Gran parte del trabajo que desarrolla un administrador de sistemas es conseguir un buen rendimiento del sistema que administra, pero …

¿Qué entendemos por rendimiento?

Rendimiento (según el Diccionario de la Real Academia Española de la Lengua)**.-** "Proporción entre el producto o el resultado obtenido y los medios utilizados".

En nuestro caso, y en general,

- "*producto o resultado obtenido*": prestaciones que da la ejecución de un conjunto de programas (sistema de información, sistema operativo, herramientas software tales como compiladores, gestores de bases de datos, etc.)

- "*medios utilizados*": en general, los recursos hardware (cpu, unidades de almacenamiento, …) y/o de red. En ocasiones puede ser otro software (compiladores, sistemas operativos, gestores de bases de datos, …).

El "conjunto de programas" que se ejecuta para ver el rendimiento de un sistema se denomina "**carga**". Según sea el uso de los "medios" por parte de la "carga" se obtendrán unas prestaciones u otras.

Las “prestaciones” de un sistema se suelen considerar, en general, en términos de tiempo de respuesta: tiempo de ejecución, tiempo de acceso a información, tiempo de transmisión en una red, … y/o utilización de recursos (memoria principal, disco, …).

Suele haber dos visiones distintas, y en ocasiones distantes, del rendimiento de un sistema informático: la del administrador del sistema y la del usuario. **Por eso es muy importante establecer un Acuerdo de Nivel de Servicio**.

Habitualmente el usuario siempre desea que mejoren las prestaciones de un sistema, lo que puede conducir a que el administrador deba mejorar el rendimiento.

Para mejorar el rendimiento de un sistema hay que:

1.- evaluar el rendimiento con una cierta carga,

2.- llevar a efecto las modificaciones oportunas,

3.- volver a evaluar el rendimiento con la misma carga para comprobar si ha mejorado.

La carga (1)

Carga: Conjunto de solicitudes de servicio que realizan los usuarios de un sistema informático en un cierto intervalo de tiempo.

En consecuencia, no se puede hablar de:

- prestaciones de un sistema,
- medida de prestaciones,
- comparación entre distintos sistemas, o
- comparación entre distintas configuraciones de un mismo sistema

sin tomar en consideración la carga durante el intervalo de tiempo que dura la observación y medida de prestaciones.

Por otra parte, la carga debe ser seleccionada en función de los **objetivos del estudio** a realizar, esto es, en función de las prestaciones que se desea observar y medir. Estos objetivos determinarán cómo caracterizar la carga para ese estudio concreto, técnicas de estudio e instrumentos de medida a utilizar y parámetros a medir.

La carga (2)

Carga de prueba: Carga que se procesa durante el intervalo de medida.

Para los estudios de comparación o sintonización (ajustes) se debe reproducir siempre la misma carga, pero esto es difícil aunque se ejecuten los mismos programas.

La carga de prueba puede ser:

- Real.
- Sintética:
  - Natural (utiliza extractos de carga real)
  - Híbrida
- Artificial: no utiliza extractos de carga real. Puede ser
  - Ejecutable: mix, kernels, programas sintéticos, benchmark.
  - No ejecutable: distribuciones de probabilidad, modelos de colas, …

La carga (3)

Carga de prueba real.- Es la carga que realmente se está ejecutando en un sistema con todos sus usuarios y aplicaciones. Lo único que hay que seleccionar son las sesiones de medida para que sean representativas del funcionamiento normal o de la situación en estudio.

Características:

- Barata
- Representativa
- La situación que se desea reproducir puede no ocurrir durante las sesiones de medida.
- Poco reproducible porque es difícil crear a voluntad las mismas situaciones.
- Poco flexible ya que no se puede modificar.
- Puede haber problemas de confidencialidad.
- No se puede usar en problemas de selección.

La carga (4)

Carga de prueba sintética natural.- También llamada benchmark (aunque en muchas ocasiones se llama benchmark a cualquier carga de prueba). Es un subconjunto de programas extraídos de la carga real. Aunque coincida plenamente con la carga real **no es** la carga real porque los usuarios no la están utilizando para realizar trabajo útil.

Características:

- Se suele usar en estudios de ampliación, reposición y selección.
- Dificultades de uso:
  - Prioridades de ejecución.
  - Parámetros diferentes en las monitorizaciones.
  - Parámetros o ampliaciones de sistemas operativos y otros programas como compiladores.

Carga de prueba sintética híbrida.- Son mezcla de programas extraídos de la carga real y de carga artificial. Permite modelar parte de carga existente y parte inexistente. Ejemplos: ampliaciones de carga (carga real + nueva carga inexistente) , programas confidenciales, etc.

La carga (5)

Carga artificial.- Se le llama “artificial” porque se construye para ser un modelo de carga. No utiliza componentes de la carga real. Sirve tanto para cargar el sistema real como un modelo del sistema. Puede ser:

- Ejecutable
  - MIX de instrucciones.
  - Mix de sentencias.
  - Kernels.
  - Programas sintéticos.
  - Secuencias conversacionales.
  - Benchmarks.
- No ejecutable

La carga (6)

Carga artificial ejecutable MIX de instrucciones.- Se usa para medir la frecuencia de aparición de las diferentes instrucciones. Un modelo representativo de la carga real es una secuencia de instrucciones tal que la proporción y frecuencia de aparición sean las mismas.

Características:

- Son muy dependientes del sistema:
  - Gestión de memoria.
  - Secuenciamiento de instrucciones.
  - Manejo de direcciones.
- Los tiempos dependen del procesador.
- Se usaron para comparar procesadores.

La carga (7)

Carga artificial ejecutable MIX de sentencias.- Se compone de sentencias de un lenguaje de alto nivel.

Características:

- Son más independientes del hardware pero muy dependientes del compilador. Hay que controlar las optimizaciones del compilador y las librerías utilizadas.
- La frecuencia de aparición de las sentencias se puede medir de forma:
  - Estática: recuento sobre un listado del programa.
  - Dinámica: durante la ejecución del programa. Es más fiable pero más costosa.

La carga (8)

Carga artificial ejecutable Kernel.- Representan un fragmento de la carga que se corresponda con su parte más característica para el estudio que se está realizando. Son programas cerrados que tienen un consumo de recursos conocido (ejemplos: función de Ackermann, inversión de matrices, algoritmos de ordenación, etc.) y que se seleccionan según su similitud con los programas de la carga real.

Precaución: Al ser parte de una aplicación, puede que quepan enteros en una caché, por lo que los resultados de la prueba pueden ser buenos pero muy diferentes a si se ejecutase la aplicación completa.

Carga artificial ejecutable "programas sintéticos".- Son programas que no realizan trabajo útil y sólo consumen recursos. Por ejemplo:

for i:= 1 to n_1 do

"consumir CPU"

for i:= 1 to n_2 do

"consumir E/S"

donde los parámetros n_1 y n_2 determinan el consumo de CPU y E/S respectivamente.

La carga (9)

Carga artificial ejecutable "secuencias conversacionales".- Son cargas transaccionales o conversacionales que usan otros ordenadores, o el propio sistema a medir, para simular las peticiones. Se suelen hacer uno más guiones que representen el comportamiento de los usuarios. Ejemplo de guión:

Conectarse al sistema.

Editar un fichero y añadir líneas.

Compilar.

Editar.

Compilar.

Ejecutar.

Aquí podríamos tener dos sistemas conectados: uno genera las peticiones como si fuesen usuarios y el otro sería el sistema bajo estudio que se está midiendo.

La carga (10)

Carga artificial ejecutable Benchmarks.- Son programas construidos con alguna de las técnicas anteriores que producen una **carga genérica que no pretende representar una carga específica y concreta**. Como es lógico, su gran inconveniente es que suelen ser muy poco representativas de una carga concreta.

Las aplicaciones de este tipo de carga son muy variadas:

- comparación sistemas,
- sintonización de sistemas,
- planificación de capacidad,
- comparación de compiladores,
- …

La carga (11)

Carga artificial ejecutable Benchmarks.

Pasos a seguir:

- Fijar los objetivos de las mediciones.
- Seleccionar los componentes apropiados para cubrir la finalidad del estudio.
- Comprobar los aspectos del sistema que pueden influir en las prestaciones.
- Presentar los resultados, incluida toda la información anterior y la fecha de aplicación.
- Analizar los resultados para comprender las razones de los índices obtenidos.

Factores que influyen:

- Sistema operativo.
- Compiladores.
- Lenguajes de programación.
- Librerías.
- Memoria caché.
- Ejecución correcta del bench (los resultados obtenidos deben ser los esperados).

La carga (12)

Carga artificial ejecutable Benchmarks.

Errores comunes (1)**:**

- Representar en la carga de prueba sólo comportamientos medios. Si se representan valores medios se pueden dar sincronizaciones en los consumos que no son reales.
- Controlar de manera inadecuada el nivel de carga. Modificamos ciertos parámetros esperando que los resultados no cambien.
- Ignorar los efectos de la caché. Las prestaciones de las memorias caché son muy sensibles al orden en el que se hacen las peticiones. En los estudios de caracterización de la carga se suele perder información relativa al orden en que se producen los consumos.
- Ignorar la sobrecarga introducida en la monitorización.

La carga (13)

Carga artificial ejecutable Benchmarks.

Errores comunes (2)**:**

- No validar las medidas. Los errores en las medidas suelen pasar inadvertidos.
- No asegurarse de que se dan las mismas condiciones iniciales.
- No medir las prestaciones del transitorio. Se suelen tomar las medidas una vez que el sistema funciona de forma estable. A veces interesa medir el coste de las operaciones de inicio.
- Almacenar muchos datos pero realizar poco análisis. Normalmente la toma de datos y la experimentación ocupan la mayor parte del tiempo del estudio. Se debe planificar el tiempo para que el análisis sea el adecuado.

La carga (14)

Cargas artificiales no ejecutables (1)**.-** Se construyen tomando como base medidas o procedimientos estadísticos utilizados en modelos de redes de colas y se resuelven mediante simulaciones o modelos analíticos.

Para “construir” estas cargas se selecciona, o construye a propósito, un modelo “matemático” genérico que se ajuste al tipo de carga real que tengamos y al estudio de prestaciones concreto que vayamos a realizar. Estos modelos suelen estar basados en distribuciones de probabilidad y modelos de colas.

Por su carácter genérico, el modelo de carga de prueba dependerá de unos parámetros. Al asignar valores a los parámetros el modelo se instanciará en una carga de prueba concreta.

Los valores de los parámetros se obtienen haciendo mediciones y estimaciones estadísticas sobre la carga real, de manera que la carga de prueba sea lo más representativa posible de la carga real. No obstante, obtenidos los valores de los parámetros e instanciado el modelo, se debe validar esta representatividad tanto para la carga real como para el tipo de estudio de prestaciones que vayamos a realizar, pues pudiera ocurrir que el modelo de partida no sea el adecuado.

Finalmente, la estimación de los diversos índices de prestaciones se obtiene mediante simulación o aplicando métodos matemáticos de carácter analítico.

La carga (15)

Cargas artificiales no ejecutables (2).-

Características de los modelos de carga:

- Repetibles o reproducibles.
- Compactos.
- Flexibles. Suelen ser fáciles de modificar para ajustarlos a variaciones de la carga real.
- Evitan problemas de privacidad y seguridad.
- Independientes del sistema: la representatividad del modelo no varía al cambiarlo de sistema.
- Deben ser representativos.
- Deben ser compatibles con el tipo de estudio a realizar.

¿Qué es evaluar el rendimiento de un sistema?

Saber de qué manera una cierta carga está utilizando “los medios” del sistema. En la mayor parte de las ocasiones consiste en saber cómo una determinada carga (conjunto de programas) está utilizando los recursos hardware y de red.

¿Para qué sirve la evaluación?

Para cuestiones tan variadas como:

- Optimizar el diseño y construcción de un sistema informático evaluando las diferentes alternativas, comparándolas y optando por la mejor.
- Seleccionar un sistema informático según sea su relación rendimiento/precio.
- Ajustar un sistema informático reconfigurando el software-hardware para mejorar su rendimiento.
- Predecir la máxima carga soportable por un sistema.
- …

por eso, el rendimiento de un sistema siempre se ha de evaluar para resolver una necesidad concreta.

¿Cómo se evalúa el rendimiento de un sistema?

Midiendo sus prestaciones para una cierta carga, pero es difícil:

- los índices utilizados pueden cambiar según el tipo de estudio que se haga o el sistema sobre el que se apliquen, lo que dificulta las comparaciones,
- los instrumentos de medida y la forma en la que se hacen las medidas pueden cambiar,
- **la carga puede no ser estática**, por lo que en el estudio de prestaciones hay que asegurarse de que todos los experimentos se llevan a cabo con la misma carga.

La carga "real" de un sistema suele ser muy variable, pero, para evaluar el rendimiento de un sistema, **la carga siempre ha de ser la misma**. Para medir las prestaciones se usa una **carga de prueba** que puede ser la carga real (si es estática) o una carga sintética. En general se suele usar carga sintética porque permite experimentar y reproducir situaciones sin modificar el sistema real. La carga de prueba suele estar formada por componentes. En cualquier caso siempre debe ser **representativa** de la carga real **y reproducible**.

Para asegurar que la carga de prueba es representativa de la carga real se deben tomar ciertas medidas sobre ambas y comprobar que tales medidas coinciden.

¿Qué se mide? (1)

Medidas para los componentes de la carga:

- Tiempo de CPU por trabajo.
- Número de operaciones de E/S por trabajo desglosado por dispositivos.
- Tiempo de CPU entre operaciones de E/S.
- Mezcla de instrucciones.
- Prioridad de ejecución.
- Memoria requerida.
- Ficheros en disco.
- Localidad de las referencias: tiempo en el que las referencias permanecen en una página.

Medidas para el conjunto de la carga:

- Tiempo entre llegadas de dos peticiones.
- Frecuencia de llegadas (inversa al tiempo entre llegadas)
- Distribución de los trabajos: proporción de los diferentes componentes de la carga.
- Para cargas interactivas con el usuario:
  - Tiempo de reflexión del usuario: tiempo para generar una nueva petición.
  - Intensidad del usuario: relación entre el tiempo de proceso de una petición y el tiempo de reflexión.
  - Número de usuarios trabajando simultáneamente en un instante dado.

¿Qué se mide? (2)

Medidas directamente relacionadas con las prestaciones: (1)

- Características físicas del sistema.
- Condiciones en las que opera el sistema durante el estudio.
- Índices externos de prestaciones (variables que percibe el usuario).
- Índices internos de prestaciones (variables que percibe el administrador).

Índices externos de prestaciones (variables que percibe el usuario):

- Productividad (throughput): trabajo útil realizado por unidad de tiempo (programas ejecutados, páginas servidas por un servidor web, correos procesados por un servidor de correo, …).
- Capacidad: máxima productividad que puede obtenerse.
- Tiempo de respuesta.

¿Qué se mide? (3)

Medidas directamente relacionadas con las prestaciones: (2)

Índices internos de prestaciones (variables que percibe el administrador):

- Factor de utilización: tiempo de uso real de un componente.
- Solapamiento: tiempo en el que varios componentes se usan simultáneamente.
- Nº de accesos por unidad de tiempo a un recurso (servidor web, de base de datos,…)
- Sobrecarga (overhead): carga procesada pero no solicitada expresamente por los usuarios.
- Factor de carga de multiprogramación: relación entre el tiempo de respuesta de un trabajo en un entorno de multiprogramación y en un entrono de monoprogramación.
- Factor de ganancia de multiprogramación: relación entre el tiempo total de ejecutar un conjunto de trabajos en multiprogramación y en monoprogramación.
- Frecuencia de fallo de página: número de fallos de página que se producen por unidad de tiempo en un sistema de memoria virtual paginada.
- Frecuencia de swapping: número de programas expulsados de memoria por unidad de tiempo.

¿Qué se mide? (4)

Medidas no directamente relacionadas con las prestaciones:

- Fiabilidad: probabilidad de funcionamiento correcto en un intervalo de tiempo.
- Disponibilidad: probabilidad de que el sistema esté funcionando en un cierto instante.
- Seguridad: probabilidad de que el sistema funcione correctamente sin perjuicios para nadie o sufra averías sin comprometer a nadie.
- Mantenimiento de servicios: en sistemas tolerantes a fallos, probabilidad de mantener un nivel de servicios (prestaciones) después de un fallo.
- Facilidad para llevar a cabo el mantenimiento del sistema.

Sesión de medida.- Intervalo de tiempo en el que se toman medidas. Puede no ser continuo.

Normalmente habrá unas sesiones de medida destinadas a ajustar la carga de prueba y otras para tomar las medidas que realmente interesan aplicando esa carga de prueba. En ambos casos es aconsejable realizar varias sesiones para evitar que incidentes particulares de una sesión se tomen como comportamiento general del sistema.

¿Cómo se puede mejorar el rendimiento de un sistema? (1)

Primero hay que determinar dónde está el problema y después hacer los cambios oportunos en el sistema:

- Usuario: modificar el código de los programas que constituyen la carga real (problemas de complejidad temporal o espacial, localidad, etc.).
- Administrador: reconfigurar el software mediante cambios en los parámetros del sistema operativo, gestor de bases de datos, etc.
- Administrador: modificar las políticas de gestión del sistema operativo, gestor de bases de datos, etc.
- Administrador: redistribuir la carga para utilizar el máximo de recursos al mismo tiempo y quitar trabajo a los más solicitados (cuellos de botella).
- Administrador: reconfigurar el hardware, y/o
- Administrador: reconfigurar la red.

Se debe estudiar particularmente si el problema se puede solucionar sin adquirir y cambiar componentes hardware o software.

¿Cómo se puede mejorar el rendimiento de un sistema? (2)

Ejemplo.- Algunos parámetros del sistema operativo son:

- Prioridad.
- Factor de multiprogramación: número máximo de trabajos simultáneos en memoria.
- Tamaño de la partición de memoria: cantidad fija de memoria principal asignada a una cola de trabajos.
- Frecuencia máxima de fallo de página: si se sobrepasa se manda el proceso a disco para no aumentar la sobrecarga.
- Índice de supervivencia de páginas: si una página no se ha usado durante un cierto tiempo se puede devolver a disco.

Técnicas de evaluación del rendimiento

- **Monitorización.-** Uso de herramientas de medida sobre el sistema real.
- **Referenciación** (benchmarking).- Comparación del rendimiento entre dos configuraciones de un mismo sistema o entre dos sistemas distintos. Suelen ser las técnicas más sencillas y, por tanto, utilizadas. Para que sea significativa, se debe cuidar especialmente que la carga de prueba sea representativa de la carga real y que las condiciones de ejecución de la carga sean las mismas.
- **Modelado.-** Construcción de un modelo que reproduzca el comportamiento del sistema
  - **Métodos analíticos.-** Construyen un modelo matemático del sistema basado normalmente en Teoría de Colas, Cadenas de Markov, Redes de Petri, … Su aplicación está muy limitada.
  - **Modelos de simulación.-** Construyen un programa que intenta reproducir el comportamiento del sistema. Son muy difíciles de validar y suelen exigir mucho tiempo de desarrollo y ejecución. Se puede simular casi todo. Los modelos se programan en lenguajes de programación convencionales o lenguajes específicos de simulación según sea el problema a resolver.

Técnicas de evaluación del rendimiento

Monitorización (1)

La medida de un índice puede variar de unas ejecuciones a otras por la imposibilidad de repetir con exactitud las condiciones de carga y el instante de ejecución de la carga en el que se hace la medida. En tales casos se realiza un seguimiento y visualización del comportamiento del sistema, una monitorización, en lugar de una medición.

Monitor.- Herramienta para observar la actividad de un sistema mientras procesa una carga:

- observa su comportamiento,
- recoge datos estadísticos sobre la ejecución de los programas,
- analiza esos datos, y
- presenta resultados.

Técnicas de evaluación del rendimiento
Monitorización (2)

Características de un monitor.-

- Sobrecarga/interferencia: el monitor consume sus propios recursos.
- Precisión: error que pueden llevar los datos registrados.
- Resolución: frecuencia máxima a la que se pueden detectar y registrar los datos.
- Dominio de medida: características que tiene capacidad de medir.
- Anchura de entrada: número máximo de registros que puede hacer en paralelo.
- Capacidad de reducción: capacidad para analizar, procesar y empaquetar los datos.
- Compatibilidad: capacidad para adaptarse a entornos o requerimientos distintos.
- Facilidad de instalación: facilidad para ser instalado y retirado
- Facilidad de uso
- Precio: de compra, instalación, mantenimiento, operación.

Técnicas de evaluación del rendimiento
Monitorización (3)

Clasificación de los monitores.-

Según su implantación:

- Monitores software
- Monitores hardware

Según su mecanismo de activación:

- Monitores de eventos.- Se activan cuando suceden ciertos eventos. Permiten hacer trazas de lo que ocurre.
- Monitores de muestreo.- Se activan a intervalos de tiempo por interrupciones de reloj.

Según la forma de presentar los resultados:

- Monitores en línea.- Presentan datos parciales según se van produciendo.
- Monitores batch.- Los datos se analizan y presentan cuando termina la ejecución de la carga.

Técnicas de evaluación del rendimiento
Monitorización (4)

Presentación de resultados.-

Los resultados deben presentarse de una forma clara, concisa, precisa, simple y fácil de usar. Para facilitar su interpretación se suele hacer por medio de gráficas.

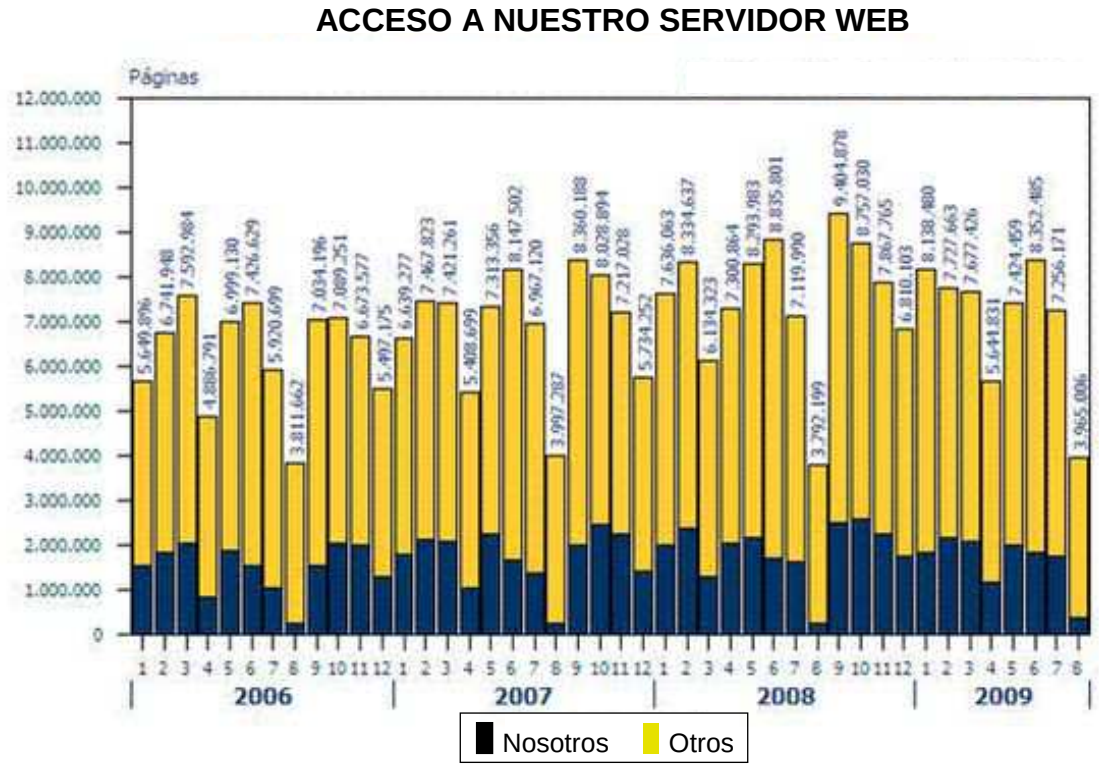
Las gráficas más usadas son:

- Diagramas de barras
- Diagramas de Gantt
- Gráficos de Kiviat
- Gráficos de Kiviat, versión Kent
- Representación funcional

Técnicas de evaluación del rendimiento

Monitorización (5)

Presentación de resultados.- Diagramas de barras

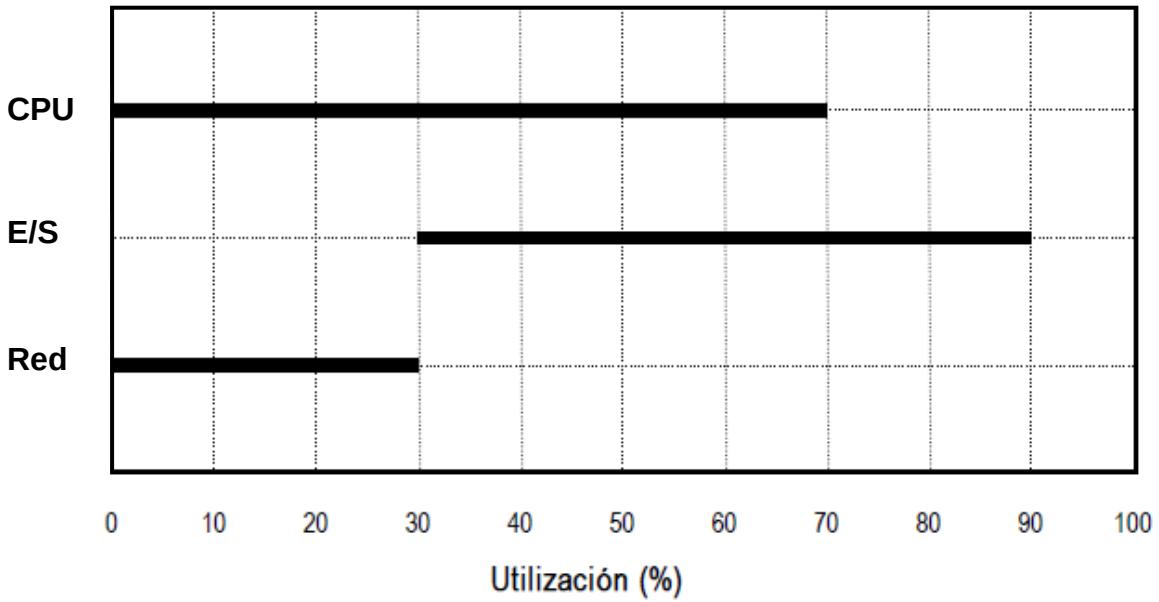


Técnicas de evaluación del rendimiento

Monitorización (6)

Presentación de resultados.- Diagramas de Gantt

Estos diagramas permiten representar el uso de varios recursos y su solapamiento



Técnicas de evaluación del rendimiento

Monitorización (7)

Presentación de resultados.- Gráficos de Kiviat

Consisten en un círculo cuyos radios representan índices de prestaciones. Partiendo del centro, sobre cada radio se marca el valor que toma el correspondiente índice, de manera que la longitud total del radio representa el valor máximo que pueda tomar.

El funcionamiento “ideal” del sistema tiene una forma determinada y, por comparación con esta forma, al ver el gráfico se observa rápidamente lo bien o mal que está funcionando y cuales son las partes que se deben mejorar.

En ese funcionamiento ideal puede haber índices “beneficiosos” cuyos valores deban ser altos (a mayor valor mejores prestaciones), e índices “perniciosos” cuyos valores deban ser bajos (a menor valor mejores prestaciones.

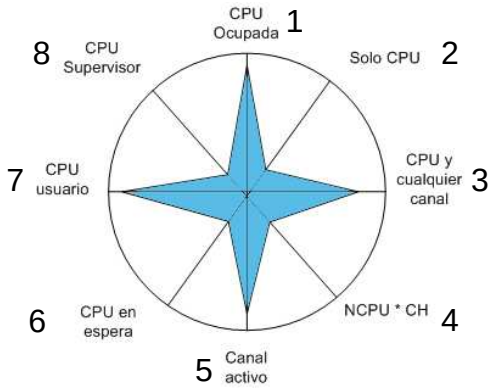
Técnicas de evaluación del rendimiento

Monitorización (8)

Presentación de resultados.- Gráficos de Kiviat, versión Kent

1. Se representa un número par de índices.
2. Se divide el círculo en tantos sectores iguales como índices a representar.
3. Se alternan índices "beneficiosos" con índices "perniciosos" comenzando por uno beneficioso.

En un funcionamiento "ideal" el gráfico tendrá forma de estrella.



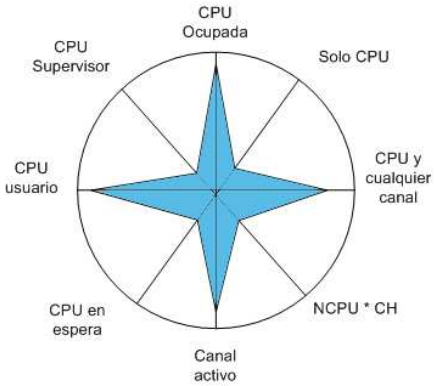
Funcionamiento ideal

1.CPU ocupada o activa.*(CPU)*
2.Sólo CPU ocupada. *(CPU*NCH)*
3.Solapamiento de CPU y canal.*(CPU * CH)*
4.Sólo canal ocupado sin solape en la CPU. *(NCPU * CH)*
5.Cualquier canal ocupado. *(CH)*
6.CPU en estado de espera. *(NCPU)*
7.CPU atendiendo a programas de usuario. *(CPU usuario)*
8.CPU en estado supervisor. *(CPU supervisor)*

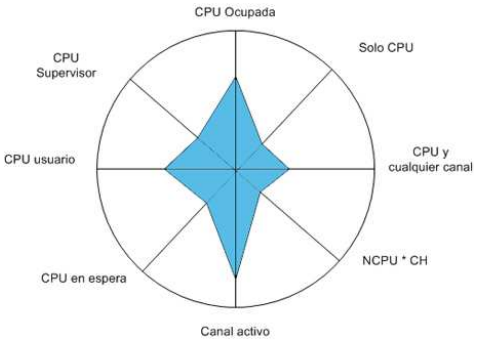
Técnicas de evaluación del rendimiento
Monitorización (9)

Presentación de resultados.- Gráficos de Kiviat, versión Kent

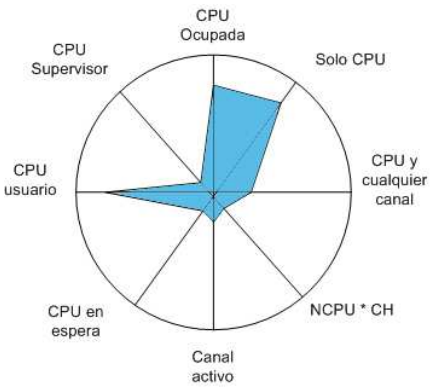
Funcionamiento ideal



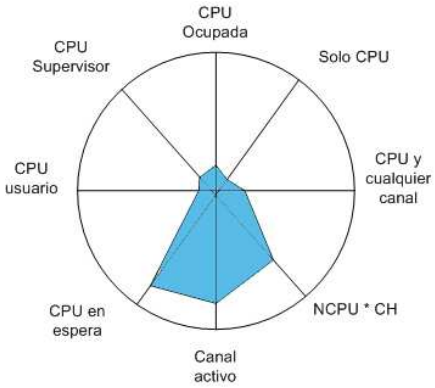
Funcionamiento "normal"



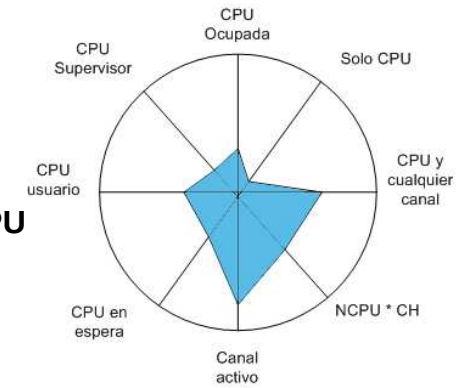
Mucha demanda de CPU, poca uso de canales E/S

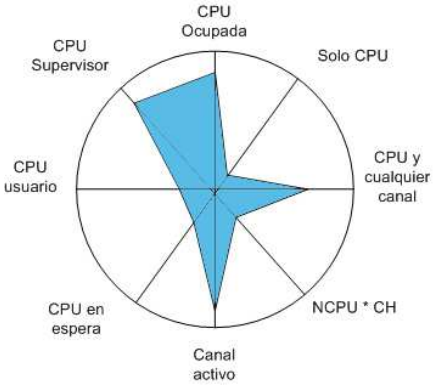


Mucho uso de E/S poco uso de CPU



Mucha E/S que, además, requiere, elevado uso de CPU



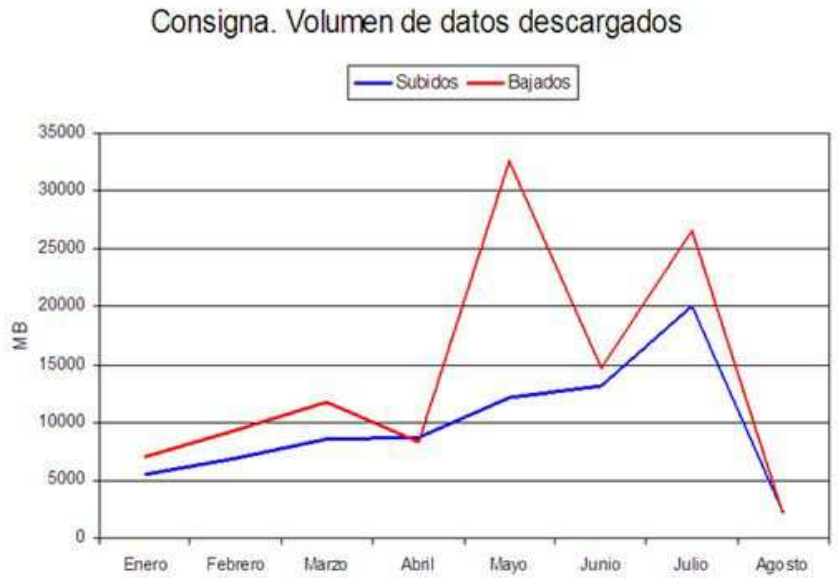


Sistema saturado por la paginación

Técnicas de evaluación del rendimiento

Monitorización (10)

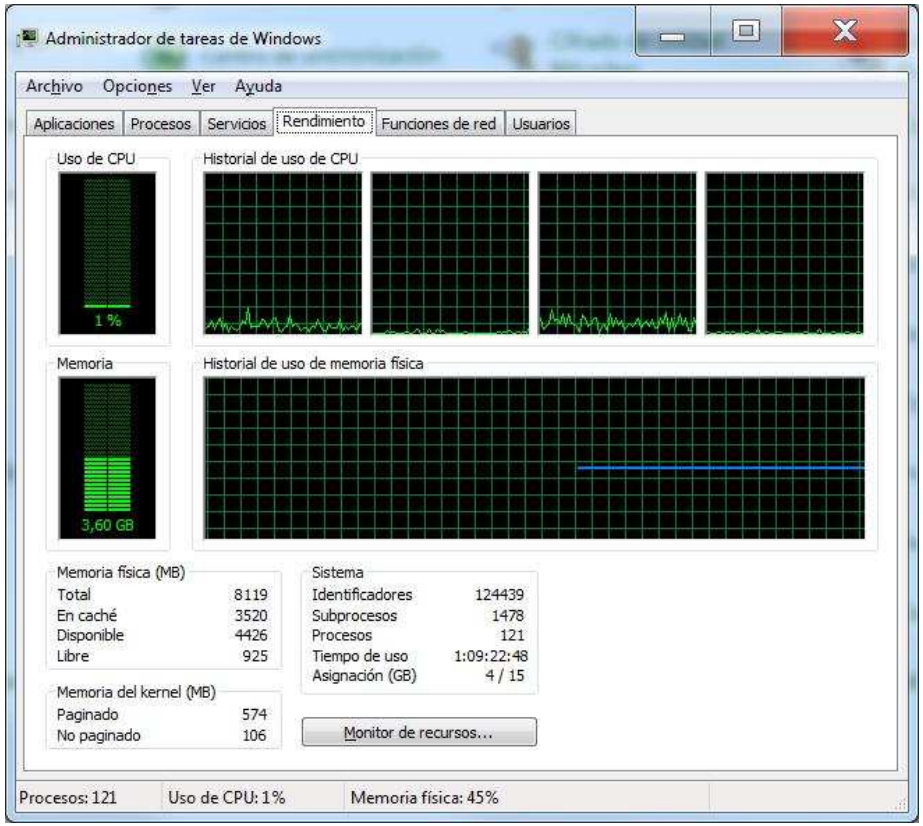
Presentación de resultados.- Función poligonal



Técnicas de evaluación del rendimiento

Monitorización (11)

Presentación de resultados.- Función poligonal

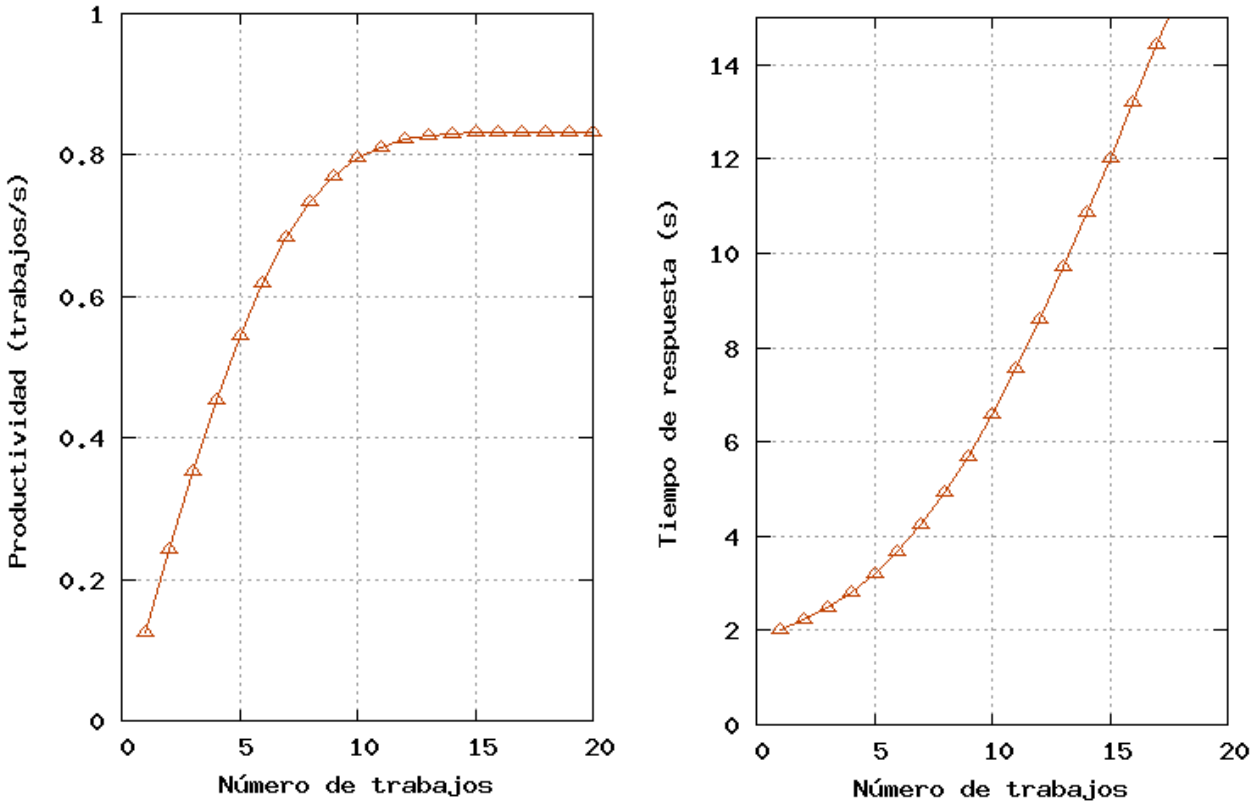


Procesador con 4 núcleos

Técnicas de evaluación del rendimiento

Monitorización (12)

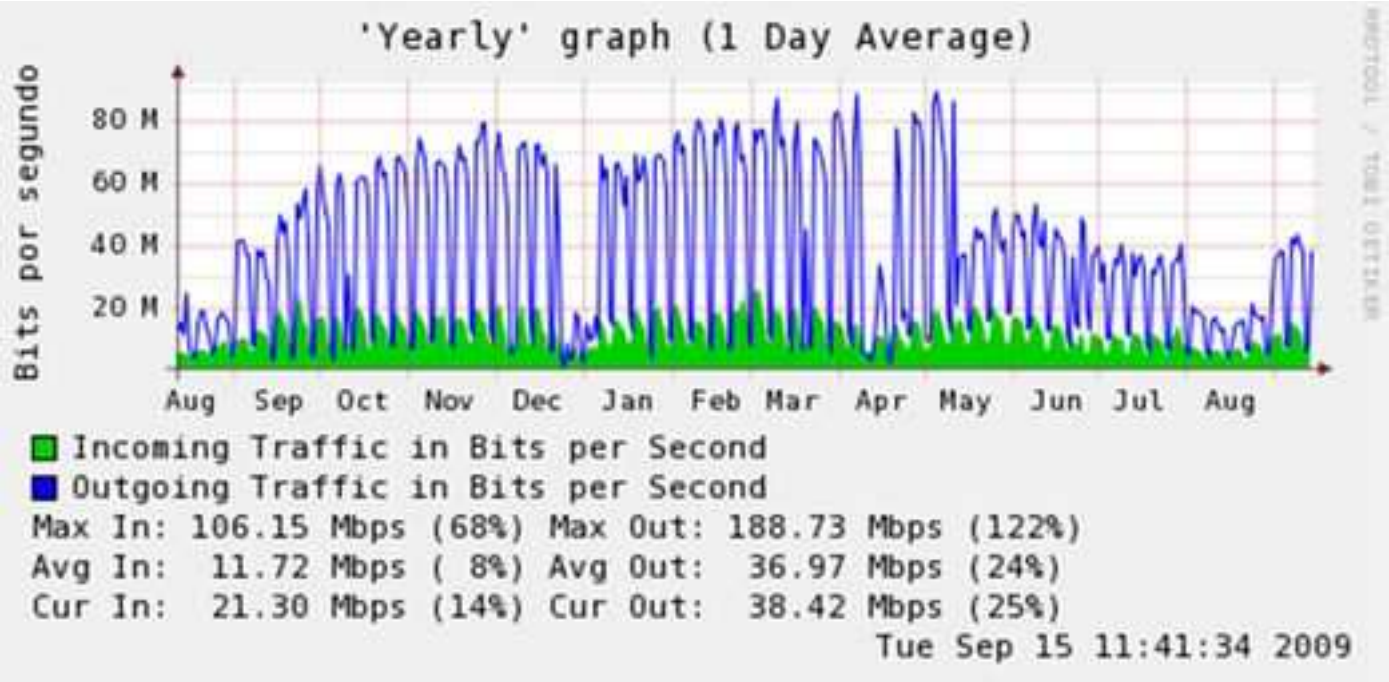
Presentación de resultados.- Función



Técnicas de evaluación del rendimiento

Monitorización (13)

Presentación de resultados.- Función



Técnicas de evaluación del rendimiento
Métodos analíticos (1)

Generalmente construyen un modelo matemático del sistema basado en distribuciones de probabilidad (teoría de colas, cadenas de Markov, …).

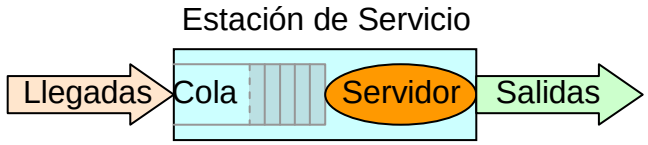
Mediante teoría de colas se intenta determinar el tiempo que los trabajos pasan en las colas de los recursos esperando ser servidos: los recursos del sistema son los servidores y los clientes son los trabajos usuarios de esos recursos.

Al tomar en consideración los diversos recursos del sistema se establece una **red de colas**: cuando un cliente (trabajo) termina de ser servido por un servidor (recurso) pasa al siguiente servidor (recurso) que necesita, si éste está libre lo ocupará y en otro caso se añadirá a su cola de espera. En el caso más simple se podría caracterizar un servidor (recurso) con dos parámetros: tasa de llegadas de clientes y tiempo que cada cliente necesita para servirse.

El análisis operacional intenta deducir los índices de prestaciones de nuestro estudio a partir de parámetros cuantificables en el sistema y de las relaciones existentes entre ellos. Estas relaciones se denominan **leyes operacionales** y son verificables mediante mediciones.

Técnicas de evaluación del rendimiento
Métodos analíticos (2)

Conceptos.-



- **Estación de servicio:** cola + servidor
- **Intensidad de carga:** tasa de llegada de clientes.
- **Demanda de servicio:** tiempo medio de recurso que necesita cada petición.
- **Utilización:** proporción de tiempo que el servidor está ocupado.
- **Tiempo de residencia:** tiempo que un cliente pasa en el servidor (cola + servicio).
- **Longitud de cola:** número medio de clientes en la estación de servicio.
- **Productividad:** tasa de salida de clientes.

Técnicas de evaluación del rendimiento
Métodos analíticos (3)

Variables operacionales básicas.- Son las que se pueden medir directamente sobre el sistema durante un tiempo de observación finito:

- T (*time*), Intervalo de observación.
- A (*arrivals*), Peticiones durante T.
- C (*completions*), Peticiones servidas durante T.
- B (*busy*), Tiempo en el que el servidor está ocupado durante T.

Variables operacionales deducidas.-

- λ, Tasa de llegada. λ= A / T
- X, Productividad (tasa de salida). X= C / T
- U, Utilización. U= B / T
- S, Tiempo medio de servicio. S= B / C

Técnicas de evaluación del rendimiento
Métodos analíticos (4)

Supuestos.-

- Todos los índices de los que se parte son valores medios.
- Una petición está únicamente en un servidor.
- Tiempo de servicio homogéneo: El tiempo que una petición está en una estación no depende del tamaño de las colas que haya en otras estaciones.
- Llegadas homogéneas: El número de peticiones que llegan a una estación es independiente del tamaño de las colas que haya en otras estaciones.

Las redes que cumplen estos supuestos se llaman **redes de colas separadas u homogéneas**

- **Supuesto del equilibrio de flujo:** Para que un sistema funcione el número de llegadas (peticiones que demandan servicio) debe ser igual al número de salidas (peticiones servidas), ésto es, A=C para periodos de observación suficientemente grandes. En tal caso, las tasas de llegada y salida también serán iguales: $\lambda = X$.

Técnicas de evaluación del rendimiento
Métodos analíticos (5)

Variables operacionales básicas en redes.- Para cada estación i:

- A_i
- B_i
- C_{ik}, trabajos que piden servicio en la estación k inmediatamente después de ser servidos en la i.
- A_{0k}, trabajos cuya primera petición es la estación k.
- C_{i0} trabajos cuya última petición es la estación i.
- $C_i = \sum_k C_{ik}$.

0 significa el exterior del modelo, esto es, en los sistemas abiertos será la fuente de clientes y donde van a parar los clientes que han terminado trabajos.

Técnicas de evaluación del rendimiento
Métodos analíticos (6)

Variables operacionales deducidas en redes.- Para cada estación i:

- $U_i = B_i / T$
- $S_i = B_i / C_i$
- $X_i = C_i / T$
- $q_{ik} = C_{ik} / C_i$, probabilidad de que un trabajo pida el servidor k después de ser servido en el i (representa una frecuencia de encaminamiento)

Técnicas de evaluación del rendimiento

Métodos analíticos (7)

Leyes operacionales (1)**.-**

Ley de la utilización

De $B_i / T = (B_i*C_i) / (T*C_i) = (C_i / T)*(B_i / C_i)$ se deduce que

$$\mathbf{U_i = X_i*S_i}$$

y también, por el equilibrio de flujo $A_i=C_i$, por lo que $\lambda_i=X_i$, de donde se deduce que

$$\mathbf{U_i = \lambda_i*S_i}$$

La utilización es igual al flujo de clientes por el servicio medio que pide cada uno.

Técnicas de evaluación del rendimiento

Métodos analíticos (8)

Leyes operacionales (2)**.-**

Ley de Little

Sean:

- W el tiempo acumulado en el sistema por todas las peticiones. Se obtiene sumando para todas las peticiones el tiempo que cada una de ellas está en el sistema.
- N = W / T el número medio de peticiones presentes en el sistema.
- R = W / C el tiempo medio de residencia en el sistema por petición.

entonces, como N = W / T = (W*C) / (T*C) = (C / T)*(W /C) = X*R, se tiene que

$$N = X*R$$

El número medio de peticiones que hay en el sistema es igual a la productividad por el tiempo medio de residencia de cada petición en ese sistema.

Técnicas de evaluación del rendimiento

Métodos analíticos (9)

Leyes operacionales (3)**.-**

Ley del Flujo Forzado

Razón de visita de un recurso: $V_i = C_i / C$, es la relación existente entre el número de peticiones servidas por el recurso i y las servidas por el sistema durante el intervalo de observación T. Se puede ver como el número de peticiones servidas por un recurso que interacciona con el sistema.

Entonces, como

$$X_i = C_i / T = (C_i * C) / (T * C) = (C / T) * (C_i / C) = X * V_i$$

se tiene

$$\mathbf{X_i = X * V_i}$$

que es la **relación entre la productividad total del sistema y la de cada recurso**.

Técnicas de evaluación del rendimiento

Métodos analíticos (10)

Leyes operacionales (4)**.- Ejemplo 1** (1)**:** Supongamos un sistema con

- 10 terminales (N_T=10 pet)
- 7.5 terminales trabajando por término medio al mismo tiempo (N=7.5 pet)
- 0.5 interacciones/seg de productividad (X=0.5 pet/seg)
- un disco 1 que
  - sirve 40 peticiones/seg (X_1=40 pet/seg)
  - con un tiempo medio de servicio por petición de 0.0225seg (S_1=0.0225 seg/pet)
  - y un número medio de peticiones presentes en el disco 1 de 4 (N_1=4 pet)

entonces:

a) Utilización del disco 1 (Ley de la utilización): número medio de peticiones recibiendo servicio en cada instante en el disco 1:

$$\mathbf{U_1} = X_1 * S_1 = 40 \text{ pet/seg} * 0.0225 \text{ seg} = \mathbf{0.9 \text{ pet}}$$

Técnicas de evaluación del rendimiento
Métodos analíticos (11)

Leyes operacionales (5)**.- Ejemplo 1** (2)**:**

b) Tiempo medio de residencia de una petición en el disco 1 (aplicando la Ley de Little):

$$\mathbf{R_1} = N_1/X_1 = 4 \text{ pet } \mathbf{/} \text{ 40 pet/seg} = \mathbf{0.1 \text{ seg}}$$

c) Tiempo medio de espera en cola de una petición en el disco 1:

$$0.1 \text{ seg} = \text{Tiempo medio en cola}_1 + 0.0225 \text{ seg}, \quad \text{de donde}$$

$$\mathbf{\text{Tiempo medio en cola}_1} = 0.1 \text{ seg} - 0.0225 \text{ seg} = \mathbf{0.0775 \text{ seg}}$$

d) Número medio de peticiones en la cola de espera del disco 1:

$$\text{Nº medio peticiones en cola} + \text{Nº medio peticiones en servicio} = 4 \text{ pet}$$

$$\text{Nº medio peticiones en cola} = 4 \text{ pet} - \text{Nº medio peticiones en servicio}$$

$$\mathbf{\text{Nº medio peticiones en cola}} = 4 \text{ pet} - 0.9 \text{ pet} = \mathbf{3.1 \text{ pet}}$$

Técnicas de evaluación del rendimiento
Métodos analíticos (12)

Leyes operacionales (6)**.- Ejemplo 1** (3)**:**

e) Tiempo medio de respuesta percibido por el usuario:

$$\mathbf{R} = N / X = 7.5 \text{ pet } \mathbf{/} \text{ 0.5 pet/seg} = \mathbf{15 \text{ seg}}$$

(en este caso se calcula a nivel de sistema, no de disco, y se considera N = 7.5 porque es la media de usuarios trabajando en el sistema)

f) Tiempo de reflexión de los usuarios (Z) (se aplica la Ley de Little a todo el sistema):

$$R_T = (N_T / X) - Z, \quad \text{de donde}$$

$$\mathbf{Z} = (N_T / X) - R_T = (10 \text{ pet } \mathbf{/} \text{ 0.5 pet/seg}) - 15 = \mathbf{5 \text{ seg}}$$

(aquí se toma N_T = 10, número total de usuarios, porque se considera que los usuarios que no están trabajando, 7.5 de media, están “reflexionando”)

Técnicas de evaluación del rendimiento

Métodos analíticos (13)

Leyes operacionales (7)**.- Ejemplo 2:**

En un sistema batch cada trabajo requiere una media de 6 accesos a un determinado disco específico. Ese disco atiende 12 pet/seg. Se pide: a) ¿Cuál es la productividad del sistema?, b) Si otro disco sirve 18 pet/seg, ¿cuántos accesos requiere un trabajo a ese disco?

Según la Ley del flujo forzado $X_k = X * V_k$, por lo que

$$X = X_k / V_k$$

y

$$V_k = X_k / X,$$

entonces

a) **$X = X_k / V_k = 12 / 6 =$ 2 pet/seg**

b) **$V_j = X_j / X = 18 / 2 =$ 9 pet**

Técnicas de evaluación del rendimiento
Métodos analíticos (14)

Cuellos de botella (1)

Los cuellos de botella se producen cuando uno o varios recursos, aún siendo utilizados al 100%, no son capaces de satisfacer a tiempo las peticiones que les llegan. En consecuencia, las peticiones pierden tiempo en espera y las prestaciones del sistema no son las esperadas. En esta situación, cualquier mejora que se introduzca en el sistema y que no afecte al recurso responsable del cuello de botella no inducirá una mejora en las prestaciones.

Se debe tener presente que:

- Al tratar un cuello de botella pueden aparecer otros. Hay que ir tratándolos hasta que el sistema quede equilibrado.
- Los cuellos de botella no dependen únicamente de la configuración hardware. También pueden aparecer en función de las características de la carga.
- Hay cuellos de botella temporales que aparecen un corto periodo de tiempo durante una sesión de medida. Son difíciles de eliminar analizando a posteriori los datos registrados anteriormente durante la sesión de medida. Lo mejor es usar métodos de monitorización y toma de decisiones en tiempo real.

Técnicas de evaluación del rendimiento

Métodos analíticos (15)

Cuellos de botella (2)

Terapias.-

Cuando se detecta un cuello de botella se deben evaluar las consecuencias y costes de cada una de las posibles terapias que puedan aplicarse. En general hay dos tipos de terapias:

- Terapias de reposición *(upgrading)*: Son modificaciones en el hardware consistentes en reemplazar o ampliar componentes.
- Terapias de sintonización *(tuning)*: Son modificaciones que tienen efecto en la organización del sistema.

Antes de llevar a efecto las reposiciones, que en general suelen tener mayor coste que las sintonizaciones y además son más difíciles de deshacer (los componentes ya se han adquirido), se debería tener alguna garantía sobre el resultado que se obtendrá después de aplicarlas. Esto lleva a utilizar modelos de simulación y/o analíticos. Los modelos analíticos permiten detectar y localizar los cuellos de botella, por un lado, y, por otro, predecir y cuantificar las mejoras que se obtendrían con las diferentes soluciones.

Técnicas de evaluación del rendimiento

Métodos analíticos (16)

Cuellos de botella (3)

Límites asintóticos (1)

Son métodos analíticos que permiten ver los límites de productividad y tiempo de respuesta del sistema en función de la intensidad de la carga. Se basan en la teoría de redes de colas. Son recomendables para evaluar diferentes configuraciones o posibles reposiciones.

Aquí se expone un caso simple en el que:

- se tiene una sola clase de clientes,
- se supone que la demanda de servicio de un cliente a una estación no depende de las demandas a esa estación u otras,
- se calculan los límites superior e inferior de las prestaciones que obtendrá el sistema.

Los valores de estos límites serán distintos para cada terapia, por lo que se puede estudiar el impacto de cada una de ellas y se puede decidir qué mejoras se van a aplicar.

Técnicas de evaluación del rendimiento

Métodos analíticos (17)

Cuellos de botella (4)

Límites asintóticos (2)

Parámetros a considerar:

- **K**, número de estaciones de servicio.
- **N**, número de clientes (para sistemas interactivos)
- $\mathbf{D_i}$= V_i* S_i, demanda de servicio de la estación i.
- $\mathbf{D_{max}}$, la mayor de las demandas de servicio de entre todas las diferentes estaciones.
- **D**, suma de las demandas de servicio en las estaciones (suma de las demandas de un cliente a todas las estaciones).
- **Z**, tiempo medio de pensar (para sistemas interactivos).

El dispositivo que tenga la máxima demanda será el candidato a ser el cuello de botella. Será el que antes llegará a una utilización del 100% (saturación).

Los parámetros utilizados para evaluar las prestaciones del sistema serán **la productividad y el tiempo de respuesta**.

Técnicas de evaluación del rendimiento

Métodos analíticos (18)

Cuellos de botella (5)

Límites asintóticos en cargas transaccionales (1)

a) Productividad (1)**.-**

El límite de la productividad indica la tasa máxima de llegada de clientes que puede procesar el sistema de forma satisfactoria. A partir de ese punto el sistema estará saturado.

Para la estación k, por la Ley de la Utilización ($U_k = X_k * S_k$), se tiene

$$U_k = X_k S_k = X_k*(1/V_k)*(V_k*S_k) = (C_k/T)*(1/(C_k/C))*D_k = C_k/T)+(C/C_k)*D_k = (C/T)*D_k = X*D_k$$

por el Equilibrio de Flujo $X=\lambda$, con lo que

$$U_k=\lambda*D_k$$

Por otro lado, y por definición, $U_k = B_k/T$, siendo B_k el tiempo en el que la estación k está ocupada durante el periodo T. De esta definición se sigue que siempre ha de ser

$$U_k \leq 1$$

Técnicas de evaluación del rendimiento

Métodos analíticos (19)

Cuellos de botella (6)

Límites asintóticos en cargas transaccionales (2)

a) Productividad (2)**.-**

Si para alguna estación ocurriera $U_k = 1$ eso significaría que la estación k está ocupada durante todo el tiempo T de estudio $\Rightarrow$ no podremos procesar una carga mayor $\Rightarrow$ **la estación k que tenga $U_k = 1$ estará saturada, su demanda de servicio será D_{max} y será el cuello de botella**. En tal caso la productividad X del sistema estará limitada por la tasa de llegada λ_{sat} que sature esa estación y tendremos

$$U_{max}(\lambda) = \lambda_{sat} * D_{max} = 1$$

de donde

$$\boxed{\lambda_{sat} = 1 / D_{max}}$$

La tasa de llegadas que satura el sistema es $1/D_{max}$ y será **la productividad máxima alcanzable con una determinada carga, recursos y configuración**.

Técnicas de evaluación del rendimiento

Métodos analíticos (20)

Cuellos de botella (7)

Límites asintóticos en cargas transaccionales (3)

b) Tiempo de respuesta.-

- En el caso más favorable sólo hay un trabajo en el sistema y el tiempo de respuesta es la suma D de las demandas de servicio D_k.
- En el caso más desfavorable se sobrepasa la saturación. A partir de ese momento no hay límite en el tiempo de respuesta.

Técnicas de evaluación del rendimiento

Métodos analíticos (21)

Cuellos de botella (8)

Límites asintóticos en cargas interactivas (1)

a) Productividad (1)**.-**

La productividad X del sistema y la utilización U_k de cada recurso serán distintas según sea el número N de clientes. Las notaremos X(N) y U_k(N) para poner de manifiesto esa dependencia. En general se tendrá que

$$U_k(N) = X(N)*D_k \leq 1$$

de donde

$$X(N) \leq 1 / D_k$$

y considerando todos los recursos

$$X(N) \leq 1 / D_{max}$$

Si ocurre que

$$X(N) = 1 / D_{max}$$

eso significa que un recurso se ha saturado y la productividad no podrá crecer más. A partir de esa carga empezarán a producirse colas (al menos en el recurso saturado).

Técnicas de evaluación del rendimiento

Métodos analíticos (22)

Cuellos de botella (9)

Límites asintóticos en cargas interactivas (2)

a) Productividad (2)**.-**

Para un solo cliente la productividad del sistema sería

$$X(1) = 1 / (D+Z)$$

Si hubiera N clientes:

- ***Caso más óptimo***: todas las peticiones de los N clientes a los recursos se solapan completamente y ninguna hace cola. En este caso la productividad será N*(1 / (D+Z)) = N / (D+Z) Como este es el caso más favorable, en general se tendrá que

$$X(N) \leq N / (D+Z)$$

Técnicas de evaluación del rendimiento

Métodos analíticos (23)

Cuellos de botella (10)

Límites asintóticos en cargas interactivas (3)

a) Productividad (3)**.-**

Si hubiera N clientes:

- ***Caso peor***: todos los clientes piden el mismo recurso a la vez. En este caso la productividad será N / (N*D+Z) y, como es el caso más desfavorable, en general se tendrá que

$$(N / (N*D+Z)) \leq X(N)$$

Así pues, para N clientes se tiene que

$$(N / (N*D+Z)) \leq X(N) \leq \min \{1 / D_{max}, N / (D+Z)\} \qquad [a]$$

Técnicas de evaluación del rendimiento

Métodos analíticos (24)

Cuellos de botella (11)

Límites asintóticos en cargas interactivas (4)

a) Productividad (4)**.-**

En el caso de saturación la productividad es

$$X(N) = 1 / D_{max}$$

a lo que corresponde un número N_{sat} de clientes tal que

$$1 / D_{max} = N_{sat} / (D+Z)$$

de donde se tiene que el **número de clientes que satura el sistema** es

$$\mathbf{N_{sat} = (D+Z) / D_{max}}$$

Técnicas de evaluación del rendimiento

Métodos analíticos (25)

Cuellos de botella (12)

Límites asintóticos en cargas interactivas (5)

b) Tiempo de respuesta (1)**.-**

Por la Ley de Little $N = X*R$, con R el tiempo medio de residencia en el sistema de un cliente, de donde $X = N / R$. En sistemas interactivos X y R dependen del número N de clientes, X(N) y R(N), y hay que considerar el tiempo Z de "pensar" de los clientes. Así se tiene

$$X(N) = N / (R(N)+Z)$$

que aplicado a [a] resulta

$$(N / (N*D+Z)) \leq (N / (R(N)+Z)) \leq \min \{1 / D_{max}, N / (D+Z)\}$$

Invirtiendo los términos de esta desigualdad se tiene

$$\max \{D_{max}, ((D+Z) / N)\} \leq ((R(N)+Z) / N) \leq ((N*D+Z) / N) \qquad [b]$$

Técnicas de evaluación del rendimiento

Métodos analíticos (26)

Cuellos de botella (13)

Límites asintóticos en cargas interactivas (6)

b) Tiempo de respuesta (2)**.-**

Multiplicando [b] por N y restando Z se llega a

$$\max \{(N*D_{max}-Z), D\} \leq R(N) \leq N*D$$

Entonces, si:

- $Z=0 \Rightarrow$ tenemos los límites para cargas batch (los clientes no "piensan").
- $\max \{(N*D_{max}-Z), D\} = (N*D_{max}-Z) \Rightarrow$ el recurso cuello de botella es el único que limita la velocidad.
- $\max \{(N*D_{max}-Z), D\} = D \Rightarrow$ no hay colas en los recursos.
- $R(N) \leq N*D \Rightarrow$ se sirven con anterioridad todos los demás clientes.