

42



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGÓN"**

**DESEMPEÑO Y SINTONIZACIÓN DE UN
SISTEMA UNIX PARA SERVICIOS DE RED**

T E S I S

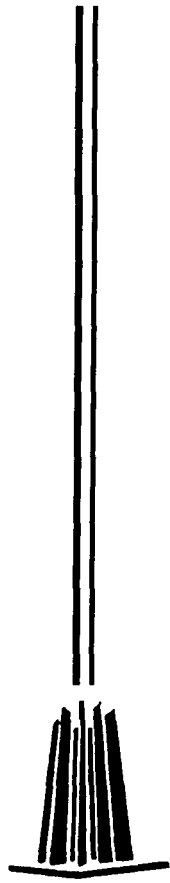
QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
CRISTINA NICOLÁS NICOLÁS

**ASESOR:
ING. DONACIANO JIMÉNEZ VÁZQUEZ**

MÉXICO

2002

**TESIS CON
FALLA DE ORIGEN**





Universidad Nacional
Autónoma de México



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.

**Desempeño y Sintonización de un Sistema UNIX
para Servicios de Red**

Cristina Nicolás Nicolás

Copyright © 2002 kpuemy Inc.
México.

TESIS CON
FALLA DE ORIGEN

A MI MADRE

*Y estás aquí,
me envuelve tu perfume
sutil y dulce
como de gardenias.*

*Veo tu faz amada,
iluminada
por la sonrisa franca y plena
que sólo tienen las personas buenas.*

*Casi no lo creo y sin embargo, siento el calor
de tu amoroso abrazo
y la ternura inconfundible de tus manos.*

*Y me invade otra vez el imperioso deseo
de cobijarme en tu regazo
y olvidarme de todo y disfrutar tus caricias
sabiéndome protegida por tus años.*

*Qué delicia es tenerte tan juntito a mi pecho,
impregnándome toda de valor y de fuerza,
que a decir la verdad, me sorprende que venga
de tu frágil y casi etérea presencia.*

*Y quisiera decirte cuánto te he extrañado
y pedirte mil veces que no te vayas más,
pero no digo nada, pues mi boca enmudece
ante la desbordante emoción de mi corazón.*

*Y en ese preciso instante... te desvaneces
en la penumbra de mi habitación
dejándome confusa y alterada.*

*Y al insistir en buscarte,
inconscientemente miro hacia el firmamento,
donde una estrella pequeña y resplandeciente
pareciera vibrar con mis latidos.*

*Y mientras ella filtra su diáfana luz
a través de mi ventana,
yo me quedo triste y anhelante,
en el vacío de ti...*

TESIS CON
FALLA DE ORIGEN

Mi primer y mayor deuda de gratitud es para mis padres porque gracias a su apoyo y consejos, he llegado a realizar una de mis metas más grandes, la cual constituye la herencia más valiosa que pudiera recibir.

A quien la ilusión de su existencia ha sido verme convertida en persona de provecho.

Doy gracias a ti Señor por haberme dado a estas personas tan queridas y por haberme dado la oportunidad de entrar en el bello y maravilloso mundo del saber.

Porque ahí donde me has puesto debo florecer y luchar por mi realización.

A los seres universalmente más queridos.

Gracias... mamá.

Gracias... papá.

TESIS CON
FALLA DE ORIGEN

AGRADECIMIENTOS

Desde este espacio deseo expresar mi más profundo agradecimiento y reconocimiento a las personas que han formado parte de este proyecto y que con su granito de amor hacen posible mejorar la vida de quién lo necesita.

A mi familia...

Por haber depositado su confianza en mí, por brindarme su apoyo y por estar siempre a mi lado. A Roger por ese legado de fortaleza que depositó en mí. A Mary, Iran y Fer que sufren mis constantes ausencias pero que saben que peleamos juntos para seguir hacia adelante.

A Eduardo...

Por compartir conmigo parte de tu existencia, por tu extraordinaria calidad humana, por poner a mi disposición tu confianza, experiencia y apoyo incondicional; pero sobre todo por ese amor y cariño que siempre me has brindado y porque sin ti este proyecto no existiría.

A todos mis amigos y colegas...

Por contribuir desinteresadamente, por su disposición, entusiasmo y consejo; especialmente al profesor Donaciano por ocuparse del contenido técnico y por su labor de coordinación.

A mi Alma Mater...

La UNAM, a quién le debo mucho más que solo mi desarrollo profesional.

El haber llegado hasta aquí no habría sido posible sin cada uno de ustedes. ¡GRACIAS!

— Cristy

TESIS CON
FALLA DE ORIGEN

Los grandes sucesos dependen de incidentes pequeños.

- Demóstenes

**TESIS CON
FALLA DE ORIGEN**

ÍNDICE

Introducción	1
01 Antecedentes	4
1.1 Sistema operativo UNIX	4
1.1.1 Breve historia del sistema operativo UNIX	5
1.1.2 Características y estructura del sistema UNIX	9
1.1.3 Administración del sistema operativo UNIX	14
1.1.3.1 Tareas del administrador del sistema	15
1.1.3.2 Herramientas básicas de administración	15
1.2 Servicios de red	20
1.2.1 Telnet	23
1.2.2 Correo electrónico	24
1.2.3 FTP	25
1.2.3.1 FTP anónimo	28
1.2.4 World Wide Web	29

**TESIS CON
FALLA DE ORIGEN**

02 Desempeño del sistema	31
2.1 Administración de procesos	32
2.1.1 Creación y ejecución de un proceso	33
2.1.2 Eliminación de un proceso	34
2.1.3 Monitoreo de procesos	36
2.1.4 Niveles de prioridad	38
2.1.5 Control de prioridades	41
2.2 Administración del desempeño	44
2.2.1 Fundamentos del desempeño de un sistema	44
2.2.2 Herramientas de monitoreo	46
2.2.3 Recursos del sistema	48
2.2.3.1 Monitoreo del desempeño del CPU	50
2.2.3.2 Monitoreo del desempeño de la memoria	53
2.2.3.3 Monitoreo del desempeño del subsistema de E/S	57
2.3 Desempeño de la red	62
2.3.1 Monitoreo y desempeño de la red	63
2.3.2 Etiqueta de red básica	68
03 Sintonización del Sistema	73
3.1 Sistemas de alto rendimiento	74
3.1.1 Terminología y conceptos	74
3.1.2 Configuración del sistema	76
3.1.3 Entendiendo el alto desempeño	77
3.1.4 Configuración y sintonización de un sistema	78
3.2 Diagnosticando problemas de desempeño	81
3.2.1 Revisando el rendimiento del sistema	81
3.2.2 Obteniendo información de desempeño	82
3.2.3 Usando herramientas adicionales de monitoreo	83
3.3 Modificación del kernel	87
3.4 Sintonización del CPU	87
3.4.1 Identificando cuellos de botella de CPU	88
3.4.2 Optimizando los recursos del CPU	89
3.4.3 Mejorando el desempeño de una aplicación	90
3.5 Sintonización de la memoria	91
3.5.1 Memoria virtual	92
3.5.2 Buffer cache	93
3.5.3 Configuración de la memoria y espacio swap	93
3.5.4 Optimizando la memoria virtual	94

TESIS CON
FALLA DE ORIGEN

3.6 Sintonización del subsistema de almacenamiento	95
3.6.1 Entendiendo el subsistema de almacenamiento	95
3.6.2 Administración de discos y archivos	97
3.7 Sintonización del subsistema de red	98
04 Monitoreo y sintonización de un sistema real	100
4.1 Estado actual	100
4.1.1 Infraestructura actual	100
4.1.1 Criterio de evaluación del software para el monitoreo del desempeño.	104
4.2 Estrategias para el mejoramiento del desempeño	106
4.3 Fase de sintonización	107
4.3.1 Carga del sistema	108
4.3.2 Estadísticas del CPU	117
4.3.3 Estadísticas de la memoria	119
4.3.4 Estadísticas de E/S	125
4.3.5 Estadísticas de la red	129
4.4 Revisión del sistema	133
4.5 Recomendaciones	134
4.6 Implementación de herramientas adicionales	136
4.7 Optimización del sistema de archivos	143
4.7.1 Activación de cuotas	144
4.7.2 Monitoreo de cuotas	146
4.8 Pruebas de simulación	146
4.9 Resultados	149
4.10 Otras consideraciones	152
4.11 Perspectivas a futuro	153
Conclusiones	155
Glosario	159
Bibliografía	163

TESIS CON
FALLA DE ORIGEN

INTRODUCCIÓN

En las grandes redes dispersas de hoy día, los administradores suelen experimentar serias dificultades al tratar de identificar problemas en sus sistemas así como en la red. Este método resulta complicado y cada vez más difícil a medida que los sistemas crecen. Hasta ahora, cuando se reciben llamadas debido a problemas de rendimiento, los administradores se sienten perdidos intentando detectar de manera rápida la causa del problema. Generalmente, los administradores dedican poco tiempo para monitorear el uso de los recursos del sistema así como el tráfico de la red a no ser que se den problemas y los usuarios comiencen a quejarse.

Desafortunadamente, muchos usuarios y administradores UNIX tienen la impresión de que los mecanismos para optimizar un sistema UNIX son tan inescrutables que prefieren mejor no hacerlo, sin embargo, es posible hacer cambios para mejorar el desempeño de un sistema aunque esta sea una tarea compleja. Los cambios más importantes involucran construir un kernel UNIX personalizado, lo cual es una tarea desalentadora y peligrosa.

TESIS CON
FALLA DE ORIGEN

Pero gran parte del proceso de sintonización de un sistema consiste en implementar buenas políticas de administración para un sistema: mantenerse alerta de cómo se utiliza el sistema, correr los procesos grandes por la noche si es posible, etc.; sin necesidad de alterar el kernel.

En este trabajo se pretende proporcionar una introducción básica a la sintonización del desempeño de un sistema: optimizar el rendimiento de un sistema UNIX por completo. Se estructurará el sistema para maximizar su desempeño total, de tal forma que sus recursos sean usados de manera eficiente y distribuidos principalmente entre los usuarios. Se intentará entender que parte del sistema esta sobrecargado, como disminuir dicha carga y saber que parte del sistema se tiene que actualizar si es necesario. Se buscarán herramientas que funcionen con las características del sistema operativo UNIX para trabajos en red que cuenten con la capacidad de medición del rendimiento de cada uno de los recursos del sistema.

Con este proyecto se espera que los administradores de sistemas cuenten con las herramientas necesarias para aislar problemas de rendimiento, detectar cuellos de botella y realizar análisis para gestionar y diagnosticar los tiempos de respuesta para que puedan ofrecer a los usuarios un nivel aceptable de desempeño.

El punto en que el estado de un sistema llega a ser crítico depende de varios factores: que tan rápidos son los discos, que tipos de aplicaciones están corriendo, cuanta paginación hay, etc. Si el tiempo ocioso es muy cercano a cero, significa que hay muchos procesos en ejecución en el sistema, pero también puede ser el primer signo de carga en el CPU.

Este proyecto esta basado en la implementación del sistema operativo Solaris, sin embargo, se puede extender a otros sistemas considerando las características propias de cada implementación debido a que difieren radicalmente unas de otras.

Este proyecto esta organizado de la siguiente manera:

Capítulo 1, *Antecedentes.*

Contiene una breve reseña histórica del sistema operativo UNIX, su estructura y sus principales características; a sí mismo se explican las tareas básicas de administración de un sistema.

Capítulo 2, *Desempeño del Sistema.*

Describe como llevar a cabo la administración de los procesos, los factores que contribuyen a mantener y administrar el desempeño de un sistema y se listan algunas herramientas útiles para el monitoreo del desempeño del sistema y la red.

Capítulo 3, *Sintonización del Sistema.*

Contiene información sobre la configuración de un sistema UNIX para obtener un alto rendimiento, además cómo monitorear, identificar y optimizar los recursos de un sistema para mejorar su desempeño.

Capítulo 4, *Monitoreo y Sintonización de un Sistema Real.*

Proporciona una introducción básica para optimizar un sistema UNIX, basado en el sistema operativo Solaris, con el fin de obtener un mejor rendimiento en sistemas que ofrecen servicios de red.

Finalmente el glosario define varios de los términos técnicos empleados a través de los capitulos aquí descritos.

1. ANTECEDENTES

TESIS CON
FALLA DE ORIGEN



Antes de entrar en detalle en los aspectos que intervienen en el análisis del desempeño de un sistema UNIX, es importante revisar algunos conceptos básicos relacionados con la operación de este sistema operativo y que son necesarios para el desarrollo del presente trabajo de tesis.

1.1 SISTEMA OPERATIVO UNIX

Un sistema operativo es un conjunto de programas que sirven para controlar y organizar los recursos de un sistema de cómputo. Dichos recursos consisten en componentes de hardware, como son las terminales e impresoras; y los programas de software, los cuales le indican a la computadora que realice determinadas tareas.

En general, los sistemas grandes tienen varios usuarios y realizan múltiples tareas; puede haber varias terminales que están conectadas a una sola computadora y el sistema operativo controla el acceso que puede tener a la computadora toda una comunidad de usuarios.

UNIX, es un sistema multiusuario y multitarea producido originalmente en los laboratorios AT&T. Cuenta con programas para la edición de textos, envío de correo electrónico, conexión de redes y muchas otras funciones especializadas.

A pesar de que la mayor parte de los sistemas operativos más importantes fueron diseñados y controlados por una sola compañía, UNIX pudo desarrollarse como consecuencia de un esfuerzo colectivo. La mayoría de las versiones funcionan también en sistemas de ventanas, lo que permite a los usuarios tener más de una "terminal" en una sola pantalla.

1.1.1 Breve historia del sistema operativo UNIX

Durante un breve período en 1969, el Departamento de Investigación Científica de Computadoras de los Laboratorios Bell utilizaron una computadora 645 de General Electric con un sistema operativo llamado Multics¹, el cual fue uno de los primeros sistemas operativos interactivos multiusuario, sin embargo, carecía de características esenciales para los programadores.

Un proyecto desarrollado por Ken Thompson, uno de los investigadores de los Laboratorios Bell, fue un programa llamado "Viaje Espacial", simulaba el movimiento de los principales cuerpos celestes del sistema solar. Como el costo era muy alto, Thompson rediseñó este proyecto para ejecutarse a un costo más bajo, sobre un PDP-7 de Digital Equipment Corporation (DEC); sin embargo, el software disponible para este equipo no tenía capacidad de memoria para almacenar los continuos desarrollos del programa fuente, por esta razón Thompson escribió un sistema operativo, un ensamblador específico para el PDP-7; el cual soportaba dos usuarios simultáneamente y al que se le llamó humorísticamente UNICS². El primer uso de este nombre se le atribuye a Brian Kernighan, posteriormente cambió ligeramente a lo que se conoce como sistema UNIX en 1970 y ha permanecido así desde entonces.

¹ *Multiplexed Information and Computing System* (Información Multiplexada y Sistema de Computación).

² *Uniplexed Information and Computing System* (Información Uniplexada y Sistema de Computación).

Ken Thompson y Dennis Ritchie gestionaron la obtención de un DEC PDP-11/20 en contrapartida a la promesa de añadir capacidades de procesamiento de texto al sistema UNIX, por lo que el programa de formateado de texto `runoff` fue portado a este nuevo sistema.

En 1972, la segunda edición del Manual del programador UNIX mencionaba que había exactamente 10 computadoras utilizando el sistema operativo UNIX, pero indicaba que se estaba esperando más.

Un sistema operativo basado en lenguaje ensamblador es "dependiente de la máquina". El programa funciona en un tipo concreto de computadora y no puede "transportarse" fácilmente a una computadora distinta, por ello Thompson desarrolló el lenguaje B, este fue modificado por Dennis Ritchie, quien desarrolló el lenguaje C.

En 1973, el sistema UNIX, basado en el Multics y escrito en ensamblador para el DPD-7, fue reescrito en C lo que hacía más fácil su mantenimiento y portabilidad a otras máquinas, esto hizo que la popularidad de UNIX creciera debido a que estaba escrito compactamente en un lenguaje de alto nivel con código que podía modificarse de acuerdo a las preferencias individuales.

Hacia 1974 comenzó a utilizarse ampliamente en los Laboratorios Bell la cuarta edición del sistema UNIX. Hacia 1977 salió la quinta y la sexta ediciones; éstas contenían bastantes más herramientas y utilidades, por lo que el número de máquinas que ejecutaban el sistema UNIX, fundamentalmente en los Laboratorios Bell y en las universidades se incrementó en más de 600 en 1978.

En 1979 los Laboratorios Bell introdujeron la séptima edición del sistema UNIX, el ascendente directo del sistema operativo UNIX disponible hoy día, esto permitió a los fabricantes de computadoras desarrollar software económico para UNIX, con el fin de utilizarlo con sistemas UNIX con licencia, y a los pequeños comerciantes utilizar la licencia del sistema UNIX para sus computadoras personales.

En noviembre de 1981, Bell lanzó el Sistema III basado en la séptima edición, como el primer sistema UNIX "comercial" de AT&T.

Para eliminar la confusión sobre las variedades del sistema UNIX, AT&T introdujo el UNIX Sistema V Versión 1 en 1983. El sistema IV de UNIX sólo existió como versión interna AT&T. Con el UNIX Sistema V Versión 1, AT&T prometió por primera vez mantener *compatibilidad ascendente* en sus futuras versiones del sistema UNIX, esto significaba que los programas construidos sobre la Versión 1 podrían continuar operando con futuras versiones del Sistema V. Esta versión incorporó algunas características de la versión del sistema UNIX desarrollado en la Universidad de California Berkeley, incluyendo el editor de pantalla vi y la biblioteca de manejo de pantalla curses.

AT&T ofreció el UNIX Sistema V Versión 2 en 1985, que incluía protección de archivos durante cortes de energía, bloqueo de archivos y registros para uso exclusivo por un programa, característica de control de trabajos y administración ampliada del sistema. La Versión 2.1 introdujo dos características adicionales de interés para los programadores: la paginación por demanda, que permite ejecutar procesos que requieren más memoria de la que se dispone físicamente y bloqueo de archivos y registros.

En 1987, AT&T introdujo el UNIX Sistema V Versión 3.0; que incluía un enfoque de redes simple y consistente, capacidades que incluyen STREAMS³ para construir software de redes; un sistema de archivos remoto, utilizado para compartir archivos a través de redes; y la *Interfaz a Nivel de Transporte* (TLI) para construir aplicaciones que utilizan redes.

La Versión 3.1 hizo al Sistema V de UNIX adaptable internacionalmente, soportando conjuntos de caracteres y formatos de hora y fecha más amplios. También incluía los diferentes aumentos de rendimiento en el uso de la memoria y en las copias de seguridad y la recuperación de archivos.

³ Flujo o secuencia de datos de entrada o salida para la comunicación de los procesos en UNIX.

La Versión 3.2 proporcionó seguridad ampliada del sistema, incluyendo la visualización de la hora de la última sesión del usuario, el registro de los intentos de conexión sin éxito y un archivo oculto de contraseñas que impide a los usuarios la lectura de contraseñas cifradas. Esta versión también introdujo el *Framed Acces Command Enviroment* (FACE) que proporciona una interfaz de usuario orientada a menús.

UNIX Sistema V Versión 4 fue desarrollada en los Laboratorios Bell en la década de los 80. Se combinaron las mejores variantes de UNIX de aquella época, UNIX Sistema V Versión 3 de AT&T, los sistemas XENIX de Microsoft, BSD y SunOS de Sun Microsystems en un solo sistema operativo potente y flexible. Esta versión se ajusta a los estándares importantes definidos para el sistema UNIX por diferentes organizaciones industriales y gubernamentales.

En 1988 un grupo de vendedores de computadoras (IBM, DEC y Hewlett-Packard), formó un consorcio denominado *Open Software Foundation* (OSF) para desarrollar una versión del sistema UNIX que compitiera con UNIX Sistema V Versión 4. Su versión del Sistema UNIX, llamada OSF/1, nunca jugó realmente un papel importante en el mercado UNIX. De todos los vendedores de este consorcio, sólo DEC basó su eje estratégico en la versión OSF de UNIX. OSF también apoyó su propia interfaz gráfica de usuario, llamada MOTIF, creada como composición de varias interfaces gráficas de algunos vendedores de OSF. Al contrario de OSF/1, MOTIF ha tenido gran aceptación en el mercado.

En 1993, algunos de los vendedores principales de sistemas UNIX crearon el consorcio *Common Open Software Enviroment* (COSE). Entre aquellos vendedores se encontraban Hewlett-Packard, IBM, SunSoft, Santa Cruz Operation, Novell y los laboratorios UNIX System. El objetivo de este consorcio fue definir estándares en la industria para los sistemas UNIX relativos a seis áreas: interfaz gráfica de usuario, multimedia, redes, tecnología de objetos, gráficos y gestión del sistema.

En octubre de 1993 Novell transfirió sus derechos de patente del sistema operativo UNIX a X/Open. Bajo este acuerdo, cualquier empresa puede usar la denominación UNIX para un sistema operativo siempre que cumpla con las especificaciones X/Open, mediante una cuota por los derechos de patente destinados a X/Open.

En septiembre de 1995, UnixWare vendió su título de propiedad sobre UNIX Sistema V Versión 4 y su versión de UNIX a SCO, quien continuó el desarrollo de la Versión 4, por su parte HP estaba desarrollando una implementación de UNIX Sistema V Versión 4 a 64 bits para entregar a SCO.

1.1.2 Características y estructura del sistema UNIX

Durante los últimos veinte años el sistema operativo UNIX se ha convertido en un sistema operativo potente, flexible y versátil. Sirve como sistema operativo para todo tipo de computadoras, incluyendo las computadoras personales monousuario, las estaciones de trabajo de ingeniería, microcomputadoras multiusuario, minicomputadoras, mainframes y supercomputadoras. El número de equipos que funcionan con el sistema operativo UNIX ha crecido de forma exponencial. El éxito del sistema UNIX se debe a diversos factores, entre los que se incluyen su portabilidad a un gran abanico de máquinas, su adaptabilidad y simplicidad, el amplio rango de tareas que puede ejecutar, su naturaleza multiusuario y multitarea y su adecuación a las redes; que han ido creciendo en importancia a medida que ha ido prosperando Internet por todo esto se espera que continúe este rápido crecimiento.

Las características que han hecho tan popular al sistema operativo UNIX incluyen:

- **Código fuente abierto**

El código fuente del sistema UNIX, ha estado disponible a usuarios y programadores. A causa de esto, mucha gente ha sido capaz de adaptar el sistema UNIX de formas diferentes, a quienes les ha resultado fácil porque el código correspondiente es sencillo, modular y compacto.

Esto ha fomentado la evolución del sistema UNIX, haciendo posible la fusión de las capacidades desarrolladas por diferentes variantes necesarias para soportar los entornos de computación de hoy en un sistema operativo.

- **Herramientas y utilidades**

El sistema UNIX proporciona a los usuarios diferentes *herramientas y utilidades* que se pueden emplear para realizar una gran variedad de trabajos. Algunas de estas herramientas son órdenes simples que se pueden utilizar para llevar a cabo tareas específicas.

Otras herramientas y utilidades son realmente pequeños lenguajes de programación que se pueden usar para construir guiones que resuelven sus propios problemas. Lo más importante es que las herramientas están diseñadas para funcionar juntas, como parte de una máquina o bloques de construcción y muchas otras están disponibles como opcionales, entre las que se incluyen todas aquellas que se encuentran disponibles desde archivos en Internet sin costo alguno.

- **Capacidades multiusuario y multitarea**

El sistema operativo UNIX puede ser utilizado por computadoras con varios usuarios o con uno solo, ya que es un sistema *multiusuario*. También es un sistema operativo *multitarea*, ya que un único usuario puede llevar a cabo más de una tarea al mismo tiempo.

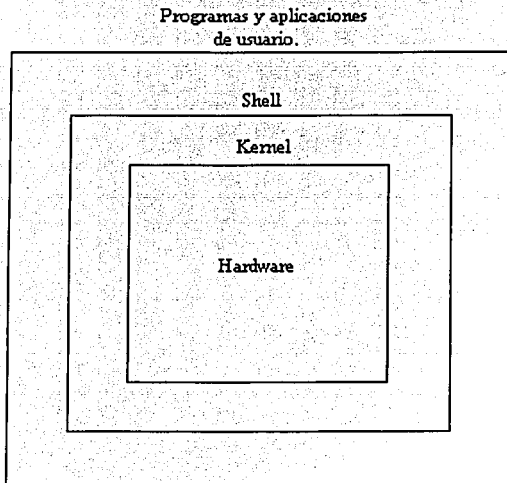
- **Entorno para redes**

El sistema UNIX proporciona un entorno excelente para *redes*. Ofrece programas y facilidades que proporcionan los servicios necesarios para construir aplicaciones basadas en red, base de la computación distribuida, donde la información es compartida por diferentes computadoras de la red en un modelo cliente/servidor. Ha sido también el sistema básico para el desarrollo de los servicios de Internet y su propio crecimiento.

- **Portabilidad**

El sistema UNIX es mucho más fácil de *portar* a nuevas máquinas que otros sistemas operativos, ya que se necesita menos trabajo para adaptarlo y ejecutarlo sobre una máquina nueva. Esta característica es consecuencia directa de estar escrito casi completamente en un lenguaje de alto nivel, el lenguaje C; y hace posible a un amplio rango de computadoras mover las aplicaciones de un sistema a otro.

Para comprender cómo opera el sistema UNIX, es necesario entender su estructura (Fig. 1.1). El sistema operativo UNIX lo forman varios componentes principales. Entre estos componentes están el *kernel*, el *shell*, el *sistema de archivos* y las *órdenes* (o *programas de usuario*).



TESIS CON
FALLA DE ORIGEN

Fig.1.1 Estructura del sistema UNIX.

- **Aplicaciones**

Se pueden utilizar *aplicaciones* construidas utilizando ordenes, herramientas y programas del sistema UNIX. Los programas de aplicación llevan a cabo diferentes tipos de tareas, ejemplos de ellos son programas tales como: procesadores de texto, compiladores, sistemas de bases de datos, hojas de cálculo, programas de análisis estadístico y programas de comunicaciones.

Varias clases de aplicaciones han experimentado un crecimiento en los últimos años; algunas de ellas son las aplicaciones de red, incluyendo aquellas que permiten a la gente hacer uso de un amplio rango de servicios disponibles en Internet. Destacan entre ellas los visualizadores para el World Wide Web tales como Internet Explorer y Netscape. Otras aplicaciones son aquellas que tratan con la multimedia, las cuales permiten a los usuarios crear y visualizar archivos multimedia como: sonido, imágenes y vídeo.

- **Programas**

El sistema UNIX contiene varios programas o utilerías de usuario. Las órdenes también se conocen como herramientas y pueden utilizarse independientemente o en forma conjunta de diversas maneras para llevar a cabo tareas útiles. Estas utilerías se pueden ejecutar invocándolas por su nombre a través del shell, por esta razón se conocen como *órdenes*. Hay programas para: edición y procesamiento de texto, gestión de información, comunicaciones electrónicas y redes, para realizar cálculos, desarrollar programas de computadoras, administración del sistema y para otros propósitos.

- **El sistema de archivos**

El sistema de archivos del sistema UNIX proporciona un método lógico para organizar, almacenar, recuperar, manipular y gestionar la información. La unidad básica utilizada para organizar la información en el sistema UNIX se denomina *archivo*, identificado este como un conjunto de datos con un nombre asociado; estos están organizados en un *sistema de archivos jerárquico* agrupados en *directorios*.

Un directorio no es más que un archivo especial que contiene información que permite encontrar otros archivos. Estos a su vez pueden contener nuevos directorios los cuales se denominan *subdirectorios*. Una característica de simplificación importante del sistema UNIX es la forma general de tratamiento de los archivos. El sistema de archivos es la parte responsable de la administración de los datos en dispositivos de almacenamiento secundarios (normalmente discos o particiones de disco).

- **El shell**

Una gran parte del uso del sistema UNIX consiste en emitir órdenes. Cuando se emite alguna se está relacionándose con el shell, la parte del sistema UNIX a través del cual se controlan los recursos del sistema operativo UNIX. El shell lee las órdenes y las interpreta como peticiones de ejecución de un programa o programas, lo que realiza posteriormente. Debido a esto, el shell se denomina *intérprete de órdenes* o *intérprete de comandos*, además el shell también es un lenguaje de programación que permite controlar cómo y cuándo se llevan a cabo las órdenes. El sistema UNIX proporciona principalmente tres shells diferentes para elegir. Además del shell del sistema conocido como Bourne Shell (**sh**) existe el C shell (**csh**) y el Korn shell (**ksh**).

- **El kernel**

El *kernel* es la parte del sistema operativo que interactúa directamente con el hardware de una computadora, a través de los controladores de dispositivo que están incorporados en el kernel. Las funciones principales del kernel son: la gestión de la memoria, el control de acceso a la computadora, el mantenimiento del sistema de archivos, el manejo de las interrupciones⁴, el manejo de errores, la realización de los servicios de entrada y salida (que permiten a las computadoras interactuar con terminales, dispositivos de almacenamiento e impresoras) y la asignación de recursos de la computadora (tales como CPU, memoria o dispositivos de E/S) entre los usuarios.

⁴ Señales que finalizan una ejecución.

Los programas interactúan con el kernel a través de *llamadas al sistema*, las cuales indican al kernel que lleve a cabo diferentes tareas para el programa, tales como: abrir un archivo, escribir en un archivo, obtener información sobre un archivo, ejecutar un programa, terminar un proceso, cambiar la prioridad de un proceso u obtener la hora y fecha del sistema.

1.1.3 Administración del sistema operativo UNIX

Cada sistema UNIX debe tener su propio administrador o persona encargada de vigilar que todo este a punto en todo momento. Esta labor requiere una serie de conocimientos que los usuarios finales no necesitan dominar, además, es necesario invertir un tiempo considerable para estos menesteres, incluso aunque seamos el único usuario del sistema.

UNIX diferencia entre los distintos usuarios, de tal manera que se controla qué es lo que podemos hacerle a otros usuarios o al propio sistema. Cada uno de ellos tiene su propia cuenta, la cual incluye nombre de usuario, grupo al que pertenece, directorio de trabajo, etc. De todas las cuentas del sistema, la más importante es la denominada cuenta del administrador o superusuario, cuyo nombre de usuario es `root`.

Las cuentas de usuarios tienen asociadas una serie de restricciones de forma que nadie pueda molestar al resto. Nadie puede borrar directorios como `/bin` o `/etc`, ni tampoco puede inhabilitar una impresora, sin embargo, el administrador tiene plenos poderes para borrar, crear o modificar cualquier archivo o directorio del sistema, para ejecutar programas especiales o para dar formato al disco.

1.1.3.1 Tareas del administrador del sistema

El administrador tiene una serie de responsabilidades relacionadas con el software, hardware y con los usuarios del sistema, sus principales actividades son:

- Verificar la correcta instalación del hardware.
- Agregar y eliminar usuarios al sistema.
- Hacer respaldos (copias de seguridad).
- Crear y mantener los sistemas de archivos.
- Restaurar archivos que por alguna razón se han borrado.
- Preocuparse por la seguridad del sistema.
- Instalar y/o actualizar el sistema operativo.
- Configurar y mantener el software de cualquier dispositivo: impresoras, tarjetas de red, etc.
- Administrar el espacio en disco.
- Levantar el sistema después de una caída.
- Instalar nuevos dispositivos de hardware (memoria, discos, terminales, etc.).
- Resolver interferencias de la red.
- Arreglar o tratar de arreglar archivos de datos.
- Escribir programas para automatizar tantas actividades como sea posible.
- *Monitorear la actividad del sistema y optimizar los recursos del sistema.*

1.1.3.2 Herramientas básicas de administración

Existe una serie de comandos más comúnmente usados como herramientas para obtener información relevante de la administración del sistema y que permiten realizar las tareas básicas administrativas. Estos comandos solo pueden ser ejecutados por root, el usuario privilegiado del sistema. Permiten construir, montar y desmontar sistemas de archivos, comprobar su integridad y solucionar algunos errores. El administrador los utiliza para poner en funcionamiento la explotación del sistema, administrar el espacio en disco o sencillamente para actualizar la fecha y hora.

Apagar el sistema

Apagar o dar de baja el sistema puede ser necesario por diversos motivos; por ejemplo, en un momento dado se tendrá que desconectar el equipo totalmente para poder darle mantenimiento y mientras tanto los usuarios no pueden estar conectados.

La orden `shutdown` permite dar de baja el sistema, provocando la muerte de los procesos que se estén ejecutando en el sistema y los programas de los usuarios, desmonta cualquier sistema de archivos que no sea el raíz (/) y vacía los buffers del sistema. Las opciones de este comando pueden variar de un sistema a otro.

Sintaxis: `shutdown [-igy] [Mensaje de aviso]`

Ejemplo:

```
#shutdown -i0 -g60 -y "El sistema se dará de baja"
```

En el ejemplo anterior, el sistema se ejecutará en el nivel de operación⁵ 0 (-i), deteniendo los servicios en 60 segundos (-g) sin pedir confirmación (-y), finalmente se enviara el mensaje de aviso.

Añadir y eliminar usuarios del sistema

La orden `useradd`, permite que el administrador del sistema añada nuevos usuarios de una forma rápida y cómoda.

Sintaxis: `useradd [-c comentario] [-d directorio] [-g grupo] [-m [-k dir_skel]] [-u uid del usuario] [-s shell] login`

Ejemplo:

```
# useradd -c Cristina Nicolás -d /export/home/cristy -g 1010  
-m -k /etc/skel -u 22 -s /bin/csh cristy
```

Aquí se crea una cuenta para el usuario `cristy`, en el directorio de trabajo `/export/home/cristy` y como shell por default el `csh`.

⁵ Estado de un sistema que define que servicios y recursos son disponibles para los usuarios.

La orden `userdel`, nos permite eliminar las cuentas de los usuarios que ya no se conectan al sistema.

Sintaxis: `userdel [-r] login`

Ejemplo:

```
# userdel -r cristy
```

Asignar contraseña a una cuenta

Para evitar huecos de seguridad a través del sistema debe definirse una contraseña para cada una de las cuentas del sistema. Para asignar una contraseña o bien cambiar una ya existente, el administrador puede hacer uso del comando `passwd`.

Sintaxis: `passwd login`

Ejemplo:

```
# passwd cristy
New password:
Re-enter new password:
```

Se solicitará que se introduzca la contraseña dos veces. Por razones de seguridad la contraseña no aparece en la pantalla.

Verificar procesos del sistema

La forma de saber que procesos se están ejecutando en el sistema en un instante determinado consiste en emplear la orden `ps`, la cual sirve para informarnos acerca de dichos procesos. Si no se da ninguna opción, solo muestra un pequeño informe de los procesos asociados a una terminal.

Sintaxis: `ps [-elf]`

Ejemplo:

```
% ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	0	0	0	Sep 20	?	0:31	sched
root	1	0	0	Sep 20	?	0:00	/etc/init -r
root	4068	252	0	15:36:53	?	0:00	in.telnetd
tuning	4079	4070	0	15:37:31	pts/1	0:00	more
tuning	4070	4068	0	15:36:53	pts/1	0:00	-csh
root	4078	4070	0	15:37:31	pts/1	0:00	ps -ef
unam	3487	3485	0	13:14:10	pts/2	0:00	-csh

Montar y desmontar un sistema de archivos

Es muy común conectar a una misma máquina varios discos físicos, cada uno de ellos probablemente con distintas particiones. En cada una de estas particiones se puede tener un sistema de archivos diferente y surge la necesidad de añadir este sistema de archivos al único disco lógico existente. Aunque se tengan distintos discos físicos, en UNIX todos forman parte de un único disco lógico.

Los comandos `mount` y `umount` sirven para montar y desmontar respectivamente un determinado sistema de archivos a un disco lógico del sistema.

Sintaxis: `mount [-F tipo_FS] [dispositivo] [punto_montaje]`

Ejemplo:

```
# mount /dev/dsk1 /mnt
```

En el ejemplo anterior, se realiza el montaje del sistema de archivos `/dev/dsk1` sobre el directorio `/mnt`.

Sintaxis: `umount [dispositivo] [punto_montaje]`

Ejemplo:

```
# umount /dev/dsk1 /mnt
```

En este ejemplo se desmonta el sistema de archivos `/dev/dsk1` montado anteriormente.

TESIS CON
FALLA DE ORIGEN

Obtener información de espacio en disco

Para ver la cantidad de espacio libre en disco podemos utilizar la orden **df** (disk free). Este comando despliega la cantidad de espacio en disco ocupado por los sistemas de archivos montados o directorios, así como la cantidad de espacio disponible y la capacidad total que el sistema de archivos ha estado usando.

Sintaxis: **df -k**

Ejemplo:

```
% df -k
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/root	4610655	1429182	3181473	31%	/
/dev/stand	48195	5737	42457	12%	/stand
/proc	0	0	0	0%	/proc
/dev/fd	0	0	0	0%	/dev/fd
/tmp	81920	24	81896	1%	/tmp
/dev/dsk/c0b0t1d0s1	4433936	1503432	2930504	34%	/u1
/dev/dsk/c0b0t1d0s2	4441968	112280	4329688	3%	/u2
/dev/dsk/c0b0t2d0s1	4442112	945728	3496384	22%	/u3
/dev/dsk/c0b0t2d0s2	4443136	2110304	2332832	48%	/u4
/dev/dsk/c0b0t3d0s1	4442112	157992	4284120	4%	/u5
/dev/dsk/c0b0t3d0s2	4443136	1172336	3270800	27%	/u6
/dev/dsk/c0b0t4d0s1	4442112	176680	4265432	4%	/u7
/dev/dsk/c0b0t4d0s2	4443136	2115224	2327912	48%	/u8

Respaldar información del sistema

Una de las principales responsabilidades del administrador del sistema es preservar los datos almacenados en el sistema, planificando y realizando copias de seguridad de forma regular. La cantidad de información que se debe copiar y cada cuánto tiempo se debe hacer, dependerá de la utilización que se haga del sistema. Sin una buena política de copias de seguridad se pueden llegar a perder datos valiosos. Un sistema es más seguro cuanto más frecuentemente se hacen copias de seguridad, lo cual permite recuperar una versión de un archivo en caso de que éste sea borrado o modificado. También se puede restaurar los archivos de configuración del sistema si resultan dañados o destruidos.

El comando **tar** se utiliza para guardar o recuperar los archivos de una cinta magnética, un disco flexible o un archivo normal.

Sintaxis: `tar [-cvf] dispositivo archivo`

Ejemplo:

```
# tar -cvf /dev/ctape1 /home/mcn
```

En este ejemplo se leen los archivos del directorio `/home/mcn` (`-f`) y todos sus subdirectorios, imprime un informe extenso (`-v`) de los procedimientos que se están realizando, empaqueta en un archivo (`-c`) y lo envía al dispositivo `/dev/ctape1`.

Especificar fecha/hora

Para especificar la fecha y hora actual en el sistema el administrador del sistema puede hacer uso del comando `date`.

Sintaxis: `date [mmddHHMMyy]`

Ejemplo:

```
# date 0425113602  
Thu Apr 25 14:21:05 CDT 2002
```

Que se descompone como Abril (04) 25 (25), 11:36 AM (1136), 2002 (02). Con esto siempre que se desee puede verse la fecha y hora tecleando `date` sin opciones.

1.2 SERVICIOS DE RED

Ahora más que nunca, una de nuestras necesidades más urgentes es desplazar con rapidez gran cantidad de información a través de enormes distancias. Cada vez más y más profesionales vienen descubriendo que la tecnología avanza a pasos agigantados y que de alguna manera, debe mantenerse el ritmo. Asimismo, los investigadores de todos los rincones encuentran que su trabajo prospera en un ambiente enlazado con redes.

Internet es una red o conjunto de redes de computadoras interconectadas entre sí a nivel mundial para la comunicación de datos. Sus usuarios, forman parte de todo tipo de instituciones, ya sea de investigación, docencia o gubernamental comercial. Esta es la red de computadoras más grande del mundo, con un crecimiento exponencial sin precedentes.

Para comunicarse entre sí, las computadoras necesitan 'hablar' un mismo lenguaje (protocolo). En la red Internet el protocolo utilizado se denomina TCP/IP (Transport Control Protocol/Internet Protocol). Por lo tanto, para conectar una computadora a Internet, además de la conexión física, se requiere que el protocolo TCP/IP este instalado en dicha computadora. A diferencia de otros protocolos de comunicación, existen implementaciones de TCP/IP para prácticamente todas las marcas y modelos de computadoras, lo que explica su aceptación y utilización en todo el mundo.

Para navegar por Internet y convertirse en *Internauta*, se utilizan varios programas entre los que destacan:

- Telnet: Para establecer sesiones interactivas en otras computadoras.
- Mail: Para enviar y recibir mensajes de correo electrónico.
- FTP: Para transferir archivos desde y hacia otras computadoras.
- WWW: Para viajar a través de documentos multimedia.

Antes de dar paso a una descripción breve de los servicios, es indispensable entender algunos conceptos importantes.

Una **dirección Internet** es un *identificador único* tanto para las claves de acceso de los usuarios como para las computadoras conectadas a la red en donde residen dichas claves. En general, la dirección de una computadora conectada a Internet tiene la forma:

computadora.dominio

Por ejemplo, si hablamos de la computadora "sella" instalada en la Dirección de Computo para la Administración Académica de la UNAM cuyo dominio es `dcaa.unam.mx`, su dirección Internet se compone de la siguiente manera:

`sella.dcaa.unam.mx`

A esta forma de dirección de Internet se le conoce como **dirección alfabética**. Internamente la red convierte esta dirección alfabética en una **dirección numérica**, por ejemplo, la dirección numérica 132.248.63.9 corresponde a la computadora sella. Se puede utilizar cualquiera de las dos formas.

Es importante aclarar que las direcciones Internet deben teclearse sin espacios en blanco y en minúsculas. De la misma manera, la dirección Internet de la clave de un usuario (dirección electrónica) en una computadora conectada a Internet se compone de la siguiente forma:

`clave@computadora.dominio`

Por lo general, las direcciones de correo electrónico aparecen en un formato en que se emplea un signo de "en" (el símbolo @). Así la dirección electrónica para el usuario cristy en la computadora sella sería:

`cristy@sella.dcaa.unam.mx`

La dirección Internet de una computadora se utiliza con los servicios Telnet y FTP. La dirección Internet de un usuario se utiliza para el correo electrónico o Mail (es importante mencionar que para el correo electrónico solamente se puede utilizar la dirección alfabética).

1.2.1 Telnet

Una de las aplicaciones más comunes de una red de computadoras es establecer sesiones remotas en otras computadoras, las cuales no están físicamente en el lugar donde se realiza la conexión. Para realizar esto se utiliza la herramienta conocida como TELNET, la cual proporciona al usuario la oportunidad de estar en un sistema de computadora y trabajar en otro, que puede estar al otro lado de la calle o a miles de kilómetros de distancia. El usuario podrá tener acceso a todos los servicios que esa máquina provee; puede realizar una sesión interactiva normal o tal vez tener acceso a muchos servicios especiales, como buscar en algún catálogo de biblioteca y aprovechar los servicios disponibles en los diferentes equipos de red.

Ejemplo:

```
sella% telnet vikingo.dcaa.unam.mx

Trying 132.248.63.10...
Connected to vikingo.dcaa.unam.mx.
Escape character is '^]'.
login: cristy          Identificándose en el sistema remoto
Password:
Last login: Tue Oct 13 14:22:05 from sella.dcaa.unam.mx
Sun Microsystems Inc. SunOS Generic August 1997
vikingo% ls           Comando que se ejecuta en el sistema remoto.
    mail              proctool          TESIS              richpae
vikingo% logout      Desconexión del sistema remoto.
Connection closed by foreign host.
sella%              De nuevo está en el sistema local
```

Telnet encuentra una computadora llamada vikingo.dcaa.unam.mx, pide los datos necesarios de identificación con el sistema y después inicia la sesión. Una vez que inicia la sesión, se puede ejecutar comandos que son apropiados para el sistema remoto. Cuando termina la sesión en el sistema remoto, telnet termina su ejecución. Cualquier comando que se de será ejecutado por el sistema local.

1.2.2 Correo electrónico

El deseo de comunicarse es la esencia de la conexión en redes. En cualquier red de computadoras, lo más importante no son las computadoras sino las personas que lo utilizan. Las personas siempre han deseado establecer correspondencia de la manera más rápida posible, es por ello que de todos los servicios que se ofrecen en una red, el correo electrónico (**Electronic Mail** o **E-Mail**) es el más utilizado. Esta aplicación permite a las personas escribir en un sentido y en otro sin tener que gastar mucho dinero por la manera en la que se envía el mensaje. Conforme la tecnología se acerca más y más a convertirse, al menos en cierto nivel, en una parte común de la vida diaria, es vital la necesidad de comprender la manera en que puede emplearse y cómo funciona.

El correo electrónico difiere de otras aplicaciones porque no es un servicio de usuario a usuario; no es necesario que las máquinas emisora y receptora del correo electrónico se comuniquen directamente entre sí. Al correo electrónico se le conoce como un servicio de "almacenaje y reenvío". El correo electrónico pasa de una máquina a otra hasta que llega a su destino final.

El correo electrónico depende del concepto de *dirección de correo electrónico*, que proporciona toda la información que se requiere para que una persona reciba mensajes de cualquier parte del mundo. Si el mensaje llega o no a su destino, dependerá únicamente de que la dirección de e-mail haya sido construida correctamente.

Los sistemas UNIX cuentan con un programa llamado **PINE**, que permite enviar, recibir y eliminar mensajes. Este programa es útil para transferir todo tipo de archivo que de otra forma será muy complicado. No se hará una revisión extensiva de sus funciones, pues cuenta con una buena ayuda en línea. Su simplicidad y el hecho de estar orientado a menús permite que sea empleado por la mayoría de los usuarios familiarizados con el sistema UNIX. Pine tiene una funcionalidad muy elemental, pero es fácil de usar, funciona en modo de pantalla completa en terminal ASCII estándar.

En la ventana de pine (Fig. 1.2) se pueden apreciar los principales comandos que nos permiten manipular los mensajes electrónicos, estos son teclas individuales. Pine arranca con el siguiente comando:

```
%pine
```

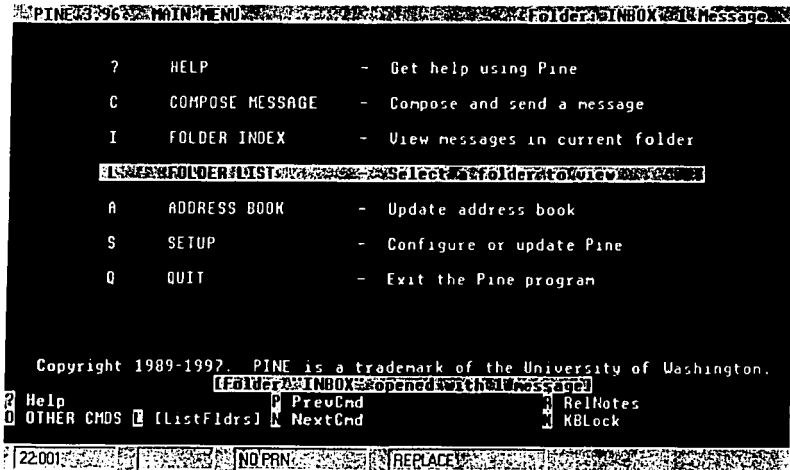


Fig. 1.2 Menú principal del programa PINE.

1.2.3 FTP

El protocolo de transferencia de archivos (**File Transfer Protocol** o **FTP**) es el método primario para la transferencia de archivos por Internet. Como su nombre lo indica, la función de este protocolo es transferir archivos de una computadora a otra, no importa donde se encuentren físicamente, cómo están conectadas o qué sistema operativo tengan siempre y cuando las computadoras tengan instalado el protocolo FTP y estén conectadas a Internet. En muchos sistemas, es también el nombre del programa que emplea el protocolo.

TESIS CON
FALLA DE ORIGEN

FTP es un protocolo complejo porque existen muchos tipos de archivos así como diferentes formas de manejo y estructuras de archivos. La diferencia de cómo se almacenan los archivos (binarios, ASCII, comprimidos o descomprimidos, etc.) introduce algunas complicaciones que requieren de un momento de análisis para que la transferencia salga bien.

En Internet están disponibles muchos archivos que sólo esperan ser llevados a una máquina a través de FTP. Probablemente encontramos un programa de dominio público que puede resolver algún problema o quizá, una receta de cocina. En cualquier caso, necesitamos transferir una copia de esa información a nuestra computadora, con el permiso adecuado, es posible copiar un archivo de una computadora a otra a velocidades muy grandes.

Normalmente FTP puede utilizarse con una cuenta de usuario en ambos sistemas o una configuración especial establecida por el administrador del sistema, por lo que será posible enviar y bajar archivos en la cuenta propia. De esta forma se puede guardar archivos de manera centralizada, intercambiar archivos con otros usuarios y abrir archivos actuales.

Al igual que telnet, FTP requiere que se especifique la máquina de la cual se quiere extraer los archivos, para ello se puede usar el comando `ftp` que permite a un usuario efectuar una conexión con la computadora remota, proporcionando una clave y una contraseña:

```
% ftp e5000.dcaa.unam.mx
Connected to e5000.dcaa.unam.mx.
220 e5000 FTP server (SunOS) ready.
User (e5000.dcaa.unam.mx:(none)): cristy
331 Password required for cristy.
Password:
230 User cristy logged in.
ftp>
```

Una vez que el sistema remoto acepta la cuenta y la contraseña, se encuentra listo para empezar a transferir archivos. FTP despliega `ftp>` como indicador para recibir comandos; y transferir archivos en dos direcciones. Es posible tomar un archivo en la máquina local (la que inicia la transferencia) y colocarlo (`put`) en la máquina remota, o extraerlo (`get`) y colocarlo en la máquina local. Los comandos `get` y `put` tienen la siguiente sintaxis:

```
ftp> get archivo-fuente archivo-destino
ftp> put archivo-fuente archivo-destino
```

Archivo fuente es el nombre del archivo existente (el archivo que se quiere copiar) y *archivo destino* es el nombre de la copia recientemente creada. El *archivo destino* es opcional, si se omite, se asigna el nombre del archivo fuente a la copia. En el siguiente ejemplo, se muestra una conexión a la máquina `sella.dcaa.unam.mx` con el usuario `cristy`. Se transferirá el archivo `ficha_tecnica` localizado en el directorio que pertenece al usuario `cristy`, de la máquina `vikingo.dcaa.unam.mx`:

```
sella% ftp vikingo
Connected to vikingo.dcaa.unam.mx.
Name (vikingo:cristy): cristy Usuario cristy en vikingo.
331 Password required for cristy.
Password:
230 User cristy logged in.
ftp> get ficha_tecnica Solicitud de copia del archivo.
200 PORT command successful. Se desplaza archivo.
226 ASCII Transfer complete.
local: ficha_tecnica remote: ficha_tecnica
3658 bytes received in 0.029 seconds (1.2e+02 Kbytes/s)
ftp> bye Finaliza la sesión.
221 Goodbye.
sella%
```

Cuando se termine de transferir los archivos, se teclea el comando `quit` para terminar la ejecución del programa `ftp`. El comando `bye` hace lo mismo; dependiendo de los conocimientos del usuario puede ser que recuerde más fácilmente el comando `bye`. Finalmente después de cada transferencia, el programa informa qué transfirió, qué tan grande es el archivo y cuál fue la tasa promedio de transferencia.

1.2.3.1 FTP Anónimo

Con lo que se ha explicado hasta aquí, es difícil poner un archivo a disposición de muchos usuarios. Por ejemplo, si se desea distribuir un paquete de software, se tendrá que poner en algún sistema y después, repartir combinaciones de claves de usuario para todo aquel que quisiera el software. Esto sería una carga especialmente para el administrador pero también para el usuario.

Con *FTP Anónimo* es posible evitar esta limitación, lo que permitiría a los usuarios no tener la necesidad de contar con una clave de usuario o una contraseña para poder tener acceso a archivos en una máquina. Obviamente, existen algunas restricciones, normalmente los usuarios anónimos sólo pueden ejecutar el comando `get` de los archivos (esto es, copiarlos), no podrán instalar archivos nuevos o modificar archivos que ya existen, lo que sería equivalente a ejecutar el comando `put`. Y existen límites estrictos sobre que archivos se pueden copiar.

Cuando se habilita un servidor FTP anónimo, se crea una clave de usuario especial llamada *anonymous*. Si el usuario inicia FTP, establece una sesión en una computadora remota y da la clave de usuario *anonymous*, se aceptará cualquier conjunto de caracteres como contraseña. Se considera de buen gusto usar como contraseña la dirección de correo electrónico, para que los administradores del servidor pueda comunicarse con el usuario si es necesario. Después de haber establecido la conexión, se podrá obtener aquellos archivos que estén permitidos expresamente a los usuarios de los servidores FTP anónimo.

Al igual que una contraseña normal, la dirección de correo electrónico no se despliega en la pantalla cuando se teclea.

1.2.4 World Wide Web

El progreso de software de hoy no está asociado solamente con el hecho de que los programas sean cada vez más capaces y ricos en nuevas herramientas, sino que también sean cada vez más fáciles e intuitivos de manejar. Esto es también válido para Internet.

El acceso más intuitivo y agradable a la información de Internet es el **World Wide Web** o **WWW**, que es un sistema universal que nos permite tener acceso de forma muy sencilla a diferentes tipos de información tales como archivos de texto, imágenes, sonidos, videos, etc.

Algo que caracteriza al Web es el uso de *Hipertexto*, es decir, un texto que contiene referencias hacia otros documentos en forma dinámica, llamadas comúnmente *ligas*, las cuales se distinguen del texto común mediante un color, generalmente azul, que nos indica que la palabra o frase que se encuentra marcado de esta manera hace referencia a un documento adicional. Para acceder a esta información basta con presionar el botón del ratón sobre la referencia. Con multimedia esto puede hacerse todavía de forma más atractiva.

Para poder explorar la información contenida en los documentos del World Wide Web en los servidores de Internet, se necesita contar con algún visualizador de Web, como Netscape o Internet Explorer. Estos programas permiten visualizar páginas Web de los más diversos temas, almacenados en diferentes servidores de Internet. Cada uno de ellos se identifica de forma única con un URL (*Universal Resource Locator*), es decir un descriptor universal de su localización dentro de Internet.

Por ejemplo, si deseamos tener acceso a la página de la UNAM, se abre el navegador de Netscape y se le indica el URL correspondiente: <http://www.unam.mx>. Una vez hecho esto Netscape contactará al servidor correspondiente y presentará la página de la Universidad (Fig. 1.3):

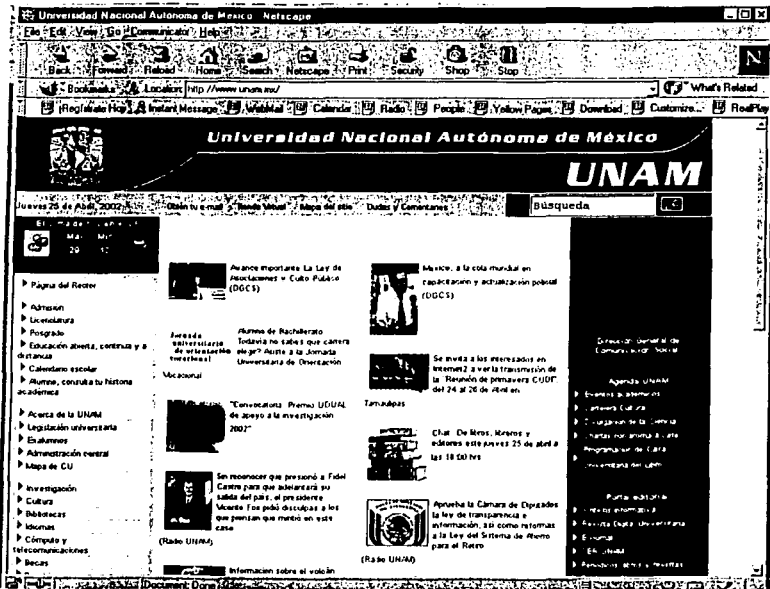


Fig. 1.3 Página principal de la UNAM.

Como se mencionó, cada página de Web tiene un identificador único, con el cual se le puede encontrar entre todas las páginas existentes en Internet. Es común que cuando se han encontrado las páginas deseadas, se quieran conservar por algún tiempo, para ello se cuenta con la facilidad de "marcar" algunas páginas y guardar su URL en una lista para que después podamos llegar a ellas rápidamente. Estas marcas de libro (*bookmarks*) se almacenan bajo el menú **Bookmarks** en caso de Netscape o el menú **Favoritos** en el caso de Internet Explorer. La dirección de la página quedará almacenada en la lista de marcas para poder regresar a ella toda las veces que se necesite.

TESIS CON
FALLA DE ORIGEN

2. DESEMPEÑO DEL SISTEMA



Obtener un buen desempeño de una computadora o de una red es una parte importante de la administración de un sistema. Aquí se describe como monitorear y administrar los procesos en ejecución en el sistema así como algunos factores que contribuyen a mantener y administrar el desempeño de un sistema UNIX particularmente; se muestra como encontrar algunos problemas, se proporcionan ejemplos y se listan algunas herramientas útiles para el monitoreo del desempeño así como las facilidades que UNIX ofrece para administrar los recursos del sistema (memoria, CPU y E/S). Es importante mencionar que una gran parte de administrar cualquier recurso del sistema es conocer su estado actual por lo que se toma algo de tiempo para monitorear el uso de los recursos. Finalmente se describe las herramientas que pueden ayudar a identificar problemas de desempeño en la red (conexiones físicas, tráfico en la red, desempeño NFS, etc.).

2.1 ADMINISTRACION DE PROCESOS

La noción de proceso es uno de los aspectos más importantes del sistema UNIX, junto con los archivos, directorios y el shell. Un *proceso*, o tarea, es una instancia de un programa en ejecución. En un sentido simple, un proceso es un programa en ejecución. Sin embargo, puesto que un programa puede crear nuevos procesos (por ejemplo, el shell genera nuevos shells), para un programa dado puede haber uno o más procesos en ejecución. Es importante hacer la distinción entre una orden y un proceso; un proceso se genera cuando se ejecuta una orden. El sistema UNIX es un sistema multitarea porque puede ejecutar muchos procesos al mismo tiempo. En cualquier momento puede haber decenas o incluso cientos de miles de procesos ejecutándose en el sistema.

En un sistema de tiempo compartido como el sistema operativo UNIX, los usuarios tienen la ilusión del uso exclusivo de la máquina, aun cuando puedan estar utilizándola simultáneamente docenas o centenares de otros usuarios. El kernel del sistema UNIX gestiona todos los procesos que se ejecutan en la máquina mediante el control de la creación, operación, comunicación y terminación de los procesos. Gestiona la distribución de los recursos de la computadora mediante la planificación de las fracciones de segundo durante las cuales el CPU está ejecutando un proceso y la suspensión y replanificación de un proceso cuando la asignación de tiempo de CPU se completa. Los procesos UNIX pueden ser de varios tipos:

1. *Procesos Interactivos.*

Son procesos iniciados y controlados desde una sesión en una terminal. Los procesos interactivos pueden correr en primer plano (*foreground*) o en segundo plano (*background*). Los procesos en primer plano permanecen asociados a la terminal; el proceso de primer plano es quien se comunica con la terminal directamente. Por ejemplo, teclear un comando UNIX y esperar su salida significa que esta corriendo un proceso en primer plano. Mientras un proceso de primer plano esta corriendo, solamente puede recibir la entrada de información directamente de la terminal.

El control de trabajos permite que un proceso en primer plano pase a segundo plano. Cuando esto pasa, el proceso es detenido temporalmente, y el control de terminal vuelve a su proceso de padre. De este modo, mientras se ejecuta el proceso en segundo plano, el shell nos devuelve el prompt y mientras se pueden ejecutar otros programas.

2. *Procesos Batch.*

Son procesos que no están asociados con una terminal. Estos procesos se guardan en una cola, en la cual los trabajos se ejecutan secuencialmente.

3. *Demonios.*

Son procesos del servidor, ejecutados generalmente en el momento de inicializar el sistema, normalmente están en segundo plano hasta que un cierto proceso requiere de sus servicios. Por ejemplo, los demonios de red están "dormidos" hasta que un cierto proceso solicita el acceso de red.

Los procesos UNIX tienen atributos asociados. Cada proceso tiene asociado un número que lo identifica. Este número es asignado por el kernel, y se denomina identificador de proceso o **PID** (*Process Identifier*). Además del PID, los procesos tienen asignado otro número denominado **PPID** (*Parent PID*), que identifica al proceso padre del proceso en cuestión. También cuentan con un dispositivo de terminal asociado o **TTY**, el cual es utilizado por el proceso para operaciones de lectura y escritura.

2.1.1 Creación y ejecución de un proceso

Un proceso se crea mediante una llamada de sistema `fork()`, la cual es una subrutina que hace que el kernel proporcione algún servicio a un programa. La llamada `fork()` crea un proceso en ejecución separado, pero casi idéntico. El proceso que hace uso de esta llamada se denomina *proceso padre*; el proceso que crea `fork()` se denomina *proceso hijo*.

Ambos procesos tienen el mismo entorno, es decir, las mismas especificaciones de manejo de señales, los mismos identificadores de grupo (GID) y de usuario (UID), la misma clase y prioridad, pero diferentes identificadores de proceso (PID). La única manera de incrementar el número de procesos que se ejecutan en un sistema UNIX es con la llamada de sistema `fork()`, es decir, cuando se ejecutan programas que generan nuevos procesos, lo hacen mediante esta llamada.

La llamada de sistema `fork()` realiza solamente la mitad del trabajo, ya que solo se encarga de duplicar el programa actual, sin embargo, el usuario desea que el proceso se ejecute. La solución es la llamada de sistema `exec()`, que se encarga de ejecutar un proceso. Esta llamada crea e inicializa el contexto para el nuevo proceso.

2.1.2 Eliminación de un proceso

Si se desea, se puede detener un proceso mientras está en ejecución. Por ejemplo, puede estar ejecutándose un programa que contiene un bucle sin fin, por lo que el proceso que se ha creado nunca se va a detener. O quizá se decida que un proceso iniciado termine su ejecución porque está acaparando recursos del sistema o porque está haciendo algo no deseado. Para ello, se usa el comando `kill`, dando el identificador de proceso (PID) como argumento. Por ejemplo, para eliminar un proceso con PID 232¹, debe teclearse:

```
# kill 232
```

El comando `kill` sirve para enviar *señales* a uno o varios procesos identificados por su PID. Esta orden envía una *señal de terminación de software* (SIGTERM) al proceso de forma asíncrona para detenerlo. Cuando un proceso recibe una señal, puede tratarla de tres formas diferentes:

¹ Para verificar los identificadores de los procesos en ejecución se utiliza el comando `ps`.

1. Ignorar la señal.
2. Invocar a la rutina de tratamiento por defecto proporcionada por el kernel.
3. Invocar a una rutina propia que se encargará de tratar dicha señal.

Para que los procesos se comuniquen entre sí y a su vez con el kernel hacen uso de las señales. Las señales son usadas para informar sobre eventos externos inesperados. Cada señal tiene asociado un número entero positivo que la identifica, las más comunes son:

1. *SIGHUP*: Hangup. Es enviada a todos los procesos asociados a una misma terminal cuando ésta se desconecta. La acción por defecto es terminar la ejecución de los procesos que la reciben.
2. *SIGINT*: Interrupción. Es enviada a todos los procesos asociados a una misma terminal cuando se pulsa la tecla de interrupción. Por defecto, provoca la terminación de los procesos que la reciben.
3. *SIGQUIT*: Salir. Es similar a *SIGINT*.
4. *SIGILL*: Instrucción ilegal. Se envía a cualquier proceso que intente ejecutar una instrucción de máquina ilegal. Por defecto, termina la ejecución del programa que la recibe.
5. *SIGTRAP*: Es enviada cuando se ejecuta instrucciones paso a paso en un programa. Su acción por defecto también es terminar el proceso que la recibe.
6. *SIGIOT*: Fallo de hardware.
7. *SIGEMT*: Fallo de hardware.
8. *SIGFPE*: Es enviada cuando el hardware detecta un error en una operación en coma flotante. Por defecto, termina la ejecución del proceso que la recibe.
9. *SIGKILL*: Provoca la terminación del proceso. Esta señal no puede ser ignorada.
10. *SIGBUS*: Error de acceso a memoria.

11. *SIGSEGV*: Violación de segmento de memoria.
12. *SIGYS*: No se usa.
13. *SIGPIPE*: Intento de escritura en una tubería en la cual no hay nadie leyendo.
14. *SIGALARM*: Es enviada al proceso cuando alguno de sus temporizadores llega a cero. Provoca por defecto la terminación del proceso.
15. *SIGTERM*: Indica a un proceso que debe terminar su ejecución. Puede ser ignorada.
16. *SIGUSR1*: Reservada para el usuario.
17. *SIGUSR2*: Reservada para el usuario.
18. *SIGCLD*: Se envía al padre de un proceso si éste muere.
19. *SIGPWR*: Fallo de alimentación.

La orden `kill` debe ser empleada por el propietario de los procesos o el superusuario. Por defecto, `kill` envía la señal número 15 al proceso especificado con intención de terminar su ejecución. Esta señal avisa al proceso que termine por sí mismo pero puede ser ignorada. Sin embargo, existen algunos procesos, tales como el shell, que no mueren cuando reciben esta señal. Se puede eliminar los procesos que ignoran la señal `SIGTERM` agregando a la orden `kill` la opción `-9`. De este modo se envía la *señal de eliminación incondicional*. Por ejemplo, para eliminar un proceso con PID 517, debe teclearse:

```
$ kill -9 517
```

2.1.3 Monitoreo de procesos

Los procesos en ejecución consumen recursos del sistema. El desempeño de un sistema se puede regular controlando y monitoreando los procesos que el CPU esta ejecutando. El comando `ps` permite monitorear los procesos activos en un sistema, dependiendo de cual opción se use reporta los siguientes campos (Tabla 2.1):

Campo	Descripción.
F	Banderas en hexadecimal, que indican el estado actual de los procesos. 00 El proceso ha terminado. 01 Proceso de sistema que siempre esta en memoria. 02 El proceso esta siendo rastreado por el padre de dicho proceso. 04 El proceso esta siendo rastreado por el padre y ha sido detenido. 08 El proceso no puede ser activado por una señal. 10 El proceso esta actualmente en memoria y es cerrado hasta que un evento sea terminado. 20 El proceso no puede ser movido a disco.
S	El estado actual del proceso. O El proceso esta ejecutándose actualmente. S El proceso esta dormido. Un proceso entra en este estado, cuando esta esperando a que un evento de E/S sea completado o no puede proseguir su ejecución por falta de algún recurso. R El proceso esta listo para ejecutarse. I El proceso no esta siendo ocupado. Z Proceso zombie; el proceso ha terminado pero aún permanece en la tabla de proceso. T El proceso es detenido por una señal de control o porque esta siendo rastreado. X El proceso esta esperando más memoria.
UID	Identificador del usuario.
PID	Identificador del proceso
PPID	Identificador del proceso padre.
C	Este campo muestra el uso de CPU, este campo no es desplegado cuando la opción <code>-c</code> es usada.
CLS	Clase asignada al proceso: tiempo real, sistema o tiempo compartido.
PRI	Prioridad asignada al proceso, los números altos significan prioridades bajas.
NI	Número nice del proceso, el cual contribuye a su asignación de prioridad. Modificar este número significa disminuir la prioridad de un proceso para que no use mucho tiempo de CPU.
ADDR	Dirección de memoria del proceso.
SZ	Cantidad de memoria virtual requerida por el proceso
WCHAN	Dirección de un evento por el cual un proceso esta dormido. Si este campo esta en blanco el proceso esta en ejecución.
TTY	Terminal en la cual el proceso fue iniciado. El símbolo ? indica que no hay una terminal de control (usualmente es un proceso de sistema).
TIME	Cantidad total de tiempo de CPU usado por el proceso desde que fue iniciado.
STIME	Tiempo de ejecución acumulativo desde que se inicio el proceso en horas, minutos y segundos.
COMD	Comando que origino el pro.-eso.

Tabla 2.1 Campos que despliega el comando ps.

Ejemplos:

Si se teclea el comando `ps` sin argumentos muestra información de los procesos asociados con la terminal de control.

```
% ps
```

```
PID  TTY  TIME  COMD
1645 pts/4 0:06  chs
2438 pts/4 0:00  ps
```

Al teclear `ps -ef` se despliega una lista completa(-f) de todos los procesos que están siendo ejecutados en el sistema(-e).

```
% ps -ef
```

```
UID  PID  PPID  C  STIME  TTY  TIME  CMD
root  0    0    0  Apr 29  ?    0:31  sched
root  1    0    0  Apr 29  ?    0:01  /etc/init
root  2    0    0  Apr 29  ?    0:00  pageout
unam 3686  460  0  May 03  ?    0:04  -csh
tuning 2128 1664  0  May 03  pts/4 0:00  ps -ef
```

La salida del comando `ps` muestra que el primer proceso ejecutado cuando se inicio el sistema es el proceso `sched` seguido por el proceso `init`, `pageout` y así sucesivamente.

2.1.4 Niveles de prioridad

A los procesos en un sistema UNIX se les asignan recursos para su ejecución de manera secuencial. El kernel asigna el CPU a un proceso durante un intervalo de tiempo; cuando el tiempo ha transcurrido, el proceso se coloca en una de varias colas de prioridad. Dependiendo de como se planifique la ejecución es el nivel de prioridad asignado al proceso. Los procesos del sistema tienen una prioridad superior a la de todos los procesos de usuario. Las prioridades de los procesos de usuario dependen de la cantidad de tiempo de CPU que hayan empleado, es decir, la prioridad asignada determina la cantidad de tiempo que el CPU dedica al proceso.

TESIS CON
FALTA DE ORIGEN

Los procesos que han utilizado grandes cantidades de tiempo de CPU tienen asignadas prioridades inferiores; los que utilizan poco tiempo de CPU tienen asignadas prioridades altas. La planificación de tareas realizada correctamente optimiza los tiempos de respuesta, ya que los procesos que acaparan el procesador reciben prioridades inferiores para asegurar que las órdenes nuevas comiencen su ejecución. En general, un sistema operativo UNIX tiene 3 clases de prioridades de procesos (Fig. 2.1):

- *Procesos de tiempo real.*

Los procesos de tiempo real tienen la prioridad más alta. Esta clase incluye aquellos procesos que deben responder a eventos externos. Por ejemplo, un proceso que recoge datos de un dispositivo puede necesitar procesar los datos y responder inmediatamente. En la mayoría de los casos, los procesos de tiempo real requieren un sistema dedicado. Otros procesos no pueden ser atendidos mientras un proceso de tiempo real tiene el control de CPU.

- *Procesos del sistema.*

Los procesos del sistema tienen prioridades medias. Esta clase consta de aquellos procesos especiales necesarios para el funcionamiento del sistema y de aquellos procesos que son ejecutados automáticamente por el kernel, tales como el demonio de swapping y paginación; de los cuales se hablara más adelante. Los procesos de esta clase son controlados por el proceso *scheduler* el cual se encarga de asignar sus prioridades y no pueden ser cambiadas.

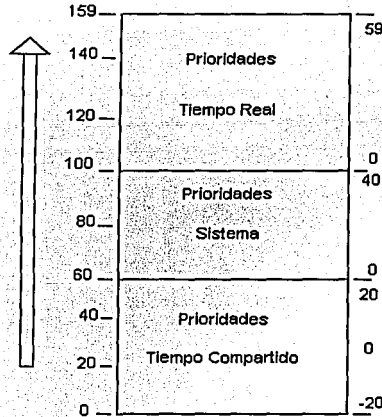
- *Procesos de tiempo compartido.*

Los procesos de tiempo compartido tienen la prioridad más baja. Esta clase incluye los procesos UNIX estándar. Normalmente, todos los procesos de los usuarios son procesos de tiempo compartido. Estos están sujetos a una política de planificación que intenta distribuir el tiempo de procesamiento equitativamente dando aplicaciones interactivas con un tiempo de respuesta rápido manteniendo así una buena carga de trabajo total en el sistema.

Estos procesos también son controlados por el proceso scheduler, el cual se encarga de asignar dinámicamente sus prioridades.

Puesto que la planificación de procesos (y las prioridades en que basan) puede afectar en gran medida a la capacidad de respuesta del sistema global, el sistema UNIX no permite mucho control al usuario sobre la planificación de procesos en tiempo compartido.

Cada clase de procesos tiene su propio conjunto de números de prioridad. Por ejemplo, las prioridades de un proceso de tiempo real van de 0 a 59. Los procesos de tiempo compartido usan números de prioridades desde -20 hasta 20 por default. Las prioridades de un proceso de sistema van de 0 a 40. Esas 3 clases específicas de conjuntos de números de prioridades son identificadas por un sólo conjunto de números de prioridad interno que van del 0 a 159.



TESIS CON
FALLA DE ORIGEN

Fig. 2.1 Prioridades de los procesos.

2.1.5 Control de prioridades de procesos

El sistema UNIX usa un algoritmo de asignación de prioridad (*scheduler*) para distribuir los recursos de CPU entre múltiples procesos. Todos los procesos tienen una prioridad de ejecución asignada, un valor entero que es dinámicamente asignado y actualizado de acuerdo a diferentes factores. Siempre que el CPU este libre, el proceso *scheduler* selecciona el proceso más favorecido para comenzar a ejecutarse, usualmente corresponde al proceso con el número de prioridad más bajo.

Aunque puede haber una multitud de procesos simultáneamente presentes en el sistema, solo un proceso usa el CPU en un momento dado (se asume que el sistema tiene un solo CPU). Una vez que un proceso comienza a ejecutarse, este continuara ejecutándose hasta que necesite esperar a que una operación de E/S se termine o reciba una interrupción, por lo tanto voluntariamente o involuntariamente cede el control al CPU. Una vez que el proceso detiene su ejecución, el algoritmo de asignación de prioridad selecciona una vez más un proceso del sistema y comienza su ejecución.

Múltiples procesos en el mismo nivel de prioridad son colocados dentro de una cola de ejecución (*run queue*) para ese nivel de prioridad. Cuando el CPU esta libre, el algoritmo de asignación comienza con los procesos con numeración más baja que se encuentran a la cabeza de la cola. Cuando el proceso que se encuentra en la cima de cola detiene su ejecución, este pasa al final de la línea y el siguiente proceso se mueve al frente.

La prioridad de un proceso es determinada por su clase y por su número *nice*, y es asignada por el proceso *scheduler*. Un proceso tiene dos números de prioridades:

- Su número *nice*, el cual es su prioridad de ejecución con respecto a otros procesos y es modificable por el propietario del proceso y por root. Este valor aparece en la columna NI al ejecutar el comando `ps -l`. Los procesos de tiempo compartido comienza con un número *nice* por omisión, el cual heredan del proceso padre.

- Su actual prioridad de ejecución, la cual es calculada y dinámicamente actualizada por el sistema operativo, se determina por varios factores: el número nice (en el caso de los procesos de tiempo compartido), el tiempo de CPU utilizado recientemente, su clase asignada y qué otros procesos están corriendo entre otros factores. Este valor aparece en la columna **PRI** al ejecutar el comando `ps -l`.

```
% ps -l
```

```
F S UID PID PPID C PRI NI ... TIME CMD
8 S 103 8934 8932 0 41 20 ... 0:00 csh
```

La opción `-l` indica que se va a generar una lista larga y completa sobre la información de un proceso.

En UNIX, menos significa más, es decir, un proceso con un número de prioridad menor se ejecuta más rápido. Un usuario puede decrementar la prioridad de un proceso incrementando su número nice, pero sólo el administrador del sistema puede incrementar la prioridad de un proceso, es decir, puede especificar números de prioridad menores que el de omisión. Esto es para prevenir que los usuarios puedan incrementar las prioridades de sus propios procesos y por lo tanto usar más recursos del CPU.

El rango de los números nice va de 0 a +40, con 0 se confiere la prioridad más alta. Existen 2 versiones del comando `nice` disponibles, las cuales pueden ser empleadas por los usuarios para iniciar un proceso con una prioridad más baja; la versión estándar `/user/bin/nice` o `/bin/nice`, y la versión que es parte del C-shell (csh) ambos trabajan de la misma manera pero difieren significativamente en su sintaxis.

- La versión estándar tiene la siguiente sintaxis:

```
% /user/bin/nice - [+ | -n ] comando
```

- La versión C shell tiene la siguiente sintaxis:

```
% nice [+ | -n ] comando
```

En la versión estándar, el valor para incrementar o decrementar es dado como una opción y es precedido por un guión. Si no se especifica ningún número, el número *nice* es incrementado en +10 (la prioridad de ejecución del proceso disminuye en 10 unidades). No es necesario preceder un número positivo con un signo más, sin embargo, es necesario preceder un número negativo con un guión. Por ejemplo, para incrementar el número *nice* de 20 a 26:

```
# /user/bin/nice -6 comando
```

Para decrementar el número *nice* de 20 a 14:

```
# /user/bin/nice -- -6 comando
```

En la versión *csh* recibe como argumento un número con signo, el cual especifica el valor que será incrementado o decrementado al número *nice* asignado, correspondiente a la prioridad del proceso tomando como base la prioridad por omisión. Si no se especifica ningún número, el número *nice* es incrementado en +4 (la prioridad de ejecución del proceso disminuye en 4 unidades). Por ejemplo, para incrementar el número *nice* de 20 a 26:

```
# nice +6 comando
```

Para decrementar el número *nice* de 20 a 14:

```
# nice -6 comando
```

Cuando los procesos son creados heredan la prioridad de ejecución de su padre, sin embargo, al cambiar la prioridad del proceso padre no se modifican las prioridades de sus hijos.

Otro comando que también sirve para modificar las prioridades de un proceso es `prionctl`. La forma del comando `prionctl` para cambiar la prioridad de un proceso es:

```
# prionctl -s -p nuevo_pri -i pid id_pro
```

donde:

`nuevo_pri` es la nueva prioridad para el proceso.

`id_pro` es el identificador del proceso deseado.

Por ejemplo, el siguiente comando configura (-s) el nivel de prioridad (-p) para el proceso con PID número 8733 (-i):

```
# prionctl -s -p -5 -i pid 8733
```

Cuando el identificador de proceso corresponde a un proceso padre, este comando configura la prioridad del padre y de todos sus hijos.

2.2 ADMINISTRACION DEL DESEMPEÑO

Antes de obtener técnicas muy específicas para la optimización de un sistema, es importante entender algunos principios involucrados en este campo, por tal motivo es necesario entender los conceptos relacionados con la administración del desempeño de un sistema.

2.2.1 Fundamentos del desempeño de un sistema

El desempeño o rendimiento de un sistema depende de cómo el sistema usa y asigna sus recursos. Los recursos más importantes de un sistema desde una perspectiva de desempeño son CPU, memoria y el sistema de E/S.

Las tareas que se ejecutan en el sistema siempre están restringidas por sus recursos y por ello es necesario monitorear el desempeño del sistema para saber cuánto trabajo se está realizando y qué recursos están siendo consumidos, para tener una idea de cómo trabaja el sistema bajo condiciones normales y su tiempo de respuesta. Cabe mencionar que muchos factores diferentes juegan un papel importante en la determinación de respuesta de un sistema. El uso de patrones, configuración de E/S, configuración de CPU y configuración de software todos contribuyen al desempeño del sistema.

Es fácil hablar de desempeño del sistema pero es difícil entenderlo. El concepto de rendimiento de un sistema de cómputo significa diferentes cosas para diferentes personas, para un administrador del sistema puede ser maximizar carga de trabajo (trabajo a ser ejecutado) y para un programador puede ser el tiempo de respuesta del sistema. El *desempeño* con relación a un sistema de cómputo es la forma en la cual el sistema ejecuta sus tareas y sus responsabilidades.

Las actividades que conforman la tarea de administración del desempeño de un sistema son:

- Monitorear el desempeño del sistema.
- Identificar problemas de desempeño.
- Hacer uso de herramientas UNIX específicas para recolectar datos del sistema.
- Diseñar e implementar modificaciones en el sistema sin son necesarias.

Considerando lo anterior, la *administración del rendimiento* de un sistema es la medición, análisis y optimización de los recursos del sistema para proporcionar un nivel adecuado de servicio para una organización y sus usuarios finales.

2.2.2 Herramientas de monitoreo

Cuando el sistema está lento o los trabajos de los usuarios se tardan en ejecutarse, se tiene que considerar la manera de mejorar la situación, particularmente si el sistema sufre un cambio repentino en su desempeño. Es difícil decir lo que constituye un bajo desempeño, sin embargo, es fácil obtener una variedad de argumentos que tienen que ver con el rendimiento del sistema y la única manera de identificar los problemas particulares de utilización del sistema es analizar rigurosamente esa información para decidir cuál es la mejor manera de resolver cualquier problema. Existen herramientas estándar disponibles en cualquier sistema UNIX que proporcionan información del desempeño del sistema en general:

<code>vmstat</code>	Reporta información de la actividad de la memoria virtual y otras estadísticas del sistema.
<code>netstat</code>	Despliega el contenido de las estructuras de datos relacionadas a la red.
<code>nfsstat</code>	Despliega estadísticas sobre el Sistema de Archivos de Red (NFS).
<code>iostat</code>	Proporciona información del sistema de E/S y de CPU.
<code>uptime</code>	Reporta la carga promedio total del sistema.
<code>ps</code>	Reporta información sobre los procesos que están corriendo en el sistema.
<code>sar</code>	Proporciona varias estadísticas sobre la actividad del sistema.
<code>nice</code>	Ejecuta un comando a una prioridad más alta o baja de la prioridad normal.
<code>renice</code>	Cambia la prioridad de uno o más procesos.
<code>ping</code>	Analiza el tiempo de respuesta de los sistemas de la red.

De esas herramientas, `ps` es indudablemente la más importante. Usando este comando, se puede obtener rápidamente información de lo que el sistema está realizando y qué está causando problemas de rendimiento. Este comando también proporciona la información más confiable de la carga de trabajo del sistema.

Esto es solo una lista parcial. Existen otras herramientas que proporcionan diferente información acerca del desempeño. A continuación se muestran algunos ejemplos de las herramientas administrativas (Tabla 2.2) que proporcionan los distintos sistemas operativos para monitorear los recursos de un sistema. Los detalles de cómo se utiliza cada una de esas herramientas se encuentran en la documentación de cada uno de los sistemas operativos.

Herramientas de monitoreo	Utilización de la herramienta	Solaris (SunOS)	AIX (IBM)	Linux	UnixWare (Caldera)
top	Muestra estadísticas de procesos en un sistema.	*	*	*	*
snoop	Captura paquetes de la red para aislar problemas de rendimiento.	*			
lsattr	Muestra atributos de los parámetros del kernel.		*		
lslv	Muestra información acerca de un volumen lógico o volumen físico.		*		
no	Despliega o activa parámetros de la red.		*		
reorgvg	Organiza la asignación de las particiones físicas con un grupo de volumen.		*		
iptraf	Utilería para mostrar el tráfico de la red.			*	
nfsppmon	Herramienta para el análisis del desempeño de NFS.			*	
idtune	Permite modificar el valor de los parámetros regulables del kernel.				*
prfstat	Mecanismo de muestreo del desempeño del sistema.				*
rptm	Monitor interactivo de desempeño en tiempo real.				*

TESIS COM
FALLA DE ORIGEN

Tabla 2.2 Herramientas administrativas para el desempeño.

También, existen herramientas distribuidas en Internet que proporcionan información relacionada con el rendimiento de un sistema. A continuación se listan algunas de ellas, las cuales operan exclusivamente para el sistema operativo Solaris:

- Proctool* Genera un reporte visual que describe varias estadísticas significantes del sistema. Esta herramienta está basada en una interfaz gráfica de usuario (GUI).
- Memtool* Es una herramienta con interfaz gráfica de usuario (GUI) que rastrea el uso de memoria generado por archivos y procesos.
- SE* Kit de herramientas para el monitoreo y optimización del desempeño de un sistema.

2.2.3 Recursos del sistema

El reto más grande de un administrador es localizar un problema de rendimiento identificando los recursos en cuestión (CPU, memoria y sistema de E/S) y entender cómo administrarlos más eficientemente. No siempre es tan obvio como parece. Estos recursos pueden ser comparados con el tráfico de una autopista. Con un tráfico normal, los carros viajan de su origen a su destino a una velocidad razonable. Con más tráfico, el tráfico es lento y posiblemente se detenga. Esto con relación a los recursos de un sistema de cómputo se conoce como *cuello de botella*. Estos ocurren cuando los recursos van más allá de su capacidad debido a actividades errantes o porque la carga de trabajo es demasiado grande.

Cuando la falta de un recurso crítico es la fuente de un problema de desempeño; hay un número limitado de propuestas para mejorar la situación, esto puede significar, obtener más (si es posible), reducir trabajo o peticiones del sistema.

A menudo es necesario distinguir entre los recursos propios del sistema y los mecanismos de control (Tabla 2.3) por los cuales ellos son asignados:

RECURSO	MECANISMOS DE CONTROL
CPU	Números nice. Prioridades de procesos. Colas batch.
Memoria	Espacio de paginación. Límites de los recursos de proceso. Parámetros de administración de memoria.
E/S	Organización de los sistemas de archivos. Colocación de archivos en disco. Parámetros de E/S.

Tabla 2.3 Mecanismos de control asociados a los recursos de un sistema.

Entender el impacto que los recursos tienen sobre el desempeño de un sistema es complicado ya que todos ellos interactúan entre sí, por lo tanto es importante saber como opera cada uno. Aunque la arquitectura de una computadora varía (Fig. 2.2), la mayoría comparte algunos componentes comunes. A continuación se muestra un diagrama a bloques de un sistema de cómputo típico.

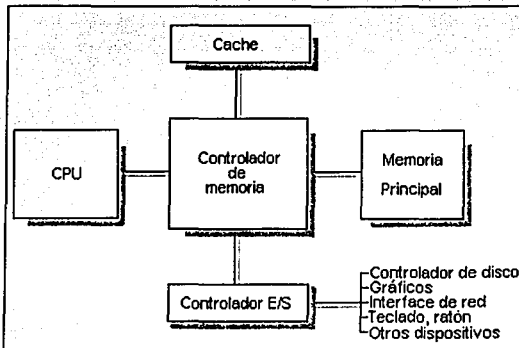


Fig. 2.2 Arquitectura básica de una computadora.

SIS CON
 PALMA DE ORIGEN

2.2.3.1 Monitoreo del desempeño del CPU

El CPU es la parte más importante de cualquier sistema de cómputo. Un sistema UNIX ejecuta los procesos enviándolos al CPU, entidad en dónde se ejecutan todas las instrucciones. En cualquier sistema de tiempo compartido, muchos programas requieren usar el CPU al mismo tiempo. En la mayoría de los casos el kernel UNIX se encarga de asignar el CPU, sin embargo, cada proceso requiere un cierto número de ciclos de CPU para ejecutarse y en un momento dado el CPU no podrá realizar todo el trabajo, por lo que es importante tener un control de los procesos que se ejecutan en el sistema.

La velocidad del CPU es controlada por un reloj (secuencia regulada de pulsos o ciclos) y se especifica en millones de ciclos por segundo o Megahertz (MHz). La velocidad del CPU no determina automáticamente la velocidad con la cual se ejecutan los programas de los usuarios, esto depende del número de instrucciones y el número promedio de ciclos por cada instrucción del programa. Por lo tanto, no debe usarse la velocidad del CPU para juzgar el rendimiento de un sistema, sin embargo, esto no quiere decir que un CPU con una velocidad alta no sea bueno. Si se cuenta con pocos ciclos de CPU, se tienen 4 alternativas básicas:

- Conseguir que los usuarios ejecuten trabajos durante la noche o cuando el sistema tiene poco uso.
- Prevenir que los sistemas ejecuten trabajos innecesarios.
- Conseguir que los usuarios ejecuten sus trabajos grandes a una prioridad baja.
- Agregar otro procesador para tener más capacidad de CPU.

El uso de CPU es usualmente el primer factor que se debe considerar cuando se trata de rastrear un problema de rendimiento o cuando se trata de estimar el estado actual de un sistema en general.

El uso general de un sistema puede ser determinado con el comando `uptime`, el cual despliega la hora actual, la cantidad de tiempo que el sistema ha estado activo, el número actual de usuarios y la carga promedio de trabajo durante los últimos minutos (1, 5 y 15 minutos respectivamente).

```
% uptime
```

```
2:55pm up 17days,19:32, 4users, load average: 0.00,0.00, 0.01
```

Si un sistema muestra una carga de trabajo alta, el comando `ps` proporciona información sobre cada proceso que está ejecutándose actualmente en el sistema y con ello se puede determinar que proceso o procesos están creando la carga de trabajo. Consideremos el siguiente ejemplo:

```
% ps -el
```

F	S	UID	PID	PPID	C	PRI	NI	...	TTY	TIME	CMD
19	T	0	0	0	0	0	SY	...	?	0:00	sched
8	S	0	1	0	0	41	20	...	?	0:03	init
19	S	0	3	0	0	0	SY	...	?	12:07	fsflush
8	S	0	163	1	0	41	20	...	?	5:24	automoun

Estas opciones son útiles para encontrar procesos que están usando mucho tiempo de CPU, denotado por la columna **TIME**.

Se dice que un proceso se "ejecuta lento" si está usando la mayoría de los recursos de CPU. Para un proceso pequeño esto puede ser un desempeño normal, sin embargo, la mayoría del tiempo indica un desempeño errante y el proceso debe ser terminado.

Es necesario usar el comando `ps` en conjunción con el comando `vmstat` para determinar los trabajos específicos que están consumiendo los recursos de CPU del sistema.

TESIS CON
FALLA DE ORIGEN

Su sintaxis más usual es:

```
% vmstat [intervalo] [contador]
```

donde:

intervalo: es el número de segundos entre reportes.

contador: es el número total de reportes a generar.

```
% vmstat 5 3
```

procs			memory		...	cpu		
r	b	w	swap	free	...	us	sy	id
0	0	0	10000	10624	...	10	12	78
0	0	0	487280	4720	...	0	0	100
0	0	0	487280	4720	...	0	0	100

En el ejemplo anterior, consideremos las últimas columnas del reporte:

us: Porcentaje de ciclos de CPU gastados en la ejecución de aplicaciones de usuarios (tiempo de usuario).

sy: Porcentaje de ciclos de CPU gastados en la ejecución de peticiones de E/S del sistema (tiempo de sistema).

id: Porcentaje de ciclos de CPU que no fueron usados durante el intervalo (tiempo ocioso).

Otra forma de medir la carga de CPU es por medio del comando `sar`, cuya sintaxis es similar al comando anterior:

```
% sar [-u] [intervalo] [contador]
```

Ejemplo:

```
% sar -u 5 4
```

```
SunOS 5.6 Generic_105181-06 sun4u 08/31/99
```

	%usr	%sys	%wio	%idle
19:18:03				
19:18:08	0	0	0	100
19:18:13	0	0	0	100
19:18:18	0	0	0	99
19:18:23	0	1	1	98
Average	0	0	0	99

La salida del ejemplo anterior muestra el uso de CPU (-u), mostrando el porcentaje de tiempo usuario (%usr), tiempo de sistema (%sys), tiempo ocioso (%idle) y tiempo de espera de un evento de E/S (%wio).

2.2.3.2 Monitoreo del desempeño de la memoria

El controlador de memoria se encarga de manipular las peticiones transferidas a memoria desde el CPU y, en la mayoría de los casos, los dispositivos de E/S. La memoria puede ser diseñada primaria o secundaria. La memoria primaria es usada para los programas y datos que están siendo ejecutados actualmente, y la memoria secundaria esta disponible en disco. La contención de memoria crece cuando los requerimientos de memoria de los procesos activos exceden la memoria física disponible del sistema. La memoria física se refiere a todas las localidades de memoria que pueden ser escritas en la computadora.

El sistema operativo usa una porción del disco duro como memoria virtual, e intercambia datos entre el disco duro y la memoria física. La memoria virtual esta asignada para programas de aplicación. Si una computadora necesita ejecutar varios programas simultáneamente, y la memoria que todos esos programas requieren excede la cantidad de memoria física disponible, el sistema operativo asigna memoria virtual para los requerimientos de memoria total de cada programa y administra la memoria física disponible para los requerimientos de memoria actual. Por lo tanto, la cantidad de memoria virtual que es asignada puede ser mucho más grande que la cantidad de memoria física que esta instalada en el sistema.

Todos los procesos UNIX usan memoria real o virtual. La administración de memoria hace posible que el CPU ejecute programas muchos más grandes que la memoria física disponible. Esto también permite que muchos programas más se ejecuten al mismo tiempo.

La cantidad de memoria real y virtual que es requerida para un sistema depende de la carga de trabajo. La cantidad de memoria puede ser incrementada para resolver las necesidades de la carga de trabajo del sistema.

Cuando la memoria física disponible no es suficiente, el sistema operativo comienza a copiar páginas de memoria hacia y desde disco. Esto se conoce como *paginación* y ocurre cuando un proceso activo requiere más memoria de la que es accesible en memoria física, entonces se mueve porciones de este proceso activo a disco para liberar memoria física que puede ser usada por otro proceso.

Otro mecanismo que el sistema operativo emplea para satisfacer los requerimientos de memoria se conoce como *swapping*, el cual ocurre cuando el espacio en memoria es bajo, entonces se busca un proceso que no se ejecutará de manera inmediata para moverlo por completo a disco para liberar su espacio en memoria que puede ser asignado a otro proceso o trabajo.

Los procesos UNIX son rastreados desde su creación hasta su terminación por sus identificadores de procesos. Los procesos son agrupados y colocados en una cola de espera dependiendo de su estado reportado por el comando `ps -l`. Los procesos que están "dormidos" (S), son aquellos que pueden ser movidos de la memoria física para permitir que un proceso que esta en espera sea ejecutado.

Utilizar memoria virtual involucra ambos mecanismos, paginación y *swapping*, los cuales indican que el sistema no puede proporcionar suficiente memoria para los procesos que actualmente se están ejecutando. El constante uso de espacio de memoria virtual degrada el rendimiento del sistema notablemente. Para prevenir la paginación se puede terminar los trabajos que requieren grandes cantidades de memoria. Esto solo se recomienda en situaciones drásticas.

La mejor herramienta para monitorear el uso de memoria, es el comando `vmstat`, el cual reporta estadísticas de memoria virtual, discos y actividad de CPU. Los reportes generados por este comando se pueden usar para determinar si el sistema esta paginando excesivamente. Si el comando se ejecuta sin ninguna opción, el reporte contiene un resumen de la actividad de memoria virtual desde que el sistema se inició. A continuación se muestra su sintaxis:

```
%vmstat [ -cisS ] [ intervalo [ contador ]
```

Opciones :

- c Reporta estadísticas de memoria cache.
- i Reporta el número de interrupciones por dispositivo.
- s Despliega el número total de varios eventos del sistema desde su inicio.
- S Reporta actividad de swapping.

intervalo: Es el número de segundos entre reportes.

contador: Es el número total de reportes a generar.

Ejemplos:

```
% vmstat 5 3
```

procs			memory		page				...		
r	b	w	swap	free	re	mf	pi	po	fr	de	...
0	0	0	4312	10368	3	1115	33	42	46	0	...
0	0	0	531992	8016	0	0	0	0	0	0	...
0	0	0	532000	8016	0	0	0	0	0	0	...

Los campos más importantes de este comando son:

- procs:** Reporta estadísticas de procesos activos.
- r Número de procesos en ejecución.
 - b Número de procesos bloqueados.
 - w Número de procesos en ejecución pero movidos a disco. Si es diferente de 0, el sistema puede estar sufriendo falta de memoria.
- memory:** Reporta el uso de memoria y disponibilidad de datos.
- swap Cantidad de espacio swap actualmente disponible en Kb.
 - free Espacio libre en Kb.

page: Reporta información sobre faltantes de páginas y actividad de paginación.

re Páginas reclamadas en unidades por segundo.

mf Faltantes menores.

pi Número de páginas movidas de disco a memoria (page ins).

po Número de páginas movidas de memoria a disco (page out).

fr Páginas libres (Kb).

dc Anticipación de faltante de memoria en Kb.

```
% vmstat -S 5 3
```

```
procs          memory          page
r  b  w  swap  free  si  so  pi  po  fr  de  ...
0  0  0  7832 10440  0  0  33 42 46  0  ...
0  0  0 527000 12304  0  0  0  0  0  0  ...
0  0  0 527000 12304  0  0  0  0  0  0  ...
```

En el ejemplo anterior se muestra la actividad de swapping (-S) denotados por los campos:

so Procesos movidos de memoria a disco (*swap ins*).

si Procesos movidos de disco a memoria (*swap outs*).

El comando `sar` proporciona datos similares al comando anterior. Su sintaxis es:

```
% sar [-prgw] [intervalo] [contador]
```

donde:

- p Actividad de paginación.
- g Actividad de paginación.
- r Datos de memoria disponible.
- w Actividad de swapping.

Los reportes se verían así:

```
% sar -p 5 3
```

```
SUNOS 5.6 Generic_105181-06 sun4u 09/03/99
14:29:29 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
14:29:34 0.00 0.00 0.00 1.20 1.79 0.00
14:29:39 0.00 0.00 0.00 0.00 0.20 0.00
14:29:44 0.00 0.00 0.00 0.00 0.00 0.00
Average 0.00 0.00 0.00 0.24 0.40 0.00
```

TESIS CON
FALLA DE ORIGEN

```
% sar -g 5 3
```

```
SunOS 5.6 Generic_105181-06 sun4u 09/03/99
14:42:29 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
14:42:34 0.00 0.00 0.00 0.00 0.00
14:42:39 7.17 21.51 22.51 2.79 0.00
14:42:44 0.60 0.60 0.60 0.00 0.00
Average 2.59 7.39 7.72 0.93 0.00
```

Los reportes muestran estadísticas de páginas movidas de disco a memoria (ppgin/s) y viceversa (ppgout/s).

2.2.3.3 Monitoreo del desempeño del subsistema de E/S

El subsistema de E/S (Entrada/Salida) esta compuesto de buffers lógicos y una interfaz de comunicación (CI). El subsistema se encarga de asignar dispositivos de E/S en respuesta a las peticiones de los procesos de los usuarios y tienen dos componentes esenciales:

- Sistema periférico: dispositivos de cinta, disco, impresora, teclado, CD-ROM, etc.
- Sistema de red: soporta interfaces Ethernet, FDDI, ATM y HPPI².

Este subsistema soporta una variedad de buses y dispositivos dependiendo de la máquina en particular y su configuración. Una de las configuraciones más comunes es el SCSI Bus (Small Computer System Interface), el cual es un estándar ANSI para la interconexión de computadoras unas con otras además de discos, floppies, cintas, impresoras, discos ópticos y scanners. El estándar SCSI incluye todos los requerimientos mecánicos, eléctricos y funcionales necesarios para la interconexión de esos dispositivos.

El subsistema de E/S proporciona funciones de lectura y escritura de archivos así como dispositivos para los eventos de E/S. Un evento de E/S es el proceso de transferir datos desde la memoria a disco, desde un dispositivo a memoria o desde un dispositivo a otro.

² High Performance Peripheral Interface.

Este proceso ocurre entre la memoria y los controladores de disco duro del sistema, aunque en algunos sistemas multiusuario, puede ser entre el sistema y terminales conectadas vía red o líneas seriales.

Cuando la velocidad a la cual un dispositivo periférico puede tener acceso y comunicarse con los datos del sistema es relativamente lento, el sistema operativo puede estar gastando la mayoría de su tiempo ocioso (idle) esperando a que un evento de E/S concluya.

Para hacer que la transferencia de datos entre la memoria y el disco sea más eficientemente, el sistema mantiene un *buffer cache* el cual contiene los accesos más recientes a los datos del disco. Esto reduce la cantidad de eventos de E/S de disco que el sistema necesita para trabajar. Cuando un usuario procesa una petición de lectura, el sistema operativo busca el *buffer cache* para la petición de los datos. Si los datos están en el *buffer*, la petición es atendida sin la necesidad de tener acceso al dispositivo físico.

El principal trabajo de un dispositivo de E/S es transferir datos desde la memoria principal o traer datos desde un dispositivo externo y almacenarlos en memoria principal. El CPU se encarga de hacer este trabajo ya que sabe como tener acceso a la memoria. En una lectura el CPU carga los datos desde el dispositivo de E/S dentro de un registro el cual posteriormente almacena los datos en la memoria principal. En una escritura, el CPU pasa los datos al dispositivo de E/S y espera a que este acepte los datos.

Una de las mediciones más importantes de rendimiento asociada con un sistema multiusuario es el tiempo de respuesta el cual esta en función del rendimiento del sistema en general. El rendimiento del subsistema de E/S representa una sensibilidad para cualquier sistema de computo, debido a que el componente más lento del subsistema de E/S es el disco duro. El dispositivo conectado típicamente a un bus de E/S es el controlador de disco duro.

El desempeño de disco es el aspecto más importante del desempeño de E/S. Esto afecta muchos otros aspectos del rendimiento del sistema. El buen desempeño de disco aumenta el desempeño de la memoria virtual, reduce el tiempo requerido para ejecutar trabajos que realizan muchas operaciones de E/S, reduce el tiempo necesario para ejecutar un programa, etc.

Un sistema puede presentar un cuello de botella de E/S, si los dispositivos periféricos (disco duro, cintas, etc.) no pueden transferir los datos tan rápido como el sistema lo requiere. UNIX cuenta con herramientas para analizar el sistema de E/S y determinar donde existen cuellos de botella relacionados al sistema de E/S, el comando `iostat` puede dar información sobre rangos de transferencia para cada controlador de disco y `vmstat` muestra información sobre cuantos procesos están bloqueados esperando a que se termine un evento de E/S.

El comando `iostat` puede ser usado para desplegar la cantidad de datos que se mueven entre el sistema y un disco o un dispositivo de terminal. La columna más interesante es la columna `svc_t`, que es el tiempo promedio en que un disco responde a una petición en milisegundos. Este comando muestra estadísticas altamente confiables de rendimiento. Reporta actividad de disco de E/S así como la utilización de CPU. Por default muestra información de la terminal, disco y CPU. La primer línea de la salida del comando es un sumario desde que el sistema se inició. Su sintaxis es la siguiente:

```
%iostat [ -cdDitx ] [ disk ... ] [ intervalo [ contador ] ]
```

Opciones:

- c Reporta el porcentaje de tiempo que el sistema ha gastado en modo usuario, en modo sistema, esperando a que un evento de E/S termine y modo ocioso.
- d Para cada disco, reporta el número de Kb transferidos por segundo, el número de transferencias por segundo y el tiempo promedio de servicio en milisegundos.
- D Para cada disco, reporta las lecturas por segundo, escrituras por segundo y porcentaje de utilización de disco.

- t Reporta el número de caracteres leídos y escritos en un terminal por segundo.
 - x Para cada disco, reporta estadísticas extendidas de disco.
- disk: Especifica explícitamente los discos a ser reportados; en adicción a cualquier disco explícito, cualquier disco activo también será reportado.

contador: Número de reportes a generar.
intervalo: Número de segundos entre reportes.

Ejemplos:

```
% iostat -xtc 5 2
```

```

                                extended device statistics
disk    r/s    w/s    kr/s    kw/s  wait actv  svc_t   %w   %b   ...
sd1     0.3  1.7    5.0    13.8  0.0  0.0    14.6   0   2   ...
sd2     0.4  0.2    4.1    1.5   0.0  0.0    32.4   0   0   ...
sd3     0.0  0.0    0.3    0.0   0.0  0.0    87.9   0   0   ...

```

```

                                extended device statistics
disk    r/s    w/s    kr/s    kw/s  wait actv  svc_t   %w   %b   ...
sd1     0.0  0.0    0.0    0.0  0.0  0.0    0.0   0   0   ...
sd2     0.0  0.0    0.0    0.0  0.0  0.0    0.0   0   0   ...
sd3     0.0  0.0    0.0    0.0  0.0  0.0    0.0   0   0   ...

```

Los campos tienen los siguientes significados:

- disk Nombre del disco
- r/s Lecturas por segundo
- w/s Escrituras por segundo
- Kr/s Kb leídos por segundo
- Kw/s Kb escritos por segundo
- wait Número promedio de transacciones en espera (tamaño de la cola de espera).
- actv Número promedio de transacciones que están siendo atendidas (eliminadas de la cola de espera pero aun no terminadas).
- svt_c Tiempo promedio de servicio (milisegundos).
- %w Porcentaje de tiempo en que hay transacciones esperando un servicio (la cola de espera no esta vacía)
- %b Porcentaje de tiempo en que el disco esta ocupado (transacciones en progreso).

Este comando también reporta el porcentaje de tiempo que el sistema ha gastado esperando a que un evento de E/S sea terminado y esta representado por la columna wt del reporte del comando.

Con el comando vmstat podemos visualizar la cantidad de procesos bloqueados esperando a que se termine un evento de E/S representado por la columna b:

```
% vmstat 5 3
```

procs			memory			faults			cpu			id
r	b	w	swap	free	...	in	sy	cs	us	sy		
0	0	0	265088	146112	...	126	139	60	0	0	100	
0	0	0	646384	73384	...	152	458	85	0	0	100	
0	0	0	646384	73384	...	149	57	86	0	0	100	

Haciendo uso del comando sar. -u, también ayuda a determinar si el sistema tiene problemas de E/S para lo cual se tiene que observar el valor de la columna %wio. Esta columna despliega el porcentaje de tiempo que cada CPU gasta esperando a que termine un evento de E/S. Si este valor es alto entonces es posible que el subsistema de E/S no pueda continuar con el resto del sistema:

```
% sar -u 5 5
```

```
SunOS 5.6 Generic_105181-06 sun4u 09/30/99
```

	%usr	%sys	%wio	%idle
18:15:49				
18:15:54	0	0	0	100
18:15:59	0	0	0	100
18:16:04	0	0	0	99
18:16:09	0	0	0	100
18:16:14	0	0	0	100
Average	0	0	0	100

Usando sar -d se puede visualizar la actividad de E/S de los dispositivos de disco:

TESIS CON
FALLA DE ORIGEN

```
% sar -d 5 2
```

```
SunOS 5.6 Generic_105181-06 sun4u 09/30/99
```

device	%busy	avque	r+w/s	blks/s	await	avserv
18:33:54						
nfs1	0	0.0	0	0	0.0	0.0
sd9	0	0.0	0	0	0.0	0.0
sd9a	0	0.0	0	0	0.0	0.0
sd9b	0	0.0	0	0	0.0	0.0
sd9c	0	0.0	0	0	0.0	0.0
18:33:59						
nfs1	0	0.0	0	0	0.0	0.0
sd9	1	0.0	1	12	0.0	31.0
sd9a	0	0.0	0	0	0.0	0.0
sd9b	0	0.0	0	2	0.0	18.9
sd9c	0	0.0	0	0	0.0	0.0
Average						
nfs1	0	0.0	0	0	0.0	0.0
sd9	1	0.0	1	11	0.0	32.3
sd9a	0	0.0	0	2	0.0	20.0
sd9b	0	0.0	0	2	0.0	32.5
sd9c	0	0.0	0	0	0.0	0.0

Los campos tienen los siguientes significados:

device: Nombre del disco.

%busy: Porcentaje del tiempo ocupado por dispositivo atendiendo una petición de transferencia.

avque: Número promedio de peticiones durante ese periodo de tiempo.

r+w/s: Número de lecturas y escrituras transferidas a dispositivo por segundo.

blks/s: Número de bloques (512 bytes) transferidos a dispositivo por segundo.

await: Tiempo promedio en milisegundos en que las peticiones de transferencia permanecen en la cola de espera.

avserv: Tiempo promedio en milisegundos para que una petición de transferencia sea terminada por el dispositivo.

2.3 DESEMPEÑO DE LA RED

Durante los últimos años, los usuarios han estado transmitiendo sistemas de archivos sobre redes distribuidas, teniendo un fácil acceso a sistemas remotos y otras facilidades basadas en red.

Mientras que las redes han llegado a ser lo más importante para los sistemas de computo, también se han convertido en parte esencial del rendimiento del sistema en su totalidad. La red puede llegar a ser un factor limitante para el desempeño de E/S, si un grupo de sistemas confía demasiado en una red, esta puede (en el peor de los casos) convertirse en el factor dominante en el desempeño total del sistema. Desafortunadamente, entender el rendimiento de la red es extremadamente difícil. Se debe observar las interacciones entre los sistemas de computadoras, no solo de unas simples computadoras. Los problemas con otros sistemas en la red pueden causar un bajo rendimiento en el sistema. Las herramientas que miden diferentes aspectos del rendimiento de la red entregan una gran cantidad de datos, aunque a veces no son muy útiles.

2.3.1 Monitoreo y desempeño de la red

Desde el punto de vista de un usuario, hay una manera fácil de saber cuando la red esta afectando la velocidad del sistema. Las operaciones que hacen uso de la red se notan lentas, mientras que las operaciones que no hacen uso de ella ocurren a una velocidad normal. Por ejemplo, si una sesión `rlogin` despliega caracteres varios segundos después de que ellos son tecleados, mientras una sesión `login` despliega caracteres normalmente, el rendimiento de la red puede ser un retardo.

La mayoría de las veces, solo una pequeña fracción del ancho de banda de la red esta en uso. Esto puede ser razonable para pedirles a los usuarios que toleren tales retardos. Diez segundos pueden ser una eternidad cuando un editor esta salvando un archivo pero ciertamente no es una pérdida crucial de tiempo. En tales situaciones, se puede tolerar el rendimiento del sistema en vez de reestructurar la red para optimizar su desempeño. Sin embargo, si se llevan minutos para salvar un archivo (en vez de segundos) o si salvar un archivo produce mensajes como "el servidor NFS no responde" ("NFS server not responding still trying"), se tienen problemas de red más significativos (posiblemente una red dañada o un servidor de archivo defectuoso) y se tendrá que tomar alguna acción.

TESIS CON
FALLA DE ORIGEN

El desempeño de la red no sólo depende de los sistemas individuales que conforman la red, sino también de la interrelación entre ellos. Cualquier miembro de la red o componentes físicos de la red ya sea cables o conectores pueden tener un impacto en el desempeño de la red.

Para determinar si el sistema esta experimentando problemas en el desempeño de la red, es necesario recolectar datos de la red para determinar el rendimiento total que el sistema es capaz de sostener con el software y hardware instalado. Las herramientas de monitoreo de la red envían paquetes hacia dentro y fuera de una máquina para medir los bits por segundo (bps) o millones de bps (Mbps), que viajan de un extremo a otro, es decir, miden la cantidad de datos que viajan hacia dentro o fuera del sistema.

Una vez que se tienen estas estadísticas, la carga de trabajo puede ser analizada. Las herramientas interactivas de monitoreo rastrean el número de paquetes de la red que llegan y salen del sistema en una instancia de tiempo actual, por lo tanto se mide la carga de trabajo de la red.

A continuación se describe las herramientas que despliegan estadísticas sobre la actividad y el rendimiento de la red, las cuales pueden usarse para rastrear e investigar la causa de que una red este lenta.

Dentro de las herramientas interactivas, se encuentra el comando `netstat`, que es usado para monitorear la actividad de la red de un sistema y reportar información variada de la misma. Este comando proporciona algunos datos básicos sobre cuánta actividad y qué tipo de actividad se está realizando en la red. Sin argumentos, lista todas las conexiones con el host³ local. A continuación se muestra la salida:

³ Nodo que pertenece a una red, este puede ser servidor o cliente.

```
% netstat
```

TCP Local Address	Remote Address	...	State
e5000.dcaa.unam.mx.33205	delfin.dcaa.unam.mx.6000	...	0 ESTABLISHED
e5000.dcaa.unam.mx.shell	simba.dgsca.unam.mx.1023	...	0 ESTABLISHED
e5000.dcaa.unam.mx.telnet	sauce.dcaa.unam.mx.1028	...	0 ESTABLISHED

En este host *e5000* hay actualmente conexiones a *delfin*, *simba* y *sauce*. El sufijo *.telnet* indica una conexión hecha con el comando *telnet*; el sufijo *.shell* indica una conexión atendiendo un simple comando. Otra forma usual de este comando es proporcionando un intervalo de tiempo, lo cual produce una salida completamente diferente:

```
% netstat -i 5
```

input hme2	output			input (Total)			output		
packets errs	packets	errs	colls	packets	errs	packets	errs	colls	
7265222 0	6888497	0	489072	7266023	0	6889298	0	489072	
1089 0	603	0	0	1089	0	603	0	0	
1071 0	596	0	0	1071	0	596	0	0	
1088 0	588	0	4	1088	0	588	0	4	
1099 0	604	0	0	1099	0	604	0	0	

Con la opción *-i* despliega una lista de las interfaces disponibles. Enseguida, evalúa la cantidad de datos que están siendo enviados y recibidos.

Este comando desplegará estadísticas de la red cada 5 segundos. La salida se divide en dos partes: incluye dos conjuntos de estadísticas de entrada y salida. Las primeras 5 columnas muestran datos para la interfaz de red primaria y las otras columnas muestran valores totales para todas las interfaces de red del sistema. En este sistema, como en muchos otros, sólo hay una interfaz, y por lo tanto ambas partes son iguales.

Las columnas "input" muestran datos que entran al tráfico de la red, y las columnas "output" muestran datos que salen del tráfico. Las columnas "errs" muestran el número de errores que ocurrieron durante la transferencia del número indicado de paquetes de red (un paquete es la unidad de transferencia de datos de red). Esos números deben ser bajos: menos del 1% del número de paquetes. Grandes números indican serios problemas de red.

Las columnas "colls" listan el número de colisiones. Una colisión ocurre cuando dos nodos de la red tratan de enviar un paquete exactamente al mismo tiempo. Cuando esto pasa, cada nodo espera una cantidad de tiempo aleatorio antes de volver a transmitir un paquete, este método elimina virtualmente colisiones repetidas por el mismo host con los mismos paquetes. El número de colisiones es una medida de cuánto tráfico de red hay en el sistema.

La herramienta más simple para monitorear la red es el comando `ping`, que es usado para observar la respuesta de los sistemas de la red. Este comando envía un paquete a un host específico en la red solicitando una respuesta. Si recibe la respuesta correcta, el comando imprime el mensaje "*host is alive*". Si la respuesta no llega, imprime el mensaje "*host is unreachable*". En este caso, no hay una ruta válida entre el sistema y el host con el que se está tratando de establecer una comunicación y puede suceder porque:

- El sistema no está conectado a la red
- El sistema tiene un coprocesador de red que puede estar fallando.
- El host con el que se trata de comunicarse se ha caído. El comando debe funcionar una vez que el host sea reinicializado.
- Alguna porción de la red entre el sistema y el host está rota. El problema puede ser desde una pieza de cable hasta un gateway dañado.
- Las tablas de ruteo del sistema están incorrectas o incompletas.

Sintaxis:

```
% ping host tiempo
```

Aquí el argumento tiempo es opcional. Este especifica la cantidad de tiempo que el comando espera antes de decidir que no hay comunicación con el host especificado. Por default `ping` se espera 20 segundos. A continuación se muestra un ejemplo:

```
% ping e5000  
e5000 is alive.
```

Con la opción `-s`, `ping` envía un paquete por segundo a un host e imprime la respuesta y tiempo para cada uno:

```
% ping -s sunsite
PING sunsite.dcaa.unam.mx: 56 data bytes
64 bytes from sunsite.dcaa.unam.mx(132.248.63.194): icmp_seq=0. time=2. ms
64 bytes from sunsite.dcaa.unam.mx(132.248.63.194): icmp_seq=1. time=0. ms
64 bytes from sunsite.dcaa.unam.mx(132.248.63.194): icmp_seq=2. time=0. ms
64 bytes from sunsite.dcaa.unam.mx(132.248.63.194): icmp_seq=3. time=0. ms
64 bytes from sunsite.dcaa.unam.mx(132.248.63.194): icmp_seq=4. time=0. ms
----sunsite.dcaa.unam.mx PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/2
```

La utilidad `nfsstat` puede ser usada para identificar problemas relacionadas con el Sistema de Archivos de Red (NFS) o con interfaces del kernel. Este comando muestra un resumen de estadísticas de servidor y cliente. Su sintaxis es la siguiente:

```
% nfsstat [ -cmnrS ]
```

Opciones:

- c: Muestra estadísticas del cliente.
- m: Muestra estadísticas para cada NFS montado en el sistema.
- n: Despliega información sobre NFS tanto para el cliente como para el servidor.
- r: Despliega información sobre RPC.
- s: Muestra estadísticas del servidor.

El siguiente ejemplo despliega estadísticas para el cliente:

```
% nfsstat -c
Client rpc:
calls      badcalls  retrans   badxid   wait     newcred
6888      132      10        51       0        0
```

El comando despliega los siguientes campos:

calls:	Número total de llamadas enviadas.
badcalls:	Número total de llamadas rechazadas por RPC.
retrans:	Número total de retransmisiones.
badxid:	Número de veces que un acuse fue recibido por una petición NFS.
wait:	Número de veces que una llamada a tenido que esperar porque no estaba disponible el cliente.
newcred:	Número de veces que la información ha sido refrescada.

El servicio de archivos distribuidos NFS utiliza la facilidad de llamada de procedimiento remoto (RPC) la cual traslada comandos locales a peticiones del host remoto. Si el servidor de archivos no puede responder a la petición de un cliente, el cliente retransmite la petición. Cada retransmisión impone una sobrecarga al sistema e incrementa el tráfico de la red. Muchas retransmisiones pueden causar problemas en el desempeño de la red. Si el rango de retransmisiones es alto se puede hacer lo siguiente:

- Buscar servidores con mucha carga de trabajo que se tomen mucho tiempo para terminar una petición.
- Buscar una interfaz Ethernet que este modificando los paquetes.
- Buscar una congestión de red la cual transmite los paquetes lentamente.

2.3.2 Etiqueta de red básica

Unas simples reglas de etiqueta ayudaran a los usuarios a obtener mucho más de lo que esperan. Si se entiende como funciona una red, estas reglas serán de sentido común. Observar estas reglas minimizará el tráfico de la red y, en muchos casos, resolverán problemas significantes de rendimiento sin requerir algún cambio de configuración.

El NFS tiene permitido un acceso virtualmente transparente a los archivos vía Ethernet. Esto es una gran ventaja, pero tiene un problema. Es fácil de descuidar u olvidar en donde están físicamente localizados los archivos. Esto rápidamente permite problemas de rendimiento en la red, por dos razones:

- La dirección Ethernet esta compartida por cada usuario en cada sistema de la red. Por lo tanto, es un recurso relativamente limitado con muchos usuarios. Esta situación puede ser arreglada agregando subredes no importando que tan compleja es la topología de red, una red básicamente consiste de muchos sistemas comunicados mediante una simple pieza de alambre. Si un usuario tiene acceso a un archivo muy grande por medio de la red, puede estar ocasionando retrasos para todos los usuarios: otros usuarios en su sistema, otros usuarios en otros sistemas, etc.
- Tener acceso a un archivo vía Ethernet nunca puede ser más rápido que hacer un login remoto y tener acceso localmente al mismo archivo. Se tiene que escribir o leer en el disco. No importa que rápida es la red, esto solo puede agregar más carga.

Los usuarios deben estar enterados de los intercambios de datos que realizan cuando tienen acceso a archivos remotos. Si un programa lee o escribe una gran cantidad de datos, es mejor abrir una sesión en el sistema donde los datos son locales que transferirlos por medio de la red usando NFS, es mejor tanto para los usuarios quienes están ejecutando el programa como para todos los usuarios de la red.

El punto no es que los usuarios deben evitar usar NFS, sino que la red es una herramienta y, como cualquier otra, tiene usos apropiados e inapropiados. Es una excelente herramienta para mover archivos entre sistemas, para navegar, para edición remota, y para compartir archivos. Si varias estaciones de trabajo pueden compartir el sistema de archivos /usr se puede liberar mucho espacio en disco. Sin embargo, NFS no es una herramienta apropiada para tener acceso a los datos con alta velocidad o para tener acceso a cintas remotas.

Si un programa muy largo presenta un cuello de botella de E/S, no hay una buena razón para hacer que el programa sea más lento incurriendo en cargas adicionales de la red. La edición, compilación y otras tareas que no presentan problemas de E/S son ideales para la red. Si se requiere transferir una gran cantidad de datos entre diferentes sistemas de cómputo, Ethernet puede no ser una medida apropiada de usar; el cable básico de Ethernet esta limitado a 10 MB por segundo.

Existen otras medias disponibles que ofrecen significativamente altos rangos de transferencia: Fast Ethernet y HPPI que trabajan a 100 MB p/s y FDDI.

El administrador del sistema debe asegurarse de que cualquier estación de trabajo tenga suficiente memoria para correr eficientemente. Si no cuentan con suficiente memoria local para minimizar paginación y swapping, pueden fácilmente consumir mucho ancho de banda de la red, aún si los usuarios obedecen los puntos aquí discutidos. Los requerimientos de memoria de cualquier estación de trabajo dependen principalmente del fabricante y de la versión del sistema operativo.

En redes de sistemas de cómputo, la red u otras computadoras conectadas a la red pueden afectar el rendimiento del sistema. Algunas sugerencias para detectar problemas de la red:

- Se puede sospechar que hay problemas con capacidad de la red o con la integridad de los datos si los usuarios notan lento el rendimiento cuando están usando el comando `rlogin` o cuando tienen acceso a archivos vía NFS.
- Ejecutar el comando `netstat -i`. Si el número de colisiones es grande, se puede pensar que existe una sobrecarga en la red. Si el número de errores en entrada o salida es grande, puede existir problemas de hardware. Un número grande de errores de entrada indica problemas en alguna parte de la red. Un número grande de errores de salida sugiere problemas con el sistema y su interfaz de red.

- Si las colisiones y el hardware de la red no son el problema, averiguar cuál sistema parece ser lento. Usar el comando `ping` para enviar una gran cantidad de paquetes al sistema lento. Si el número de paquetes perdidos es grande, muy probablemente el sistema remoto no puede responder lo suficientemente rápido a la entrada de datos.
- Observar si hay problemas de CPU, memoria o E/S de disco en el sistema remoto. Si no es así, quizás el sistema no es capaz de soportar cargas de trabajo pesadas de la red. Intentar reorganizar la red de modo que el sistema no sea un servidor de archivos.
- Ejecutar `nfsstat` y observar los datos del cliente de RPC.
 - ✓ Si el campo `retrans` es mayor al 5% del campo `calls`, la red o un servidor de NFS está sobrecargado.
 - ✓ Si el campo `timeout` es alto, al menos un servidor de NFS está sobrecargado, la red puede estar fallando, o uno o más servidores pueden estar abajo.
 - ✓ Si el campo `badxid` es igual al campo `timeout` (misma magnitud), al menos un servidor de NFS está sobrecargado.
 - ✓ Si los campos `timeout` y `retrans` son altos pero el campo `badxid` es bajo, alguna parte de la red entre el cliente de NFS y el servidor está sobrecargado y perdiendo paquetes.
- Si el sistema tiene una implementación de red basada en STREAMS, ejecutar `netstat -m` y verificar que existan suficientes buffers STREAMS.

Algunos tips para reducir la carga de la red son:

- Tratar de prevenir a los usuarios de ejecutar programas intensivos de E/S a través de la red.
- Reorganizar las computadoras y discos de la red de manera que la mayoría de usuarios como sea posible puedan hacer tanto trabajo como sea posible en un sistema local.
- Minimizar el número de sitios de trabajos sin disco duro (*diskless workstations*). Si es posible eliminarlas.
- Usar sistemas con un buen rendimiento de red como servidores de archivos.
- Si se tiene un número corto de STREAMS reconfigurar el kernel con más buffers.

Si hay problemas con la integridad de los datos, la única solución es encontrar la pieza de hardware dañada y reemplazarla. Un analizador de red es una pieza indispensable de equipo para esta tarea.

Para un desempeño óptimo de la red, esta debe cumplir con las siguientes condiciones:

- Debe ser capaz de transferir datos correctamente (integridad de los datos).
- Debe proporcionar suficiente ancho de banda para satisfacer las necesidades de los usuarios de la red. Si la red no tiene suficiente ancho de banda, la cantidad de tiempo requerido para transferir datos entre dos puntos cualquiera será excesivamente grande.
- Cada sistema de la red debe ser lo suficientemente rápido para manipular el tráfico de la red.

3. SINTONIZACIÓN DEL SISTEMA

Este capítulo contiene información sobre la configuración de un sistema UNIX para obtener un alto rendimiento. Describe como sintonizar el sistema para mejorar el desempeño mediante el monitoreo del sistema, identificando problemas de rendimiento, configurando y optimizando los subsistemas de almacenamiento, CPU, memoria y red.

Actualmente los negocios requieren un ambiente de computó que sea confiable y capaz de soportar la carga de trabajo que se genere en un momento dado. Los usuarios y las aplicaciones generan diferentes demandas en un sistema, y ambos requieren de un desempeño consistente con un mínimo de tiempo de respuesta. Un sistema también debe ser capaz de absorber un incremento en la carga de trabajo sin alterar su desempeño.

3.1 SISTEMAS DE ALTO RENDIMIENTO

3.1.1 Terminología y conceptos

El rendimiento de un sistema depende de una eficiente utilización de los recursos del sistema que están disponibles para los usuarios y las aplicaciones, estos incluyen componentes de hardware y software. Un sistema debe trabajar bien bajo una carga de trabajo ejercida en el sistema tanto por las aplicaciones como de los usuarios.

La carga de trabajo del sistema cambia en un cierto plazo. Se puede agregar usuarios o ejecutar aplicaciones adicionales. Quizá sea necesario configurar de nuevo el sistema para poder manejar un incremento de carga de trabajo. La *escalabilidad* se refiere a la capacidad de un sistema para utilizar recursos adicionales con un incremento fiable en funcionamiento, o la capacidad de absorber un aumento en la carga de trabajo sin una degradación significativa del rendimiento.

Un problema de desempeño en un área específica de la configuración es llamado *cuello de botella*. Los cuellos de botella potenciales incluyen el subsistema de memoria virtual y buses de E/S. Un cuello de botella puede ocurrir si la carga de trabajo exige más de un recurso que va más allá de la *capacidad* del sistema, que es el rendimiento de procesamiento teórico máximo de un recurso.

El desempeño frecuentemente se describe en términos de dos valores. El *ancho de banda* que es el valor en el cual un subsistema o componente de entrada/salida puede transferir bytes de datos. El ancho de banda a menudo es llamado *valor de transferencia*. El ancho de banda es especialmente importante para las aplicaciones que realizan grandes transferencias de datos secuenciales.

**TESIS CON
FALLA DE ORIGEN**

El *rendimiento de procesamiento (throughput)* es el valor en el cual un subsistema o componente puede realizar operaciones de entrada/salida. El rendimiento de procesamiento es especialmente importante para las aplicaciones que realizan muchas operaciones pequeñas de entrada/salida.

El desempeño también es medido en términos de *latencia*, que es la cantidad de tiempo para realizar o llevar a cabo una operación específica. La latencia es a menudo llamada *retardo (delay)*. El alto desempeño de un sistema requiere un tiempo bajo de latencia, es decir, poco *tiempo de espera*. El tiempo de espera de la entrada/salida se mide en milisegundos; el tiempo de espera de la memoria se mide en nanosegundos. El tiempo de espera de la memoria depende de la configuración del banco de memoria y de los requerimientos de memoria del sistema.

La *alta disponibilidad* es la capacidad de un recurso para soportar un incidente de hardware o software. Los recursos se pueden hacer altamente disponibles usando una cierta forma de duplicación del recurso o *redundancia*. Por ejemplo, se puede hacer que un disco de datos sea altamente disponible haciendo un *mirror* de disco, es decir, un reflejo de ese disco; replicando los datos en un disco diferente. Si el disco original falla, la copia todavía está disponible para los usuarios y las aplicaciones. Si se utiliza un *RAID* (Redundant Array of Independent Disks), los datos redundantes son almacenados para que sean utilizados para regenerar datos si ocurre una falla del disco.

Además, se puede hacer que la red sea altamente disponible usando conexiones de red redundantes. Si una conexión no está disponible, se usa otra conexión para el acceso de red. La disponibilidad de la red depende de las aplicaciones, la configuración de red y del protocolo de red.

3.1.2 Configuración del sistema

La configuración de un sistema consiste de una combinación de hardware y software. Por ejemplo CPUs, tarjetas de memoria, discos y el sistema operativo son parte de una configuración. Para configurar un sistema, se necesita instalar un nuevo hardware o software, o modificar una configuración ya existente. Por ejemplo, la configuración del subsistema de almacenamiento puede incluir instalar más discos.

Para planear la configuración que satisfaga las necesidades de desempeño, se debe identificar qué recursos tendrán el más grande impacto en el desempeño. Por ejemplo, si las aplicaciones mantienen al CPU con una actividad muy intensa, se puede considerar un sistema con múltiples CPUs y suficiente ancho de banda.

Un sistema *multiprocesador* permite dos o más procesadores para compartir la memoria física común. Un ejemplo de sistema multiproceso es un *sistema de multiprocesamiento simétrico (SMP)*, en el cual los CPUs ejecutan la misma versión de sistema operativo, tienen acceso a la memoria común y ejecutan instrucciones simultáneamente.

Si las aplicaciones requieren mucha memoria, se debe configurar suficiente memoria para el sistema; si ejecutan un gran número de operaciones de E/S, se debe configurar el subsistema de almacenamiento para prevenir cuellos de botella en disco. Si el sistema es un servidor de Internet, debe asegurarse que pueda manejar muchas peticiones de red. Si se requiere una alta disponibilidad además de un alto desempeño, se requieren que el sistema sea escalable y se debe determinar el impacto que tendrá en el desempeño del sistema una configuración con esta característica. Un sistema escalable permite agregar hardware para mejorar el desempeño o para absorber un incremento en la carga de trabajo.

3.1.3 Entendiendo el alto desempeño

Después de que un sistema es configurado, se puede hacer ciertos cambios de tal forma que el sistema proporcione un alto rendimiento o desempeño asegurando que los usuarios finales nunca experimentarán un servicio inestable. Por supuesto, esto es en teoría pues en la mayoría de las circunstancias no es posible garantizar un servicio adecuado todo el tiempo bajo cualquier situación. Sin embargo, se puede conseguir tener un nivel aceptable de servicio constante para la mayoría de los usuarios.

Un sistema debe tener un nivel confiable de desempeño para satisfacer las necesidades de los usuarios y las aplicaciones. La configuración del sistema es crucial para que pueda responder rápidamente a las demandas de una carga de trabajo normal y mantener un nivel adecuado de desempeño si la carga de trabajo incrementa.

Se debe entender las características de la carga de trabajo para determinar el nivel de desempeño que se requiere y la configuración necesaria para cubrir las necesidades de desempeño. Aunque algunos ambientes requieren el más alto desempeño posible, este nivel puede ser no necesario o muy costoso. El desempeño del sistema depende de la interacción entre la configuración del hardware y del software además de la carga de trabajo. Un sistema que trabaja bien debe utilizar los recursos de CPU, memoria y E/S eficientemente. Si un recurso alcanza su capacidad, se convierte en un cuello de botella y puede degradar el rendimiento. Los cuellos de botella se relacionan entre sí de manera frecuente; por ejemplo, insuficiente memoria puede causar excesiva paginación y swapping, y esto a su vez ocasionar un cuello de botella en el subsistema de disco de E/S.

Una vez que se planea e implementa una configuración, se puede mejorar el desempeño sintonizando el sistema, sin embargo, la sintonización podría proporcionar un desempeño marginado, por lo que se debe asegurar que la configuración es la apropiada para la carga de trabajo de un sistema en particular.

Sintonizar es el proceso de observar la operación de un sistema de cómputo por completo y ajustar varios componentes del sistema basándose en esas observaciones logrando así que el sistema sea más eficiente y pueda brindar así un alto desempeño.

Si al sintonizar un sistema no mejora su rendimiento, entonces tiene que ser configurado nuevamente, lo cual involucra agregar CPUs o memoria, cambiar la configuración de almacenamiento o modificar el software de aplicación.

3.1.4 Configuración y sintonización de un sistema

Implementar y mantener un sistema de alto rendimiento y de alta disponibilidad requiere de ciertos pasos:

- Configurar el sistema.

Para configurar (o reconfigurar) un sistema, se debe determinar los requerimientos de su ambiente y seleccionar una configuración para satisfacer las necesidades. Una vez hecho esto, se debe implementar el hardware, sistema operativo, productos y aplicaciones.

- Ejecutar cualquier tarea de sintonización inicialmente recomendada.

Para algunas configuraciones, se tiene que realizar algunas tareas de optimización inmediatamente después de configurar el sistema. Por ejemplo, si el sistema se utiliza como servidor de Internet, seguir las recomendaciones para modificar los valores por defecto de los parámetros y atributos del sistema.

- Monitorear el desempeño del sistema.

Monitorear el desempeño sistema como se describe en el capítulo 2. Si es aceptable, continuar revisando constantemente para evitar que el desempeño disminuya si los recursos alcanzan su máxima capacidad o si hay un cambio significativo en el ambiente. Si el desempeño no es aceptable, se debe determinar la fuente del problema.

- Determinar si hay una solución de sintonización que elimine el problema de desempeño.

Si no hay una solución o si se han agotado todas las soluciones posibles, reconfigurar el sistema para eliminar el problema de desempeño.

- Eliminar el problema de desempeño.

Para eliminar el problema, primero hay que considerar soluciones no costosas, tales como correr aplicaciones por las noches o restringir el acceso a disco. Posteriormente intentar soluciones más complejas y caras, tales como reconstruir el kernel o agregar más hardware.

Si las aplicaciones y el CPU no son el problema, la optimización del subsistema de memoria virtual proporciona la mejor ventaja de desempeño y debe ser el área de enfoque. Si la optimización de la memoria no elimina el problema, sintonizar el subsistema de E/S. La sintonización usualmente requiere modificar los atributos del kernel. Sin embargo, se puede mejorar desempeño del sistema realizando algunas tareas administrativas del sistema.

- Monitorear el desempeño del sistema.

Después de sintonizar el sistema, se debe monitorear cuidadosamente el sistema para asegurarse de que el problema de desempeño ha sido eliminado.

Si no se puede reducir o eliminar un problema de desempeño sintonizando el sistema, se debe reconfigurar el sistema nuevamente. La figura 3.1 describe el proceso de configuración y sintonización de un sistema:

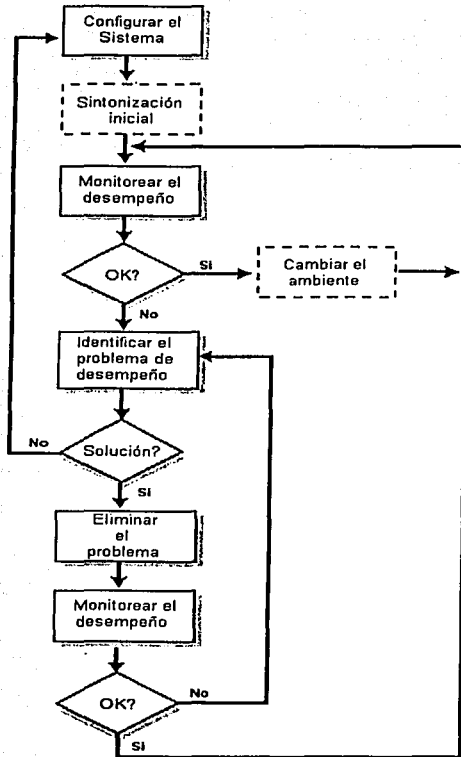


Fig. 3.1 Proceso de configuración y sintonización.

TESIS CON
FALLA DE ORIGEN

3.2 DIAGNOSTICANDO PROBLEMAS DE DESEMPEÑO

Para obtener el máximo rendimiento de un sistema, se debe eliminar cualquier cuello de botella que tenga que ver con el desempeño. Diagnosticar problemas de desempeño involucra identificar el problema (por ejemplo, excesiva paginación), y determinar la fuente del problema (por ejemplo, insuficiente memoria o valores incorrectos para los atributos del subsistema de memoria virtual).

3.2.1 Revisando el rendimiento del sistema

Aunque los problemas de rendimiento son evidentes aparentemente (por ejemplo, las aplicaciones se ejecutan lentamente), otros problemas pueden no ser tan obvios para los usuarios o administradores. Esto puede ser un obstáculo para identificar la fuente del problema. Hay varias maneras de determinar si un sistema tiene un problema de desempeño o si se puede mejorar el desempeño de un sistema. Algunas indicaciones de problemas de desempeño son:

- Tiempo de respuesta del sistema lento. Si los usuarios se quejan del tiempo de respuesta del sistema, esto puede indicar que los procesos están siendo movidos de memoria a disco por falta de memoria.
- Tiempo de ejecución de aplicaciones lento. Si el tiempo de terminación de una aplicación es inadecuado, esto puede indicar que el CPU o la memoria son inadecuados, que existe un cuello de botella de E/S o que es una aplicación pobremente diseñada.
- Uso inadecuado de los discos. La actividad excesiva de solo unos cuantos discos puede indicar una distribución desequilibrada de disco.
- Excesiva paginación e intercambio de procesos. Un valor alto de paginación y swapping puede indicar una memoria inadecuada para la carga de trabajo.
- Conexiones de red lentas o incompletas. Las conexiones de red que se cortan prematuramente pueden ser causa de problemas de desempeño de la red.

- Mensajes de eventos indicando problemas en el sistema. La información de los archivos ASCII o archivos binarios de registro del sistema pueden ser una alerta de que existen problemas de desempeño.
- Información de las herramientas de análisis de programas y del kernel. La salida de las herramientas que se utilizan para analizar código puede indicar las áreas del código que están degradando el desempeño o que están haciendo un uso excesivo de los recursos.

3.2.2 Obteniendo información de desempeño

Para determinar como está trabajando el sistema y poder diagnosticar problemas de desempeño, es necesario obtener información del sistema. Para hacer esto, se necesitan las estadísticas del sistema y monitorear los recursos.

Es conveniente recolectar estadísticas de desempeño bajo diferentes condiciones. Por ejemplo, recolectar información cuando el sistema esta trabajando y cuando tiene un desempeño bajo. Esto permitirá comparar diferentes conjuntos de datos.

Después de configurar el ambiente, inmediatamente comenzar a recolectar la información de desempeño ejecutando las siguientes tareas:

- Configurar el registro de eventos. El sistema operativo UNIX usa la facilidad de registros de eventos para diagnosticar problemas de desempeño. Dicha facilidad usa la función `syslog` la cuál envía un mensaje al demonio `syslogd`¹ para que recolecte los mensajes registrados del kernel, comandos, utilidades y programas de aplicación. Posteriormente el demonio escribe los mensajes en un archivo de configuración local especificado en archivo `/etc/syslog.conf`².

¹ Programa encargado de guardar informes sobre el funcionamiento de una máquina.

² Archivo de configuración que define las reglas a seguir para gestionar el almacenamiento de mensajes del sistema.

- Activar las estadísticas del sistema y cuotas de disco para rastrear la utilización de recursos por cada usuario. Las estadísticas del sistema permiten obtener información de cómo los usuarios emplean los recursos. Se puede obtener información de la cantidad de tiempo de CPU, el número de procesos, el número de operaciones de E/S, uso de memoria y disco.
- Monitorear el desempeño continuamente. Para hacer esto, se puede hacer uso del visualizador de desempeño, que muestra una gráfica de desempeño de todos los componentes significantes del sistema. Además permite ver el impacto que las aplicaciones tienen en todo el sistema.
- Obtener estadísticas iniciales de desempeño. Se puede hacer uso de los comandos descritos en el capítulo 2 para recolectar información e identificar problemas de desempeño. Se debe obtener estadísticas en diferentes condiciones. Por ejemplo:
 - ✓ Inmediatamente después de instalar el sistema y antes de correr cualquier aplicación para obtener información del desempeño del sistema.
 - ✓ Cuando el sistema está trabajando bien bajo una carga de trabajo normal.
 - ✓ Cuando el sistema tiene un bajo desempeño bajo una carga de trabajo normal.
 - ✓ Después de que el sistema se haya optimizado o reconfigurado.

3.2.3 Usando herramientas adicionales de monitoreo

El rápido incremento de los sistemas de cómputo de misión crítica demandan al mismo tiempo un alto rendimiento. Esto a originado que proliferen soluciones para la administración del desempeño ofreciendo características adicionales para la planeación de capacidad del sistema a largo plazo, la mayoría de ellas tiene un alto costo y otras son de dominio público. Estas herramientas también nos permiten obtener estadísticas de desempeño en tiempo real y alguna otra información del sistema, generalmente presentan un mejor cuadro comprensivo del sistema con diferentes cargas de trabajo y esto ayuda a diagnosticar los problemas de desempeño e incluso anticiparlos.

Como una alternativa a las herramientas comerciales, algunos desarrolladores han programado algunas aplicaciones de software libre o gratuito para UNIX y Solaris particularmente. La mayoría de esas herramientas se concentran en aspectos de administración del desempeño.

Un ejemplo de tales herramientas es la utilidad `top`, que es un programa que despliega una pantalla completa del status del sistema y la mayoría de los procesos activos. Aquí se muestra el despliegue desde un sistema UNIX:

```
% top
load averages: 0.00, 0.05, 0.21      19:04:53
61 processes: 59 sleeping, 1 stopped, 1 on cpu
CPU: 98.4%idle,0.0%user,1.6%kernel,0.0%iowait,0.0%swap
Memory: 64M real, 4044K free, 42M swap in use, 159M swap free

PID  USER  THR  PRI  NI  SIZE  RES  STATE  TIME  CPU  COMMAND
1040  cristy  1    39   0   1444K 912K  cpu    0:00  0.63% top
309   root   12   38   0   2072K 536K  sleep  13:38  0.04% mibiisa
3385  root   1    58   0   1632K 848K  sleep  3:15  0.01% rpc
2030  root   1    59   0    11M 3556K  sleep  27:43  0.00% Xsun
7910  admn   1    480  0    23M 10M    sleep  13:4  10.00% netscape
237   root   9    53   0   2352K 1140K  sleep  0:55  0.00% nscd
142   root   1    58   0   4436K 536K  sleep  0:39  0.00% skipd
266   root   1    58   0   2192K 416K  sleep  0:09  0.00% sendmail
219   root  11    58   0   3108K 1148K  sleep  0:06  0.00% syslogd
265   root   1    58   0    936K 264K  sleep  0:06  0.00% utmpd
1     root   1    58   0    708K 108K  sleep  0:03  0.00% init
2811  root   4    13   0    11M 1360K  sleep  0:01  0.00% pine
.
.
.
```

Proctool es otra herramienta gratuita desarrollada por Walter Nielsen y Morgan Herrington de Sun Microsystems. Proctool (Fig. 3.2) monitorea y controla los procesos de un sistema, proporcionando una interfaz gráfica para el comando `ps` y proporciona más flexibilidad que la utilidad `top`. Esta herramienta extrae una gran cantidad de información de las estructuras de datos mantenidas en `/proc`³ para cada proceso y puede esquematizar una gran variedad de datos gráficamente por proceso o por sistema.

³ Sistema de archivos que permiten un fácil acceso a la información sobre los procesos.

El kit de herramientas llamado SE (*Symbolic Engine*), consiste de un lenguaje de programación, un conjunto de librerías y de scripts que imitan la funcionalidad de los comandos, vmstat, iostat, netstat, netstatc y otras utilerías. Su elemento fundamental es el lenguaje de programación SymbEL (*Symbol Engine Language*) que proporciona las bases para desarrollar herramientas personalizadas para la administración del desempeño. Se diseña en respuesta a las necesidades de simplificar el acceso a los datos contenidos dentro del kernel. SymbEL es similar a C visualmente y contiene muchas estructuras similares.

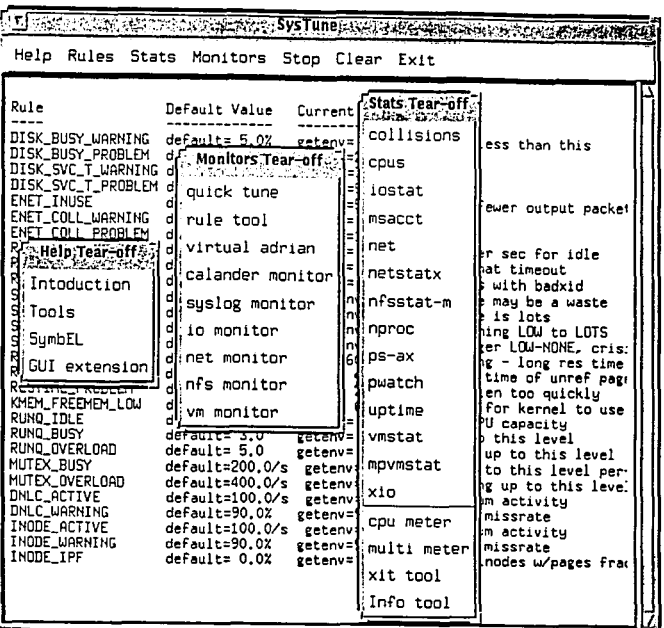


Fig. 3.3 Ventana principal de SE.

TESIS CON
FALLA DE ORIGEN

3.3 MODIFICACION DEL KERNEL

Las **variables del kernel** o *parámetros regulables*, determinan el rendimiento del sistema operativo UNIX y los subsistemas, algunas de ellas son usadas solo para monitorear el estado actual del sistema. Cuando se instala el sistema operativo o se agregan subsistemas adicionales, las variables del kernel tienen un valor por default. Modificar los valores de ciertas variables puede optimizar el desempeño del sistema, sin embargo, debido a que es una tarea compleja debe ser el último recurso al que debe recurrirse.

Estas variables afectan el funcionamiento y desempeño del kernel, el subsistema de memoria virtual, el subsistema de E/S y las aplicaciones. Se puede modificar temporalmente el kernel cambiando estas variables mientras el sistema esta en ejecución, o se puede modificar permanentemente cambiando los valores de los atributos.

Se usan **atributos** para modificar el kernel sin reconstruirlo. En algunos casos, se pueden modificar cambiando los valores de los **parámetros** en el archivo de configuración del sistema; sin embargo, en este caso se debe construir el kernel nuevamente para usar los nuevos valores. El archivo de configuración `/etc/system` describe los parámetros del sistema y puede ser editado para modificar los valores asignados actualmente a los parámetros o agregar otros parámetros.

Cada parámetro regulable se representa con un nombre y tiene asignado un valor específico en el sistema. Este valor define el límite correspondiente a un recurso particular del sistema que los usuarios o los procesos pueden consumir. El sistema también almacena el valor mínimo y máximo permitido correspondiente al parámetro.

3.4 SINTONIZACION DEL CPU

El sistema debe tener un CPU suficiente poder de procesamiento para responder a las necesidades de desempeño de los usuarios y las aplicaciones. El administrador debe ser capaz de mejorar el desempeño optimizando el CPU y sus aplicaciones.

Un sistema debe ser capaz de asignar eficientemente los ciclos de CPU disponibles entre los procesos. UNIX soporta sistemas multiprocesamiento y procesadores con diferentes velocidades. Los sistemas multiprocesamiento permiten expandir el poder de procesamiento de un sistema agregando procesadores. Las cargas de trabajo que se benefician del multiprocesamiento constan de múltiples procesos que pueden ejecutarse simultáneamente.

Es importante mencionar que incrementar el número de procesadores no siempre es necesario y puede incrementar las demandas en los sistemas de E/S y memoria ocasionando cuellos de botella innecesarios. Por ejemplo, incrementar el número de procesadores no mejorará el desempeño en un sistema que pierde recursos de memoria. Si el sistema abre una gran cantidad de archivos pequeños y tiene acceso a ellos repetidamente, hay que asegurarse que el problema de desempeño no es causado por la memoria virtual o el sistema de E/S.

3.4.1 Identificando cuellos de botella de CPU

Usar el comando `vmstat` para determinar el uso de CPU y observar las siguientes características:

- Un alto porcentaje de tiempo ocioso (*idle time*) en uno o más procesadores puede indicar:
 - ✓ Los procesos están bloqueados debido a que el CPU está en espera de un evento o algún recurso (por ejemplo, memoria o E/S).
 - ✓ El CPU no está ocupado.
- Un bajo porcentaje de tiempo ocioso es la primera indicación de un cuello de botella.

- Un alto porcentaje de tiempo usuario puede ser una característica de un sistema con un buen desempeño. Sin embargo, si el sistema tiene un desempeño bajo, un alto porcentaje de tiempo usuario puede indicar un cuello de botella originado por un código ineficiente de algún usuario, insuficiente poder de CPU, excesiva latencia de memoria o falta de cache. Hay que determinar que secciones de código consumen la mayoría de tiempo de procesamiento, si este fuera el caso.

3.4.2 Optimizando los recursos de CPU

Después de configurar el número apropiado de CPUs para el sistema, hay que sintonizar el desempeño del sistema optimizando los recursos de CPU. Antes de optimizar este recurso, hay que asegurarse que la memoria virtual o el sistema de E/S no están causando los problemas de desempeño. Si optimizando el CPU no se resuelve el problema de desempeño, se debe actualizar el CPU por uno más rápido. Para optimizar los recursos de CPU, se pueden usar los siguientes métodos:

- Usar un planificador de prioridades (Scheduler) para asignar los recursos de CPU. Esto permite asignar un porcentaje de tiempo de CPU a una tarea o aplicación. Asignar la mayor parte de tiempo de CPU para los procesos más importantes, mientras que se limita el uso de CPU a los procesos menos críticos. Para usar el planificador, agrupar todos procesos en una clase y asignar a cada clase un porcentaje de tiempo de CPU.
- Dar prioridad a los trabajos de tal manera que las aplicaciones importantes se ejecuten primero. Usar el comando `nice` y `renice` para modificar la prioridad a un proceso.
- Planificar los trabajos en diferentes tiempos (usar los comandos `at` y `cron`).
- Optimizar las aplicaciones. Se pueden usar varios compiladores y perfiladores para generar códigos más eficientes.

- Incrementar los límites de uso de los recursos del sistema para un programa. Los programas extremadamente grandes se pueden ejecutarse más eficientemente si se incrementa los valores de los parámetros que controlan los límites de uso de los recursos disponibles en el sistema, dichos recursos pueden ser memoria virtual, espacio en el sistema de archivos, tiempo de CPU, etc. Los comandos `limit` y `unlimit` pueden ser empleados para afectar estos límites.

3.4.3 Mejorando el desempeño de una aplicación

Si una aplicación está degradando el desempeño del sistema, se debe identificar las secciones de código que consumen grandes porciones de tiempo de ejecución. En un programa típico, la mayoría de tiempo es gastado en pocas secciones de código. Las aplicaciones bien escritas usan los recursos de CPU, memoria y E/S eficientemente. El administrador debe ser capaz de mejorar el desempeño de las aplicaciones y del sistema siguiendo estas indicaciones:

- Usar la última versión de sistema operativo, compilador, firmware⁴ y parches. Revisar el software del sistema para asegurarse que está usando la última versión de compilador y sistema operativo para desarrollar las aplicaciones. En general, las nuevas versiones de un compilador realizan optimizaciones avanzadas, y las nuevas versiones de sistemas operativos operan eficientemente.
- Asegúrese que las aplicaciones se ejecutan sin errores. Cuando se desarrolla una nueva aplicación, nunca debe ser optimizada hasta que haya sido rigurosamente depurada y probada, es decir, se debe probar las aplicaciones para asegurarse que no tienen errores.
- Optimizar las aplicaciones.

Esto involucraría modificar el código fuente de la aplicación.

⁴ Software interno que se almacena en una memoria ROM (*Read Only Memory*).

- Modificar las prioridades de ejecución de las aplicaciones.
- Usar librerías compartidas.

El uso de librerías compartidas reduce la necesidad de memoria y espacio en disco. Cuando múltiples programas están enlazados a una librería compartida, la cantidad de memoria física usada para cada proceso puede ser reducida significativamente. Sin embargo, el tiempo de ejecución de las librerías compartidas es más lento que si se usan librerías estáticas.

3.5 SINTONIZACION DE LA MEMORIA

La cantidad total de *memoria física* es determinada por la capacidad de las tarjetas de memoria instaladas en el sistema. El sistema distribuye su memoria en unidades llamadas *páginas*.

El desempeño del sistema depende en gran parte de la cantidad total de memoria física y también de la distribución de los recursos del sistema. UNIX permite controlar la asignación de memoria modificando los valores de los atributos del sistema. Sintonizar la memoria usualmente involucra las siguientes tareas:

- Incrementar la asignación de un recurso del sistema para mejorar el desempeño de una aplicación.
- Modificar la forma en que el sistema asigna la memoria.
- Modificar la forma de cómo el sistema de archivos de datos del sistema es almacenado en memoria.
- Configurar el *espacio swap*⁵.

⁵ Área para almacenar ciertos procesos que han sido movidos de la memoria temporalmente para que otros procesos se ejecuten.

3.5.1 Memoria virtual

La memoria virtual describe métodos que se encargan de asignar más memoria de la que esta disponible físicamente, por tal motivo esto hace parecer que el sistema tiene mucho más memoria física. El sistema operativo usa una porción del disco duro como memoria virtual, e intercambia datos entre el disco duro y la memoria física. Hay varias técnicas de memoria virtual, la más común es la paginación.

El subsistema de memoria virtual realiza las siguientes instrucciones:

- Asigna memoria para los procesos, es decir, se asigna un espacio de dirección virtual, la cual es un arreglo de páginas que una aplicación puede mapear dentro de la memoria física.
- Rastrea y administra todas las páginas en el sistema. El subsistema de memoria virtual usa listas de página para rastrear la localización de las páginas de la memoria física. Se puede emplear el comando `vmstat` para determinar la actividad de paginación del sistema.
- Usa paginación y swapping para asegurarse que hay suficiente memoria para que los procesos se ejecuten y se almacenen en el sistema de archivos. Cuando la demanda de memoria es alta y un proceso activo requiere más memoria, la memoria virtual usa paginación para mover porciones de procesos (páginas) a disco de tal forma que la memoria física pueda ser usada por otros procesos. El swapping copia completamente un proceso a disco para liberar grandes cantidades de memoria física para nuevos procesos. Es importante hacer notar, que de vez en cuando debido a la carga de trabajo, la memoria en el sistema puede ser insuficiente y debido a ello se incrementa la actividad de paginación. Dicha actividad debe considerarse un problema si se presenta continuamente.

3.5.2 Buffer cache

Un buffer es un área de memoria usada para almacenamiento de datos. El sistema operativo UNIX usa un buffer cache como parte de la memoria del kernel y es usado para almacenar datos temporalmente para mejorar el desempeño reduciendo la latencia. Este buffer fue diseñado para tener un rápido acceso a los datos subsecuentemente, por lo tanto se lleva menos tiempo para realizar un acceso que una memoria normal. Debido a esta característica la cache puede mejorar el desempeño de subsistema de E/S reduciendo el número de operaciones de E/S a disco.

El administrador del sistema puede mejorar el rendimiento optimizando el buffer cache. Sin embargo, incrementar la cantidad de memoria disponible para el buffer cache afectará el subsistema de memoria virtual y puede incrementar la actividad de paginación y swapping.

Las siguientes salidas pueden indicar que el tamaño del buffer cache es demasiado pequeño para una configuración:

- La salida del comando `vmstat` muestra que muchas páginas son movidas de disco a memoria y algunas o ninguna son movidas de memoria a disco.
- La salida del comando `iostat` muestra poca o ninguna actividad de disco.

3.5.3 Configuración de la memoria y espacio swap

Las tareas para configurar la memoria y el espacio swap incluye:

- Determinar los requerimientos de memoria física. La cantidad de memoria instalada en el sistema debe ser capaz de proporcionar un nivel aceptable para el desempeño de las aplicaciones de usuarios. Para determinar los requerimientos de memoria, se debe obtener la siguiente información:

- ✓ La cantidad de memoria que el subsistema de memoria virtual requiere para almacenar los procesos.
- ✓ La cantidad de memoria que se requiere para almacenar los datos del sistema de archivos.
- Determinar la cantidad de espacio requerida para el almacenamiento temporal de datos que se intercambian entre el disco duro y la memoria física (espacio swap). El desempeño del sistema depende en gran parte de la configuración del espacio swap. El comando `iostat` puede ser de utilidad para ver el uso de disco. Considerar los siguientes pasos para configurar el espacio swap:
 - ✓ Configurar el espacio swap al inicializar el sistema.
 - ✓ Usar discos rápidos para reducir la latencia de paginación.
 - ✓ Dividir el espacio swap en múltiples discos.

3.5.4 Optimizando la memoria virtual

El subsistema de la memoria virtual es una fuente primaria para problemas de desempeño. El desempeño se puede degradar si el subsistema de memoria virtual no puede proseguir con la demanda de memoria y la actividad de paginación, ocasionando así demasiada actividad de swapping.

Un cuello de botella de memoria puede causar un cuello de botella de E/S, ya que el swapping y paginación excesiva degradan el rendimiento e indican que la cantidad de trabajo natural ha excedido la memoria disponible. La actividad de paginación puede ser indicada por las siguientes salidas:

- El comando `vmstat` muestra información de las páginas que son movidas de memoria a disco y viceversa.
- El comando `iostat` muestra información de la actividad excesiva de E/S.

Las recomendaciones de optimización que pueden proporcionar un mejor desempeño involucran las siguientes áreas:

- Asignación de recursos del sistema.
 - ✓ Incrementar el espacio disponible.
 - ✓ Incrementar los recursos del kernel disponibles para los procesos.
- Asignación de memoria.
 - ✓ Modificar el porcentaje de memoria asignada al buffer cache.

3.6 SINTONIZACION DEL SUBSISTEMA DE ALMACENAMIENTO

Aunque muchos dispositivos son parte del subsistema de E/S, el disco es el más afectado. La configuración de almacenamiento en disco puede tener un impacto significativo en el desempeño del sistema, esto debido a que las operaciones de E/S son usadas para las operaciones del sistema de archivos y también por la memoria virtual para la paginación y swapping.

Para configurar el subsistema de almacenamiento que responderá a las necesidades de desempeño y disponibilidad, primero hay que entender los requerimientos de E/S de los usuarios y las aplicaciones. Después de configurar el subsistema de almacenamiento, puede ser optimizado para mejorar el desempeño.

3.6.1 Entendiendo el subsistema de almacenamiento

Las operaciones de E/S a disco son significativamente más lentas que la transferencia de datos. Debido a que los discos son usados para almacenar los datos y para el espacio swap de la memoria virtual, una configuración o una sintonización incorrecta puede degradar el desempeño del sistema completo.

El desempeño del subsistema de E/S puede afectar las siguientes variables:

- Características de carga de trabajo. El desempeño depende de cómo los usuarios ejecuten las aplicaciones. Por ejemplo, una carga de trabajo puede involucrar operaciones de E/S de lectura y escritura. Además algunas cargas del sistema requieren una latencia baja y un alto rendimiento, mientras que otras requieren de una rápida transferencia de datos (alto ancho de banda).

Una latencia baja es importante para que múltiples transferencias pequeñas de datos. Un ancho de banda alto es importante para los sistemas que ejecutan grandes secuencias de transferencias de datos, tales como servidores de bases de datos.

- Capacidad de desempeño del hardware. Se recomienda que se use hardware con características de alto grado de desempeño. Por ejemplo, los discos con un alto rango de revoluciones por minuto (RPM) proporcionan un mejor desempeño.
- Asignación de memoria al buffer cache. El buffer cache es asignado a una porción de memoria física y almacena temporalmente los datos del sistema de archivos actual para lecturas y escrituras. El buffer cache reduce el número de operaciones de disco para los sistemas de archivos sirviendo como una capa entre el disco y el sistema operativo.
- Valores de las variables del kernel. El desempeño del disco para las operaciones de E/S depende de que los valores de las variables del kernel sean apropiados para la carga del trabajo del sistema. Estos valores pueden ser modificados para obtener un óptimo desempeño del sistema.
- Configuración del sistema de archivos. El sistema de archivos es usado para organizar y administrar los archivos.

3.6.2 Administración de discos y archivos

El método tradicional de administración de discos y archivos es dividir cada disco en áreas lógicas llamadas *particiones*, y crear un sistema de archivos sobre una partición. Un disco puede consistir de 8 particiones con un tamaño determinado. Cada disco tiene un esquema de partición por default. El tamaño de la partición determina la cantidad de datos que puede almacenar.

Un método alternativo de administrar discos con particiones estáticas es usar un Administrador de Almacenamiento Lógico o LSM (Logical Storage Manger). Este administrador permite crear discos virtuales de acuerdo al rendimiento y capacidad del sistema, además de agregar y borrar discos, balancear la carga y ejecutar otras tareas de administración de almacenamiento. Este administrador puede mejorar el rendimiento de E/S, proporcionando una alta disponibilidad de los datos y permite administrar el almacenamiento más eficientemente. Sus características incluyen:

- Espacio de almacenamiento.
- Distribuye la carga a través de los discos.
- Concatenación de discos (crea un gran volumen desde múltiples discos).
- Interfaz gráfica de usuario (GUI) para una fácil administración de los discos.
- Información detallada del desempeño del sistema.

Usar hardware de alto rendimiento ayuda a obtener un mejor desempeño del disco. Por ejemplo un disco con un número alto de revoluciones por minuto (RPM) tiene un bajo tiempo de acceso a disco (latencia), por lo tanto son especialmente benéficos para el desempeño de transferencias de datos secuenciales. Por otro lado, es conveniente distribuir la carga de las operaciones de E/S a través de los discos para obtener máximos beneficios. Distribuir el espacio swap a través de diferentes discos permite que la paginación y swapping sean más eficientes y ayudar a prevenir cuellos de botella.

3.7 SINTONIZACION DEL SUBSISTEMA DE RED

La mayoría de los recursos usados por la red son asignados o ajustados dinámicamente, sin embargo, hay algunas recomendaciones de sintonización que pueden emplearse para mejorar el desempeño:

- Revisar los procesos que ejecutan los usuarios. Una red llega a tener un tráfico excesivo cuando la transferencia de datos es muy extensa o pesada. Por ejemplo si un usuario empieza a transferir grandes archivos simultáneamente, esto ocasionará tráfico en la red.
- Emplear el comando `netstat` para monitorear la actividad de la red.
- Particularmente en los sistemas que fungen como servidores de Internet, los cuales incluyen servidores Web, servidores de correo y servidores de FTP, el desempeño de la red es afectado cuando los recursos disponibles no son capaces de continuar con la demanda de trabajo. Para obtener el mejor rendimiento en la red, se debe entender la carga de trabajo y las características del desempeño del hardware de la red (ruteadores, hubs, interfaz de red, etc.). Diferentes interfaces de red tienen diferentes características de rendimiento. Por ejemplo, una interfaz FDDI proporciona mejor desempeño que una interfaz Ethernet.
- Se debe determinar si la fuente de un problema de desempeño es una aplicación, una conexión de red, un controlador de red o un patrón de comunicación.
- El sistema de archivos de red o NFS (Network File System), comparte el buffer cache con el subsistema de memoria virtual y los sistemas de archivos local. La mayoría de los problemas de desempeño del NFS pueden ser atribuidos a cuellos de botella en la memoria virtual, la red o el subsistema de disco.

- La pérdida de paquetes en la red puede degradar severamente el desempeño del sistema. Estas pérdidas pueden ser causadas por un servidor congestionado, corrupción de paquetes durante la transmisión (causados por malas conexiones eléctricas o ruido en la interfaces Ethernet) y por ruteadores que no reenvían la información rápidamente. Esto ocasiona que la red sea muy lenta debido a retransmisiones repetidas y excesivas. Emplear el comando ping para determinar si hay pérdida de paquetes.

4. MONITOREO Y OPTIMIZACIÓN DE UN SISTEMA REAL

Este apartado proporciona una introducción básica para optimizar un sistema UNIX basado en el sistema operativo Solaris, con el fin de obtener un mejor desempeño en servidores que ofrecen servicios de Red incluyendo servidores Web, servidores FTP y servidores de correo. Incluye estrategias que se pueden usar para mejorar el desempeño de un sistema. Esto no quiere decir que un programa individual correrá mucho más rápido, sin embargo, si el sistema tiene un mejor rendimiento, los procesos individuales se ejecutarán más rápido. Lo que se pretende es estructurar el sistema para maximizar su desempeño total, es decir, estructurar el sistema de tal manera que sus recursos sean usados y distribuidos eficientemente entre los usuarios. Para ello hay que averiguar que parte del sistema tiene una sobrecarga, entender como disminuir dicha carga y realizar una actualización si es necesario.

4.1 ESTADO ACTUAL

La Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM es la entidad universitaria encargada de supervisar los sistemas centrales de cómputo académico; de operar y extender las telecomunicaciones, y de ofrecer a los diferentes sectores de la población múltiples y variados servicios que giran en torno a 4 ejes que fundamentan su existencia: docencia, telecomunicaciones, apoyo a la investigación y desarrollo de sistemas computacionales. La DGSCA a su vez cuenta con diversas entidades para cumplir con este compromiso con un alto nivel de profesionalismo y sentido de responsabilidad; entre ellas se cuenta la Coordinación de Servicios de Red (CSR), el cual es un departamento que ha crecido muy rápidamente debido a la demanda de los servicios de que ofrece y que busca soluciones para manejar este crecimiento controlando la carga de trabajo para brindar un mejor servicio a los usuarios. Por este motivo la CSR ha decidido implementar herramientas que permitan localizar los problemas de desempeño que se presenten y al mismo tiempo averiguar como solucionar dichas averías sin tener que gastar en compras ineficaces (compra de costosas máquinas). Afortunadamente, hay un cierto número de herramientas que pueden ayudar a resolver problemas, incluyendo monitoreo de rendimiento, planeación de capacidad, administración de cuentas de usuarios, administración de archivos, etc. y que proporcionan soluciones a la carga de trabajo.

4.1.1 Infraestructura actual

La infraestructura actual de la CSR esta basada en el modelo Cliente/Servidor, donde un número de servidores que corren largas aplicaciones y resguardan una gran cantidad de datos ofrecen servicios para diversos clientes. El departamento cuenta con máquinas de escritorio con ambiente operativo Windows además de los servidores SUN que soportan aplicaciones de UNIX, aplicaciones VMS, Windows NT y bases de datos como Sybase y Oracle.

La CSR tiene máquinas específicas que soportan las aplicaciones en red, estas necesitan ser monitoreadas para determinar su carga actual de trabajo así como su configuración actual. A continuación se muestra las características (Tabla 4.1) de cada una de ellas:

Máquinas para ser monitoreadas				
Nombre	Hardware	SO	Aplicaciones	#CPUs
E5000	Ultra Enterprise 5000	Solaris 2.7	Sybase, Oracle, Web, Correo, FTP, Telnet, PHP, Perl, Java, SQL.	2
E4000	Ultra Enterprise 4000	SunOS 2.7	Sybase, Oracle, Web, Correo, FTP, Telnet, PHP, Perl, Java, SQL.	4
Sunsite	SPARCserver 1000E	SunOS 2.7	Web, FTP anónimo, correo, Telnet.	2
Tzetzal	SPARCstation 20	SunOS 2.8	Correo	1
Mitla	SPARCstation 10	SunOS 2.8	Correo	2
Sella	SPARCstation 4	SunOS 2.6	Correo	1

Tabla 4.1 Características de los servidores UNIX.

El monitoreo del rendimiento en esos servidores es limitado, en el mejor de los casos, y las herramientas que están disponibles son en gran parte inusuales debido a la complejidad y a la carencia de datos para la planeación de una tendencia análisis/capacidad. Por esta razón, es conveniente realizar una evaluación de las herramientas disponibles actualmente, con el fin de seleccionar los programas que habrán de utilizarse de acuerdo a los requerimientos del sistema. Algunas herramientas que estos servidores proporcionan y que proporcionan estadísticas de como trabaja un sistema son:

TESIS CON
 FALLA DE ORIGEN

- Las utilidades `sar` y `sadc`, obtienen y reportan estadísticas de la actividad del sistema.
- El comando `ps`, el cual proporciona información de los procesos activos.
- El `meter` de rendimiento, muestra de forma gráfica el estado actual de los sistemas de la red.
- Los comandos `vmstat` e `iostat`, proporcionan la actividad total del sistema proporcionando información sobre la actividad de memoria virtual, uso de disco y actividad de CPU.
- El comando `swap`, puede ser usado para desplegar información sobre el espacio swap disponible en el sistema.
- Los comandos `netstat` y `nfsstat`, los cuales despliegan información acerca del rendimiento de la red.

Con esto nos damos cuenta que la CSR no tiene suficientes herramientas para realizar un monitoreo eficiente y por ende una sintonización adecuada de los servidores, sin embargo, como ya se vio, existen herramientas adicionales que se pueden utilizar debido a los datos que proporcionan. Varias de las cosas que se pueden hacer para obtener estadísticas relacionadas al desempeño (scripts, filtros, gráficas, etc.), desperdician tiempo de diseño, configuración y se limitan a una máquina al mismo tiempo; mientras que estas herramientas ya tienen integrados la mayoría de estos beneficios.

Por lo anterior, hubo necesidad de realizar una revisión de las herramientas adicionales disponibles vía Internet, para supervisar y evaluar el rendimiento de un sistema Solaris:

➤ SE

Es un kit de utilerías que corre bajo ambiente Solaris, y contiene muchos scripts predefinidos los cuales pueden ser usados para evaluar el estado de uno o varios sistemas a la vez. Se puede emplear como herramienta de monitoreo y optimización. Se obtienen diversas estadísticas por la gran cantidad de datos que se filtran consiguiendo que sean claros y comprensibles.

➤ Proctool

Proctool es una herramienta para la administración y monitoreo de procesos. Está orientado a la administración del sistema, análisis del desempeño, así como observaciones generales de usuarios. Proctool abarca muchas de las funciones proporcionadas por los comandos `ps`, `renice`, y `pricntl`. También proporciona algunas funciones de los comandos `dispadmin` y `sar`.

➤ SyMON.

Es una aplicación que se puede usar para monitorear el desempeño de uno o varios servidores. Esta herramienta hace uso del protocolo SNMP (Simple Network Management Protocol) internamente.

➤ PATROL.

Proporciona un análisis de desempeño actual e histórico de las aplicaciones críticas y optimiza el uso de los recursos del sistema para obtener una máxima productividad. Los resultados del análisis estadístico avanzado se presentan gráficamente.

4.1.1.1 Criterio de evaluación del software para el monitoreo del desempeño.

La manera más fácil de ordenar la información obtenida de la evaluación y conseguir la mejor comparación es hacer una lista de los criterios que se pueden evaluar de los paquetes de software, asegurando un conjunto estandarizado de características para buscar las cosas que deseamos del software y la manera en que los paquetes sobrealgan de sus competidores:

- Plataformas. Las herramientas deben correr en el sistema operativo Solaris Versión 2.6, 7 y 8.
- Interfaz gráfica de monitoreo en tiempo real. Monitorear la utilización de CPU, actividad del disco, uso de memoria, etc.; mediante una interfaz gráfica de usuario(GUI) con gráficas a colores, para tener una visión general y rápida del desempeño del sistema.
- Definición de umbrales¹ para generar alertas. El administrador debe definir reglas relacionadas a la actividad del sistema para que las alarmas se disparen. Por ejemplo, cuando el uso de CPU sobrepase el 80%.
- Definición de acciones de alertas. El administrador debe ser capaz de definir las acciones que se llevaran a cabo cuando las alarmas son generadas. Esas acciones pueden ser: fijación de paginación, enviar un e-mail o ejecución de un script para hacer un arreglo automatizado del problema.
- Lenguaje de escritura. El producto debe tener su propio lenguaje de escritura para editar las acciones de alerta, definiciones de alertas, monitoreo de casos especiales, monitoreo de aplicaciones, etc.
- Distribución. Las herramientas deben tener una arquitectura de red donde los programas deben correr en muchas máquinas enviando datos para una o más consolas. La consola puede monitorear solo la máquina en la cual esta corriendo.
- Capacidad de historia de análisis. El software debe tener algún archivo donde pueda guardar datos de manera que puedan ser consultado para un análisis posterior.

Después de evaluar las herramientas, queda claro que existen diversas estrategias disponibles, así como ventajas y desventajas. Por lo tanto se buscará la forma de combinar las herramientas de tal manera que se obtenga un mayor beneficio.

¹ Mínimo necesario para que un suceso sea perceptible.

4.2 ESTRATEGIAS PARA EL MEJORAMIENTO DEL DESEMPEÑO

El objetivo esencial es la sintonización del desempeño, lo cual involucra el manejo y administración de los recursos del sistema. La administración de recursos involucra las tareas de:

- Identificar cargas de trabajo.
- Definir y priorizar metas.
- Identificar los recursos requeridos.
- Minimizar los requerimientos de un recurso.
 - Utilizar el recurso adecuado.
 - Estructurar el uso en paralelo de los recursos.
- Controlar la asignación de recursos.
- Aplicar recursos adicionales como sea necesario.

La evaluación objetiva del rendimiento del sistema se realizará en dos fases: una fase de optimización y una fase de pruebas. En la fase de optimización, se obtendrá una serie de datos antes de la sintonización, se harán modificaciones manuales y si es necesario algunos parámetros del kernel serán regulados hasta obtener resultados satisfactorios. Finalmente en la fase de prueba, se recolectaran nuevamente una serie de estadísticas para evaluar el desempeño antes y después de la sintonización. Durante estas dos fases se incluyen las siguientes actividades:

- Análisis de todo el funcionamiento del sistema. Entender para que está diseñado el sistema, que hacen los usuarios y cuales son sus requerimientos.
- Análisis del servidor. Recolección de estadísticas del servidor para determinar la carga de trabajo actual del sistema y ver que cambios se pueden realizar.

- Revisión del sistema para proporcionar un reporte del estado actual del sistema. Este reporte se realiza después del análisis del sistema y proporcionará información de la capacidad actual del sistema y su capacidad de proporcionar servicios a los clientes o usuarios.
- Ejecutar pruebas de simulación. Diseño de pruebas de simulación basadas en las aplicaciones y carga de trabajo actuales del sistema.
- Presentar recomendaciones de optimización.

4.3 FASE DE SINTONIZACION

La CSR contribuye a la optimización de todos los procesos en beneficio de la propia institución y de sus usuarios, estos se reflejan en el ahorro de tiempo y en el aprovechamiento de los recursos técnicos y financieros. Dentro de los servicios que ofrece se cuentan:

- Evaluación de hardware y software
- Consultoría en configuraciones avanzadas en UNIX
- Aplicaciones avanzadas para académicos e investigadores de la UNAM
- Administración y monitoreo de servidores UNIX
- Alojamiento de servidores Web
- Correo electrónico
- Telnet
- FTP
- Desarrollo y asesoría de sitios Web
- Bases de datos
- Sistemas de búsqueda
- Publicaciones electrónicas
- Seguridad en cómputo

Para cumplir con estas tareas es muy importante que el departamento cuente con servidores potentes con un alto grado de rendimiento con el fin de brindar soporte y satisfacer las necesidades de los usuarios. Para cumplir con este objetivo, después de asimilar los conocimientos de los capítulos anteriores, es importante realizar un análisis de la carga de los servidores UNIX, obtener estadísticas de su rendimiento y realizar los cambios que sean necesarios para obtener el mayor rendimiento posible de tal forma que los usuarios que emplean los servicios antes mencionados trabajen en un ambiente en donde se obtengan una respuesta deseada.

4.3.1 Carga del sistema

La carga promedio del sistema proporciona un resumen conveniente de la actividad de un sistema, y es la primer estadística que se debe de observar cuando hay problemas de desempeño. Las cargas de trabajo tienden a caer en un número pequeño de clases:

- Estación de trabajo. Una carga de trabajo que consiste de un simple usuario que envía el trabajo mediante un teclado y recibe resultados en la pantalla del sistema. El objetivo en una carga de trabajo de este tipo es minimizar el tiempo de respuesta de las peticiones que hace el usuario.
- Multiusuario. Una carga de trabajo que consiste en un número de usuarios que envían trabajo por medio de terminales individuales. El objetivo aquí es maximizar el rendimiento total del sistema preservando un tiempo de respuesta deseado.
- Servidor. Una carga de trabajo que consiste de una serie de peticiones desde otros sistemas. Por ejemplo, la modificación de un archivo es una petición de lectura/escritura, la ejecución de programas grandes, transacciones de bases de datos, etc. En esencia, es la actividad de E/S de una carga de trabajo de tipo multiusuario; por lo que el objetivo es el mismo.

De acuerdo a la clasificación anterior, nos damos cuenta que la carga de trabajo de los sistemas de la CSR es de tipo servidor. Una vez que se ha determinado el tipo de carga de trabajo es necesario analizar dicha carga.

Para la realización de mediciones se utilizó la máquina Solaris Ultra Enterprise 5000 denominada e5000, con sistema operativo SunOS 7, el cual cuenta con 2 procesadores y 380 MB de RAM. Dicho servidor es el que presenta la mayor carga de trabajo actualmente y de aquí se tomarán las bases para los cambios necesarios en los otros servidores.

La carga promedio del sistema se puede ver en primera instancia usando el meter de rendimiento (`perfmeter`) que ofrece una manera de visualizar y monitorear varios parámetros de rendimiento para uno o varios sistemas en la red. El meter despliega gráficamente el status de los siguientes parámetros:

- Carga promedio
- CPU
- Swap
- Paquetes Ethernet
- Actividad de paginación por segundo
- Número de interrupciones de dispositivos por segundo
- Colisiones por segundo detectadas en la Ethernet
- Errores por segundo en paquetes recibidos
- Tráfico en disco en transferencias por segundo

La principal ventaja de un meter es que da un panorama inmediato sobre el rendimiento del sistema. Se puede ejecutar un comando o un programa, y observar la gráfica para ver si el uso de CPU se incrementa. Se puede desplegar la carga de varias máquinas de la red y seleccionar uno con la carga más pequeña en la cual ejecutar un programa intensivo de CPU.

El siguiente meter (Fig. 4.1) muestra la actividad de CPU del servidor e5000. Las curvas se mueven de derecha a izquierda, mostrando la carga promedio del sistema.

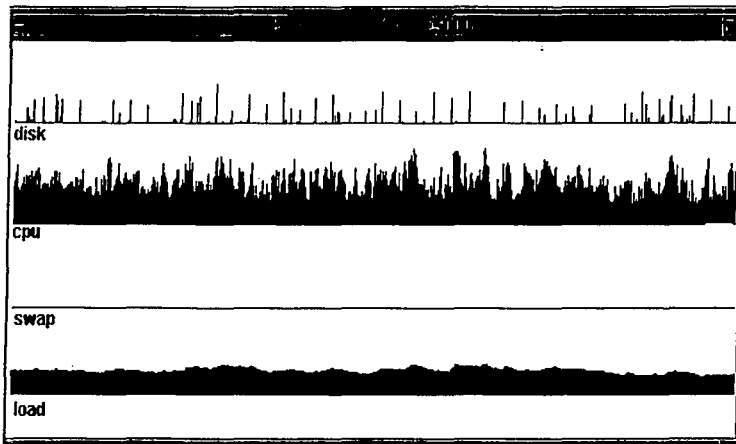


Fig. 4.1 Perfmeter.

La carga promedio del sistema también se puede ver mediante los comandos `uptime` y `w`. A continuación se muestran las primeras estadísticas obtenidas del servidor e5000:

TESIS CON
FALLA DE ORIGEN


```
e5000% uptime
```

```
3:34pm up 31 days, 9 users, load average: 3.32, 2.83, 2.38
```

Este reporte muestra el tiempo en que el servidor ha estado trabajando. Reporta 3 promedios de cargas. La primera carga (3.32) se mide durante el último minuto, la segunda (2.83) durante los últimos 5 minutos, mientras que la tercera (2.38) se midió durante los últimos 15 minutos. Es muy frecuente notar que la carga promedio sube o baja. En este caso está subiendo, ya que la carga promedio obtenida durante el último minuto es más grande que la que se obtuvo durante los últimos 15.

```
e5000% w -u
```

```
3:52pm up 31 days, 8 users, load average: 3.80, 3.55, 3.08
```

El reporte es similar, en este caso los resultados desplegados muestran más claramente que la carga de trabajo va en ascenso. Un sistema con una carga pesada de más de 10 es extremadamente alta para un sistema de computo grande, si un sistema trabaja regularmente con este tipo de carga, se debe realizar algunos pasos para distribuir el uso de CPU. Cuando la carga está entre 3 y 7, en este nivel los usuarios se empiezan a quejar que el sistema está lento y que los comandos tardan mucho en ejecutarse. En este sentido como medida inmediata se puede sugerir a los usuarios correr sus trabajos en la noche. Una carga menor de 3 no es causa de alerta, ya que la carga en este momento es ligera, sin embargo esto depende de los procesos y aplicaciones que se corren en el sistema.

Debido a que el sistema tiene una carga promedio que va en ascenso, se analizarán los trabajos que se están ejecutando actualmente. El comando `ps` proporciona la información necesaria para determinar que procesos son responsables de la carga del sistema.

e5000% ps -el

F	S	UID	PID	PPID	C	PRI	NI	SZ	TTY	TIME	CMD
19	T	0	0	0	0	0	SY	0	?	0:31	sched
8	S	0	1	0	0	41	20	169	?	6:38	init
19	S	0	2	0	0	0	SY	0	?	0:01	pageout
19	S	0	3	0	0	0	SY	0	?	715:55	fsflush
8	S	0	649	1	0	40	20	185	?	0:00	sac
8	S	0	429	418	0	49	20	69	?	0:00	vxrelocd
8	S	0	294	1	0	51	20	252	?	1:44	inetd
8	S	0	11	1	0	41	20	245	?	0:02	vxconfig
8	S	0	351	1	0	47	20	332	?	0:00	lpsched
8	S	0	268	1	0	49	20	251	?	0:00	keyserv
8	S	1508	14226	7149	0	41	20	259	?	0:05	httcpd
8	S	0	266	1	0	41	20	266	?	0:04	rpebind
8	S	0	301	1	0	41	20	214	?	0:00	lockd
8	S	0	317	1	0	41	20	476	?	0:12	automoun
8	S	0	418	1	0	89	20	69	?	0:00	vxrelocd
8	S	0	299	1	0	41	20	273	?	0:00	statd
8	S	0	243	1	0	41	20	189	?	2:16	cryptora
8	S	1507	1720	12179	0	51	20	289	?	0:02	httcpd
8	S	0	335	1	0	55	20	199	?	3:34	cron
8	S	0	341	1	0	47	20	342	?	2:12	nscd
8	S	500	29536	29527	0	75	20	279	?	0:00	httcpd
8	S	0	387	1	0	41	20	113	?	0:03	utmpd
8	S	0	390	386	0	40	20	217	?	0:00	suntechd
8	S	0	374	1	0	98	20	149	?	0:00	powerd
8	S	0	386	1	0	41	20	207	?	0:04	lmgrd.st
8	S	0	405	1	0	41	20	296	?	0:02	vold
8	S	0	400	1	0	41	20	274	?	0:08	sendmail
8	S	0	456	1	0	41	20	300	?	0:00	mountd
8	S	2000	584	578	0	41	20	290	?	0:17	httcpd
8	S	1007	21472	613	0	41	20	283	?	0:00	httcpd
8	S	0	652	649	0	40	20	194	?	0:00	ttymon
8	S	0	430	429	0	41	20	110	?	0:00	vxnotify
8	S	505	11763	7520	0	41	20	629	?	0:01	httcpd
8	S	0	458	1	0	41	20	214	?	0:00	nfsd
8	S	1007	615	613	0	51	20	287	?	0:00	httcpd
8	S	107	472	1	0	62	20	114	?	0:00	RUN_SYBASE
8	S	700	499	1	0	41	20	248	?	0:01	httcpd
8	S	107	474	1	0	62	20	114	?	0:00	RUN_SYBA
8	S	0	475	1	0	41	20	723	?	0:00	dtlogin
8	S	107	476	472	0	41	20	981	?	1:05	backupse
8	S	107	477	474	0	41	20	6561	?	312:51	dataserv
8	S	2000	578	1	0	40	20	280	?	0:00	httcpd
8	S	2000	585	578	0	41	20	290	?	0:15	httcpd
8	S	1007	616	613	0	51	20	286	?	0:00	httcpd
8	S	1000	12810	27932	0	41	20	580	?	0:00	httcpd
8	S	2000	587	578	0	41	20	290	?	0:15	httcpd
8	S	1007	613	1	0	40	20	280	?	0:00	httcpd
8	S	1007	617	613	0	41	20	286	?	0:00	httcpd
8	S	2000	588	578	0	41	20	290	?	0:16	httcpd
8	S	2000	583	578	0	41	20	290	?	0:17	httcpd
8	S	1007	619	613	0	41	20	282	?	0:00	httcpd

TESIS CON
 FALLA DE ORIGEN

8	S	1007	620	613	0	41	20	282	?	0:00	httpd
8	S	0	630	1	0	51	20	234	?	0:30	snmpdx
8	S	503	627	1	0	40	20	278	?	0:00	httpd
8	S	0	640	1	0	51	20	396	?	0:00	snmpXdm1
8	S	503	653	627	0	41	20	282	?	0:00	httpd
8	S	0	639	1	0	65	20	318	?	0:00	dmispd
8	S	503	654	627	0	41	20	282	?	0:00	httpd
8	S	503	655	627	0	41	20	282	?	0:00	httpd
8	S	503	656	627	0	41	20	282	?	0:00	httpd
8	S	503	657	627	0	41	20	282	?	0:00	httpd
8	S	503	658	627	0	41	20	282	?	0:00	httpd
8	S	503	659	627	0	41	20	281	?	0:00	httpd
8	S	0	13813	294	0	41	20	197	?	0:27	rpc.rsta
8	S	505	27891	7520	0	41	20	634	?	0:01	httpd
8	S	0	666	630	0	41	20	209	?	0:13	mibiisa
8	S	505	27893	7520	0	41	20	633	?	0:01	httpd
8	S	1000	27948	27932	0	41	20	594	?	0:00	httpd
8	S	1507	1719	12179	0	41	20	290	?	0:02	httpd
8	S	3003	3874	3871	0	41	20	143	pts/4	0:00	csH
8	S	2301	8192	11445	0	41	20	288	?	0:01	httpd
8	S	500	29531	29527	0	41	20	281	?	0:00	httpd
8	S	2301	11472	11445	0	41	20	288	?	0:01	httpd
8	S	2301	11450	11445	0	41	20	289	?	0:01	httpd
8	S	111	28431	24634	0	41	20	779	?	1:59	httpd
8	S	2000	3154	578	0	41	20	290	?	0:14	httpd
8	S	0	3871	294	0	81	20	194	?	0:00	in.telne
8	S	1007	21469	613	0	41	20	283	?	0:00	httpd
8	S	2301	11446	11445	0	41	20	288	?	0:01	httpd
8	S	0	5633	294	0	81	20	194	?	0:00	in.telne
8	S	2000	3156	578	0	41	20	290	?	0:14	httpd
8	S	1507	12186	12179	0	51	20	289	?	0:02	httpd
8	S	1007	21473	613	0	41	20	287	?	0:00	httpd
8	S	505	27895	7520	0	41	20	634	?	0:01	httpd
8	S	700	4719	499	0	41	20	252	?	0:00	httpd
8	S	505	27897	7520	0	41	20	634	?	0:01	httpd
8	S	2301	11471	11445	0	41	20	288	?	0:01	httpd
8	S	1507	8195	12179	0	51	20	302	?	0:02	httpd
8	S	500	29533	29527	0	41	20	279	?	0:00	httpd
8	S	1508	7373	7149	0	41	20	259	?	0:05	httpd
8	S	5027	5060	5059	0	41	20	4054	?	93:45	dataserv
8	S	1508	9304	7149	0	41	20	260	?	0:05	httpd
8	S	1508	7154	7149	0	41	20	259	?	0:05	httpd
8	S	2301	11448	11445	0	51	20	288	?	0:01	httpd
8	S	1000	27937	27932	0	41	20	579	?	0:00	httpd
8	S	111	24635	24634	0	41	20	781	?	1:04	httpd
8	S	1508	7152	7149	0	41	20	259	?	0:05	httpd
8	S	2000	3155	578	0	41	20	290	?	0:14	httpd
8	Z	3004	2186	2175	0	0				0:00	<defunct>
8	S	2000	3956	578	0	41	20	290	?	0:15	httpd
8	S	0	2175	294	0	48	20	194	?	0:00	in.telne
8	S	0	7520	1	0	40	20	571	?	0:00	httpd
8	S	1508	14220	7149	0	41	20	259	?	0:05	httpd
8	S	2301	11447	11445	0	41	20	288	?	0:01	httpd
8	S	500	29535	29527	0	65	20	279	?	0:00	httpd

TESIS CON
FALLA DE ORIGEN

8	S	1508	25502	7149	0	41	20	259	?	0:04	httpd
8	S	1508	7153	7149	0	41	20	260	?	0:05	httpd
8	S	1508	7149	1	0	40	20	250	?	0:00	httpd
8	S	1508	7150	7149	0	41	20	259	?	0:05	httpd
8	S	1507	1721	12179	0	51	20	305	?	0:02	httpd
8	S	1508	151	7149	0	41	20	259	?	0:06	httpd
8	S	0	24634	1	0	41	20	571	?	0:00	httpd
8	S	1007	21470	613	0	51	20	288	?	0:00	httpd
8	S	1507	13118	13108	0	41	20	1986	?	0:00	.netscape
8	S	1507	12181	12179	0	41	20	304	?	0:02	httpd
8	S	1000	27936	27932	0	41	20	597	?	0:00	httpd
8	S	2301	11445	1	0	40	20	280	?	0:00	httpd
8	S	1507	12179	1	0	40	20	280	?	0:00	httpd
8	S	1507	12180	12179	0	41	20	289	?	0:02	httpd
8	S	2301	11449	11445	0	41	20	288	?	0:01	httpd
8	S	2301	11470	11445	0	41	20	288	?	0:01	httpd
8	S	0	27622	294	0	45	20	194	?	0:00	in.telne
8	S	1507	12183	12179	0	41	20	304	?	0:02	httpd
8	S	1507	12182	12179	0	41	20	303	?	0:02	httpd
8	S	1507	12184	12179	0	41	20	289	?	0:02	httpd
8	S	500	29527	1	0	40	20	278	?	0:00	httpd
8	S	700	4639	499	0	41	20	252	?	0:00	httpd
8	S	1507	13108	1	0	41	20	3292	?	3:17	.netscape
8	S	145	26782	1	99	41	20	5986	?	1259:51	.netscape
8	S	145	11888	1	0	41	20	1968	?	1:20	.netscape
8	S	500	29534	29527	0	55	20	279	?	0:00	httpd
8	S	505	27706	27624	0	41	20	152	pts/0	0:01	pico
8	S	700	4641	499	0	41	20	252	?	0:00	httpd
8	S	1000	27947	27932	0	41	20	582	?	0:00	httpd
8	S	505	23295	7520	0	41	20	631	?	0:00	httpd
8	S	500	29532	29527	0	41	20	281	?	0:00	httpd
8	S	5027	5059	1	0	51	20	117	?	0:00	RUN_syba
8	S	505	27892	7520	0	41	20	630	?	0:01	httpd
8	S	0	27932	1	0	40	20	573	?	0:00	httpd
8	S	1000	27935	27932	0	41	20	597	?	0:00	httpd
8	S	1007	21471	613	0	41	20	283	?	0:00	httpd
8	S	500	29528	29527	0	41	20	281	?	0:00	httpd
8	S	505	27894	7520	0	41	20	629	?	0:01	httpd
8	S	1000	27933	27932	0	41	20	597	?	0:00	httpd
8	T	111	9391	1	0	41	20	383	?	0:01	isql
8	S	0	8454	294	0	81	20	194	?	0:00	in.telne
8	S	1000	27934	27932	0	41	20	579	?	0:00	httpd
8	S	2301	2273	11445	0	41	20	288	?	0:01	httpd
8	S	505	27896	7520	0	51	20	633	?	0:01	httpd
8	S	0	2522	294	0	71	20	194	?	0:00	in.telne
8	T	111	29600	1	0	41	20	383	?	0:00	isql
8	S	2000	29538	578	0	41	20	290	?	0:10	httpd
8	S	700	4642	499	0	41	20	252	?	0:00	httpd
8	T	1507	27378	1	0	41	20	383	?	0:00	isql
8	S	0	1774	294	0	61	20	194	?	0:00	in.telne
8	S	700	4021	499	0	41	20	252	?	0:00	httpd
8	S	103	8483	8457	0	45	20	112	pts/5	0:00	more
8	T	3004	3782	2186	0	41	20	104	pts/23	0:00	a.out
8	S	3003	2527	2522	0	41	20	143	pts/2	0:00	csH

TESIS CON
FALLA DE ORIGEN

El reporte es largo debido a que el sistema tiene muchos procesos activos de varios usuarios. Desde el punto de vista de optimización del desempeño, los campos más importantes son:

UID	Identificación del usuario que creó el proceso (dueño del proceso).
PID	El número de identificación del proceso.
PPID	El número de identificación del proceso padre.
C	Porcentaje de utilización de CPU.
PRI	Prioridad de ejecución del proceso: números bajos indican prioridades altas.
NI	Número nice del proceso, empleado para definir la prioridad de ejecución.
SZ	Cantidad de memoria virtual (en páginas) requerida por el proceso
TIME	Cantidad total de tiempo de CPU usado por el proceso.
COMD	Comando actual.

Se puede obtener información del sistema observando los comandos que se están ejecutando. Se puede identificar los procesos que están consumiendo la mayoría de tiempo de CPU y memoria del sistema. Del reporte anterior se tiene los siguientes problemas:

- El sistema está ejecutando muchos procesos idénticos creados por el mismo usuario. Dependiendo de la situación, es conveniente terminar esos procesos usando el comando `kill`.

```
e5000% ps -el | grep 503
```

```
 8 S 503 627 1 0 40 20 278 ? 0:00 httpd
 8 S 503 653 627 0 41 20 282 ? 0:00 httpd
 8 S 503 654 627 0 41 20 282 ? 0:00 httpd
 8 S 503 655 627 0 41 20 282 ? 0:00 httpd
 8 S 503 656 627 0 41 20 282 ? 0:00 httpd
 8 S 503 657 627 0 41 20 282 ? 0:00 httpd
 8 S 503 658 627 0 41 20 282 ? 0:00 httpd
 8 S 503 659 627 0 41 20 281 ? 0:00 httpd
```

```
e5000# kill -9 627 653 654 655 656 657 658 659
```

- Hay un proceso que tiene acumulado una gran cantidad de tiempo de CPU. Esto puede indicar que el proceso se encuentra en un ciclo infinito o que algo esta mal.

```
e5000% ps -el |grep 145
```

```
8 S 145 26782 1 99 41 20 5986 ? 1259:51 .netscape
```

Como se puede ver este proceso esta gastando el 99% del CPU, por lo que es necesario eliminarlo, de lo contrario provocaría que el sistema se caiga.

```
e5000# kill -9 26782
```

- El sistema esta corriendo varios procesos grandes progresivamente, se puede sugerir que el usuario los ejecute a una prioridad baja (nice) o en la noche (at).
- En general, el administrador solo tiene que preocuparse por los trabajos que están en ejecución, que estén bloqueados por falta de algún recurso del sistema. Los procesos dormidos (S), que han sido detenidos (T) o zombies (Z) no afectan el rendimiento total del sistema.

```
e5000% ps -el |grep defunct
```

```
8 Z 3004 2186 2175 0 0 0:00 <defunct>
```

- El tamaño de la memoria (columna SZ) de algunos procesos en ejecución es muy grande, aquí se tiene que revisar los problemas de uso de memoria debido a que el proceso esta consumiendo un gran porcentaje de la memoria del sistema.

```
8 S 107 477 474 0 41 20 6561 ? 312:51 dataserv
8 S 5027 5060 5059 0 41 20 4054 ? 93:45 dataserv
8 S 145 26782 1 99 41 20 5986 ? 1259:51 .netscape
```

TESIS CON
FALLA DE ORIGEN

4.3.2 Estadísticas del CPU

El primer componente del desempeño del sistema a verificar, es la utilización de CPU. Esta es una de las áreas más fáciles de verificar. El reporte del comando `ps` es solo una imagen de la actividad del sistema y de lo que pasa con cada proceso en el momento en que se ejecuto, sin embargo, es necesario tener una idea de cómo esta el sistema en general. Hay varias formas de obtener un cuadro más general del uso de los recursos del sistema incluyendo la red. La herramienta `iostat` puede ayudar a determinar si el sistema está o no limitado en CPU. Proporciona una medición del tiempo `idle` del CPU (columna `id`).

e5000% iostat 5

tty	sd9			sd10			sd21			ssd0			cpu				
	tin	tout	kps	tps	serv	kps	tps	serv	kps	tps	serv	tps	serv	us	sy	wt	id
1	24	9	1	33	4	1	50	0	0	174	1	0	14	20	2	3	75
0	47	4	1	12	0	0	0	0	0	0	0	0	0	76	23	0	1
0	16	0	0	0	0	0	0	0	0	0	0	0	0	75	25	0	0
0	16	2	0	18	0	0	0	0	0	0	0	0	0	74	26	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	77	23	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	80	20	0	0
0	16	10	2	31	15	2	46	0	0	0	0	0	0	76	24	0	0
0	16	1	0	13	0	0	0	0	0	0	0	0	0	80	20	0	1
0	16	5	1	13	0	0	0	0	0	0	0	0	0	75	24	0	1
0	16	1	0	16	0	0	0	0	0	0	0	0	0	78	22	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	78	21	0	1
0	16	14	2	40	18	2	60	0	0	0	0	0	0	76	24	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	76	24	0	0
0	16	2	0	18	0	0	0	0	0	0	0	0	0	76	24	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	76	23	0	1
0	16	0	0	0	18	2	6	0	0	0	0	0	0	71	29	0	0
0	16	2	0	12	3	0	9	0	0	0	0	0	0	73	27	0	0

Este reporte muestra que el sistema esta gastando casi el total de tiempo CPU ejecutando. Con la revisión anterior mediante el comando `ps` podemos ver qué procesos están consumiendo los recursos de CPU. Esta estadísticas en combinación con la información de la carga promedio de trabajo, se obtiene un cuadro global de lo que esta pasando en el sistema. En este caso se observa que se tiene un cuello de botella de CPU ya que el tiempo ocioso es nulo.

El comando `sar`, también nos proporciona información general de la actividad del sistema completo:

```
e5000% sar 5 15
```

```
SunOS e5000 5.7 Generic_105181-06 sun4u 11/27/00
```

	%usr	%sys	%wio	%idle
15:37:17	72	28	0	0
15:37:22	70	30	0	0
15:37:27	70	30	0	0
15:37:32	70	30	0	0
15:37:37	70	30	0	0
15:37:42	68	32	0	0
15:37:47	70	30	0	0
15:37:52	72	27	0	0
15:37:57	73	27	0	0
15:38:02	71	29	0	0
15:38:07	69	31	0	0
15:38:12	69	30	0	0
15:38:17	69	31	0	0
15:38:22	68	32	0	0
15:38:28	67	32	0	0
15:38:33	70	30	0	0
Average	70	30	0	0

El reporte muestra que el sistema tiene una carga alta, el porcentaje de tiempo ocioso es bajo y además que el porcentaje más alto es la cantidad de tiempo gastado en la ejecución de procesos de los usuarios. Como el sistema esta limitado en CPU, entonces se procede a encontrar cuáles procesos o aplicaciones están dominando el uso de CPU. Esto se lleva a cabo ejecutando periódicamente el comando `ps`.

Una vez que el culpable es identificado se realizan los mismos cambios mencionados anteriormente, y posteriormente se analiza nuevamente la carga promedio del sistema, para ver si tuvieron efecto sobre el desempeño del sistema.

```
e5000% uptime
```

```
5:37pm up 31days, 2:16, 12 users, load average:0.80,0.55, 0.08
```


4.3.3 Estadísticas de la memoria

Si el sistema no está limitado en CPU, la siguiente opción es examinar el uso de memoria. Hay herramientas que reportan las estadísticas de paginación y swapping. Para UNIX, `vmstat`, proporciona la manera más exacta de determinar si hay problemas de la memoria. En cualquier herramienta que se use, se obtendrá información sobre:

- *Page-ins*, o páginas movidas de disco a memoria.
- *Page-outs*, o páginas movidas de memoria a disco.
- *Swap-ins*, o procesos movidos de disco a memoria.
- *Swap-outs*, o procesos movidos de memoria a disco.

Como ya se sabe, la memoria virtual es un método por medio del cual, la memoria física aparenta ser más grande que su tamaño real. El sistema de memoria virtual está compuesto de memoria real más espacio en disco, donde se almacenan las porciones de un archivo que no están actualmente en uso. El reporte de `vmstat` proporciona amplia información sobre la memoria virtual del sistema:

```
e5000% vmstat 5
```

procs			memory				page				disk				faults				...	
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	sg	sl	sd	sd	sd	sd	sd	sd	...
0	0	0	120	368	0	17	6	1	2	0	0	1	1	0	0
1	0	0	481368	5400	0	43	64	0	0	0	10	0	0	0	0
3	1	0	481368	5488	0	43	62	0	0	0	3	7	2	0	0
2	3	0	481368	3400	0	43	128	16	0	0	0	0	0	0	0
4	2	0	481368	5400	6	43	0	0	0	0	2	1	0	0	0
4	3	0	481368	3440	0	43	0	16	0	0	0	0	2	2	0
4	3	0	481368	2490	11	43	258	26	0	0	5	16	0	0	0
3	0	0	481368	4400	0	60	96	1	0	0	3	20	1	0	0
2	1	0	481368	5400	4	43	64	0	0	0	0	2	1	0	0
4	0	0	481368	4392	3	43	1	0	0	0	0	0	0	0	0
2	1	0	481368	4392	10	43	0	0	0	0	0	0	0	0	0
3	1	0	481368	3392	9	43	256	29	0	0	4	6	1	0	0
4	2	0	481368	3392	9	43	256	29	0	0	0	1	2	0	0
4	1	0	481368	3372	8	43	184	32	0	0	0	0	0	0	0
3	0	0	481368	3996	9	43	159	38	0	0	3	1	2	0	0
2	2	0	481368	3492	7	43	132	10	0	0	0	0	0	0	0
3	2	0	481368	3389	0	43	125	18	0	0	3	7	0	0	0
4	2	1	481368	5392	1	43	0	256	0	0	0	2	1	0	0
2	3	0	481368	1995	4	86	288	58	0	0	8	1	0	0	0

El reporte anterior tomado del sistema e5000, muestra que el sistema tiene una carga pesada. Observando las estadísticas anteriores se puede apreciar si el sistema esta realizando paginación o swapping, en este caso sucede constantemente pues hay varias páginas de procesos activos que están siendo movidos de memoria a disco y viceversa (columnas *pi* y *po*) y la cantidad de memoria física (columna *fre*) esta disminuyendo; por otro lado se puede ver que hay procesos bloqueados (columna *b*).

Otro hecho importante, es que también se observa que hay un proceso que ha sido movido completamente de memoria a disco (columna *w*), aquí se puede ver que la cantidad de memoria física disminuye drásticamente a casi 2 MB, lo cual quiere decir que hay actividad de swapping y que un proceso entero se movió a disco para recompensar la falta de memoria.

Esto se puede ver mejor ejecutando el comando `vmstat -S`, el cual desplegara las estadísticas de swapping directamente. La columna *so*, debe ser siempre 0 al menos que el sistema tenga un serio problema de memoria, lo cual sucede si la cantidad de memoria disponible no es suficiente para los procesos que se están ejecutando en el sistema.

Con el comando `sar -r` podemos visualizar cuanta memoria esta disponible:

```
e5000% sar -r 5 5
SunOS e5000 5.7 Generic_105181-06 sun4u    11/27/00
18:43:29 freemem freeswap
18:43:34      4909      984383
18:43:39      4829      988329
18:43:44      4909      984383
18:43:49      3983      986352
18:43:54      3779      986352
Average      4917      985958
```

La columna *freemem*, reporta cuanta memoria libre esta disponible en páginas. Cuando el sistema empieza a paginar la memoria libre disminuye. Aquí se puede apreciar que el número de páginas libres de memoria va disminuyendo por lo tanto se dice que el sistema esta paginando. Para ver las estadísticas de swapping se ejecuta el comando `sar -w`. Aquí se muestra el reporte tomado de servidor e5000:

```
e5000% sar -w 5 5
SunOS e5000 5.7 Generic_105181-06 sun4u      11/27/00
18:58:25 swpin/s  bswin/s  swpot/s  bswot/s  pswch/s
18:58:30  0.00    0.0    0.00    0.0    128
18:58:35  0.00    0.0    0.00    0.0    126
18:58:40  0.00    0.0    0.00    0.0    124
18:58:45  0.00    0.0    0.00    0.0    123
18:58:50  0.00    0.0    0.00    0.0    122
Average    0.00    0.0    0.00    0.0    124
```

Las columnas en este reporte muestran:

- `swpin/s`

Número promedio de transferencias a memoria por segundo durante el intervalo.

- `bswin/s`

Número promedio de bloques de 512 bytes transferidos a memoria por segundo.

- `swpot/s`

Número promedio de transferencias a disco por segundo durante el intervalo.

- `bswot/s`

Número promedio de bloques de 512 bytes transferidos a disco por segundo.

- `pswch/s`

Número de procesos movidos por segundo durante el intervalo.

Usando estos comandos nos damos una idea de cómo se mantiene la memoria del sistema. En este caso podemos ver que el sistema mantiene una actividad constante de paginación, sin embargo, en este intervalo de tiempo, esto no llega a ser un problema serio. Por el contrario, si el sistema se mantiene haciendo paginación o swapping constantemente en intervalos más largos, se puede tener problemas de memoria.

Para tener un cuadro más exacto sobre el desempeño de la actividad de paginación del sistema, se debe ejecutarse el comando `sar -g`:

```
e5000% sar -g 5 5
```

```
SunOS e5000 5.7 Generic_105181-06 sun4u 11/27/00
19:23:11 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
19:23:16 0.20 0.20 0.20 0.00 0.00
19:23:21 0.00 0.00 0.00 0.00 0.00
19:23:26 0.40 0.40 0.40 0.00 0.00
19:23:31 0.00 0.00 0.00 0.00 0.00
19:23:36 0.00 0.00 0.00 0.00 0.00
Average 0.12 0.12 0.12 0.00 0.00
```

Las columnas en este reporte son:

- `pgout/s`

Número promedio de peticiones de páginas movidas a disco por segundo durante el intervalo, es decir, el número de veces que el subsistema de memoria detecto que le hacia falta memoria.

- `ppgout/s`

Número promedio de páginas por segundo que fueron movidas a disco. Más de una página puede ser movida a disco en una simple petición por lo que este campo a veces es mayor que `pgout/s`.

- `pgfree/s`

Número promedio de páginas por segundo que fueron liberadas.

- `pgscan/s`

Número promedio de páginas revisadas para liberar memoria.

A continuación se obtiene un reporte que muestra los reportes anteriores juntos. Este reporte muestra cuánta memoria libre hay (-r), estadísticas de swapping (-w) y si el sistema tiene o no actividad de paginación (-g).

e5000% more sar-rwg.out

SunOS e5000	5.7	Generic_105181-06	sun4u	11/27/00			
18:09:39	freemem	freeswap	swpin/s	bswin/s	awpot/s	bswot/s	pswch/s
	pgout/s	ppgout/s	pgfree/s	pgscan/s	%ufa_ipf		
18:09:44	459	142091					
	0.00	0.0	0.00	0.0			406
	0.40	0.40	0.40	0.00			0.00
18:09:49	457	142091					
	0.00	0.0	0.00	0.0			410
	0.20	0.20	0.20	0.00			0.00
18:09:54	451	141789					
	0.00	0.0	0.00	0.0			373
	0.00	0.00	0.00	0.00			0.00
18:09:59	459	142406					
	0.00	0.0	0.00	0.0			433
	0.20	0.60	0.60	0.00			0.00
18:10:04	491	142457					
	0.00	0.0	0.00	0.0			417
	0.20	0.20	0.20	0.00			0.00
18:10:09	549	142938					
	0.00	0.0	0.00	0.0			486
	0.00	0.00	0.00	0.00			0.00
18:10:14	535	142556					
	0.00	0.0	0.00	0.0			462
	0.20	0.20	0.20	0.00			0.00
18:10:19	442	142268					
	0.00	0.0	0.00	0.0			451
	1.00	1.59	1.59	0.00			0.00
18:10:24	460	141875					
	0.00	0.0	0.00	0.0			475
	0.60	0.60	0.60	0.00			0.00
18:10:30	417	139012					
	0.00	0.0	0.00	0.0			479
	28.80	160.75	242.80	159.76			0.00
Average	472	141946					
Average	0.00	0.0	0.00	0.0			439
Average	3.18	16.56	24.83	16.09			0.00

TESIS CON
FALLA DE ORIGEN

Este reporte mantiene un cuadro interesante de estadísticas. Después de que es iniciado el muestreo, la memoria tenía 459 páginas de memoria libres y no hay actividad de swapping, en este punto la memoria empieza a incrementarse hasta alcanzar 549 páginas disponibles, una vez que se inician algunos procesos, la cantidad de memoria disminuye hasta 417 páginas, debido a esto se observa que el sistema empieza a paginar. La columna `pgfree/s` se incremento hasta 242.80, lo que quiere decir que aproximadamente 242 páginas fueron liberadas mediante el mecanismo de paginación. Una vez que los procesos terminan, la memoria disponible se incrementa nuevamente, por lo tanto, la actividad de paginación disminuye.

Con el comando `ps`, se puede saber cuáles procesos están utilizando la mayor cantidad de memoria, la columna `SZ` muestra el tamaño de algunos procesos activos en páginas:

```
8 S 1507 13108 1 0 41 20 3292 ? 3:17 .netscape
9 S 145 26782 1 99 41 20 5986 ? 1259:51 .netscape
8 S 145 11888 1 0 41 20 1968 ? 1:20 .netscape
8 S 107 477 474 0 41 20 6561 ? 312:51 dataserv
8 S 5027 5060 5059 0 41 20 4054 ? 93:45 dataserv
8 S 145 26782 1 99 41 20 5986 ? 1259:51 .netscape
```

Con esto podemos afirmar que el proceso que más consume recursos de memoria es la ejecución del navegador Netscape.

Es importante hacer notar que cuando un proceso hace referencia a una página de memoria virtual que esta en disco, la página referenciada deberá ser movida a memoria y probablemente una o más páginas deberán ser movidas hacia el disco, creando tráfico de E/S y retardando el progreso del proceso.

4.3.4 Estadísticas de E/S

El siguiente paso es verificar si no existe un cuello de botella ocasionado por la actividad de E/S en disco. En la mayoría de los sistemas, el desempeño del disco es el aspecto más importante y crítico dentro de la administración del desempeño del subsistema de E/S, ya que esto afecta muchos otros aspectos del desempeño del sistema, como el almacenamiento de datos, el intercambio de datos de disco a memoria y viceversa; y la cola de impresión. Un buen desempeño de disco incrementa el rendimiento de la memoria virtual y reduce el tiempo necesario para correr procesos que requieren muchos recursos del subsistema de E/S.

En la optimización del desempeño del subsistema de E/S se deben considerar dos factores básicos:

- Rendimiento total del disco del sistema, determinar la velocidad en la cual los procesos pueden transferir datos hacia o desde los discos, es decir, la velocidad en que los procesos pueden leer y escribir en varios discos simultáneamente con el propósito de satisfacer un requerimiento a una aplicación.
- Eficiencia del almacenamiento del disco.

Un cuello de botella en disco puede ser ocasionados por excesiva actividad de E/S debido a la actividad de paginación o E/S a archivos locales. Si no se tienen problemas con accesos a disco local, hay que verificar si hay mucha actividad de E/S a archivos remotos.

La herramienta primaria para monitorear la carga de trabajo del sistema de E/S, es el comando `iostat`, el cual produce medidas muy exactas del rendimiento, utilización, rangos de transferencia y tiempos de servicio de los dispositivos de E/S del sistema.

e5000% iostat 5 15

tty	fd0			sd3			sd6			sd15			cpu			
	kps	tps	serv	kps	tps	serv	kps	tps	serv	kps	tps	serv	us	sy	wt	id
..	0	0	0	3	0	73	0	67	0	13	1	57	2	3	33	63
..	0	0	0	0	0	15	0	66	0	35	6	12	2	5	36	57
..	0	0	0	0	0	0	0	66	0	0	0	0	0	0	33	66
..	0	0	0	0	0	0	0	67	0	2	0	16	0	0	33	66
..	0	0	0	0	0	0	0	67	0	1	0	17	0	1	33	66
..	0	0	0	0	0	0	0	66	0	0	0	0	0	0	33	66
..	0	0	0	0	0	0	0	67	0	0	0	0	0	1	33	66
..	0	0	0	17	2	73	0	66	0	21	3	80	4	8	33	55
..	0	0	0	0	0	0	0	67	0	0	0	0	0	0	33	66
..	0	0	0	0	0	0	0	67	0	1	0	22	0	0	33	66
..	0	0	0	0	0	19	0	67	0	0	0	0	0	0	33	66
..	0	0	0	0	0	0	0	67	0	0	0	0	0	1	33	66
..	0	0	0	0	0	0	0	66	0	0	0	0	0	0	33	66
..	0	0	0	0	0	0	0	67	0	0	0	0	0	1	33	66
..	0	0	0	18	3	78	0	65	0	16	2	72	0	2	33	65

Las dos primeras columnas del reporte están relacionadas a la actividad de la terminal. Lo siguientes cuatro grupos de columnas reportan información de los primeros cuatro discos del sistema. Para cada dispositivo, esas columnas detallan:

- kps Número de kilobytes transferidos por segundo.
- tps Operaciones transferidas por segundo.
- serv Tiempo promedio de servicio en milisegundos.

El grupo final de columnas reporta el uso de CPU:

- us Porcentaje de tiempo en estado usuario.
- sy Porcentaje de tiempo en estado sistema.
- wt Porcentaje de tiempo en estado de espera.
- id Porcentaje de tiempo ocioso.

El reporte indica que no hay una situación de limitación en CPU y el tiempo de espera (wt) es mayor al 20%, por lo que se ve que hay un problema con la actividad de E/S a disco. Esto es ocasionado debido a una carga desbalanceada de los discos. Para obtener un mejor desempeño de E/S, la mejor política es dividir la carga en los discos como sea posible.

Otro formato del comando `iostat` usa la opción `-D`. En este formato la salida es diferente y se enfoca en los rangos de transferencia de los dispositivos y su uso. El reporte es similar, para cada disco, se detalla:

- `rps` Operaciones leídas por segundo
- `wps` Operaciones de escritura por segundo
- `util` Uso promedio del dispositivo

```
e5000% iostat -D 5 8
```

fd0			sd3			sd6			sd15		
rps	wps	util	rps	wps	util	rps	wps	util	rps	wps	util
0	0	0.0	0	0	0.7	0	67	0.0	0	1	1.3
0	0	0.0	0	0	0.0	0	67	0.0	0	0	0.0
0	0	0.0	0	0	0.0	0	67	0.0	0	0	0.5
0	0	0.0	0	2	2.6	0	65	0.0	0	2	2.6
0	0	0.0	0	0	0.0	0	67	0.0	0	0	0.0
0	0	0.0	0	0	0.0	0	66	0.0	0	0	0.0
0	0	0.0	0	1	1.0	0	67	0.0	0	1	1.5
0	0	0.0	0	0	0.0	0	66	0.0	0	0	0.0

Un tercer formato emplea la opción `-x` para desplegar información detallada sobre los discos. En este formato, se despliega una línea por cada disco e identifica los siguientes indicadores de desempeño:

- `device` Información del dispositivo.
- `r/s` Operaciones de lectura transferidas por segundo.
- `w/s` Operaciones de escritura transferidas por segundo.
- `kr/s` Número de kilobytes leídos por segundo.
- `kw/s` Número de kilobytes escritos por segundo.
- `wait` Número promedio de comandos en la cola de espera en el dispositivo.
- `actv` Número promedio de comandos en proceso.
- `svc_t` Tiempo de servicio (tiempo que se tarda en atender un proceso, incluye el tiempo de espera).
- `%w` Porcentaje de tiempo en la cola de espera.
- `%b` Porcentaje de tiempo en que esta ocupado el dispositivo.

```
e5000% iostat -x 5 2
```

extended device statistics									
device	r/s	w/s	kr/s	kw/s	wait	actv	svc_t	%w	%b
fd0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd3	0.2	0.2	1.0	1.6	0.0	0.0	73.2	0	1
sd6	0.0	67.0	0.0	0.0	0.0	0.0	0.0	0	0
sd15	0.4	0.6	6.4	6.7	0.0	0.0	56.9	0	1
sd16	0.0	0.0	0.2	0.0	0.0	0.0	12.4	0	0
sd17	0.0	0.0	0.4	0.1	0.0	0.0	8.7	0	0
sd18	0.0	0.0	0.1	0.0	0.0	0.0	6.6	0	0
st1	0.0	0.9	0.0	4.2	0.0	0.0	18.8	0	2
nfs1	0.0	0.0	0.0	0.0	0.0	0.0	10.6	0	0

extended device statistics									
device	r/s	w/s	kr/s	kw/s	wait	actv	svc_t	%w	%b
fd0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd6	0.0	66.6	0.0	0.0	0.0	0.0	0.0	0	0
sd15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
sd18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
st1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
nfs1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0

Para identificar cualquier problema de desempeño, se tienen que monitorear los tiempos de servicio (Svc_t) y tiempo en que los dispositivos están ocupados (%b).

En general, los discos que han mantenido tiempos mayores a 50 milisegundos o que están ocupados en un porcentaje mayor al 30%, probablemente ocasionarán un cuello de botella. Eso indica que hay problemas de desempeño de los discos.

Cuando el tiempo de espera llega a ser excesivo, el problema no es el disco individual, sino el controlador SCSI de E/S a disco, en este caso se reduce el número de dispositivos en el controlador para mejorar el desempeño, es decir, si todos los discos están sobre un mismo SCSI, podría beneficiar el hecho de tener varios adaptadores SCSI. Debido a que los controladores SCSI pueden manejar múltiples transferencias simultaneas, ofrecen un buen desempeño y permiten rangos de transferencias más rápidos.

4.3.5 Estadísticas de la red

La administración del rendimiento de la red es el aspecto más difícil de la sintonización del desempeño. El desempeño de la red no solo depende del desempeño de las máquinas individuales que la componen, sino también de cualquier miembro de la red incluyendo los componentes físicos tales como cables y conectores, todos ellos pueden tener un impacto en el rendimiento de una red.

Los factores principales en el desempeño de la red son la latencia, ancho de banda y la utilización. La latencia es la cantidad de tiempo que se lleva un simple paquete de datos en viajar de un sistema a otro. Esta medida no se relaciona con la cantidad de datos enviados. Es una simple medida del índice de propagación de una señal desde un sistema a otro. La manera más efectiva de reducir la latencia en un ambiente altamente distribuido, podría involucrar la creación de sitios espejos o mejor conocidos como *mirror sites*².

El ancho de banda es un factor límite en el desempeño de la red. Es la medición de la cantidad máxima del tráfico de datos que pueden ser enviados a través de la red en un periodo de tiempo dado. Por ejemplo, en una Ethernet 10Base-T, la velocidad máxima teórica es de 10 Mbps, la cual equivale a 1250 Kbps. El problema es que esta velocidad máxima nunca podrá ser alcanzado. La sobrecarga, el medio de control de acceso y la congestión introducen retardos que disminuyen el índice de ancho de banda total efectivo.

La utilización es la medición de la cantidad de ancho de banda de la red que es usada. Así como el número de transferencias de datos sobre la red se incrementa, también la utilización de la red. El incremento en la utilización reduce la disponibilidad de la red, porque la red está ocupada. La pérdida de disponibilidad disminuye el índice de rendimiento total de la red. Los problemas de utilización pueden ser resueltos de dos formas: incrementar el ancho de banda y/o redistribuir la carga de trabajo de la red.

² Servidores que duplican el contenido de otro servidor en un punto remoto.

Hay herramientas para el sistema operativo Solaris que permiten identificar problemas de latencia y utilización. Para latencia comúnmente se usa el comando ping. Este comando determina la latencia en una red. La forma básica del comando es usada solo para saber si una máquina esta arriba, es decir, si esta activa. Manda paquetes especiales de datos a una máquina remota y contabiliza el tiempo que tarda el paquete en ser devuelto. En este caso la prueba se ha realizado desde la máquina e4000.

La versión más simple del comando ping envía un simple paquete al servidor en la red. Si el comando recibe la respuesta correcta, imprime el mensaje "host is alive".

```
e5000% ping e4000
e4000.dcaa.unam.mx is alive
```

Con la opción -s, se envía un paquete cada segundo y se produce un reporte después de que han sido enviados en el cual se detalla los tiempos mínimo, promedio y máximo en que tardan los paquetes en ser devueltos, es decir, el tiempo en que tarda en viajar la información, así como el porcentaje de paquetes perdidos.

```
e5000% ping -s e4000

PING e4000.dcaa.unam.mx (132.248.63.10): 56 data bytes
64 bytes from e4000.dcaa.unam.mx: icmp_seq=0. time=3. ms
64 bytes from e4000.dcaa.unam.mx: icmp_seq=5. time=3. ms
64 bytes from e4000.dcaa.unam.mx: icmp_seq=6. time=3. ms
64 bytes from e4000.dcaa.unam.mx: icmp_seq=7. time=3. ms
^C
----e4000.dcaa.unam.mx PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/2/4

e5000% ping -s e4000 4096

PING e4000.dcaa.unam.mx(132.248.63.10): 4096 data bytes
4104 bytes from e4000.dcaa.unam.mx: icmp_seq=0. time=22. ms
4104 bytes from e4000.dcaa.unam.mx: icmp_seq=7. time=14. ms
4104 bytes from e4000.dcaa.unam.mx: icmp_seq=8. time=14. ms
^C
----e4000.dcaa.unam.mx PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 14/15/22
```

Como se puede ver, se contacto con el equipo e4000 cuya dirección es 132.248.63.10. El tamaño del paquete por default es 64 bytes, sin embargo, para probar diferentes cargas de trabajo y tráfico, este valor puede ser modificado especificando el tamaño deseado como se muestra en el último reporte. Es importante notar que el tiempo de latencia puede incrementarse si la cantidad de datos transferidos también se incrementa. Cuando se indica un alto porcentaje de paquetes perdidos, esto indica un problema grave.

Para monitorear la actividad de la red, se tiene que ejecutar el comando `netstat`, esto permite obtener información del estado de la red y además muestra el número de colisiones y errores.

```
e5000% netstat -I
```

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
lo0	8232	loopback	localhost	4347284	0	4347284	0	0	0
hme2	1500	e5000	e5000	23218124	0	10138288	0	83368	0

En este reporte se obtiene detalles del estado de cada interfaz de la red. La primer línea despliega información desde que el sistema se inicializo. Para cada interfaz la siguiente información es desplegada:

mtu³ Unidad máxima de transmisión-tamaño máximo del paquete.
net/dest Red a la cual esta conectada la interfaz.
address Nombre o dirección de esta interfaz en Internet.
ipkts Número total de paquetes de entrada recibidos por esta interfaz.
ierrs Número de errores en la entrada de paquetes.
opkts Número de paquetes de salida enviados por esta interfaz.
oerrs Número de errores en la salida paquetes.
collis Número de colisiones.

TESIS CON
FALLA DE ORIGEN

³ Este valor difiere de acuerdo a la topología de red, para Ethernet el valor es 1500, para token-ring es 4089 y para módems es 576.

El número de entrada y salida de paquetes es útil para determinar la actividad de la interfaz. La columna `ierrs` representa un número de diferentes problemas: error de tamaño del paquete, errores de consistencia, interfaz o cable dañado. Este número debe ser pequeño, menos de 1% de los paquetes que están entrando. De lo contrario esto podría indicar que alguna corrupción de paquete esta ocurriendo en algún lugar de la red o el número de buffers de red es inadecuado. La columna `oerrs` representa el número de veces en que un paquete no ha podido ser transmitido. Este valor no debe ser mayor al 1% de los paquetes que están saliendo.

Para obtener un monitoreo en tiempo real se debe especificar un intervalo de tiempo:

```
e5000% netstat -i 5
```

input		hme2		output		input (Total)		output	
packets	errs	packets	errs	packets	colls	packets	errs	packets	errs
42589369	0	20820311	0	260748	56539788	0	34770730	0	260748
18	0	3	0	0	18	0	3	0	0
11	0	1	0	0	11	0	1	0	0
7	0	1	0	0	7	0	1	0	0
9	0	1	0	0	9	0	1	0	0
12	0	1	0	0	12	0	1	0	0
19	0	1	0	0	19	0	1	0	0

Con esto se obtiene estadísticas de la interfaz de red cada 5 seg. En nuestro caso solo tenemos una sola interfaz (`hme2`).

Se puede hacer uso del comando `spray` para probar la confiabilidad de los tamaños de los paquetes. El comando `spray` se usa para generar tráfico en la red. Se puede escoger un tamaño de paquete, el retardo entre paquetes y el número total de paquetes a ser transmitidos. Esto hace al comando, una herramienta útil para detectar si una red, junto con un servidor remoto, puede manejar grandes volúmenes de datos. También es usado para ver como una aplicación de red se comporta bajo una carga muy alta en la red simulando tráfico.

La forma más simple del comando `spray` es simplemente especificarle un nombre del algún servidor. El comando enviará continuamente un paquete de 86 bytes (tamaño por omisión) a la máquina remota:

```
e5000% spray e4000
sending 1162 packets of length 86 to e4000 ...
no packets dropped by e4000
10953 packets/sec, 942008 bytes/sec
```

Aquí, el campo clave es el número de paquetes perdidos (`packets dropped`). Idealmente debe ser 0. Para especificar el tamaño de un paquete se emplea la opción `-l`:

```
e5000% spray -l 1024 e4000
sending 97 packets of length 1026 to e4000 ...
no packets dropped by e4000
1175 packets/sec, 1206415 bytes/sec
```

```
e5000% spray -c 100 -d 20 0 -l 2048 e4000
sending 100 packets of length 2048 to e4000 ...
no packets dropped by e4000
579 packets/sec, 1186902 bytes/sec
```

En el último ejemplo se enviaron 100 paquetes al servidor `e4000` (`-c`) con un tamaño de 2048 bytes (`-l`) y un tiempo de retardo de 20 microsegundos (`-d`).

En estos ejemplos, el emisor de los paquetes es el servidor `e5000` y el servidor remoto es `e4000`. Como resultado, la pérdida de paquetes fue de 0 y esto no genero carga adicional para la red Ethernet.

4.4 REVISION DEL SISTEMA

Es claro que el servidor UNIX `e5000` ofrece mayor velocidad, confianza y muchas otras características; sin embargo, la misión central de este sistema es aprovechar sus recursos al máximo para obtener mayor desempeño para satisfacer las necesidades de los usuarios, así como la creciente demanda de aplicaciones.

Del análisis hecho a partir de las estadísticas anteriores, se observa que el sistema presenta frecuentemente problemas de CPU, una vez que se realizan algunos cambios manuales la situación mejora, sin embargo, cuando se inicia algún otro proceso pesado que requiere usar una parte considerable de los recursos del sistema, se vuelve a presentar nuevamente problemas de desempeño produciendo un cuello de botella de CPU lo cual a su vez en ocasiones genera una disminución en la memoria disponible en el sistema.

Como resultado de una reexaminación cuidadosa del análisis, se muestra que el subsistema de CPU muestra sobrecarga cuando se trabaja con bases de datos, se consultan sitios con muchos gráficos o se lleva a cabo los respaldos de datos debido a que se procesan grandes cantidades de datos.

En este caso de estudio los usuarios reportan que el tiempo de respuesta es ocasionalmente muy lento durante la jornada de trabajo. Esto lo toman como un desempeño inaceptable puesto que durante la jornada de trabajo normal, no había problemas de desempeño causado por aplicaciones que se corren en el sistema, por lo tanto, es necesario hacer unos ajustes al sistema.

4.5 RECOMENDACIONES

En base al análisis realizado anteriormente se proponen los siguientes cambios para mejorar el tiempo de respuesta del servidor, las cuales están basadas en las condiciones presentes en el momento de la recolección de datos:

- Monitorear todos los procesos que están corriendo actualmente, para eliminar aquellos que no son necesarios y que están empleando recursos en el sistema.
- Realizar una evaluación del desempeño del sistema, monitoreando los recursos del sistema en tiempo real, con las herramientas adecuadas que permitan notar problemas potenciales de rendimiento.

- Implementar programas o herramientas que nos permitan:
 - ✓ Obtener estadísticas de forma automática para tener un historial y registro de las mismas para su análisis posterior.
 - ✓ Obtener estadísticas gráficas del uso de los recursos del sistema para una visualización de manera rápida.
 - ✓ Administrar los procesos del sistema de forma automática.
 - ✓ Tener un control de acceso de los usuarios.
 - ✓ Implementar mensajes de advertencia y alarmas para los recursos que están próximos a saturarse y evitar de esta forma un impacto en el desempeño del sistema.
- Se recomienda hablar con los usuarios para hacerles notar la importancia en cuanto a las consideraciones pertinentes respecto al diseño y programación de aplicaciones.
- Correr los respaldo del servidor durante la noche.
- Optimizar las operaciones de Netscape Communicator y su uso de la red, para ello debe cumplir con los siguientes requerimientos:
 - ✓ 32 MB en RAM mínimo.
 - ✓ Asignar un tamaño adecuado a la memoria cache de Netscape Communicator.
 - ✓ Tener suficiente espacio en disco para que se almacenen fácilmente los registros de páginas visitadas, correos recibidos y enviados, y todas las actividades de los usuarios.
 - ✓ Tener suficientes ciclos de CPU para los procesos que demanda el uso de Netscape.
- Eliminar todos los archivos innecesarios del sistema para liberar espacio en memoria y así evitar hasta donde sea posible el uso de la paginación y swapping.
- Revisar el número de conexiones de red simultaneas, por default deben ser cuatro.

- Optimizar el desempeño del sistema de archivos.
- Monitorear las cuotas del sistema.
- Establecer límites en el número de archivos que pueden ser abiertos en un simple proceso a la vez. Para sitios muy ocupados, incrementar el número a 1024. En el archivo `/etc/system` modificar el parámetro `rlim_fd_max` y reinicializar el servidor.

4.6 IMPLEMENTACION DE HERRAMIENTAS ADICIONALES

En base al análisis previo de herramientas adicionales para supervisar y evaluar el rendimiento de un sistema, dos de las herramientas fueron añadidas al sistema e5000, Proctool y SE. Estas dos herramientas son de gran valor para el administrador del sistema, las cuales incluyen un agente y un administrador del rendimiento para entornos cliente/servidor, además de la facilidad de generar reportes de problemas de rendimiento. También permiten establecer umbrales de utilización escogiendo valores óptimos para identificar fácilmente cuando han sido rebasados los límites, generando así automáticamente alarmas cuando se exceden los umbrales de utilización de recursos. Esto permite actuar rápidamente y evitar la parada del sistema.

Con la implementación de ambas herramientas se puede obtener un análisis del desempeño en forma precisa y oportuna, la cual hace posible que se entienda de forma rápida la causa más común de los problemas de desempeño mostrando en forma gráfica los resultados; estos pueden estar basados en filtros y modos de ordenamiento definidos por el administrador, con ello se obtienen informes personalizados de acuerdo a las necesidades que se tengan. Esto nos ahorra tiempo en estar investigando cual es la fuente del problema, automatiza tareas tediosas de recolección de datos y reemplaza múltiples herramientas y procesos.

La primera tarea que Proctool hace es desplegar de forma gráfica estadísticas de uso de CPU y uso de memoria de todos los procesos que están corriendo en el sistema, reportando los procesos que consumen más recursos de CPU. Estos datos son evaluados regularmente para identificar y anticipar problemas de desempeño en Solaris.

Para la instalación de este programa:

- Bajar el archivo `proctool_1999_04.tar.Z` del sitio www.solaris4you.dk/monitoringSS.html en el directorio `/opt/proctool`.

- Ejecutar el comando:

```
% zcat proctool_1999_04.Z | tar xvf -
```

- Asegurarse de que `pmon` pueda ejecutarse como `root`:

```
% su root
passwd:
# chown root bin/pmon
# chmod u+s bin/pmon
# exit
% ls -l bin/pmon
-r-sr-xr-x 1 root staff 524288 Aug 16 2001 pmon*
```

Una vez que el paquete ha sido instalado, debe ser habilitado para comenzar la recolección de datos. Proctool es habilitado mediante la ejecución del comando `proctool&`. Puede ser ejecutado por un usuario con o sin privilegios de `root`. Las primeras estadísticas que se obtuvieron se muestran en la ventana principal (Fig. 4.2):

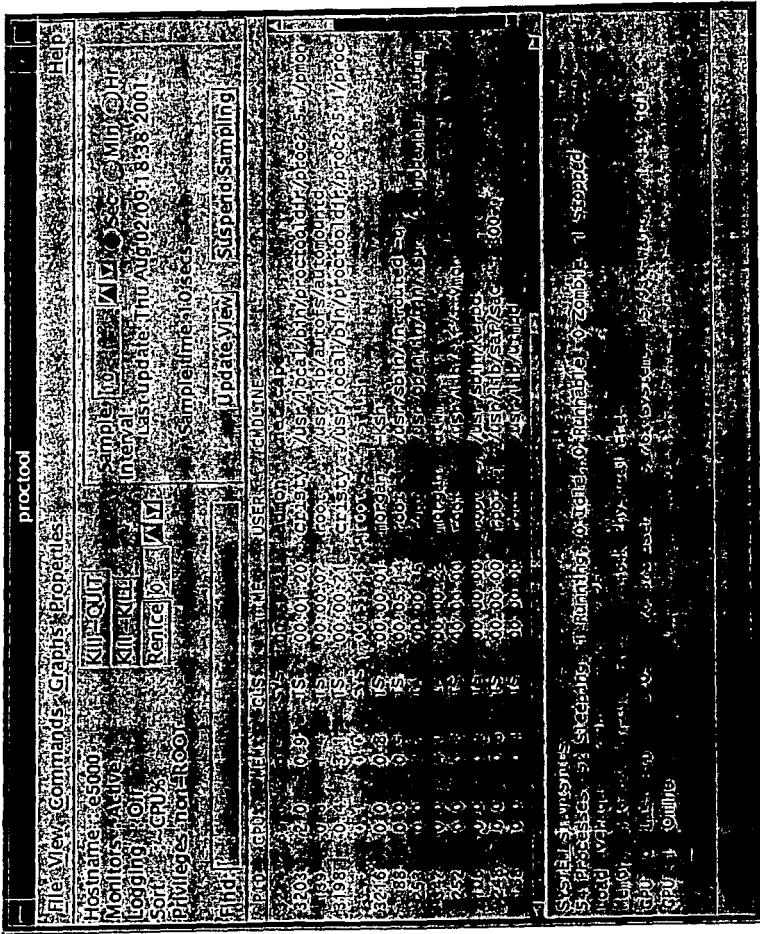


Fig. 4.2 Proctool en el servidor e5000.

TESIS CON
FALLA DE ORIGEN

Aquí vemos que el primer recurso evaluado es el CPU, se muestra que el proceso número 3234 (Netscape) del usuario *admon* esta consumiendo el 69% de CPU, lo cual retarda otras tareas en ejecución; además se muestra el promedio de la carga actual del sistema la cual se va incrementando según las estadísticas mostradas en la gráfica, así como información del hardware del sistema.

Con esto primer reporte se ve claramente porque se han venido presentando problemas de desempeño en el servidor, uno de los CPU esta apagado; lo cual ocasiona que toda la carga de trabajo sea procesada en un solo CPU originando un incremento en el uso del mismo y esto a su vez generando cuellos de botella en el sistema.

Esto se puede comprobar, mediante la ejecución el comando `psrinfo`, el cual muestra información más exacta sobre los procesadores del sistema, se emplea para examinar el estado administrativo de cada CPU:

```
e5000%psrinfo
0 on-line since 10/01/00 15:36:50
1 off-line since 02/26/01 18:37:58
```

El CPU 0 ha estado activo desde el día 1 de octubre del 2000 y CPU 1 se apago el día 26 de febrero del 2001.

El kit de herramientas SE además de obtener estadísticas de forma automática, ajusta los parámetros necesarios y sugiere recomendaciones en base a una serie de reglas establecidas para hacer que el sistema trabaje en forma óptima. Este kit nos proporciona un interprete para un lenguaje de programación (SymbEL) que es una combinación de C y C++. Su sintaxis es similar a C e incluye un conjunto de scripts que extraen datos del sistema para el análisis del desempeño del sistema. El interprete se encarga de tener acceso a estos datos y los presenta en forma gráfica.

Para la instalación de este programa:

- Bajar el archivo `RICHPse.tar.gz` del sitio www.sunfreeware.com o www.setoolkit.com.

- Situarse en el directorio donde se guardo el archivo y ejecutar:

```
e5000%gunzip RICHPse.tar.gz
```

```
e5000%tar xvf RICHPse.tar
```

Esto creara un directorio llamado `RICHPse`.

- Como root ejecutar:

```
e5000# pkgadd RICHPse
```

El archivo será instalado en `/opt/RICHPse`.

- Ir al directorio `DOC` y leer las instrucciones.

```
e5000% cd /opt/RICHPse/DOC
```

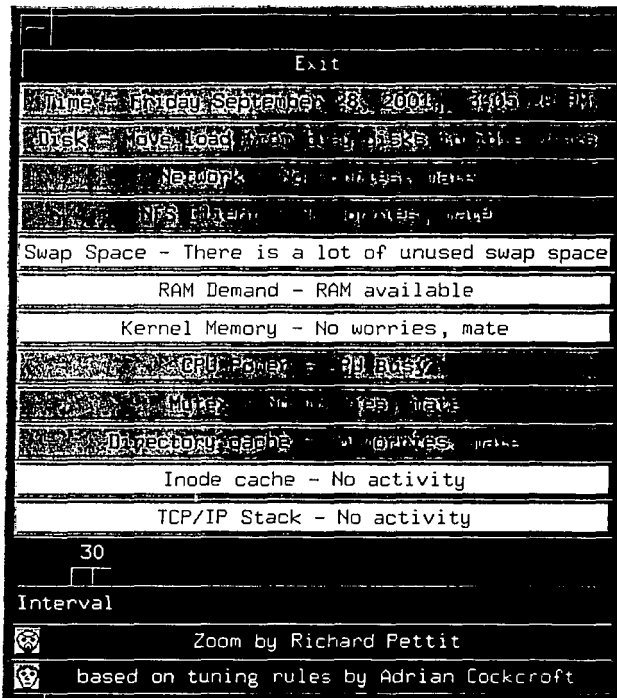
- Monitorear el sistema con:

```
e5000% /opt/RICHPse/bin/se zoom.se
```

```
o
```

```
e5000% /opt/RICHPse/bin/se toptool.se
```

Al ejecutar el primer comando de monitoreo, las barras de la pantalla (Fig. 4.3) cambian de color, indicando los diferentes niveles de carga que hay en el servidor, en el caso del CPU (color ámbar) indica que no hay suficientes ciclos de CPU lo cual reduce el desempeño total del sistema e incrementa los tiempos de respuesta.



TESIS CON
FALLA DE ORIGEN

Fig. 4.3 Zoom.

Ambas herramientas proporcionan una solución única y poderosa que permite al administrador del sistema detectar y corregir problemas de desempeño de una manera rápida y efectiva antes de que los usuarios finales sean afectados. Además de que permiten visualizar estas estadísticas mediante gráfica lineales o de barras, como se muestra en las figuras 4.4 y 4.5 las cuales despliegan las estadísticas de uso de CPU y actividad de paginación respectivamente.

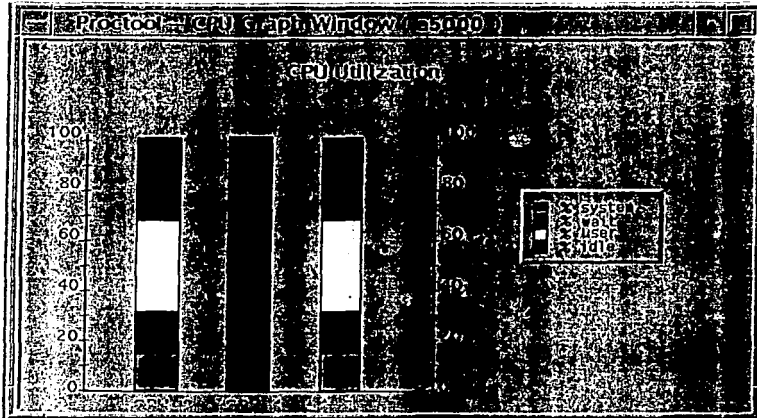


Fig. 4.4 Uso de CPU.

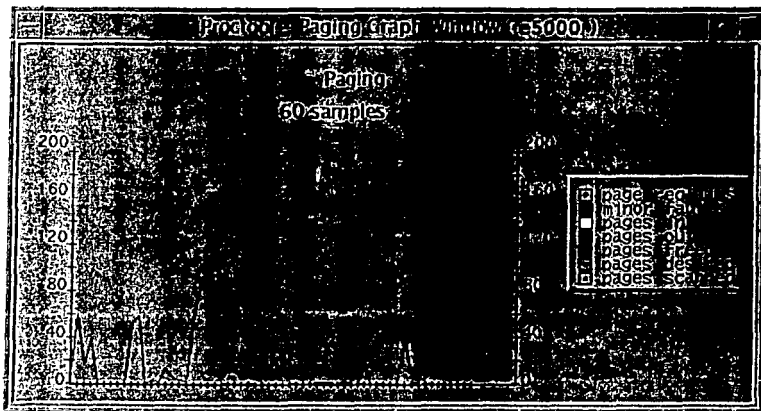


Fig. 4.5 Actividad de paginación.

TESIS CON
FALLA DE ORIGEN

4.7 OPTIMIZACION DEL SISTEMA DE ARCHIVOS

El comando `tunefs` cambia los parámetros dinámicos de un sistema de archivos. Todos los parámetros dinámicos están relacionados en la forma en que los datos son proyectados físicamente dentro del disco. Aunque el comando puede ser usado en un sistema en modo multiusuario, no puede ser usado en un sistema de archivos montado. Por lo tanto, antes de usar el comando `tunefs`, el sistema de archivos debe ser desmontado.

Los sistemas de archivos no deben ser optimizados al menos que el sistema de archivos sobrepase el 90% de su capacidad, en este caso debe ser optimizado para espacio.

Otra forma de ahorrar espacio, es imponer límites de uso de espacio en disco. En Solaris se puede controlar el espacio en disco a través del sistema de cuotas del disco. Las cuotas en disco activan los límites máximos para cada usuario: el número máximo de archivos que pueden ser creados y la cantidad máxima de espacio en disco que puede ser usado.

Cada cuota tiene dos valores: el límite duro y el límite suave. El límite suave sirve como un nivel de alerta, sin embargo se les asigna espacio adicional. Cuando este valor es excedido, el usuario recibe un mensaje de alerta para que libere espacio en un periodo de tiempo específico. Cuando este tiempo expira, el límite suave es tratado como un límite duro. Cuando el nivel del límite duro⁴ es alcanzado, el sistema detiene la asignación de recursos hasta que el nivel de uso se reduzca por debajo del límite suave.

Para verificar el espacio en disco se puede usar el comando `df -k`. A continuación se despliega la salida de este comando, la columna `capacity` es la que nos muestra el porcentaje total de uso de cada sistema de archivos.

⁴ Este límite no aplica a root.

```
%df -k
```

filesystem	kbytes	used	avail	capacity	mounted on
/dev/root	4610655	1653066	2957589	36%	/
/dev/stand	48195	5737	42457	12%	/stand
/dev/dsk/c0b0t0d0s4	3036285	208818	2827467	7%	/home
/tmp	81920	28	81892	1%	/tmp
/var/tmp	40960	0	40960	0%	/var/tmp
/dev/_tcp	0	0	0	0%	/dev/_tcp
/dev/dsk/c0b0t1d0s1	4433936	1503416	2930520	34%	/u1
/dev/dsk/c0b0t1d0s2	4441968	112280	4329688	3%	/u2
/dev/dsk/c0b0t2d0s1	4442112	945728	3496384	22%	/u3
/dev/dsk/c0b0t2d0s2	4443136	3468496	974640	79%	/u4
/dev/dsk/c0b0t3d0s1	4442112	157992	4284120	4%	/u5
/dev/dsk/c0b0t3d0s2	4443136	632104	3811032	15%	/u6
/dev/dsk/c0b0t4d0s1	4442112	176680	4265432	4%	/u7
/dev/dsk/c0b0t4d0s2	4443136	3473408	969728	79%	/u8

4.7.1 Activación de cuotas

Para implementar el sistema de cuotas en los diferentes sistemas de archivos, se tiene que realizar algunas tareas de administración:

- Modificar el archivo `/etc/rc2.d/S01MOUNTFSYS` para habilitar las cuotas a la hora de inicializar el sistema:

Las cuotas se activan agregando cuatro líneas al final del archivo:

1. `echo -n 'Checking quotas:' > /dev//console`
2. `/usr/sbin/quotacheck -a > /dev/console 2>&1`
3. `echo 'Enabling quota limits' > /dev/console`
4. `/usr/sbin/quotactl -a > /dev/console 2>&1`

- Habilitar las cuotas en un sistema de archivos individual en el archivo `/etc/vfstab` y crear el archivo `quota` en el directorio padre de cada sistema de archivos afectado:

Para cada sistema de archivos que tienen que ser protegido por cuotas, se tienen que realizar dos operaciones:

1. Agregar una opción a la entrada del sistema de archivos en el archivo `/etc/vfstab`. El cambio agrega el valor `rq` al campo `mount options` para cada sistema de archivos.
2. Después de modificar el archivo `/etc/vfstab`, agregar el archivo `quotas` al directorio padre del sistema de archivos afectado:

```
> cd /sistema_de_archivos
> touch quotas
> chmod 600 quotas
```

El primer comando nos cambia al directorio del sistema de archivos, el segundo crea un archivo `quotas` y el tercero cambia los permisos de lectura/escritura solo para `root`.

- Asignar los límites con el comando `edquota`:
Asignar cuotas involucra editar un archivo que define las cuotas para un usuario. En la mayoría de los casos, un archivo prototipo es creado y es usado como base para los usuarios reales del sistema:

```
% edquota usuario
```

La información desplegada consiste de una línea para cada sistema de archivos que tiene cuotas activadas. Por ejemplo:

```
fs /home blocks (soft=0, hard=0) inodes (soft=0, hard=0)
```

En este caso, para este usuario los límites en el sistema de archivos `/home` son 0 en todos los casos.

- Inicializar y activar los límites:

En este paso, los comandos agregados al archivo `/etc/rc2.d/S01MOUNTFSYS` son ejecutados manualmente para habilitar el chequeo de las cuotas:

1. `/usr/sbin/quotacheck -a`; para inicializar el sistema de cuotas en disco.
2. `/usr/sbin/quotactl -a`, para habilitar el monitoreo de cuotas⁵.

4.7.2 Monitoreo de cuotas

Monitorear las cuotas puede realizarse manualmente o como parte de un script. El comando `quota` despliega la cuota en disco de un usuario. Cuando se usa sin opciones, visualiza solamente alertas sobre los sistemas de archivos montados donde se han saturado los límites de espacio. Con la opción `-v` despliega la cuota del usuario actual en todos los sistemas de archivos montados donde existen cuotas habilitadas. Esta opción, permite al usuario `root` o al administrador del sistema desplegar la cuota de un usuario específico. A continuación se muestra la cuota en disco para el usuario `adminweb`, quien ya tiene lleno su espacio en `/export/home`:

```
# quota -v adminweb
Disk quotas for adminweb (uid 1000):

Filesystem      usage   quota   limit   files  quota   limit
/export/home    55848  55849   55849   2836   0       0
```

4.8 PRUEBAS DE SIMULACION

Hay una forma crucial que solo el administrador del sistema puede proporcionar: una prueba de simulación o *benchmark* que le dice si cualquier cambio particular en la configuración del sistema mejora o degrada el desempeño. El diseñar benchmarks apropiados es tan difícil como describir los síntomas de un mal desempeño.

⁵ Las cuotas pueden desactivarse por completo con el comando `quotaoff`.

El benchmark apropiado para cualquier situación depende de qué problemas tiene el sistema. Algunas ideas son:

- Cualquier benchmark que prueba el desempeño del sistema generalmente corre varias copias de uno o más trabajos en background (segundo plano). Esto es solo una manera práctica de simular una carga típica.
- Para medir la carga completa de E/S, escribir un programa en C que cree un archivo grande. La expresión $\text{tiempo-transcurrido}/\text{tiempo-total-cpu}$ es una medición de la carga de trabajo de E/S. Comparar los resultados cuando no hay carga con los resultados de una carga pesada, esto ayudara a determinar como esta trabajando el subsistema de E/S.
- Para el desempeño de E/S de un solo sistema de archivos, un benchmark que copie uno o más archivos grandes puede ser apropiado. Esta prueba simula un programa de aplicación grande e intensa actividad de E/S.
- Para simular el efecto de varios procesos simultáneamente usando diferentes discos, crear varios procesos que creen y copien archivos grandes en diferentes sistemas de archivos.
- Para el desempeño de la memoria virtual, un benchmark que ejecute varias compilaciones simultáneamente es apropiado.
- Al correr en combinación las pruebas descritas anteriormente puede ayudar a simular varios tipos de cargas de trabajo. Para simular la carga de trabajo del sistema se puede emplear varias aplicaciones usadas por los usuarios del sistema.

Aquí se muestra un simple programa en lenguaje C que crea un archivo grande haciendo repetidamente la llamada de sistema *write*. Se puede usar para medir el rendimiento total de un solo archivo o se puede agregar a un script más general que pruebe el sistema de E/S bajo cierta carga.

```
#include <fcntl.h>
#define FAIL -1
#define BSIZE 8192 /*escribe el tamaño del buffer*/
#define FSIZE 1000 /*número de veces que escribe el buffer*/
main()
{
  if (-1 == writeit(FSIZE)) printf("OPERACIÓN DE ARCHIVO FALLIDO")
}
int writeit(blocks)
int blocks;
{
  char buf[BSIZE];
  int i;
  int written;
  if ((fdes = open("./testfile.bm", O_WRONLY | O_CREAT, 0777) ) < 0)
    return FAIL;
  for (i=0; i<blocks; i++)
    {
      if (written = write(fdes,buf,BSIZE) !=BSIZE) return FAIL;
    }
  return 0;
}
```

El factor más importante en el diseño de un benchmark es que sea *repetible*. La manera más fácil de crear una benchmark repetible es escribir un shell script que automáticamente inicie trabajos de prueba y obtenga estadísticas del sistema. Se deben seleccionar comandos de uso común en el sistema que se tomen un largo tiempo y tienden a saturar el CPU.

No obstante, estas evaluaciones en ocasiones ofrecen poca información, es por ello que mientras más cerca se simule la carga actual del sistema, más relevantes serán los resultados. Definitivamente los mejores programas para usar en una prueba de rendimiento son las aplicaciones que comúnmente ejecutan los usuarios en dicho sistema. El concepto importante es que no hay que esperar a que el sistema este mal para obtener estadísticas. Con algo de inteligencia se puede hacer que el sistema entre en un estado conveniente.

4.9 RESULTADOS

Después de los cambios efectuados se ejecutaron nuevamente algunos comandos para verificar los resultados. En primer instancia se puede verificar una vez más el estado de cada CPU:

```
e5000%psrinfo
0 on-line since 10/01/00 15:36:50
1 on-line since 08/10/01 16:17:20
```

Con esto comprobamos que el problema del CPU dañado quedo resuelto y que nuevamente esta trabajando en condiciones normales.

Se procede a revisar la carga promedio actual del sistema, con lo cual podemos comprobar que el servidor muestra mayor potencia y flexibilidad para que los usuarios ejecuten sus aplicaciones sin ningún problema:

```
e5000% uptime
4:48pm up 47 days, 12 users, load average: 0.00, 0.04, 0.02
```

Una vez más podemos analizar las estadísticas actuales del CPU, para comprobar que estas han mejorado de manera significativa comparadas con las estadísticas anteriores. Estas estadísticas las podemos obtener directamente de las bitácoras que genera el programa Proctool:

Proctool log. Started on Mon Nov 19 11:50:49 2001

```
# CPU      cpu-id %user %kernel %wait %idle
# PAGING   reclaims minor-faults KB-paged-in KB-paged-out KB-freed KG-
#          desired pages-scanned
# VM       swap-avail free-physical-memory
# LOADAVG  avg1 avg2 avg3
# PROC ST  #-sleeping #-running #-idle #-runnable #-zombie #-stopped
```

```
TIME      Mon Nov 19 11:51:12 2001
CPU       0      1.29      1.09      0.00 97.61
CPU       1      1.99      1.99      0.00 96.02
PAGING    0 108 0 0 0 0 0
VM        1729912      598296
LOADAVG   0.27 0.11 0.08
PROC ST   105 1 0 0 0 1
```

```
TIME      Mon Nov 19 11:51:22 2001
CPU       0      0.20      1.19      0.00 98.61
CPU       1      0.30      1.09      0.00 98.61
PAGING    1 26 0 0 0 0 0
VM        1729904      598256
LOADAVG   0.23 0.11 0.07
PROC ST   104 1 0 0 0 1
```

```
TIME      Mon Nov 19 11:51:32 2001
CPU       0      0.40      5.67      0.89 93.04
CPU       1      0.50      1.29      0.00 98.21
PAGING    2 36 0 0 0 0 0
VM        1729936      598128
LOADAVG   0.20 0.11 0.07
PROC ST   104 1 0 0 0 1
```

```
TIME      Mon Nov 19 11:51:42 2001
CPU       0      2.29      0.50      0.00 97.22
CPU       1      0.89      0.99      0.00 98.11
PAGING    1 47 0 0 0 0 0
VM        1729936      597872
LOADAVG   0.19 0.11 0.08
PROC ST   104 1 0 0 0 1
```

```
TIME      Mon Nov 19 11:51:52 2001
CPU       0      2.88      1.39      0.50 95.23
CPU       1      0.60      0.70      0.00 98.71
PAGING    0 82 0 0 0 0 0
VM        1729928      597712
LOADAVG   0.18 0.11 0.08
PROC ST   104 1 0 0 0 1
```

En estas estadísticas podemos ver la carga de trabajo para cada CPU, las estadísticas de paginación, la cantidad de memoria disponible, la carga promedio del servidor y por último el status de los procesos del sistema.

Con estas estadísticas finales podemos ver que la carga de trabajo se reparte en ambos CPUs permitiendo así que los procesos en ejecución no tengan que esperar a que algún otro proceso u aplicación termine, lo cual permite una actividad más estable de ambos procesadores.

Este avance en términos de rendimiento se logró, en parte, utilizando el software Proctool y SE junto con las herramientas propias del sistema, lo cual nos permitió obtener el máximo nivel de rendimiento y disponibilidad del servidor. Los resultados obtenidos incluyen:

- Los usuarios obtuvieron tiempos de respuesta más rápidos.
- Los usuarios ya no obtuvieron mensajes "*connection refused*", lo cual indica una alta disponibilidad del servidor.
- El servidor e5000 mostró menos actividad del sistema de E/S aproximadamente en un 15%.
- Se encontró que el servidor e5000 mejora su desempeño entre 13 y 29% sobre los servidores que ejecutan sus respaldos durante el día.
- Aumento en la confiabilidad, ya que el monitoreo de los procesos del sistema con ayuda de estas herramientas nos permite tener un mayor control de acceso de manera activa de usuarios tanto internos como externos a la organización, consiguiendo así una protección contra usuarios no autorizados del sistema.

La obtención de estos resultados de rendimiento significa que los usuarios pueden tener acceso a la información de una manera más rápida, mejorar notablemente las operaciones de sus aplicaciones y a su vez el sistema soportar una mayor carga de trabajo sin que esto degrade notablemente su desempeño. Actualmente el servidor ofrece mayor rendimiento con capacidad de actualización y la más alta disponibilidad en entornos en los que se trabaja 7 días a la semana, 24 horas al día.

4.10 OTRAS CONSIDERACIONES

A parte de las recomendaciones especiales hay algunos factores de desempeño comunes para todos los sistemas operativos que deben tomarse en cuenta:

- Revisar de manera frecuente las bitácoras del sistema para detectar posibles fallas en cuanto al hardware para evitar futuros problemas de rendimiento. En este caso en particular, se hubiera descubierto a tiempo que había problemas con la temperatura de un CPU:

```
WARNING: CPU/Memory board 0 is warm (temperature: 73C). Please check
system cooling
```

```
WARNING: CPU/Memory board 0 is very hot (temperature: 83C)
```

```
WARNING: System shutdown scheduled in 20 seconds due to over-
temperature condition on CPU/Memory board 0
```

```
WARNING: CPU/Memory board 0 still too hot (temperature: 83C). Overtemp
shutdown started
```

```
NOTICE: Processor 0 powered off.
```

Esto indica que uno de los procesadores está en condiciones excesivas de temperatura, lo cual ocasionó que el CPU se apagará para que la temperatura se regule o vuelva a su estado normal.

- Asegurarse de tener instalado todos los parches recomendados para el sistema operativo. El propósito de esos parches es corregir algunos problemas que pueda afectar el desempeño del servidor o comprometer la seguridad del sistema.
- Mantenerse informado sobre los problemas y situaciones relacionadas con el sistema operativo y ver en que forma pueden afectar la operación del servidor.
- Dar mantenimiento preventivo y correctivo de manera regular para evitar una degradación en el desempeño del sistema.

Es importante saber donde obtener una amplia variedad de información sobre el sistema operativo de tal forma que se puedan encontrar soluciones para inevitables problemas anticipados.

4.11 PERSPECTIVAS A FUTURO

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 IBM y 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 han tenido que aprender a construir sus propias supercomputadoras conectando computadoras y desarrollando software para enfrentar tales problemas.

Por otro lado, dado que día con día los servidores tienen mayor poder de cómputo y las redes de comunicación mayor ancho de banda, se vuelve cada vez más atractivo y factible hacer uso de sistemas que ofrecen cómputo de alto rendimiento tales como los clusters que también emplean el procesamiento en paralelo.

Un cluster es una colección de nodos (máquinas físicas) que ejecutan sus propios procesos. Esos procesos se comunican uno con otro de tal forma que pareciera un solo sistema el que proporcionan un acceso continuo tanto a los recursos del sistema como a las aplicaciones y a los datos de los usuarios.

Un cluster ofrece varias ventajas sobre los servidores tradicionales. Esas ventajas incluyen tolerancia a fallas, servicios escalables, capacidad para un crecimiento modular, y un precio bajo comparado con sistemas que emplean el procesamiento en paralelo como las supercomputadoras. Un cluster asegura la disponibilidad de los datos y de las aplicaciones de los usuarios e incrementa el desempeño al habilitar nuevos servicios o al agregar nuevos procesadores.

El procesamiento en paralelo se refiere, al concepto de acelerar el tiempo de ejecución de un programa dividiéndolo en múltiples fragmentos que pueden ser ejecutados simultáneamente, cada uno en su propio procesador. Un programa que es ejecutado en N procesadores puede correr N veces más rápido que si corriera en uno solo.

En otras palabras, la disponibilidad, alta confiabilidad, el alto rendimiento y la eficiencia de los clusters, ha sido una gran fortuna y su popularidad crece cada día en los medios académicos.

CONCLUSIONES

Partiendo del diseño de hardware del servidor e5000 y de los servicios que ofrece se ha podido valorar que una actualización de CPU no es recomendada debido a que los CPUs tienen suficiente capacidad para el sistema en cuestión. Con esto se ha mostrado que no siempre es necesario adquirir nuevo hardware para lograr que un sistema mejore su desempeño.

Esto permitió evitar una de las inversiones a realizarse en dicho sistema y al mismo tiempo aumentar su capacidad de servicio garantizando que se mantenga a niveles aceptables, ambas características importantes considerando la falta de recursos de la Dirección General de Servicios de Cómputo Académico.

A lo largo de todo el trabajo expuesto, ha quedado clara la necesidad de incorporar herramientas adicionales al sistema si se pretende analizar de forma justa el rendimiento de un sistema UNIX.

El kit de herramientas SE (System Engineers) realiza un monitoreo de todo el sistema y puede realizar cambios inmediatos a algunos parámetros regulables del kernel sin necesidad de hacerlo manualmente. Proctool es una herramienta principalmente para la administración de procesos, que nos permite saber si algún recurso del sistema esta en problemas. La principal ventaja de estas herramientas es que son una muy buena opción para administrar el desempeño, pues permiten tener de manera rápida, consistente, confiable y oportuna un panorama del comportamiento del sistema.

Como se vio, antes de cualquier otro análisis, siempre se debe inspeccionar el sistema para ver que es lo que esta pasando. Esto redundará en una visión de lo que podría estar contribuyendo o causando el problema, siempre y cuando se tenga conocimiento de cómo trabaja el sistema y qué aplicaciones ejecutan los usuarios. Por esta razón, quizá las herramientas más importantes empleadas en la optimización del desempeño son la *observación* y el *sentido común*.

En base a las estadísticas obtenidas del servidor e5000 se puede concluir lo siguiente:

- Todas las tareas son restringidas por los recursos.
- Los problemas de rendimiento pueden aparecer rápidamente y con poco aviso.
- El hecho de que el tiempo ocioso sea casi nulo, generalmente es el primer síntoma de carga de CPU.
- Los comandos `iostat` y `vmstat` pueden ser el primer paso para buscar problemas de rendimiento. Con esto en mente, se puede seleccionar un comando específico para un recurso dado y echar un vistazo al desempeño del sistema.
- Todas las utilerías del sistema operativo pueden ser empleadas en conjunto como un instrumento de medición para obtener un cuadro de estadísticas más extenso sobre el uso del sistema, permitiendo así evaluar requerimientos de software y hardware.

- La memoria llega a ser un factor crítico en un sistema cuando los procesos requieren más memoria física de la que hay disponible. Debido a ello se implementa la memoria virtual la cual es posible a través del mecanismo de paginación y/o swapping para soportar procesos adicionales.
- El subsistema de red es el más complejo y en muchas formas el más difícil, pues además de la tarjeta de red, su software y su configuración; se debe considerar los dispositivos intermedios de la red local (rutadores, hubs y otros tipos de dispositivos de conexión), los protocolos de red usados, la arquitectura, las conexiones a redes externas y el rendimiento de la red externa en sí misma.

Con los resultados obtenidos se considera que con el uso de las herramientas propias del sistema en conjunto con el software para la gestión del desempeño dan una mejor evaluación del rendimiento de un sistema informático. Se ha demostrado que las herramientas adicionales ofrecen una poderosa herramienta de trabajo, por su cómodo diseño, además que son de fácil instalación y extremadamente intuitivos en su manejo. Desde un inicio, se desconocía la razón del problema de desempeño, dado que no se tenía ninguna estadística en cuanto al uso de los recursos del sistema. Sin embargo, el trabajo no fue en vano, ya que el objetivo se logró obteniendo una mejora sustancial en el desempeño. Considero que estos resultados sin ser espectaculares alientan a seguir trabajando sobre el tema.

Por otro lado, debemos saber que los problemas de desempeño no se limitan estrictamente al hardware. El sistema operativo y los programas de aplicación afectan de manera importante el desempeño total de un sistema. En algunos casos donde el rendimiento es menos que óptimo, el hardware está funcionando adecuadamente como está diseñado, sin embargo, las configuraciones del sistema operativo o el diseño de las aplicaciones y su implementación son realmente el problema. Aquí sería un grave error agregar más hardware para solucionar el problema de software. Primero habrá que encontrar y solucionar el problema de software para brindar un nivel aceptable de desempeño.

No podemos pasar por alto la revisión constante de las bitácoras, ya que esto es muy importante al permitirnos detectar a tiempo hasta un 90% de los posibles problemas que pudiesen presentarse en cualquier sistema UNIX.

Es importante mencionar que no existe una guía que nos indique paso a paso como detectar problemas de desempeño y de cómo sintonizar un sistema que presente estos problemas, los ejemplos mostrados aquí son solo una forma de hacerlo para garantizar que los resultados sean fiables. La metodología empleada aquí nos permite saber, de forma bastante precisa, el grado de complejidad para determinar problemas de rendimiento, ya que existen múltiples factores que pueden alterar en mayor o menor medida el desempeño de un sistema.

Siempre será prudente considerar factores como el crecimiento que pueden tener las aplicaciones, bases de datos y los componentes físicos de la red, puesto que estos también pueden afectar el desempeño de un sistema.

GLOSARIO

Archivo

Conjunto de datos con un nombre asociado que suele residir en dispositivos de almacenamiento secundario como discos, cintas o disquetes.

Buffer

Memoria intermedia para la transferencia de datos.

Cache

Una porción de la memoria reservada para almacenar temporalmente información leída en un disco.

Cluster

Agrupamiento de computadoras independientes, grupo, cúmulo.

Colección de computadoras independientes que se usa como un único sistema y que proporciona servicios de alta disponibilidad y desempeño. El software reparte los datos y la computación entre los nodos del cluster. Cuando un nodo falla, otros nodos proporcionan los servicios y los datos que proporcionaban el nodo fallido. Cuando se añade o repara un nodo, el software de cluster migra algunos datos y computación a ese nodo.

Cola (Queue)

Conjunto de paquetes en espera de ser procesados.

Conexión remota (Remote login)

Operación realizada en un sistema remoto a través de una red de computadoras, como si se tratase de una conexión local.

Demonio (Daemon)

Aplicación UNIX que está alerta permanentemente en un servidor Internet para realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página web.

Desconexión por tiempo (Time out)

Situación que se produce cuando un usuario es desconectado de la red por haber transcurrido un tiempo prefijado.

Direcciones de E/S

Posiciones dentro del espacio de direcciones de entrada/salida de la computadora que son utilizadas por un dispositivo, como puede ser una impresora o un módem. La dirección es utilizada para la comunicación entre el software y el dispositivo.

Directorio

Es un archivo especial que contiene información que permite encontrar otros archivos. Los directorios pueden contener a su vez, nuevos directorios, los cuales se denominan subdirectorios. A esta estructura de organización se le conoce como estructura de árbol.

E/S

Acónimo de 'Entrada/Salida'. Suele aplicarse al flujo de datos.

Espacio swap

En un disco duro, es un área usada para almacenar partes de programas en ejecución que han sido movidos de la memoria temporalmente para que otros procesos se ejecuten.

Ethernet

Especificación de red LAN de banda base que opera a 10 Mbps.

FDDI

Interfaz de datos distribuidos por fibra óptica que opera a 100 Mbps.

Hilo (Thread)

Hilo de ejecución en un programa o hilo de mensajes en una lista, hebra.

Interfaz (Interface)

Conexión mecánica o eléctrica que permite el intercambio de información entre dos dispositivos o sistemas. Habitualmente se refiere al 'software' y 'hardware' necesarios para unir dos elementos de proceso en un sistema o bien para describir los estándares recomendados para realizar dichas interconexiones.

Llamadas al sistema

Conjunto de funciones proporcionadas por el sistema operativo para manipular los archivos, procesos y dispositivos.

Modelo cliente-servidor

Modelo de comunicación entre computadoras conectados a una red en el cual hay uno, llamado servidor, que satisface las peticiones realizadas por el otro llamado cliente.

NFS (Network File System)

Sistema de archivos conectado a red, con el cual se tiene la impresión de que los archivos de una computadora remota forman parte del árbol de directorios de la computadora local.

Pérdida de paquetes (Packet loss)

Pérdida de alguna de las unidades de información, o paquetes, que componen un mensaje transmitido a través de Internet.

Programa

Conjunto de instrucciones y datos que se encuentran almacenados en un archivo ordinario. Cuando un programa es leído en el disco por el sistema operativo y cargado en memoria para ejecutarse se convierte en un proceso. A los procesos, el sistema operativo les asigna recursos para que puedan ejecutarse correctamente. Entre estos recursos están la memoria, CPU y los dispositivos de E/S.

Protocolo (Protocol)

Descripción formal de formatos de mensaje y de reglas que dos computadoras deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina-a-máquina o intercambios de alto nivel entre programas de asignación de recursos.

RPC

Llamadas de procedimientos remotos.

TESIS CON
FALLA DE ORIGEN

Scheduler

Proceso permanente que se encarga de asignar prioridades a un proceso.

Script (shell script)

Archivo de texto que contiene una lista de comandos, se utiliza para automatizar tareas muy habituales como, por ejemplo, la conexión a la red (*login*).

Servidor (Server)

Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la Red.

Shell

Programa que se encarga de interpretar todas las ordenes ejecutadas por el usuario. El shell indica que esta presente, esperando nuestras ordenes mostrando una marca o prompt. Este prompt es por defecto el caracter \$, en el caso del Bourne shell (sh), Korn shell (ksh) y en Cshell (csh) es el caracter %. El shell es la interfaz entre el usuario y el Sistema Operativo.

Sistema de Archivos

Parte del sistema responsable de la administración de los datos en dispositivos de almacenamiento secundarios, normalmente discos o particiones de disco. En UNIX los archivos están organizados en directorios.

Swap

Proceso de mover temporalmente un proceso de memoria a disco duro, en un área especial denominada espacio swap, de manera que otro proceso pueda usar ese espacio de memoria. Cuando el espacio llega a ser nuevamente disponible, el proceso regresa a memoria. Esto permite que otros procesos sean cargados y ejecutados simultáneamente independientemente del espacio de memoria física.

Tiempo de respuesta (Response time)

Lapso de tiempo que transcurre entre que un usuario hace una petición a la red y la información pedida es recibida por éste. En Internet depende de múltiples factores, tales como ancho de banda, calidad de la línea, velocidad de módem, congestión de la red. Por definición un usuario nunca está satisfecho con el tiempo de respuesta de la red y se acostumbra rápidamente a las mejoras de éste.

BIBLIOGRAFÍA

AIX/6000, Análisis del Rendimiento.
Manual del Estudiante.
IBM Education, IBM de México S.A.

Essential System Administration.
A Eleen Frisch.
O'Reilly & Associates, Inc.
2nd Edition, 1995.

High Performance Computing.
Kevin Dowd, Charles Severance.
O'Reilly & Associates, Inc.
2nd Edition, July 1998.

High Performance Computing, Challenges For Future Systems.
Davis J. Kuck.
Oxford University Press.
USA, 1996.

HP-UX Tuning and Performance: Concept, Tools and Methods.
Robert F. Sauers, Peter S. Weygant.
Prentice Hall.
July 1999.

TESIS CON
FALLA DE ORIGEN

Managing Internet Information Services.

Cricket Liu, Jerry Peck, Russ Jones, Bryan Buus & Adrian Nye.
O'Reilly & Associates, Inc.
First Edition, 1994.

Optimizing UNIX for Performance.

Amir H. Majidimehr.
Prentice Hall.
October 1995.

Sistema Operativo Unix, Guía del usuario.

Rebecca Thomas PhD, Jean Yate.
Ed. McGraw-Hill.
México, 1985.

Solaris 7 Performance Administration Tools.

Frank H. Cervone.
McGraw-Hill.
USA, 2000.

Solaris 8 Administrator's Guide.

Paul Watters.
O'Reilly & Associates, Inc.
January 2002.

Solaris Internals

Jim Mauro, Richard McDougall.
Prentice Hall.
Nov. 2000.

Solaris™ Solutions for System Administrators.

Time-Saving Tips, Techniques, and Workarounds.
Sandra Henry-Stockner, Evan R. Marks.
WILEY.
March 2000.

Sun Performance and Tuning.

Adrian Cockcroft, Richard Pettit.
Sun Microsystems Press.
2nd Edition, 1999.

System Performance Tuning.

Gian-Paolo D. Musumeci, Mike Loukides.
O'Reilly & Associates, Inc.
2nd Edition, 2002.

TESIS CON
FALLA DE ORIGEN

The Whole Internet, User's Guide & Catalog.

E. Krol.

O'Reilly & Associates, Inc.

2nd Edition, 1992.

UNIX Performance Tuning.

Sys Admin-Essential Reference Series.

By Sys Admin Magazine.

R & D Books / April 1997.

Web Performance Tuning . Speeding Up the Web.

Patrick Killelea.

O'Reilly & Associates, Inc.

October 1998.

TESIS CON
FALLA DE ORIGEN

RECURSOS ELECTRÓNICOS

MENSAJES ELECTRONICOS

- *Walter Nielsen*, [electronic mail], Sun Engineer, walter@west.sun.com.
- *Morgan Herrington*, [electronic mail], Market Development Engineering, Sun Microsystems, morgan.herrington@west.sun.com.

PERFORMANCE TOOLS

- *Proctool*, [online], 18/Jul/2001, 2:31, 1.4M, GZIP compressed document, disponible en: <http://www.ibiblio.org/pub/packages/solaris/freeware/SOURCES/PROCTOOL/>.
- *Proctool*, [online], Comsats Internet Services, 1999, README, disponible en: http://download.comsats.net.pk/unix/proctool1999_04.htm.
- *SymbEL, Sun Performance*, [online], Sun Microsystems, Inc., 1994-2002, disponible en: <http://www.sun.com/sun-on-net/performance/se3>.
- *The SE Performance Toolkit*, [online], Richard Pettit, Adrian Cockcroft, 1999-2001, disponible en <http://www.setoolkit.com>.

TESIS CON
PALA DE ORIGEN

SYSTEM PERFORMANCE

- *Tuning your Site: performance Issues and Actions*, [online], Sven Sjoberg, Web Site Team, Netscape Communications Corporation, disponible en: http://developer.netscape.com/docs/presentations/archivedconf/fall96/trc_8.pdf.
- *UNIX System Performance Tuning*, [online], Computation Center Newsletter, Vance Strickland, UNIX Services, 1996, disponible en: <http://www.utexas.edu/cc/newsletter/mar96/unixperf1.html>
- *Performance and Tuning of the UNIX Operating System*, [online], Petr Zemánek, 2000, disponible en: <http://cs.felk.cvut.cz/~zemanekp/datasem/DATASEM.HTM>.
- *HP-UX Tuning Guide*, [online], Stephen Ciullo, Daniel Hinojosa, June 1, 1999, disponible en: http://www.hp-partners.com/edaweb_public/html/technical_support/tuning.html.
- *Performance Guide*, [online library], SCO Handbook, The Santa Cruz Operation Inc., 1983-1995, disponible en: <http://osr5doc.ca.caldera.com:457/PERFORM/CONTENTS.html>.
- *Improving Compilation Times: A Case Study in Performance Analysis*, [online], Gian-Paolo D. Musumeci, Feb. 21, 2002, disponible en: <http://www.onlamp.com/pub/a/onlamp/2002/02/21/sysperf.html>.
- *Sun Performance*, [online], Collection of performance related information and links, Sun Microsystems, 1994-2002, disponible en: <http://www.sun.com/sun-on-net/performance.html>.

UNIX ADMINISTRATION

- *Unix system Administration*, [online], Frank G. Fiamingo, University Technology Services, The Ohio State University, 1996, disponible en: http://wks.uts.ohio-state.edu/sysadm_course/html/sysadm-1.html.
- *Anatomy of a Unix System*, [online], The Trustees of Indiana University, 1997, disponible en: <http://www.uwsg.indiana.edu/edcert/session1/concepts/system.html>.

TESIS CON
FALLA DE ORIGEN