve automático.

Así la continuidad del negocio no se ve afectada, porque si un nodo falla, automáticamente los otros puntos entran a suplir el defecto, brindando total garantía de que la producción no padecerá y los usuarios recibirán su producto.

La apropiación de la GRID *computing* o computación bajo demanda significará, para el sector empresarial de los países de la región, pensar en la infraestructura no solamente como un tema de hardware y redes, sino mucho más que eso: cuestionar las restricciones actuales de integración, desempeño, disponibilidad, seguridad, portabilidad y algunas más relacionadas con los servicios que al final presta la tecnología. Validar estas funciones implica pensar en la labor del software no solamente como una herramienta de especificación, desarrollo de pruebas e implantación, sino como un instrumento de uso en donde el valor para quien la emplea está en su facilidad y capacidad de adopción.

El beneficio de asociarse a este enfoque se traduce entonces en un mejor precio total de propiedad, pues se están atacando de entrada temas como el costo de integración, ensayo y puesta a punto, así como el mantenimiento de soluciones informáticas, la capacidad de modificación y la automatización. Al final, el mayor beneficio será para el negocio que apropie este modelo, pues le generará por encima de todo una mayor competitividad.

Bases de Datos

Oracle comenzó desde el año 2003 a generar capacidades GRID para sus productos. Con el lanzamiento de su base de datos y el *Application Server 10g*, la empresa ofrece nuevos recursos de gestión, almacenamiento y *cluster* que proporcionan eficiencia inmediata camino al GRID *computing* empresarial. El enfoque a largo plazo de la organización es lograr que las empresas puedan utilizar aplicaciones orientadas al servicio dentro de sus arquitecturas manejadas por eventos con el fin de aprovechar al máximo las capacidades de computación distribuida de una GRID.

Oracle tiene tres pilares fundamentales para la implantación de una GRID empresarial en la *suite 10g*. La base de datos Oracle Data Base, el Oracle Application Server y el Enterprise Manager GRID Control.

Oracle Data Base 10g, virtualiza la forma como la base de datos utiliza los componentes de hardware (de almacenamiento o servidores). Automáticamente provee almacenamiento en *clusters* a las diferentes bases de datos que se encuentren en el GRID. Como proveedor de datos, el manejador puede agrupar, virtualizar y distribuir datos a los usuarios y aplicaciones que estén en la GRID.

El Oracle *Application Server 10g*, está diseñado para ejecutar aplicaciones empresariales en grupos de servidores y sistemas de almacenamiento de bajo costo para proveer alto rendimiento, escalabilidad y alta disponibilidad. Esta plataforma automatiza el proceso de instalación, configuración y aprovisionamiento de software; genera las identidades y accesos de los usuarios a través de múltiples recursos y aplicaciones de la GRID; afina y monitorea estas aplicaciones para lograr un óptimo rendimiento; administra automáticamente las cargas de trabajo; se encarga de proveer alta disponibilidad y escalabilidad a medida que limita el tiempo de procesamiento ocioso de los equipos; y finalmente trabaja directamente con Oracle GRID Control para facilitar la administración y monitoreo de la GRID.

El Enterprise Manager GRID Control, está diseñado para administrar GRID desde una interfaz Web de usuario que utiliza protocolos como el http, https, ssl o JDBC para establecer conectividad TCP/IP con cada agente de monitoreo que reside en cada equipo de la GRID. Puede escalar hasta 100 nodos sin costo adicional, realiza tareas de aprovisionamiento automático de parches y actualizaciones, puede ser accedido desde cualquier navegador dentro o fuera de la organización por el administrador, permite disminuir costos de administración de un gran número de sistemas y puede hasta monitorear la experiencia de cada usuario final para garantizar los niveles de calidad de servicio.

La solución GRID de Oracle es independiente del hardware por lo que se puede incluir dentro del grupo de servidores de un sistema informático de la misma forma que se pueden agregar servidores Unix, Windows o Linux. Adicionalmente, la infraestructura Oracle 10g permite integrar a la GRID aplicaciones y datos que no residan en sistemas propietarios de la organización.

Dell, EMC, Intel y Oracle pretenden desarrollar un modelo estándar de diseño e implantación de infraestructuras utilizando GRID *computing* en un proyecto que tiene por nombre Proyecto Mega GRID.

Los primeros retos del mundo de TI que este proyecto busca solucionar son: la reducción de costos de la infraestructura de TI, mejoras a la calidad de servicio, garantizar la continuidad del negocio, ofrecer escalabilidad y administración centralizada y una acelerada implantación basada en mejores prácticas pre-evaluadas en los laboratorios.

En el proyecto Mega GRID, Oracle provee la base de datos, el servidor de aplicaciones, el software de administración y las herramientas diseñadas para correr ambientes GRID. Dell por su parte facilita toda la infraestructura de servidores, EMC hace lo propio con su sistema de almacenamiento en red (NAS y SAN) e Intel suministra los procesadores de tipo empresarial.

Poder de Procesamiento

Sun Microsystems enfoca su visión de GRID en el poder de procesamiento. Sun GRID provee una estructura para conectar y consolidar recursos de cómputo en un poderoso GRID, para maximizar su utilización, flexibilizar la administración del poder computacional a través de diferentes plataformas y lograr un incremento en la productividad.

El Service Provisioning System, automatiza el aprovisionamiento de aplicaciones permitiendo al administrador distribuir, actualizar y configurar los servicios de red a través de un *Web browser*. El ejecutivo comentó que con este sistema se puede ajustar el centro de datos a las cargas de trabajo en un momento determinado para mantener los niveles de calidad de servicios. Se define cuáles son los parámetros de calidad de cada servicio y cuando esa

cifra se excede automáticamente el Provisioning System toma acción y administra la carga de trabajo incorporando o desincorporando recursos.

Por su parte SAP ha venido trabajando en incorporar funcionalidades de computación adaptable a su plataforma SAP Netweaver. Estas funcionalidades establecen un puente entre el hardware y el software, ayudan a los clientes a consolidar y racionalizar sus infraestructuras de tecnología de la información, con el propósito de administrar mejor y más eficientemente la demanda concurrente de recursos de hardware.

Las soluciones de software son desligadas de sus infraestructuras de hardware, posibilitando una más flexible utilización de los recursos de hardware, a través de la sincronización de las necesidades de desempeño de las aplicaciones de negocios, en toda la empresa. Gracias a las funcionalidades de computación adaptable, los clientes pueden operar cualquier aplicación sobre cualquier servidor, en cualquier momento, para mejorar el desempeño del sistema a unos muy bajos costos de operación.

Sun Microsystems apoya a organizaciones que están trabajando en la región para promover el uso de la GRID de manera científica en las universidades. La GRID Latinoamericano y del Caribe (GRIDLAC), es una institución cuyo objetivo es compartir recursos computacionales creando ambientes y facilidades para resolver problemas de la región y promover el intercambio de experiencias e integración. Algunas aplicaciones que el GRIDLAC ha implementado están relacionadas con *E-learning* o educación a distancia y bioinformática.

En el área de *E-learning* el fin es crear una plataforma que permita ir más allá de lo tradicional: páginas Web, correo electrónico, chat y video conferencias, para crear un verdadero ambiente de cooperación interactiva. Con la llegada de Internet 2, las perspectivas de la educación a distancia bajo GRID presentan un panorama antes inimaginable. En materia de bioinformática hay un buen trecho recorrido y en la actualidad hospedan el meta portal bioinformático que integra distintos servidores y portales relacionados, así como *La Red Bioinformática Iberoamericana*.

Una solución GRID trae consigo optimización en el manejo de la infraestructura tecnológica, la capacidad de agregar recursos de procesamiento sin importar su localización geográfica, resolver situaciones en las que un servidor se encuentre trabajando al máximo rendimiento mientras otros mantienen ciclos ociosos, disminución en los tiempos de procesamiento y por supuesto la consolidación de los recursos y la consecuente reducción de costos de tecnología.

Los empresarios se ven beneficiados de la GRID porque pueden acelerar sus procesos de negocio, incrementar su productividad, nivelar los costos de inversión y agilizar la adaptación de nuevos modelos de negocio resultantes de cambios en el mercado.

Acelerador De Partículas

Una red equivalente a 20,000 ordenadores personales de 1 GHz. Cuando el Large Hadron Collider (LHC), un potentísimo acelerador de partículas, entre en servicio en 2007 en Europa, será necesario un increíble sistema informático para poder analizar y manipular la enorme producción de datos que generarán sus experimentos, equivalentes a 20 millones de CDs al año. Los científicos ya han empezado a poner las bases para crear esta infraestructura informática, esencial para el éxito del proyecto.

La opción elegida no consiste en construir una supercomputadora masiva, sino una red que permitirá conectar a numerosos computadores para que trabajen juntos. La citada red ha sido bautizada como European GRID, y estará formada por redes más pequeñas dedicadas a la física de partículas.

Una de estas redes menores es la que va a preparar el Particle Physics and Astronomy Research Council británico. Se llamará GRIDPP2 y su potencia combinada será equivalente a la que tendría la segunda supercomputadora más grande del planeta, tras el Japan Earth Simulator, que como su nombre indica es una supercomputadora japonesa pensada para simulaciones geofísicas. La GRIDPP2 estará conectada a la European GRID y colaborará en el procesamiento de datos del LHC, propiedad del CERN.

La GRIDPP2 ha estado funcionando en forma de prototipo dentro de una red de pocas computadoras, situadas en 10 localidades británicas. El objetivo era probar el buen funcionamiento de los programas que deben gobernarla, aprovechando al máximo la capacidad de las máquinas conectadas. Este software también debe hacerse cargo de la seguridad del intercambio de datos y de la subdivisión de éstos para distribuirlos hacia cada una de las computadoras disponibles.

Gracias al modo en que se comparten los recursos, los científicos podrán realizar simulaciones y análisis en una fracción del tiempo que se necesitaría si sólo se empleara una sola máquina. Este trabajo se puede hacer también en supercomputadoras, pero dado que éstas se fabrican a medida y son muy caras, además de estar muy solicitadas, la GRID es la mejor opción para programas como el LHC.

Entre los primeros proyectos GRID surge la *Information Power* GRID de la NASA, la iniciativa de la *National Science Foundation* con los centros de supercomputación de NCSA y SDC, y la *Advanced Strategic Computing Initiative* del DOE.

3. Proyectos de la GRID

Las GRID computacionales representan en la actualidad uno de los temas de investigación de mayor auge en el desarrollo de la infraestructura para el cómputo científico de alto rendimiento y de alto desempeño. A nivel mundial, en los últimos años se ha observado un avance significativo en los proyectos de GRID, avance que se ha visto grandemente impulsado, entre otras cosas, por la estandarización de la arquitectura propuesta en el proyecto Globus.

Aunque la investigación en cómputo GRID ha estado liderada por EEUU a través del Globus Project (origen de la idea), el Global Grid Forum (centrado en el desarrollo de estándares para la GRID) y TeraGRID (centrado en las redes WAN de alta velocidad, y almacenamiento distribuido), también en Europa se han producido avances a través de proyectos europeos como CrossGRID (centrado en supercomputación y análisis de grandes volúmenes de datos), EuroGRID (centrado en la aplicación de tecnologías GRID en el ámbito científico e industrial), DataGRID (centrado en el análisis de grandes volúmenes de datos) y UK e-Science (proyecto del Reino Unido centrado en investigación distribuida).

3.1 Proyectos GRID en Estados Unidos

El desarrollo de los proyectos GRID en EEUU es imparable: entre los científicos cabe citar PDG y GriPhyN (Física de Partículas), DOE ScienceGRID, Earth System GRID (meteorología), Fusion Collaboratory (fusión nuclear), NEESGRID (simulación para el estudio de los terremotos) así como un centro de soporte GRID del NSF. El International Virtual Data GRID Laboratory (iVDGL) y el proyecto TeraGRID, que unirá cuatro centros de EEUU de supercomputación, son dos de los más relevantes.

3.1.1 GADU/GNARE. Usos de TeraGRID para análisis de secuencias de proteínas

GADU/GNARE, el Análisis del Genoma y el desarrollo del sistema de actualización de la base de datos en la división de ciencias Matemáticas y Computacionales del Laboratorio Nacional de Argonne, ha usado los recursos de TeraGRID con éxito por realizar el análisis periódico de alto rendimiento de todas las secuencias de proteínas públicamente disponibles usando herramientas bioinformáticas. Por ejemplo, el tamaño de la base de datos no redundante de la proteína NCBI es actualmente de 2.3 Millones de secuencias. El análisis de estos datos que usan “Explosión y Bloques” requiere en el orden de 7 millones de procesos. Un típico flujo de trabajo de “Explosión y Bloques” incluye varios pasos: dividir el archivo de entrada en pequeños archivos que serán enviados a nodos individuales en el TeraGRID, ejecutar la herramienta bioinformática por el nodo, posteriormente analizar los resultados de cada nodo. Después de que todos los nodos terminan el análisis, los resultados se regresan al *host* que lo solicitó. Todo el flujo de trabajo de la GRID es manejado por el Sistema de Datos Virtual GriPhyN, usando Cóndor-G y Globus. Los resultados del análisis se guardan entonces en una base de datos relacional (Oracle). Los datos almacenados se usan para construir las diferentes aplicaciones bioinformáticas. PUMA2 es un ejemplo de tales aplicaciones. Contiene el análisis de 1031 genomas pre-computarizados en TeraGRID y otros recursos GRID. Los resultados son usados por los algoritmos para la anotación automatizada de la secuencia de datos y son desplegados al usuario para el análisis extenso interactivo.

GADU es una herramienta de análisis y actualización de la base de datos del Genoma para el departamento de Ciencias Matemáticas y de la Computación (MCS) en los Laboratorios Nacionales de Argonne (ANL). GADU es una herramienta automatizada que busca periódicamente a través de diferentes cadenas de ADN y bases de datos de proteínas para encontrar nuevos genomas de diferentes organismos.

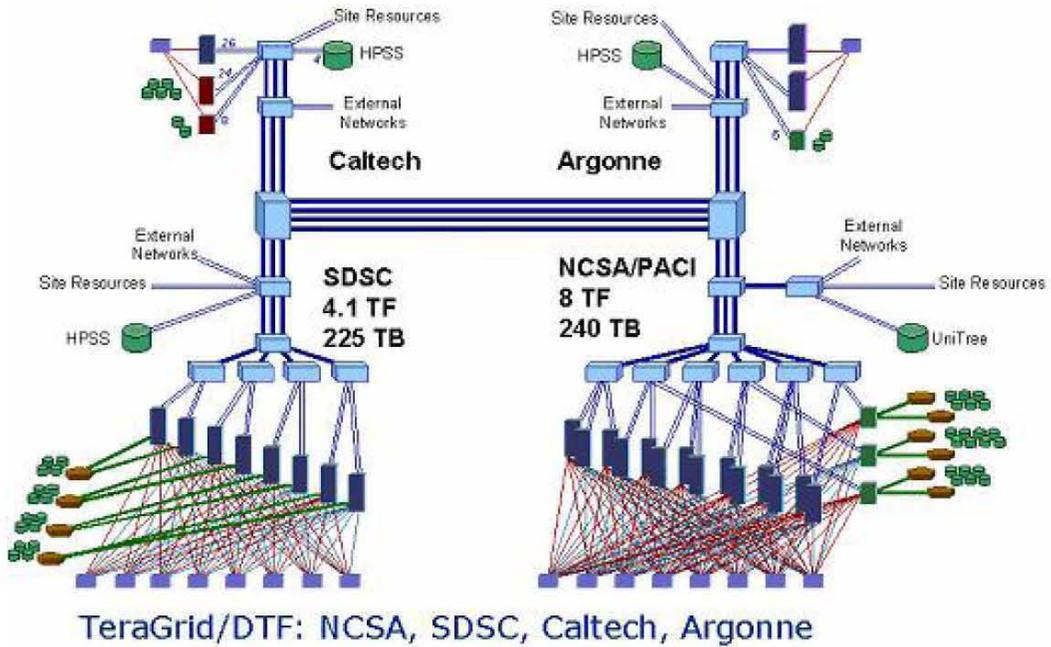


Figura 3.1. Esquema del proyecto TeraGRID



Figura 3.2 Mapa del proyecto NSF (National Science Foundation) TeraGRID

3.1.2 Condor

El objetivo del proyecto Condor es desarrollar, implementar, utilizar y evaluar mecanismos y políticas que soporten Cómputo de Alto Rendimiento (HTC) en grandes cantidades de recursos de cómputo distribuidos. Dirigido tanto por los desafíos tecnológicos como sociológicos de un ambiente de cómputo, el equipo de Condor ha construido herramientas de software que permiten aumentar el rendimiento de cómputo a científicos e ingenieros.

Condor es un sistema especializado en la administración de trabajo para aplicaciones de cómputo intensivo. El mecanismo de trabajo que proporciona Condor consiste en el uso de colas, programa políticas, el esquema de prioridad, monitorea y administra recursos. Los usuarios presentan sus tareas consecutivas o paralelas a Condor, y éste las coloca en una cola, elige cuándo y dónde dirigir las tareas basándose en una política, supervisa cuidadosamente su progreso, y por último informa al usuario sobre la finalización de su tarea.

Condor puede ser utilizado para manejar un *cluster* con nodos dedicados al cómputo. Además, sus mecanismos permiten a Condor disponer con eficacia de la energía desperdiciada de los CPUs inactivos y aprovecharla en otras estaciones de trabajo. Por ejemplo, Condor puede ser configurado para utilizar solamente las máquinas de escritorio donde están inactivos el teclado y el ratón. Si Condor detecta que una máquina no está disponible, produce un punto de comprobación en forma transparente y emigra el trabajo a una máquina diferente que se encuentre desocupada. Condor no requiere un sistema de archivos compartidos a través de las máquinas - si no hay sistema disponible de archivos compartidos, Condor puede transferir los archivos de datos del trabajo a la cuenta del usuario, o podría ser capaz de volver a dirigir transparentemente todas las peticiones de trabajo de regreso a la máquina que las envió. Como resultado, Condor puede ser utilizado para aprovechar toda la energía y recursos de cómputo de una organización.

Condor puede ser usado para construir ambientes de cómputo GRID que van más allá de aplicaciones administrativas. Condor incorpora muchas de las metodologías y protocolos emergentes, basados en el cómputo GRID.

Condor es el producto del Proyecto de Investigación Condor de la Universidad de Wisconsin-Madison (UW-Madison), y fue instalado inicialmente como un sistema de producción del departamento de Ciencias de la Computación de UW-Madison hace 15 años. Actualmente en su departamento, Condor maneja más de 1000 estaciones de trabajo. Las instalaciones adicionales se han establecido a través del campus y del mundo. Centenares de organizaciones en la industria, gobierno, y la academia ha utilizado a Condor para establecer instalaciones de cómputo que se extienden a miles de estaciones de trabajo.

Descripción del Sistema Condor en el Cómputo de Alto Rendimiento

Para muchos científicos, la calidad de sus investigaciones es fuertemente dependiente del rendimiento de cómputo. No es raro encontrar problemas que requieran semanas o meses de cómputo para ser resueltos. Los científicos involucrados en este tipo de investigación necesitan un ambiente que brinde gran cantidad de poder computacional durante un largo periodo. Un ambiente así es llamado “*High-Throughput Computing*” (HTC). En contraste, un ambiente “*High-Performance Computing*” (HPC) ofrece una gran cantidad de poder de cómputo durante

un corto periodo. Los ambientes HCP son medidos frecuentemente en términos de Operaciones de Punto Flotante por Segundo (FLOPS). Muchos científicos hoy no se preocupan por los FLOPS; sus problemas son a mayor escala. Esta gente está preocupada por operaciones de punto flotante por mes o por año. Ellos están interesados en cuantas tareas pueden ser completadas durante un periodo.

La clave para el alto rendimiento es el uso eficiente de los recursos disponibles. Hace años, la comunidad científica utilizaba grandes computadoras centrales para hacer el trabajo computacional. Un gran número de individuos y grupos tendrían que reunir sus recursos financieros para poder adquirir una computadora así. Es bastante común encontrar una máquina así en las instituciones de investigación más grandes. Los científicos tenían que compartir el tiempo de la computadora central, por lo que tenían que asignar una cantidad de tiempo específica para la resolución de sus problemas. Por ello tuvieron que limitar el tamaño y el alcance de sus problemas para asegurar la obtención de resultados. Mientras este ambiente era inconveniente para los usuarios, era muy eficiente, porque la computadora central estaba ocupada casi todo el tiempo.

Como las computadoras llegaron a ser más pequeñas, rápidas y menos costosas, los científicos cambiaron las grandes computadoras centrales y compraron computadoras personales o estaciones de trabajo. Un individuo o un grupo pequeño podían proporcionar un recurso de cómputo que estuviera disponible siempre que lo desearan. El recurso podía ser más lento que la computadora central, pero proporcionaría acceso exclusivo. Recientemente, en vez de una computadora grande para una institución, hay muchas estaciones de trabajo. Cada estación de trabajo es propiedad de su usuario y al mismo tiempo es propiedad distribuida. Mientras que la propiedad distribuida es más conveniente para los usuarios, es también menos eficiente. Las máquinas están desocupadas por largos periodos de tiempo, mientras los usuarios están ocupados haciendo otras cosas. Condor toma este tiempo perdido de cómputo y le da un mejor uso. La situación actual se semeja a la anterior, con la adición de *clusters* en la lista de recursos. Estas máquinas están frecuentemente dedicadas a resolver tareas. Condor maneja el trabajo del *cluster* eficientemente, así como la manipulación de otros recursos.

Para alcanzar este alto rendimiento, Condor proporciona dos funciones importantes. Primero, hace más eficientes a los recursos disponibles poniendo a las máquinas inactivas a trabajar. En segundo lugar, amplía los recursos disponibles para los usuarios, funcionando bien en un ambiente de propiedad distribuida.

3.1.3 Entropia

Entropia es el líder en el cómputo GRID. Las soluciones de la GRID de Entropia preparan y manejan el poder de procesamiento sin explotar de computadoras personales de escritorio dentro de la red de la empresa para tratar tareas computacionalmente intensivas para aplicaciones comerciales y científicas. Las soluciones de Entropia benefician a empresas que utilizan aplicaciones de cómputo intensivo como aquellas extensamente usadas en farmacéuticas, ciencias materiales, químicas e industrias de servicios financieros.

Liberando el poder sin explotar de las computadoras personales existentes, las soluciones de Entropía aceleran la innovación y aumentan dramáticamente el retorno en la inversión para una empresa de cómputo e infraestructura de red.

Entropía y Cambridge Crystallographic Data Centre

La combinación de productos brinda un mejor rendimiento, y una solución más rentable al descubrimiento de medicamentos en adelante de las ciencias naturales. Entropía y el Cambridge Crystallographic Data Centre (CCDC) están realizando un diseño para acelerar la investigación virtual en la industria farmacéutica. Entropía y CCDC están trabajando juntos para perfeccionar el cómputo GRID para la investigación virtual. Esto les permitirá a investigadores acelerar el análisis de grandes cantidades de datos de una manera rentable, al usar el DCGRID de Entropía.

El diseño de medicamentos y la investigación virtual son componentes importantes en el descubrimiento de medicamentos farmacéuticos. Sin embargo, el tamaño de las bases de datos continúa aumentando y pueden requerir un significativo tiempo y poder de cómputo para procesarlas. Las soluciones más comunes al encontrarse con las restricciones de tiempo en un proyecto son comprar más hardware como un gran servidor o un *cluster* de Linux o limitar el alcance de la investigación.

La plataforma de DCGRID es una alternativa rentable al adquirir nuevo hardware. DCGRID recolecta los ciclos de proceso de las PCs de escritorio desocupadas de la red de la empresa para procesar los grandes trabajos de muchas aplicaciones como, por ejemplo, la investigación virtual de moléculas biológicamente activas que usa GOLD. Los usuarios pueden dirigir la investigación extensa a un muy bajo costo y a un ritmo más rápido con la GRID informática.

La comprensión de Entropía sobre las aplicaciones de la ciencia en la vida le hace un compañero ideal. Entropía demostró cómo GOLD podía ejecutar tareas excepcionalmente rápido, demostrando un aumento en el rendimiento en un ambiente de GRID. GOLD escaló a varios cientos de PCs en DCGRID con incrementos lineales en el rendimiento. La solución de Entropía trae una nueva etapa de alto-rendimiento. La GRID computacional ofrece una manera eficaz de incrementar la utilidad de GOLD debido al aumento en el rendimiento y la realización más rápida de grandes trabajos. GOLD, junto con DCGRID, permiten a los investigadores hacer substancialmente mejor ciencia, accesible y económica.

El objetivo de Entropía no es competir con los proveedores de aplicaciones, sino colaborar con los proveedores que son líderes en sus campos. Lo que se desea es atraer a más proveedores de diferentes mercados como las ciencias naturales, las ciencias materiales, los servicios financieros e ingeniería. Es por ello que CCDC se agregó al programa de Entropía.

El CCDC es una organización no lucrativa independiente del Reino Unido que compila y distribuye la Base de Datos estructural de Cambridge, la extensa base de datos de estructuras orgánicas y metal-orgánicas determinadas por la radiografía y técnicas de difracción del neutrón. La CCDC y sus colaboradores también desarrollan los programas de las aplicaciones como SuperStar y GOLD, para investigar interacciones de cadenas de proteínas. Un sistema de base de datos paralelo, Relibase+, también está disponible para el análisis de las cadenas de proteínas de las estructuras de proteínas en el Banco de Datos.

Entropía y Evotec OAI

El objetivo de Evotec OAI AG es el descubrimiento de medicamentos de alto-valor-agregado integrado a la industria farmacéutica y biotecnológica, ha utilizado el DCGRID de Entropía para adicionar poder de cómputo en lugar de comprar soluciones de cómputo de alto rendimiento más costosas. DCGRID es una plataforma poderosa y rentable para el cómputo en GRID que proporciona capacidades de cómputo de alto rendimiento utilizando los ciclos de procesamiento de las computadoras sin usar de una red.

Lo atractivo del cómputo en GRID es que toma el poder de cómputo disponible de las PCs con las que ya cuentan, sin incurrir en el desembolso de un capital substancialmente mayor al adquirir un *cluster* con sistema Linux.

Los científicos de la computación de Evotec OAI actualmente tienen una capacidad de cómputo masiva disponible para el uso de los programas de sus clientes. Se calculó el potencial de cómputo usando 250 de sus PCs y se verificó en el sitio Web de la lista de las 500 mejores supercomputadoras. Lo sorprendente fue averiguar que esta compañía tendría el poder de cómputo equivalente para pertenecer a esta lista.

Evotec OAI compró Entropía por la firmeza de la solución del cómputo GRID, la especialización del dominio en el descubrimiento y los mercados farmacéuticos, y el modelo comercial. Al final la solución del cómputo GRID de Entropía venció a los productos de cómputo GRID disponibles en Internet. Evotec OAI está seguro que seleccionando a Entropía les daría una ventaja agregada a su GRID interna y se enfrentaría a sus metas incrementando la productividad y el rendimiento. Entre más usa la GRID Evotec OAI, más crece su realización, mientras las aplicaciones pueden estar corriendo en muchos sistemas, hacer a la GRID útil requiere más que sólo simples aplicaciones. La especialización de Entropía en estas áreas ha reforzado el por qué Evotec OAI escogió a Entropía.

Evotec OAI es una compañía de servicio cuya habilidad es entregar la calidad más alta y rápida en los resultados científicos que las otras compañías que compiten en el mercado. Evotec OAI ha demostrado claramente que incluso con una GRID modesta de unos cientos de PCs pueden brindar un poder de cómputo de un rendimiento superior. Usar una GRID es la solución para las empresas con bajo presupuesto que buscan más poder de cómputo sin mayor costo.

Evotec OAI ofrece un rango extenso de alto-valor agregado de servicios y productos que se requieren para incrementar la eficiencia y al mismo tiempo reducir el riesgo en la identificación de nuevos medicamentos. Integrando tecnologías innovadoras y procesos en la biología, química e investigación, la compañía ha establecido una única posición para todos los elementos críticos en el descubrimiento de medicamentos y en el proceso de desarrollo - del objetivo al desarrollo clínico. Debido a su extenso conocimiento y experiencia Evotec OAI es a nivel mundial la mejor opción para el área farmacéutica y biotecnológica.

La compañía emplea más de 600 personas, principalmente a dos de sus principales sitios en Hamburgo en Alemania y Abingdon en el Reino Unido. Las subsidiarias se localizan en Europa y América del Norte.

Entropía e IBM

La DCGRID de Entropía, es una plataforma de cómputo GRID poderosa y rentable que proporciona las capacidades de cómputo de alto rendimiento agregando los ciclos de procesamiento no usados de las redes de Windows basadas en computadoras personales existentes, ha sido instalada en el Centro de Diseño de la IBM a petición del negocio electrónico, donde está disponible para demostraciones y revisión por clientes junto con otras tecnologías GRID. Además, las ventas de IBM mundiales, los equipos técnicos y sus clientes que quieren entender más sobre el cómputo GRID pueden tener acceso a DCGRID, por la Intra-GRID de IBM sin necesidad de viajar.

La clave a la estrategia GRID de una compañía es la interoperabilidad de su completa infraestructura de cómputo. Las GRID de PCs proporcionan un recurso disponible de inmediato que puede incrementar inmediatamente el poder de cómputo de una organización.

Las GRID de PCs brindan más opciones del uso eficaz de aplicaciones de cómputo distribuidas. Más opciones permiten a las empresas decidir dónde ejecutar aplicaciones distribuidas basadas en sus características, disponibilidad de recursos y necesidades de calidad de servicios. El centro de Diseño de IBM en Francia es de un inmenso valor cuando permite que compañías vean y comparen diferentes soluciones de alto rendimiento de cómputo.

Entropía y compañías de asistencia médica

Una de las cinco mejores compañías de asistencia médica del mundo compró DCGRID para incrementar su capacidad de cómputo de alto rendimiento disponible para ejecutar aplicaciones que ayuden a la investigación y descubrimiento de medicinas. El objetivo de la compañía es utilizar la GRID de PCs para explorar y aprovechar el potencial sin explotar de sus computadoras personales conectadas a una red como un complemento a sus otras plataformas de cómputo.

DCGRID de Entropía, es utilizada en uno de los sitios de investigación farmacéuticos estadounidenses de este cliente. Los sitios adicionales dentro de su organización global serán traídos entonces en línea con el tiempo para encontrar las necesidades comerciales de la compañía.

Con un apetito creciente de poder de cómputo en los centros de organización R&D, las GRID tendrán un impacto inmediato para ejecutar aplicaciones de cómputo intensivo y satisfarán las peticiones de recursos para sus necesidades de crecimiento. Una razón por la que DCGRID fue elegida es que es la única solución de GRID diseñada para trabajar dentro de la empresa y apoyar la ejecución de una amplia variedad de aplicaciones comerciales y propias sin la necesidad de la modificación del código fuente. Esto permite el uso rápido de aplicaciones tanto al principio como con las revisiones del software subsecuentes.

Mientras las organizaciones sigan sintiendo la presión de incrementar su capital en los tiempos económicos desafiantes de hoy, la capacidad de preparar el poder de cómputo sin explotar dentro de sus computadoras personales se hace crítica. La especialidad de Entropía y su enfoque por revelar el valor no descubierto atrapado dentro de las computadoras personales de una organización y convertir un recurso gastado en uno que proporciona el poder de cómputo solicitado.

Entropía y *The Scripps Research Institute*

La fase inicial del proyecto proporciona el equivalente a más de 1,400 años de cómputo continuo hacia la investigación del tratamiento de SIDA.

Entropía y el Laboratorio Olson en *The Scripps Research Institute* (TSRI), la organización de investigación biomédica privada, no lucrativa nacional más grande, lograron la finalización de la fase inicial del proyecto de investigación FightAIDS@Home (FAAH). El FAAH acelera la investigación de SIDA generando y probando compuestos de medicina contra modelos detallados del virus evolucionado del SIDA.

Con el cambio a la siguiente etapa, el programa se hará completamente auto-manejado por TSRI, que ya ha comenzado a convocar participantes. Usando el software desarrollado por Entropía, los miles de miembros contribuyen al proyecto con sus recursos de cómputo desocupados. Casi 60,000 máquinas en 20 países están implicadas en el proyecto, registrando tan sólo 1,400 años de cómputo continuo, y realizando más de nueve millones de tareas. Estos recursos distribuidos con eficacia funcionaron como una computadora con 14 terabytes de memoria y 1,335 terabytes de espacio de disco.

El proyecto FightAids@Home ha sido un enorme beneficio a la investigación de la medicina terapéutica de VIH y Entropía ha sido un compañero clave en esta importante causa.

La investigación es un elemento crítico en la lucha contra el SIDA y este proyecto muestra cómo al juntar las contribuciones individuales se puede brindar un suministro del verdadero músculo de cómputo. No habría sido factible para TSRI realizar las masivas cantidades del análisis que hizo sin la GRID FightAIDS@Home.

Gracias a los recursos de los miembros de cómputo distribuidos, millones de cálculos sobre medicamentos han sido dirigidos en la red de FAAH. El proyecto ha demostrado hábilmente que con tales capacidades de cómputo masivo, los investigadores pueden utilizar acercamientos intensivos para identificar medicamentos candidatos que sucumben a la resistencia de mutaciones y aquellos que son más resistentes a tales mutaciones.

La siguiente fase establecerá la fundación de investigación pero dirigirá cálculos co-evolutivos aún más masivos para buscar características óptimas de medicamentos en la evasión de resistencia de mutaciones. En cuanto haya más datos disponibles, serán integrados en los cálculos con el objetivo de desarrollar estrategias aún más sofisticadas para las terapias de VIH.

The Scripps Research Institute es una de las organizaciones de investigación científicas más grandes en el mundo, privada y no lucrativa. Está a la vanguardia en la ciencia biomédica básica, un segmento vital de la investigación médica que procura entender los procesos más fundamentales de la vida, y es reconocido para su investigación en biología molecular y celular, química, inmunología, neurociencia y medicina molecular.

Entropía y Sistemas BAE

DCGRID aumenta y amplía las capacidades del cómputo GRID de los sistemas BAE para un amplio rango de aplicaciones de diseño de software de ingeniería.

Entropía trabaja en colaboración con Sistemas BAE, el líder en sistemas y compañía aeroespacial. Esta relación combina la especialización global de los Sistemas BAE en la ingeniería y el diseño mecánico con la dirección tecnológica de la GRID de Entropía para llevar más allá la meta.

DCGRID de Entropía será integrada dentro de la infraestructura de los Sistemas BAE. Los Sistemas BAE tienen la intención de usar su GRID como una plataforma de investigación para sus aplicaciones existentes y aplicaciones de terceros. Se desarrollarán capacidades adicionales que permitirán insertar DCGRID dentro de sus propias aplicaciones durante esta colaboración.

Se han empleado durante algún tiempo metodologías GRID pero la adición de la tecnología de Entropía proporciona una solución profesionalmente diseñada que fue construida para el escritorio. La arquitectura abierta del DCGRID facilita la interoperación, usando OGSA y Globus Toolkit, con las capacidades existentes. Además, el modelo de integración de aplicación del DCGRID permitirá que se desarrollen conjuntamente capacidades específicas de ingeniería y del sector de diseño mecánico.

Las diversas aplicaciones de los Sistemas BAE los hacen un compañero ideal para sacar adelante las capacidades de la familia del producto de Entropía para el sector del mercado de ingeniería. Considerando sus objetivos mutuamente estratégicos, los Sistemas BAE se beneficiarán directamente de un esfuerzo de desarrollo de software dedicado, mientras DCGRID está a prueba de presión en verdaderas condiciones de producción mundial en el área de ingeniería.

Los Sistemas BAE son una compañía de sistemas, innovando para un mundo más seguro. Y emplean a casi 100,000 personas incluyendo empresas conjuntas, y tienen ventas anuales de alrededor de 12 mil millones de libras esterlinas. La compañía ofrece una capacidad global en aire, mar, tierra y espacio con una capacidad de contratación principal de categoría mundial apoyada por una variedad de habilidades claves. El diseño de los Sistemas BAE, fabricación y soporte de aviones militares, superficies de barcos, submarinos, radares, aeroelectrónica, comunicaciones, electrónica, sistemas de armas guiadas y un rango de otros productos de la defensa. Los Sistemas BAE están dedicados a la fabricación de conexiones inteligentes necesarias para entregar soluciones innovadoras.

3.1.4 La ciencia de SETI@home

SETI (*Search for Extraterrestrial Intelligence*) es un área científica cuya meta es descubrir la vida inteligente fuera de la Tierra. Un acercamiento, conocido como radio SETI, usa radio telescopios para escuchar radio señales del espacio. Tales señales no pueden producirse naturalmente, así que una detección podría proporcionar evidencia de tecnología extraterrestre.

Las señales del radio telescopio consisten principalmente en ruido (de las fuentes celestes y la electrónica del receptor) y señales artificiales como las estaciones de televisión, radar, y satélites. Los proyectos modernos de radio SETI analizan los datos digitalmente. Más poder de cómputo permite a las búsquedas cubrir un rango de frecuencias mayores. Por consiguiente, radio SETI tiene un apetito insaciable de poder de cómputo.

Los proyectos anteriores de radio SETI han usado supercomputadoras de propósito específico, localizadas en el telescopio, para hacer el análisis del volumen de datos. En 1995, David Gedye propuso transmitir por radio SETI utilizando una supercomputadora virtual integrada por un gran número de computadoras conectadas a través de Internet, y él organizó el proyecto SETI@home para explorar esta idea. SETI@home se lanzó originalmente en mayo de 1999.

Visitar otra civilización en un mundo distante podría ser fascinante, pero actualmente está más allá de la capacidad humana. Sin embargo, está perfectamente dentro del alcance desarrollar un sistema de comunicación que use un transmisor poderoso y un receptor sensible, y se utilice para investigar el cielo y buscar mundos extraterrestres cuyos ciudadanos tengan una inclinación similar a la de los terrestres. Una suposición básica de SETI es la idea que la humanidad no es exótica en el Cosmos pero en cierto sentido es "típica" o "elemental" comparada con otras especies inteligentes. Esto significaría que la humanidad tiene bastante similitud con otros seres inteligentes que la comunicación será mutuamente deseable y entendible. Si esta suposición básica es correcta y otras especies inteligentes están presentes en cualquier número en la galaxia en nuestro nivel tecnológico o anterior, entonces la comunicación entre los dos mundos debe ser inevitable.

SETI no es una tarea trivial. La vía láctea tiene 100,000 años luz, y contiene cien billones de estrellas. Investigar el cielo entero y las débiles señales es un trabajo agotador, sobre todo para descifrar las señales que se reciben de los radio telescopios, es por ello que se requiere de gran poder de cómputo para analizar todos los datos recibidos. Y una gran opción para el procesamiento de esas señales es la GRID.

La existencia del proyecto SETI@home significa que cualquier individuo puede involucrarse con la investigación de SETI simplemente descargando de Internet un software protector de pantalla. El software realiza el análisis de las señales y regresa el reporte de los resultados a través de Internet. Más de 5 millones de computadoras de los usuarios en más de cien países han firmado para SETI@home y han contribuido con más de 19 mil millones de horas de tiempo de procesamiento colectivamente.

El propósito de SETI@home es analizar los datos entrantes del radio telescopio de Arecibo (Puerto Rico), mientras busca posible evidencia de transmisiones de radio de inteligencia extraterrestre. Con más de cinco millones de participantes a nivel mundial, el proyecto es la GRID de cómputo con más participantes actualmente.

Software

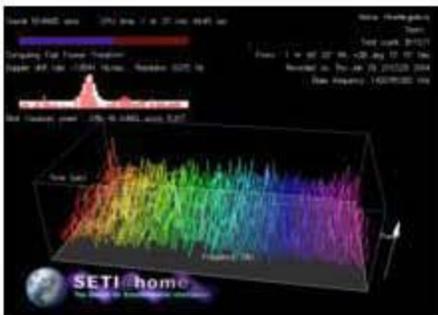


Figura 3.3.1 SETI@home versión 4.18

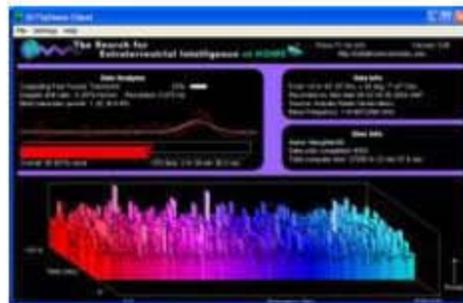


Figura 3.3.2 SETI@home de un cliente clásico (versión 3.08)

El software de cómputo distribuido de SETI@home, está disponible para la mayoría de los sistemas operativos, se ejecuta como un protector de pantalla o continuamente mientras un usuario trabaja, haciendo uso de poder de procesamiento desperdiciado para la investigación. SETI@home fue la primera aplicación popular del cómputo GRID.

SETI@home, además de su uso altruista para ayudar a SETI, es bastante útil como una herramienta de comprobación de stress para las estaciones de trabajo. SETI@home se usa a menudo para inspeccionar la fiabilidad de la configuración de una computadora cuando se presenta un *overclocking*.

Hay planes futuros de recibir datos del Observatorio de Parkes en Australia para analizar el hemisferio sur.

3.2 Proyectos GRID europeos

En el año 2001 se lanzan varios proyectos como GRIDLab, Grace, DataTag y CrossGRID. El CSIC (IFCA, IFIC y RedIRIS) es uno de los socios principales, responsable del apartado de *testbed* en el que participan la UAB (Barcelona) y la USC (Santiago) en colaboración con el CESGA. El IFCA es también responsable de la aplicación interactiva de Física de Partículas; la USC y la Universidad de Cantabria participan en el bloque de aplicaciones meteorológicas, y la UAB (Barcelona) desarrolla el software que permita la asignación de recursos distribuidos para aplicaciones interactivas. El proyecto propone además desarrollar para el entorno GRID aplicaciones como el análisis de imágenes y simulación para operaciones de cirugía vascular en medicina, el control de inundaciones, etc.

Uno de los puntos clave del proyecto es el acceso a bases de datos distribuidas, y el desarrollo de técnicas de minería de datos como redes neuronales, que requieren técnicas de computación distribuidas, sobre protocolos adaptables al entorno GRID como el *MPI*. La figura 3.4 muestra el *testbed* del proyecto CrossGRID, incluyendo la infraestructura de red básica de GÉANT.

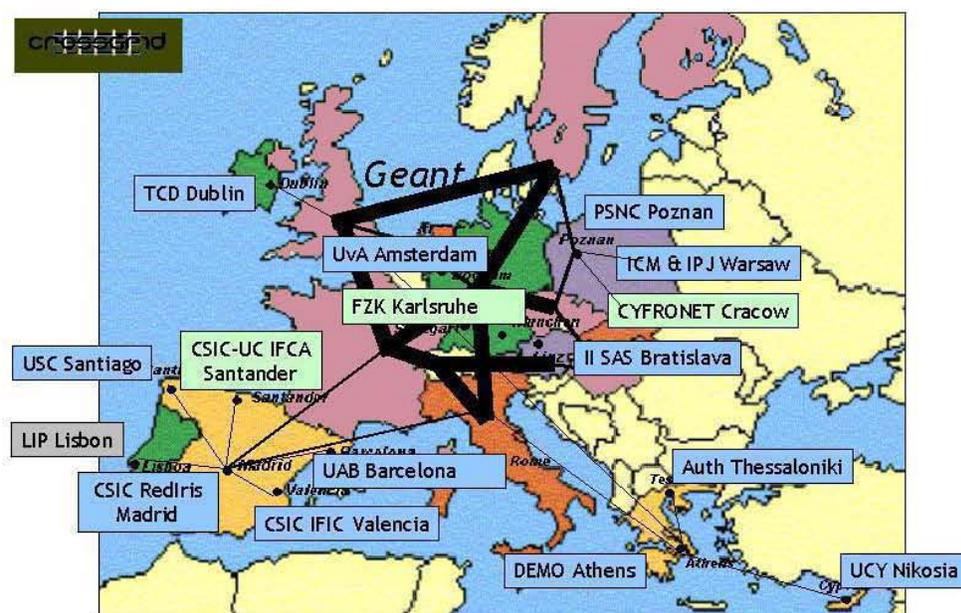


Figura 3.4. Mapa del *testbed* del proyecto europeo CrossGRID

Cada centro incluye una configuración mínima, necesaria para permitir la ejecución de aplicaciones en el entorno GRID. La configuración actual, definida por el proyecto DataGRID, está basada en PCs con sistema operativo Linux. En la GRID de producción los centros pueden aportar los recursos disponibles incrementando el número de elementos de cómputo y almacenamiento. Éste es el caso por ejemplo de dos centros del CSIC, IFCA e IFIC, que cuentan con granjas de más de cien procesadores: “Santander GRID Wall” (SGW), y “Granja orientada a GRID” (GoG).

3.2.1 GÉANT

El proyecto de GÉANT se lleva a cabo con la colaboración de 26 Naciones de investigación y redes de Educación, representando a 30 países de Europa, la Comisión europea, y DANTE. Su propósito principal es desarrollar la red GÉANT, una red multi-gigabit de comunicación de datos europea, reservada específicamente para el uso de la investigación y la educación. El proyecto también cubrió varias otras actividades relacionadas en la investigación de redes. Estas incluyen pruebas de la red, desarrollo de nuevas tecnologías y apoyo para algunos proyectos de investigación con requerimientos de redes específicas.

GÉANT es el nombre dado al proyecto y la red que surgieron de él. GÉANT2, el sucesor a GÉANT, empezó oficialmente el 1º de septiembre de 2004. Los dos proyectos se ejecutaron uno al lado de otro hasta junio de 2005. Aunque GÉANT sigue proporcionando servicio hasta que la construcción de GEANT2 esté completada. La red conecta a un total de 34 países. La red GÉANT es construida y operada por DANTE, el cual también coordina el proyecto GÉANT en su rol de administrador.

El propósito del proyecto GÉANT ha sido mejorar la red de investigación europea anterior (TEN-155) creando un nuevo *backbone* con una velocidad en gigabits. El proyecto tiene cuatro objetivos principales:

- Velocidad en Gigabits
- Expansión Geográfica
- Conectividad Global
- Calidad garantizada de servicio

El proyecto continúa extendiendo la red tanto aumentando la velocidad de transmisión como extendiendo su cobertura geográfica.

Mejorar la conectividad global ha sido un continuo esfuerzo y GÉANT ahora tiene una conectividad de 12 Gbps con América del Norte y 2.5 Gbps con Japón. Las conexiones adicionales con GÉANT se han establecido con el mediterráneo sur a través del proyecto de EUMEDCONNECT. El trabajo también consiste en establecer las conexiones adicionales a GÉANT otras regiones mundiales, incluso América Latina (a través de ALICE) y la región de Asia-Pacífico (a través de TEIN2).

El proyecto es innovador adquiriendo e integrando los sistemas de la transmisión más avanzados y el equipo de asignación de ruta disponible para crear una red que permanezca a la vanguardia de los desarrollos de las redes de investigación. GÉANT se compromete a proporcionarles una red de calidad mundial a los investigadores europeos.

Desde que se puso en operación el servicio de GÉANT, se pusieron varios servicios disponibles en la red. El uso de servicios de IPv6 en GÉANT, además, se han comenzado otros desarrollos para proporcionar:

- La Calidad de Servicio de IP
- IP MULTICAST
- Redes Privadas Virtuales



Figura 3.5. Mapa del proyecto GÉANT

GÉANT2

GÉANT2 es la séptima generación de la red de investigación y educación Europea, sucesor de la red de investigación multi-gigabit GÉANT. GÉANT2 fue fundada por la Comisión europea y redes de investigación y educación nacional de Europa, y es administrada por DANTE.

La red de GÉANT2 conectará 34 países a través de 30 redes de investigación y educación nacional (NRENs), usando longitudes de onda de múltiplos de 10 Gbps. GÉANT estableció las conexiones del *backbone* europeo a América del Norte y a Japón. Racionalizando la interconexión, GÉANT era capaz de proporcionar el acceso mejorado a eslabones

intercontinentales a muchos NRENs que habían luchado individualmente para permitirse instalaciones de tránsito intercontinentales rápidas para esta escala.

Las redes de investigación alrededor del mundo usan una variedad diversa de tecnologías, acercamientos operacionales y procedimientos. Sin embargo, el aumento del interés global a la investigación internacional y la cooperación en la educación conducirá y apoyará el desarrollo de la investigación de servicios de redes a una escala global. El GÉANT2 consolidará arreglos existentes, racionalizando y ampliándolos para desarrollar un concepto integrado para la interconexión global.

GÉANT2 también se beneficiará de las conexiones de otras regiones mundiales que han sido (o serán) alcanzadas a través de otros proyectos de investigación. Estos proyectos han tomado el modelo de red de investigación europea y lo han aplicado a otras regiones del mundo. Esto ha ayudado a desarrollar NRENs dentro de estas regiones y ha ampliado eslabones de GÉANT's a importantes áreas del mundo:

- Europa del Sur y del Este por SEEREN
- El Mediterráneo por EUMEDCONNECT
- América Latina por ALICE
- La región de Asia-Pacífico por TEIN2.

La realización de la interconexión global coordinada entregará ventajas enormes para investigación internacional y colaboración en la educación.

3.2.2 DataGRID

En el año 2000 el programa comunitario IST lanzó el proyecto *European DataGRID* (EDG) coordinado por el CERN, con el objetivo de construir la próxima generación de infraestructura de computación que permita el cálculo intensivo y análisis de bases de datos compartidas a gran escala, desde cientos de terabytes a petabytes, entre comunidades científicas ampliamente distribuidas. El proyecto utiliza Globus como software básico, y desarrolla nuevo “*middleware*” para construir aplicaciones que manejan un gran volumen de datos, como las de Física de Partículas (CERN, LHC), de Bioinformática, y de la ESA (Observación de la Tierra). El proyecto cuenta como socios principales con entidades de investigación nacionales con instalaciones significativas de cálculo, como el INFN (Italia), PPARC (Reino Unido), CNRS (Francia) o NIKHEF (Holanda). Su interconexión está basada en la nueva red gigabit europea GÉANT, instalada desde finales del 2001.

Uno de los objetivos básicos del proyecto es el desarrollo de una GRID distribuida por toda Europa.

y el Reino Unido. DataGRID es un proyecto ambicioso. Su desarrollo se beneficia de muchas diferentes clases de tecnología y especialización.

El primer y principal desafío que afronta el proyecto es compartir enormes cantidades de datos distribuidos sobre la infraestructura de red que está disponible actualmente. El proyecto DataGRID no comienza desde el principio en este desafío: éste surge a partir de tecnologías de GRID computacionales que se esperaban para hacer factible la creación de un gigantesco ambiente computacional de un conjunto de archivos distribuidos, bases de datos, computadoras, instrumentos y dispositivos científicos.

El proyecto de DataGRID está dividido en doce paquetes de trabajo distribuidos en cuatro grupos de funcionamiento: *Testbed*, Infraestructura, *Middleware* y Aplicaciones. La figura 3.7 ilustra la estructura del proyecto y la interacción entre los paquetes de trabajo.

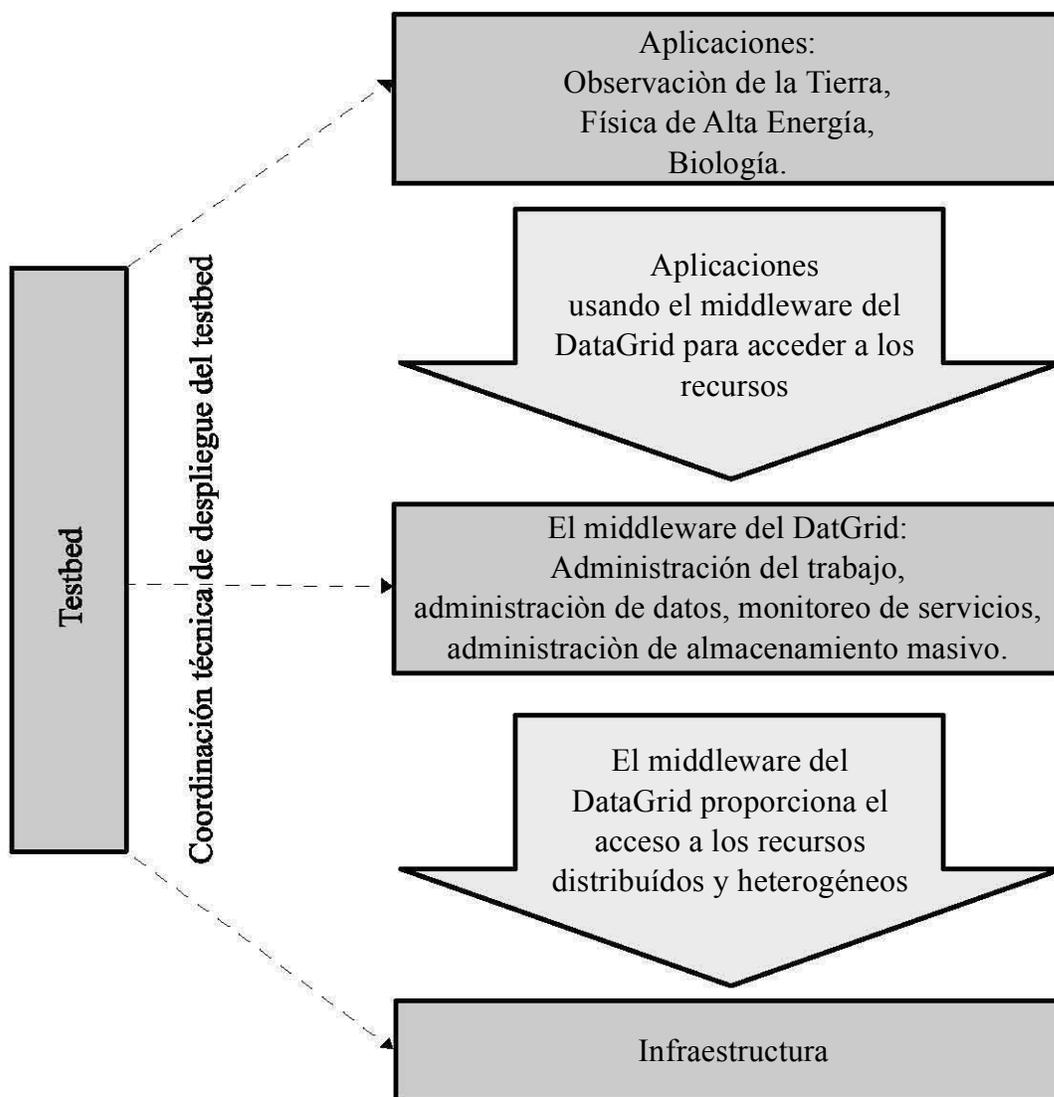


Figura 3.7 Organización de los paquetes de trabajo técnicos en el proyecto de DataGRID

Áreas de Aplicación

Las tres áreas de aplicación de verdadero cómputo intensivo se encuentran en los proyectos:

- **Física de Alta Energía (HEP)**, dirigido por el CERN,
- **Procesamiento de Imágenes Biológicas y Medicas**, dirigido por CNRS (Francia),
- **Observaciones de la Tierra (EO)** dirigido por la Agencia del Espacio Europeo.

3.2.3 CrossGRID

Desarrollo de nuevos componentes GRID

El proyecto CrossGRID está orientado hacia las aplicaciones de cómputo intensivo de datos que se caracterizan por procesamiento de ciclos. Tales aplicaciones requieren una respuesta de la GRID a una acción por un agente humano en diferentes escalas de tiempo.

Hay muchos problemas a gran escala que requieren nuevos acercamientos al cómputo, como la observación de la tierra, la administración ambiental, biomedicina, modelado industrial y científico. El proyecto de CrossGRID se dirige a problemas realistas en medicina, protección del medio ambiente, predicción de inundaciones, y análisis físicos. Estas aplicaciones están orientadas hacia usuarios finales específicos.

El proyecto CrossGRID implica a 21 compañeros y es coordinado por CYFRONET, el Centro de Cómputo Académico en Cracovia, Polonia. La diseminación y la explotación de los resultados del proyecto de CrossGRID son coordinadas por Algosystems, una compañía establecida en Atenas, Grecia.

Medicina

Los médicos, podrían obtener nuevas herramientas para ayudar a obtener diagnósticos correctos y dirigirlos durante simulaciones de operaciones y visualización para procedimientos quirúrgicos.

Se está desarrollando un prototipo de un sistema para la planificación de intervenciones vasculares y procedimientos quirúrgicos en tiempo real de la estructura vascular y el flujo sanguíneo. El sistema consistirá en un ambiente de simulación distribuido de tiempo real, con el cual un usuario interactúa en la Realidad Virtual (VR). Un modelo 3D de las arterias de un paciente, se obtiene usando técnicas de imágenes médicas, y servirá como entrada al ambiente para los cálculos de flujo sanguíneo. El usuario podrá cambiar la estructura de las arterias, imitando una intervención o procedimiento quirúrgico. Los efectos de esta adaptación serán analizados en tiempo real y el resultado será presentado al usuario en un ambiente virtual.

El papel de la GRID

La simulación avanzada con base en la GRID y las técnicas de visualización interactivas requieren que el acceso del usuario sea transparente y seguro a recursos distribuidos y cómputo de alto rendimiento. La infraestructura CrossGRID brinda los datos y exploraciones médicas, el ambiente de visualización y los recursos computacionales requeridos para el flujo de cálculos situados en posiciones geográficas separadas y en dominios administrativos distintos. El uso de la GRID permite que la aplicación tenga acceso a los recursos de cómputo de alto rendimiento, sin requerir de una gran inversión de capital.

La Sección de Ciencia Computacional de la Universidad de Amsterdam, Holanda, es responsable del desarrollo de la aplicación biomédica dentro del proyecto de la CrossGRID.

Inundaciones

Los equipos de crisis de inundación, que podrían predecir el riesgo de una inundación con base en archivos históricos y datos hidrológicos y meteorológicos actuales de equipos de crisis de inundaciones apoyarían la decisión del sistema.

La utilidad de la tecnología GRID para apoyar equipos de crisis de inundación en ríos internacionales está siendo estudiada. Los desafíos en esta tarea son la adquisición de recursos significativos con poca anticipación, respuesta rápida en tiempo real, la administración de combinación de datos distribuidos y cómputo distribuido, los requerimientos computacionales para la combinación de modelos hidrológicos e hidráulicos y, eventualmente, acceso móvil en condiciones adversas. La visualización 2D de los resultados de la simulación es posible actualmente, pero necesita la adaptación al ambiente GRID; las posibilidades adicionales para la visualización 3D están siendo investigadas.

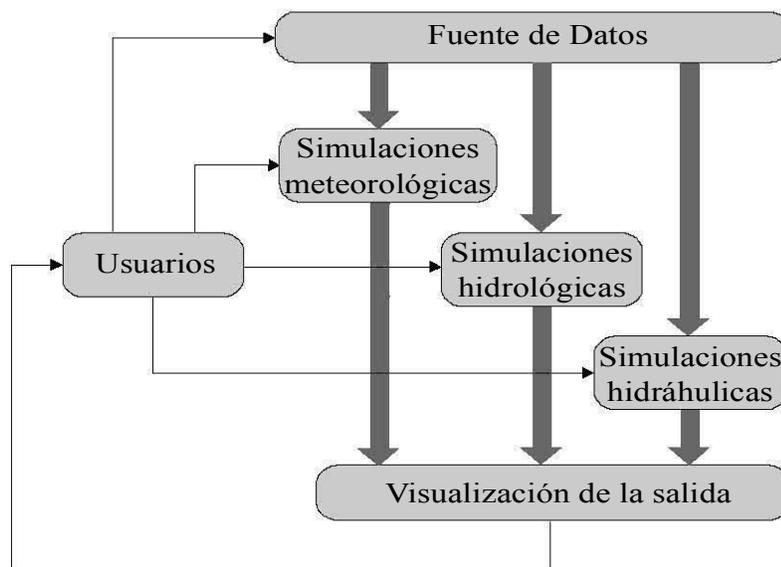


Figura 3.8 Tecnología GRID en apoyo a inundaciones

El Instituto de Informática de la Academia de Ciencia de Eslovaquia, en colaboración con el Instituto Meteorológico de Eslovaquia, es responsable del desarrollo del sistema dentro del proyecto CrossGRID.

Física de partículas

Los físicos son quienes podrían optimizar el análisis de volúmenes masivos de datos distribuidos a través de países y los continentes distribuyendo el análisis de datos en física de alta energía.

La siguiente generación de experimentos de física de alta energía requerirá recursos de cómputo sin precedentes para el análisis de datos. El objetivo de CrossGRID es desarrollar aplicaciones para el usuario final y para el análisis físico que se ejecuta de modo distribuido, en un ambiente GRID, usando grandes bases de datos distribuidas. Las técnicas de minería de datos distribuida reducirán el tiempo de espera en este análisis de horas a minutos, permitiendo pruebas de estrategias poderosas en grandes bases de datos. El uso de recursos de cómputo distribuidos para la filtración de tiempo real también está siendo estudiado. Estas aplicaciones serán empleadas en el campo de la física de alta energía.

Las técnicas de filtración en línea, así como algoritmos matemáticos sofisticados y redes neuronales, serán usadas para seleccionar aquellos acontecimientos que serán analizados por físicos que trabajan en centros de investigación a través del mundo entero. Ellos compartirán sus recursos de cómputo en una gran GRID.

El Instituto de Física de Cantabria (IFCA, CSIC) en Santander, España es el líder del desarrollo de esta aplicación, en colaboración con IFIC (Valencia), INP (Cracovia), UAB (Barcelona), INS (Varsovia) y FZK (Karlsruhe).

Contaminación meteorológica

El uso interactivo y la escalabilidad de la tecnología GRID están siendo investigados, a fin de reunir investigación atmosférica y aplicaciones para los requerimientos de la comunidad de usuarios. La especificación de requerimientos típicos para la investigación atmosférica, así como su desarrollo y realización, es el objetivo de esta actividad. Una aplicación completa implica poner en servicio las herramientas remotas de la GRID, coordinada la retroalimentación de modelos atmosféricos y modelos de onda, basados en datos costeros locales y forzados por campos de viento generados por los componentes atmosféricos del sistema. Los estudios de química de contaminación atmosféricos también serán incluidos.

El sistema emplea la plataforma del CrossGRID de servicios GRID y herramientas para conectar transparentemente a los expertos, los datos y recursos de cómputo necesarios para diferentes aplicaciones meteorológicas. El sistema integrará técnicas de comunicación avanzadas, permitiéndoles a los expertos en contaminación aérea y marítima ejecutar las simulaciones con diferentes parámetros y analizar el impacto de condiciones meteorológicas en la contaminación del aire y del mar.

El Centro Interdisciplinario para el Modelado Matemático y Computacional, la Universidad de Varsovia, Polonia, la Universidad de Coruña y el Consejo Superior de Investigaciones Científicas, España, junto con la Universidad de Cantabria, son responsables del desarrollo del sistema dentro del proyecto de CrossGRID.

3.2.4 EuroGRID

Aplicaciones en cómputo GRID para los europeos

El proyecto de EuroGRID, fue un proyecto de costos compartidos para la Investigación y Desarrollo de Tecnología (RTD, por sus siglas en inglés) aprobado por la Comisión europea. Es parte del Programa de la Sociedad de Tecnologías de Información (IST). El proyecto de EuroGRID demostró el uso de las GRID en comunidades científicas e industriales seleccionadas, destinado hacia exigencias específicas de estas comunidades, y resaltando los beneficios de usar las GRID.

Los objetivos del proyecto de EuroGRID son:

- Establecer la red de GRID europea guiada por centros de cómputo de alto rendimiento en diferentes países europeos.
- Operar y apoyar la infraestructura de software de EuroGRID. El software de EuroGRID usará la red de Internet existente y ofrecerá un acceso seguro y continuo para los usuarios de EuroGRID.
- Desarrollar componentes de software importantes de GRID e integrarlos en EuroGRID (transferencia rápida de archivos, agente de recursos, la interfaz para aplicaciones acopladas y acceso interactivo).
- Demostrar códigos de simulación distribuidos en diferentes áreas (simulaciones biomoleculares, predicción del tiempo, simulaciones acopladas de CAE, análisis estructural y procesamiento de datos en tiempo real).
- Contribuir al desarrollo internacional de las GRID y vincularse con los principales proyectos de GRID internacionales.
- Producir los componentes de software para la EuroGRID. Al final del proyecto, el software de EuroGRID estará disponible como un producto de soporte.

Todos estos objetivos se han logrado durante el proyecto. Los componentes de software EuroGRID se explotarán por partes para las GRID de cómputo, dominando áreas científicas específicas de biología, química, meteorología y computación, incluyendo la ingeniería.

Bio GRID

La Bio GRID desarrolló un portal de acceso para los recursos de modelado biomolecular. Se desarrollaron interfaces para permitir a químicos y a biólogos enviar su trabajo a los centros de alto rendimiento (HPC). Esta tarea se enfocó en el desarrollo de varias interfaces de aplicaciones biomoleculares y bases de datos. La principal tarea de la Bio GRID fue integrar aplicaciones selectas a la infraestructura de UNICORE y proporcionar herramientas para los inexpertos en la informática de alto rendimiento. Se desarrollaron conjuntos de herramientas e interfaces de usuario para la simulación y visualización de biomoléculas.

Como paso final el portal de cómputo se estableció y probó con varias aplicaciones biomoleculares en ICM y varios usuarios y proyectos de biomoléculas.

Para lograr estas metas el software de UNICORE fue evaluado y se establecieron las características requeridas por la Bio GRID. El siguiente paso fue seleccionar paquetes de

software usados ampliamente en la investigación biomolecular y adaptarlos a la infraestructura de UNICORE. Esta tarea requirió el desarrollo de las interfaces apropiadas y protocolos de la simulación. La información sobre los paquetes de software instalados y los recursos disponibles de computación es recolectada y proporcionada al usuario a través del software de UNICORE. El desarrollo se enfocó en paquetes seleccionados, bien establecidos y ampliamente usados en diferentes áreas de investigación biomolecular, por ejemplo, dinámica molecular o química cuántica. Son considerados los códigos escalares, vectoriales y paralelos. Los usuarios tienen acceso a la interfaz de varias bases de datos, normalmente accedidas mediante Internet y publicadas alrededor del mundo. Esto permite simplificar el acceso a los bases de datos e integrarlas a las simulaciones.

El Bio GRID estuvo disponible primero para los usuarios de las instituciones asociadas a la EuroGRID y después, para la comunidad biomolecular en general, tan pronto como las interfaces requeridas fueran desarrolladas.

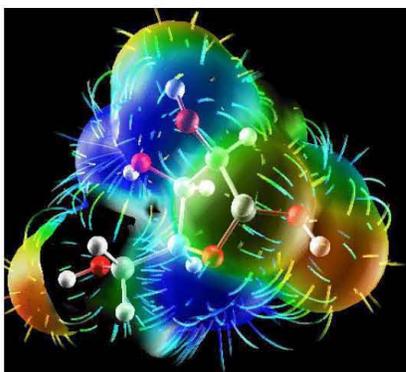


Figura 3.9 Bio GRID

Meteo GRID

Basado en un proyecto de desarrollo de tecnología, un GUI fue desarrollado e integrado en el ambiente de EuroGRID. Este GUI permite un fácil control de la ejecución de LM, por ejemplo, con relación al modelo de dominio (en cualquier parte del globo), al tamaño de la red (normalmente entre 25 y 2 km), a la longitud de previsión (arriba de las 48 horas), fecha y tiempo inicial, y el rendimiento de archivos de resultados.

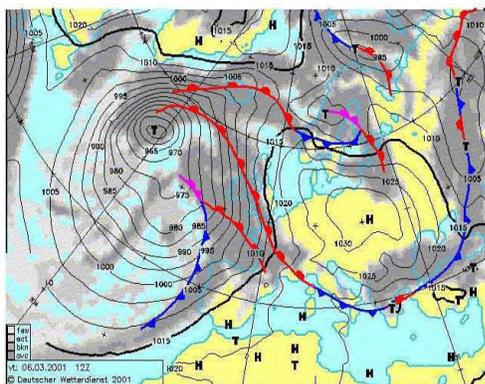


Figura 3.10 Meteo GRID

CAE GRID

Este paquete de trabajo está enfocado en las aplicaciones de la industria CAE (*Computer Aided Engineering*). Proporciona servicios de HPC a los clientes que requieren un amplio poder de cómputo para resolver problemas de Ingeniería. Los socios mayoritarios en este paquete de trabajo son: T-Systems y el centro EADS Corporate Research. Ellos comenzaron utilizando la funcionalidad inicial de UNICORE e investigaron dos importantes aspectos del uso industrial del CAE, acoplamiento de código y tipo de servicios ASP.

El código de acoplamiento u optimización multi-física es el punto clave para mejorar el diseño principal del sistema reduciendo los costos de prototipos y pruebas. El soporte general para el acoplamiento de las aplicaciones fueron integradas en el sistema EuroGRID.

Los servicios de tipo ASP diseñaron aplicaciones específicas de las interfaces de usuario para el trabajo sometido, por ello la complejidad del uso de sistemas HPC se encuentra oculto a los ingenieros de la aplicación. Al mismo tiempo, este paquete de trabajo se dirigió a algunos problemas relacionados a la predicción exacta de costos estimados para simulaciones CAE que le permitiera al usuario escoger si continuaba o cancelaba el cálculo.

Esta investigación sobre HPC con GRID fue usada como una prueba para el desarrollo de aplicaciones distribuidas, y como un ruedo entre el trabajo cooperativo y socios de EuroGRID. El último objetivo fue la integración de un sistema de producción completo en la GRID, y una cercana colaboración entre diferentes sitios resolviendo los mayores desafíos científicos, usando recursos computacionales distribuidos en una escala Europea.

El paquete de trabajo se concentró en el desarrollo de aplicaciones distribuidas generales. Las aplicaciones distribuidas que tenían el valor agregado más grande por la investigación científica, son aquellas involucradas con simulaciones multi-físicas o confiando en cadenas computacionales complejas, basadas en el acoplamiento de módulos de diferente software. Estas aplicaciones distribuidas exigen el uso coherente y complementario de recursos computacionales, y trazan muy naturalmente un ambiente de supercómputo heterogéneo.

El acoplamiento secuencial de módulos del software se logró de la entrada, a través de la funcionalidad inicial de UNICORE. Se desarrollaron aplicaciones más ambiciosas que involucran interacción del tiempo real de módulos del software usando estándares portátiles y bien establecidos como CORBA (Common Object Request Broker Architecture). Este paquete de trabajo se dirigió a algunos problemas relacionados a la calidad de servicios del HPC en la investigación de la GRID, como el ancho de banda de Internet, la administración, operación y seguridad de problemas.

Desarrollo de tecnología

El paquete de trabajo 5 de la EuroGRID fue diseñado para construir en base al funcionamiento de UNICORE y extenderlo para mantener el *middleware* necesario para los otros paquetes de trabajos antes mencionados (Bio GRID, CAE GRID, Meteo GRID). El *middleware* desarrollado en el paquete de trabajo 5, soporta las diversas aplicaciones de los otros paquetes de trabajo ya que es apto para las varias aplicaciones de una GRID computacional.

Transferencia Eficiente de Datos

Desde que se diseñó la tecnología EuroGRID para ser utilizada sobre rutas de Internet, el problema de la transferencia de datos rápida y segura se convirtió en un factor de gran importancia. Por lo que surgen varias consideraciones.

- Optimización del ancho de banda de la transferencia y costo. Puede haber más de una ruta para la transferencia de un archivo punto a punto. Puede haber intercambios en las garantías de ancho de banda y el costo. Por ejemplo, las rutas ATM PVC proporcionan garantías de ancho de banda y muy baja latencia pero son costosas. Una decisión puede tomarse entre una ruta de Internet que ofrece un ancho de banda variable o un vínculo dedicado.
- A prueba de fallos y transferencia cifrada. Si un vínculo de red se cae en una transferencia, los archivos no deben ser modificados sin advertencia. La seguridad de los datos en redes abiertas es una preocupación importante, especialmente para el uso comercial de la GRID.
- El traslape de transferencia y procesamiento. Esta investigación se realiza en términos de procesamiento en tiempo real. Esta transferencia de datos necesita ser una "aplicación conocida".

En general, los requisitos de la aplicación en la transferencia de datos cae en dos categorías: transferencias de estallido, donde la latencia es crítica y transferencia a granel donde el ancho de banda disponible debe ser utilizado. Ambas formas de transferencia son importantes en la computación GRID.

Agente de recursos

Un agente de recursos en una GRID computacional es donde hay una variedad de máquinas con arquitecturas especializadas, necesita ser capaz de acceder y tomar decisiones basadas en información estática y dinámica. La información estática se refiere a los recursos tales como el tipo de arquitectura, rango de compiladores y software, medios de almacenamiento de datos, que permanecen constantes a lo largo de grandes periodos. La información dinámica se refiere a la puesta en marcha de las máquinas, versiones de compiladores y software, planificación actual y estrategias de costo. Éstos pueden cambiar rápidamente, en algunos casos continuamente. El agente de recursos necesita esta información para relacionar a los requisitos computacionales de los usuarios, tradicionalmente representado como un "trabajo", a lo que está disponible en la GRID. También el agente necesita ser capaz de negociar el balance de clientes necesarios para el bajo costo y rápido regreso. El agente tiene que leer y actualizar la información acerca de las concesiones de los clientes y cuentas en las máquinas por la GRID. De esta manera el agente de recursos de una GRID computacional es un concepto sofisticado de software.

Infraestructura ASP

El costo de una tarea enviada a la GRID debe incluir el costo de la licencia de software necesaria para trabajar. Esto aplica particularmente al uso comercial donde el costo de la licencia para el software poderoso puede dominar sobre el costo para la tarea. Puede ser que una organización necesite usar el software en las máquinas grandes pero no podría justificar la compra de una máquina y una licencia del software para su uso exclusivo.

Acoplamiento de la Aplicación

Muchas aplicaciones computacionales a gran escala involucran problemas entrelazados los cuales pueden ser localizados en diferentes máquinas. Un buen ejemplo son los sistemas donde los fluidos interactúan con estructuras sólidas y diferentes soluciones óptimas para el fluido dinámico y las estructuras mecánicas. Otro ejemplo son las simulaciones ambientales las cuales involucran resolución de ecuaciones para el comportamiento de gases, fluidos y sólidos en conjunto con químicos y posibles reacciones biológicas. En las GRID computacionales con un rango de arquitecturas especialistas, diferentes submodelos podrían situarse en máquinas donde las soluciones corren particularmente eficientemente, esto es conocido como afinidad de máquina. Este paquete investigará la integración a la comunicación en el *middleware* para tal acoplamiento de las aplicaciones. Eso necesita tener una sincronía algo débil para el acoplamiento de tal sub-modelo. Así se investigarán interfaces para fijar máquinas y otros recursos.

Acceso Interactivo

Algunos trabajos computacionales requieren acceso interactivo a los recursos. Un ejemplo podría ser correr una simulación con visualización de los resultados proporcionándose como la corrida de la simulación. Esto puede ser extendido a situaciones donde la simulación se dirige según el rendimiento visualizado. Este acercamiento está incrementándose a un importante rango de usuarios de medios computacionales: industriales, científicos y civiles. El prototipo rápido y virtual del nuevo plan de necesidades de productos y los equipos diseñados para ser capaces de observar el comportamiento del producto propuesto y la información visual es muy importante a este proceso. Los procedimientos de emergencia civil, probando simulaciones y los estudios de impacto en el medio ambiente también confían en la interacción del personal experimentado con una simulación basada en observaciones de datos hipotéticos. Muchos de los medios de computación a gran escala han operado tradicionalmente sus máquinas en modo de lote y el modelo actual de UNICORE en el cual está basado. Este paquete de trabajo agregará funcionalidad para permitir la posibilidad de interactuar el uso de la computación y la visualización de los medios.

Actualmente, las aplicaciones del cómputo GRID están muy orientadas al mundo científico. Pero las empresas empiezan a interesarse por estas tecnologías. El futuro de la GRID es que exista una GRID Pública a nivel mundial, en la que cualquier persona puede disponer, si lo necesita, del poder computacional de una supercomputadora. Además, todos podemos ser también proveedores de ciclos de CPU.

3.2.5. DAMIEN

Aplicaciones Distribuidas y Software intermedio para el uso Industrial de Redes Europeas.

El proyecto DAMIEN (*Distributed Application and Middleware for Industrial Use of European Networks*), proviene del área de cómputo de alto desempeño y trata el problema de cómo desarrollar aplicaciones con GRID. En el cómputo de alto desempeño el desarrollador de aplicaciones está acostumbrado a algunos estándares, como MPI y varias herramientas, las cuales facilitan el desarrollo y análisis de su aplicación. El objetivo central del proyecto es por lo tanto

proporcionar estas herramientas conocidas y muy aceptadas por las GRID computacionales. Para alcanzar este objetivo, las herramientas tienen que ser extendidas a las propiedades de la GRID. Además, estas herramientas serán probadas en ambientes industriales usadas en la producción diaria.

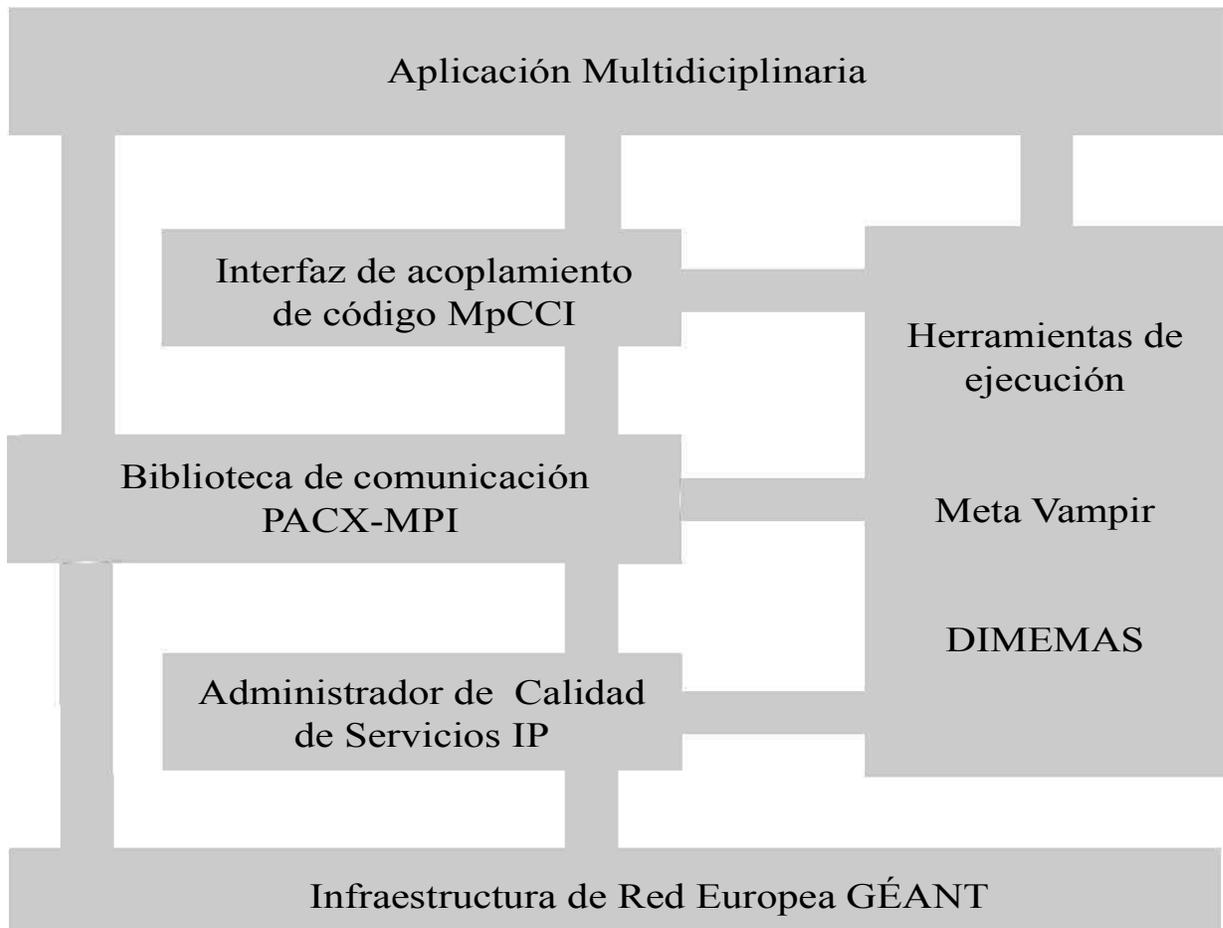


Figura 3.11 Arquitectura DAMIEN

La parte central de la arquitectura DAMIEN son estándares usados por la industria, bibliotecas y herramientas, con los cuales los usuarios del cómputo de alto desempeño están familiarizados. Basado en estas herramientas ya existentes (DIMEMAS, MpCCI, PACX-MPI, Vampir), DAMIEN define las extensiones necesarias para soportar ambientes de cómputo GRID que crean un ambiente de desarrollo de software para el cómputo GRID. Las extensiones pueden ser divididas básicamente en tres partes:

- La integración de una capa de comunicación adicional que tiene que ser introducida para reflejar las características de ambientes distribuidos.
- La integración de calidad-de-servicio (QoS) manejando las herramientas para permitir una administración flexible de los recursos de la red.
- Facilita el uso de herramientas distribuidas y ambientes distribuidos.

Pruebas

La caja de herramientas desarrollada en el marco de DAMIEN está siendo validada en las pruebas basadas en las redes de alta velocidad europeas entre los tres centros de investigación del proyecto (CEPBA, CRIHAN, HLRS). La utilidad de las herramientas también ha sido demostrada en las pruebas de HPC-recursos distribuidos por todo el mundo para el HPC.

Aplicación

EADS CCR proporciona una aplicación multi-física. La aplicación es un nuevo método fuertemente acoplado para simulación vibro-acústica, la cual puede ser usada para simular la propagación de ruido durante la fase de lanzamiento de cohetes o para las simulaciones de reducción de ruido dentro de las cabinas de un avión. Debido al gran número de entidades en la compañía y su distribución geográfica por Europa, este proyecto valida la GRID informática como una solución para los usuarios industriales.

Innovación

Dos innovaciones principales se logran dentro del proyecto de DAMIEN: Primero, el proyecto proporciona el primer ambiente de desarrollo para usuarios finales incluso estándares, bibliotecas y herramientas, con los cuales ellos están acostumbrados al trato de su ambiente de trabajo regular. Instrumentos como Vampir, DIMEMAS, MpCCI son muy aceptados en el área del cómputo paralelo, y también están puestos a disposición ahora en ambientes de GRID. La segunda innovación principal en el proyecto es la integración de calidad en el módulo de servicio en el software de comunicación así como en las herramientas. Ya que la interpretación de aplicaciones GRID están fuertemente bajo la influencia de la red entre máquinas diferentes, la administración de los recursos de la red es una cuestión clave para ambientes de GRID y sus aplicaciones.

Resultados

Las herramientas desarrolladas en el proyecto han sido usadas exitosamente para ejecutar y verificar la aplicación industrial en el banco de pruebas instalado en EADS CCR.

3.3 Construcción de una GRID Interinstitucional en México

En México, bajo el marco del evento "Día Virtual CUDI sobre GRID-Supercómputo" varias de las instituciones participantes integraron una mesa de trabajo en la cual se presentaron los intereses en GRID computacionales, se discutieron ventajas y beneficios de este proyecto así como posibles estrategias para llevarlo a cabo. Fue así como surgió este proyecto apoyado por CUDI, con la intención de estrechar una colaboración que permitiese capitalizar la experiencia que dichas instituciones han adquirido.

El objetivo primordial de este proyecto es la construcción de una infraestructura de GRID nacional, inicialmente a nivel experimental pero con la intención de que posteriormente alcance la madurez que le permita servir como recurso de cómputo de gran escala para la investigación científica.

Esta iniciativa surge de la responsabilidad que asumen las instituciones participantes (UDG, UAM, CINVESTAV, UNAM, CICESE) para proveer herramientas de cómputo de alto rendimiento, que sirvan de instrumento para la investigación en diversas áreas del conocimiento.

Los objetivos generales del proyecto son, al término del mismo, contar con:

- Una infraestructura de hardware, consistente en al menos un sistema de cómputo de alto rendimiento en cada institución, conectado a los demás equipos mediante una red con altas capacidades de ancho de banda y calidad de servicio.
- Un conjunto de servicios (*middleware*) que gestionarán el acceso y uso compartido de la infraestructura de hardware.
- Resultados de aplicaciones ejecutadas en la GRID, que demuestren la factibilidad de su uso.

Uno de los aspectos más importantes de este proyecto es la colaboración entre las instituciones participantes para la construcción de la GRID nacional. Y como parte de esta labor interinstitucional se determinaron los siguientes objetivos específicos:

- Intercambiar las ideas, asimilaciones y experiencias que se han generado gracias a los esfuerzos particulares de cada institución en trabajos relacionados con las GRID.
- Obtener una mayor experiencia y entrenamiento en la implementación y operación de infraestructura de una GRID.
- Establecer una GRID local en cada una de las instituciones participantes.
- Examinar la capacidad de la red en cada institución dado el tráfico generado por aplicaciones de GRID, tanto en comunicación entre procesos como en transferencia de archivos.
- Aprovechar esta experiencia para sugerir las modificaciones necesarias a la red CUDI para lograr un mayor rendimiento de las aplicaciones en GRID.
- Construir una GRID de pequeña escala, que sirva como infraestructura de experimentación de diversos esquemas de operación y uso.
- Evaluar la factibilidad de la construcción de una GRID nacional para uso académico, tanto técnica como organizacionalmente.
- Fomentar, tanto entre los usuarios potenciales como entre los potenciales proveedores de recursos, las tecnologías GRID.

La infraestructura que se tiene contemplada incorporar a la GRID nacional, es heterogénea, de diversas capacidades de cómputo y de almacenamiento, geográficamente distribuida en los Centros de Alto Rendimiento de cada institución participante.

Las GRID computacionales requieren de las capacidades que provee Internet 2, como el ancho de banda y las características de calidad servicio, multicast y transporte diferenciado. Las aplicaciones que se ejecutarán sobre la GRID requieren de una conectividad segura, estable y de alta velocidad que solo Internet 2 puede proporcionar.

En la figura 3.12 se muestra cómo están interconectados los equipos de alto rendimiento participantes en el proyecto GRAMA (GRID Académica Mexicana) dentro de la red de Internet 2.



Figura 3.12 Diagrama de conectividad de GRAMA.

Los proyectos que se presentaron anteriormente son sólo algunos de los proyectos que se están llevando a cabo alrededor del mundo, pero en la tabla 3.1 se listan la mayoría de proyectos en desarrollo, cada uno con su respectiva página Web, en caso de que se desee mayor información.

ASTROGRID	http://www.astrogrid.org/
BIOGRID	http://www.biogrid.jp/
BIOINFORMATICS	http://www.bbsrc.ac.uk/science/initiatives/bep.html
BIRN	http://www.nbirn.net/
CONDOR	http://www.cs.wisc.edu/condor/
CROSSGRID	http://www.crossgrid.org/
DAMIEN	http://www.hlrs.de/organization/pds/projects/damien/
DATAGRID	http://datatag.web.cern.ch/datatag/
DATATAG	http://datatag.web.cern.ch/datatag/
DOE SciDAC	http://www.osti.gov/scidac/
DOE SCIENCE GRID	http://doesciencegrid.org/

Earth System GRID II	http://www.earthsystemgrid.org/
ECOGRID	http://www.buyya.com/ecogrid
EGEE	http://public.eu-egee.org/
ESnet	http://www.es.net/
EUROGRID	http://www.eurogrid.org/
FUSIONGRID	http://www.fusiongrid.org/
GÉANT	http://www.dante.net/geant/
GLOBUS	http://www.globus.org/
GRACE	http://www.grace-ist.org/
GRIDBUS	http://www.gridbus.org/
GriDis	http://web.datagrid.cnr.it/GriDis/jsp/index.jsp
GRIDPP	http://www.gridpp.ac.uk/
GRIP	http://www.grid-interopability.org/
GriPhyN	http://www.griphyn.org/index.php
Inca	http://inca.sdsc.edu/
IPG	http://www.ipg.nasa.gov/
ivDgl	http://www.ivdgl.org/
LEAD	http://lead.ou.edu/
LEGION	http://www.cs.virginia.edu/~legion/
MCNC	http://www.mcnc.org/gcns2/
myGRID	http://www.mygrid.org.uk/
NAREGI	http://www.naregi.org/
National GRID Service (UK)	http://www.ngs.ac.uk/
NEES grid	http://www.neesgrid.org/
NextGRID	http://www.nextgrid.org/
NORDUGRID	http://www.nordugrid.org/
NSF MIDDLEWARE INITIATIVE	http://www.nsf-middleware.org/
Open Science GRID Consortium	http://www.opensciencegrid.org/
PPDG	http://www.ppdg.net/
RealityGRID	http://www.realitygrid.org/
SimDat	http://www.simdat.org/
TERAGRID	http://www.teragrid.org/
UNICORE Plus	http://www.fz-juelich.de/unicoreplus/index.html
WestGRID	http://www.westgrid.ca/home.html

Tabla 3.1. Proyectos GRID en el mundo

4. Implementación de la GRID

En este caso se aplicará un proceso cíclico para desarrollar el producto. En cada ciclo se deciden cuáles funcionalidades desarrollar. En el primero de ellos se diseña e implementa la primera versión del producto; en el segundo ciclo, se incrementan las funcionalidades del producto.

Ciclo 1

Durante el primer ciclo se realizará el trabajo correspondiente a la implementación de los *clusters* necesarios para construir la GRID.

Ciclo 2

Se procederá a la instalación y configuración del software necesario para poder utilizar el conjunto de herramientas de Globus. Para construir la GRID, en este ciclo se interconectarán los *clusters* construidos en el ciclo anterior y se realizará la gestión de sus recursos mediante la herramienta Globus.

4.1 Ciclo 1. Construcción de *clusters*

Dado que se utilizará la herramienta Globus para construir la GRID, el sistema operativo instalado debe ser capaz de soportar dicha herramienta de software. En este caso el sistema operativo que se instalará será Red Hat 9.0, pero puede ser cualquiera de los sistemas que soportan a Globus Toolkit.

Diseño e implementación del *Cluster* tipo *Beowulf*

El sistema *Beowulf* necesita de muchos parámetros de diseño a considerar para su terminación exitosa. En los apartados siguientes se resumen los más significativos.

4.1.1 Diseño

Es importante considerar los aspectos de diseño para tomar ciertas decisiones entre las que destacan:

- Escoger el tipo de los procesadores que van a conformar las CPUs que se van a utilizar y su velocidad de procesamiento.
- La cantidad de memoria RAM, tanto para el servidor como para los nodos
- Capacidades de Disco Duro, para el servidor y clientes
- Tipo de interconexión entre los nodos y el servidor.

En general, si se puede contar con el tipo de recursos que uno desea, sin importar el costo, entonces se piensa en la aplicación en que se va a trabajar y tomando esto como base se decide cuáles de las características antes mencionadas merecen una prioridad mayor; por ejemplo, puede que la aplicación necesite más memoria que disco duro o viceversa.

También existen casos en que no se cuenta con recursos muy amplios, por lo que el *cluster* se construye basándose en el equipo existente, y es en la parte de programación donde se puede o no ajustar la aplicación al sistema

4.1.2 Requerimientos mínimos

CPU

Básicamente un nodo deberá tener como mínimo, una CPU 386 y una tarjeta madre compatible con Intel. Existe la posibilidad de trabajar con otras arquitecturas que soporten Linux. Los procesadores Intel 386 funcionarían pero no tendrán un buen desempeño, debido a que sus velocidades de procesamiento están muy por debajo del requerido.

Memoria

Los requerimientos de memoria se deciden dependiendo del procesamiento de datos en la aplicación. Para cumplir con los requerimientos mínimos del sistema operativo y las aplicaciones cada nodo deberá contar al menos 8MB de memoria RAM.

Disco duro

Usando un espacio de disco centralizado, los nodos se pueden inicializar desde un floppy, un disco duro de poca capacidad o un sistema de archivos compartidos a través de la red. Entonces los nodos pueden acceder a su partición raíz desde un servidor de archivos a través de la red, normalmente usando el *Network File System* (NFS). Esta configuración funciona mejor en un entorno con un gran ancho de banda que agiliza la conexión y que permita un buen rendimiento en el servidor de archivos. Para un mejor rendimiento bajo otras condiciones, cada nodo tendría que tener suficiente espacio en el disco local para el sistema operativo, la memoria virtual y los datos. Cada nodo debería tener un mínimo de 200 MB de espacio en disco.

Red de interconexión

Como mínimo cada nodo tiene que incluir una tarjeta de red ethernet o fast ethernet. Alternativamente, interconexiones de más alto rendimiento, incluyendo las de los tipos Gigabit ethernet y Myrinet, podrían ser usadas en conjunto con CPU's más rápidas.

Requerimientos del servidor

El servidor es recomendable que tenga recursos mayores que los nodos, además de que es el único que contará con un teclado, monitor, mouse (opcional) así como otros dispositivos que se consideren necesarios.

4.1.3 Implementación

Tomando en cuenta los recursos con que se contaban en ese momento y considerando los parámetros de diseño mencionados anteriormente, se prosigue a la implementación del *Sistema Beowulf*.

A continuación se presentan los puntos más representativos de la implementación.

4.1.4 Configuración del servidor y nodos

Antes de instalar el Sistema Operativo se tiene que planificar una serie de aspectos para poder contestar a todas las preguntas durante la instalación. Para realizar esta planificación podemos basarnos en lo siguiente:

- Medios con los que se cuenta. En esta versión de Linux Red Hat 9.0 se detectan automáticamente todos los periféricos sin necesidad de pedirnos información adicional, pero en algunos casos no, y es necesario especificarlos.

Se debe tener a la mano las especificaciones relativas a:

- Modelo de la tarjeta de video y memoria que contiene
 - Tipo de monitor y frecuencias de barrido horizontal y vertical
 - Modelo de dispositivos y controladores, del ratón, tarjeta de red y dispositivos de hardware con que se cuente y deban funcionar en los servidores de Linux.
- Nombre del sistema, así como el software y servicios que se van a instalar (Colocar las herramientas de programación, desarrollo y *clustering*)
 - Definir cómo se va a distribuir el disco ya que al momento de la instalación se requiere y es mejor planearlo con anticipación. Se recomienda dejar el mayor espacio posible a (/) raíz, aquí se creará el sistema de archivos para los nodos y en segundo lugar a la partición /usr, donde se instalarán las herramientas o software de paralelización como PVM.

Posteriormente de haber planeado lo anterior se procede a la instalación del Sistema Operativo Linux Red Hat en las computadoras que van a ser servidores.

Ya instalado el sistema operativo se procede a la configuración del equipo, es decir, configuración de los ambientes de red y los diferentes servicios que prestarán los servidores a los nodos.

4.1.5 Configuración inicial

Para configurar los *clusters* se deben hacer modificaciones a los siguientes archivos tanto en el servidor como para el o los nodo(s):

Configuración de la tarjeta de red

En el directorio `/etc/sysconfig/network-scripts` debe existir un archivo con el nombre `ifcfg-ethx`, donde se darán de alta las características propias de cada tarjeta.

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-eth0
```

El contenido del archivo será parecido al siguiente:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
BROADCAST=192.168.1.255
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
```

Para verificar si la tarjeta responde, se utiliza el comando *ping* para probar la respuesta de la tarjeta.

```
# ping eth0
# ping 192.168.1.1
```

En el archivo `/etc/sysconfig/network` se puede configurar el nombre de la máquina, el dominio y la puerta de enlace:

```
#cd ..
#vi network
```

```
NETWORKING=yes
HOSTNAME=SERVIDOR1
GATEWAY=132.248.54.254
NISDOMAIN=CLUSTER1
```

Editar el archivo `/etc/hosts`. En este archivo se colocarán la dirección IP, el nombre de la máquina con su dominio, y el alias por el cual se conocerá a la máquina.

```
127.0.0.1    localhost.localdomain    localhost
192.168.1.1  SERVIDOR1                SERVIDOR1
```

Después de haber realizado los cambios a los archivos anteriores es necesario reiniciarlos para que tengan efecto, lo cual se realiza de la siguiente manera:

```
#cd /etc/rc.d/init.d/
#./network restart
```

Para probar que la tarjeta funcione correctamente y que la red fue configurada con éxito se hace un *ping* para verificar la conexión con el o los nodos conectado(s) a cada servidor, escribiendo la dirección de cada nodo:

```
#ping 192.168.1.2
```

4.1.6 Configuración de servidor NFS

Para la configuración del servidor hay que seguir básicamente dos pasos:

El primer paso es crear el archivo `/etc/exports`. Este archivo define qué partes del disco del servidor se comparten con el resto de la red, es decir, lista las particiones compartidas, las máquinas con las que comparte y con qué permisos.

```
# vi /etc/exports
```

Al final de este archivo se agregan las líneas:

```
/home 192.168.1.0/255.255.255.0(rw,no_root_squash)
/usr 192.168.1.0/255.255.255.0(rw,no_root_squash)
```

Al realizar este cambio se debe inicializar el servicio *nfs* para que los cambios tengan efecto en el servidor

```
# cd /rc.d/init.d
# ./nfs restart
```

Configuración del cliente

Para montar un sistema de archivos, se utiliza el comando *mount* por medio del archivo `/etc/fstab`. Este archivo permite que se monten los sistemas de archivos remotos al arranque.

```
# cd /etc
# vi fstab
```

se agrega al final, la línea:

```
192.168.1.1:/home      /home      nfs     defaults    0 0
192.168.1.1:/usr      /usr       nfs     defaults    0 0
```

Al igual que para el caso anterior se debe reiniciar el servicio *nfs*, de la misma manera.

Para montar el home del servidor en el cliente:

```
mount -t nfs 192.168.1.1:/home /home
mount -t nfs 192.168.1.1:/usr /usr
```

4.1.7 Configuración del servidor NIS

Los servicios de NIS son simplemente una base de datos en un esquema cliente - servidor que los clientes pueden consultar. Consta de una serie de tablas independientes. Cada tabla se origina como un archivo de texto, como el */etc/passwd*, el cual tiene una naturaleza tabular y tiene al menos una columna que es la única por cada archivo (lo que permite generar una base de datos de valores *clave/par*). NIS accede a esas tablas por nombre y permite búsquedas de dos maneras:

- Listado de la tabla
- Presentación de una entrada específica basada en una búsqueda por una clave dada.

Una vez establecidas las bases de datos en el servidor, los clientes pueden buscar en el servidor las entradas de la base de datos. Normalmente esto ocurre cuando un cliente se configura para que busque en un mapa NIS cuando no encuentra una entrada en su base de datos local. Una máquina puede tener sólo las entradas que necesite para que el sistema trabaje en modo usuario único (cuando no hay conectividad por red), por ejemplo, con el archivo */etc/passwd*. Cuando un programa hace una petición de búsqueda de la información de contraseña de un usuario, el cliente comprueba su archivo *passwd* local y si el usuario no existe allí, entonces el cliente hace la petición a su servidor NIS para buscar la entrada correspondiente en la tabla de contraseñas. Si NIS tiene la entrada, volverá al cliente y al programa que pidió la información. El propio programa no se da cuenta de que ha usado NIS. Lo mismo es cierto si el mapa NIS devuelve la respuesta de que la entrada de contraseña del usuario no existe. El programa debería recibir la información sin que sepa qué ha ocurrido.

El servidor NIS proporcionará la facilidad de poder tener una administración centralizada para la configuración de ciertos archivos para todas las máquinas que se encuentran en la red.

Los archivos de configuración sobre los cuales se generan los mapas son:

/etc/group, */etc/hosts*, */etc/netgroup*, */etc/protocols*, */etc/rpc*, */etc/networks*, */etc/services*,
/etc/passwd.

Los mapas pueden generarse de diferentes maneras al ordenarlos por un registro diferente cada uno. Así, para el archivo *passwd*, se pueden generar los siguientes mapas:

passwd.byname (ordenado por nombre) *passwd.bynumber* (ordenado por número) *passwd.byuid*
(ordenado por UID de usuario)

NIS puede tener un único servidor autorizado donde permanecen los archivos de datos originales. Este servidor autorizado se llama servidor NIS maestro. Si la carga en la red es muy grande, se puede distribuir en varios servidores NIS secundarios o esclavos.

El servidor NIS primario establece dominios que abarcan el área en que trabajan. El servidor de NIS sólo envía los datos que le pide el cliente, pero no realiza autenticación, de ese paso se encargan las máquinas individualmente.

Las características que presentan los dominios en NIS son las siguientes:

- Todas las máquinas pertenecientes a un dominio tienen el mismo nombre de dominio.
- Un dominio NIS tiene un único servidor maestro.
- Un dominio de NIS puede tener o no servidores esclavos.
- Los mapas con igual nombre pero distinto dominio son diferentes.
- El dominio de NIS es diferente a los dominios que se definen en Internet.
- Los dominios suelen aparecer en los sistemas como subdirectorios del directorio donde se localiza *yp*.

Configuración del servidor NIS maestro

La forma de definir un dominio NIS es con el comando *domainname*:

```
domainname dominio
```

Donde "dominio" es el dominio NIS empleado. Es necesario que cada vez que se reinicie el sistema, el dominio se establezca. Se puede añadir la siguiente línea al archivo */etc/sysconfig/network*:

```
NIS_DOMAIN=nombre_de_dominio
```

La configuración del nombre de dominio debe de ocurrir antes que se inicien los servicios del NIS.

El nombre del dominio que utilizamos será *cluster1* y *cluster2*

Arranque de *ypserv*.

EL programa *ypserv* es el responsable del manejo de las peticiones que se hacen hacia el servidor NIS. Para no reiniciar el servidor, es posible iniciar el demonio de *ypserv* por medio del comando:

```
$ /etc/rc.d/INIT.d/ypserv start
```

Edición del archivo *Makefile*

Para que el NIS funcione debidamente, es necesario crear los mapas de los cuales se servirá para realizar su función. Para esto, en el directorio */var/yp* se encuentra un archivo llamado *Makefile*. Este archivo lista los archivos que se compartirán mediante NIS, así como algunos parámetros adicionales sobre cómo compartirlos.

Uso de *ypinit*

Una vez que se tenga preparado el *Makefile* se inicializa el servidor YP usando el comando:
/usr/lib/yp/ypinit -m

donde la opción *-m* le dice a *ypinit* que configure el sistema como servidor de NIS maestro.

Configuración de un cliente NIS

El archivo `/etc/yp.conf` contiene la información necesaria para que el demonio del cliente, `ypbind`, arranque y encuentre el servidor NIS. El cliente puede encontrar el servidor de diferentes maneras:

- El uso de broadcast
- La especificación de nombre de la máquina del servidor

Se edita el archivo `/etc/yp.conf` y se le agrega la siguiente línea

```
ypserver nombre_servidor (ypserver 192.168.1.1)
```

Actualización de mapas

Una vez que se tienen configurados el servidor y el cliente, se puede probar que funcionan correctamente por medio de la utilidad `ypcat`, que mostrará en pantalla el mapa que se comparte. Para esto se debe de escoger un mapa que se comparta en el dominio, como por ejemplo, `pas5wd`.

Una vez que se tiene todo esto configurado, al dar de alta a un usuario nuevo, se hará directamente desde el servidor, y ya no nodo por nodo. Para actualizar todos los mapas con los nuevos datos, sólo se deberá ejecutar `make` dentro del directorio `/var/yp` con lo que se actualizarán todos los mapas que se comparten.

4.1.8 Instalación de PVM (Parallel Virtual Machine)

Al realizar la instalación del sistema operativo, entre los paquetes que se tienen dentro de las opciones para instalar, está PVM. Para verificar que el paquete ya esté instalado, se puede utilizar el comando:

```
# rpm -qi pvm
```

lo que devolverá una de dos posibles salidas:

1. La información acerca del paquete, o la lista de los paquetes instalados con los que coincide en su nombre la cadena dada.
2. Un mensaje que indica que el paquete no está instalado.

Cabe mencionar que este comando sólo nos proporcionará esta información si se realizó la instalación por medio de un paquete rpm.

Si no está instalado se puede instalar por rpm de la siguiente manera:

```
# rpm -i pvm-3.4.3-4.i386.rpm  
# rpm -i make_pvm-3.78.1-2.i386.rpm  
# rpm -i pvm-gui-3.4.3-4.i386.rpm
```

Una vez instalado entonces sólo hay que agregar las siguientes líneas en el archivo `/etc/skel/.bashrc`

```
export PVM_ROOT=/usr/share/pvm3
export PVM_ARCH=LINUX
```

Al crearlo en `/etc/skel/.bashrc` se permite que al agregar más usuarios al sistema, estas variables se copien a su directorio Home.

Si se tienen cuentas ya creadas, se deben agregar estas líneas al `.bashrc` de cada usuario.

Para que funcione el PVM, es necesario que se puedan ejecutar comandos de manera remota por medio del comando `rsh`. Para ello se agrega en el directorio `/etc` el archivo `hosts.equiv`, que contiene los nombres y direcciones de las máquinas en las cuales pueden ejecutarse comandos de manera remota. De esta manera cada vez que se agrega un nuevo nodo habrá que agregar su nombre y dirección IP a este archivo.

4.1.9 Instalación de MPICH (MPI/Chameleon)

Una vez que se tiene la última versión de *mpich* continúan los siguientes pasos:

- Se descomprime el archivo de instalación de *mpich* en el directorio compartido `/usr/local/mpich`
- Se va al directorio `mpich` y se teclea:
`./configure --arch=LINUX device =ch_p4 --rsh=ssh`
- Se teclea `make`
- Se teclea `util/tstmachines`
- Se va al directorio `mpich/util/machines`
- Se edita `machines.LINUX` y se borran todas las líneas del servidor y luego se teclean los nombres de todos los nodos
- Se va a `mpich/util`
- Se teclea `./tstmachines`

Compilación y ejecución de los programas

Los programas que utilizan las bibliotecas de MPI deben ser compilados antes de poder ser ejecutados; el programa que permite compilar programas de MPI con base en el lenguaje C se llama `mpicc`. Para compilar los programas con MPICH, se teclea lo siguiente en la línea de comandos:

```
# mpicc -lm -o nombre_del_programa_de_salida nombre_del_programa_en_c
```

Después, para ejecutar el programa de salida que se obtuvo mediante `mpicc`, se puede utilizar el guión de comandos `mpirun`. Los tipos de ejecución del programa de salida se describen a continuación:

```
# mpirun n a-b nombre_del_programa parámetros
# mpirun N nombre_del_programa parámetros
# mpitun -c NUM nombre_del_programa parámetros
```

donde:

- n:** indica que se va a utilizar un intervalo de nodos
- a:** indica el primer nodo
- b:** indica el último nodo
- N:** indica que se utilizarán todos los nodos
- c:** indica que se ejecutarán copias del programa
- NUM:** indica el número de copias del programa que se van a ejecutar
- nombre_programa:** es el archivo ejecutable en cuestión
- parámetros:** son los parámetros del programa

4.2 Ciclo 2. Construcción de una GRID usando Globus Toolkit 4.0.1 (GT4)

Para construir una GRID se utiliza un software que permita la gestión de recursos de los nodos de la GRID. Globus Toolkit es el middleware de código abierto mejor conocido actualmente. La versión más reciente de este Toolkit es la 4.0.1. El juego de herramientas de Globus permite realizar la administración de dichos recursos. Las principales divisiones de Globus son:

- Administración de tareas
- Seguridad
- Descubrimiento de Recursos

El GT4 se compone de dos partes:

1. Componentes PreWS
2. Componentes WS

Los componentes WS (Web Service) son completamente independientes de la plataforma en la que fueron desarrollados usando Java y están basados en WSRF Framework. Contiene WS-GRAM, WS-MDS, CAS, RLS, etc.

Los componentes Pre-WS son completamente dependientes de la plataforma UNIX, como GRAM, GIIS, GRIS, MDS, etc.

Para elegir el sistema operativo en que se desea instalar Globus, se deben considerar los servicios que se desean instalar y verificar que sean compatibles con ese sistema operativo elegido.

Para Windows no pueden instalarse todos los paquetes de Globus, sólo el software basado en Java. En tanto que para Unix además de esos paquetes, se pueden instalar los de seguridad para los servicios Web (WSRF), el GridFTP (Grid File Transfer Protocol), el administrador de recursos y servicios de ejecución (GRAM, Globus Resource Allocation Manager), el servicio de réplica de posición (RLS) y los servicios de información (MDS, Metacomputing Directory

Service). Los servicios de WSRF y GRAM requieren de una infraestructura que sólo puede ser ejecutada en Unix.

Debido a que la mayoría de las herramientas de software de Globus pueden ser utilizadas en un sistema operativo Unix, la GRID que se implementará en esta tesis se hará bajo una versión y distribución de dicho sistema: RedHat 9.0, que es la distribución instalada en los *clusters* que formarán parte de esta GRID. Además de que en esta tesis sólo se implementarán los Pre-WS.

Planeación

Los módulos en que se dividirá este ciclo son:

1. Instalación de Globus Toolkit y del software necesario para su funcionamiento.
1. Instalación del módulo de seguridad, el cual incluye obtener el CA (Certificate Authority) para confirmar, obtener el certificado de los hosts, la identificación de los usuarios y la creación del archivo de mapa de la GRID.
2. Usar CAs
3. Iniciar el servidor de GridFTP
4. Iniciar Globus Gatekeeper

4.2.1 Instalación del Globus Toolkit

Descargando el GT4

El GT4 puede ser descargado de <http://www.globus.org/toolkit/downloads/4.0.1>, el paquete que debe descargarse es `gt4.0.1-all-source-installer.tar.gz`. La distribución del código fuente es compatible con varios sistemas operativos basados en UNIX como:

- MAC OS X
- Solaris APARC
- Linux (RedHat, Fedora, Debian, Core, SuSE, FreeBSD)
- Alpha Machines (HP-UX y AIX)

Para instalar el GT4 en una máquina con Windows se requiere descargar `ws-core-4.0.1-bin.zip`. En el sistema operativo Windows sólo funcionan los componentes WS del GT4.

Antes de la Instalación

Antes de comenzar con la instalación, se debe preparar efectivamente el sistema como se indica a continuación:

- 1) Verificar la plataforma y prerequisites compatibles
- 2) Crear un usuario llamado “globus” en la máquina local, globus debe ser un usuario común en todos los nodos del *cluster*.

- 3) Es necesario crear un directorio en blanco como *root* y luego cambiar al usuario de *globus*, para que éste último tenga todos los permisos y control sobre dicho directorio.

```
# mkdir /usr/local/globus
# chown globus:globus /usr/local/globus
```

- 4) Si se instala dentro de un NFS, será más fácil compartir la misma instalación de Globus entre varios sistemas.
- 5) Asignar la variable de ambiente básica: *GLOBUS_LOCATION* al directorio seleccionado para la instalación del GT4.

```
$ cat >> .cshrc
export $GLOBUS_LOCATION /usr/local/globus
$ source .cshrc
```

- 6) Crear una carpeta para la versión compilada del GT4 y extraer el GT4 en ella.

Instalando los prerrequisitos del GT4

La lista de requerimientos necesarios para la instalación de middleware se muestra en la siguiente tabla

Programas	Página Web	Descripción	Requerido
GNU tar,sed,Make,gcc	http://www.gnu.org	Herramientas básicas.	Sí
Ant 1.5.1+	http://ant.apache.org/	Herramienta de construcción independiente de la plataforma.	Sí
J2SE 1.4.2+	http://java.sun.com/j2se	Edición estándar de java.	Sí
zlib 1.1.4+	http://www.gzip.org	Bibliotecas de compresión requeridas por GPT.	Sí
sudo	http://www.courtesan.com/sudo	Herramienta que permite a los usuarios normales ejecutar comandos de nivel de superusuario usados por GRAM	Sí
JDBC compliant database	http://www.postgresql.org	Recomendada para RFT, CAS, etc.	Sí
IODBC	http://www.iodbc.org	Para RLSC (<i>Replica Location Service</i>).	No
Tomcat	http://jakarta.apache.org/tomcat/	Opcional para necesidades de tiempo de ejecución.	No
Junit	http://www.junit.org	Necesario para pruebas unitarias de <i>ant</i> .	No
Servidor de correo		Necesario para enviar y recibir certificados.	No

Tabla 4.1 Lista de requerimientos para el GT4

Para el caso de la distribución de Linux utilizada en los *clusters*, algunos de los requerimientos de software pueden instalarse al mismo tiempo que se instala el sistema operativo.

Instalación y configuración de Apache Ant

El paquete Apache Ant es una herramienta de construcción basada en Java. En teoría es algo como **make**. En vez de un modelo que se extiende comandos basados en el intérprete de comandos, Ant se extiende usando clases Java. En vez de escribir comandos del intérprete, los archivos de configuración se basan en XML, llamando a un árbol de objetivos en el que se ejecutan diversas tareas. Cada tarea es ejecutada por un objeto que implementa una interfaz de tarea particular.

Ant es una herramienta de construcción independiente de la plataforma requerida, Windows o Unix, antes de la instalación del GT4. Toda la compilación de los componentes de WS está basada en Ant.

Los pasos que deben seguirse para su instalación y configuración son los siguientes:

- 1) Descargar la distribución binaria de Ant
- 2) Extraerla en la carpeta donde se instalará

```
root# cd /usr/local
root# mkdir apache-ant
root# tar -zxvf apache-ant.tar.gz
```

- 3) Asignar la variable de ambiente ANT_HOME a /usr/local/apache_ant (localización de la instalación)

```
root# export ANT_HOME=/usr/local/apache_ant
```

Instalación y configuración de J2SE

J2SE es importante para todos los servicios basados en Java. Los pasos a seguir para poder instalar y configurar java son los siguientes:

- 1) Descargar el J2SE Binario para la plataforma que se utilizará y extraerlo en una carpeta

```
root# cd /usr/local
root# mkdir j2sdk
root# cd j2sdk
root# tar -zxvf j2sdk-linux.tar.gz
```

- 2) Asignar la variable de ambiente JAVA_HOME (preferentemente en .cshrc)

```
root# export JAVA_HOME=/usr/local/j2sdk1.4.2_09
```

Instalación y configuración de PostgreSQL

PostgreSQL es una implementación de código abierto del sistema RDBMS que soporta comunicación TCP/IP. Los pasos a seguir para poder instalar PostgreSQL son los siguientes:

- 1) Descarga PostgreSQL
- 2) Extraerlo en una carpeta
- 3) Configurarlos
- 4) Hacer make y make install

```
root# cd /usr/local
root# mkdir pgsq
root# cd $HOME
root# tar -zxvf postgresql-base-8.0.3.tar.gz
root# cd postgresql-base-8.0.3
root# ./configure --prefix=/usr/local/pgsq
root# make
root# make install
```

La configuración de PostgreSQL necesita los siguientes pasos:

- 1) Agregar un nuevo usuario “postgres” con privilegios normales.
- 2) Crear como *root* una carpeta para almacenar la base de datos y luego cambiar al usuario de postgres, para que este último tenga todos los permisos y control sobre dicha carpeta.

```
# mkdir /usr/local/pgsq/data
# chown postgres /usr/local/pgsq/data
# su - postgres
```

- 3) Asignar la variable de ambiente PGDATA=/usr/local/pgsq/data y también actualizar la variable PATH en .csh

```
# export PGDATA=/usr/local/pgsq/data
# export PATH=${PATH}:/usr/local/pgsq/bin
```

- 4) Ejecutar el comando \$ /usr/local/pgsq/bin/initdb, el cual debe ser exitoso y sin ningún error. Este comando crea el archivo postgresql.conf en el directorio data.
- 5) Crear un nuevo archivo en /etc/INIT.d llamado postgresql.

Agregar la siguiente línea en ese archivo:

```
su -postgres -c “/usr/local/pgslq/bin/pg-ctl Stara -o “-i” -l logfile -D /usr/local/pgsq/data”
```

- 6) Crear una liga en el directorio /etc/rc3.d como

```
# ln -s ../init.d/postgresql S99postgresql
```

7) Hacer a postgresql como ejecutable usando estos comandos

```
# chmod u+x postgresql
# chown postgres postgresql
```

Sudo

Sudo es usado por usuarios ordinarios para ejecutar comandos que requieren privilegios de superusuario. Es un requerimiento importante para WS-GRAM, así que configurar sudo es muy importante para la ejecución exitosa de los trabajos usando WS-GRAM. Sudo puede instalarse como se describe a continuación.

1) Descargar el archivo binario y extraerlo en un directorio apropiado

2) Editar el archivo `/usr/sbin/visudo` y agregar la siguiente entrada:

```
# Globus GRAM

globus ALL=(username1,username2) NOPASSWD: /opt/globus/GT4.0.1/libexec/globus-
gridmap-and-execute -g
/etc/grid-security/grid-mapfile
/opt/globus/GT4.0.1/libexec/globus-job-manager-script.pl *

globus ALL=(username1,username2) NOPASSWD:
/opt/globus/GT4.0.1/libexec/globus-gridmap-and-execute -g
/etc/grid-security/grid-mapfile
/opt/globus/GT4.0.1/libexec/globus-gram-local-proxy-tool *
```

Las entradas anteriores deben escribirse solo en dos líneas.

3) visudo reportará si hay algún error en las entradas

4.2.2 Configuración e instalación del GT4

Una vez que se han terminado de instalar y configurar todos los prerequisites, se puede continuar con la configuración de Globus.

Para configurar el GT4, después de que se haya descargado el archivo tar, deben seguirse los siguientes pasos:

Como usuario globus, y dentro de la carpeta donde se descomprimirán los archivos:

```
globus$ cd globus_compilado
globus$ tar -zxvf gt4.0.1-all-source-installer.tar.gz
```

Se ejecutan los siguientes comandos:

```
globus$ ./configure - -prefix=GLOBUS_LOCATION
globus$ make | tee build.log
globus$ make install
```

Instalando MPICH-G2

MPICH-G2 es sólo un parche para la versión más reciente de MPICH y puede ser usada para enviar las tareas que pueden ejecutarse en paralelo en una GRID. MPICH puede descargarse de la página <http://www-unix.mcs.anl.gov/mpi/mpich/>. Los comandos para instalar MPICH para Globus son los siguientes:

```
$ configure --with-device=globus2:flavor=gcc32dbg
--prefix=<install location>
$ make | tee mpi.log
$ make install
```

Se puede instalar por ejemplo en /usr/local/mpich-1.2.4/globus2.

Antes de continuar con la configuración de Globus es necesario que todas las rutas estén correctas en el archivo .cshrc

Una vez que se terminó de instalar el Globus Toolkit, el siguiente paso es instalar el módulo de seguridad. Para instalar el módulo de seguridad se debe iniciar un servidor GridFTP, configurar una base de datos para RFT y configurar WS-GRAM. También se debe iniciar el demonio de GSI-OpenSSH, instalar un servidor MyProxy, ejecutar RLS y usar CAS (Certificate authority).

1.1.1 Implementando la seguridad

Uno de los pilares de la infraestructura de la GRID es la seguridad. El nivel de seguridad debe ser muy alto antes de permitir el uso de cualquier recurso público de la GRID. Es bien sabido que muchas páginas Web muestran una firma certificadora que garantiza al usuario que el sitio es más confiable. Esos certificados son necesarios para probar que esos sitios son seguros. De manera similar, en la GRID, cada recurso debe tener certificados apropiados para hacerlo más confiable. Estos certificados pueden ser emitidos por una organización autorizada la cual es más confiable para el usuario y también la autoridad debe conocer bien al usuario. Esta organización es en general llamada Autoridad de Certificación (CA). Actualmente, todo sistema de seguridad debe estar dirigido por una Autoridad de Certificación.

Creando una nueva Autoridad de Certificación

La Autoridad de Certificación es el intercambio para emitir certificados tanto para los equipos de cómputo como para los usuarios de la GRID. La Autoridad de Certificación puede ser una organización externa muy confiable o incluso una Autoridad de Certificación personal la cual

puede ser creada usando el software llamado SimpleCA que provee Globus. Cada sistema en la GRID debe seguir la misma CA y un solo sistema puede tener múltiples CAs para formar parte de dos o más GRID.

A continuación se listan los pasos a seguir para instalar el SimpleCA que provee Globus. Estos pasos se siguen para la instalación inicial de la GRID. Si lo que se desea es formar parte de una GRID existente, entonces no debe instalarse el SimpleCA, pasar a ***Confiability y uso de una CA existente***

- 1) Ejecutar el comando `$GLOBUS_LOCATION/setup/globus/setup-simple-ca` como usuario globus.

```
globus$ bash $GLOBUS_LOCATION/setup/globus/setup-simple-ca
```

- 2) `GPT_LOCATION` tiene que ser especificada antes de ejecutar el comando, pero `GPT_LOCATION` se asignará automáticamente a la ubicación correcta `$GLOBUS_LOCATION`.
- 3) El host donde se instale el SimpleCA actuará como el CA principal de la GRID.
- 4) El sujeto de CA instalado en el host es "**cn=Globus Simple CA, ou=simpleCA, ou=nombrehost, o=GRID**". De manera similar se puede especificar cualquier sujeto que se desee.
- 5) Enviar un correo electrónico al administrador del SimpleCA. Ese correo puede ser usado para enviar los certificados que pidan los clientes.
- 6) La fecha de expiración de los certificados emitidos por este CA es de 5 años y puede cambiar en el futuro.
- 7) La frase *PEM pass* es la contraseña o la clave que se usa para identificar al CA. Es importante ya que tiene que ser emitido por el administrador siempre que firme los certificados.
- 3) El *hash* (firma) del CA se despliega al resumen de esta instalación. Se debe anotar el *hash* que se desplegó, el cual se usará para configurar el CA en otros sistemas.
- 9) El paso final es instalar la Infraestructura de Seguridad de la GRID (GSI).

```
globus$ cd $GLOBUS_LOCATION/setup/simpleca-<hash>/
globus$ bash setup-gsi --default
```

Confiability y uso de una CA existente

Confiar en un CA requiere tomar una buena decisión. Si un sistema necesita participar en una GRID externa entonces debe confiar en el CA en que ya confían todos los sistemas de la GRID. Para unirse a una GRID existente, debe bajarse el archivo *tar* que provea la GRID. Y hacer la instalación de GSI en la máquina.

Pasos a seguir para la instalación del CA:

- 1) Bajar el archivo `.tar` que proporcione la GRID existente y utilizar el comando `gpt-build <nombre_archivo> gcc32dbg`.
- 2) y posteriormente `cd $GLOBUS_LOCATION/setup/<archivo generado por gpt-build>`
`bash setup-gsi –default`

Obtención de los certificados requeridos por el CA

GT actualmente sigue la autenticación mutua para la ejecución del trabajo y sumisión del trabajo de una máquina a otra. La autenticación mutua requiere de certificados de seguridad para ser comparados. Estos certificados son de tres tipos:

➤ Certificados del Host

Estos certificados deben estar en cada sistema que forme parte de la GRID. Estos certificados son archivos de sólo lectura incluso para el root y pueden localizarse en `/etc/grid-security` o `$GLOBUS_LOCATION/etc`. El lugar más seguro es `/etc/grid-security` que se crea durante la instalación del simpleCA.

➤ Certificados de usuario

Estos certificados se usan para identificar a un usuario particular en la GRID. Los certificados del usuario, por defecto, se ponen bajo el directorio `$(HOME)/globus`.

➤ Certificados del proxy

Éstos son certificados cortos que están abiertamente disponibles y se usan para la autenticación mutua. Son necesarios para cada operación en Globus y normalmente se ponen en el directorio de `/tmp`.

Todos estos certificados están protegidos por sus archivos clave correspondientes. El certificado del proxy es similar al certificado de usuario comparten el archivo clave del certificado del usuario.

Obteniendo el certificado del host

El certificado del Host se obtiene como sigue:

- 1) Como root ejecutar, `# grid-cert-request –host ‘hostname’`
- 2) Se crean tres archivos en el directorio `/etc/grid-security`
 - a. `hostcert.pem` – archivo vacío para ser reemplazado con el certificado del CA
 - b. `hostkey.pem` – lleva a cabo la frase de paso en formato encriptado en RSA 1024-bit.
 - c. `hostcert_request.pem` – para ser enviado al CA para un certificado firmado.

- 3) Posteriormente, enviar por correo electrónico el `hostcert_request` al correo electrónico del CA.
- 4) Firmando el Certificado del Host por el CA:
 - d. `grid-ca-sign -en hostcert_request.pem - salida hostsigned.pem`
 - e. enviar por correo al cliente `< cat hostsigned.pem`
- 5) Después de eso, copiar el certificado firmado `hostsigned.pem` en `/etc/grid-security/hostcert.pem`.

Obteniendo el certificado de usuario

Los certificados de usuario se obtienen como sigue:

- 1) Todos los usuarios necesitan un certificado de usuario para correr trabajos de Globus. Primero, es necesario entrar al sistema con el *login* de usuario.
- 2) Escribir, **grid-cert-request** – para pedir su nombre DN deseado y frase de paso. DN no es nada más que el Nombre Distinguido similar al *username* en un sistema local.
- 3) Creará tres archivos en el directorio `$(HOME)/.globus`.
 - a. `usercert.pem`
 - b. `userkey.pem`
 - c. `usercert_request.pem`
- 4) Los pasos son similares a los de la obtención del certificado del host.
- 5) Eso es, enviar por correo electrónico el `usercert_request` al CA y devolver el certificado firmado y copiarlo en `$(HOME)/.globus/usercert.pem`.

Obteniendo el certificado del proxy

Los certificados del proxy están en el nivel más bajo de proceso de autenticación.

También puede usarse para probar si la seguridad entera se configuró correctamente. El comando para hacerlo es el siguiente, \$ **grid-proxy-init -debug -verify**

El éxito de este comando significa que el GSI está configurado perfectamente y la seguridad no será ningún problema para este usuario en particular. El paso final es asociar el DN con un nombre de usuario local en la máquina.

Acerca del `grid-mapfile`

El `grid-mapfile` es un archivo que contiene la información de varios DN que pueden usar ese sistema como un recurso de la GRID, asociado con el nombre del usuario local.

La manera de agregar entradas en el grid-mapfile es la siguiente:

```
grid-mapfile-add-entry -dn <distinguished-name> -ln <local-name>
```

Se puede verificar la consistencia del grid-mapfile usando el comando:

```
root# bash $GLOBUS_LOCATION/sbin/grid-mapfile-check-consistency
```

Creando el contenedor de certificados y claves

Ya que sólo *root* tiene acceso a los certificados del *host*, algunos de los servicios de la GRID necesitan un certificado de *host* por separado para una ejecución exitosa. Un servicio es el ContainerRegistryService el cual es el mecanismo de arranque para todos los otros servicios cuando *globus-start-container* es invocado.

- 1) Entrar como *root*, y hacer un *containercert.pem* y *containerkey.pem*

```
root# cd /etc/grid-security
root# cp hostkey.pem containerkey.pem
root# cp hostcert.pem containercert.pem
root# chown globus:globus containerkey.pem containercert.pem
```

- 2) De manera similar, para algunos servicios definidos por el usuario, un certificado del *host* tiene que ser creado con el nombre del servicio en lugar del nombre del *host* en el *filename*.

4.2.4 Configurando GridFTP y Globus Gatekeeper

Existen dos servicios básicos de la GRID como parte de los Pre-WS de Globus. Ellos son:

1. GridFTP
2. Globus Gatekeeper

GridFTP es el FTP por default usado para la transferencia de archivos entre los *hosts* de la GRID. Globus Gatekeeper es la que se encarga de verificar y administrar la seguridad entre la sumisión y ejecución del trabajo.

Configurando inetd o xinetd

En lugar de iniciar el servidor del GridFTP y gatekeeper cada vez que sea necesario, se puede sólo agregar como un demonio dentro de los scripts *init*. Para realizar eso, se deben seguir los siguientes pasos:

- 1) *gsiftp* está configurado por default en el puerto 2811. *gsigatekeeper* está configurado por default en el puerto 2119. Si no está allí, debe teclearse *gsiftp* <TAB> después el número de puerto deseado y el protocolo a usar.

- 2) Si se configurará el gridftp y gatekeeper para iniciar como un proceso inetd, entonces se deben seguir los siguientes pasos:
- 3) Como usuario *root*, entrar al directorio */etc*.
- 4) Editar el archivo *inetd.conf* y agregar las entradas que se dan a continuación.

Un ejemplo de *inetd.conf*

```
gsiftftp stream tcp nowait root /usr/bin/env
env GLOBUS_LOCATION=/usr/local/globus
LD_LIBRARY_PATH=/usr/local/globus/lib
/usr/local/globus/sbin/globus-gridftp-server -i
```

```
gsigatekeeper stream tcp nowait root /usr/bin/env env
LD_LIBRARY_PATH=/usr/local/globus/lib
/usr/local/globus/sbin/globus-gatekeeper-conf
/usr/local/globus/etc/globus-gatekeeper.conf
```

Los ejemplos anteriores deben estar escritos sólo en dos líneas. Si el archivo se actualiza mientras se está ejecutando *inetd*, se tiene que reiniciar *inetd* para usar el *archivo inetd.conf* actualizado, para hacerlo se deben escribir los siguientes comandos:

```
# ps -aux | grep inetd
# kill -HUP (process-number)
```

Si se desea actualizar el archivo *xinetd*, entonces se deben agregar las siguientes entradas al archivo de configuración:

Crear un archivo llamado *gsiftftp* en */etc/xinetd.d* y agregar lo siguiente:

```
service gsiftftp
{
instances                = 100
socket_type              = stream
wait                     = no
user                     = root
env                      +=
GLOBUS_LOCATION          = /usr/local/globus
env                      +=
LD_LIBRARY_PATH          = /usr/local/globus/lib
server                   = /usr/local/globus/sbin/globusgridftpserver
server_args              = -i
log_on_success           += DURATION
nice                     = 10
disable                  = no
}
```

Crear un archivo llamado *gsigatekeeper* en */etc/xinetd.d* y agregar lo siguiente:

```
service gsigatekeeper
{
instances                = 100
socket_type               = stream
wait                     = no
user                     = root
env                      += GLOBUS_LOCATION=/Users/globus/globus
env                      += LD_LIBRARY_PATH=/Users/globus/globus/lib
server                   = /Users/globus/globus/sbin/globusgatekeeper
server_args               = -conf /Users/globus/globus/etc/globusgatekeeper.conf
log_on_success            += DURATION
nice                     = 10
disable                  = no
}
```

Ahora, para iniciar el servicio, escribir:

```
# sbin/service gsift
# sbin/service gsigatekeeper
```

Actualizando */etc/services*

Este archivo almacena varios puertos reservados y los protocolos que se ejecutan en esos puertos, *gsift* y *gsigatekeeper* son los dos protocolos como TCP. Este archivo es importante porque si el *firewall* está activo, habrá varias restricciones de puertos las cuales podrán ser identificadas en este archivo.

Las entradas dentro de */etc/services*,

```
gsift                2811/tcp
gsigatekeeper        2119/tcp
```

En algunos sistemas muchos de los puertos están reservados para varios servicios. Por eso es mejor ejecutar GridFTP en un puerto más alto, por ejemplo 5000:

Configurando GridFTP

Debe crearse un archivo llamado *gridftp.conf* dentro de *\$GLOBUS_LOCATION/etc/* con un contenido básico como:

```
banner "Welcome"
inetd 1
daemon 1
detach 0
```

Existen muchas más opciones, pero éstas son las básicas.

Probando GridFTP:

1. Iniciar el servidor GridFTP:
globus-gridftp-server
2. Iniciar sesión como usuario normal.
3. También se puede usar el mismo comando para transfererir algunos archivos.

Probando Gatekeeper:

Lo siguiente es la base para probar *gatekeeper*. Se pueden realizar más pruebas si se tiene un *shell* ejecutable.

```
$ globus-job-run localhost /bin/date
```

Esta instrucción proporcionará la fecha actual del sistema.

De manera similar, se podrá probar cualquier otro comando del *shell* utilizando el comando *globus-job-run*.

Una vez que se han instalado, configurado y probado los Pre-WS, se puede decir que se ha concluido con la instalación de una GRID con los servicios básicos.

4.2.5 Enviando trabajos a la GRID

El siguiente paso es enviar trabajos a la GRID. Para enviar los trabajos se pueden utilizar los siguientes comandos

Acerca del *Resource Specification Language (RSL)*

RSL es el lenguaje de especificación de recursos usado para representar la descripción del envío trabajo en Globus. Los diferentes atributos en RSL son:

arguments – usado para especificar argumentos al ejecutable.

count – número de procesos.

directory – directorio inicial.

executable – nombre del ejecutable con la ruta completa.

environment – variables de ambiente necesarias para el ejecutable.

jobType – tipo de trabajo (single, multiple, mpi, condor).

maxTime – tiempo máximo en que puede ser ejecutado el trabajo.

maxWallTime – tiempo máximo suspendido.

maxCpuTime – tiempo máximo en CPU.

gramMyjob – para interactuar con la interfaz *gramMyJob*.

stdin – entrada estándar.

stdout – salida estándar.

stderr – error de flujo.

queue – apunta el trabajo a una cola.

project – apunta el trabajo a una cuenta de proyecto como definido por el planificador.

dryRun – no enviará el trabajo, sólo verificará si se satisfacen las condiciones.

maxMemory – máxima memoria permitida.

minMemory – mínima memoria permitida.

hostCount – sólo si se ejecuta en máquinas SMP (*Symmetric Multiprocessing*), usado para especificar el número de host a usar.

Ejemplos de RSL:

Relación	Ejemplo	Significado
(executable=string)	(executable="/bin/ls")	Un archivo de programa ejecutable.
(arguments=list)	(arguments= -l -d .)	Argumentos al programa.
(directory=string)	(directory="/tmp")	El directorio activo de trabajo.
(environment=list)	(environment= (LD_FLAGS "-o") (COPTS "-g -c"))	Variables de ambiente para ser asignadas al trabajo.
(stdin=string) (stdout=string) (stderr=string)	(stdin=/tmp/cmds)	Donde stdin, stdout, o stderr debería venir de/ir a (pueden ser archivos o URLs).
(count=integer)	(count=5)	Número de procesos a ejecutarse (por default es 1).
(hostCount=integer)	(hostCount=5)	Número de CPUs en el cual se ejecutaran los procesos.
(maxCpuTime=integer)	(maxCpuTime=10)	Tiempo máximo de ejecución de CPU en minutos
(maxMemory=integer)	(maxMemory=100)	Máxima memoria por proceso en Megabytes
(minMemory=integer)	(minMemory=5)	Mínima memoria por proceso en Megabytes
(jobType=string)	(jobType=condor)	Especifica cómo comenzar los procesos dentro del trabajo

Tabla 4.2. Especificación RLS

Usando globus-job-run (Pre-WS-GRAM)

La sintaxis del comando se da a continuación:

```
% globus-job-run 'contact string' command
```

Ejemplos:

```
% globus-job-run localhost /bin/date
```

```
% globus-job-run localhost/jobmanager-condor /bin/date
```

```
% globus-job-run surya -stdin /tmp/myin.txt -stdout /tmp/myout.txt -s myprog
```

Usando globusrun

La sintaxis del comando se da a continuación:

```
% globusrun [-help] [-f RSL file] [-s][-b][-d][...] [-r RM] [RSL]
```

Ejemplo:

```
% globusrun -f test.rsl
```

Contenido de test.rsl:

```
&(executable=/bin/date)  
(stdout=/tmp/output.date)  
(count=2)
```

En este momento, se cuenta con un sistema habilitado para la GRID con las configuraciones básicas.

Conclusión

Durante el proceso de investigación de esta tesis tuve una visión general acerca de la computación GRID, y de todos los proyectos que se están llevando a cabo utilizando GRID computacionales. Pero sobre todo las grandes ventajas que ofrece la GRID al permitir compartir tanto recursos como poder y velocidad de cómputo desde diversos lugares geográficos.

Si embargo durante el proceso de construcción de la GRID me di cuenta de la complejidad que puede presentarse al no haber suficiente información al respecto. Y eso no es sólo un problema para poder construir la GRID, también existe el problema de programar en paralelo para poder utilizar ya sea un cluster o una GRID.

El objetivo principal de este trabajo de tesis fue la construcción de una GRID de clusters experimental. Aunque también se esperaba obtener resultados concretos al finalizar esta tesis, como:

- a) Mostrar los diferentes proyectos que se están llevando a cabo con la computación GRID en todo el mundo.
- b) Crear una infraestructura de hardware, consistente de al menos dos clusters de alto rendimiento, conectado entre sí mediante una red.
- c) Instalar una infraestructura de software, que gestionara el acceso y uso compartido de los recursos de hardware.
- d) Contar con un esquema de autenticación y seguridad para el uso de la infraestructura de GRID (hardware y software).

La infraestructura de hardware de esta GRID experimental se construyó utilizando dos clusters, cada uno formado por un servidor y un nodo. La ventaja de construir una GRID con clusters, es que pueden utilizarse las máquinas existentes en cualquier centro de cómputo.

Para la infraestructura de software se instaló Globus Toolkit 4.0.1, se eligió este software porque es el *middleware* más utilizado en la construcción de las GRIDS, además de ser código abierto a todo el público. Para poder instalar este software, los clusters debían contar con un sistema operativo compatible con esta versión de Globus. En este caso se instaló el sistema operativo RedHat 9.0.

Finalmente la infraestructura de comunicación se llevó a cabo con una red local.

Después de la construcción y configuración de los clusters, se prosiguió con la instalación y configuración de los servicios proporcionados por el Globus Toolkit 4, que es el software básico para la construcción de una GRID. Este software proporciona: los servicios de autenticación (de usuarios, equipos de cómputo y otros servicios) necesarios para la interacción segura y confiable entre elementos participantes en la infraestructura GRID; los servicios de manejo de recursos para ejecutar y controlar procesos en máquinas remotas; los servicios de manejo de información de los recursos pertenecientes a la infraestructura de la GRID; los servicios de manejo de archivos desde y hacia los recursos pertenecientes a la infraestructura de la GRID.

Uno de los problemas que se presentaron durante la construcción de la GRID, es que no se puede paralelizar una tarea y repartirla hacia los diferentes nodos de dos o más clusters al mismo tiempo, debido a un problema que tiene MPICH-G2. Esto sólo podría resolverse si cada nodo de los clusters tuviera una IP pública, lo cual no es factible, ya que el principal objetivo de un cluster es que el servidor y el conjunto de sus nodos se vean como un solo equipo. De tal manera que la función de la GRID para trabajar con clusters se reduce a poder utilizar uno u otro cluster, pero no dos o más al mismo tiempo. A pesar de este problema, la GRID de clusters sigue teniendo la ventaja de que si un cluster esta muy ocupado y otro no lo está se puede elegir el que está desocupado, o también podría elegirse el cluster que tuviera más recursos o mayor poder de cómputo.

En esta tesis se muestran las ventajas del cómputo GRID en general y no sólo de las GRID de clusters, es por ello que por el momento es preferible tener una GRID donde además intervengan computadoras personales y supercomputadoras, para que puedan aprovecharse al máximo los recursos (al mismo tiempo) de todos los equipos de cómputo que formen parte de la GRID.

Con este trabajo de tesis se logró construir una GRID de pequeña escala, que servirá como plataforma de experimentación de diversos esquemas de operación y uso. La infraestructura actual de la GRID contiene todos los elementos necesarios para continuar con la experimentación de proyectos de computación GRID. Y permitirá poder construir una GRID de mayor escala o poder integrarse a alguna GRID existente.

No está por demás comentar que el Laboratorio de Telemática podría formar parte de la GRID Académica Mexicana, ya sea aportando servicios, poder de cómputo o recursos humanos para el desarrollo de proyectos.

Espero que el presente trabajo de tesis represente una buena fuente que sirva de apoyo a todo aquel interesado en la programación en paralelo, y en la computación GRID. Pero sobre todo espero que en un futuro existan proyectos de diversas instituciones que puedan aprovechar los recursos y poder de cómputo que ofrecen las GRIDS.

Bibliografía:

Juárez Sosa Julio, Sáenz García Karén y Valdez Casillas Oscar. “*Análisis del desempeño de un cluster Beowulf en diversos algoritmos de tipo concurrente*”. Directora de Tesis: Ing. Laura Sandoval Montaña. Facultad de ingeniería. UNAM, 2001.

Balderas Jiménez Roy, Moreno Sánchez Jeanette y Plácido Mojica Martín. “*Administración del cluster tipo Beowulf de la Facultad de Ingeniería*”. Directora de Tesis: Ing. Laura Sandoval Montaña. Facultad de ingeniería. UNAM, 2003.

González Flores Christian, González Flores Lissette y Ramírez Ponce Eryk. “*Elaboración de prácticas de programación distribuida en el cluster tipo Beowulf de la Facultad de Ingeniería*”. Directora de Tesis: Ing. Laura Sandoval Montaña. Facultad de ingeniería. UNAM, 2004.

Mesografía:

Sistemas Distribuidos

<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

Introducción a los sistemas distribuidos

<http://members.fortunecity.es/lrmdl/SO7.htm#Intro>

Sistemas Distribuidos, Conceptos y Características

<http://mx.geocities.com/alfonsoaraujocardenas/sistemasdistribuidos.html>

Sistemas Distribuidos

http://gsync.escet.urjc.es/simple_com/phd-thesis-es/node11.html

Arquitecturas Paralelas

<http://www.microsoft.com/spanish/MSDN/estudiantes/computadores/paralelas/default.asp>

Clusters de Linux

http://clusters.fisica.uson.mx/clusters_de_Linux.htm

Clusters de Linux

<http://clusters.fisica.uson.mx/>

Clusters

http://www.consol.org.mx/2002/ponencias/conferencias/Edgar_Govea_-_Clusters.html

Programación Concurrente

<http://www.willydev.net/descargas/codigofuente/concurre.pdf>

Arquitecturas GRID orientadas a la gestión de recursos

<http://ifc.uv.es/grid/computacion-grid-ifc/doc/ArquitecturasGrid-AlvaroFernandez.pdf>

GRID Computing, Un nuevo paradigma de computación distribuida

http://www.eside.deusto.es/eventos/semana/eventos/pdf/grid_computing.pdf

GRAMA, GRID Académica Mexicana

<http://www.grama.org.mx>

GRIDs y e-CIENCIA

<http://grid.ifca.unican.es/English/Presentations/rediris02/GRID-Final-Boletin-RedIRIS.doc>

Grid Computing, un Salto Cuántico

http://www.gerente.com/revistas/manager/0205/venezuela/informe1_0205.html

La Gran Matriz

http://www.gerente.com/revistas/manager/0205/colombia/informe1c_0205.html

European GRID

<http://www.hispacluster.org/modules.php?op=modload&name=News&file=article&sid=519>

Global Grid Forum

<http://www.globalgridforum.org>

The Data GRID Project

<http://eu-datagrid.web.cern.ch/eu-datagrid/>

EUROGRID, Application Testbed for European GRID computing

<http://www.eurogrid.org/>

EGEE, Enabling GRIDs for E-science

<http://www.eu-egee.org>

CrossGrid

<http://www.eu-crossgrid.org>

Condor, High Throughput Computing

<http://www.cs.wisc.edu/condor>

Entropía, PC GRID Computing

<http://www.entropia.com>

SETI, Search for Extra-Terrestrial Intelligence

<http://setiathome.ssl.berkeley.edu>

Geant

<http://www.dante.net/geant>

DAMIEN, Distributed Applications and Middleware for Industrial Use of European Networks

<http://www.hlrs.de/organization/pds/projects/damien>

Construction of IISc-Grid using Globus Toolkit 4.0.0

<http://garl.serc.iisc.ernet.in/docs/2005/interns/pushparajan-report.pdf>

The Globus Toolkit

<http://www.globus.org/toolkit/docs/4.0/admin/docbook/>