DISCA
DEPARTAMENT D'INFORMÀTICA
DE SISTEMES I COMPUTADORS

Arquitectura y Prestaciones de la WEB

Tema 2: Medidas y evaluación de prestaciones

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Medidas y evaluación de prestaciones

- Parte 1: El fondo de la cuestión
  - Introducción
  - Pasos para el estudio de rendimiento de un sistema
- Parte 2: Métricas de prestaciones en los Sistemas Web
  - Medidas de prestaciones
  - Análisis de qué debe ser medido
- Parte 3: Características de la carga
  - Características del Servidor
  - Características del Cliente
  - Características de la Red
  - Características del Protocolo
- Parte 4: Entorno de Test
  - Configuración hardware
  - Configuración software

Parte 1:
El fondo de la cuestión

- Introducción
- Pasos para el estudio de rendimiento de un sistema

Introducción

- Por qué es importante el análisis de prestaciones
  - Permite a los administradores verificar, ajustar el tamaño, gestionar y sintonizar los sistemas.
  - Permite a los investigadores mejorar los sistemas
- Qué debe tenerse en cuenta cuando se analizan las prestaciones
  - Cómo se pueden crear condiciones realistas
  - Cómo se pueden validar los resultados
  - Cómo se pueden interpretar los resultados

Introducción

- Cúando se necesitará evaluar objetivamente las prestaciones de un sistema:
  - Diseño del sistema.
  - Adquisición del sistema.
  - Explotación y ampliación del un sistema
- Los objetivos de una evaluación suelen ser alguno de los siguientes:
  - Comparar alternativas.
  - Determinar el impacto de una nueva funcionalidad.
  - Sintonizar el sistema, es decir, hacer que funcione mejor según algún punto de vista.
  - Identificar los fallos del sistema que hacen que vaya más lento
  - Poner unas expectativas sobre el uso del sistema, por ejemplo, cuántas conexiones es capaz de soportar un sitio web.

Pasos

- Cualquier estudio de rendimiento de un sistema deberá seguir *explícitamente* estos pasos.
  1. *Definir el sistema y especificar los objetivos*.
  2. *Hacer una lista de los servicios que ofrece el sistema y sus posibles resultados*.
  3. *Seleccionar las métricas*.
  4. *Listar los parámetros que pueden afectar a las prestaciones*.
  5. *Factores a estudiar.*
  6. *Seleccionar las técnicas de evaluación*: modelización, simulación y/o medición.
  7. *Seleccionar la carga de trabajo*.
  8. *Diseñar los experimentos*.
  9. *Analizar e interpretar los datos*.
  10. *Presentar los resultados*.

Paso 1

- Especificar los objetivos y definir el sistema:
  - Se debe de definir claramente cuál es el sistema, para medir exclusivamente eso y eliminar en lo posible de la medición la influencia de todos los demás factores.
  - Una medición de prestaciones no tiene sentido sin objetivos.

Paso 2

- Hacer una lista de los servicios que ofrece el sistema y sus posibles resultados
  - Un sistema puede dar un resultado válido, inválido o simplemente no dar ningún resultado, en cualquier caso, habrá que medir la tasa de sucesos de uno u otro tipo.
  - Se puede empezar listando los servicios dados por el sistema;
  - Por ejemplo, para cada petición:
    - La petición se ha realizado correctamente.
    - La petición se ha realizado incorrectamente.
    - La petición no se ha podido realizar.

Paso 3

- Seleccionar las métricas
  - Esto es, los criterios para comparar prestaciones.
  - Para los servicios anteriores
    - La petición se ha realizado correctamente.
      - la velocidad.
    - La petición se ha realizado incorrectamente.
      - la fiabilidad
    - La petición no se ha podido realizar.
      - la disponibilidad

Paso 3

Seleccionar las métricas

- Cada servicio que ofrece un sistema debe de tener una serie de métricas de velocidad, fiabilidad y disponibilidad.
- Por ejemplo,
  - la fiabilidad se puede medir en tiempo medio entre fallos (MTBF, mean time between failures)
  - disponibilidad en el número de horas al año que no está disponible debido a un fallo. Generalmente, estos factores son difíciles de evaluar. A veces se habla de disponibilidad cinco nueves, para indicar que el sistema está disponible el 99.999%.

Seleccionar las métricas

- La velocidad (llevado a cabo correctamente) se mide por:
  - por el tiempo que se ha tardado en realizar la petición, "**responsividad**"
  - la tasa a la cual el servicio ha sido realizado, **productividad**
  - y los recursos consumidos mientras se lleva a cabo el servicio, **utilización**
- Por *ejemplo*, para una pasarela de red (*gateway*),
  - la "responsividad" se mide por su tiempo de respuesta, es decir, el tiempo entre la llegada y la salida de un paquete;
  - su productividad por el número de paquetes que envía por unidad de tiempo,
  - y su utilización el porcentaje de tiempo que los recursos se usan en una unidad de tiempo determinada.

Paso 3

Seleccionar las métricas

- Los tres criterios que se suelen seguir para elegir un subconjunto de todas las métricas suelen ser:
  - variabilidad baja (para que no haya que repetir las mediciones muchas veces),
  - que no haya redundancia (que no haya métricas que dependan unas de otras),
  - y complitud (que definan de forma completa las prestaciones de un sistema).

Paso 3

Seleccionar las métricas

- Las métricas de prestaciones se suelen clasificar de la forma siguiente:
  - *Mayor es mejor*, HB, higher is better; es decir que es mejor cuanto más alta, como la velocidad, o el throughput de un sistema.
  - *Menor es mejor*, LB, Lower is better; es decir, que los valores inferiores son los mejores, como el tiempo de respuesta o el número de fallos de página.
  - *Nominal es mejor*, NB, Nominal is best, no son buenos lo valores altos ni los bajos; por ejemplo, la utilización es un valor de este tipo. Utilización baja significa infrautilización, y utilización alta hace que los tiempos de respuesta sean altos.

Paso 4

- Listar los parámetros que pueden afectar a las prestaciones
  - Estos parámetros se dividen entre las características del sistema, y la carga de trabajo a la cual está sometido; las primeras no varían para todos los sistemas que tengan las mismas características; pero la segunda varía entre diversas instalaciones.

Paso 5

- Factores a estudiar
  - De los parámetros anteriores, algunos se variarán durante el estudio, los denominados factores. Los diferentes valores que tomarán durante el estudio se denominan niveles.

Paso 6

- Seleccionar las técnicas de evaluación
  - Medición
    - Consiste en tomar medidas directamente sobre el sistema en el que uno está interesado.
    - Hay que utilizar la carga adecuada y que la toma de medidas no influya sobre el sistema a medir.
  - Modelado Analítico.
    - usando fórmulas y ecuaciones, tratar de hallar a partir de los valores conocidos o estimados de ciertos parámetros, los valores de los que nos van a interesar.
  - Simulación.
    - usando algún lenguaje de simulación, simular el sistema original.

Paso 6

Seleccionar las técnicas de evaluación

- Qué técnica usar.
  - La mayoría de las veces se recurre a la medición: las herramientas existen ya, y sólo hay que aplicarlas a nuestro sistema; sin embargo, si el sistema no existe, la única forma de medir sus prestaciones es mediante simulación y/o modelización.
  - También habría que tener en cuenta el coste (normalmente la medición es bastante cara),
  - En cuanto al tiempo que se tarda en obtener resultados, lo más rápido es usar un modelo analítico: simplemente se aplican ecuaciones; las mediciones tardan un poco más (sobre todo, teniendo en cuenta la variabilidad de las cargas de trabajo durante el tiempo); por último, la simulación es lo más lento, pues hay que diseñar y escribir un programa y evaluar los resultados.

Paso 6

Seleccionar las técnicas de evaluación

- Qué técnica usar.
  - En cuanto a la exactitud, por supuesto, realizar mediciones sobre el propio sistema da el resultado más exacto (siempre que se mida lo correcto, y se extrapolen correctamente), seguido por la simulación, ya que en ella se ponen casi todos los elementos del sistema real, y por último, el modelo analítico, porque requiere gran cantidad de suposiciones.
  - y, por supuesto, lo vendibles que son los resultados (en este caso, lo mejor son mediciones).

Paso 7

- Seleccionar la carga de trabajo
  - es decir, la carga a la que se va a someter el sistema para medirlo. Siempre se hará en función de los objetivos establecidos
  - en particular si el objetivo es mejorar las prestaciones para una carga determinada.

Paso 8

- Diseñar los experimentos
  - dividiéndolos en niveles o valores que tomarán los factores.
  - Inicialmente, se suele diseñar un experimento con muchos factores, pero pocos niveles, para, una vez vistos cuáles son los factores que influyen más en el experimento, concentrarse en esos.

Paso 9

- Analizar e interpretar los datos
  - no basta con medir, sino que hay que sintetizar los datos de las medidas, y extraer conclusiones de ellos.

Paso 10

- Presentar los resultados
  - Es muy importante, tanto si se presenta en un congreso como si es a un gerente que debe de tomar una decisión sobre qué comprar.

Cuestiones clave

- ¿Qué métricas de prestaciones son importantes?
  - Correcto funcionamiento
  - Tamaño, sintonización y planeamiento de capacidad
- ¿Cómo afecta a los resultados el entorno de test?
- ¿Qué técnicas de medida y análisis deben ser empleadas?
- ¿Cómo se puede validar el análisis?
- ¿Puede utilizarse la simulación?

Parte 2:
Métricas de prestaciones en los Sistemas Web

- Medidas de prestaciones
- Análisis de qué debe ser medido

Decidiendo qué medir

- Exactitud funcional
  - ¿Responderá un servidor web con el conjunto correcto de objetos web cuando se le hayan pedido por parte de un cliente?
- Prestaciones
  - El tamaño del servidor ¿es el adecuado para la población de clientes estimada?

Medidas clave de prestaciones

- Latencia:
  - Tiempo transcurrido desde que se hace una petición hasta que se empieza a ver el resultado.
- Productividad
  - Número de ítems procesados por unidad de tiempo
  - Si se mide en bits/seg. se denomina ancho de banda
- Medir la latencia y la productividad requiere mantener registros (logs) en el servidor y el cliente.

Medidas clave de prestaciones

- Diferencias entre productividad y latencia:
  - Ejemplo (The Mythical Man-Month by Frederic P. Brooks):
    - Una vaca puede tener una productividad de 1 becerro cada 9 meses
      - Latencia 9 meses
    - Si se toman 9 vacas se puede tener 9 becerros cada 9 meses
      - Productividad 1 becerro cada mes
      - Latencia 9 meses
  - Las trasferencias de ficheros grandes pueden tener alta productividad, pero baja latencia.

Medidas clave de prestaciones

- Midiendo la latencia
  - La mayoría de la latencia en el acceso a un servidor web se debe a la naturaleza de almacenamiento-retransmisión de los routers influyendo el número de routers en el camino.

Medidas clave de prestaciones

- Utilización
  - Fracción de la capacidad de un componente que se está utilizando actualmente.
  - Una utilización alta de un componente respecto del resto puede indicar un cuello de botella.
  - Las mejores prestaciones se suelen obtener alrededor de una utilización del 70%
  - Herramientas:
    - perfmeter

Medidas clave de prestaciones

- Eficiencia
  - Productividad partido por utilización.
  - Con la misma utilización es mejor el que da mayor productividad.
  - Con la misma productividad es mejor el que tiene menor utilización.

Medidas con detalle

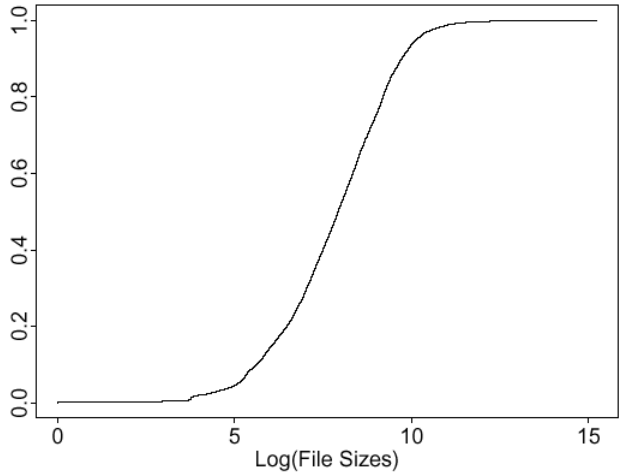
- Para una detallada evaluación de prestaciones en base a ajustar la capacidad y sintonizar los servidores se requiere instrumentación adicional.
- El análisis incluye utilización de la CPU, Disco, Memoria, Red, e incluso perfilado (profiling) del núcleo.
  - Webmonitor para Linux
  - Perfmon para NT
  - DCPI para Alfa

Herramientas de medida

- Webmonitor para Linux
  - Monitoriza una amplia variedad de estadísticas de CPU, incluyendo información específica de procesos HTTP
  - Sobrecarga de menos del 2% y almacena los datos en formato comprimido
- Perfmon para NT
  - Herramienta estándar en NT
  - Robusta pero difícil de monitorizar uso de memoria
- DCPI para Alfa
  - Permite el profiling del sistema completamente (S.O., aplicaciones, drivers, etc.)
  - Sobrecarga entre el 1% y el 3% y almacena en formato comprimido

Analizando datos

- Estadísticas simples
  - Cuentas, media, mediana, varianza, covarianza, …
- Distribuciones
  - Función de Distribución Acumulativa (CDF)
    - Relaciona un valor con la probabilidad de que la variable tome un valor menor o igual a ese valor



$$F_x(a) = P(x \leq a)$$

Distribuciones (Cont.)

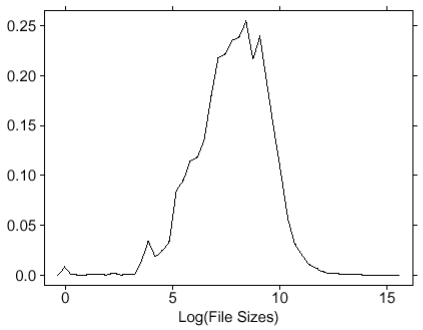
- Función de Densidad de Probabilidad (PDF)
  - Es la derivada de la CDF

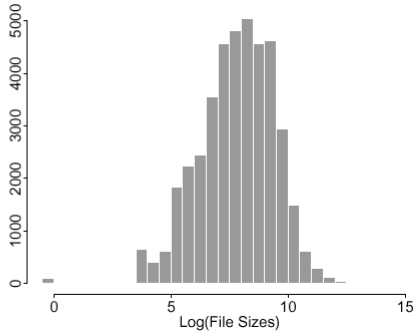
$$f(x) = dF(x)/dx$$

  - Dada una pdf f(x), la probabilidad de que x esté en el intervalo (x1, x2) es

$$P(x_1 < x < x_2) = F(x_2) - F(x_1) = \int_{x_1}^{x_2} f(x)dx$$

  - Otra manera de analizar densidad es mediante histogramas

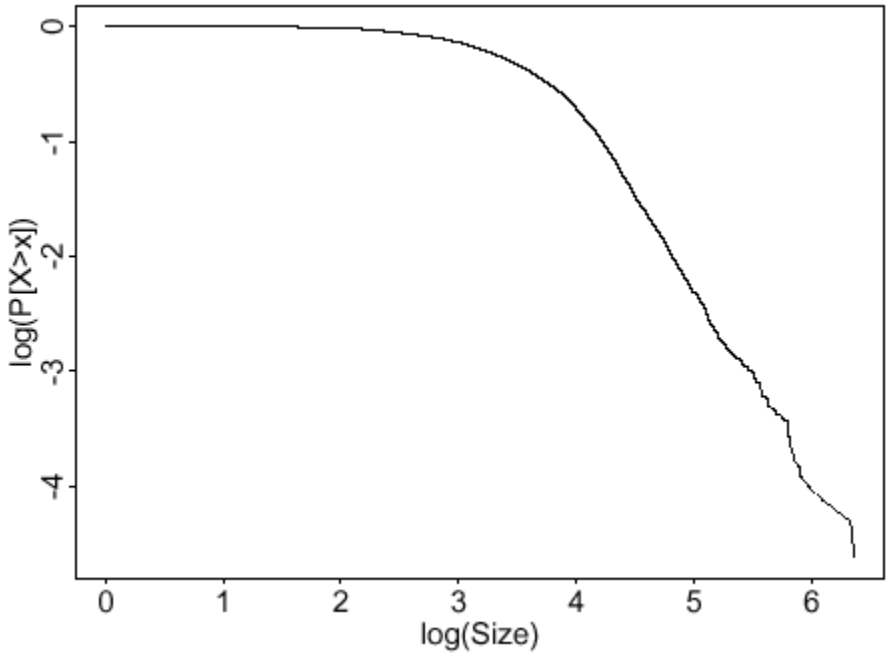




Analizando datos

Distribuciones (Cont.)

- Análisis de la cola
  - Algunas características web tienen una gran variabilidad
  - La distribución complementaria log-log (LLCD) es un medio para ver las características de la cola



Analizando datos

- Modelando:
  - Ajustar distribuciones estándar a las distribuciones empíricas
  - Métodos visuales
    - Gráficas CDF, PDF, LLCD, Quantiles-Quantiles, etc.
    - A veces es muy duro distinguir entre distribuciones estándar
    - La realidad es que la mayoría de modelos se ajustan visulamente.
  - Métodos matemáticos de bondad de ajuste
    - Incluyen Chi Cuadrado y λ^2
    - Mejor que los métodos visuales, dado que existen medias numéricas para comparar los ajustes
    - El tamaño de la muestra es crítico.

Analizando datos

- Modelando (Cont.):
  - Tests de bondad de ajuste
    - Estadísticos de la función de densidad empírica (EDF)
    - Método de Kolmogorov
    - Método de Anderson-Darling
    - Los test se rompen con conjuntos grandes de datos

Parte 3:
Características de la carga

Características del Servidor
Características del Cliente
Características de la Red
Características del Protocolo

Características del Servidor

Popularidad

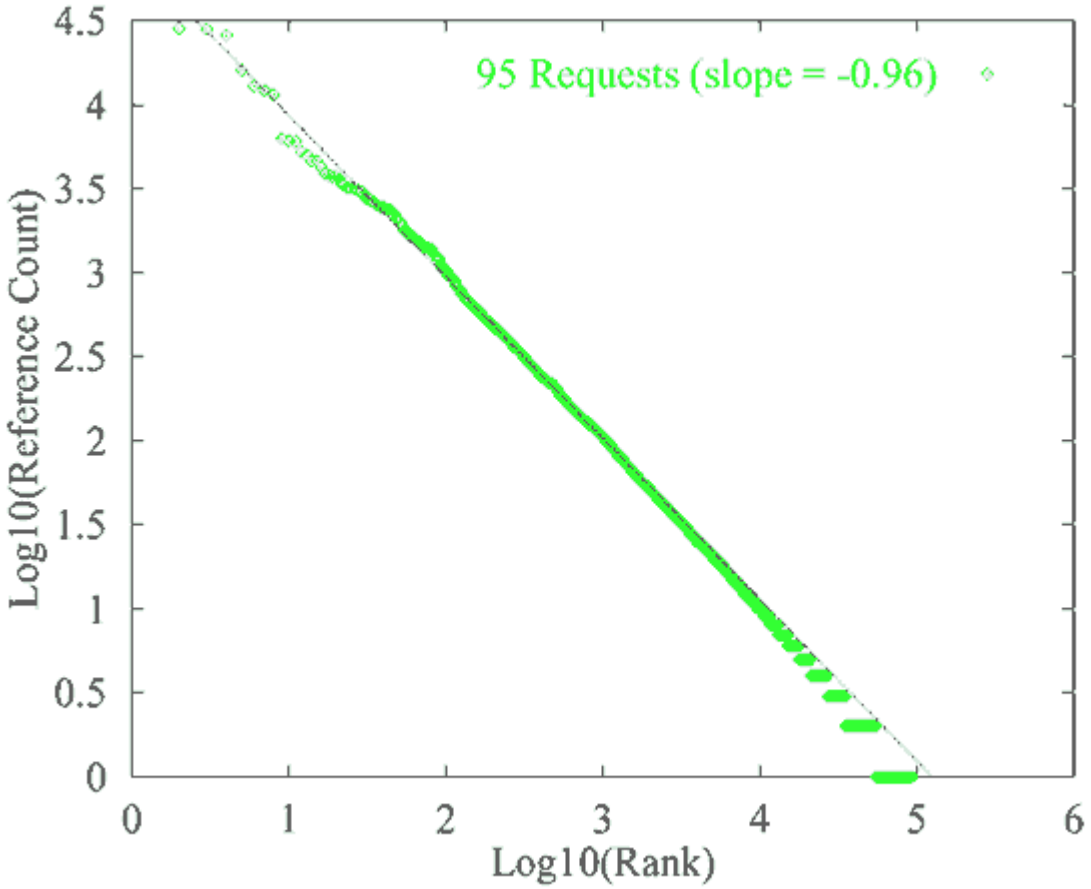
- Definido como el número de veces que un objeto ha sido demandado en un periodo dado.
- Se necesita un modelo detallado para poder especificar de forma realista caches y buffers.
- Se modela típicamente por la Ley de Zipf [*]
  - El número de referencias a un fichero (P) es inversamente proporcional a su ranking (r) (o jerarquía), para alguna constante positiva (k)

$$P = \frac{k}{r^{\beta}}$$

  - Se han hallado valores empíricos $0.6 < \beta < 1$

Características del Servidor

- Polularidad



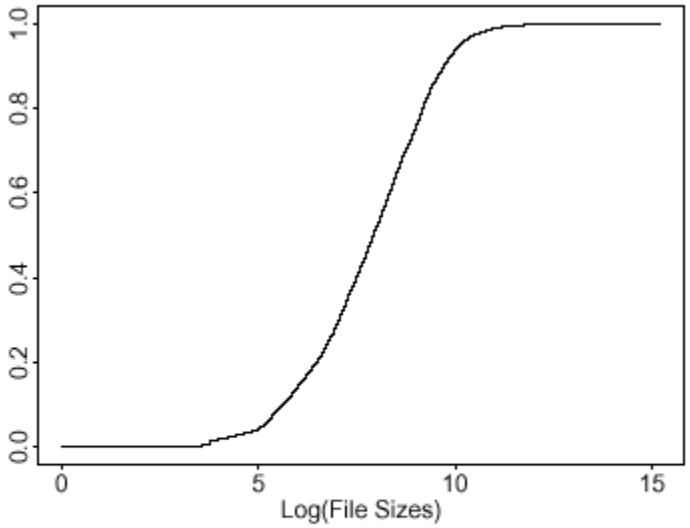
Características del Servidor

- Tamaños de ficheros
  - Tamaños de todos los ficheros únicos que residen en el servidor
  - Se necesita un modelo detallado para poder especificar de forma realista el sistema de ficheros del servidor.
  - Se ha demostrado que exhiben "colas pesadas" (heavy tails).
    - La cola superior de la distribución decae con una ley de potencia

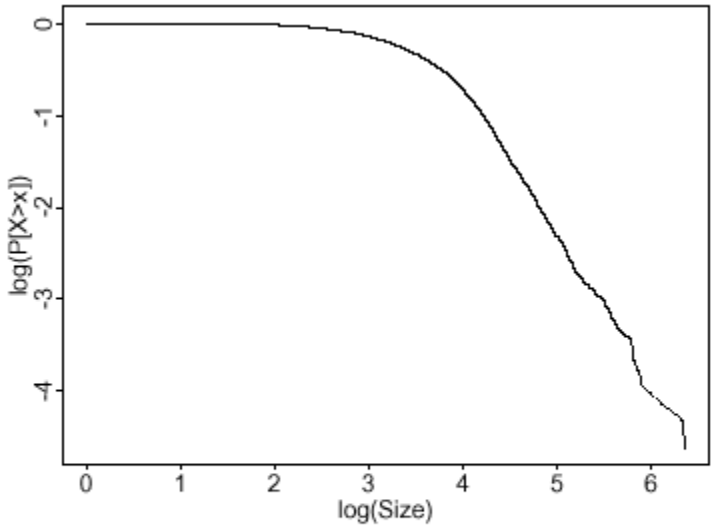
$$P[X < x] \approx x^{-\alpha} \quad 0 < \alpha \leq 2$$

Características del Servidor

- Tamaños de ficheros



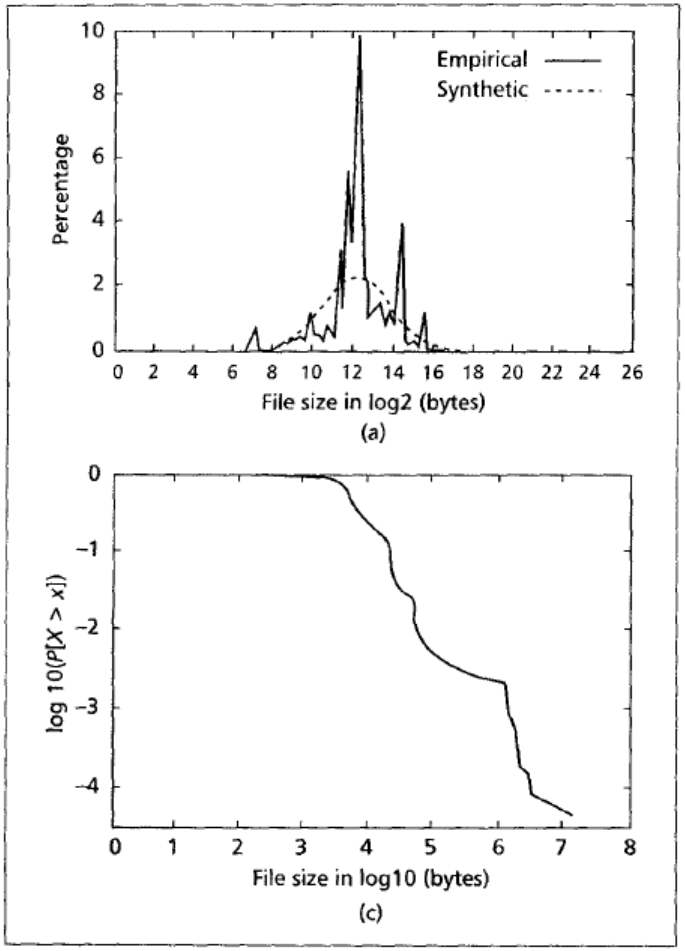
CDF File Sizes

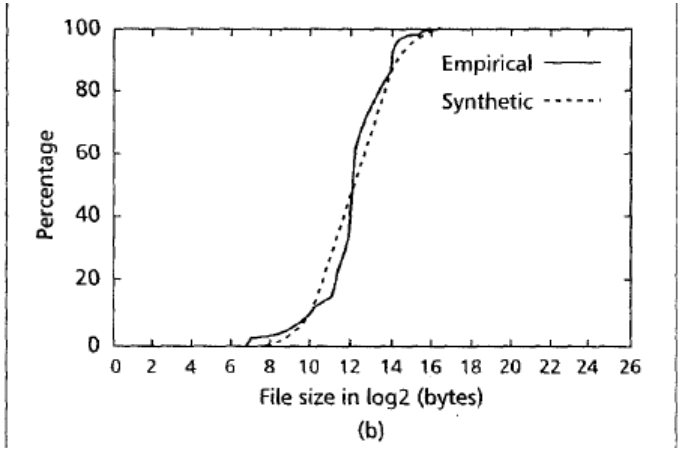


LLCD File Sizes

Características del Servidor

Tamaños de ficheros





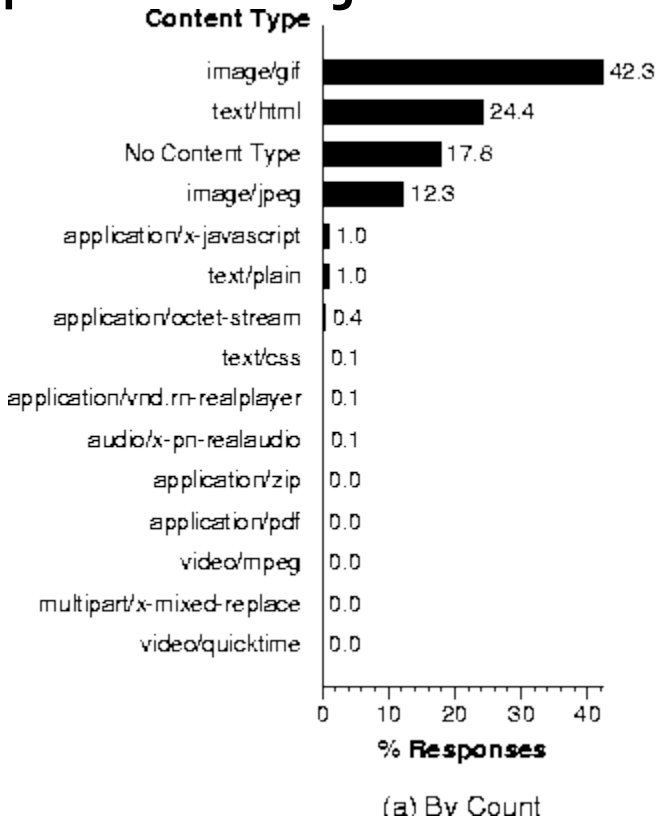
Distribución de tamaños de ficheros únicos
(a) frecuencias, (b) frecuencia acumulada
(c) cola.
A Workload Characterization Study of the 7998 World Cup Web Site
Martin Arlitt and Tai Jin,

Características del Servidor

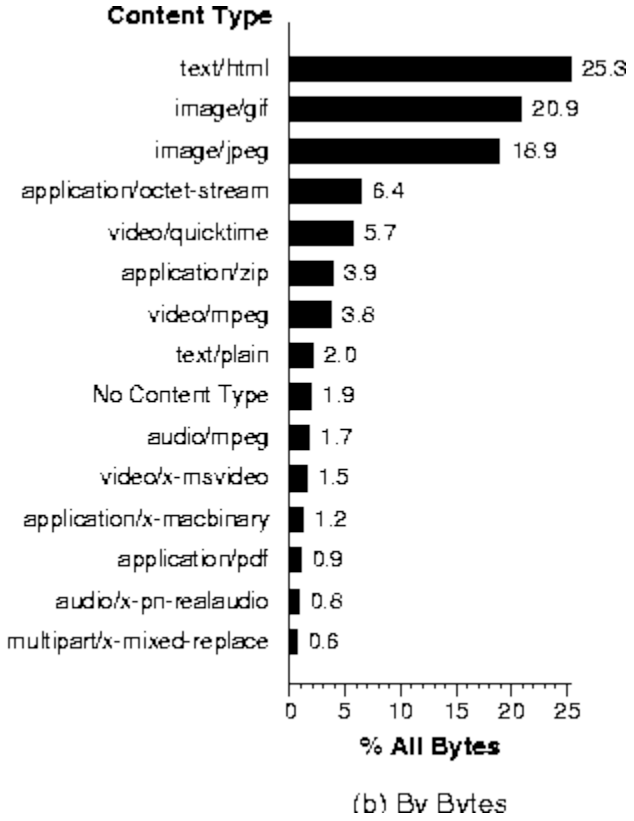
- Tamaños de ficheros solicitados
  - Tamaños de todos los ficheros transferidos por el servidor.
  - Puede ser bastante diferente de la distribución de tamaños de ficheros, dado que el mismo fichero puede ser transferido varias veces
  - Se necesita un modelo detallado para poder especificar de forma realista la red.
  - Se ha demostrado que también exhiben "colas pesadas" (heavy tails).

Características del Servidor

Tipos de objetos



(a) By Count

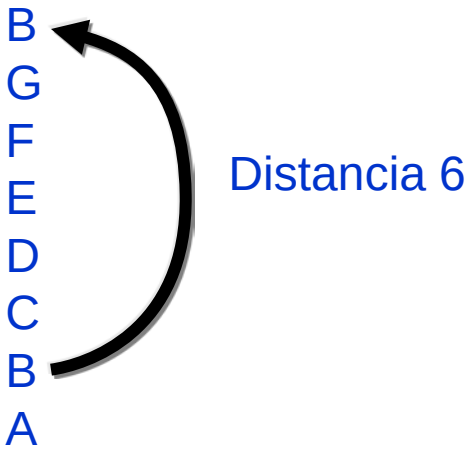


(b) By Bytes

"Organization-Based Analysis of Web-Object Sharing and Caching ". Alec Wolman, Geoff Voelker, Nitin Sharma, Neal Cardwell, Molly Brown,
Tashana Landray, Denise Pinnel, Anna Karlin, Henry Levy .
Department of Computer Science and Engineering
University of Washington

Características del Servidor

- Localidad Temporal
  - Se refiere a la posibilidad de que una vez que se ha solicitado un fichero, se vuelva a solicitar en un futuro cercano.
  - Se necesita un modelo detallado para poder especificar de forma realista caches y buffers.
  - Se puede medir en términos de distancia LRU



Características del Servidor

Localidad Espacial

- Se refiere a la estructura de los objetos web:
  - Hay ficheros embebidos que se transfieren siempre que se solicita un archivo HTML.
- Se necesita un modelo detallado para poder especificar de forma realista caches, buffers y la red.
- Se puede caracterizar por el número de referencias embebidas en los objetos web

Características de los Clientes

- Población de clientes
  - Número de clientes con sesiones de navegación establecidas en un servidor
- Longitud de Sesión
  - Número de objetos web que son solicitados por un cliente en una sesión de navegación.
  - Se necesita un modelo detallado para poder especificar de forma realista la red

Características de los Clientes

- Proceso de llegada de clientes
  - Frecuencia con la que los clientes establecen nuevas sesiones de navegación.
  - Se necesita un modelo detallado para poder especificar de forma realista la CPU y los buffers del servidor y la red
  - Las medidas empíricas sugieren que exhiben una alta variabilidad.

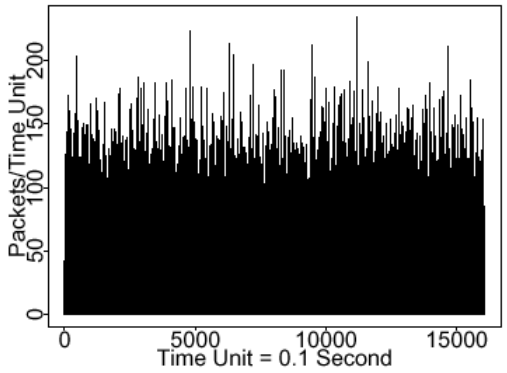
Características de los Clientes

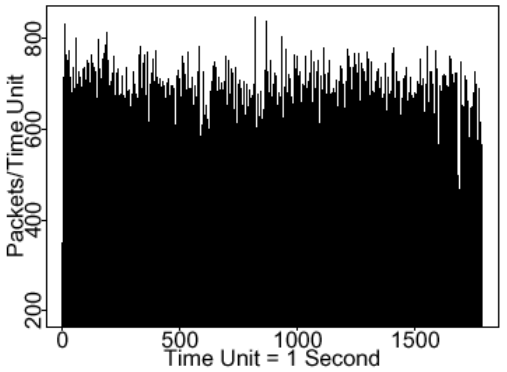
- Periodos de inactividad
  - Tiempos de inactividad (tiempo de pensar) entre peticiones de objetos web durante una sesión de navegación.
  - Se necesita un modelo detallado para poder especificar de forma realista el tráfico en la red.
  - Las medidas empíricas sugieren que exhiben "colas pesadas" (heavy tails).

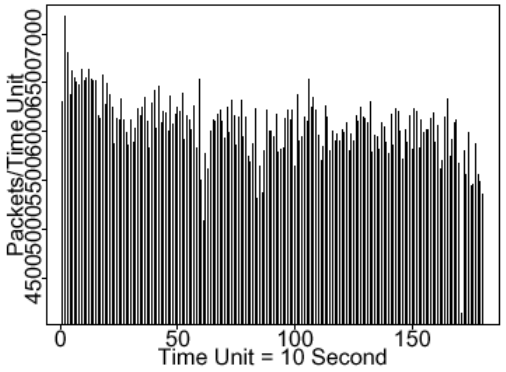
Características de la Red

Utilización del ancho de banda

- Número de bits por segundo que fluyen entre el servidor y el cliente (cuenta, media, varianza)







Características de la Red

Utilización del ancho de banda

- Auto-similitud
  - Se refiere al escalado de la variabilidad (burstiness).
  - Una serie estacionaria en el tiempo de media cero X_t, t = 1,2, … se dice que es "exactamente de segundo-orden auto-similar" si

$$X_t \overset{d}{=} m^{-H} \sum_{i=m(t-1)+1}^{mt} X_i$$

  - Para ½ ≤ H < 1 y todo m > 0
  - Donde $\overset{d}{=}$ significa igualdad de distribución

Características de la Red

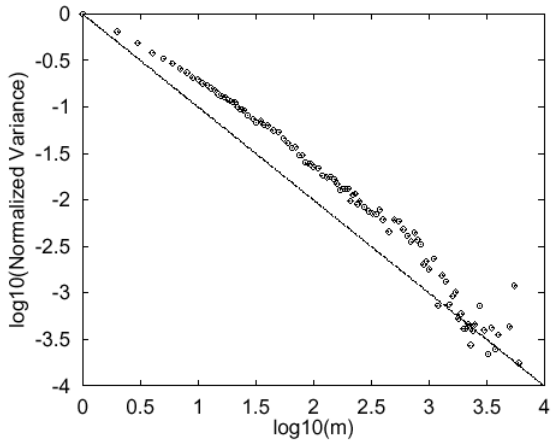
- Utilización del ancho de banda
  - Gráficas Varianza-Tiempo
    - Son un mecanismo para evaluar la auto-similitud.
    - Se dibuja la

$$\mathrm{var}\left(\sum_{i=m(t-1)+1}^{mt} X_i \right)$$

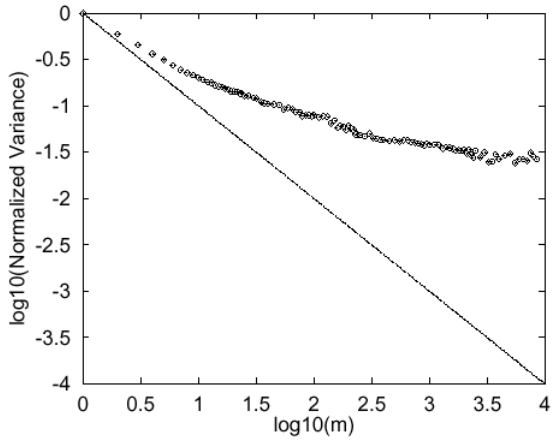
    - En función de m en ejes log-log, donde Xi son medidas de tráfico en bytes o paquetes por unidad de tiempo.
    - Un comportamiento lineal con una pendiente mayor de -½ sugiere auto-similitud

Características de la Red

- Utilización del ancho de banda
  - Gráficas Varianza-Tiempo



No Indication of Self-similarity



Consistent with Self-similarity

Características del protocolo

- HTTP/1.0
  - La versión utilizada más ampliamente
  - Protocolo "stop and wait"
  - Caching
    - Cabecera de expiración y peticiones "if-modified-since".

Características del protocolo

- HTTP/1.1
  - Reduce la latencia cuando la red es el cuello de botella.
  - Conexiones persistentes
    - Reduce el "overhead" de establecimiento/cierre de conexión
    - El servidor tiene que manejar gran número de conexiones abiertas.
  - Pipelining
    - Ganancia en eficiencia depende del número de ficheros embebidos en un objeto web
  - Caching
    - Requerimientos más explícitos (cabecera "cache control")
  - Compresión
    - Distinción entre codificación del contenido y codificación de transferencia.

Parte 4:
Entorno de Test

Configuración hardware

Configuración software

Entrono hardware

- Configuración del servidor
  - Nodo simple o configuración distribuida
  - Velocidad de CPU
  - Tamaño de RAM (comparada con el sistema de ficheros)
  - Configuración del disco (importante si el conjunto de ficheros >> RAM)
  - Configuración de red

Emulación WAN

- NISTnet
  - Simula retardos y pérdidas de paquetes
  - Corre en un sistema separado entre el servidor y los clientes
  - http://www.ncsl.nist.gov/itg/nistnet/index.html
- dummynet
  - Parte de FreeBSD
  - Simula los efectos de limitaciones de ancho de banda, retardos de propagación, colas de tamaño limitado, pérdidas de paquetes,
  - Intercepta paquetes en el S.O. y los pasa por pipes.
  - http://www.iet.unipi.it/~luigi/ip_dummynet/

Software del Servidor Web

- Servidores de conexión por proceso
  - Un conjunto de procesos está listo para servir las peticiones entrantes
  - Intensivo en recursos
  - Apache/Unix
- Servidores Multi-hilo
  - Servidores de hilo por conexión
    - Apache/NT
  - Servidores de hilo por función
    - IIS
- Servidores de hilo único
  - Zeus, Flash

Configuración del servidor

- Time outs
  - Muchos navegadores y servidores soportan "connection keep alive" en HTTP/1.1
  - Los servidores pueden decidir cuándo cerrar la conexión
  - El timeout típico son 15 segundos
- Máximo num. de conexiones
  - Debe configurarse por debajo de la sobrecarga en condiciones normales (dependiente del sistema)
  - Debe configurarse muy alto si se desea testar condiciones de sobrecarga

Configuración del S.O.

- Límites del sistema
  - Altamente dependientes del sistema
  - En linux, hay un límite en descriptores de procesos e i-nodes por proceso (se cambian en /proc)
  - En NT, se deben añadir una gran cantidad de parches y valores del registro.
  - En open.specbench.org/osg/web96/tunings/ se encuentran parámetros de sintonización para DEC, IBM, HP
- TCP
  - En linux los parámetros no están accesibles fácilmente
  - En NT se requieren algunos cambios

Benchmarks y Generadores de Carga

- Los benchmark se utilizan para generar estadísticas de prestaciones que puedan utilizarse para comparar productos distintos.
- Los generadores de carga genéricos se pueden utilizar para generar carga de acuerdo con las especificaciones que se les suministre

Especificaciones de benchmark

WebStone

- Creado por Silicon Graphics y adquirido por Mindcraft
  - http://www.mindcraft.com/webstone/
  - Simula la actividad de múltiples clientes sobre un servidor, cada uno pidiendo un conjunto standard de ficheros.
  - Cada máquina cliente puede ejecutar múltiples instancias.
  - La última versión puede testear HTML, CGI's y API's
  - Problemas:
    - Deja muchas opciones libres.
    - La velocidad de la red no se especifica.
    - El conjunto de objetos es tan pequeño que puede ser cacheado.
    - Solo hace GET, no hace POST

Especificaciones de benchmark

- SPECweb99
  - Creado por Standard Performance Evaluation Corporation (SPEC)
    - http://www.specbench.org/osg/web99/
    - Un poco más riguroso que el WebStone
    - La carga modela un ISP típico
      - Medida de conexiones simultáneas en lugar de operaciones HTTP
      - Simulación de conexiones con velocidad limitada
      - GETs dinámicos y estáticos; operaciones POST.
      - Conexiones Keepalive (HTTP 1.0) y persistentes (HTTP 1.1).

Especificaciones de benchmark

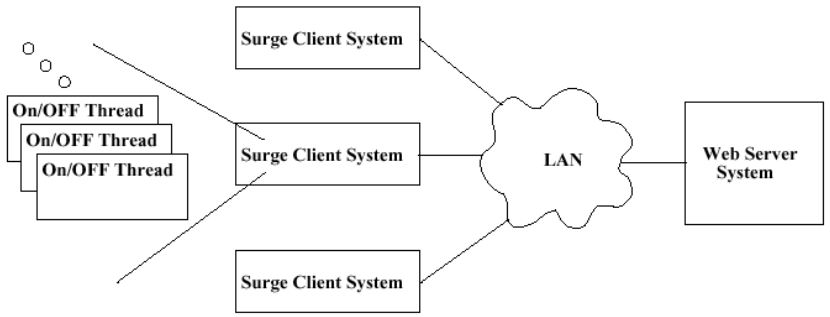
- TPC-W
  - Benchmark para el procesado de transacciones web de comercio electrónico creado por el Transction Processing Council.
  - http://www.tpc.org/tpcw/
  - Características
    - Múltiples sesiones de navegación on-line
    - Generación de páginas dinámicas con acceso y actualización de bases de datos
    - Objetos web consistentes
    - Ejecución simultánea de múltiples tipos de transacciones on-line
    - Las bases de datos contienen muchas tablas con variedad de tamaños, atributos y relaciones
    - Integridad de transacciones
    - Contienda en el acceso y actualización de datos.

Generadores de carga

- Conducidos por traza
  - Webjamma
    - De Virginia Tech's Network Research Group
    - Repite ficheros de trazas de servidor (una URL por línea)
    - Soporta múltiples procesos de cliente por sistema
    - No soporta localidad temporal (orden de petición) entre sistemas cliente separados
    - http://www.cs.vt.edu/~chitra/tools.html
  - S-Clients
    - http://www.cs.rice.edu/~gaurav/gaurav_research.html

Generadores de carga

- Otros
  - SURGE
    - Scalable URL Request Generator
    - http://www.cs.bu.edu/fac/crovella/links.html
    - La carga se basa en la idea de Usuario Equivalente (UE)
      - La carga de un UE es aproximadamente igual a la que generaría un cliente
    - Cada UE es un hilo ON/OFF
    - Soporta GET HTTP/1.0 y HTTP/1.1



Generando Carga

- Radview's *Webload*
  - Herramienta de test para NT
  - Repite guiones (proporcionados por los diseñadores) de usuarios que acceden a un conjunto de objetos web.
  - Provee algunas capacidades de generación de carga distribuida.
  - Proporciona visualización gráfica de resultados de prestaciones.
  - Soporta tests básicos y de regresión con una gran variedad de capacidades adicionales (cookies, passwords, etc.)
  - http://www.radview.com/

Generando Carga

- GUERNICA (http://www.gii.upv.es/web_architecture/)
  - GUERNICA es la solución de iSOCO en el campo del software generador de carga para aplicaciones web.
  - Principales funcionalidades:
    - Definición de carga de usuario basada en navegaciones, con las que se refleja el comportamiento dinámico.
    - Ejecución de carga mediante modelo distribuido y modelo local.
    - Representación gráfica de resultados y exportación a otros formatos.
    - Interpretación 3D de resultados.
    - Contempla comandos especiales del navegador y el protocolo HTTP (redireccionamiento automático, acceso a páginas cacheadas, etc).
    - Simulación de tiempos de reflexión de los usuarios.
    - Soporte a protocolo HTTP 1.0, HTTP 1.1, HTTPS, cookies, etc.
    - Portabilidad completa entre sistemas operativos, ya que se trata de una aplicación 100% puro Java.

Generando Carga

Arquitectura GUERNICA