

Ingeniería de Servidores

Héctor Pomares Cintas (hector@ugr.es)

Para tutorías, mandar e-mail o mensaje vía SWAD

1. Introducción a la Ingeniería de Servidores

Concepto de servidor

Un servidor es un caso específico de un sistema informático ; es decir, el hardware (lo físico, cables y componentes) ; el software (lo lógico, el SO y las aplicaciones que gestionan los recursos hardware) ; y el peopleware (los recursos humanos que intervienen)

Se clasifican según su uso! Pueden ser

De uso general como los PC , que los usa un usuario para ejecutar diversas aplicaciones

De uso específico como servidores o sistemas empotrados

► **Sistemas empotrados** Sistemas informáticos acoplados a un dispositivo o aparato que se diseñan para realizar una o varias funciones dedicadas con restricciones de tamaño y tiempo de respuesta, coste y consumo . Se componen de un microprocesador, memoria e interfaces de comunicación . Ej's Coche, Cam. Vigilancia.

► **Servidores** Sistema informático que , formando parte de una red , proporciona servicios a otros sistemas informáticos (clientes) . No es precisamente un computador dedicado especial , un portátil puede hacer de servidor , o un clúster de computadoras .

Tipos de servidores

Servidor web Almacena webs estáticas y las distribuyen a los clientes

Servidor de archivos Almacena archivos para acceso remoto

Servidor de BD Almacenan bases de datos a otros PCs o program

Servidor de e-commerce Almacenan webs dinámicas

Servidor de correo electrónico Almacenan e-mails de clientes

Servidor de impresión Controla las impresoras de los clientes

Servidor DHCP Proporciona una dirección IP dinámica a un cliente

Server DNS (Domain Name System) Resuelve nombres de webs y las redirecciona al servidor del dominio

Server proxy Servidor intermedio en las peticiones de recursos de un cliente a otro servidor.

Clasificación de SI según arquitectura de servicio -

► Sistema aislado Sistema que no interactúa con otros, donde no existe la distribución de la información.

► Sistema cliente / servidor En este sistema varios clientes reciben recursos de varios servidores que reciben sus solicitudes. El cliente no sabe que está interactuando con otros clientes.

► Sistema cliente / servidor de varios niveles. Similar al anterior pero el servidor se divide en varios niveles con funcionalidades diferentes en cada nivel. Gracias a ello se mejora la distribución de carga entre los diversos servidores ofreciendo más escalabilidad; pero pone más carga de red y es más difícil de programar y administrar. No necesariamente se necesitan más infraestructura física, pueden comunicarse varios MV o contenidores.

► Arquitectura cliente -cola -cliente habilita a todos los clientes para desempeñar tareas semejantes interactuando cooperativamente para hacer una actividad distribuida, y el servidor actúa de cola para capturar las peticiones y sinergizar el funcionamiento del sistema.

Ej: BitTorrent, Skype.
(los clientes comparten trozos de un archivo a otros)

Es importante que el cliente sepa que hay otros clientes interactuando con él.

Fundamentos de Ingeniería de Servidores

Para montar el servidor, se toman decisiones teniendo en cuenta el hardware, los recursos físicos, las aplicaciones que se van a usar...

Se resumen en los siguientes requisitos funcionales:

► Prestaciones / Rendimiento (Performance)

Miden la velocidad con la que se realiza una carga de trabajo. Se mide con el tiempo de respuesta o latencia, que mide el tiempo desde que se solicita hasta que se obtiene resultado. Cuanto más pequeño mejor.

También se mide la productividad o ancho de banda realizada por el servidor o un componente del mismo por unidad de tiempo. Cuanto más grande mejor.

El tiempo de respuesta inicial se denota R_0 ; y la productividad inicial, X_0 . Al llegar a cierto punto el servidor se saturará, y es trabajo del ingeniero

aumentar la productividad y reducir saturaciones.

Las prestaciones pueden ser afectadas bien por fallas de hardware (los cuales pueden reemplazarse o cambiarse reemplazando o añadiendo componentes); por el sistema operativo y sus planificaciones y configuraciones mal optimizadas; o por aplicaciones que tienen menor optimización que pueden ajustarse y optimizarse por permisos del SO o optimizando programas.

► **Disponibilidad (Availability)** Permite la operatividad del servidor. Se puede medir normalmente según el tiempo en el que el servidor no está disponible (downtime). Este puede ser planificado por una actualización que requiere un rearrange; o no planificado surgido de un fallo físico anomalia o efecto físico. Una alta disponibilidad requiere que el sistema sea tolerante a fallos (como un SO modular que requiere reinicios de partes del sistema sin reiniciar). También;

Podemos insertar componentes "en caliente" (sin necesidad de rearrange) (hot plugging); o de componentes (hot swapping).

También podemos redundar los discos (RAID), la alimentación (a través de más servicios de electricidad o SAT), de red (que tienen cable y WiFi para cubrir peticiones), y en general, la corja (load balancing).

► **Fiabilidad (Reliability)** Permite desarrollar su actividad sin presencia de errores. Se mide por el MTTF o tiempo medio que tiene un sistema hasta que ocurre un error.

Para solucionarlo se puede hacer uso de somas de comprobación para detectar y/o corregir errores (ECC, MDS), comprobar la recepción de paquetes de red y retransmitirlos (UDP, TCP).

► **Seguridad** Un servidor debe ser seguro ante la invasión de individuos no autorizados (confidencialidad) corrupción o alteración no autorizada de datos (integridad), e interferencias o ataques que impiden el acceso a los recursos (como en los casos de ataque DOS). Como soluciones tenemos una autenticación segura, encriptar los datos, antivirus, cortafuegos y parches de seguridad.

► **Extensibilidad** Facilidad que ofrece el sistema para aumentar sus recursos, como tener bahías libres para añadir mas memoria, usar SO de código abierto (modulares), usar interfaces de Entrada/Salida extenderlo con el objetivo de aumentar la escalabilidad o facilidad de ofrecer al sistema para poder aumentar de forma significativa sus características o recursos de cara a un aumento significativo de la carga.

Una solución se basa en la virtualización donde un hipervisor controla los recursos hardware al los que los sistemas operativos quieren usar, de forma que trabaja ellos con un subconjunto del hardware.

La escalabilidad puede ser vertical (donde se aumentan las prestaciones ~~sobre~~^{en} la misma máquina física, como en cloud computing) u horizontal (aumentando el nº

de servidores atendiendo una aplicación)

- **Mantenimiento** Refiere a todas las acciones que tienen como objetivo prolongar el funcionamiento correcto del sistema. Es importante que el servidor sea fácil de mantener. Es conveniente usar SOs con actualizaciones automáticas, cloud computing y automatizar las copias de seguridad (backup) y las tareas de configuración y administración (Ansible, Chef, Puppet)
- **Coste** El diseño se debe ajustar al presupuesto donde no sólo se tiene en cuenta el hardware y el software sino también otros costes como mantenimiento, personal, alquiler del local del servidor, consumo eléctrico... Hay soluciones como el cloud computing, software de código abierto y reducir costes de electricidad, bien a través del consumo de potencia de los componentes electrónicos según la carga; o con free cooling, usando bajas temperaturas en el exterior para refrigerarse gratuitamente.

Introducción a la comparación de prestaciones entre Sistemas Operativos

El computador con mejores prestaciones es aquel que ejecuta un conjunto de programas en el tiempo más corto.

Sea t_A el tiempo de ejecución en un programa en la máquina A, y t_B en la máquina B.

Cambio absoluto $\frac{t_A}{t_B}$

Cambio relativo de t_A sobre t_B

$$t_A = t_B + \frac{\Delta t_{A,B} (\%)}{100} \cdot t_B$$

$$\left(\Delta t_{A,B} = \frac{t_A - t_B}{t_B} \cdot 100 \right)$$

En el lenguaje común podremos ver:

" t_A es un 100% mayor que t_B " (Cambio relativo)
 " t_A es 2 veces mayor que t_B " (Cambio absoluto,
 " 1 vez mayor quiere decir "iguales")

Al hacer comparaciones, se miden las velocidades, definidas como la inversa proporcional a los tiempos.
 $(V = \text{Cálculo realizado} / \text{Tiempo})$

A esa división de las velocidades se denomina ganancia de velocidad o speedup

$$S_B(A) = \frac{V_A}{V_B} = \frac{t_B}{t_A}$$

Cambio relativo de velocidad

$$(S_B(A) - 1) \cdot 100$$

Relación prestaciones /coste La división de ambas:

$$\frac{\text{Prestaciones}}{\text{Coste}} = \frac{V}{\text{Coste}} = \frac{\frac{1}{f}}{\text{Coste}} = \frac{1}{f \cdot \text{Coste}}$$

Límites en la mejora del tiempo de respuesta, La ley de Amdahl

La mejora del tiempo de respuesta es limitada y hasta se puede hacer al reemplazar un componente del S/I

por otro más rápido.

(T ₀)	Toriginal	(1-f)T ₀	fT ₀
Recurso no usado			Recurso usado
Tiempo total			

Un computador puede que al ejecutar el programa una parte del tiempo no use el recurso, y otra parte sí. Si mejoramos esa parte, notaremos la mejora en los computos donde lo necesite. Si ese recurso se mejora K veces, el límite de mejorar se medirá con la ley de Amdahl

$$S = S_{\text{original}} (\text{modificado}) = \frac{V_m}{V_0} = \frac{T_0}{f_m} = \frac{(1-f)T_0 + fT_0}{(1-f)T_0 + fT_0} = \frac{(1-f)T_0 + fT_0}{K((1-f)T_0 + fT_0)} = \frac{1}{K(1-f) + f}$$

$$\frac{1}{K(1-f) + f}$$

Por esa ley:

Si $f=0$, porque $S=1$, por lo que no hay mejoras en el sistema.

Si $f=1$, es porque $S=K$, por lo que se mejora tanto como el componente

$$\text{Si } K \rightarrow \infty, S \rightarrow \lim_{K \rightarrow \infty} S = \frac{1}{1-f}$$

► **Ejemplo** La utilización de un disco duro es del 60% para un programa monohebra donde ejecutada sobre un determinado SI.

a) ¿Cuál sería la máxima ganancia en velocidad si se duplica la velocidad del disco?

b) ¿Cuál es la ganancia máxima actuando sólo sobre los discos?

0'4T ₀	0'6 · T ₀	T ₀
Resto	K=2	Se duplica la velocidad del disco
0'4T ₀	0'6 · T ₀	T _m

$$S = \frac{V_m}{V_0} = \frac{T_0}{T_m} = \frac{T_0}{\frac{0'4T_0 + 0'6T_0}{2}} = \frac{T_0}{1'0 \cdot (0'4 + 0'3)} = \frac{1}{0'7} = \underline{\underline{1'43}}$$

b) Son igual (K → ∞)

$$= \lim_{K \rightarrow \infty} \frac{1}{0'4 + \frac{0'6}{K}} = \frac{1}{0'4} = \underline{\underline{2'5}}$$

Si quisieramos generalizarlo para n mejoras:

$$S = \frac{1}{(1 - \sum_{i=1}^n f_{ci}) + \sum_{i=1}^n f_{ci} / K_i}$$

Ejemplo 4] Un computador tarda 100 segundos en ejecutar un programa de simulación de una red de interconexión entre multicomputadores. El programa tarda el 20% en operaciones de Áritmética Entera, el 30% en operaciones en coma Flotante, mientras que el resto se emplea en E/S. Calcula la ganancia en velocidad y el tiempo de ejecución si las operaciones aritméticas enteras y reales se mejoran de manera simultánea 2 y 3 veces respectivamente.

E/S 50%	AE 20% = 20s	CF 30% = 30s	$T_0 = 100s$
	K _{AE} =2	K _{CF} =3	
E/S 50s	AE 10% = 10s	CF 10% = 10s	$T_m = 70s$

$$f_{E/S} \cdot T_0 = (1 - (f_{AE} + f_{CF})) \cdot T_0$$

$$f_{E/S} \cdot T_0 = (1 - (f_{AE} + f_{CF})) \cdot T_0 \Rightarrow f_{E/S} = 1 - (f_{AE} + f_{CF})$$

$$\rightarrow f_{E/S} = 1 - (0'2 + 0'3) = 0'5$$

$$K_{AE} T_{m(AE)} = \frac{T_0}{K_{AE}} = 20/2 = 10s$$

$$T_{m(CF)} = \frac{T_0}{K_{CF}} = 30/3 = 10s$$

$$S = \frac{V_m}{V_0} = \frac{T_0}{T_m} = \frac{100}{70} = 1'43 \quad \% \text{ mejora (v)} = 43\%$$

$$\text{Generalizando, } S = \frac{T_0}{T_m} = \frac{T_0}{T_0 \cdot (1 - (f_{AE} + f_{CF})) + T_0 \cdot \frac{f_{AE}}{K_{AE}} + T_0 \cdot \frac{f_{CF}}{K_{CF}}}$$

$$\geq \frac{1}{(1 - (f_{AE} + f_{CF})) + \frac{f_{AE}}{K_{AE}} + \frac{f_{CF}}{K_{CF}}}$$

Sustituyendo $f_{AE} = 0'1$ y $f_{CF} = 0'1$

$$\frac{1}{(1 - 0'1) + 0'1 + 0'1} = \frac{1}{0'7} = 1'43$$

Al final, lo mejor es mejorar aquellas partes que más se van a utilizar. La ley de Amdahl en este momento ayuda a entender la mejora pero si el servidor fuera con un CPU mononúcleo.

► **Ejercicio 13)** Una aplicación informática se ejecuta en una computadora durante un total de 70s. Mediante el uso de un monitor de actividad se ha podido saber que el 85% del tiempo de ejecución se utiliza la CPU, mientras que el resto del tiempo hace uso del Disco Duro.

Determine cuántas veces debe ser como mínimo más rápido un CPU mejorado para que, costeando el doble, hubiese valido más la pena de cara a la relación prestaciones del sistema / coste del CPU.

$$DD \boxed{CPU_0} \quad T_0 = 70s$$

$$DD \boxed{CPU_m} \quad T_m = ?$$

Coste doble, por lo que Prestaciones mejorado > 2 · Prestaciones orig.

$$V_m > 2 V_0 \rightarrow S = \frac{V_m}{V_0} > 2 \text{ ó } S = \frac{T_0}{t_m} > 2$$

$$S = \frac{T_0}{(1-g)\frac{T_0}{K} + \frac{f}{K} \cdot T_0} = \frac{1}{0'1S + 0'8S} > 2$$

$$1 > 0'30 + \frac{1'7}{K} \Rightarrow K = 0'3K + 1'7$$

$$\Rightarrow K - 0'3K \geq 1'7 \Rightarrow 0'7K \geq 1'7 \Rightarrow K \geq \frac{1'7}{0'2}$$

$$\rightarrow K > 2'43$$

$$T_m = 0'1S \cdot T_0 + \frac{0'8S}{2'43} \cdot T_0 = (0'1S + 0'3S) T_0 \\ \approx 0'59 T_0 = 34'99 S$$

► Ejercicio III Ante la necesidad de reducir el tiempo de ejecución de un programa de cálculo de trayectorias espaciales, un equipo de arquitectos de computadoras ha diseñado un nuevo procesador que mejora 3 veces la ejecución de las operaciones de coma flotante. El programa, cuando se ejecuta utilizando el nuevo procesador, emplea el 65% del tiempo en la realización de operaciones de coma flotante.

a) Calcule qué tanto por ciento del tiempo

de ejecución necesitan las operaciones de coma flotante en el sistema con el procesador original

b) Indique la ganancia en velocidad global conseguida por el nuevo procesador

\geq

1

$$(1 - (f_{AE} + f_{CE})) + \frac{f_{AE}}{K_{AE}} + \frac{f_{CE}}{K_{CE}}$$

Sustituyendo $f_{AE} = 0'1$ y $f_{CE} = 0'1$, $\frac{1}{(1 - 0'1) + 0'1 + 0'1} = \frac{1}{0'7} = 1'43$

Al final, lo mejor es mejorar aquellas partes que más se van a utilizar. La ley de Amdahl en este momento ayuda a entender la mejora pero si el servidor fuera con un CPU mononúcleo.

► **Ejercicio 13)** Una aplicación informática se ejecuta en una computadora durante un total de 70s. Mediante el uso de un monitor de actividad se ha podido saber que el 85% del tiempo de ejecución se utiliza la CPU, mientras que el resto del tiempo hace uso del Disco Duro.

Determine cuántas veces debe ser como mínimo más rápido un CPU mejorado para que, costeando el doble, baje ese uso del 85% a un 20%.

DD	CPU0	T0 = 70s
----	------	----------

DD	CPUm	Tm = ?
----	------	--------

Coste doble, por lo que Prestaciones mejorado > 2 · Prestaciones orig.

Resto	CP <u>0</u>	T_0
	K=3	
Resto	CP <u>m</u>	T_m

1 35% 1 65% 1

$$S_0 \text{ (m)} = \frac{V_m}{V_0} = \frac{T_0}{T_m} = \frac{0'35 \cdot T_m + 0'65 \cdot 3 \cdot T_m}{T_m} = \frac{0'35 + 0'65 \cdot 3}{1} = 0'35 + 0'65 \cdot 3 = 2'3$$

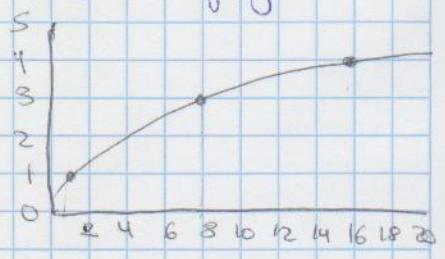
$$(0) \frac{T_m}{T_0} = \frac{T_m}{0'35 T_m + 0'65 \cdot 3 T_m}$$

► Operación 10j Un equipo de biólogos que investiga sobre clonación de células utiliza el procesador ALIANT para ejecutar un simulador que se puede parallelizar en una fracción f de su tiempo de ejecución. La figura

presenta la ganancia en velocidad conseguida por la máquina paralela en la ejecución del simulador para diferentes valores del nº de procesadores.

a) ¿Cuál es la fracción parallelizable f del programa de simulación?

Severable		Paralelizable
(1-f)T ₁	fT ₁	T ₁
	K=p	
(1-f)T ₁	$\frac{f}{p} \cdot T_1$	T _p



$$S = \frac{V_p}{V_i} = \frac{T_1}{T_p} \Rightarrow \frac{\frac{1}{R}}{(1-f)\frac{1}{R} + f \cdot \frac{1}{P}} = \frac{1}{(1-f) + \frac{f}{P}}$$

Con la formula se ayuda de la gráfica...

$$S_{p=6} = 3 = \frac{1}{1-f + \frac{f}{6}} \Rightarrow 3 - 3f + \frac{f}{2} = 1$$

$$\rightarrow 6 - 6f + f = 2 \Rightarrow 4 = 5f \Rightarrow f = \frac{4}{5} = 0.8 (80\%)$$

2. Componentes Hardware de un Servidor

La Placa Base

Una placa base es una tarjeta de circuito impreso de la que se basa un computador. Su objetivo es interconectar los distintos elementos hardware para implementar el servidor.

Tarjeta de circuito impreso Es una tarjeta en la que se dibujan un circuito o pista de cobre uniendo los contactos. Están hechas de láminas de un substancial no conductor en las que se intercalan pistas de cobre. Las placas base son multi-capa, es decir, las pistas pueden pasar por una de las capas conductoras. Para conectar las pistas se usen unos agujeros o vias entre capas.

Las placas base tienen distintos tamaños y formas normalmente estandarizados:

Una placa base suele tener un nº de serie o modelo, de la cual se pueden extraer qué características y compatibilidades hay a través de un manual de usuario, de las especificaciones técnicas del producto o de un

Data Sheet.

Cuando alguien manipula la placa base o sus componentes ya que la electricidad electrostática puede dañar cualquier componente o transistor resultando en que deje de funcionar. Es recomendable no tocar nunca ningún contacto metálico; descargar la electricidad estática; asegurarse que el equipo esté apagado antes de instalar o quitar un componente (o no ser que sea hot plugging o que admite swapping) y no forzar la inserción de cables (hacer una forma única de conectarse)

La fuente de alimentación

La misión de esta es convertir la corriente continua en alternativa. Pasa de 230 V AC a pines de 3,5 o 12V en DC. Esta alimenta la placa base y los periféricos. La potencia es variable, dependiendo del fabricante.

La certificación 80+ certifica la eficiencia de esta, haciendo ver que transforme la corriente ~~mientras~~^{en vez de} calentarse.

Módulo regulador de voltaje

Su misión es adaptar la tensión continua que arroja en la fuente de alimentación para tensiones menores y ofrecen estabilidad.

Se comparan de transistores, bobinas y condensadores.

El zócalo de la CPU

Puede ser de dos tipos

PGA (Pin Grid Array) El zócalo tiene un array de contactos al que se inserta (sin presión) los pins que lleva la CPU. Es utilizada en los sockets Ryzen de AMD actualmente.

LGA (Land Grid Array) El zócalo tiene un array de pins que tocan los conectores o contactos de la CPU. Es usada en procesadores Intel.

CPU : Microprocesador

El CPU o microprocesador es quien hace los operaciones que lleva dentro uno o varios cores o núcleos.

Históricamente se sigue la ley de Moore hasta 2005, donde el nº de transistores crece exponencialmente cada generación (18 meses a 2 años). La frecuencia en 2005 se tuvo que dejar de aumentar por la barrera térmica, haciendo así que hubiera que emplear en usar varios núcleos que se ejecutaran juntos con voltajes, frecuencias y temperaturas para incrementar su rendimiento.

(las CPUs hoy en día ~~sobregan~~ contienen controladores gráficos y de memoria, teniendo más que ser un SoC (System on a chip))

- ¿Y qué diferencia a un microprocesador de servers de uno para escritorio en la misma generación?

Un mayor nº de cores ya que los clientes de los servidores suelen usar varios hilos en paralelo con una gran eficiencia. Los procesadores llevan SMT o HyperThreading para poder trabajar en 2 hilos por core.

Menor frecuencia debido a que son muchos más cores y la barra térmica no permite sobralo, si bien se permite una frecuencia turbo en algún core concreto.

Mayor caché, menor precio

Mayor tamaño ^{RAM} que puede direccionar, y acepta memoria con corrección de errores, aumentando así la fiabilidad

Más canales de memoria que puede trabajar simultáneamente

Posibilidad de trabajar con varios núcleos en la misma placa base.

Más líneas de E/S

Mayor consumo

No contienen controladores gráficos, ya que no son necesarios en el mundo de los servidores.

Están más preparados y testeados ya que deben tener suficiente calidad para durar 24/7.

Más tecnologías dedicadas a facilitar tareas propias de servidores como virtualización, encriptación o Hyper-threading, si bien algunas tecnologías aparecen en el mundo del PC.

► Sobre AMD ...

AMD históricamente fue una empresa que clonaba CPUs de Intel hasta la creación de las instrucciones x64.

Su serie de CPUs de escritorio fue primero Opteron (de hecho fue el primer CPU en tener más de 1 núcleo); y hoy en día, Epyc. Su filosofía se basa en añadir los cores en chiplets.

Otra serie de CPUs fue Opteron X, más sencilla que Opteron, pero añadió un controlador gráfico, siendo así una APU.

Y otra más, Opteron A, cuyo repertorio de instrucciones es ARM en vez de x64, haciendo así que se reduzcan los consumos y añadiendo controladores PCIe, Ethernet, SATA... en el chip, convirtiéndose en un SoC.

Sobre IBM

IBM tiene a otra filosofía basada en que da igual el precio, por lo que no escatiman en recursos, haciendo que sus supercomputadoras sean carísimas.

Hacen clusters de servidores denominados mainframes que son disponibles, fiables y con mayores prestaciones por vatio. Es usado por grandes empresas.

Algunas mejoras son:

Possibilidad de hyperThreadings 1 x 8!

Mayor posibilidad de poner más CPUs.

Disipadores

Su misión es transmitir el calor al aire.

El disipador de la CPU puede ser:

Pasivo No requiere electricidad. Son placas metálicas con glicos para transmitir el calor y ser llevado por el aire.

Activo Requiere electricidad. Se trata de ventiladores o bombas de agua para fluir un fluido por un circuito.

Los ventiladores se pueden controlar dando sus revoluciones y permitiendo así consultar la velocidad de esta, subir o bajarla.

La RAM (Random Access Memory)

La memoria RAM dinámica (DRAM) es de lectura/escritura, volátil (se borra si se va la electricidad) pero necesitan refrescarse (son un condensador y transistores que se descarga y carga, por lo que la memoria debe, cada cierto tiempo, ser recargada)

La memoria RAM estática (SRAM) tienen más densidad por bit y son más caras por lo que necesitan más realimentación constante, por lo que no necesitan refresco.

En un principio las memorias eran asíncronas y establecían protocolos para transportar datos. Después se hizo síncrona, es decir, que debe esperar a una señal de reloj. Hoy en día se usa la tecnología DDR (Double Data Rate) que permite leer la memoria cada cuatro ciclos del reloj. Así, DDR2 lee 4 bits por ciclo, DDR3, 8 bits por ciclo y DDR4, 16 bits por ciclo.

Los módulos SDRAM tienen los conectores ^{o pastillas} en un sentido, y los DIMM ahora los tienen en ambos sentidos, aumentando el ancho de banda. El bus de datos es de 32 o 64 bits half-duplex (siendo así que los datos, o se leen, o se escriben, no pueden pasar simultáneos). El voltaje varía según la tecnología que sea. Si las pastillas están en 1 lado, cada 1 de ellas contiene 8 bits.

los tipos de DIMM son:

SODIMM (Small Outline). Son memorias con tamaño más pequeño, se usan en portátiles.

DIMM o U-DIMM (Unbuffered) Se usan en PCs de escritorio.

ED-DIMM: Es una memoria con ECC (Error Correction Code) lo que permite, con muy bajo de control extra, aumentar la fiabilidad e intentar reconstruir el dato o volver a pedirlo si es posible. Se usa en servidores.

R-DIMM (Registered) Añade un chip de registro que almacena los señales de control para que las direcciones se registren y se transmita a la memoria, y permiten tener más memoria en el mismo módulo, a costa de una mayor latencia.

LR-DIMM (Load Reduced) Añade un registro que almacene las señales de control y el dato a leer/escibir, siendo así el módulo con menor tamaño, ya que cargan direcciones hasta cierto punto y de ahí parten.

La DRAM se agrupa en:

Canales Memorias que puede usar la CPU a la vez

Ranuras Cada uno de los baches donde conectar la memoria RAM

Rango de memoria Es una agrupación de chips que proporcionan una palabra completa. Puede ser en el mismo módulo,

1R x 4

Un rango con chips de memoria de 4 bytes

2R x 8

Dos rangos con chips de memoria de 8 bytes

Info 1
Comprob
de una
RAM

A la hora de instalar la placa base habrá que alinear los agujeros de ésta con los standoffs de la caja.

Ranuras de Expansión

Permiten la conexión a la placa base de otras tarjetas de circuito impreso denominadas tarjetas hijas, o bien un riser, que hace que el conector se ponga rotado 90°.

En principio eran de estándar ISA, donde las placas precisaban de jumpers entre pines para su configuración.

Hoy en día se usa el estándar PCI (Peripheral Component Interface), el cual usa un bus entre todos, transmitiendo datos en paralelo. Es de 32 o 64 bits que se transmiten según la señal de reloj. Son

half-duplex, por la cual las pistas se usan sólo para leer o sólo para escribir (no ambas a la vez).

Es plug and play, es decir, se puede poner y quitar (tras un reinicio).

El ancho de banda que puede transmitir se calcula

Frecuencia x bits. Así, los habla de $33 \text{ MHz} \times 32 \text{ b}$,
(133 MBps)

$66 \text{ MHz} \times 32 \text{ b}$, $66 \text{ MHz} \times 64 \text{ b}$ y en caso de
(266 MBps) (533 MBps)

Servidores, el estandar PCI-X $133 \text{ MHz} \times 64 \text{ b}$,

la cual transmite $\approx 16 \text{ Bps}$

Un poco más tarde salió el estandar AGP

(Accelerated Graphics Port). Similar al PCI (32 b, bus paralelo) con conectores de 132 pinillas que se usaban específicamente para tarjetas gráficas. Su última versión,

la AGP 8x, tenía un ancho de banda de 26 Bps

El estandar usado hoy día es el PCI Express.

► Características

Protocolo serie (PCIe) - Conexión punto-a-punto (ya no es un bus con líneas compartidas) a través de vías (lanes) compuestas de 4 cables (2 por cada sentido) + lo que lo hace

Protocolo paralelo (PCI, AGP) full-duplex. Es serial, es decir, los bits vienen por la Lane de uno en uno

- La transmisión es sincrona estando el reloj sincronizado según la fluctuación de los datos.
- Es hot plug (se puede pinchar con el PC en funcionamiento) y permite la intercalación de Entrada/Salida.
- Al ser una conexión serie, y para una correcta comunicación, hacen una codificación añadiendo bits para redundancia.
 - + En las versiones 1 y 2, por cada 30 bits transmitidos, 8 son de datos y 2 de redundancia. Por ello, en PCI 1.0 podía tener un ancho de banda de 250 MB/s y en PCI 2.0, de 500 MB/s.
 - En las versiones 3 y 4, de 130 bits transmitidos, 128 son de datos y 2 de redundancia, por lo que PCIe 3.0 llegaba a 16 Gbps y PCIe 4.0, a 26 Gbps aprox.
- Es escalable, ya que se puede añadir o quitar lanes, llegando a tener hasta 16 lanes. Este nº se negocia con el dispositivo.

El uso más práctico del PCIe x16 es su uso en tarjetas gráficas.

► Ventajas de interfaces serie respecto a paralelo:
Las expansiones paralelas no podían sobrar las frecuencias mucho ya que las pistas no llegaban todas al mismo

tiempo y podría haber corrupciones o fallos de lectura. Las expansiones serie entran ese rendimiento o SKew, además de un menor nº de señales para que el rendimiento sea similar. Por consecuencia, es más escalable y facilita conexiones full duplex.

Almacenamiento permanente (no volátil).

La memoria DRAM y SRAM es eso, una memoria, pero volátil (en cuanto se apague se pierde la información).

Hay varios tipos de almacenamiento: magnéticos, ópticos y NVRAM. Se suelen ofrecer en varios factores de forma, medidos en pulgadas; y se conectan a la placa base con conectores estandarizados.

Los principales son los discos duros, SSD y cintas.

Discos duros

Almacenamiento basado en unos discos rígidos de material magnético. Se lee y escribe con unos cabezales magnéticos controlados por un brazo motor y el giro de los discos. Los datos se distribuyen en pistas.

Discos sólidos o "Discos duros SSD"

Se basa en un transistor MOSFET donde se le añade una "puerta flotante" donde se almacena el dato permanentemente si se le asigna un voltaje más alto que borre ese dato. Se tardan así el mismo tiempo en llegar a un dato. El nº de chips importa ya que así se pueden hacer lecturas y escrituras en paralelo. A esta memoria se le conoce como NV RAM o Non volatile RAM. Lo malo es que si se hacen muchas reescrituras, o con el tiempo, esa puerta deja de funcionar. Para equilibrar el desgaste se distribuye la dirección lógica de los celdas de memoria, conocido como wear leveling, a través de un controlador.

Las celdas pueden ser de un solo nivel o de varios, ya que en SLC la celda es de un bit y en MLC de varios (según la medición de carga)

RAID

En el mundo de los servidores se necesitan mecanismos de redundancia para aumentar la tolerancia a fallos. Así se hace el concepto de RAID o Redundant Array

of Independent Disks y hay varias configuraciones.

RAID 0 : Aumenta el ancho de banda poniendo una parte de la unidad en un disco, y la otra en otro.

RAID 1: Se pone el mismo dato en dos discos distintos aumentando la redundancia.

RAID 10: RAID 1+0. Un RAID 0 de RAID 1.

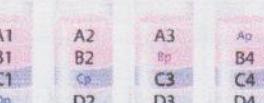
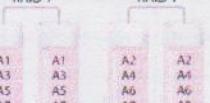
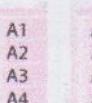
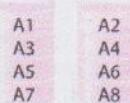
RAID 5:

RAID 0

RAID 1

RAID 10

RAID 5



Para controlar el RAID puede hacerse una software (gestionandolo el SO) ; o una hardware con una placa de expansion que lo controle.

Unidades Opticas

Es una alternativa para almacenar datos en un largo tiempo. Son discos que en contacto con un laser son grabados grabando huecos, haciendo así ls y os. Sin embargo, el desgaste y su baja velocidad no los hacen aptos para backups de servidores.

	CD	DVD	Blu-ray
Capacidad (GB)	0,64-0,7	4,7-8,5	25-128
Ancho de banda ref. 1X (Mbps)	1,4	11	36
Ancho de banda máx. (Mbps) (X-factor)	88 (72X)	266 (24X)	576 (16X)

Onduladores de cinta

Son medios con una gran densidad de bits por área. Permiten decenas de TB por cinta y velocidades de lectura/señal de 150 MB por segundo.

Estandares de conexión

IDE / Parallel ATA Es un conector de 40 pines corto, basado en Advanced Technology Attachment, funcionando con 2 conectores (maestro/esclavo)

	SATA I	SATA II	SATA III
Frecuencia	1500 MHz	3000 MHz	6000MHz
Bits/clock	1	1	1
Codificación 8b10b	80%	80%	80%
bits/Byte	8	8	8
Velocidad real	150 MB/s	300 MB/s	600 MB/s

SATA Es un conector serie de 1 m de longitud con 7 pines que usa la codificación 8b/10b. La tercera versión de SATA, con 6 Gbps, solo puede transmitir hasta 600 MB/s. Utiliza una interfaz AHCI (Advanced Host Controller Interface) que permite el hot plug, y una cola NCA (Native Command Queuing) que optimiza los ordenes de E/S y evita así que los pistones estuvieran cercanos y no sobreutilizar el cabezal.

En el mundo de los servidores...

SCSI: Small Computer System Interface. Paralelo 16 bits half-duplex más veloz que PATA. Permitía hot-plug y conectar varias unidades en cadena usando daisy-chain.

Actualmente se usa una derivada **Ultra-SCSI** con conectores de 50 pines que tiene longitudes de 32 metros.

SAS Serial Attached SCSI. Permite leer y escribir a la vez (es full-duplex). Es compatible con discos SATA pero no al revés. Tienen grandes voltajes y eso permite longitudes de cable más grandes, de hasta 30 m.

Los conectores se parecen a los de SATA. De hecho, usa también codificación 8b10b, con velocidades reales hasta 1200MB/s.

Si se quisiera añadir muchos discos duros se podría hacer uso de conectores **mini-SAS**, de los que se pueden parar hasta 4 conectores SAS o SATA a través de un splitter.

O aún más discos con tarjetas de expansión

SAS Expanders.

Pde con protocolo NVMe (Non-Volatile Memory express) Usado por SSD son anchos de banda mayores a los SATA, se conectan a los PCI express, M.2 (un conector más pequeño que internamente es un PCIe x4), el U.2 (similar a M.2 pero se usa poco) o SATA Express (combinación de PCIe y SATA)

Conectores del panel trasero

Conector PS/2 para conectar teclado y ratón

VGA para conexión de video analógico

HDMI / Display Port para conexión de video digital (con audio)

eSata Conector SATA externo. Tiene suficiente corriente para alimentar un cable de 2 metros.

USB 2.0 / 3.0

Puerto de audio

Conector Ethernet para conexión a Internet por cable.

En un servidor no hacen falta estos conectores, siendo los básicos:

Conector RJ-45 / Ethernet Se usa en comunicaciones a larga distancia por Redes de Área Local. Hay muchos estándares compatibles entre sí, full-duplex. Además, estándares como SCSI se pueden conectar vía internet como iSCSI.

Puerto serie, o en su defecto, Ethernet de bajas prestaciones.

VGA para un video sencillo & de bajas prestaciones.

USB A pesar de ser serie, son half-duplex. Los ZO suelen usarse

Otros conectores internos

System panel Es el conjunto de conectores desde la carcasa. Por ejemplo, PW Switch enciende la fuente de alimentación, Reset Switch reinicia el PC, LEDs de estado y un Speaker que emite sonidos.

Hay también conectores serie (para sacar un puerto serie) o USB (conjunto de 9 pines para conectarlo al frontal de un chasis o en tarjetas de expansión).

La Bios

Es una memoria ROM (Read-Only Memory) que contiene instrucciones de arranque del ordenador. Lo que hace es:

[Identifica los dispositivos instalados]

Instala los drivers básicos (Bios = Basic I/O System)

Prueba el sistema con POST (Power-on Self-Test)

Mira si se han pulsado los teclas.

Si se hace se entra al programa de la placa base.

Si no, se comienza el sector de arranque del disco. Esta secuencia de arranque se mete en una RAM alimentada por una pila (punto con otras configuraciones)

También dispone de otras para resetearse en caso de problemas de overclock

Chipset : El puego de chips

El chipset es un conjunto de circuitos integrados de la placa base encargados de controlar las comunicaciones entre los componentes de la placa base.

Hay dos grandes componentes:

Puente norte , que se encarga de las transferencias de mayor velocidad

Puente sur , que se encarga de las transferencias entre el puente norte y el resto de periféricos con menores exigencias .

Hoy en dia , el propio CPU se encarga a su vez de ser el puente norte y hay solo un chipset con un chip puente sur, pero solo en algunas plazas modernas.

3. Monitoreo de servicios y programas

¿Por qué lo tengo que monitorizar?

Como administrador / ingeniero:

Hay que conocer los recursos para saber...

qué hardware hay que reconfigurar / sustituir / añadir

qué parámetros del sistema hay que ajustar

Poder predecir cómo va a evolucionar la carga con el tiempo

Poder dar servicios a los clientes

Obtener un modelo del componente de cara a decidir

qué posib. si p..

Como programador, se podrían conocer las partes críticas de una aplicación de cara a su optimización, observando hot spots

El SO se monitoriza para adaptarse a la carga.

Pero... ¿qué es la carga?

Larga: Conjunto de tareas que ha de realizar un sistema, es decir, todo aquello que demande recursos.

Es más fácil interpretarlo como la actividad de un sistema, o conjunto de operaciones que se realizan en el sistema.

Como consecuencia de la carga que soporta.

Para monitorizar esa actividad existen monitores de actividad que miden esa actividad y facilitan así su análisis.

Algunas acciones son medir algunas variables que definen la actividad, procesar y almacenar la información y mostrar los resultados.

Se puede medir de dos formas:

Monitor por eventos. Se activa cada vez que ocurre un evento. Es variable, dependiendo de cuando se haga tal evento.

Monitor por muestreo. Se activa cada cierto tiempo. La información es estadística cuyo volumen varía según su periodo de muestreo que puede ser fijo o variable.

Los monitores pueden ser software (o sea, programas instalados en el sistema), hardware (con dispositivos físicos de medida que dejan una mayor sobrecarga) o híbridos.

Los monitores pueden o no tener interacción con el analista o administrador.

Si no existe, la consulta se hace usando otra herramienta independiente al proceso de monitorización (batch monitors)

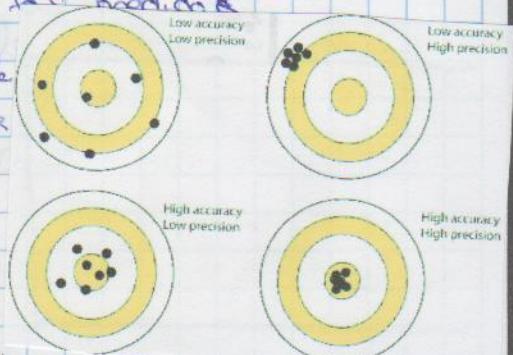
Si existe, se pueden consultar los valores durante la monitorización y/o interactuar con ellos mediante representaciones gráficas diversas. (Monitores interactivos / on-line monitors)

Atributos que caracterizan a un sensor o monitor

Exactitud / Accuracy, Offset Mide cuanto se aleja el valor medido del valor real

Precision Mide la dispersión de los resultados

Resolución del sensor Es la medida que tiene que cambiar el valor a medir cada vez que se detecte un cambio.



Tasa máxima de Entrada / Max Input Rate Frecuencia máxima de ocurrencia de los eventos que el monitor puede soportar
Anchura de entrada (Input Width) Mide cuanta información almacena por cada medida que toma el monitor

Sobre carga (Overhead) Mide los recursos que gasta el monitor al sistema ya que ejecutar el monitor carga más el sistema.

$$\% \text{ Sobre carga} = \frac{\text{Uso del recurso por parte del monitor}}{\text{Capacidad total de recurso}} \cdot 100$$

Eso si, habrá veces que en monitores por muestreo se usa el tiempo empleado y el tiempo entre muestras.

Monitores a nivel de sistema

▷ El directorio /proc (en sistemas basados en Unix)

Es una carpeta del sistema que se aloja en DRAM y almacena toda información relativa a monitoreos a nivel de sistema. Gracias a la estructura de Unix, se puede acceder a ella desde el sistema de archivos.

A través de la carpeta podemos:

Acceder a información sobre el SO

Acceder a la información de cada uno de los procesos en uso del sistema con [/proc/[PID]]

Acceder (y modificar) algunos parámetros del Kernel con la carpeta [/proc/sys]

▷ Uptime

Mide el tiempo que lleva el sistema en marcha y la carga media que soporta.

Unix denomina Carga al nº de procesos en modo running, runnable o I/O blocked que almacena en la cola de procesos del núcleo.

Los estados básicos de un proceso son:

Running Ejecutando.

Runnable Listos para ejecutar

I/O blocked o Uninterruptible sleep

Espesan al SO para desbloquearse
cuando lo solicita por I/O

Interruptible sleep Esperan a la interacción del usuario

La Carga media no se basa en una medida simétrica, sino en una medida ponderada. No es ni volver de fácil interpretación.

▷ ps

Da información sobre el estado actual de los procesos del sistema

USER Usuario que lanzó el proceso

%CPU, %MEM Porcentaje de procesador y memoria física usada

RSS Memoria física ocupada por el proceso en KiB

STAT Estado en el que se encuentra:

▷ strace

Crea una lista con las llamadas al sistema.

Sigue se van realizando en el sistema.

[Running, Sleeping, Zombie, Nicd, < higher priority, Swapped, Stopped]

▷ top

Es un monitor interactivo que muestra cada 1 segundo dando información parecida a ps.

En %Cpu(s) podemos ver algunas cargas como usuario, system, idle (enroscado), dando una idea de la carga del sistema. También está Wait for E/S y time stolen by hypervisor.

▷ vmstat

Monitorea la paginación, el swap, la CPU y las interrupciones del sistema.

Aquí pueden verse de forma fija los cuadros procesos estan corriendo y cuales están bloqueados, haciendo así una carga media por medida antimétrica.

▷ Paquetes de monitorización

Hoy en día se usan paquetes de monitorización como sysstat. Dentro de este está sar (System Activity Response) que recopila información sobre la actividad del sistema.

Hace uso de sade, que recoge los datos estadísticos y construye un registro en formato binario con formato saDD (DD = día del mes); y el propio Sar, que lee los datos que recoge Sade y los traduce en el frontend a texto pleno.

Tiene muchos parámetros: ➔

(sar equivale a sar -u)

Si quisieramos, por ej, ver el uso desglosado de cada CPU recopilada cada segundo:

sar -P ALL 1

Los datos almacenados por sade se almacenan en un fichero histórico de datos dano. Esto se ejecuta con un cron añadiendo, cada 5 minutos, las medidas.

Modo interactivo: [tiempo_muestreo, nº muestras]
-f Fichero de donde extraer la información, por defecto: hoy
-s Hora de comienzo de la monitorización
-e Hora de fin de la monitorización
u Utilización global del procesador (opción por defecto)
-P Mostrar estadísticas por cada procesador (-P ALL: Todas)
-I Estadísticas sobre interrupciones
-C Cambios de contexto
-q Tamaño de la cola y carga media del sistema
-B Estadísticas globales de transferencias de E/S
-T Transferencias para cada disco
-d Conexión de red
-n Utilización de memoria
-R Estadísticas sobre la memoria
-A Toda la información disponible
...

► Ejemplo ¿Cuál es la anchura de entrada de sede?

-r w -r -- r -- l root root 3049952 23:55 5a02

Se supone que la primera muestra se toma a las 0:00

Si se toma cada 5 minutos, se cogen 12 muestras
cada hora, 288 muestras diarias.

Si el fichero pesa 3049952 bytes ($\approx 3'05\text{ MB}$),

$$\frac{3'05\text{ MB}}{288} = 10'59 \text{ KB}$$

► Otras herramientas para monitoreo de sistema

Los sistemas Unix tienen muchas herramientas de monitoreo que permiten medir el uso de cualquier cosa.

A nivel de paquetes, hay algunos como CollectL.

Nagios

► Sistematización de la monitoreo: método USG

Para no tener que lidiar con todos los datos se aplica el método USE, para el cual, por cada recurso, midemos:

Utilización % de utilización del recurso

Saturación: Ocupación de los colas de tareas que usan el recurso

Errores: Mensajes de error del Kernel sobre el uso de estos

Monitorización a nivel de aplicación (profilers)

El objetivo de un profiler es monitorizar la actividad generada por una aplicación con el fin de obtener información para poder optimizar código.

La información que suele proporcionar un profiler es:

Tiempo de ejecución del programa, de uso por usuario y por sistema, tiempo de E/S

Hot spots, parte de código donde pasa la mayor parte de su tiempo de ejecución.

Veces que ejecuta cada línea de programa

Veces que se llama a un procedimiento, y dónde y cuánto tiempo de ejecución

Fallos de caché (página genera cada línea del programa).

time

Es una primera aproximación, llamada /usr/bin/time

(no confundir con %\$ time). Este programa mide el tiempo

de ejecución de un programa y algunos estadísticas sobre la ejecución.

Una cosa interesante son los cambios de contexto voluntarios e involuntarios

Voluntarios: Cambios debidos a la necesidad de uso de recursos externos

Involuntarios: Cambios que el programa no quería hacer

Monitorización a nivel de aplicación (profilers)

El objetivo de un profiler es monitorizar la actividad generada por una aplicación con el fin de obtener información para poder optimizar código.

La información que suele proporcionar un profiler es:

Tiempo de ejecución del programa, de uso por usuario y por sistema, tiempo de E/S

Hot spots, parte de código donde pasa la mayor parte de su tiempo de ejecución.

Veces que ejecuta cada línea de programa

Veces que se llama a un procedimiento, y dónde y cuánto tiempo de ejecución

Fallas de caché (página genera cada línea del programa)

time

Es una primera aproximación, llamada /usr/bin/time

(no confundir con %\$ time). Este programa mide el tiempo

de ejecución de un programa y algunos estadísticas sobre la ejecución.

Una cosa interesante son los cambios de contexto voluntarios e involuntarios

Voluntarios: Cambios debidos a la necesidad de uso de recursos externos

Involuntarios: Cambios que el programa no quería hacer

Gprof

Da información sobre el tiempo de CPU que consume cada función de un programa y el nº de veces que se ejecuta cada función en proceso o hilo para programas escritos en C/C++.

Para compilar usamos el flag -pg que instrumenta el programa con herramientas de profiling. Al ejecutarlo, se recoge información en gmon.out y se visualiza con Gprof.

► ¿Cómo funciona?

- Genera una tabla de direcciones físicas en memoria de cada función del programa, incluso las de bibliotecas.
- Se inicializan 2 contadores de profiling a 0 en cada función. C1 mide el nº de veces que se ejecuta; y C2 estima su tiempo de CPU.
- El SO programa un temporizador que desencadena cada vez que se ejecuta código de un programa.
- Cada vez que se ejecuta una función se incrementa C1, se multa la función que la ha llamado y se guarda.
- Cada vez que el temporizador llega a 0 s, se interrumpe el programa y se incrementa el C2 de la función interrumpida.

- Se estima el tiempo de CPU de cada función con el tiempo de CPU y el contador c_2
- También se generan los perfiles planos (flat) y de llamadas (call)

▷ El flat profile

% time : % Tiempo de CPU que usa el código propio de la subrutina.

Cumulative seconds : Suma acumulada de segundos consumidos en CPU por la subrutina y las que aparecen encima de esta en la tabla.

Self seconds : Tiempo que consume el código propio de la subrutina, se calcula (N° llamada \cdot self seconds por llamada)

Self s/call (Self seconds por llamada) Tiempo CPU medio de ejecución del código propio por cada llamada a la subrutina.

Total s/call (Total seconds por llamada) Tiempo (PV) medio total de ejecución por cada llamada a la subrutina, es decir, contando las subrutas a las que llama desde esa subrutina.

► El call profile

Index Índice de cada función (en el mismo orden que en el perfil)

Name Relación de todos las funciones llamadas por la de la fila.

Called indica cuántas veces es llamada ^{ha sido por} la función y, a veces, cuántas veces de ellas la función de Index (con formato Veces llamadas desde subrutina / Veces llamadas totales)

gcov

No se considera como profiler, pero sirve para ver cuántas veces se ejecuta una línea de código de un programa. Sirve para baterías de pruebas. Está dentro de las herramientas de gcc, instrumentando el programa.

Para ello se crean los flags -fprofile-arcs
-fprofile-coverage

Perf

Es un conjunto de herramientas para el análisis de rendimiento en Linux basadas en eventos software y hardware. Permiten analizar el rendimiento de

▷ El call profile

Index Índice de cada función (en el mismo orden que se encuentra en el perfil)

Name Relación de todos los funciones llamadas por la de la fila.

Called Indica cuántas veces es llamada ^{ha sido por} la función y, a veces, cuántas veces de ellas ^{ha sido por} la función de Index (con formato $\text{V} \text{e} \text{c} \text{e} \text{s} \text{ l} \text{l} \text{a} \text{m} \text{o} \text{d} \text{o}$ desde subrutina / $\text{V} \text{e} \text{c} \text{e} \text{s} \text{ l} \text{l} \text{a} \text{m} \text{o} \text{d} \text{o}$ totales)

gcov

No se consideran como profilers, pero sirve para ver cuántas veces se ejecuta una línea de código de un programa. Sirve para baterías de pruebas. Está dentro de las herramientas de gcc, instrumentando el programa.

Para ello se crean los flags -fprofile-ares
-ftest-coverage

Perf

Es un conjunto de herramientas para el análisis de rendimiento en Linux basadas en eventos software y hardware. Permiten analizar el rendimiento de

- en hilo individual
- en proceso y sus hilos
- todos los procesos ejecutados en una CPU
- Todos los procesos del sistema

▷ Algunos parámetros

perf list : lista los eventos disponibles con perf

perf stat Cuenta los eventos que han sucedido durante el tiempo de ejecución. Cuenta tanto eventos software como eventos hardware.

perf record : Hice a través de temporizadores de forma que cada vez que se hace un evento se mide. Lo registrado se puede ver en perf.data

perf report: Da el flat profile de los eventos pasados como argumento en perf record (Un profile para cada evento a analizar). Su profile llega a un grado muy fino de análisis.

perf annotate Analiza perf.data dando información sobre el consumo de tiempo en cada línea.

Valgrind

Es un conjunto de herramientas para el análisis y mejoría del código. Pasa el lenguaje máquina o lenguaje de Valgrind, y lo instruye directamente para poder recompilarlo y usarlo una SO. Eso sí, al hacerlo, el overhead es muy grande. Se usa más para debugging.

4. Análisis comparativo del rendimiento

Características de un buen índice de rendimiento

Representatividad y fiabilidad: Si un sistema A siempre presenta un índice de rendimiento mejor que el sistema B, es porque su rendimiento real en A será mejor siempre que en B.

Repetibilidad: Siempre que se mida el índice en las mismas condiciones, el valor de este será el mismo.

Consistencia y facilidad de medición: El índice se debe poder medir en cualquier sistema informático y la medida sea fácil de tomar.

Unrealidad: Si el índice de rendimiento aumenta, el rendimiento real del sistema debe aumentar en la misma proporción.

Tiempo de ejecución, frecuencia de reloj y CPI

Basándose en la fórmula de AC: $Tej = NI \cdot CPI \cdot T_{CPU}$ vemos que como índices de rendimiento valdrían la frecuencia de reloj del CPU y el CPI. Pero no lo son, ya que hay contraprestaciones.

Y si usamos Tejición en un determinado programa?

Para que sea consistente, el programa deberá estar escrito en un lenguaje de alto nivel. Pero la repetitividad deberá ejecutar el programa en un entorno muy controlado. Y la representatividad y fiabilidad dependerá del programa a ejecutar.

MIPS

$$\text{MIPS} = \frac{N_1}{\text{Tiempo} \cdot 10^6}$$

Sus inconvenientes son:

la representabilidad y facilidad depende del peso de instrucciones ya que en un RISC aumenta los N₁ pero en CISC aumenta los CPI.

La repetibilidad también puede peligrar ya que en la misma máquina los MIPS pueden ser distintos.

MFLOPS

$$\text{MFLOPS} = \frac{\text{Op. coma flotante}}{\text{Tiempo} \cdot 10^6}$$

Los MFLOPS solo se basan en operaciones en coma flotante, entiendo I/O y suspensiones.

Sin embargo se generan inconvenientes:

Repetibilidad: No todas las operaciones son igual de complejas. Pero para solventarlo se proponen las operaciones normalizadas, donde las operaciones tienen ciertos puntos o otros. Cuanto más complejas, más puntos o operaciones normalizadas vale.

Representabilidad y facilidad En distintas máquinas la coma flotante vería ya que habría COPS con floats de más o menos bits, lo que no les hace muy representativo.

No confundamos con comparar los tiempos en distintas cargas.

la carga real

Es una carga difícil de tratar en la evaluación de sistemas, ya que

Varia a lo largo del tiempo: No sabremos decir qué carga ver, si la que tiene en uno u otro momento.

resulta complicado reproducirla: En un servidor único puede que llegue una solicitud que se basa en una respuesta no recibida.

Interacciona con el sistema,

Nos contendremos con un modelo de la carga real o test workload

> Representatividad (los modelos de carga son aproximaciones que representan una abstracción de la carga que recibe un sistema informático. Este modelo debe ser lo más representativo posible a la par de ser lo más simple/compacto posible)

Para ello, podremos usar un modelo paramétrico o personalizado a partir de la monitorización del sistema ante la carga (características de la carga) o usar un modelo

ya existente similar al que se tiene en la máquina a través de referencias o benchmarking

Benchmarking

Se basa en usar uno o varios programas con el fin de comparar algunas características del rendimiento entre equipos. Tiene dos características:

Carga de prueba específica con la que se entra el sistema a evaluar

Configuro de unas reglas que se tienen que seguir para la correcta ejecución, obtención y validación de los resultados.

Ventajas de usar benchmarking

Alta probabilidad de encontrar un benchmark que encuentra reproduce condiciones similares a las que el servidor experimenta.

(as compilaciones entre el rendimiento de servidores similares son con reglas iguales y éstas (a pesar de que hacen veces en los que se da una ventaja hacia trampas)
Hay benchmarks que permiten ajustar la carga para medir la escalabilidad del servidor)

Con ello obtenemos información muy valiosa sobre cómo diseñar y/o configurar nuestros servidores.

Típos

- Programas que miden el tiempo necesario para ejecutar una cantidad preestablecida de tareas (como la mayoría)
- Programas que miden la cantidad de tareas ejecutadas en un tiempo preestablecido (como SLACOM)
- Programas que permiten modificar sus parámetros para adaptarlos a cada sistema (como TPC-C)

Según la generalidad del test

Microbenchmarks / Benchmarks para componentes específicos

Componentes o agrupaciones de estos componentes concretos. Es difícil saber cómo estrucutarlo.

Macrobenchmarks / Benchmarks de sistema completo (de aplicación real)

La carga intenta imitar situaciones reales. Tipos de un área.

▷ Sobre SPEC

Es un consorcio sin ánimo de lucro que hace conjuntos de micro benchmarks y herramientas para evaluar el rendimiento de un sistema. Es soportado por muchísimas empresas.

▷ SPEC CPU 2017

Es un paquete compuesto de 4 conjuntos de benchmarks distintos, siendo de tipo Speed, que mide el tiempo de respuesta, o Rate, que mide cuantos programas se ejecutan en un tiempo determinado (productividad).

Este se distribuye como un .ISO que contiene el código fuente, datasets que necesitan algunos programas para

Tips

- Programas que miden el tiempo necesario para ejecutar una cantidad preestablecida de tareas (como la mayoría)
- Programas que miden la cantidad de tareas ejecutadas en un tiempo preestablecido (como SLACOM)
- Programas que permiten modificar sus parámetros para adaptarlos a cada sistema (como TPC-C)

Según la generalidad del test

Microbenchmarks / Benchmarks para componentes Especificos

Componentes o agrupaciones de estos componentes concretos. Es difícil saber cómo extraerlo.

Macrobenchmarks / Benchmarks de sistema completo (de aplicación real)

La carga intenta imitar situaciones reales. Tipos de un área.

▷ Sobre SPEC

Es un consorcio sin ánimo de lucro que hace conjuntos de microbenchmarks y herramientas para evaluar el rendimiento de un sistema. Es soportado por muchas empresas.

▷ SPEC CPU 2017

Es un paquete compuesto de 4 conjuntos de benchmarks distintos, siendo de tipo Speed, que mide el tiempo de respuesta; o Rate, que mide cuantos programas se ejecutan en un tiempo determinado (productividad).

Este se distribuye como un ISO que contiene el código fuente, datasets que necesitan algunos programas para

ejecutarse, herramientas para compilar, ejecutar, obtener resultados, validar y corregir errores; y la documentación de los datos.

Los criterios generales para obtener los programas son:

Ten de ser aplicaciones reales

Se deben portar a distintas arquitecturas como x86,
x86-64, IBM Power, IA64.

Normalmente están escritos en C, C++ o Fortran, pero si un lenguaje es más popular, se pondrá en próximas versiones (como Python)

Los índices de prestaciones pueden ser de tipo Speed-base, Speed-peak, Rate-base, Rate-peak para coma flotante y enteros.

Base: Se compila en modo conservador, usando los mismos operadores de compilación

Peak: Es el rendimiento pico, que para ello se toman las configuraciones más óptimas para el sistema.

Para calcular el índice SPEC, el benchmark se ejecuta 3 veces y se pilla el resultado del medio. El índice final es la media geométrica entre el sistema evaluado y un sistema de referencia, usando las generaciones en velocidad

$$\text{Índice SPEC} = \sqrt[n]{\frac{t_{ref1}}{t_{base1}} \cdot \dots \cdot \frac{t_{refn}}{t_{basen}}}$$

▷ Sobre TPC (Transactions Processing Performance Council)

Es una organización sin ánimo de lucro especializada en benchmarks relacionados con comercios electrónicos y BD,

OLTP (Online Transaction Processing) Tratan de simular entornos con transacciones de comercio electrónico.

► Benchmarks de TPC

TPC-C Simula una gran compañía con varios almacenes con muchos productos. Los pedidos se envían a servidores locales y remotos.

TPC-E Simula una correduría de bolsa con solo una BD central. El benchmark es escalable de modo que pueden simularse transacciones de compraventa de diversos tamaños.

TPC-H, TPC-DS. Se basan en apoyo a la decisión. Se ejecutan consultas muy complejas a una gran BD con muchos datos.

Las métricas se hacen con pedidos procesados por unidad de tiempo superando ciertas requisitos de tiempo de respuesta. También miden el consumo de potencia y el coste por pedido procesado.

► Media aritmética

$$\sum_{k=1}^n t_k \cdot \frac{1}{n} = \bar{t}$$
 Dado un conjunto de n medidas, la media aritmética sera

Pero se puede asignar también pesos, $\bar{t}_w = \sum_{k=1}^n t_k w_k \cdot \frac{1}{\sum w_k}$ donde la suma de los pesos sea 1.

Así, por ejemplo; para no bajar la media por un t más lento se pide hacer un peso inversamente proporcional al tiempo.

emplado en hacerse en una máquina de referencia.

Pero claro, esa constante varía según la máquina de referencia, y con ello los pesos.

▷ Media geométrica

$$S_g = \sqrt[n]{\prod_{k=1}^n S_k}$$

Para un conjunto de n medidas

Si a S_n , su media geométrica es

Cuando las medidas son generadas en velocidad, el índice mantiene el mismo orden en las comparaciones independientemente de la máquina de referencia elegida.

Si premian así las mejores sustitutas pero no se castigan los empeoramientos no tan sustanciales.

Comparación de prestaciones en presencia de aleatoriedad

▷ La distribución de probabilidad normal o gaussiana

Independientemente de qué índice se escoga, el ingeniero deberá determinar si las diferencias entre las medidas obtenidas por un test de rendimiento en presencia de aleatoriedad son estadísticamente significativas. Así que repasaremos algo de estadística.

Distribución normal. Es una distribución caracterizada por la media μ y la varianza σ^2 . Cuya función de probabilidad es

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Así, la probabilidad de obtener un elemento del rango $[x-2\sigma, x+2\sigma]$ es del 95%.

! Teorema del Límite Central La suma de un conjunto grande de muestras aleatorias de cualquier distribución e independientes entre sí pertenece a una distribución normal

▷ Distribución t de Student

Si extraemos n muestras de pertenecientes a una distribución

Normal de media \bar{x}_{real} , y calculo estandarizamos la

medida

$$t_{exp} = \frac{\bar{x} - \bar{x}_{real}}{(s/\sqrt{n})}$$

Siendo \bar{x} la media muestral y s la desviación típica
muestra

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

y repitiendo el experimento muchas veces veremos que
 t_{exp} se distribuyen según la T de Student con
 $n-1$ grados de libertad

Gracias al test T se puede determinar un intervalo de confianza para T_{real} y no simplemente el valor medio medido

Diseño de experimentos de comparación de rendimiento.

Supongamos que queremos determinar cuáles de los siguientes factores afectan

Terminología estadística

Variable respuesta/dependiente o medida Índice de rendimiento que usamos para las comparaciones

Factor Cada variable respuesta que pueda afectar a la variable respuesta

Nivel Valores que cada factor puede asumir

Interacción Sucede cuando el efecto de un factor cambia para diferentes niveles de otro.

Tipos de diseños experimentales

Diseño con un solo factor Se usa una configuración fija de la que solo varía un factor cada vez, midiendo resultados para cada nivel de este factor.

Diseño multifactorial completo Se proveen cada posible combinación de niveles para todos los factores.

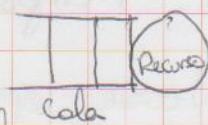
5. Análisis operacional en servidores

Redes de colas de espera

Modelo de un SI

Un modelo es una abstracción del sistema informático real, como un conjunto de dispositivos relacionados entre sí y los trabajos que "devoran" esos recursos.

Se consideran recursos o dispositivos a cosas como la CPU, la red... y se consideran devoradores o trabajos todo tipo, proceso o petición.



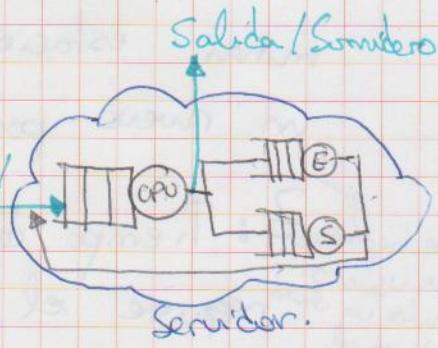
El modelo que vamos a usar es un core lógico que puede trabajar en un recurso, por lo que los demás deben esperar en una cola.

A partir de ahí podemos llegar a una red de colas, que es un conjunto de estaciones de servicio conectadas entre sí.

De entre todos los modelos, el más usado es el modelo de servidor control, el cual representa el comportamiento básico de los programas del servidor de cara a extraer información sobre su rendimiento.

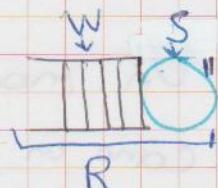
Para ello su comportamiento sería:

- 1 Llega un trabajo al servidor y usa la CPU (o espera a poder usarla) Entrada/Fuente
- 2 Tras usarla, el proceso pede acceso a E/S
- 3 Si va a E/S, espera a poder acceder y tras ejecutarse vuelve a CPU



Para poder trabajar con el modelo se usan algunas variables:

W: Tiempo de espera en cola Tiempo transcurrido desde que el trabajo solicita usar el recurso hasta poder usarlo.



S: Tiempo de servicio Tiempo transcurrido desde que accede al recurso hasta que lo libera (o sea, lo que tarda en procesar)

R: Tiempo de respuesta Suma de ambos tiempos ($R=W+S$)

(Véase ejercicio pág. 11)

!(Los valores medidos no han de ser necesariamente enteros)

Pero claro, no siempre es el caso de que haya una cola por dispositivo. Puede pasar:

Una cola para acceder a 3 dispositivos idénticos, lo que genera más probabilidad de ser atendido en ese instante, o sea, que la espera podría ser más corta.

¿Y si hubiera infinitos dispositivos? Entonces denotamos con que no habría espera en cola ($R=S$). Se denominan estaciones de tipo retraso. Con ello se da un nuevo parámetro

Con hipótesis: Tiempo de reflexión: Representa el tiempo que un usuario envía un ^{trabajo} _{trabajo} que requiere el usuario para mandar otra respuesta a lo trabajado al servidor que recibe del servidor.

Como no siempre es necesario que el usuario esté ~~conectado~~ comprobando el servidor, se pueden dar varias arquitecturas

Redes de colas cerradas

Son presentan un n° constante de trabajos que recorren por la red, denominados N_r . Los trabajos pueden ser de tipo batch / por lotes donde se llevan los trabajos en bucle por el servidor (de forma que $N_r = N_l$) ; o de tipo interactivo , donde también interactúa el usuario decidiendo en el trabajo, por lo que el n° de trabajos contendrá los trabajos de ~~los~~ ^{✓ NZ} los usuarios ($1 \text{ usuario} = 1 \text{ trabajo}$) y los que crean por el servidor ^{✓ NO} ($N_r = N_l + N_u$)

Redes abiertas

Los trabajos llegan a la red a través de una fuente externa , se procesan y luego salen a través de sumideros, sin retroalimentación entre fuente y sumidero . El n° de trabajos en el servidor ^{puede} ~~variar~~ con el tiempo. (N_l)

Redes mixtas

Aquellos que pueden combinar ambas, o no corresponden a ninguno de los anteriores.

Variables y leyes operacionales



Análisis operacional

Es la técnica de análisis de redes de colas basada en valores medios de diferentes variables medibles del servidor.

Esto nos proporcionará unas relaciones generales entre las variables operacionales denominadas leyes operacionales, que nos permitirán calcular las prestaciones del servidor para casos de baja y alta carga usando cálculos muy sencillos para permitirnos evaluar los efectos en el rendimiento de diferentes configuraciones hardware del servidor.

Variables del servidor y de cada Estación de Servicio

Sabemos que el servidor contiene K estaciones de servicio, denominando a todo el servidor globalmente como el dispositivo cero.

► Variables operacionales básicas Estos no se obtienen de valores medios, sino de la monitorización del sistema.

T : Duración del periodo de medida para el que se extrae el modelo.

A_i : Arrivals (i) N° de trámites solicitados a la estación i

B_i : Busy time (i) Tiempo donde se está ocupado la estación i
 $\rightarrow (Se\alpha \leq T)$

C_i : Completed (i) N° de trámites completados de la estación i

► Variables operacionales dedicadas Se pueden estimar de las variables básicas.

λ_i : Arrival rate (i) Tasa media de llegada ($\lambda_i = \frac{\Delta_i}{T}$) de la est. i

S_i : Service time (i) Tiempo medio de servicio ($S_i = \frac{B_i}{C_i}$) de la est. i

U_i : Utilization (i) Utilización media de la est. i ($U_i = \frac{B_i}{T}$)

X_i Productividad media de la estación i ($X_i = \frac{C_i}{T}$) %

W_i : Waiting time in queue (i) Tiempo de espera en cola media de la est. i

R_i : Response time (i) Tiempo de respuesta medio de la estación i ($R_i = W_i + S_i$)

N_i : N° medio de trabajos en la estación de servicio i

Q_i : In Queue (i) N° medio de trabajos en la cola de espera de la estación de servicios i

U_i N° medio de trabajos servidos por el dispositivo, que coincide numéricamente con Utilization (i) ya que:

$$\text{Nº medio trabajos } (i) = \frac{B_i \cdot 1 + CT - B_i \cdot 0}{T} = \frac{B_i}{T} = U_i$$

Pero son conceptos distintos.

! Puede pasar que $C_i > \Delta_i$ ya que antes de montarlos ya podía haber trabajos en proceso.

Variabes globales del servidor

► Básicas

A_0 : Arrivals N° de trabajos solicitados al servidor

C_0 : Completed N° de trabajos completados en el servidor.

► Detalladas

λ_0 : Arrival rate Tasa media de llegada al servidor

X₀: Productividad media del servidor.

N₀: N° medio de trabajos en el servidor ($= N_1 + \dots + N_k$)

(Response time)

R₀ Tiempo medio de respuesta del servidor en procesar una petición.

V_i: Visit Ratio (i) Razón media de visita. Representa la proporción entre el n° de trabajos completados ^{por} ~~entre~~ el servidor y el n° de trabajos completados por la estación i. ($V_i = \frac{C_i}{C_0}$)

D_i: Demand rate (i) Cantidad de tiempo que, de media, ha estado ocupado la estación i a cada trabajo que ha procesado el servidor

$$D_i = \frac{B_i}{C_0}$$

$$D_i = V_i \cdot S_i$$



$$D_i = V_i \cdot S_i = \frac{\alpha}{C_0} \cdot \frac{B_i}{\mu_i} \Rightarrow \frac{B_i}{C_0}$$

leyes operacionales

El valor de todos los variables utilizados en el análisis operacional del intervalo de observación T. Pero además hay una serie de relaciones entre las variables que se verán según T ni suposiciones de distribución o forma donde llegar

los trabajos. Estas se denominan leyes operacionales.

Estas leyes son más útiles si se cumple el hipótesis del equilibrio de flujos, el cual dice que si el intervalo T es suficientemente largo:

$$- \bar{C} \approx A_0, \text{ es decir, } X_0 \approx A_0$$

El nº de trabajos que completa el servidor coincide aproximadamente con los solicitados. La productividad media coincide aproximadamente con la tasa de llegada.

$$- C_i \approx A_i \rightarrow X_i \approx \lambda_i \quad i = 1 \dots K$$

El nº de trabajos que completa cada estación de servicio coincide aproximadamente con los que se solicitan.

La ley de Little

Originalmente era para ver cuánta gente hay en un sitio en función de cuántos están y cuánto tiempo están ahí.

Esta ley, aplicada al entorno de los servidores, dice que independientemente de cuánto tarde el servidor en hacer un proceso, y de cómo sea el flujo de entrada, se puede establecer una relación entre X_0 y R_0

$$N_0 = \lambda_0 \cdot R_0 = X_0 \cdot R_0$$

Evidentemente, esta ley sólo es válida si el servidor está en equilibrio de flujo.

La ley no sólo puede aplicarse en general, además...

$$- \text{En una estación de servicio } N_i = \lambda_i \cdot R_i = X_i \cdot R_i$$

$$- \text{En una cola de la estación de servicio } Q_i = \lambda_i \cdot W_i = X_i \cdot W_i$$

Ley de utilización

Relaciona la utilización de un dispositivo con el nº de trabajos que es capaz de realizar por unidad de tiempo (productividad) y el tiempo que dedica a cada uno de ellos (tiempo de servicio, S_i)

$$U_i = X_i \cdot S_i = \lambda_i \cdot S_i \quad (\text{si está en equilibrio de flujo})$$

$$S_i = \frac{B_i}{T_i} = \frac{B_i/T}{C_i/T} = \frac{\lambda_i}{X_i} \Rightarrow X_i \cdot S_i = U_i \Rightarrow U_i = X_i \cdot S_i$$

Como consecuencia inmediata, la productividad media de un dispositivo está limitada a la inversa de su tiempo de servicio:

$$U_i \leq 1 \Leftrightarrow X_i \leq \frac{1}{S_i}, \forall i \in 1..K$$

Ley del flujo forzado

Flujo N° de tareas que entran o salen del dispositivo. En este caso, sería el de salida.

La ley relaciona la productividad del servidor y los dispositivos que integren el mismo

Imagine que fuera el proceso de montaje de un coche. Recorrerá 4 veces el proceso de montar las ruedas.

$$X_i = X_0 \cdot V_i \quad [= \lambda_0 \cdot V_i = \lambda_i]$$

Si hay equilibrio de flujo

$$V_i = \frac{C_i}{C_0} = \frac{B_i/T}{C_0/T} = \frac{X_i}{X_0} \Rightarrow X_0 V_i = X_i \Rightarrow \lambda_i = X_0 \cdot V_i$$

Relación Utilización - Demanda

Como consecuencia de la ley anterior, las utilizaciones de cada dispositivo son proporcionales a las demandas de servicio del mismo, cuya constante es la productividad global.

$$U_i = X_0 \cdot D_i = \lambda_0 \cdot D_i$$

$$D_i = \frac{B_i}{C_0} = \frac{\beta_i / \tau}{C_0 / \tau} = \frac{U_i}{X_0} \rightarrow X_0 D_i = U_i \Rightarrow U_i = X_0 \cdot D_i$$

Ley general del tiempo de respuesta

Proporciona una forma fácil de calcular el tiempo medio de respuesta de un servidor. Solo es válida cuando se cumple el equilibrio de flujo.

$$R_o = V_1 \cdot R_1 + V_2 \cdot R_2 + \dots + V_n \cdot R_n = \sum_{i=1}^n V_i \cdot R_i$$

$$N_o = \sum N_i \xrightarrow{\text{Ley Little}} X_0 R_o = \sum X_i \cdot R_i \xrightarrow[\text{Por tanto}]{\text{Ley Flujo}} X_0 R_o = \sum X_0 V_i \cdot R_i$$

Ley del tiempo de respuesta interactiva

Se obtiene aplicando la ley de Little a una red cerrada interactiva. En este caso, una red cerrada siempre está en equilibrio de flujo, así que siempre se podrá aplicar la ley de Little.

$$R_o = \frac{N_f}{X_0} - z$$

Para demostrarlo:

$$N_T = X_0 Z \rightarrow N_T = X_0 Z + X_0 R_0 = X_0 (Z + R_0)$$

$$N_T = X_0 R_0$$

$$N_T = N_T + N_Z$$

$$N_T = X_0 (Z + R_0) \rightarrow \frac{N_T}{X_0} = Z + R_0$$

$$\rightarrow \frac{N_T}{X_0} - Z = R_0$$

Limitaciones en el rendimiento

Cuello de botella

Todo servidor presenta alguna limitación en su rendimiento, ya que cada cola lleva a un recurso, y puede que haya un recurso que si no llega a hacerse a tiempo, podría causar una saturación en la cola y haciendo que su productividad decaiga. A este recurso que tiene más posibilidades de causar saturación se le denomina cuello de botella o bottleneck (puede haber más de uno).

La única manera de mejorar las prestaciones de un servidor de manera significativa es trabajando sobre ese cuello de botella.

► Identificación Sería aquel elemento que se saturaría primero ante una gran demanda de los recursos.

Conforme aumente la carga λ , aumentaría la demanda. La que tenga una mayor demanda D_i será ese cuello de

botella, ya que V_i depende de D_i . Esto nos permite evitir tener que llevar al límite la carga.

Como $D_i = V_i \cdot S_i$, no solo depende de la rapidez que sea un cuello de botella, sino de cuántas veces tiene que usarse.

Por ello, como notación:

D_b es el máximo D_i , $D_b = V_b \cdot S_b$

V_b es el máximo V_i , $V_b = X_b \cdot D_b$

Saturación del servidor

El servidor se saturará en cuanto el recurso con cuello de botella lo haga, ya que será el primero en alcanzar $V_b = 1$ cuando aumente la carga.

No no tiene por qué poner el máximo por cuello de cuello de botella, sino que se rompería el equilibrio de flujo (si estuviera)

En saturación, la productividad del cuello de botella estrena a su máximo

$$V_b = X_b \cdot S_b \xrightarrow{V_b=1} 1 = X_b \cdot S_b \rightarrow X_b = \frac{1}{S_b}$$

Mientras no esté saturado, habría **equilibrio de flujo**

Servidor equilibrado

Sea un caso de servidor donde todos los dispositivos son cuellos de botella, donde al aumentar la carga estos van saturándose uniformemente hasta que todos se saturan a la vez.

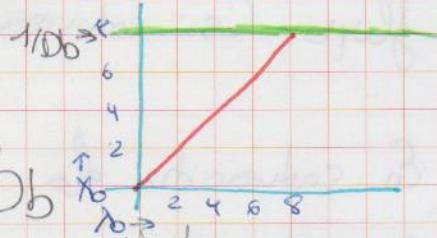
Límites del rendimiento de un servidor

Se trata de estimar las prestaciones límite de un servidor en casos extremos de cargas altas y bajas. Para ello habrá que estimar una cota superior a la productividad X_0^{\max} y una inferior para el tiempo de respuesta R_0^{\min} , que se denominan límites optimistas del rendimiento.

Límites optimistas en redes abiertas

El valor máximo de la productividad del servidor será el producido por una tasa de llegada que sature el cuello de botella.

$$\text{Si } V_b = 1, X_0^{\max} = \frac{1}{D_b}$$

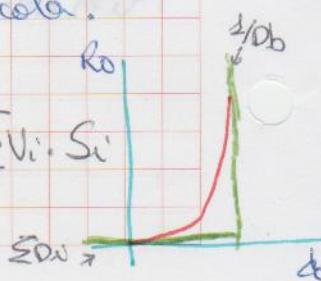


$$V_b = X_0 \cdot D_b \approx \lambda_b \cdot D_b \rightarrow V_b = X_0 \cdot D_b \rightarrow 1 = X_0 \cdot D_b \rightarrow \frac{1}{D_b} = X_0$$

El valor mínimo del tiempo de respuesta será el que experimente un trabajo sin esperas en la cola.

$$R_0 = \sum V_i \cdot R_i \Rightarrow R_0^{\min} = \sum V_i \cdot R_i^{\min} \rightarrow R_0^{\min} = \sum V_i \cdot S_i$$

$$\rightarrow R_0^{\min} = \sum D_i \Rightarrow$$



Límites optimistas en redes cerradas.

Para ello usaremos en especial la ley de Little para redes completas:

$$R_D = \frac{N_T}{X_0} - Z$$

► En valores de carga altos

Supongamos que N_T es grande. A mayor N_T , más cerca esté el cuello de botella a 1.

Si $U_b \rightarrow 1$, el valor máximo de la productividad tenderá a $1/D_b$

El valor mínimo del tiempo de respuesta usar la ley de Little, siendo así que

$$R_O \text{ tendería a } \frac{N_T}{X_0} - Z = D_b \cdot N_T - Z$$

► En valores de carga bajas

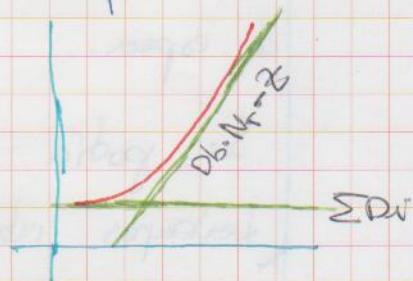
Supongamos que N_T es pequeño. A menor N_T , más cerca esté U_b a 0

El valor mínimo del tiempo de respuesta será el tiempo cuando los dispositivos no tienen color, $\sum D_i$

El valor máximo de la productividad tenderá a ~~$D_b \cdot N_T$~~

$$R_O = \frac{N_T}{X_0} - Z = R_O \cdot X_0 = N_T - X_0 Z \rightarrow X_0 (R_O + Z) = N_T \Rightarrow X_0 \rightarrow \frac{N_T}{R_O + Z}$$

$$\approx X_0 \rightarrow \frac{N_T}{\sum D_i + Z}$$



Punto teórico de saturación N_T^*

En redes cerradas, es el punto donde las asintotas coinciden.

$$\sum D_i = Db \cdot N_T^* - Z \rightarrow D = Db \cdot N_T^* - Z$$

$$\rightarrow D + Z = Db \cdot N_T^* \rightarrow \frac{D + Z}{Db} = N_T^*$$

Para $N_T > N_T^*$, las límites asintóticos están impuestos por el cuello de botella del servidor.

Propiedades

A partir de N_T^* trabajos, ya no se puede conseguir el tiempo de respuesta mínimo ya que habrá colas de espera.

Se podría decir entonces que sería el número de trabajos ideal.

Se puede derivar de la ley de Little para redes completas,

$$R_0 = \frac{N_T}{X_0} - Z \rightarrow X_0(R_0 + Z) = N_T \xrightarrow[X_0 \text{ max y } R_0 \text{ min}]{S: N_T^*}$$

$$X_0^{\max} (R_0^{\min} + Z) = N_T \xrightarrow{R_0^{\min} = \sum D_i} \frac{R_0^{\min} + Z}{X_0} = N_T^*$$

$$\xrightarrow{R_0^{\min} = \sum D_i} \frac{\sum D_i + Z}{Db} = N_T^* \xrightarrow{\sum D_i \geq 0} \frac{D + Z}{Db} = N_T^*$$

Técnicas para mejorar las prestaciones

Para mejorar las prestaciones habría que mejorar el cuello de botella, bien ajustando ese recurso, o comprando nuevo hardware.

Sintonización o ajuste (tuning)

Es la optimización del funcionamiento de componentes existentes.

En caso del hardware, podríamos configurar voltajes, frecuencias y otros parámetros de la placa.

En las aplicaciones, podemos usar archivos de configuración o hacer profiling si tenemos el código fuente.

A nivel de SO, habría que ir configurando políticas de gestión de procesos, memoria, almacenamiento y red.

Algunas inconvenientes serían:

Possible alteración de la fiabilidad

Requiere conocer muy bien el sistema operativo y el funcionamiento de los componentes hardware.

Deberíamos realizar tests estadísticos para ver qué factores realmente influyen en las prestaciones.

Actualización y/o ampliación

Se trata de reemplazar recursos con cuello de botella por otros más rápidos, para bajar el tiempo de servicio D_b; o añadir más dispositivos para bajar la razón de visita V_b.

Algunos problemas son:

la compatibilidad de los nuevos elementos con los existentes.

Comprobar las facilidades del servidor para depurarse actualizaciones (de cara a la escalabilidad).

Algoritmos de resolución de modelos de redes de colas

Requeriremos una metodología o algoritmo para resolver modelos de redes de colas. Suponemos conocido

K N° de estaciones

Per cada estación,

V_i Su razón de visita

Si su tiempo de servicio

Si la red es abierta...

de la tasa de llegada

y si es cerrada...

N_T N° total de trabajos.

\bar{z} Tiempo medio de reflexión

Hipótesis del "peor escenario posible"

Para el caso de redes abiertas en equilibrio de flujo; suponemos que el momento en el que llega un trabajo no depende de cuándo llegaron los anteriores.

También suponemos que todos los llegados y ~~solicitudes~~ tiempos de servicio se rigen por una distribución exponencial $f(x) = \lambda e^{-\lambda x}$

En ese caso se puede demostrar que el tiempo de espera viene dado por $W_i = N_i \cdot S_i$. Si hay N_i trabajos que están en proceso en trabajo de la cola. Si, en el peor caso se tardaría S_i ese tiempo.

El tiempo de respuesta sería $R_i = W_i + S_i$

En el caso de la hipótesis, $R_i = W_i + S_i = N_i S_i + S_i$

Y si aplicamos la ley de Little a una estación de servicio

$$N_i = \lambda_i \cdot R_i \approx X_i R_i$$

$$R_i = N_i \cdot S_i + S_i \xrightarrow{\text{Ley Little}} R_i = X_i R_i S_i + S_i \rightarrow$$

$$R_i - X_i R_i S_i = S_i \rightarrow R_i (1 - X_i S_i) = S_i \rightarrow$$

$$R_i = \frac{S_i}{1 - X_i S_i} \xrightarrow{\substack{\text{Ley de Fort.} \\ D_i = V_i \cdot S_i}} R_i = \frac{S_i}{1 - \lambda_i} \xrightarrow{\substack{\lambda_i = D_i / X_i \\ D_i = X_i D_i}} R_i = \frac{S_i}{1 - X_i D_i} \xrightarrow{\substack{\lambda_i = D_i \\ X_i = D_i}} R_i = \frac{S_i}{1 - D_i}$$

$$R_i = \frac{S_i}{1 - \lambda_i D_i} \xrightarrow{D_i = V_i \cdot S_i} R_i = \frac{S_i}{1 - \lambda_i V_i \cdot S_i}$$

Resolución de redes de colas

(en equilibrio de flujo)

1 Calculamos la demanda media de servicio de cada estación (D_i)

2 Calculamos el tiempo medio de respuesta de cada estación usando la hipótesis del peor caso. (R_i)

3 Calculamos el tiempo medio de respuesta del servidor. (R_o)

! El resto de variables se pueden calcular usando sus expresiones habituales

Resolución de redes cerradas

El algoritmo supone conocidos V_i , S_i , N_T y f . En este caso, el algoritmo es iterativo, donde se van resolviendo la red para valores incrementales del n.º de trabajos hasta N_T .

$$N_T \text{ con la hipótesis } R_i(N_T) = (N_i(N_T - f) + 1) \cdot S_i$$

donde $n_T \in \{1 \dots N_T\}$

6. Pliegos de prescripciones técnicas

Supongamos que somos los responsables de los Servicios de Informática de una empresa pública y nuestra labor ahora es buscar contratar a otra empresa para realizar una determinada instalación informática.

Evidentemente, no podemos llamar a lo informático de confianza, decírle cuatro cosas y ya. Deberíamos pues estipular un documento donde se detalla exactamente lo que necesitamos y los requisitos técnicos mínimos de cada material, obra o servicio que se emplee.

Junto con ello, hay que redactar otro documento que especifique todas las condiciones administrativas y jurídicas que han de cumplirse durante el contrato.

No se puede contratar a cualquiera, sino que emplearemos una licitación.

En caso de empresas públicas, debemos garantizar/potenciar:

El uso eficiente del dinero público

La transparencia

La igualdad de oportunidades

La protección del medio ambiente

El empleo de personas con discapacidad

El empleo juvenil.

Algunos términos

Licitación : Se trata de una subasta pública para la realización del proyecto.

Licitador o licitante Las empresas que participan en la licitación ofreciendo ejecutar un servicio a cambio de dinero o beneficios

Contrato: Documento que recoge las condiciones de un pacto entre ambas partes sobre una materia que es obligatorio cumplir.

Contractante Aquel que contrata a la empresa.

Contractista Es aquél licitante que resulta adjudicatario de la licitación y quien se encargará de ofrecer el servicio establecido por contrato.

Tipos de contratos

De obras , aquellas que han de ejecutar una obra de construcción o ingeniería civil cumpliendo una función económica o técnica , y que tenga por objeto un bien mueble.

De suministro , aquellos que tienen por objeto adquirir o arrendar , con o sin opción a compra , productos o bienes muebles.

De servicios , aquellos que se prestan a ser consistentes en el desarrollo de una actividad u obtener un resultado distinto de una obra o suministro . Ej ; Mantenimiento , reparaciones...

Algunos términos

Licitación : Se trata de una subasta pública para la realización del proyecto.

Licitador o licitante Las empresas que participen en la licitación ofreciendo ejecutar un servicio a cambio de dinero o beneficios

Contrato: Documento que recoge las condiciones de un pacto entre ambas partes sobre una materia que es obligatorio cumplir.

Contractante Aquel que contrata a la empresa.

Contractista Es aquél licitante que resulta adjudicatario de la licitación y quien se encargará de ofrecer el servicio estipulado por contrato.

Tipos de contratos

De obras , aquellas que han de ejecutar una obra de construcción o ingeniería civil cumpliendo una función económica o técnica , y que tenga por objeto un bien mueble.

De suministro , aquellos que tienen por objeto adquirir o arrendar , con o sin opción a compra , productos o bienes muebles.

De servicios , aquellos que se prestan a ser consistentes en el desarrollo de una actividad u obtener un resultado distinto de una obra o suministro . Ej ; Mantenimiento , reparaciones...

Licitaciones

Aertura y publicidad del expediente

En el caso de administraciones públicas requiere tramitar previamente el expediente de licitación, que lo iniciará el contratante justificando la necesidad del contrato, y junto con él, el pliego de condiciones jurídicas, técnicas y administrativas.

Este se publica en el BOE o BOJA, según sea a nivel nacional o autonómico.

Información de los anuncios

Datos de identificación del contratante.

Descripción general de las obras, suministros y/o servicios, y posibles variantes.

Presupuesto total estimado del contrato. Si está dividido en lotes, el de cada lote.

Calendario de entrega para los suministros u obras; o de la prestación de servicios

Criterios usados para adjudicar el contrato

Plazo de recepción de ofertas

CPV

Dirección de Internet para consultar los pliegos de condiciones.

Pliego de condiciones

El pliego de condiciones es un conjunto de artículos o cláusulas que regulan los derechos, responsabilidades, obligaciones y garantías mutuas entre los distintos agentes involucrados en el contrato. Estas exigencias son de carácter jurídico, económico, administrativo y técnico; que rigen la ejecución del trabajo.

Se divide en:

PCAP : Pliego de cláusulas administrativas particulares

PPT : Pliego de prestaciones técnicas

PCAP

Estos pliegos contendrán aquellas declaraciones jurídicas, económicas y administrativas específicas del contrato y del procedimiento y forma de adjudicación.

De carácter general contendrán los siguientes datos

Objeto del contrato y códigos CPV

Presupuesto de la licitación

Plazo de ejecución o de duración del contrato

Procedimiento y forma de adjudicación del contrato

Documentos a presentar por el licitador

Derechos y obligaciones de ambas partes

Régimen de pagos

Opcionalmente contendrá información sobre confidencialidad, modificaciones del contrato, mordencias en la ejecución del contrato como motivos de resolución de contrato y la inspección de los contratos,

PPT

Este contendrá, al menos, los siguientes elementos:

Descripción y especificaciones técnicas de los bienes o prestaciones del contrato.

Las especificaciones técnicas definen las características exigidas de un producto o servicio como estíndares, niveles de calidad, rendimiento, instrucciones de uso, accesibilidad universal, comportamiento ambiental, seguridad, etiquetado...

En su caso, descripción y especificaciones de las variantes

Precio de cada una de las unidades que se descompone en el presupuesto y nº de unidades a suministrar.

En ningún caso en este pliego deberá haber cláusulas que pertenezcan al PCAP

Reglas para redactar un PPT

En carácter general, se definirán criterios de sostenibilidad, protección ambiental, accesibilidad universal e igualdad de oportunidades.

No habrá referencia a una fabricación o marca concretas para evitar desfavorecer a otros productos. Si no es posible evitarlo, se debe añadir "o equivalente"

Se formularán de alguna de estos maneras (o una mezcla)

- En términos de rendimiento o requisitos funcionales
- En referencia a especificaciones técnicas de normas nacionales o internacionales.

Secundarios, habituales de un PPT

Objeto, contexto en el que se ubica el pliego de prescripciones técnicas y su objetivo principal.

Descripción general de la obra, servicio o producto a realizar, incluidas las posibles variantes.

Especificaciones técnicas mínimas de cada uno de los elementos de la obra / servicio / suministro a realizar y sus variantes.

Precio de cada unidad o lote que descomponer el presupuesto y nº de unidades a suministrar.

Garantías de los componentes y obras a realizar.

Servicio de ayuda o soporte técnico: Tipo, disponibilidad...

Documentos, manuales y formación que se debe aportar para el uso y mantenimiento de los bienes.

Instalación y puesta a punto para comprobar que los requisitos que se deben cumplir para considerar que lo contratado se ha instalado correctamente.

Adjudicación de los contratos

Entrega de los objetos

Las proposiciones de los interesados deberán ajustarse a lo previsto en los pliegos de condiciones. Presentar esa proposición implica que estén de acuerdo con las condiciones del contrato.

Sin excepciones. Si se quisiera expresar desacuerdo, se trataría de impugnar previamente.

Una oferta contiene habitualmente estos puntos:

Documentación sobre la empresa donde se verifique que cumple los requisitos legales y administrativos de los pliegos

Memoria técnica con la solución propuesta. Aquí si se puede hablar de marcas.

Precio por el que se compromete a realizar los trabajos indicados en la memoria técnica

Valoración de la oferta

De haber sólo un criterio de adjudicación, será el proyecto más barato. Los criterios básicos de un contrato lo determinarán el contratante y se detallarán en los PCAP.

No se puede declarar desierto una licitación cuando existe alguna oferta o proposición que sea admisible de acuerdo a los criterios del pliego.

Firma del contrato

El documento de formalización será firmado por el adjudicatario y se unirá al mismo, como anexo, un ejemplar de los PCAP y del PPT. En este documento no se puede incluir estipulaciones que establezcan derechos y obligaciones para las partes distintas de lo dicho en los pliegos.

los efectos de los contratos administrativos se regirán por las normas de la Ley 9/2017 de Contratos del Sector Público y por los pliegos de cláusulas administrativas y de prescripciones técnicas o documento técnico que lo sustituya.

! Ten cuidado con el lenguaje! Unquier ambigüedad en los términos podría llevar a que se deba aceptar lo que ofrece el licitante (si ya hubiera alguien dispuesto a ello)

Memoria técnica de la propuesta de cada licitante

Cada licitante debe entregar la documentación técnica solicitada dentro del plazo establecido en la convocatoria.

La memoria técnica de su propuesta deberá ser compatible con el PPT, y su coste debe respetar el presupuesto de la licitación. En esta memoria, el licitante detalla su solución concreta, donde detalla exhaustivamente de todos componentes, obra y/o servicio a desarrollar; para poder competir por conseguir el contrato.