Diseño y Evaluación de Configuraciones

III Workload, medidas y benchmark

J.M. Drake

Notas:

Medida de Performance

- Evaluar el comportamiento temporal de un sistema requiere monitorizar el sistema bajo una carga de trabajo determinada.
- Para evaluar el comportamiento de un sistema, o comparar su comportamiento con otros con diferentes arquitecturas y/o diferentes configuraciones, hay que conocer:
  - ¿Que tipo de workload se va a producir?
  - ¿Que workload son o han sido utilizados por otros analistas?
  - ¿Cual es el workload adecuado a un sistema determinado?
  - ¿Como se puede formular el workload a partir de datos medidos?
  - ¿Como presentar los resultados de la medidas de la performance?
  - ¿Como puede cargarse un sistema con un determinado workload?

Notas:

The workload is the most crucial part of any performance evaluation project. It is possible to reach misleading conclusions if the workload is not properly selected. When the conclusions of a study are found to be unacceptable, the inappropriateness of the workload is commonly used as a criticism of the study. Like other aspects of performance evaluation, proper selection of workloads requires many considerations and judgments by the analyst, which is a part of the art of performance evaluation that comes with experience. In this chapter, a number of considerations are described that will help you make the right selection and justify your choice.

Definiciones de workload

- El workload (*workload real*) de un sistema describe los patrones y las frecuencias de requerimientos a las que está sometido un sistema por el entorno en que opera.
- Un workload de prueba (*test workload*) es el workload controlado al que se somete un sistema a fin de medir su comportamiento.
- Un benchmark es un conjunto de workloads de prueba estandarizados que se define para comparar el comportamiento de diferentes sistemas por diferentes analistas.
- Una característica clave de un workload de prueba es que sea repreducible. Por ello los workload de prueba suelen ser sintéticos y generados a partir de un modelo del comportamiento del entorno en el que opera el sistema.
- Un workload de prueba debe ser fácilmente modificable y fácilmente portables a diferentes sistemas.
- La selección del workload es la parte mas cuestionada de una análisis de performance. La selección de uno u otro workload, conduce a resultados muy diferentes.

Notas:

The term **test workload** denotes any workload used in performance studies. A test workload can be real or synthetic. A **real workload** is one observed on a system being used for normal operations. It cannot be repeated, and therefore, is generally not suitable for use as a test workload. Instead, a **synthetic workload**, whose characteristics are similar to those of the real workload and can be applied repeatedly in a controlled manner, is developed and used for studies. The main reason for using a synthetic workload is that it is a representation or model of the real workload. Other reasons for using a synthetic workload are no real-world data files, which may be large and contain sensitive data, are required; the workload can be easily modified without affecting operation; it can be easily ported to different systems due to its small size; and it may have built-in measurement capabilities.

Workload para evaluación de un monoprocesador

- En los primeros tiempos de los sistemas informáticos, el computador era el elemento relevante de la evaluación del sistema. Workload utilizados para comparar computadores y arquitectura de computadores son:
  - Mezcla ponderada de instrucciones (*Instruction Mixes*): Se evalúan el tiempo de ejecución de ciertas ejecuciones, y se dá como parámetro de evaluación una combinación ponderada de estas instrucciones o su inversa MIPS.
  - Operaciones núcleo (*kernel operation*): Orientado hacia la evaluación de los servicios que ofrece el computador y el S.O.
  - Programa sintéticos (*Synthetic programs*): Permite generar el workload a parti de un modelo parametrizado.
  - Aplicaciones benchmark (*application benchmarks*): Workload estandarizado formulados para la comparación de un determinado aspecto de diferentes sistemas.

Notas:

Mexcla ponderada de instrucciones

- Mezcla ponderada de instrucciones (*Instruction Mixes*): Se evalúan el tiempo de ejecución de ciertas instrucciones básica de la CPU, y se dá como parámetro de evaluación una combinación ponderada de estas instrucciones o su inversa MIPS.
- Existen varias listas. La mas utilizada es la de Gibson:

Load and store	31.2%	F.P. multiply	0.6%
F.P. Add and substract	6.1%	F.P. Divide	0.2%
Compares	3.8%	Shifting	4.4%
Branches	16.6	Loggical and and or	1.6%
Floating add and substract	6.9	Instruction not using registers	5.3%
Floating multiply	0.6	Indexing	18.0%
Floating divide	0.2		

- Hoy día es poco útil, ya que sólo mide la velocidad de la CPU, e incluso en los procesadores modernos la lista de instrucciones es muy reducida.

Notas:

As the number and complexity of instructions supported by the processors grew, the addition instruction was no longer sufficient, and a more detailed workload description was required. This need led several people to measure the relative frequencies of various instructions on real systems and to use these as weighting factors to get an average instruction time.

An **instruction mix** is a specification of various instructions coupled with their usage frequency. Given different instruction timings, it is possible to compute an average instruction time for a given mix and use the average to compare different processors. Several instruction mixes are used in the computer industry; the most commonly quoted one is the Gibson mix.

The Gibson mix was developed by Jack C. Gibson in 1959. The Gibson mix extended the averaging to 13 different classes of instructions, shown in the table. The average speed of a processor can be computed from the weighted average of the execution times of instructions in the 13 classes listed in the table.

Operaciones de núcleo

- Se basa en formular el workload en la medida de pruebas orientadas a la ejecución de ciertas operaciones complejas como inversión de grandes matrices búsquedas en árboles o la aplicación de algoritmos complejos como Sieve, Puzzle, Ackermann, etc.
- Es una extensión del método basado en mezcla de operaciones, pero que al estar combinadas en programas, también evalúan características como encadenamientos (pipelines), uso de memorias caché, modos de direccionamientos, etc.
- Las principales desventajas son:
  - No contempla operaciones de entrada/salida o llamadas al sistema.
  - Sólo mide la velocidad de la CPU y ciertos servicios hardware.

Notas:

The introduction of pipelining, instruction caching, and various address translation mechanisms made computer instruction times highly variable. An individual instruction could no longer be considered in isolation. Instead, it became more appropriate to consider a set of instructions, which constitutes a higher level function, a *service* provided by the processors. Researchers started making a list of such functions and using the most frequent function as the workload. Such a function is called a **kernel**. Since most of the initial kernels did not make use of the input/output (I/O) devices and concentrated solely on the processor performance, this class of kernels could be called the **processing kernel**.

A kernel is a generalization of the instruction mix. The word *kernel* means nucleus. In some specialized applications, one can identify a set of common operations, for example, matrix inversion. Different processors can then be compared on the basis of their performance on this kernel operation. Some of the commonly used kernels are Sieve, Puzzle, Tree Searching, Ackermann's Function, Matrix Inversion, and Sorting. However, unlike instruction mixes, most kernels are not based on actual measurements of systems. Rather, they became

popular after being used by a number of researchers trying to compare their processor architectures.

Most of the disadvantages of instruction mixes also apply to kernels, although some of the disadvantages related to parameter values, such as frequency of zeros and frequency of branches, no longer apply. The main disadvantage of kernels is that they do not typically make use of I/O devices, and thus, the kernel performance does not reflect the total system performance.

Programas sintéticos

- Son programas constituidos por bucles que incluyen operaciones específicas de evaluación de la CPU, operaciones de entrada/salida, llamadas al sistema, etc.
- La principal ventaja es que pueden ser desarrolladas de forma muy rápida e incorpora fácilmente las características que se quieren medir.
- Sus principales desventajas son:
  - No siempre están estandarizadas (Cada analista utiliza una diferente).
  - Sólo mide la capacidad de la CPU

Notas:

The processing kernels do not make use of any operating system services or I/O devices. As the applications of computer systems are proliferating, they are no longer used for processing-only applications. Input/output operations have become an important part of the real workloads. Initial attempts to measure I/O performance lead analysts to develop simple exerciser loops that make a specified number of service calls or I/O requests. This allows them to compute the average CPU time and elasped time for each service call. In order to maintain portability to different operating systems, such exercisers are usually written in high-level languages.

The first exerciser loop was proposed by Buchholz (1969) who called it a synthetic program. By adjusting the control parameters, one can control the number of times the request is made. Exerciser loops are also used to measure operating system services such as process creation, forking, and memory allocation.

The main advantage of exerciser loops is that they can be quickly developed and given to different vendors. It is not necessary to use real data files, which may contain proprietary information. The programs can be easily modified and ported to different systems. Further, most exercisers have built-in measurement capabilities. Thus, once developed, the measurement process is. automated and can be repeated easily on successive versions of the operating systems to characterize the relative performance gains/losses.

The disadvantages of exercisers are that they are generally too small and do not make representative memory or disk references. The mechanisms of page faults and disk cache may not be adequately exercised. The CPU-I/O overlap may not be representative. In particular, exercisers are not suitable for multiuser environments since the loops may create synchronizations, which may result in better or worse performance.

Aplicaciones benchmark populares

- **Algoritmo Sieve**: Determina todos los números primos inferiores a un gran número N. Consiste en contemplar inialmente los N primeros números, y luego eliminar los que son múltiplos de 2,3,4,..., $\sqrt{N}$
  - Se utiliza para comparar microprocesadores y lenguajes de alto nivel
- **Función de Ackermann**: Es una función recursiva basada en dos parámetros Ackermann(m,n):

$$Ackermann(m,n) = \begin{cases} m = 0 & => return \;\; n+1 \\ m > 0 & => \begin{cases} n = 0 & => return \;\; Ackerman(m-1,1) \\ n > 0 & => return \;\; Ackermann(m, n-1) \end{cases} \end{cases}$$

  Internamente la función Ackermann hace un conjunto seleccionado de operaciones representativas. La profundidad de recursión de la función Ackermann es 2^{n+m}-m
  - Mide los tiempos de ejecución de las instrucciones, los tiempos de invocación de funciones y la gestión del stack.
  - Fue propuesta para medir la eficiencias de la invocación de funciones en lenguajes de alto nivel.
- Otros benchmark utilizados son Whetsstone, Linpack, Dhrystone, etc.

Notas:

If the computer systems to be compared are to be used for a particular application, such as banking or airline reservations, a representative subset of functions for that application may be used. Such benchmarks are generally described in terms of functions to be performed and make use of almost all resources in the system, including processors, I/O devices, networks, and databases.

In trade presses, the term *benchmark* is almost always used synonymously with workload. Kernels, synthetic programs, and application-level workloads, for example, are all called benchmarks. Although the instruction mixes are a type of workload, they have not been called benchmarks. Some authors have attempted to restrict the term benchmark to refer only to the set of programs taken from real workloads. This distinction, however, has mostly been ignored in the literature. Thus, the process of performance comparison for two or more systems by measurements is called **benchmarking**, and the workloads used in the measurements are called **benchmarks**.

Selección de servicios que se prueban en el workload.

- Un sistema se componen de múltiple elementos con los que ofrece múltiples servicios, el workload debe concernir a la evaluación de los servicios y no de los elementos del sistema.
- Sin embargo en función del objetivo del estudio se deben realizar las pruebas de forma que muestren mas eficientemente las características de comportamiento del sistema que se están analizando.

Ejemplo: *En un sistema de aplicaciones bancarias, se presentan una jerarquía de componentes que son parte de él:*

- Si se evalúan los servicios de la ALU el workload apropiado debe establecer la proporción de operaciones aritméticas.

- Si se evalúan las CPU el workload debe especificar operaciones de núcleo que exprese instrucciones y sus asociaciones.

- Si se evalúan los S.O. la base del workload debe enfocarse hacia la descrición de servicios del S.O. que se usan y sus relaciones.

- Si se evalúan los servicios de procesamiento de transacciones el workload debe describir la frecuencias de los diferentes tipos de transacciones.

Notas:

The best way to start the workload selection is to view the system as a service provider. The metrics chosen should reflect the performance of services provided at the system level and not at the component level. For example, the MIPS is a justifiable metric for comparing two CPUs, but it is not appropriate for comparing two timesharing systems. The CPU is only one component of the timesharing system. A timesharing system may provide services such as transaction processing, in which case the performance would be measured by transactions (as opposed to instructions) per second.

The basis for workload selection is also the system and not the component. For example, the services provided by the CPUs are the so-called instructions, and the CPU designers may want to use instruction frequency as a possible representation of workload. The services provided by the turn-key banking systems are generally called "transactions," and so the bank may use the transaction frequencies as the workload.

Notice that using instruction frequencies to specify the workload of a banking system is not appropriate, since the performance of the banking system depends on several components in addition to that of the CPU. Similarly, using transactions to compare two CPUs may or may not be appropriate, since the performance may be affected by other components such as I/O devices. However, if a manufacturer offers two banking systems that are identical except for the CPUs, the two systems can be compared using transaction frequencies as the workload. This latter study may sometimes be inaccurately referred to as the comparison of the two CPUs.

Selección de los servicios en los que se evalúa la performance

- El sistema cuya performance se evalúa debe concebirse como un proveedor de servicios. Se diferencian dos elementos:
  - **Sistema bajo prueba (SUT)**: Representa el sistema completo sobre el que se realizan las evaluaciones.
  - **Componente bajo estudio (CUS)**: Representa un servicio concreto del SUT cuyas alternativas se están analizando.

  Ejemplo: *Si se está diseñando una CPU, y dentro de ella se estudia diferentes alternativas de ALUs, la CPUes el SUT y la ALU es el CUS.*

  Ejemplo: *Si dentro de un sistema de gestión bancaria se está estudiando las diferentes opciones de discos para el almacenamiento persistente de la información, en este caso el sistema de procesamientos de transacciones es el SUT y los sistemas de discos son el CUS.*
- La confusión entre sistema (SUT) y componente (CUS) es frecuente y conduce habitualmente a resultados erróneos.
- La base de selección del workload es el sistema, sin embargo la métrica debe ser establecida para que refleje la naturaleza del servicio que se evalúa.

Notas:

Often the term **System Under Test (SUT)** is used to denote the complete set of components that are being purchased or being designed by the organization. Sometimes there is one specific component in the SUT whose alternatives are being considered. This component is called **Component Under Study (CUS).**

Often the term **System Under Test (SUT)** is used to denote the complete set of components that are being purchased or being designed by the organization. Sometimes there is one specific component in the SUT whose alternatives are being considered. This component is called **Component Under Study (CUS).**

A single request at ahigher level may result in one or more requests at the lower level. As shown in the figure, the interface levels are:

1. Arithmetic-logic unit **2.** Central processing unit **3.** Operating system **4.** Applications

The workload could be described by summarizing the requests at any one of these interface levels, dependingupon what constitutes the SUT. If two ALUs are being compared, that is, the ALUs are the systems, the arithmetic instructions constitute the services or requests. The appropriate workload in this case is to specify the frequency of various arithmetic instructions or to specify the most frequent arithmetic instruction, which may very well be the addition instruction.

If two CPUs are being compared, the instruction set of the processors constitutes the service provided. One possibility in this case is to use the instruction mix. However, if the performance of one instruction depends upon that of other neighboring instructions, the workload should be specified in terms of a set of instructions that are commonly used together.

If two systems are being compared at the operating system level, the services provided by the operating systems, including the operating system commands and system services, should form the basis of the workload.

If two turn-key transaction processing systems are being compared, the application interface is the SUT level and the requests at this level, namely, the transactions, would form the basis of workload. The workload could be described by specifying the frequency of various types of transactions or the most frequent transaction.

Ejemplo: Sistema de backup (1)

- Un sistema de backup consiste en un sistema compuesto de varios sistemas de datos en disco. Cada sistema en disco se compone de varios discos físicos, con diferentes drivers, con subsistemas de lectura y escritura independientes. Este sistema se puede jerarquizar en los siguientes niveles:
  - **Sistema Backup:**
    - Servicios: Ficheros de backup, operaciones de backup, restauración de ficheros, etc.
    - Parámetros: Tamaño del sistema de ficheros, tipos de backup (incremental o completo)
    - Métricas: Tiempo de backup, tiempo de restauración.
    - Workload: Descripción de un sistema para realizar su backup o para ser restaurado.
  - **Sistema de datos**:
    - Servicios: Lectura/escritura de ficheros, lectura de etiquetas
    - Parámetros: Tipos de disco físico
    - Métricas: Velocidad, fiabilidad , tiempo entre fallos.
    - Workload: Un programa sintético de operaciones de entrada/salidas representativo de los procesos de backup.

Notas:

A magnetic tape backup system consists of several tape data systems, each containing several tape drives. The drives have separate read and write subsystems. Each subsystem makes use of magnetic heads. Thus, starting from a high level and moving down to lower levels, the services, factors, metrics, and workloads are as follows:

1. *Backup System:*

(a) Services: Backup files, backup changed files, restore files, list backed-up files.

(b) Factors: File system size, batch or background process, incremental or full backups.

(c) Metrics: Backup time, restore time.

(d) Workload: A computer system with files to be backed up. Vary frequency of backups.

2. *Tape Data System:*

(a) Services: Read/write to the tape, read tape label, autoload tapes.

(b) Factors: Type of tape drive.

(c) Metrics: Speed, reliability, time between failures.

(d) Workload: A synthetic program generating representative tape I/O requests.

Ejemplo: Sistema de backup (2)

- **Sistema de discos físicos:**
  - Servicios: Lectura de pistas, escritura de pistas, lectura de directorio.
  - Parámetros: Número de caras, Tamaño del disco, formato de la información.
  - Métricas: Tiempo de respuesta de cada tipo de servicio, tasa de fallos, disipación de potencia.
  - Workload: Programa sintético que requiera los diferentes servicios con las proporciones adecuadas.
- **Subsistema de lectura/escritura**:
  - Servicios: Lectura de datos, escritura de datos (como señales digitales)
  - Parámetros: Técnica de codificación de datos, Tecnología de implementación
  - Métricas: densidad de datos, anchura de banda de escritura y lectura (bits/s),
  - Workload: Plan de prueba de lectura/escritura de datos con diferentes amplitudes, plan de prueba de escritura con diferentes velocidades de giro del disco..

Notas:

3. *Tape Drives:*

(a) Services: Read record, write record, rewind, find record, move to end of tape, move to beginning of tape.

(b) Factors: Cartridge or reel tapes, drive size.

(c) Metrics: Time for each type of service, for example, time to read record and to write record, speed (requests per unit time), noise, power dissipation.

(d) Workload: A synthetic program exerciser generating various types of requests in a representative manner.

4. *Read/Write Subsystem:*

(a) Services: Read data, write data (as digital signals).

(b) Factors: Data-encoding technique, implementation technology (CMOS, TTL, and so forth).

(c) Metrics: Coding density, I/O bandwidth (bits per second).

(d) Workload: Read/write data streams with varying patterns of bits.

Nivel de detalle del workload

De menor a mayor nivel de detalle se identifican las siguiente opciones:

- Requerimiento mas frecuente: Se considera como workload la frecuencia del requerimiento que se demanda con mayor frecuencia. Se utiliza en los análisis tempranos en los que no hay información disponible sobre el sistema. Es representativo sólo si el servicio es requerido con mucho mayor frecuencia que los restantes, o si el requerimientos es el principal consumidos de recursos.
- Frecuencias de cada tipo de requerimiento: El workload consiste en una lista con las características y frecuencia de cada uno de los requerimientos de los servicios del sistema. Un ejemplo es el conjunto ponderado de instrucciones. No es representativo cuando son servicios sensibles a la asociación histórica como los fenómenos de caché.
- Traza detallada de los requerimientos: El workload consiste en una lista de requerimientos datados con el tiempo en que se han producido. Pueden corresponde a cantidades masivas de datos. No son útiles en los estudios basados en modelos.
- Promedio de las demanda de recursos: El workload se describe en función de las frecuencia de requerimiento de los recursos, y no de los requerimientos de servicios. Son principalmente utilizados en análisis basados en modelos.
- Distribución de las demandas de recursos: El workload se describe a través de la lista de las funciones de distribución de los requerimientos de recursos. Es especialmente necesario si los requerimientos tienen una gran variabilidad.

Notas:

The least detailed alternative is to select the most frequently requested service and use it as the workload. While this may not provide enough information about the system, this is commonly used as the initial workload. The addition instruction to compare early ALUs, various kernels to compare processors, and the debit-credit benchmark to compare transaction processing systems are examples of this approach. It is particularly valid if one type of service is requested much more often than others or is a major consumer of resources in the system.

The second alternative is to list various services, their characteristics, and frequency. The instruction mixes are examples of this approach. If the performance of a service depends upon the context, that is, on the services required in the past, the set of services that are expected to be context free are more appropriate than the individual services. History-sensitive mechanisms, such as caching in computer systems, often make this grouping necessary.

The third alternative is to get a time-stamped record (called a **trace**) of requests on a real system and use it as a workload. The problem with this alternative is that it may be too detailed. It is certainly inconvenient for analytical modeling. Also, for simulation it may require exact reproduction of component behavior to maintain the timing relationships.

In analytical modeling, the resource demands placed by the requests, rather than the requests themselves, are used as the workload. For example, the statement that each user has an average CPU time of 50 milliseconds and makes 20 I/O's per request may be a sufficient workload description for analytical modeling. In case of multiple services, several similar services can be grouped into a class, and each class may be characterized by its average resource demands.

The average demand may not be sufficient in some cases, and it may be necessary to specify the complete probability distribution for resource demands. This is particularly the case if there is a large variance in resource demands or if the distribution is different than that used in the model. Particularly, in simulations, it is easy to use different distributions. The analytical models are generally restricted to a given distribution.

The workload descriptions used for analytical and simulation modeling are also referred to as **nonexecutable** workloads since they are not in a form suitable for execution on a real system. On the other hand, a trace of user commands that can be executed directly on a system would be called an **executable** workload.

Representatividad y reproducibilidad

- Los workloads de prueba tienen que ser representativos del uso que el entorno real realiza del sistema que se analiza. La representatividad puede hacer referencia a uno de los siguientes niveles:
  - Frecuencia de requerimientos: Las frecuencias con las que se realizan los requerimientos de los servicios en el workload debe coincidir con el que ocurre en el sistema bajo estudio.
  - Demanda de los recursos: La demanda total de los recursos en el workload debe coincidir con los que ocurren en el sistema real.
  - Perfil de uso de los recursos: El perfil de uso de los recursos debe coincidir con el real. Él workload en un entorno de multiprogramación concurrente, no sólo debe ser representativo del uso de los recursos en cada aplicación de forma independiente, sino que también debe serlo cuando concurren las diferentes aplicaciones.
- Los workload deben tener las características necesarias para que los resultados de un análisis puedan ser reproducibles. Workload que contienen altas aleatoriedad en los requerimientos de los servicios y en la demanda de los recursos, suelen presentar problemas porque generan una alta variabilidad de los resultados.

Notas:

A test workload should be representative of the real application. One definition of representativeness is that the test workload and the real application match in the following three respects:

1. *Arrival Rate:* The arrival rate of requests should be the same or proportional to that of the application.

2. *Resource Demands:* The total demands on each of the key resources should be the same or proportional to that of the application.

3. *Resource Usage Profile:* Resource usage profile relates to the sequence and the amounts in which different resources are used in an application. In a multiprogramming environment, it is important that the test workloads have a resource usage profile similar to that of the applications. Otherwise, it is possible that total resource demands of individual workloads may be representative of their respective

applications, but when several workloads are run together, they may not produce results representative of combined applications.

Nivel de carga

- Un workload puede llevar al sistema hasta la utilización plena de su capacidad (mejor caso) o llevarlo por encima de su capacidad (peor caso) o al nivel observado en el sistema real (caso típico).
- La correcta elección del nivel de carga debe ser estudiado en cada caso y en función de los objetivos del análisis:
  - Para analizar la efectividad de una política para gestionar las situaciones de congestión del sistema hay que utilizar workload de peor caso que lleven al sistema mas allá de su capacidad.
  - Para estudiar el efecto de las diferentes políticas de planificación de transferencia de paquetes por la red de comunicaciones del sistema hay que utilizar workload que corresponda a uso típico de la red, así como a workload que representan tráfico mas densos, si estos también se pueden presentar.

Notas:

Loading Level: A workload may exercise a system to its full capacity (best case), beyond its capacity (worst case), or at the load level observed in real workload (typical case). For procurement purposes, a workload measured in a similar existing environment may be good enough. However, for computer system design, you may have to identify all the environments where the system might be used and study performance under best, worst, and typical workloads. The correct choice of the loading level varies from case to case. For example, to measure the effectiveness of a congestion control scheme, the network should be exercised beyond its capacity, while the packet etransmission schemes should be tested for normal as well as heavy load, since the retransmissions may be required under both circumstances.

Caracterización del workload

- Cuando se analiza o compara el comportamiento de varios sistemas, se necesita utilizar un workload de prueba que:
  - Sea reproducible.
  - Refleje las características esenciales del workload real.
  - Pueda ser modificado de forma controlada
- Se denomina caracterización del workload al proceso de modelado y análisis del workload real de un sistema a fin de generar un workload de prueba que refleje sus características esenciales
- Para caracterizar un workload se necesita identificar y caracterizar los clientes que realizan requerimientos al sistema. A estos clientes se suelen denominar componentes del workload.
- Los componentes del workload se pueden organizar con diferentes criterios:
  - Aplicaciones: Las aplicaciones que hacen uso concurrente del sistema.
  - Lugares: La localizaciones desde las que se hacen requerimientos al sistema.
  - Cliente de sesiones: Sesiones completas desde (el login hasta el logout).

Notas:

In order to test multiple alternatives under identical conditions, the workload should be repeatable. Since a real-user environment is generally not repeatable, it is necessary to study the real-user environments, observe the key characteristics, and develop a workload model that can be used repeatedly. This process is called **workload characterization**. Once a workload model is available, the effect of changes in the workload and system can be studied in a controlled manner by simply changing the parameters of the model.

In workload characterization literature, the term **workload component** or **workload unit** is used instead of the user. The workload characterization consists of characterizing a typical user or workload component. Other examples of workload components are as follows:

• *Applications*: If one wants to characterize the behavior of various applications, such as mail, text editing, or program development, then each application may be considered a workload component and the average behavior of each application may be characterized.

• *Sites*: If one desires to characterize the workload at each of several locations of an organization, the sites may be used as workload components.

• *User Sessions*: Complete user sessions from login to logout may be monitored, and applications run during the session may be combined.

Parámetros del workload

- Un aspecto clave para simplificar el modelo del workload es poder agrupar los clientes en grupos con características homogéneas.
- Incluir en un mismo grupo clientes con características muy diversas conduce a incrementar la variabilidad del workload y a reducir la reproducibilidad.
- Con este objetivo se deben elegir adecuadamente los parámetros con los que se caracterizan los componentes del workload:
  - Se deben elegir parámetros que describan características propias de los clientes, y no dependan del modelo del sistema.

    Ejemplo: *Es preferible utilizar frecuencias de requerimientos de servicios que tanto por ciento de uso de recursos del sistema.*
  - Deben escogerse las características que afecten a la demanda de recursos del sistema.

    Ejemplo: *Si el tamaño de los paquetes no tiene efectos sobre la latencia de un router, es preferible utilizar otro parámetro para caracterizar el workload como es la frecuencia de paquetes que si afecta el comportamiento del router.*

Notas:

The key requirement for the selection of the workload component is that it be at the SUT interface. Another consideration is that each component should represent as homogeneous a group as possible.

The measured quantities, service requests, or resource demands, which are used to model or characterize the workload, are called **workload parameters** or **workload features**. Examples of workload parameters are transaction types, instructions, packet sizes, source destinations of a packet, and page reference pattern.

In choosing the parameters to characterize the workload, it is preferable to use those parameters that depend on the workload rather than on the system. For example, the elasped time (response time) for a transaction is not appropriate as a workload parameter, since it depends highly on the system on which the transaction is executed. This is one reason why the number of service requests rather than the amount of resource demanded is preferable as a workload parameter.

Técnicas para caracterizar los workload

- Caracterización estadística: Promediado y especificación de la dispersión.
- Histogramas de un único parámetro.
- Diagramas de dispersión. Histograma de múltiples parámetros.
- Análisis de componentes principales.
- Modelos de Markov.
- Agrupamientos (clustering)

Notas:

There are several characteristics of service requests (or resource demands) that are of interest. For example, arrival time, type of request or the resource demanded, duration of the request, and quantity of the resource demanded by each request may be represented in the workload model. Particularly those charcteristics that have a significant impact on the performance should be included in the workload parameters, and those that have little impact should be excluded. For example, if the packet size has no impact on packet forwarding time at a router, it may be omitted from the list of workload parameters, and only the number of packets and arrival times of packets may be used instead.

The following techniques have been used in the past for workload characterization:

1. Averaging

2. Specifying dispersion

3. Single-parameter histograms

4. Multiparameter histograms

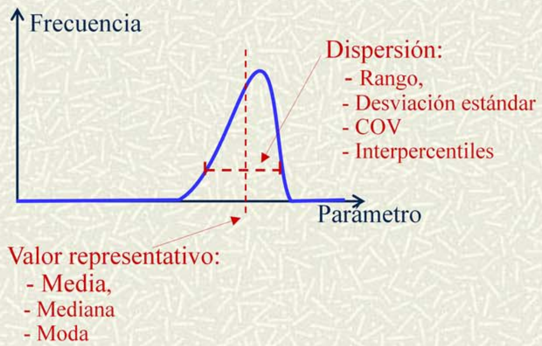
5. Principal-component analysis

6. Markov models

7. Clustering

Promediado y dispersión

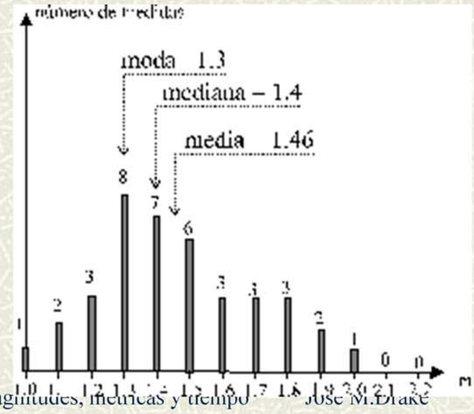
- La forma mas simple de describir un parámetro de workload que tenga una cierta variabilidad es representando lo por dos magnitudes estadísticas que describan su:
  - Valor mas representativo: Valor medio, mediana o moda
  - Valor que describe su dispersión: Rango, desviación estándar, COV, interpercentiles del 10% al 90%



Notas:

Magnitudes estadísticas

- Cuando se mide múltiples veces una misma magnitud m, se obtienen un conjunto valores con una cierta **distribución estadística** de valores (continua o discreta). En estos casos, se necesita caracterizar estadísticamente el resultado de la medida con dos objetivos:
  - Estimar cual es el valor medido **mas representativo** de la medida.
  - Caracterizar la **dispersión de valores** que han resultados, y con ello informar sobre la **falta de seguridad** o **incertidumbre** que existe.



Notas:

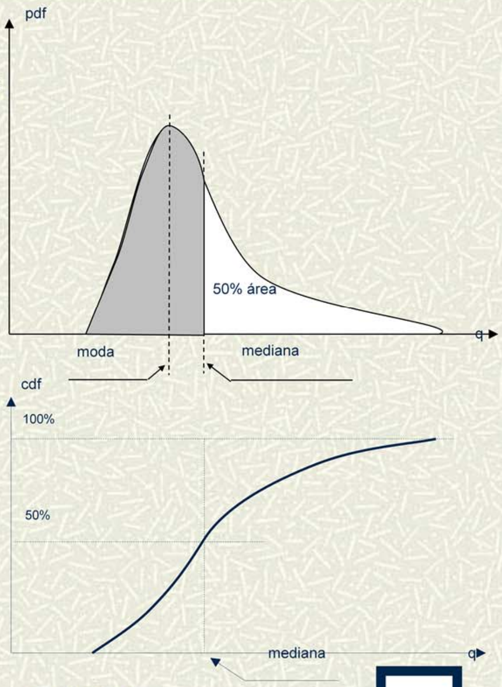
Valor mas representativo

- **Valor medio** (*mean*) de N medidas q_1 (n_1 ocurrencias), q_2 (n_2 ocurrencias),.. ,q_k (n_k ocurrencias). Se calcula como,

$$\overline{q} = \frac{1}{N}\sum_{1<i<k} n_i q_i \qquad \overline{q} = \int_{-\infty}^{\infty} q\ p(q)\ dq$$

$$siendo\ p(q)\ la\ densida\ despectral$$

- **Mediana** (*median*): de un conjunto de valores es el valor de la muestra que tiene tantos valores del conjunto por encima como por debajo de ella.
- **Moda** (*mode*): es el valor de la muestra que se presenta con mayor frecuencia (o tiene una densidad espectral máxima), lo que representa el valor más común en la medida.

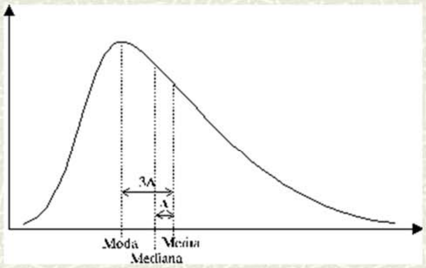


Notas:

In the most condensed form, a single number may be presented that gives the key characteristic of the data set. This single number is usually called an **average** of the data. To be meaningful, this average should be representative of a major part of the data set. Three popular alternatives to summarize a sample are to specify its mean, median, or mode. These measures are what statisticians call **indices of central tendencies**. The name is based on the fact that these measures specify the center of location of the distribution of the observations in the sample.

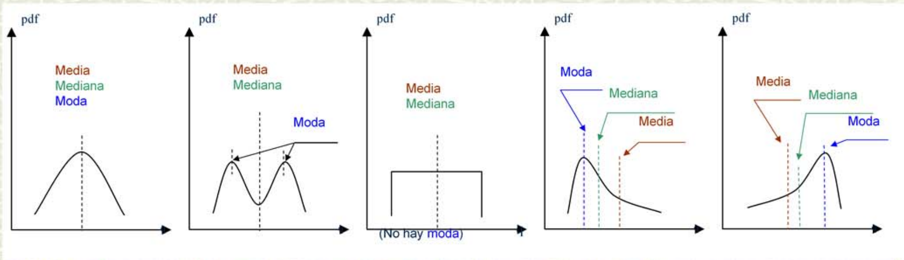
Relación entre los valores mas representativos

- Para curvas de frecuencia unimodales y simétricas la media, la mediana y la moda coinciden. En curvas unimodales asimétricas, se suele verificar la siguiente relación empírica:

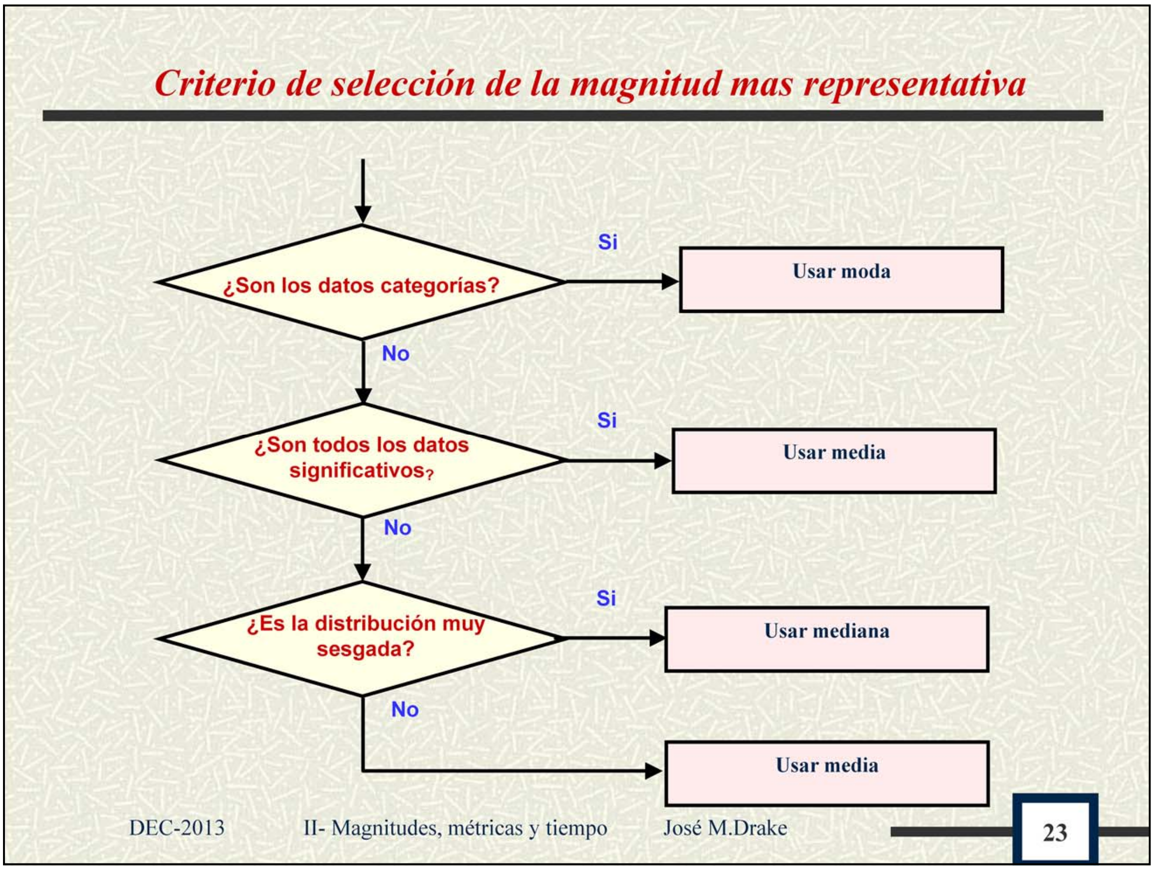


|Moda-Media| ≈

≈ 3 |Media-Mediana|



Notas:


Criterio de selección de la magnitud mas representativa
¿Son los datos categorías?
Si
Usar moda
No
¿Son todos los datos significativos?
Si
Usar media
No
¿Es la distribución muy sesgada?
Si
Usar mediana
No
Usar media
DEC-2013
II- Magnitudes, métricas y tiempo
José M.Drake
23

Notas:

Magnitudes que miden la dispersión (I)

- **Rango** (*range*): de un conjunto de datos es la diferencia entre el mayor y el menor de ellos
- **Desviación media** es la media de las desviaciones de las diferentes medidas respecto del valor medio.

$$\bar{D} = \frac{1}{n}\Sigma|d_i| = \frac{\Sigma\,|\,m_i - \bar{m}\,|}{n}$$

- **Desviación estándar experimental** (*Standard desviation*) Corresponde a una desviación cuadrática media de un conjunto de N muestras.

$$S = \sqrt{\frac{\Sigma(q_i - \bar{q})^2}{N-1}} = \sqrt{\sum p_i q_i} = \sqrt{\int_{-\infty}^{\infty} q\, f(q)\, dq} \quad \textit{siendo } f(q) \textit{la función densidad de probabilidad}$$

- **Coeficiente de variación** (*COV*) representa la dispersión relativa de los resultados de una medida obtenida a partir de una serie de mediciones de una misma magnitud. Esto es,

$$V = \frac{S}{\bar{q}}$$

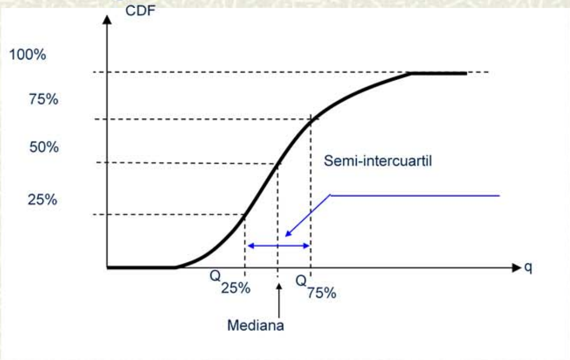
Notas:

Magnitudes que miden la dispersión (II)

- **Semi-intercuartil** (*Semi-interquartile*): es la mitad de la diferencia entre los semicuartiles $Q_{75\%}$ y $Q_{25\%}$

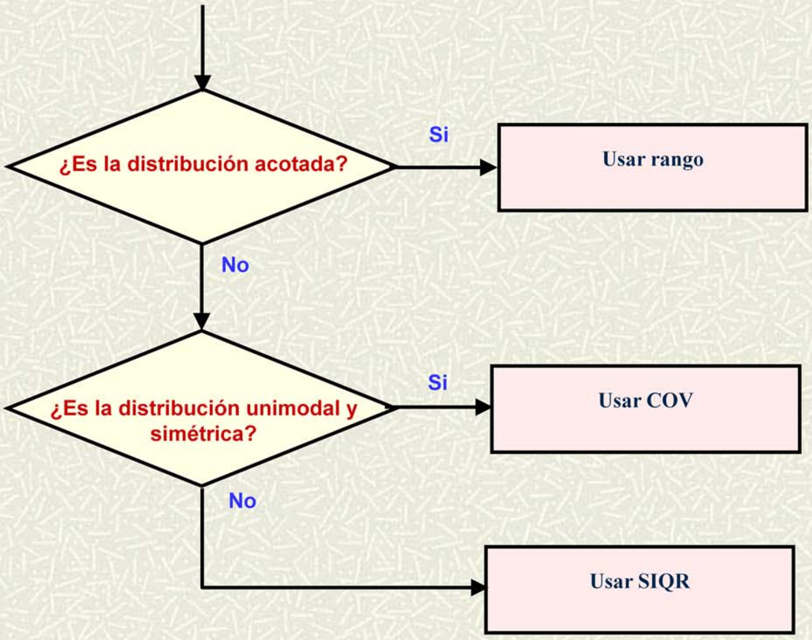
$$SIQR = \frac{Q_{75\%} - Q_{25\%}}{2}$$

Se denomina x-cuartil el valor de la magnitud para el que la función de distribución integrada toma el valor x.



Notas:

Criterio de selección de la medida de dispersión



Notas:

Ejemplos de caracterización estadística

- Workload organizada por lugares

Caracterización Universidad de Cantabria

Parámetro	Prom	COV
CPU time	2.19 s	40.23%
Number of outputs	820	53.59%
Number of writes	10210	82.14%
Number of inputs	226	25.65%
Number of reads	4970	21.01%

Alta dispersión => *Los componentes no deben ser caracterizados conjunta-mente*

- Workload organizadas por aplicaciones

Caracterización Aplicación Editor

Parámetro	Prom	COV
CPU time	2.57 s	3.54%
Number of outputs	1974	4.33%
Number of writes	13460	3.87%
Number of inputs	377	3.73%
Number of reads	3693	3.16%

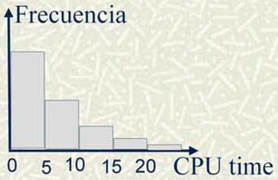
Baja dispersión => *Los componentes son buenos candidatos a ser caracte-rizados conjuntamente*

Notas:

The resource demands of various programs executed on six university sites were measured for 6 months. The average demand by each program is shown in Table 6.1. Notice that the C.O.V. of the measured values are rather high, indicating that combining all programs into one class is not a good idea. Programs should be divided into several classes. The table shows the average demand for all editors in the same data. The C.O.V. are now much lower.

Histograma de parámetros independientes

- Los histogramas de un parámetros describen las frecuencia de los valores organizados en secciones. Describen de forma discreta la distribución de los valores de los parámetros.



CPU time

Lugar	0-5	5-10	10-15	15-20	20-
Universidad Cantabria	980	412	189	60	24

- Los histogramas simples son útiles para caracterizar las distribuciones estadísticas de los parámetros en los modelos analíticos.
- Problemas:
  - No describen las correlaciones entre parámetros
  - No describen la dependencia en el tiempo entre los valores.
  - Dan lugar a un exceso de datos (Nº secciones x Nº parámetros x Nº componentes)

Notas:

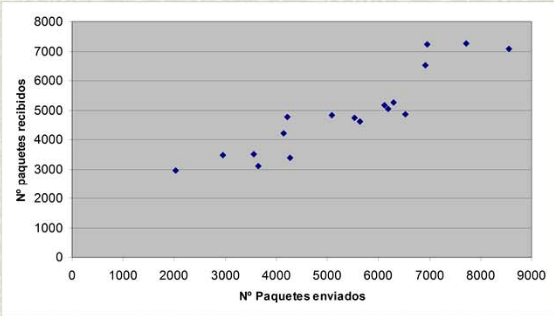
A histogram shows the relative frequencies of various values of a parameter. For continuous-value parameters, this requires dividing the complete parameter range into several smaller subranges called *buckets* (or *cells*) and counting the observations that fall in each cell. An example is shown in the figure for CPU time. The results can also be presented in tabular form, as shown in the table.

Given *n* buckets per histogram, *m* parameters per component, and *k* components, this method requires presenting *nmk* numerical values. This may be too much detail to be useful. Thus, this should be used only if the variance is high and the averages cannot be used.

The key problem with using individual-parameter histograms is that they ignore the correlation among different parameters.

Histograma de parámetros múltiples (diagramas de dispersión)

- Representan en una misma gráfica n parámetros del workload Muestra la correlación que existen entre los diferentes parámetros.
- En la figura se muestra de dispersión que relaciona los parámetros Nº de paquetes enviados y Nº de paquetes recibidos por diferentes equipos en una determinada ventana de tiempos:



- De la inspección del diagrama se observa la correlación entre ambos parámetros; "Los equipos que envían mas paquetes, también reciben mas paquetes"
- Cuando el número de parámetros es mayor que 2, la visualización es difícil y se requiere técnicas matriciales y herramientas de análisis de las tendencias.
- Son caracterizaciones basadas en demasiados número de valores de parámetros.

Notas:

If there is a significant correlation between different workload parameters, the workload should be characterized using multiparameter histograms. An *n*-dimensional matrix (or histogram) is used to describe the distribution of *n* workload parameters.

The figure shows an example of a simplistic plot of a two-parameter joint histogram. The number of frames sent and received by stations on a local-area network are plotted. Each dot in the figure represents a station. The number of dots in a square represents the number of stations that sent and received the frames in the range corresponding to the cell. Generally, the stations sending a large number of frames are also the ones that receive a large number of frames. Thus, there is a significant correlation between the two parameters.

It is difficult to plot joint histograms for more than two parameters. Also, as discussed before, even single-parameter histrograms are too detailed in some cases. Multiparameter histograms are even more detailed; they are therefore rarely used.

Análisis de las componentes principales

- Trata de reducir el número de parámetros que se requieren para describir la variabilidad de un workload. Para ello, se definen nuevos parámetros que describen la variabilidad de forma mas:
  - Sencilla: se requieren menos parámetros para describir la variabilidad.
  - Significativa: reflejan de forma mas directa la demanda de los recursos que requiere el workload.
- Los nuevos parámetros se formulan en función de los parámetros primitivos medidos sobre el workload real:

$$y_i = f_i(x_1, x_2, .., x_n) \xrightarrow[\text{función lineal}]{} y_i = \sum_{j=1}^{n} a_{ij} x_j$$

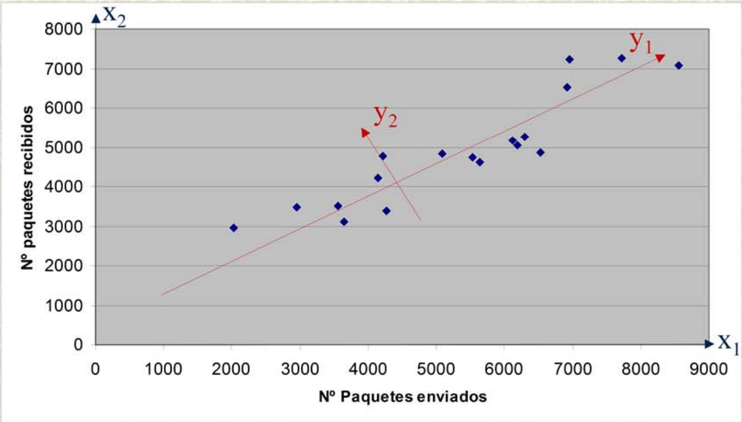
- Esta técnica requiere métodos sistemáticos para definir las funciones que definen los nuevos parámetros.

Notas:

One technique commonly used to classify workload components is by the weighted sum of values. Using *aj* as weight for the *j*th parameter *xj*, the weighted sum *y* is $y=\sum_j a_j x_j$ This sum can then be used to classify the components into a number of classes such as low medium demand. Although this technique is commonly used in performance analysis software, the person running the software is asked to choose the weight. Without any concrete guidelines, may assign weights such that workload components with very different characteristics may together, and the mean characteristics of the group may not correspond to any member.

Análisis de parámetros principales

- El análisis de parámetro principales es un método matemático sistemático que proporciona nuevos parámetros no correlacionados que representen mejor la variabilidad de los datos del workload.



Notas:

Método de evaluación de los parámetros principales

- A partir de un conjunto de casos descritos por sus parámetros medidos $\{x_1,x_2,..,x_n\}$, se trata de identificar los casos mediante otro conjunto de parámetros $\{y_1,y_2,..,y_n\}$, con las siguientes características:
  - Los nuevos parámetros están relacionados con los primitivos mediante relaciones lineales

$$y_i = \sum_{j=1}^{n} a_{ij} x_j$$

  - El conjunto de los parámetros y forman un conjunto ortogonal (equivale a decir que las y_i no están correlacionadas entre sí).

$$\langle y_i, y_j \rangle = \sum_k a_{ik} a_{kj} = 0$$

  - La y_i resultan ordenadas en cuanto al nivel de dependencia de los casos respectos de ellas. Esto es, los diferentes casos representan una variabilidad respecto de y_1, en menor medida respecto de y_2, aún en menor medida respecto de y_3, etc. De acuerdo con el nivel de detalle, podremos considerar sólo los primeros parámetros y_i

Notas:

One method of determining the weights in such situations is to use the principal-component analysis, which allows finding the weights *wi*'s such that *yj*'s provide the maximum discrimination among the components.

The quantity *yj* is called the **principal factor(*)**. Statistically, given a set of *n* parameters {*x*1,*x*2,...,*xn*}, the principal-component analysis produces a set of **factors** {*y*1,*y*2,...,*yn*} such that the following holds:

1. The *y*'s are linear combinations of *x*'s.
2. The *y*'s form an orthogonal set, that is, their inner product is zero.
3. The *y*'s form an ordered set such that *y*1 explains the highest percentage of the variance in resource demands, *y*2 explains a lower percentage, *y*3 explains a still lower percentage, and so forth. Thus, depending upon the level of detail required, only the first few factors can be used to classify the workload components.

(*) The correct term is principal component. However, to avoid confusion with workload components, the term

principal factor, which has a slightly different meaning in factor analysis, is used here.

Ejemplo de calculo de las componentes principales

Caso	x_1	x_2	x'_1	x'_2	y_1	y_2
1	7718	7258	1,359	1,717	2,175	-0,253
2	6958	7232	0,922	1,698	1,853	-0,549
3	8551	7062	1,837	1,575	2,413	0,186
4	6924	6526	0,903	1,186	1,477	-0,200
5	6298	5251	0,543	0,262	0,570	0,199
6	6120	5158	0,441	0,195	0,450	0,174
7	6184	5051	0,478	0,117	0,421	0,255
8	6527	4850	0,675	-0,029	0,457	0,497
			-0,156	-0,047		
14	3562	3497			-1,441	-0,013
15	2955	3480	-1,377	-1,022	-1,696	-0,251
16	4261	3392	-0,627	-1,085	-1,211	0,324
17	3644	3120	-0,981	-1,283	-1,601	0,213
18	2020	2946	-1,914	-1,409	-2,349	-0,357
Media	5352,0	4889,4	0,0000	0,0000	0,000	0,000
DesvStd	1741,0	1379,5	1,0000	1,0000	1,384	0,290

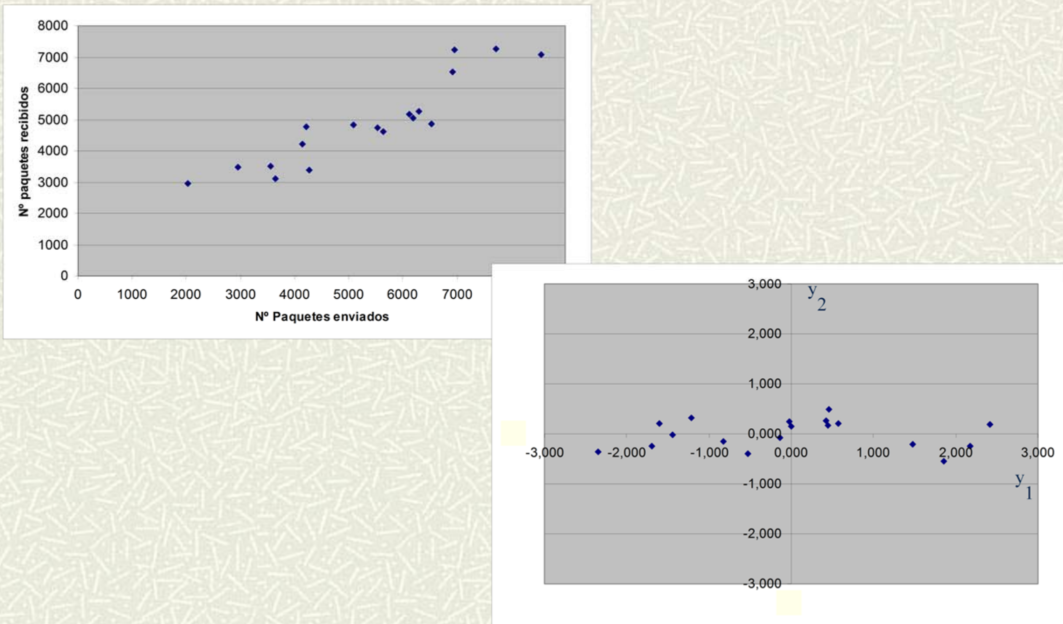
1) Se calculan la media y la desviación típica.
2) Se normalizan los parámetros
3) Se calculan los coeficientes de correlación entre x_1 y x_2
4) Matriz de correlación
5) Se determinan los autovalores de la matriz de correlación.
6) Se calculan los autovalores de la matriz de correlación
7) Se evalúan las ecuaciones de transformación
8) Se calculan los valores de los casos en y_1 e y_2
9) Se determinan los valores medios y desviación estándar en los parámetros y_1 e y_2
10) Se dibujan los datos en el nuevo sistema de coordenadas.

Notas:

The number of packets sent and received, denoted by *xs* and *xr*, respectively, by various stations on a local-area network were measured. The observed numbers are shown in the second and third columns of the table. A scatter plot of the data is shown in Figure (pgna 24). As seen from this figure, there is considerable correlation between the two variables. The steps in determining the principal factors are as follows:

El calculo de los factores principales de este ejemplo, puede verse completo en las Pgnas 78-80 del libro de Raj Jain.

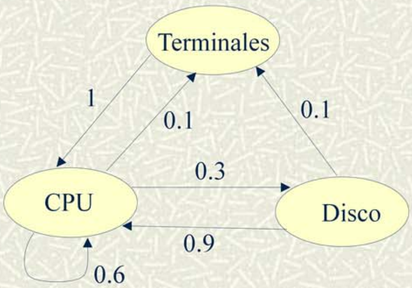
Resultado de la representación con parámetros principales



Notas:

Modelo de Markov

- Se utiliza cuando es relevante describir las relaciones históricas entre los sucesivos requerimientos que se realizan en el workload.
- El modelo de Markov es una simplificación en la que sólo se considera que la probabilidad de transición depende del estado actual, y no de los anteriores.



Matriz de probabilidades de transición

From/To	CPU	Disk	Terminal
CPU	0.6	0.3	0.1
Disk	0.9	0	0.1
Terminal	1	0	0

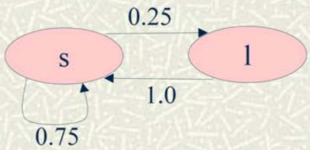
Notas:

Sometimes, it is important to have not only the number of service requests of each type but also their order. The next request is generally determined by the last few requests. If it is assumed that the next request depends only on the last request, then the requests follow a **Markov model**. Actually, the term is used in a more general sense of system states rather than for user requests. That is, if the next system state depends only on the current system state, the system follows a Markov model. The models can be described by a *transition matrix*, which gives the probabilities of the next state given the current state. For example, the transition probability matrix for a job's transition between the CPU, disk, and terminal is shown in the table. The corresponding state transition diagram is shown in the figure. After each visit to the CPU, the probability that the job will move to the disk is 0.3, the probability of it going to the terminal is 0.1, and so on.

Transition matrices are used not only for resource transitions but also for application transitions. For example, if the users in a software development environment run editors, compilers, linkers, and applications, a transition probability matrix can be used to characterize the probability of a user running software of type *j* after running the software of type *i*.

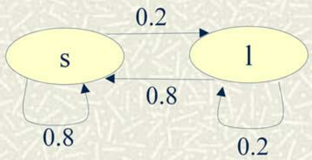
Ejemplo de modelado de tráfico de una red

- Considerar una red por la que se transmiten dos tipos de paquetes de tamaño muy diferentes (Largos (l) y cortos (s)). Entre ellos existen ciertas cadencias de ocurrencia:
  - Caso A; En promedio, a cada cuatro paquetes pequeños le sigue uno grande (sssslsssslssssl...)



	Post. s	Post. l
Previo s	0.75	0.25
Previo l	1	0

  - Caso B: El 80% de los paquetes son pequeños y el 20% de los paquetes grandes



	Post. s	Post. l
Previo s	0.8	0.2
Previo l	0.8	0.2

Notas:

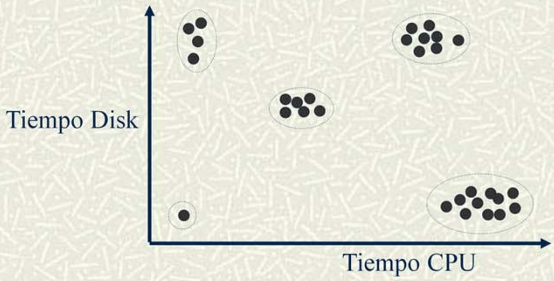
Traffic monitoring on a computer network showed that most of the packets were of two sizes—small and large. The small packets constituted 80% of the traffic. A number of different transition probability matrices will result in an overall average of 80% of small packets. Two of the possibilities are as follows:

1. An average of four small packets are followed by an average of one big packet. A sample sequence, using s for small and b for big, is ssssbssssbssss. In this sequence, three of the four small packets are followed by another small packet. Also, every big packet is followed by a small packet. The corresponding transition probability matrix is the upper table..

2. Another alternative is to generate a random number between 0 and 1. If the number is less than or equal to 0.8, generate a small packet; otherwise, generate a large packet. This assumes that the next packet size does not depend upon the current packet size. The transition probability matrix in this case is the lower table.

Agrupamiento (clustering)

- Habitualmente un workload está compuesto por un gran número de componentes, y su análisis se simplifica si se reducen el número de ellos.
- En este caso hay que reducir el número en base a identificar con un sólo modelo, todos aquellos que tienen un comportamiento semejante.
- Ejemplo: en la gráfica se muestra el caso de un computador (CPU + Disco) que son demandado por 30 aplicaciones. Los parámetros que se utilizan para caracterizar las aplicaciones son Tiempo de CPU y Tiempo de disco que requieren.
- Cuando los parámetros son muchos y las regiones de los grupos se confunden hay que tener un método sistemático de clustering



Notas:

Generally, the measured workload consists of a large number of components. For example, several thousand user profiles may have been measured. For analysis purposes, it is useful to classify these components into a small number of classes or clusters such that the components within a cluster are very similar to each other. Later, one member from each cluster may be selected to represent the class and to study the effect of system design decisions on the entire class.

The figure shows the CPU and disk I/O demands of 30 jobs. As shown, the jobs can be classified into five clusters. Thus, instead of using 30 jobs for each analysis, just 5 jobs can be used to represent the average resource demands of each cluster.

To characterize measured workload data using clustering, the steps are as follows:

1. Take a sample, that is, a subset of workload components.

2. Select workload parameters.

3. Transform parameters, if necessary.

4. Remove outliers.

5. Scale all observations.

6. Select a distance measure.

7. Perform clustering.

8. Interpret results.

9. Change parameters, or number of clusters, and repeat steps 3 to 7.

10. Select representative components from each cluster.

Pasos para ejecutar el proceso de clustering(1)

1. Toma una muestra de los componentes que se quieren agrupar.
   - Su adecuación se evalúa al final
2. Seleccionar los parámetros de workload con los que se caracterizan los componentes.
   - Se deben seleccionar parámetros que sean significativos respecto a la demanda de recursos del sistema.
3. Transformar los parámetros si es necesario
   - Si los valores son muy sesgados, conviene utilizar transformaciones de los parámetros para que la variaciones de cada uno de ellos sea equivalente. Por ejemplo aplicar escalas logarítmicas.
4. Eliminar casos muy extremos
   - Los valores extremos distorsionan los rangos, y es mejor tratarlos como casos singulares.
5. Escalar todas las observaciones.
   - El proceso de clustering va a depender de ponderar los valores de los diferentes parámetros, y para equilibrar su efecto conviene escalarlos para que todos ellos tengan la relevancia que se le quiera dar. Criterios de escalado:
     - Normalizar a media nula y varianza unidad.
     - Asignar pesos.
     - Normalizar los rangos.

Notas:

Pasos para ejecutar el proceso de clustering(2)

6. Seleccionar una distancia de medida.
   - El criterio de agrupamiento va a consistir en reunir aquellos cuya distancia entre si en el espacio paramétrico sea equivalente. A tal fin hay que definir una métrica de distancia en el espacio paramétrico:
     - Distancia euclidea
     - Distancia euclidea con pesos
     - Distancia chi-Cuadrado
7. Realizar el agrupamiento
   - El agrupamiento se puede realizar por dos criterios:
     - Formular el agrupamiento de forma que la varianza interna de los grupos sea mínima.
     - Formular el agrupamiento de forma que las varianza entre las medias de los grupos sea máxima

     Ambas técnicas conducen a resultados similares
8. Interpretar los resultados
   - Conviene analizar los resultados y eliminar aquellos grupos que sean irrelevantes. El criterio de descarte debe hacerse en base a la demanda de recursos que represente el grupo, y no al número de casos que contenga.
9. Si se quiere, cambiar los parámetros del proceso y repetir los pasos 3 a 7.
10. Seleccionar el tipo representativo de cada grupo.

Notas: