

Técnicas de medida y de presentación de los resultados

- Introducción
 - > Principios de medida
 - > Proceso de medida
 - > Técnicas de medida
 - Eventos
 - Muestreo
- Herramientas de medida
- Monitorización
- Presentación de resultados

Medir

- ¿Cuáles son los recursos más utilizados en mi sistema?
- ¿Cuál es el ancho de banda real de la red?
- ¿Cuál es el tiempo de respuesta medio de las transacciones de venta?
- ¿Cuál es el ancho de banda de la red corporativa consumido por el tráfico de Internet?

Debemos medir ...

- Para saber cómo se está comportando mi sistema.
 - Ajuste de sistema. (tuning)
- Para modelar el sistema.
 - Modelado analítico
- Para conocer la capacidad real de mi sistema.
 - Planificación de capacidad.
 - Mejora.
- Para comprobar el nivel de servicio del sistema (SLA)

Medir ... ¿Cómo?

- ¿Cuáles son las fuentes de información de los datos de rendimiento?
- ¿Cuáles son las herramientas disponibles para la medición de tiempos y de utilización de recursos?
- ¿Qué técnicas utilizaré para calcular los parámetros de entrada del modelo?

Principios de medida

- Medida permite tratar los fenómenos de naturaleza compleja de forma cuantitativa.
- Medir un sistema: recoger información de la actividad del sistema mientras da servicio a los usuarios (reales o simulados).
- Los experimentos de medida necesitan elegir la carga de trabajo a procesar durante la recogida de datos.
- Para realizar evaluaciones comparativas es necesario que los experimentos de medida sean repetibles.

Proceso de medida.

- Especificar las medidas
- Especificar los Puntos de prueba
- Instrumentar herramientas de medida
- Recolectar Medidas
- Analizar y Transformar los datos
 - Programas de aplicación
 - Comunicaciones
 - Sistema Operativo
 - Hardware

Proceso de medida.

- Especificar las medidas
 - Se debe decidir qué variables han de ser medidas.
 - - Paquetes enviados por segundo
 - -Tiempo de utilización de CPU
 - - N° de faltas de página
- Especificar los Puntos de prueba
- Instrumentar herramientas de medida
- Recolectar Medidas
- Analizar y Transformar los datos

Proceso de medida.

- Especificar las medidas
- Especificar los Puntos de prueba
 - Dónde se recogerán los datos.
 - En la LAN
 - En el servidor
- Instrumentar herramientas de medida
- Recolectar Medidas
- Analizar y Transformar los datos

Proceso de medida.

- Especificar las medidas
- Especificar los Puntos de prueba
- Instrumentar herramientas de medida
 - Se deben desarrollar/parametrizar las herramientas para monitorizar el sistema.
 - Configurar las herramientas para que las variables especificadas sean medidas durante el periodo de observación y la información requerida se guarde.
- Recolectar Medidas
- Analizar y Transformar los datos

Proceso de medida.

- Especificar las medidas
- Especificar los Puntos de prueba
- Instrumentar herramientas de medida
- Recolectar Medidas
- Analizar y Transformar los datos

Proceso de medida.

- Especificar las medidas
- Especificar los Puntos de prueba
- Instrumentar herramientas de medida
- Recolectar Medidas
- Analizar y Transformar los datos
 - Se suelen recoger grandes cantidades de datos lineales que corresponden a la observación detallada del sistema en operación.
 - Agregar
 - Cálculo de porcentajes, ratios, medias,....

Técnicas de Medida: Categorías

- Dirigidas por eventos (event-driven)
 - Genera un registro cada vez que un acontece un evento de un conjunto preestablecido.
 - Los datos de las actividades del sistema se recogen capturando todos los acontecimientos(eventos) asociados a los cambios de estado y se registran en el mismo orden en el que se producen.
- Traza
 - Es similar a la detección de eventos pero se registra el evento y parte del estado del sistema que identifica unívocamente al sistema.

Técnicas de Medida: Categorías

- Muestreo
 - Inspecciona periódicamente el estado del sistema o del componente y calcula métricas en base a estas observaciones.
 - Se interrumpe el sistema a intervalos regulares o aleatorios para detectar el estado de alguno de los componentes.
 - Técnica que interfiere menos en el sistema que la de detección de eventos.
 - El tamaño de la muestra influye en la precisión de la medida.
- Indirectas.
 - Se utiliza cuando la métrica en la que se está interesado no puede obtenerse de forma directa.
 - Ej: Verificación de la velocidad de acceso a memoria L1

Detección de Eventos

- *Evento*: cambio de estado del sistema.
 - Comienzo de una operación E/S; su finalización
 - Transición de la CPU de ocupada a desocupada.
 - Fallo de página, inicialización de un programa,...
- El estado de un sistema está determinado por un conjunto de variables.
- La recogida de información se produce cuando se detecta la ocurrencia del *evento*.
- Los eventos asociados a una función de programa se denominan *eventos software (software event)*.
- Detección de *eventos software* mediante la inserción de un código suplementario en puntos específicos del sistema operativo.
 - Ocurrencia evento a detectar => activación de rutina de recogida de datos.
 - Información recogida: fecha, hora y tipo de evento.

Detección de eventos

- El conjunto de instrucciones y datos recogidos para este fin constituye un monitor.
- Los basados en la detección de eventos se denominan *Event-driven monitor* y los datos recogidos traza de eventos.
- Problemas:
 - Grandes volúmenes de datos
 - Ralentización del funcionamiento del sistema. Sobrecargas de hasta el 30%
- Ejemplo.
 - Sistemas UNIX: se inserta un registro en el accounting log cada vez que finaliza un proceso.

Muestreo

- Técnica estadística para analizar solamente una parte de la población objetivo. Esta parte se denomina muestra.
- Información recogida en instantes predeterminados.
- La precisión depende del tamaño muestral y de una adecuada selección de la muestra.
- Equilibrio entre precisión y sobrecarga (overhead).

Muestreo

- **Objetivos**
 - Medir las fracciones de un intervalo de tiempo que cada componente del sistema consume en sus diferentes estados.
 - Herramientas de medida utilizadas por los usuarios del sistema
 - Se supone que la carga es estacionaria.
 - Situaciones casi estacionarias se consiguen dividiendo el intervalo de medida en periodos relativamente cortos (algunos minutos) y agrupando los bloques de *datos homogéneos*.
 - Seguir la evolución del sistema y predecir su comportamiento futuro.
 - La medida es parte integrante del proceso de control de la actividad del sistema operativo
 - Las funciones del sistema operativo tienen que adaptarse dinámicamente a las variaciones de las condiciones del sistema.
 - Estimaciones de los parámetros en función de la historia reciente de sus valores.
 - La evolución no se puede predecirse de forma determinista, se crean modelos basados en la estructura estocástica subyacente.

Herramientas de medida: Monitores

- **Monitor:** es una herramienta utilizada para observar las actividades de un sistema.
- La función principal de un monitor es recoger datos relativos a la operación de un sistema.
- El monitor debe ser un observador de la actividad del sistema y nunca un participante en la misma.
 - No debe afectar a la operación del sistema que ha de medirse.
- Los monitores están caracterizados por
 - Modo: evento, traza, muestreo.
 - Tipo: hardware, software, firmware e híbrido.

Monitores: Términos

- Evento : Cambio en el estado del sistema
- Traza: Es un registro de eventos que normalmente incluye el instante del evento, el tipo de evento y los parámetros más importantes asociados con dicho evento.
- Sobrecarga (Overhead): Perturbación de la operación del sistema. Consumo de CPU y disco para almacenamiento.

$$\text{Sobrecarga} = \text{Tiempo_ejecución_monitor} / \text{Intervalo_de_medida}$$

- Dominio: conjunto de actividades observables por el monitor.
- Tasa de entrada: La frecuencia máxima de eventos que el monitor puede observar de forma correcta.
- Resolución: El nivel de desagregación con la que la información puede ser observada.
- Tamaño de entrada: El nº de bits/bytes de información recogida en un evento

Tipos de monitores

- Nivel de implementación
 - Monitor software
 - Monitor hardware
 - Monitor firmware (microcódigo)
 - Monitor híbrido
- Mecanismo
 - Dirigido por eventos (*event-driven*)
 - Dirigido por temporización (*time-driven*) monitor de muestreo.
- Capacidad de mostrar resultados
 - Monitores *en línea*
 - Monitores *batch*

Bloques funcionales de los monitores

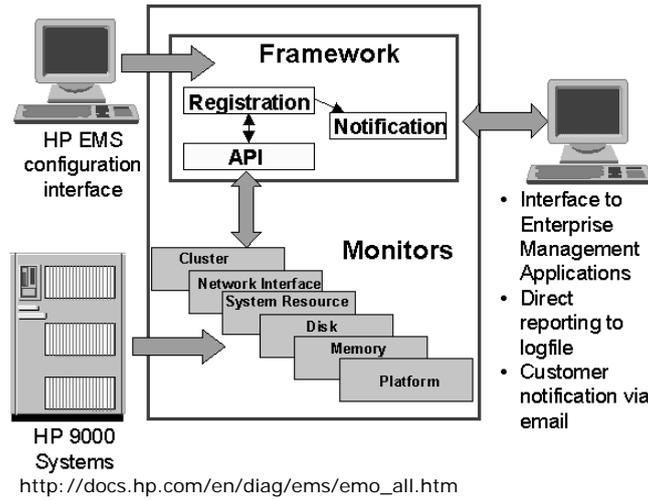
- Observación
 - Recoge datos de los componentes individuales del sistema.
- Recolección
 - Recoge los datos provenientes de varios observadores.
- Análisis
 - Analiza los datos recolectados por los diferentes recolectores. Rutinas estadísticas que resumen las características principales de los datos.
- Presentación
 - Relacionado con la interfaz de usuario. Informes, pantallas, alarmas.
- Interpretación
 - Entidad inteligente que obtiene interpretaciones significativas de los datos.
- Consola
 - Interfaz para controlar los parámetros y estados del sistema.
- Gestión
 - Toma de decisiones para configurar o cambiar los parámetros del sistema

Monitores hardware

- Equipos que se incorporan al sistema mediante sondas.
- No consumen recursos del sistema. Sobrecarga menor.
- Tipos
 - *Sondas*: Utilizadas para observar señales en puntos del hardware del sistema.
 - *Contadores*: Cuando sucede un determinado evento se incrementa un contador.
 - *Elementos lógicos*: Combinan señales enviadas por diferentes sondas mediante puertas lógicas. Las combinaciones se utilizan para indicar eventos que pueden incrementar los contadores.
 - *Comparadores*: se utilizan para comparar contadores o valores de señales con valores establecidos de antemano.
 - *Mapping Hardware*: Establece histogramas de las cantidades observadas.
 - Combinación de varios contadores y comparadores.
 - *Timer*: Utiliza para “marcas de tiempo” o para disparar operaciones de muestreo.

Monitores hardware:
Ej.: Servicios de monitorización de eventos HP

HP Event Monitoring Service

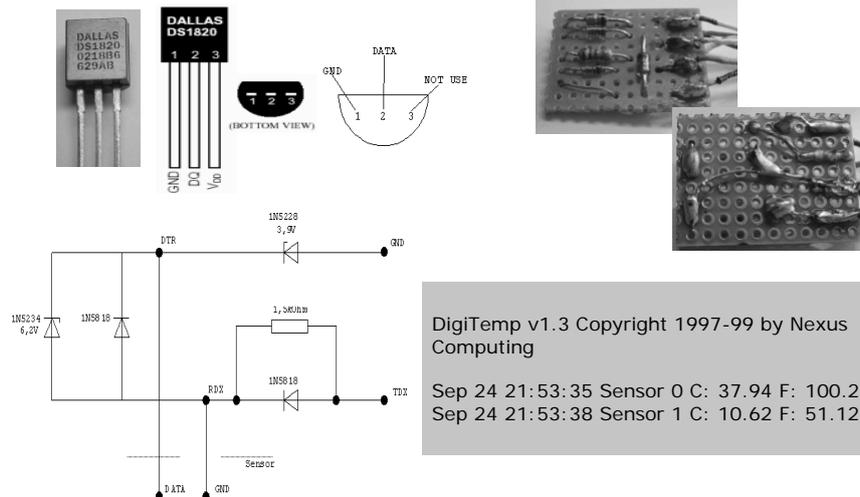


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

23

Monitores hardware:
Ej.: Monitor de temperatura para Linux



<http://www.linuxfocus.org/English/November2003/article315.shtml>

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

24

Monitores hardware: Ej.: μGURU Clock



Unidad externa de monitorización de hardware para placas base AV8
(ABIT AV8 3rd Eye)

<http://www.tech-hounds.com/review1/ReviewsComplete.html>

Monitores Software

- Conjunto de rutinas embebidas en el software de un sistema cuya finalidad es recoger estado y eventos del sistema.
- Monitorización de sistemas operativos, redes y bases de datos.
- Activación implica ejecución de varias instrucciones lo que puede provocar una sobrecarga importante.
- Dependencia del sistema operativo.
- Medidas
 - Nivel de sistema: estadísticas de utilización del sistema
 - Utilización de CPU, utilización de discos, nº total de operaciones E/S.
 - Nivel de programa: muestran información relacionada con programas.
 - Identificación del programa, *elapsed time*, t° de cpu,
 - Los sistemas de *accounting* constituyen la fuente principal de información acerca de la ejecución de programas

Monitores Software: Elementos del diseño

- Mecanismo de activación
 - *Trap instruction*: Mecanismo de interrupción software que transfiere el control a una rutina de recolección de datos.
 - *Trace mode*: Cambio del procesador al modo traza. En este modo, la ejecución de la instrucción se interrumpe cada instrucción y el control se transfiere a la rutina de recolección de datos. Procedimiento con una gran sobrecarga. Se utiliza para monitorizar aplicaciones donde no se medirá el tiempo entre eventos.
 - *Timer interrupt*: Se transfiere el control a la rutina de recolección de datos en intervalos de tiempo fijos. Este mecanismo se denomina muestreo. Se recomienda para eventos frecuentes ya que la sobrecarga es independiente de la tasa de eventos.
- Tamaño del buffer.
 - Los datos se recogen inicialmente en buffer y se transfieren posteriormente al disco.
 - Tamaño del buffer en función de la tasa de entradas, tamaño de la entrada y tasa de vaciamiento.

Monitores Software: Elementos del diseño

- Número de buffers.
 - Al menos dos. El proceso se organiza de forma circular, el proceso de recogida (vaciamiento de buffer) sigue al proceso de monitorización (llenado del buffer).
- *Buffer Overflow*
 - Puede darse el caso de que todos los buffers se llenen. El proceso de monitorización puede sobrescribir el buffer o detenerse.
- Análisis de datos
 - Capacidad de procesamiento de los datos a medida que son observados.
 - Reduce la cantidad de espacio necesario.

Monitores Software: Elementos del diseño

- Interruptor on/off
 - Capacidad de activar / desactivar el proceso de monitorización
- Lenguaje de programación
 - Es recomendable que los monitores desarrollados a medida estén escritos en el mismo lenguaje del sistema monitoreado, si es posible.
- Prioridad
 - Ejecución asíncrona : prioridad baja. Operaciones del sistema poco afectadas
 - Si el retraso de las observaciones de los eventos es importante la prioridad alta para que los retrasos en su ejecución no den valores de tiempo sesgados.
- Monitorización de eventos anormales
 - El monitor tiene que ser capaz de observar eventos normales y anormales.

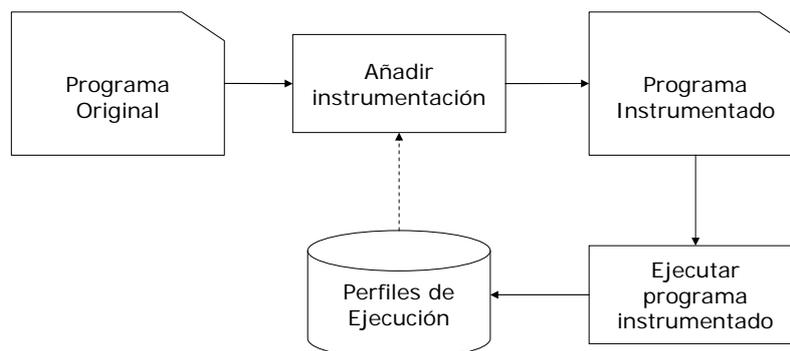
Consideraciones temporales

- Un monitor software ha de proporcionar medidas temporales, duración de los fenómenos observados.
- Muestreo o medidas directas
- Medida directa: instante de finalización – instante de comienzo.
 - En ocasiones no muy preciso y de muy difícil implementación.
 - Procesos con nivel de prioridad muy alto.
- Es necesario un reloj de sistema.
 - Accesible vía software. En este caso podemos tomar medidas directas.
 - Para muestreo ha de ser capaz de provocar la interrupción del sistema en intervalos regulares.
- Frecuencia del reloj da la resolución del sistema o cantidad mínima de tiempo medible por él.

Monitores de ejecución de programas

- Son monitores software diseñados para observar programas de aplicación. Se conocen también como analizadores u optimizadores de programas.
- Útiles en la construcción de programas de alto rendimiento y en la mejora del rendimiento de programas existentes.
- Las principales tareas de ayuda al analista de rendimiento
 - Traza (*Trace*)– proporciona el perfil de ejecución del programa.
 - Temporización (*Timing*)– localiza el tiempo que consumen los diferentes módulos.
 - Ajuste (*Tunning*)– optimización de código, E/S y paginación.
 - Verificación de asserts (*Assertion checking*) – verificación de las relaciones asumidas/definidas entre las variables de un programa.
 - Análisis de cobertura (*Coverage analysis*)– Para determinar el nivel de adecuación de una prueba de ejecución.

Pasos en la monitorización de ejecución de programas



Sistema de Seguimiento (*Accounting Logs*)

- Proporcionan información acerca de la utilización de los recursos y del rendimiento del sistema.
- Principales ventajas
 - Suelen estar incorporados al Software proporcionado con el S.O
 - Los datos se recogen durante la operación normal del sistema
 - Sobrecarga pequeña, se pueden recoger datos durante periodos largos de tiempo.
- Principales inconvenientes
 - No incorporan una utilidad de análisis
 - Los datos pueden no ser del nivel de granularidad deseada.
 - Información a nivel de usuario y tareas de usuario, no de nivel de sistema.

Sistema de Seguimiento (*Accounting Logs*)

- Los datos que se suelen recoger son:
 - Nombre del programa, tiempo de comienzo y finalización, tiempo de CPU
 - Nº y bytes totales leídos y escritos en disco.
 - Nº y total de bytes leídos y escritos en el terminal.
 - Nº de faltas de página y páginas leídas desde/hacia el dispositivo de paginación.
- Se pueden calcular los parámetros de carga a partir de los datos recogidos.

Generadores de ficheros *LOG*

- Contienen información acerca de los servicios solicitados a un sistema, la respuesta proporcionada y el origen de las peticiones.
- Los ficheros *log* generados por servidores funcionales software tales como HTTP y DBMS contienen información útil para el analista de rendimiento.
- Ejemplo
 - El *access log* es un tipo de *web log*, que puede ser útil para calcular parámetros de carga.
 - Contiene información acerca del origen de la petición, del servicio solicitado, y de la respuesta proporcionada.
 - Se pueden calcular el intervalo de medida, la tasa de llegadas (pet./seg) y estadísticas acerca del tamaño del documento.

Web Server Log File

- Son ficheros de texto plano (ASCII) independientes de plataforma. Los campos separados por tabs y espacios.
 - Si un campo no tiene valor aparece un guión.
- Cuatro tipos de logs en el servidor
 - Transfer (access) log
 - Error Log
 - Referer Log
 - Agent Log
- Los ficheros Transfer y Error son los estándar. Los otros pueden añadirse como formato extendido en el transfer.
- Cada transacción de protocolo http, se complete o no, se registra en los logs y en algunos caso en varios logs.
 - Un error se registra en el transfer y en error.
- Logs en servidores apache
 - <http://httpd.apache.org/docs/2.0/logs.html>

Extended Log File Format

- W3C Working Draft WD-logfile-960323 (<http://www.w3.org/TR/WD-logfile>)
- Formato
 - El carácter # al inicio de cada línea indica una directiva. Puede ser:
 - Version: <integer>. <integer>. La versión del formato extendido de fichero log empleado. En el ejemplo : versión 1.0.
 - Fields: [<specifier>...] Especifica los campos registrados en el log.
 - Software: <string> Identifica el software que ha generado el log.
 - Start-Date: <date> <time> La fecha y hora en que el log fue iniciado.
 - End-Date: <date> <time> La fecha y hora en que el log fue finalizado.
 - Date: <date> <time> La fecha y hora en la que la entrada fue añadida.
 - Remark: <text> Comentarios. Usualmente deben ignorarse por las herramientas de análisis

```
#Version: 1.0
#Date: 12-Jan-1996 00:00:00
#Fields: time cs-method cs-uri
00:34:23 GET /foo/bar.html
12:21:16 GET /foo/bar.html
12:45:52 GET /foo/bar.html
12:57:34 GET /foo/bar.html
```

Web Server Log File

dos.xyz.com -- [24/Jan/2003:13:43:14 -0400] "GET 2.gif HTTP/1.0" 200 2555

- Dirección o DNS
 - La del ordenador que efectúa la petición HTTP
 - Recoge el IP y el DNS (si está configurado)
- Time Stamp
 - Formato: Fecha, hora y offset de la hora Greenwich (GMT) [GMTx100]
 - Se puede deducir cuanto tiempo tardó un visitante en cambiar de página.

Web Server Log File

dos.xyz.com -- [24/Jan/2003:13:43:14 -0400] "GET 2.gif HTTP/1.0" 200 2555

- Tipo de petición
 - GET – petición estándar de documento o programa
 - POST – Indica al servidor que a continuación vienen datos.
 - HEAD – Enlazar programas de comprobación y descargas.
- Status Code
 - Estado de la respuesta que proporciona el servidor
 - Éxito – series 200
 - Redirección – series 300
 - Fallo – series 400
 - Error servidor – series 500
- Volumen transferido en bytes.

Extended Log File Format FIELDS

- Cada campo FIELDS puede ser de una de las siguientes formas:
 - identifier
 - Identifier relates to the transaction as a whole.
 - prefix-identifier
 - Identifier relates to information transfer between parties defined by the value prefix.
 - prefix(header)
 - Identifies the value of the HTTP header field header for transfer between parties defined by the value prefix. Fields specified in this manner always have the value <string>.
 - The following prefixes are defined:
 - c Client
 - s Server
 - r Remote
 - cs Client to Server.
 - sc Server to Client.
 - sr Server to Remote Server, this prefix is used by proxies.
 - rs Remote Server to Server, this prefix is used by proxies.
 - x Application specific identifier.
- The identifier cs-method thus refers to the method in the request sent by the client to the server while sc(Referer) refers to the referer: field of the reply. The identifier c-ip refers to the client's ip address.

Extended Log File Format

Identificadores

- No requieren prefijo:
 - date Date at which transaction completed, field has type <date>
 - time Time at which transaction completed, field has type <time>
 - time-taken Time taken for transaction to complete in seconds, field has type <fixed>
 - bytes bytes transferred, field has type <integer>
 - cached Records whether a cache hit occurred, field has type <integer> 0 indicates a cache miss.

- Requieren prefijo
 - ip IP address and port, field has type <address>
 - dns DNS name, field has type <name>
 - status Status code, field has type <integer>
 - comment Comment returned with status code, field has type <text>
 - method Method, field has type <name>
 - uri URI, field has type <uri>
 - uri-stem Stem portion alone of URI (omitting query), field has type <uri>
 - uri-query Query portion alone of URI, field has type <uri>

Extended Log File Format

Ejemplo

- Fecha y hora
- Dirección ip (c-ip)
- Nombre del servidor (s-computername)
- Método (cs-method)
- Recurso visitado (cs-uri-system)
- Consulta solicitada (cs-uri-query)
- Estado del protocolo (sc-status)
- Bytes enviados (sc-bytes)
- Bytes recibidos (cs-bytes)
- Tiempo consumido (time-taken)

- Información sobre el proceso.
 - Suceso de proceso (s-event)
 - Tipo de proceso (s-process-type)
 - Total tº usuario (s-user-time)
 - Total tº núcleo (s-kernel-time)
 - Total procesos (s-total-procs)
 - Procesos activos (s-active-procs)
 - Total procesos terminados (s-stopped-procs)

Extended Log File Format Ejemplo

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2003-10-01 10:06:08
#Fields: date time c-ip s-computername cs-method cs-uri-stem cs-uri-query
sc-status sc-bytes cs-bytes time-taken
2003-10-01 10:06:08 157.88.124.59 INFISS GET /iss/index.html - 304 164 539 10
2003-10-01 10:06:08 157.88.124.59 INFISS GET /iss/scripts/estilo2.css - 304 162 438 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/scripts/menu.js - 304 163 435 10
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/scripts/abrir.js - 304 164 437 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/dptoWeb.gif - 304 163 440 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/home06.gif - 304 162 438 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/period.gif - 304 164 440 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/pizarra.gif - 304 164 441 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/info02.gif - 304 164 440 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/scripts/fecha.js - 304 163 436 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/scripts/fecha.js - 304 163 436 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/dptoWeb.gif - 304 163 440 0
2003-10-01 10:06:10 157.88.124.59 INFISS GET /iss/imagenes/home06.gif - 304 162 438 0
```

Extended Log File Format Ejemplo

Monthly Statistics for August 2006		
Total Hits	6786	
Total Files	4483	
Total Pages	2716	
Total Visits	312	
Total KBytes	45.95 MB	
Total Unique Sites	185	
Total Unique URLs	132	
Total Unique Referrers	21	
Total Unique Usernames	1	
Total Unique User Agents	69	
	Avg	Max
Hits per Hour	21	299
Hits per Day	522	903
Files per Day	344	598
Pages per Day	208	412
Visits per Day	24	37
KBytes per Day	3.53 MB	11.19 MB
Hits by Response Code		
Code 200 - OK	4483	66.06%
Code 206 - Partial Content	17	0.25%
Code 301 - Moved Permanently	22	0.32%
Code 302 - Found	767	11.30%
Code 304 - Not Modified	1386	20.42%
Code 400 - Bad Request	14	0.21%
Code 401 - Unauthorized	23	0.34%
Code 404 - Not Found	74	1.09%

Extended Log File Format Ejemplo

Code 404 Monthly Statistics for August 2006 (Top 20 of 33 URLs)

#	Hits	URL
1	5	/mysql-test/main.php
2	3	/PMA/main.php
3	3	/mysql/main.php
4	3	/test/main.php
5	3	/db/main.php
6	3	/dbtest/main.php
7	3	/web/test/main.php
8	3	/test/pma/main.php
9	3	/test/test/main.php
10	3	/test/mysql/main.php
11	3	/test2/main.php
12	3	/mysqltest/main.php
13	3	/test/main.php
14	3	/main.php
15	3	/test-2.5.6/main.php
16	3	/test-2.5.4/main.php
17	3	/test-2.5.1/main.php
18	3	/test-2.2.3/main.php
19	3	/test-2.2.6/main.php
20	2	/mytest/main.php

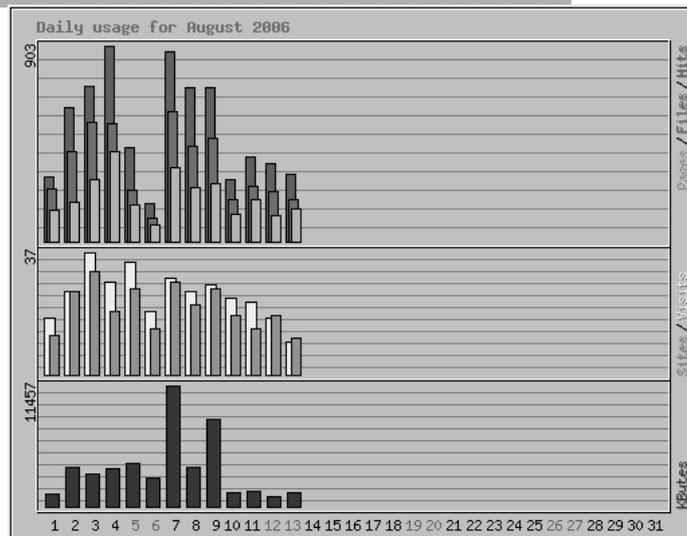
[View All URLs](#)

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

45

Extended Log File Format Ejemplo



M.A.V.S. oct-10

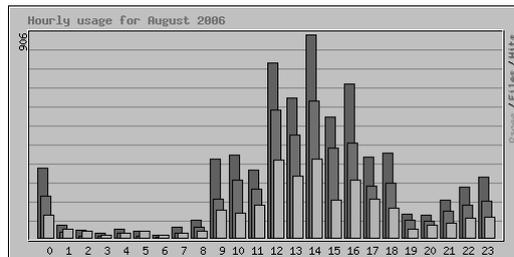
Dpto. Informática – ETSII – U. Valladolid

46

Extended Log File Format Ejemplo

Daily Statistics for August 2006												
Day	Hits		Files		Pages		Visits		Sites		KBytes	
1	299	4.41%	240	5.35%	145	5.34%	17	5.45%	12	6.49%	1.14 MB	2.48%
2	617	9.09%	415	9.26%	183	6.74%	25	8.01%	25	13.51%	3.66 MB	7.96%
3	718	10.58%	548	12.22%	284	10.46%	37	11.86%	31	16.76%	3.06 MB	6.66%
4	903	13.31%	544	12.13%	412	15.17%	28	8.97%	19	10.27%	3.46 MB	7.53%
5	433	6.38%	236	5.26%	171	6.30%	34	10.90%	26	14.05%	4.02 MB	8.74%
6	176	2.59%	107	2.39%	76	2.80%	19	6.09%	14	7.57%	2.67 MB	5.80%
7	877	12.92%	598	13.34%	343	12.63%	29	9.29%	28	15.14%	11.19 MB	24.35%
8	711	10.48%	442	9.86%	249	9.17%	25	8.01%	21	11.35%	3.64 MB	7.92%
9	712	10.49%	478	10.66%	265	9.76%	27	8.65%	26	14.05%	8.09 MB	17.61%
10	286	4.21%	196	4.37%	125	4.60%	23	7.37%	18	9.73%	1.31 MB	2.85%
11	389	5.73%	254	5.67%	193	7.11%	22	7.05%	14	7.57%	1.47 MB	3.20%
12	357	5.26%	232	5.18%	122	4.49%	17	5.45%	18	9.73%	1005.52 KB	2.14%
13	308	4.54%	193	4.31%	148	5.45%	10	3.21%	11	5.95%	1.27 MB	2.75%

Extended Log File Format Ejemplo



Daily Statistics for August 2006							
Files	Pages		KBytes		Total	%	
	Total	Avg	Total	Avg			
187	4.17%	7	100	3.68%	67.01 KB	871.16 KB	1.83%
25	0.56%	3	40	1.47%	8.30 KB	107.95 KB	0.23%
7	0.16%	2	31	1.14%	4.62 KB	60.00 KB	0.13%
6	0.13%	0	11	0.41%	566 bytes	7.19 KB	0.02%
22	0.49%	1	22	0.81%	6.85 KB	89.08 KB	0.19%
7	0.16%	2	28	1.03%	7.93 KB	103.10 KB	0.22%
6	0.18%	0	11	0.41%	1.36 KB	17.68 KB	0.04%
7	0.16%	1	20	0.74%	6.31 KB	82.03 KB	0.17%
8	0.16%	3	48	1.07%	22.21 KB	288.80 KB	0.61%
9	5.16%	13	173	3.86%	88.53 KB	1.12 MB	2.45%
10	5.44%	19	257	5.73%	85.15 KB	1.08 MB	2.33%
11	4.42%	16	215	4.80%	93.23 KB	1.18 MB	2.58%
12	11.43%	43	569	12.69%	263.78 KB	3.35 MB	7.29%
13	9.18%	35	458	10.22%	257.79 KB	3.27 MB	7.12%
14	13.35%	47	611	13.63%	517.37 KB	6.57 MB	14.30%
15	7.96%	30	399	8.90%	189.70 KB	2.41 MB	5.24%
16	10.06%	32	423	9.44%	785.42 KB	9.97 MB	21.70%
17	5.39%	17	228	5.09%	150.75 KB	1.91 MB	4.17%
18	5.34%	18	245	5.47%	638.97 KB	8.11 MB	17.63%

Extended Log File Format Ejemplo

Top 10 of 132 Total URLs					
#	Hits		KBytes		URL
1	1106	16.30%	6.97 MB	15.17%	/test/test.php3
2	436	6.42%	6.66 MB	14.50%	/test/
3	228	3.36%	251.85 KB	0.54%	/test/css/default.css
4	90	1.33%	7.90 MB	17.20%	/stat/usage_200608.html
5	75	1.11%	380.42 KB	0.81%	/test/loginredirect.php3
6	57	0.84%	11.24 KB	0.02%	/robots.txt
7	39	0.57%	7.54 KB	0.02%	/favicon.ico
8	38	0.56%	1.11 MB	2.41%	/test/pdf/system_mac.pdf
9	30	0.44%	288.19 KB	0.61%	/stat/
10	28	0.41%	323.50 KB	0.69%	/stat/site_200608.html

Top 10 of 132 Total URLs By KBytes					
#	Hits		KBytes		URL
1	90	1.33%	7.90 MB	17.20%	/stat/usage_200608.html
2	1106	16.30%	6.97 MB	15.17%	/test/test.php3
3	2	0.03%	6.91 MB	15.04%	/download/ie5en.bin
4	436	6.42%	6.66 MB	14.50%	/test/
5	1	0.01%	5.17 MB	11.24%	/download/MFJ_2.2.5.smu.bin
6	38	0.56%	1.11 MB	2.41%	/test/pdf/system_mac.pdf
7	15	0.22%	1000.30 KB	2.13%	/test/classes/bm.jar
8	15	0.22%	750.22 KB	1.59%	/test/classes/zm.jar
9	5	0.07%	730.65 KB	1.55%	/stat/site_200607.html
10	7	0.10%	703.23 KB	1.49%	/stat/usage_200607.html

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

49

Extended Log File Format Ejemplo

Top 6 of 6 Total Entry Pages					
#	Hits		Visits		URL
1	436	6.42%	9	32.14%	/test/
2	12	0.18%	7	25.00%	/test/loginhelp.php3
3	1106	16.30%	4	14.29%	/test/test.php3
4	3	0.04%	3	10.71%	/test/disclaimer/disclaimer.php3
5	90	1.33%	3	10.71%	/stat/usage_200608.html
6	8	0.12%	2	7.14%	/test/info.php3

Top 10 of 12 Total Exit Pages					
#	Hits		Visits		URL
1	436	6.42%	168	69.71%	/test/
2	26	0.38%	18	7.47%	/test/logout.php3
3	90	1.33%	16	6.64%	/stat/usage_200608.html
4	1106	16.30%	15	6.22%	/test/test.php3
5	12	0.18%	8	3.32%	/test/loginhelp.php3
6	28	0.41%	4	1.66%	/stat/site_200608.html
7	3	0.04%	3	1.24%	/test/disclaimer/disclaimer.php3
8	8	0.12%	3	1.24%	/test/info.php3
9	75	1.11%	2	0.83%	/test/loginredirect.php3
10	16	0.24%	2	0.83%	/test/subscription.php3

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

50

Monitor Windows NT/2000/XP

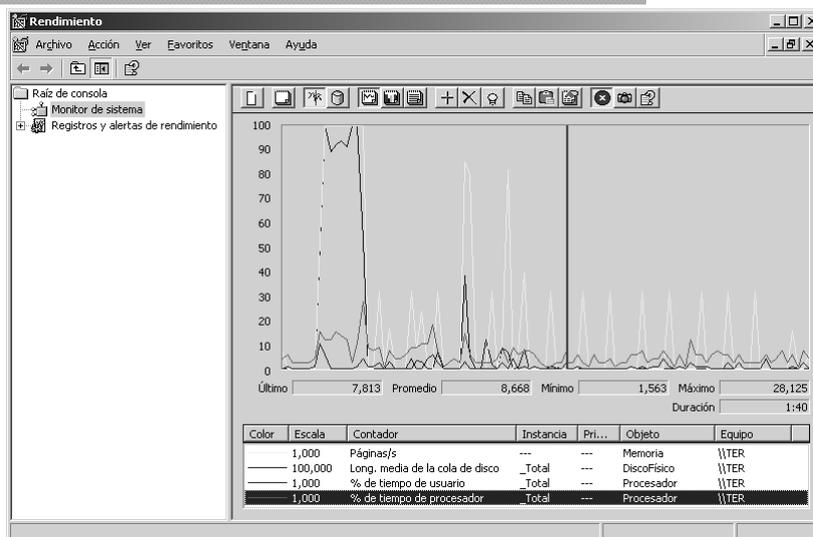
- Mide el rendimiento de los objetos del sistema.
- Define los datos de rendimiento que recolecta en términos de objetos, contadores e instancias.
- Un objeto es un mecanismo estándar para identificar y utilizar un recurso del sistema. Representan
 - Procesos
 - Hilos (threads)
 - Memoria
 - Dispositivos físicos.
- Los objetos tienen instancias
 - Si el sistema tiene n procesadores el objeto Processor podrá tener tantas instancias como procesadores
- Si el disco físico tiene dos particiones el objeto Logical Disk tendrá dos instancias.

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

51

Perfmon



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

52

Monitor Windows NT/2000/XP

- Un contador es una entidad para la que están disponibles los datos de rendimiento.
 - Los contadores están organizados por tipo de objeto.
 - Existe un único conjunto de contadores para cada objeto tipo.
- Contadores
 - Instantáneos: muestran las medidas más recientes.
 - Medias: representan un valor de un intervalo de tiempo y muestran la media de las dos últimas medidas.
 - Diferencia: muestran la diferencia en la dos medidas consecutivas. La última y la anterior.
 - Porcentaje.

Monitor Windows NT/2000/XP

- Los datos recogidos se muestran como
 - Gráfico : Muestra el valor del contador a lo largo del tiempo.
 - Informe : Muestra el valor del contador
 - Alerta: Dispara un evento cuando el valor del contador alcanza un umbral determinado
 - Log : Almacena los datos de los contadores en disco. A posteriori se pueden leer con perfmon para elaborar los gráficos y los informes.
- Detección de problemas y ajuste de rendimiento (performance tuning)
 - Vistas gráfico, informe, alerta
- Planificación de la capacidad y análisis del rendimiento
 - Vista log.

Monitor Windows: Objetos

- Logical disk
 - Informa de la actividad y utilización de las particiones y volúmenes de los discos del sistema.
 - Hay que activar explícitamente la monitorización de los discos: diskperf
 - diskperf -y : activa los contadores del disco (-n los desactiva)
 - diskperf -yd : activa discos físicos (-yv activa los discos lógicos)
- Memoria
 - Utilización de la RAM para el almacenamiento de código y datos.
- Process
 - Actividad del proceso, que es un objeto software que representa un programa en ejecución.
- Processor
 - Actividad del procesador, CPU.
- System
 - Estadísticas de los contadores del sistema.
- ... etc.

Monitor Windows: Configuración

- Disco
 - Utilización
 - LogicalDisk [%Free Space ; %Disk Time]
 - Physical Disk [Disk Reads/seg; Disk Writes/seg]
 - Cuellos de Botella
 - LogicalDisk [Avg Disk Queue Length]
 - Physical Disk [Avg Disk Queue Length (all instances)]
- Memoria
 - Utilización
 - Memory [Available Bytes ; Cache Bytes]
 - Cuellos de Botella
 - Memory [Pages/seg; Page Faults/seg; Page Inputs/seg; Page Reads/seg]
 - Paging File [%Usage Object (all instances)]

Monitor Windows: Configuración

- Red
 - Utilización
 - Network Segment [% Net Utilization]
 - Throughput
 - Network Interface [Bytes total/seg ; Packets/seg.]
 - Server [Bytes Total/seg; Bytes Sent/seg; Bytes Received/seg]
- Procesador
 - Utilización
 - Processor [% Processor Time (all instances)]
 - Cuellos de Botella
 - System [Processor Queue Length (all instances); Interrupts/seg]
 - System [Context switches/seg.]

Control de la sobrecarga del monitor: Recomendaciones

- Utilizar logs en lugar de mostrar los gráficos.
- Espaciar los intervalos de recolección. En general, intervalos de 600 segundos pueden ser suficientes para la monitorización normal.
- Recoger datos en el periodo de actividad punta, en lugar de sobre un periodo extenso.
- Reducir el número de objetos a monitorizar, salvo que sean críticos para el estudio a realizar.
- Guardar el fichero de log en el disco que no se esté monitorizando.
- Limitar al máximo la utilización de los logs de traza.
- Evitar seleccionar datos no establecidos por defecto (valor medio para los logs, último valor para los gráficos). Las estadísticas se calculan para cada periodo de muestreo.

Monitorización sistemas Unix

- **Uname** Proporciona información básica del sistema

```
[ian@localhost ian]$ uname -a
Linux localhost 2.6.8.1-10mdktmp #1 SMP Wed Sep 8 16:41:52 CEST 2004 i686 Intel(R)
Celeron(R) CPU 2.00GHz unknown GNU/Linux
      Nombrehost      versión      plataforma hardware

[ian@localhost ian]$ uname --help
Usage: uname [OPTION]...
Print certain system information.  With no OPTION, same as -s.

-a, --all            print all information, in the following order:
-s, --kernel-name   print the kernel name
-n, --nodename      print the network node hostname
-r, --kernel-release print the kernel release
-v, --kernel-version print the kernel version
-m, --machine       print the machine hardware name
-p, --processor     print the processor type
-i, --hardware-platform print the hardware platform
-o, --operating-system print the operating system
--help             display this help and exit
--version          output version information and exit
```

Monitorización sistemas Unix

- **uptime** Proporciona el valor de la carga del sistema (*system load average*, número medio de procesos en la cola del núcleo)

```
14:49:07 up 18 days, 4:15, 78 users, load average: 0.00, 0.06, 0.14
Hora-actual Est. (Dias,Horas,Min)-up Usuarios carga media (ultimo min, 5min, 15min)
```

- **time** Proporciona información acerca del tiempo de ejecución de un programa. (usuario, sistema, tiempo real)

```
$ time host wikipedia.org wikipedia.org has address 207.142.131.235 0.000u
0.000s 0:00.17 0.0% 0+0k 0+0io 0pf+0w
```

Monitorización sistemas Unix

■ **df** Información acerca del uso del sistema de ficheros

```
$ df -k
Filesystem      1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4         32768         16016   52%     2271   14% /
/dev/hd2        4587520      1889420   59%    37791    4% /usr
/dev/hd9var      65536         12032   82%      518    4% /var
/dev/hd3        819200       637832   23%     1829    1% /tmp
/dev/hd1        524288       395848   25%      421    1% /home
/proc            -              -        -         -        - /proc
/dev/hd10opt     65536        26004   61%      654    4% /opt
```

Monitorización sistemas Unix

■ **ps** Estado de los procesos en ejecución en el sistema

```
ps
```

```
  PID  TTY  TIME  CMD
 6874 pts/9  0:00  ksh
 6877 pts/9  0:01  csh
  418 pts/9  0:00  csh
```

```
ps -ef
```

```
UID PID PPID C STIME TTY TIME CMD
hope 29197 18961 0 Sep27 ? 00:00:06 sshd: hope@pts/87
hope 32097 29197 0 Sep27 pts/87 00:00:00 -csh
hope 7209 32097 0 12:17 pts/87 00:00:00 ps -ef
```

Monitorización sistemas Unix

- **top** Actividad de los procesos. Muestra de forma dinámica los procesos que están consumiendo tiempo de procesador. Valor actualización por defecto 5 segs.

```
$ top
 8:48am  up 70 days, 21:36, 1 user, load average 0.28, 0.06, 0.02
47 processes: 44 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 99.6% user,  0.3% system, 0.0% nice, 0.0% idle
Mem:  256464K av, 234008K used,  22456K free, 0K shrd, 13784K buff
Swap: 136512K av,  4356K used, 132156K free,          5240K cached

  PID USER   PRI  NI  SIZE  RSS SHARE STAT  LC  %CPU  %MEM  TIME  COMMAND
 9826 carlos  0   0   388   388   308 R    0  99.6  0.1  0:22 simulador
 9831 miguel 19   0   976   976   776 R    0  0.3  0.3  0:00 top
      1 root   19   0    76    64    44 S    0  0.0  0.0  0:03 init
      :
      4 root   20  19     0     0     0 SWN  0  0.0  0.0  0:00 ksoftiq
     11 root    0 -20     0     0     0 SW<  0  0.0  0.0  0:00 recoved
```

Monitorización sistemas Unix

- Información aportada por **top**

Campo Descripción

PID Identificador del proceso
USER Nombre del usuario propietario del proceso
PRI Prioridad del proceso
NI Valor del parámetro *nice* del proceso
SIZE Memoria total en KB que ocupa (datos, código y pila)
RSS Memoria física en KB que ocupa
SHARE Memoria compartida en KB
STAT Estado del proceso: R (*running*), S (*sleeping*), Z (*zombie*), D (*uninterruptible sleep*), T (*stopped*)
Modificadores: W (*swapped out*), N (*running nicely*), > (*memory soft limit exceed*), < (*high niced level*)
LC Identificador del último procesador usado por el proceso
%CPU Uso del procesador desde la última actualización
%MEM Uso de la memoria desde la última actualización
TIME Tiempo de CPU que ha utilizado el proceso desde su inicio
COMMAND Nombre del proceso (línea de órdenes)

Monitorización sistemas Unix

- **vmstat** Muestra información relativa al sistema de memoria. Se puede especificar el intervalo entre muestras y el número de muestras.
 - r – procesos esperando a ser ejecutados
 - b – procesos durmiendo ininterrumpidamente
 - swpd – memoria virtual en uso (Kb)
 - free – memoria física libre (Kb)
 - buff – memoria usada como buffer
 - cache – memoria usada como cache
 - si – memoria intercambiada (KB/s) desde disco
 - so – memoria intercambiada (KB/s) hacia disco
 - bi – bloques de memoria por segundo enviados a disco
 - bo- bloques de memoria por segundo recibidos desde disco
 - in – interrupciones por segundo
 - cs – cambios de contexto por segundo
 - us – uso del procesador ejecutando código de usuario
 - sy – uso del procesador ejecutando código del sistema operativo
 - id – porcentaje de tiempo con el procesador ocioso.

Monitorización sistemas Unix

- **vmstat con cinco segundos de retraso**
vmstat 5
- **vmstat con cinco segundos de retraso y 10 actualizaciones**
vmstat 5 10

- **Sistema libre de actividad de paginado**

```
procs                memory      swap          io          system cpu
r  b  w  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id
0  0  0  29232 116972 4524 244900 0  0  0  0  0  0  0  0  0
0  0  0  29232 116972 4524 244900 0  0  0  0 2560 6  0  1  99
0  0  0  29232 116972 4524 244900 0  0  0  0 2574 10 0  2  98
```

- **Sistema con actividad de paginado**

```
procs                memory      swap          io          system cpu
r  b  w  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id
. . .
1  0  0  13344 1444 1308 19692 0 168 129 42 1505 713 20 11 69
1  0  0  13856 1640 1308 18524 64 516 379 129 4341 646 24 34 42
3  0  0  13856 1084 1308 18316 56 64 14 0 320 1022 84 9 8
```

Monitorización sistemas Unix

- **iostat** Monitorización de dispositivos de E/S del sistema

- Muestra estadísticas para el primer y tercer disco cada segundo.

```
iostat -w 1 disk0 disk2
```

- Muestra estadísticas totales cada 3 segundos

```
iostat -Iw 3
```

- **netstat** Monitorización de las comunicaciones de Red

- **mpstat** Monitorización del procesador principal

Monitorización sistemas Unix

sar

- **sar** System activity report

- Herramienta de muestreo sencilla que registra datos acumulados y promedio de la actividad del sistema.
- Tiempo en el que se tomó la medida.
- Modo usuario, procesos de baja prioridad, sistema, tiempo esperando por e/s bloqueadas y tiempo de cpu ociosa.
- Por defecto da información del procesador.

Monitorización sistemas Unix

sar

- Para observar la actividad de la CPU
sar
- Para observar la actividad de la CPU durante 10 minutos y almacenar los datos.
sar -o temp 60 10
- Para observar la actividad del disco
sar -d

SunOS unknown 5.10 Generic_118822-23 sun4u 01/22/2006

```
00:00:01 device      %busy  avque  r+w/s  blks/s  await  avserv
...
14:00:02 dad0        31     0.6    78    16102   1.9    5.3
          dad0,c      0     0.0    0      0       0.0    0.0
          dad0,h     31     0.6    78    16102   1.9    5.3
          dad1        0     0.0    0      1       1.6    1.3
          dad1,a      0     0.0    0      1       1.6    1.3
          dad1,b      0     0.0    0      0       0.0    0.0
          dad1,c      0     0.0    0      0       0.0    0.0
```

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

69

Monitorización sistemas Unix

sar

- sar uso de CPU

```
$ sar -u 5 10
SunOS colossus 5.8 Generic_108528-07 sun4u 03/25/05
09:48:13 %usr  %sys  %wio  %idle
09:48:18 41    35    2     22
09:48:23 41    35    3     21
09:48:28 39    36    3     22
09:48:33 41    34    0     25
09:48:38 41    34    0     25
09:48:43 39    36    0     25
09:48:48 41    34    0     25
09:48:53 41    35    0     24
09:48:58 44    38    1     18
09:49:03 42    33    2     23
Average 41    35    1     23
```

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

70

Más acerca de **sar**

- <http://www.softpanorama.org/Admin/Monitoring/Solaris/sar.shtml>
- <http://www.princeton.edu/~psg/unix/solaris/troubleshoot/sar.html>
- Inside Solaris 9
 - Bill Calkins, Dec, 2002.
 - C. 9 System Monitoring and Tuning
 - Being Proactive
 - Performance Measurement Tools
 - Measuring CPU Load
 - Monitoring Disk I/O
 - Monitoring RAM and Swap
 - **sar**
 - Sun SyMON
 - Fixing a Performance Problem
 - Summary

Monitorización de programas: **gprof**

- En sistemas Unix la herramienta más usada es **gprof**.
- Sea un programa prog.c, se tienen las siguientes etapas:
 - \$ gcc prog.c -o prog **-pg -g**
 - \$ prog

 - \$ gprof prog

 - o

 - \$ gprof prog > prog.gprof
- Opciones
 - -p : muestra la distribución del tiempo de ejecución entre los procedimientos del programa (flat profile)
 - -q : muestra la información relativa al grafo de dependencias entre los procedimientos (call profile)

Ejemplo (I)

■ bucles.c

```
#include <math.h>
double a=3.14; b=6.34; c=-3.03;
long i;

void main () {
    producto(); producto(); producto();
    division(); division();
}

producto(){
    for (i=0;i <50000000; i++)
        c=a*b;
}

division(){
    for (i=0; i<30000000; i++)
        c=a/b;
}
```

Ejemplo (II)

■ bucles.gprof

Flat profile:

Each sample counts as 0.01 seconds.

%	cumulative	self	self	self	total	
time	seconds	seconds	calls	ms/call	ms/call	name
58.85	1.00	1.00	3	333.47	333.47	producto
42.20	1.72	0.72	2	358.74	358.74	division

:

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.58% of 1.72 seconds

index	% time	self	children	called	name
[1]	100.0	0.00	1.72		<spontaneous>
		1.00	0.00	3/3	main [1]
		0.72	0.00	2/2	producto [2]
					division [3]
[2]	58.2	1.00	0.00	3	main [1]
		0.72	0.00	2/2	producto [2]
[3]	41.8	0.72	0.00	2	main [1]
		0.72	0.00	2	division [3]

:

Index by function name

[3] division [2] producto

Presentación de los resultados

- Archibald Putt (pseudónimo)
 - **It is not what you say, but how you say it ...**
 - Technology is dominated by two types of people: those who understand what they do not manage, and those who manage what they do not understand [Research/Development magazine, Jan 1976]
 - **Putt's Law and the Successful Technocrat: How to Win in the Information Age** (NY: Exposition Press 1981), (April 2006)

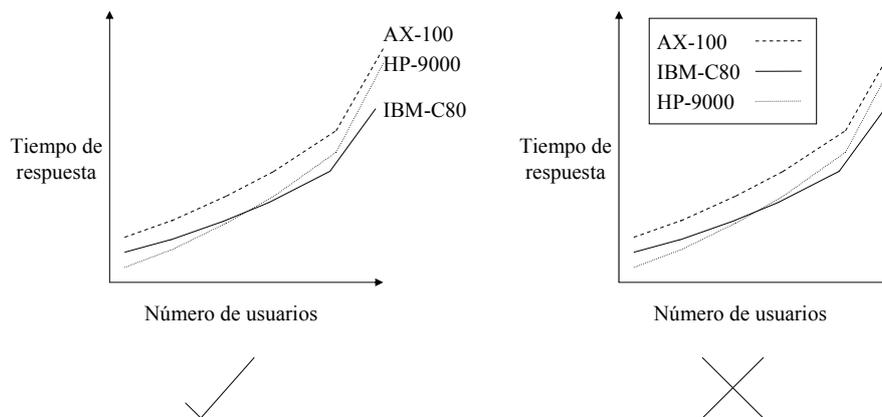
Tipos de Variables

- Variables
 - Cualitativas / Categóricas
 - Ordenadas
 - Pequeño
 - Mediano
 - Grande
 - No ordenadas
 - Carga de trabajo administrativo
 - Carga de trabajo I+D
 - Carga de trabajo de producción
 - Cuantitativas
 - Discretas
 - Continuas

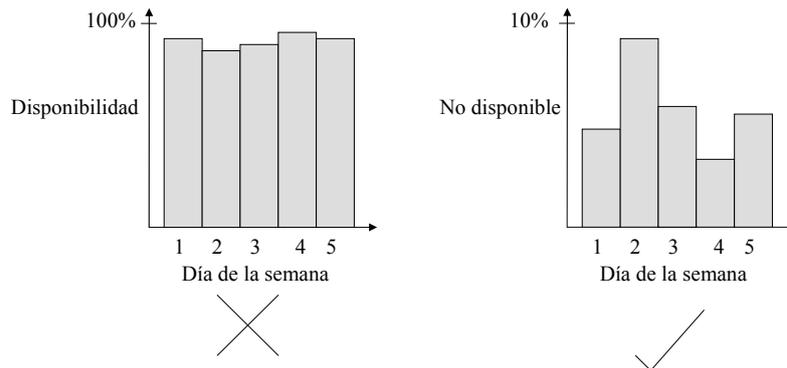
Criterios prácticos

- Mínimo esfuerzo de lectura
- Maximizar la información ofrecida
- Minimizar la carga del gráfico
- Aplicar prácticas comúnmente aceptadas
 - Variable independiente en el eje x
 - Origen representa (0, 0)
 - Escalas incrementan de izquierda a derecha y de abajo hacia arriba.
- Evitar la ambigüedad

Criterios prácticos: Mínimo esfuerzo y carga gráfica mínima



Criterios prácticos: Maximizar la información



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

79

Errores comunes

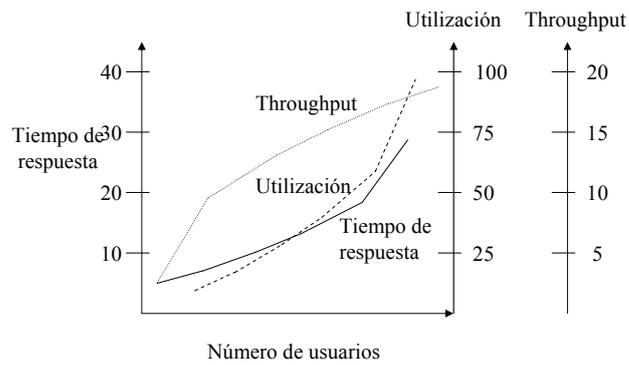
- Presentar muchas alternativas en un solo diagrama
- Presentar muchas variables “y” en un solo diagrama
- Uso excesivo de símbolos
- Introducir mensajes “extraños” en el gráfico
 - Las líneas guía solo deben introducirse si es necesario
- Selección de escalas inapropiadas
 - Elegir si se emplean escalas logarítmicas o normales
 - Si se genera automáticamente a veces será necesario configurar el ancho de cada intervalo de muestra
- Uso de gráficas de línea en lugar de gráficas de bloque

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

80

Errores comunes: Muchas variables

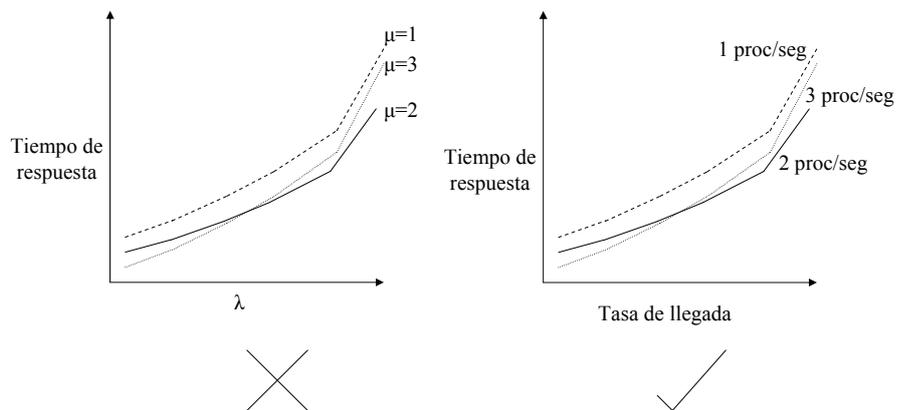


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

81

Errores comunes: Símbolos vs. Palabras clave



M.A.V.S. oct-10

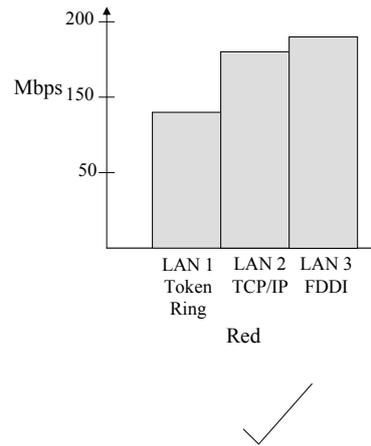
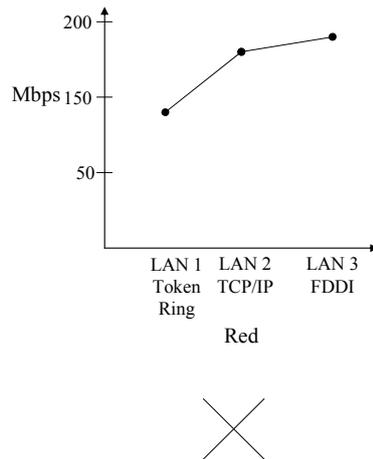
Dpto. Informática – ETSII – U. Valladolid

82

Errores comunes:

Uso diagramas de línea en lugar de diagrama de bloques

- No se puede interpolar la velocidad de las redes del ejemplo



M.A.V.S. oct-10

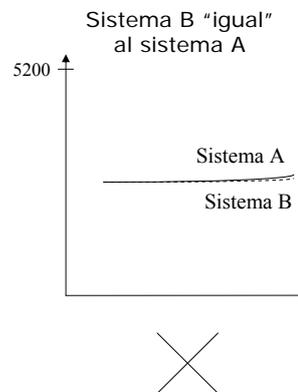
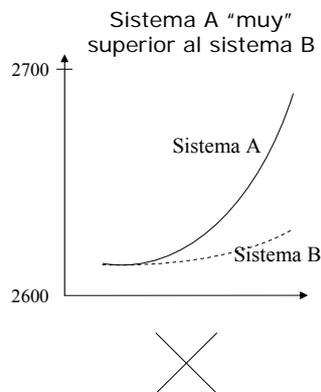
Dpto. Informática – ETSII – U. Valladolid

83

Distorsión de la información:

Uso de orígenes no-cero para resaltar la diferencia

- Se intenta exagerar o minimizar las diferencias
- El origen debe representar (0,0) a menos que haya una razón justificable para no hacerlo
- Aplicar la regla de "altura de tres cuartos"



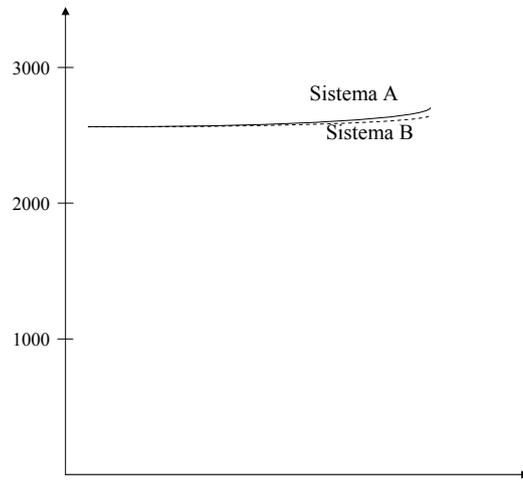
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

84

Distorsión de la información: Uso de orígenes no-cero para resaltar la diferencia

- La gráfica aplicando la regla de "altura de tres cuartos"



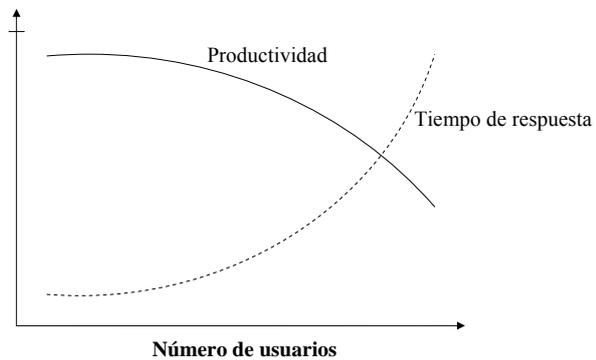
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

85

Distorsión de la información: Uso de gráficos de doble desventaja

- Incluyen dos o más gráficos que muestran resultados desventajosos.



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

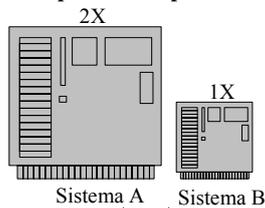
86

Distorsión de la información:

Pictogramas escalados solo en altura

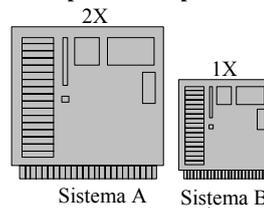
- Presentan usualmente el escalado basado solo en doblar la altura del pictograma. Dependiendo de la figura esto puede “representar” cuadruplicar la diferencia en la comparación.
- La relación de tamaños debe tomar en cuenta la superficie del pictograma utilizado.
- En el ejemplo el sistema A tiene 2 veces la productividad del sistema B.

Comparación de productividad



La superficie del primer pictograma es 4 veces la del segundo

Comparación de productividad



La superficie del primer pictograma es 2 veces la del segundo

M.A.V.S. oct-10

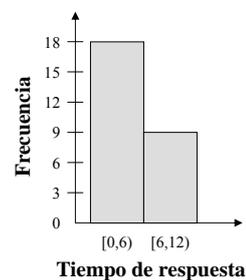
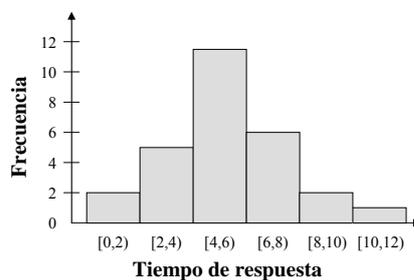
Dpto. Informática – ETSII – U. Valladolid

87

Distorsión de la información:

Uso inapropiado de tamaño de intervalos en histogramas

- Tamaños de intervalo inapropiados pueden causar el ocultamiento o pérdida de información, o pueden presentar un exceso de detalle.



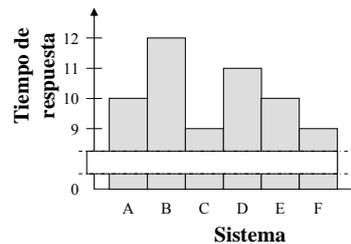
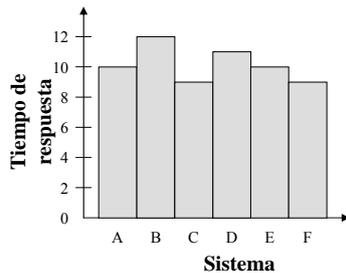
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

88

Distorsión de la información: Uso de saltos de escala en diagramas de barras

- Se usan a menudo para exagerar pequeñas diferencias



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

89

Diagramas de Gantt: Introducción

- Se espera que los recursos de un sistema sean utilizados óptimamente.
- Un recurso que tiene una utilización muy alta puede ser un cuello de botella y puede degradar el rendimiento.
- Un recurso cuya utilización es muy baja representa ineficiencia en el sistema.
- Para una apropiada utilización de los recursos del sistema es necesario que la carga represente una mezcla de tareas usando diferentes recursos y que haya un solapamiento significativo en el uso de estos recursos.
- El solapamiento en el uso de recursos puede ser mostrado a través de perfiles en diagramas de Gantt.

M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

90

Diagramas de Gantt

- Son gráficos introducidos por Henry Gantt (1861-1919) en 1917 como una herramienta para el control de la producción y para la gestión de actividades paralelas.
- En EESI se usan para mostrar la duración relativa de cualquier número de condiciones booleanas.
 - Red: ocupada o libre
- Cada condición es presentada como un conjunto de segmentos de línea.

Red

Ocupada	Libre	Ocupada	Libre
15%	5%	40%	40%
- Un diagrama Gantt usualmente muestra varios recursos.
- La posición de los segmentos se debe ordenar de forma tal que la sobreposición de varios segmentos represente el solapamiento de condiciones.
- El diagrama no indica dependencias entre tareas/recursos o condiciones.

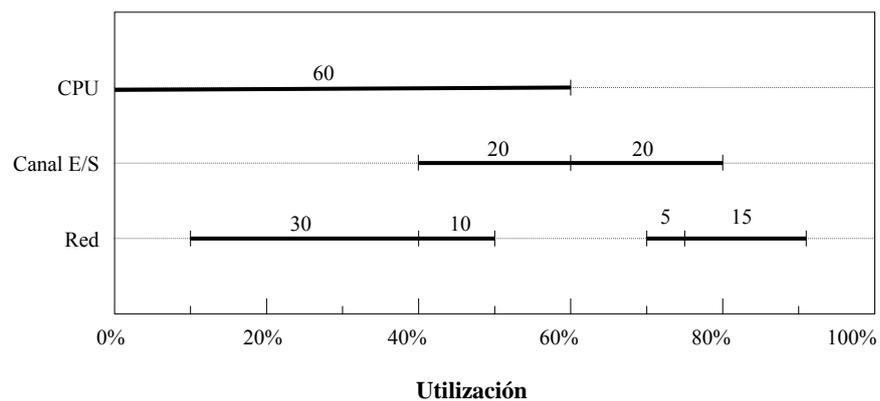
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

91

Diagramas de Gantt

- Ejemplo:



M.A.V.S. oct-10

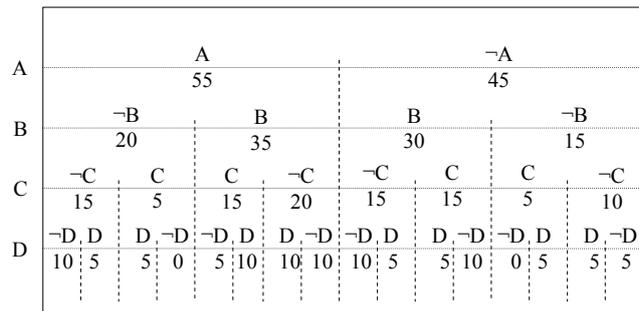
Dpto. Informática – ETSII – U. Valladolid

92

Diagramas de Gantt

- A CPU
- B Canal E/S 1
- C Canal E/S 2
- D Enlace de red

A	B	C	D	Tiempo de uso (%)
0	0	0	0	5
0	0	0	1	5
0	0	1	0	0
0	0	1	1	5
0	1	0	0	10
0	1	0	1	5
0	1	1	0	10
0	1	1	1	5
1	0	0	0	10
1	0	0	1	5
1	0	1	0	0
1	0	1	1	5
1	1	0	0	10
1	1	0	1	10
1	1	1	0	5
1	1	1	1	10
				100



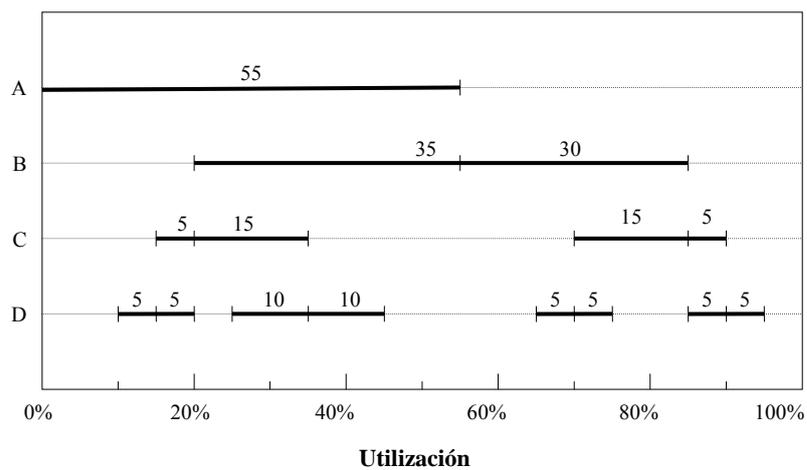
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

93

Diagramas de Gantt

- Ejemplo:



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

94

Gráficas de Kiviat: Introducción

- Introducidas por Kolence y Kiviat en 1973.
- Kiviat sugirió que se podía representar los valores de varias variables (a escala predeterminada) como semiejes irradiados desde un punto central, llamado *polo*.
- Los puntos correspondientes a los valores representados son unidos mediante rectas formando así una *gráfica de Kiviat*.
- Los ejes radiales representan 3 o más índices de rendimiento.
- Los puntos que señalan los valores de los índices representados se conectan mediante rectas con los puntos de los ejes vecinos.
- La interpretación libre de la idea de Kiviat dio origen a la aparición de varias versiones del diagrama.
- La más común es la conocida como "*versión de Kent*" (1974).

Gráficas de Kiviat: Versión de Kent

- Indica como ordenar los índices de rendimiento considerados.
 1. Seleccionar un número par de variables a se estudiadas.
 - La mitad de ellas deben ser índices de rendimiento "buenos" (HB).
 - La otra mitad deben ser índices de rendimiento "malos" (LB).
 2. Dividir el círculo en tantos sectores como variables se consideren. Un lado del primer sector debe coincidir con el semieje vertical superior.
 3. Numerar secuencialmente los semiejes partiendo por el semieje vertical superior (preferiblemente en sentido de las agujas del reloj).
 4. Asociar los índices HB a los semiejes impares y los índices LB a los semiejes pares.
- Bajo éste criterio la gráfica de Kiviat del rendimiento de un sistema ideal tiene la forma de estrella.

Gráficas de Kiviat: Versión de Kent

- Indica como ordenar los índices de rendimiento considerados.
 1. Seleccionar un número par de variables a se estudiadas.
 - La mitad de ellas deben ser índices de rendimiento "buenos" (HB).
 - La otra mitad deben ser índices de rendimiento "malos" (LB).
 2. Dividir el círculo en tantos sectores como variables se consideren. Un lado del primer sector debe coincidir con el semieje vertical superior.
 3. Numerar secuencialmente los semiejes partiendo por el semieje vertical superior (preferiblemente en sentido de las agujas del reloj).
 4. Asociar los índices HB a los semiejes impares y los índices LB a los semiejes pares.

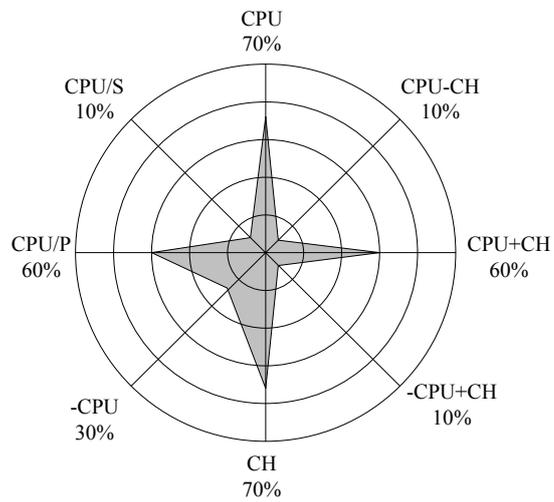
- Bajo éste criterio la gráfica de Kiviat del rendimiento de un sistema ideal tiene la forma de estrella.

Gráficas de Kiviat: Ejemplo

- Se ha monitorizado el comportamiento de la CPU y dos canales de E/S.
- Se desea representar 8 métricas de rendimiento.
- Las métricas representan el porcentaje de tiempo que el sistema permanece en determinado estado.
 - CPU CPU ocupada (HB)
 - CPU-CH Solo CPU ocupada (LB)
 - CPU+CH CPU y cualquier canal ocupados (HB)
 - -CPU+CH Cualquier canal ocupado y CPU en espera (LB)

 - CH Cualquier canal ocupado (HB)
 - -CPU CPU en espera (LB)
 - CPU/P CPU en estado problema: ejecutando instrucciones de usuario (HB)
 - CPU/S CPU en estado supervisor: ejecutando instrucciones de S.O. (sobrecarga del S.O.) (LB)

Gráficas de Kiviati: Ejemplo

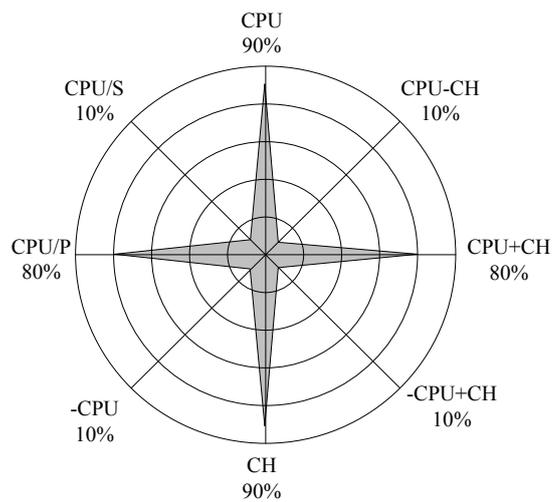


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

99

Gráficas de Kiviati: Tipo 0: Sistema óptimo

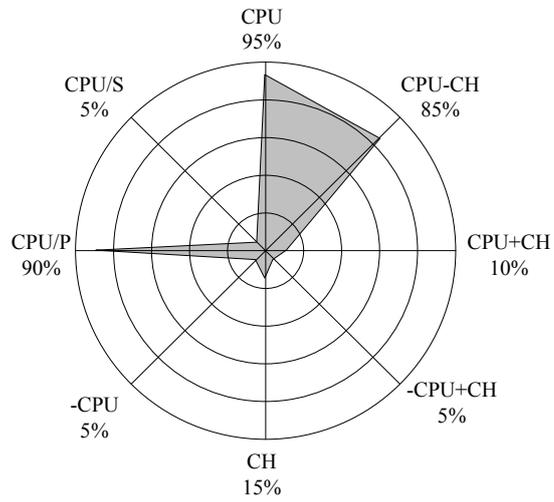


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

100

Gráficas de Kiviati:
 Tipo 1: Velero. Sistema limitado por CPU

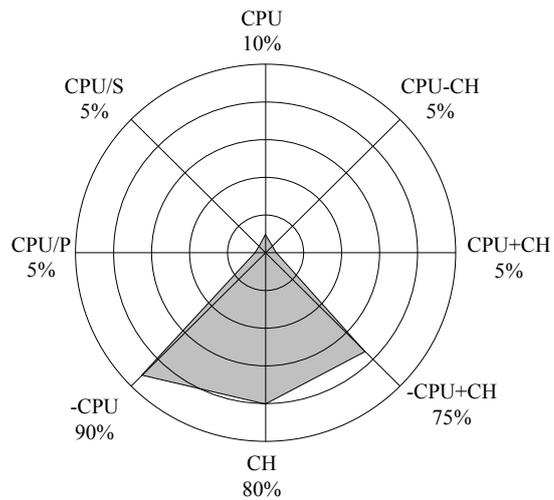


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

101

Gráficas de Kiviati:
 Tipo 2: Cuña. Sistema limitado por E/S

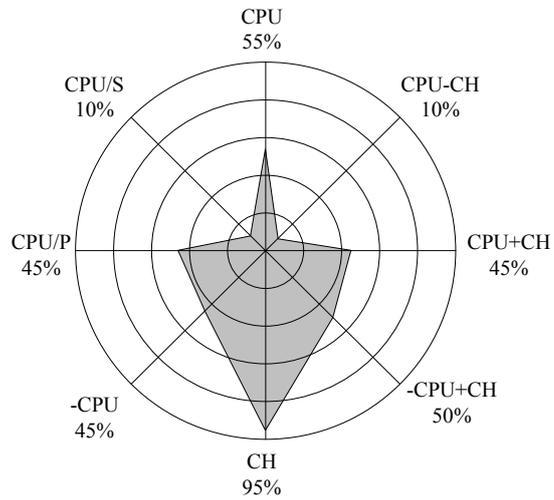


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

102

Gráficas de Kiviat: Tipo 3: Flecha. Sistema limitado por CPU y E/S

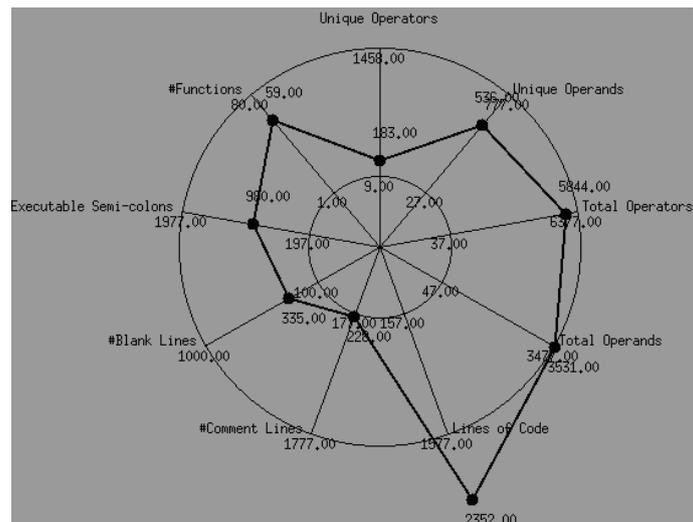


M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

103

Gráficas de Kiviat: Otras aplicaciones



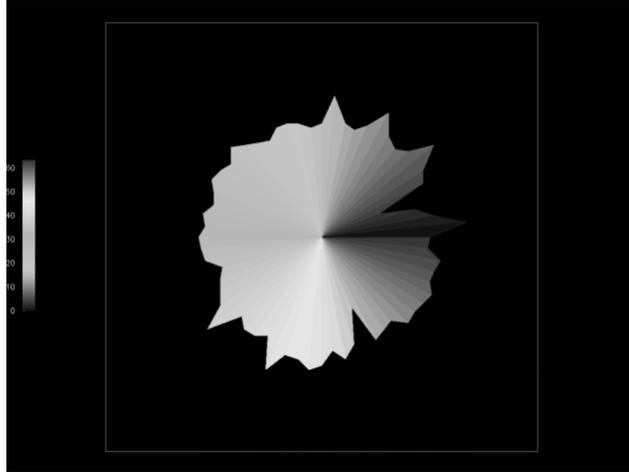
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

104

Gráficas de Kiviat: Otras aplicaciones

- Visualizing Parallel Programs and Performance
<http://www.cs.uoregon.edu/research/paraducks/papers/ieeecga95.d/>



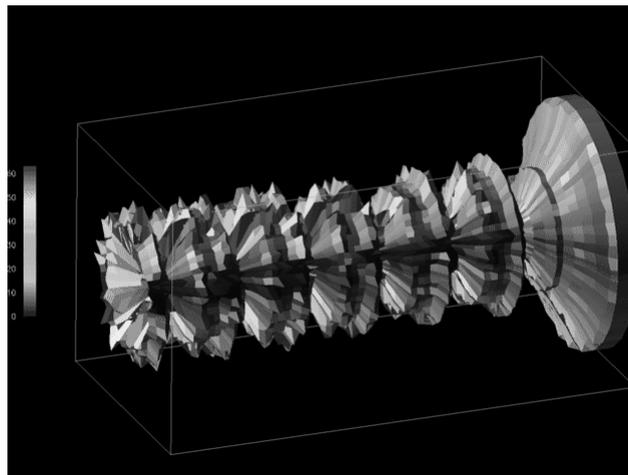
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

105

Gráficas de Kiviat: Otras aplicaciones

- Visualizing Parallel Programs and Performance



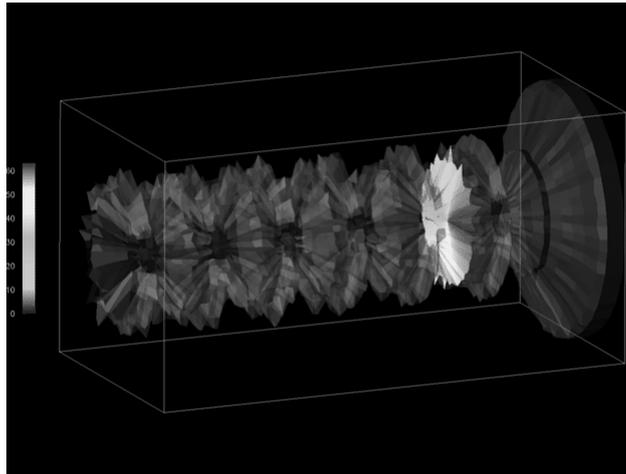
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

106

Gráficas de Kiviat: Otras aplicaciones

■ Visualizing Parallel Programs and Performance



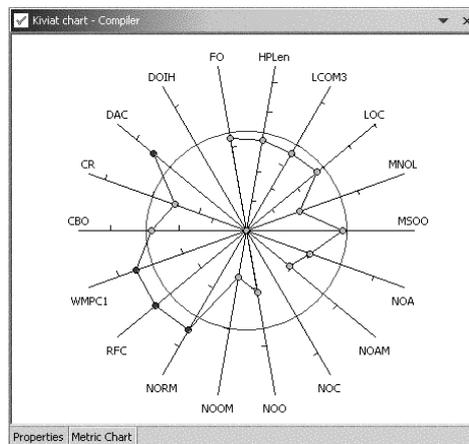
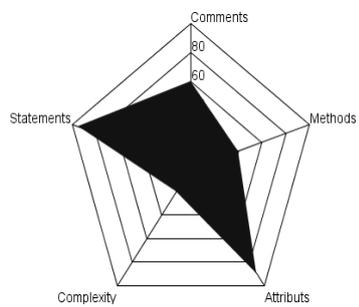
M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

107

Gráficas de Kiviat: Otras aplicaciones

■ Visualización del rendimiento de compiladores



M.A.V.S. oct-10

Dpto. Informática – ETSII – U. Valladolid

108