Diseño y Evaluación de Configuraciones

II Magnitudes y métricas

J.M. Drake

Notas:

Concepto de métrica

- Una métrica de performance es una magnitud que es útil para describir características no funcionales tales como, comportamiento temporal, confiabilidad o calidad de servicio que ofrece un sistema.
- Cada análisis de performances requiere elegir el conjunto de métricas con el que se realiza.
- Se debe distinguir entre medida y métrica:
  - Una medida proporciona información de un caso que ha ocurrido.
  - Una métrica es generada desde el análisis de muchas medidas, y corresponde a una interpretación de las medidas para ser utilizadas por el analista.
- Una buena métrica debe ser SMART (specific, measurable, attainable, repeatable, and time-dependent)

Notas:

A widely accepted management principle is that an activity cannot be managed if it cannot be measured.

For each performance study, a set of performance criteria or metrics must be chosen. One way to prepare this set is to list the services offered by the system. For each service request made to the system, there are several possible outcomes.

It helps to understand what metrics are by drawing a distinction between metrics and measurements. Measurements provide single-point-in-time views of specific, discrete factors, while metrics are derived by comparing to a predetermined baseline two or more measurements taken over time.

Measurements are generated by counting; metrics are generated from analysis. In other words, measurements are objective raw data and metrics are either objective or subjective human interpretations of those data.

Good metrics are those that are SMART, i.e. specific, measurable, attainable, repeatable, and time-dependent, according to George Jelen of the International Systems Security Engineering Association.

Truly useful metrics indicate the degree to which performance goals, are being met, and they drive actions taken to improve an organization's overall performance program. Distinguishing metrics meaningful primarily to those with direct responsibility for performance management from those that speak directly to executive management interests and issues is critical to development of an effective performance metrics program.

Smart metric

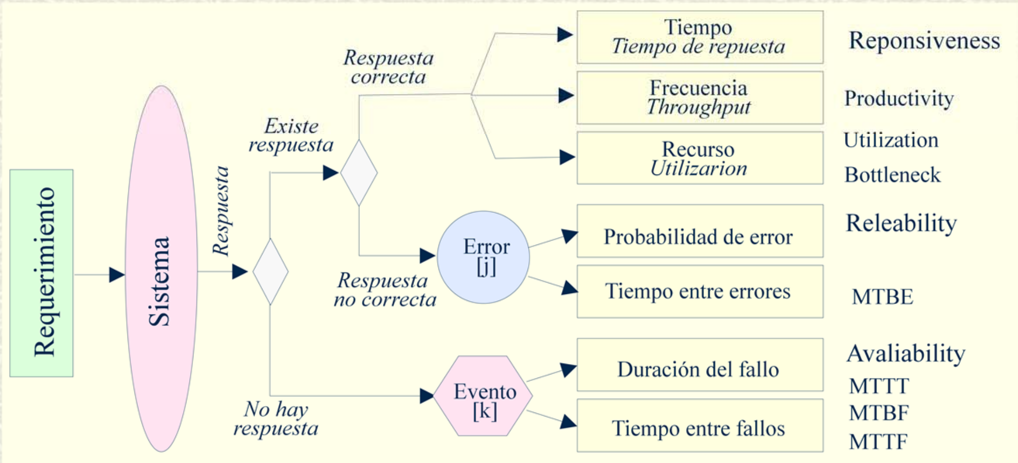
Specific	Metrics should be well defined, using unambiguous language that requires no judgment or interpretation by measurement takers.
Measurable	Metrics by definition must be quantitative in nature. If something can be counted or weighed, it is measurable.
Attainable	Some measurements are specific and theoretically measurable, but repeated measurement is not practical. Metrics, therefore, must be within both the budgetary and technical limitations of the measurement takers.
Repeatable	Metrics, by definition, consist of measurements. Those measurements are often gathered by different people at different times, and potentially, across many organizations. The key to good science is repeatability, i.e., when two different measurement takers look at the same phenomenon, they should each record the same measurement.
Time-Dependent	Time-dependence is particularly important when measuring a dynamic process, such as security. Metric contexts are typically time-dependent, both because the setting of baselines requires multiple time slices and because measurements themselves are only valid for finite periods of time.

Notas:

These measurement strategies, however, do not necessarily imply good metrics. In fact, many of them are biased, subjective, and not repeatable. If measurements are instantaneous snap-shots of a particular parameter, then metrics are more complete pictures, typically comprised of several measurements, baselines, and other supporting information that provide context for interpreting the measurements. Good metrics are goal oriented and exhibit, according to George Jelen of the International Systems Security Engineering Association, SMART characteristics: Specific, Measurable, Attainable, Repeatable, and Time dependent.

Clasificación de las métricas de performance

- Un primer criterio de clasificación de las métricas se puede basar en si se mide la no operatividad o la operatividad, y en este caso si se mide la operación correcta o no.



MTBE:Mean Time Between Failures MTTF:Mean Time To Failure MTBE:Mean Time Between Error MTTT:Mean Time To Transition

Notas:

Generally, these outcomes can be classified into three categories:

The system may perform the service correctly, incorrectly, or refuse to perform the service.

If the system performs the service correctly, its performance is measured by the time taken to perform the service, the rate at which the service is performed, and the resources consumed while performing the service. These three metrics related to **time-rate-resource** for successful performance are also called **responsiveness**, **productivity**, and **utilization** metrics, respectively. For example, the responsiveness of a network gateway is measured by its response time—the time interval between arrival of a packet and its successful delivery. The gateway's productivity is measured by its throughput—the number of packets forwarded per unit of time. The utilization gives an indication of the percentage of time the resources of the gateway are busy for the given load level. The resource with the highest utilization is called the **bottleneck**. Performance optimizations at this resource offer the highest payoff. Finding the utilization of various resources inside the system is thus an important part of performance evaluation.

If the system performs the service incorrectly, an **error** is said to have occurred. It is helpful to classify errors and to determine the probabilities of each class of errors. For example, in the case of the gateway, we may want to find the probability of single-bit errors, two-bit errors, and so on. We may also want to find the probability of a packet being partially delivered (fragment).

If the system does not perform the service, it is said to be *down, failed*, or *unavailable*. Once again, it is helpful to classify the failure modes and to determine the probabilities of each class. For example, the gateway may be unavailable 0.01% of the time due to processor failure and 0.03% due to software failure.

Número de métricas en un sistema

- Por cada servicio del sistema puede existir un requisito relativo a cada unos de los aspectos:
  - Velocidad o productividad (*speed, productivity*)
  - Fiabilidad (*reliability*)
  - Disponibilidad (*availability*)

  Como un sistema ofrece muchos servicios, el número de métricas que deben utilizarse en el análisis de un sistema pueden ser muchas.
- Para algunas métricas su valor medio es suficientes, pero para otras su variabilidad es muy relevante y requiere varios parámetros: Valor medio, varianza, desviación típica, sesgo, percentiles.
- Es habitual que los recursos del sistema sean compartidos por muchos usarios. Y en base a ello, hay dos tipos:
  - Métricas individuales: Reflejan la performance que corresponde a cada usuario
  - Metricas globales: Reflejan la utilizacióngloba l del sistema en su conjunto.

  La optimización de unas y otras pueden ser diferentes y/o contrapuestas.

Notas:

The metrics associated with the three outcomes, namely successful service, error, and unavailability, are also called **speed, reliability**, and **availability** metrics. It should be obvious that for each service offered by the system, one would have a number of speed metrics, a number of reliability metrics, and a number of availability metrics. Most systems offer more than one service, and thus the number of metrics grows proportionately.

For many metrics, the mean value is all that is important. However, do not overlook the effect of variability. For example, a high mean response time of a timesharing system as well as a high variability of the response time both may degrade the productivity significantly. If this is the case, you need to study both of these metrics.

In computer systems shared by many users, two types of performance metrics need to be considered: individual and global. Individual metrics reflect the utility of each user, while the global metrics reflect the systemwide utility. The resource utilization, reliability, and availability are global metrics, while response time and throughput may be measured for each individual as well as globally for the system. There are cases when the decision that optimizes individual metrics is different from the one that optimizes the system metric. For example, in computer networks, the performance is measured by throughput (packets per second). In a system where the total number of packets allowed in the network is kept constant, increasing the number of packets from one source may lead to increasing its throughput, but it may also decrease someone else’s throughput. Thus, both the systemwide throughput and its distribution among individual users must be studied. Using only the system throughput or the individual throughput may lead to unfair situations.

Given a number of metrics, use the following considerations to select a subset: low variability, nonredundancy, and completeness. Low variability helps reduce the number of repetitions required to obtain a given level of statistical confidence. Confidence level that are ratios of two variables generally have a larger variability than either of the two variables and should be avoided if possible.

Criterios de selección de las métricas

- Cuando hay excesivas métricas y hay que elegir un subconjunto, debe seguirse los criterios de:
  - Baja variabilidad: Permiten obtener los niveles de confianza estadistica con un número pequeño de muestras.
  - No redundantes: Si dos métricas dan una información equivalente, se debe seleccionar sólo una de ellas.
  - Completitud: Todos los aspectos de interés deben estar cubiertos por el conjunto de las métricas seleccionadas.

Notas:

Given a number of metrics, use the following considerations to select a subset: low variability, nonredundancy, and completeness.

Low variability helps reduce the number of repetitions required to obtain a given level of statistical confidence. Confidence level that are ratios of two variables generally have a larger variability than either of the two variables and should be avoided if possible.

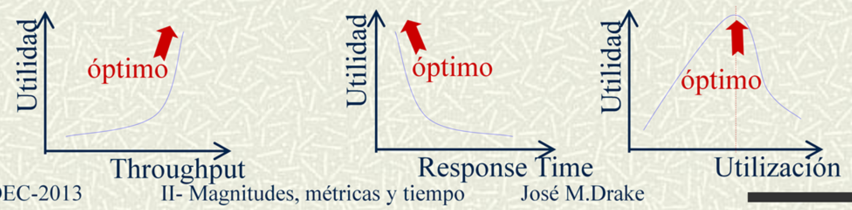
If two metrics give essentially the same information, it is less confusing to study only one. This is not always obvious, however. For example, in computer networks, the average waiting time in a queue is equal to the quotient of the average queue length and the arrival rate. Studying the average queue lengths in addition to average waiting time may not provide any additional insights.

Finally, the set of metrics included in the study should be complete. All possible outcomes should be reflected in the set of performance metrics. For example, in a study comparing different protocols on a computer network, one protocol was chosen as the best until it was found that the best protocol led to the highest number of premature circuit disconnections. The *probability of disconnection* was then added to the set of

performance metrics.

Clasificación de las métricas por lo que se desea

- Los valores de una misma métrica pueden tener diferentes interpretaciones en función del efecto de utilidad que se busca:
  - Cuanto mas alto mejor:
    - Cuanto mas alto sea el throughput el sistema tiene mayor capacidad
  - Cuanto mas bajo mejor:
    - Cuanto mas baja sea el tiempo de respuesta el sistema es mas rápido.
  - Cuento mas próximo a un valor nominal mejor:
    - Una utilización alta conduce a un tiempo de respuesta bajo, mientras que un valor bajo conduce a un bajo throughput. Mantener la utilización en un nivel determinado es lo óptimo.



DEC-2013 II- Magnitudes, métricas y tiempo José M.Drake 7

Notas:

Depending upon the utility function of a performance metric, it can be categorized into three classes:

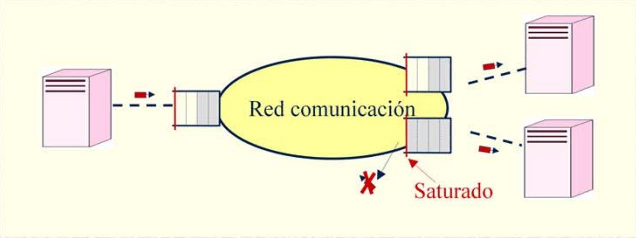
• *Higher is Better* or **HB**. System users and system managers prefer higher values of such metrics. System throughput is an example of an HB metric.

• *Lower is Better* or **LB**. System users and system managers prefer smaller values of such metrics. Response time is an example of an LB metric.

• *Nominal is Best* or **NB**. Both high and low values are undesirable. A particular value in the middle is considered the best. Utilization is an example of an NB characteristic. Very high utilization is considered bad by the users since their response times are high. Very low utilization is considered bad by system managers since the system resources are not being used. Some value in the range of 50 to 75% may be considered best by both users and system managers.

Ejemplo de selección de métricas

- Se comparan dos algoritmos de control de la saturación de una red de computadores.



- Cuando un terminal envía un conjunto de paquetes a otro por la red:
  - Algunos paquetes son entregados al destinatario en el orden previsto.
  - Algunos paquetes son entregados al destinatario en un orden erróneo.
  - Algunos paquetes son entregados duplicados
  - Algunos paquetes son perdidos

Notas:

Consider the problem of comparing two different congestion control algorithms for computer networks. A computer network consists of a number of **end systems** interconnected via a number of **intermediate systems**. The end systems send packets to other end systems on the network. The intermediate systems forward the packets along the right path. The problem of congestion occurs when the number of packets waiting at an intermediate system exceeds the system's buffering capacity and some of the packets have to be dropped.

The system in this case consists of the network, and the only service under consideration is that of packet forwarding. When a network user sends a block of packets to another end station called **destination**, there are four possible outcomes:

1. Some packets are delivered in order to the correct destination.

2. Some packets are delivered out of order to the destination.

3. Some packets are delivered more than once to the destination (duplicate packets).

4. Some packets are dropped on the way (lost packets).

Ejemplo de selección de métricas

- Métricas de performance útiles para el nudo de comunicación pueden ser:
  - Tiempo medio de respuesta: Retraso de un mensaje desde que se envía a que se recibe.
  - Variabilidad del tiempo de respuesta (p.e. varianza σ^2): Una alta varianza puede requerir mayores buffers.
  - Flujo (throughput): Número de paquetes por segundo que envía un terminal.
  - Tiempo de entrega de un paquete en el terminal que envía.
  - Tiempo de recepción de un paquete en el terminal de recepción.
  - La probabilidad de paquetes fuera de orden: Evita la entrega inmediata lo que requiere buffer de salida mas grandes.
  - La probabilidad de paquetes duplicados: Hacen uso de recursos de forma inútil.
  - Factor de imparcialidad (fairless): Todos los usuarios reciben un throughput equivalent.

$$f(\phi_1,\phi_2,..) = \frac{\left(\sum \phi_i\right)^2}{n\sum\left(\phi_i^2\right)} \quad \begin{array}{ll} \to 1 & \textit{Todos reciben el mismo throughput} \\ \to \frac{k}{n} & \textit{Sólo k de los n elementos reciben paquetes} \\ \geq 0 & \textit{En cualquier caso} \end{array}$$

- Tras varios análisis se descubre que tiempo de respuesta y el throughput están relacionadas, y se sustituyen por Potencia =Throughput/Tiempo de respuesta.

Notas:

For packets delivered in order, straightforward application of the time-rate-resource metrics produces the following list:

1. Response time: the delay inside the network for individual packets.

2. Throughput: the number of packets per unit of time.

3. Processor time per packet on the source end system.

4. Processor time per packet on the destination end systems.

5. Processor time per packet on the intermediate systems.

The response time determines the time that a packet has to be kept at the source end station using up its memory resources. Lower response time is considered better. The throughput is the performance as seen by the user. Larger throughput is considered better.

5. The variability (variance) of the response time is also important since a highly variant response results in unnecessary retransmissions.

6. the probability of out-of-order arrivals :Out-of-order packets are undesirable since they cannot generally be delivered to the user immediately. In many systems, the out-of-order packets are discarded at the destination end systems. In others, they are stored in system buffers awaiting arrival of intervening packets. In either case, out-of-order arrivals cause additional overhead.

7. Probability of duplicate packets: Duplicate packets consume the network resources without any use

8. The probability of lost packets: Lost packets are undesirable for obvious reasons.Excessive losses result in excessive retransmissions and could cause some user connections to be broken prematurely.

The network is a multiuser system. It is necessary that all users be treated fairly. Therefore, fairness was added as the eleventh metric. It is defined as a function of variability of throughput across users. For any given set of user throughputs ($\phi 1$, $\phi 2$,..., ϕn), the following function can be used to assign a fairness index to the f($\phi 1$, $\phi 2$,..., ϕn) function. The fairness index always lies between 0 and 1. If all users receive equal throughput, the fairness index is 1. If only *k* of the *n* users receive equal throughput and the remaining $n - k$ users receive zero throughput, the fairness index is *k/n*. For other distributions also, the metric gives intuitive fairness values.

After a few experiments, it was clear that throughput and delay were really redundant metrics. All schemes that resulted in higher throughput also resulted in higher delay. Therefore, the two metrics were removed from the list and instead a combined metric called **power**, which is defined as the ratio of throughput to response time, was used. A higher power meant either a higher throughput or a lower delay; in either case it was considered better than a lower power.

Principales métricas para especificar performance

- Las métricas para especificar el comportamiento o la calidad de servicio de un sistema son muy dependientes de su naturaleza.
- Aquí describimos algunas que son de gran aplicabilidad:
  - Tiempo de respuesta.
  - Throughput
  - Eficiencia
  - Utilización
  - Fiabilidad
  - Disponibilidad
  - Seguridad
  - Escalabilidad

Notas:

For each performance study, a set of performance criteria or metrics must be chosen.

Tiempo de respuesta

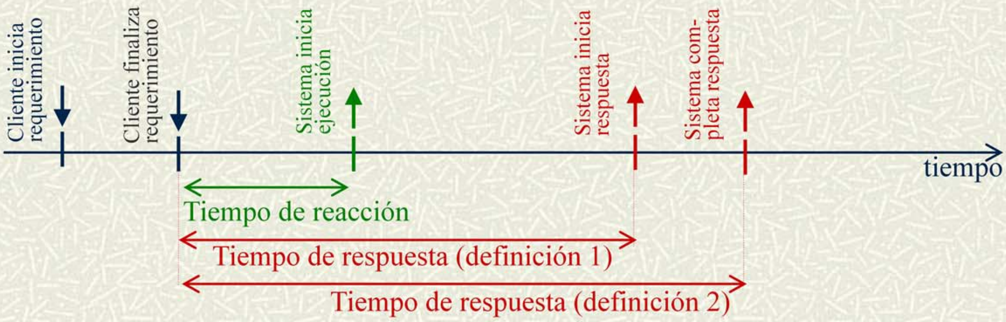
- Es el tiempo que transcurre entre que el cliente realiza un requerimiento y el sistema proporciona (o ejecuta) la respuesta.

Esta definición es simplista e imprecisa, ya que el requerimiento y la respuesta nos son acciones instantáneas:

Notas:

Response time is defined as the interval between a user's request and the system response, as shown in the Figure. This definition, however, is simplistic since the requests as well as the responses are not instantaneous.

Componentes del tiempo de respuesta



Tiempo de reacción: Es el tiempo que transcurre entre la finalización de la invocación del servicio y el comienzo de las actividades del servicio en el sistema.

Tiempo respuesta (Definición 1): Es el tiempo que transcurre entre la finalización de la invocación del servicio, y el inicio de generación de la respuesta

Tiempo respuesta (Definición 2): Es el tiempo que transcurre entre la finalización de la invocación del servicio y la finalización de la generación de la respuesta.

Notas:

The users spend time typing the request and the system takes time outputting the response, as

shown in Figure 3.2b. There are two possible definitions of the response time in this case. It can be defined as

either the interval between the end of a request submission and the beginning of the corresponding response

from the system or as the interval between the end of a request submission and the end of the corresponding

response from the system. Both definitions are acceptable as long as they are clearly specified. The second definition is preferable if the time between the beginning and the end of the response is long. Following this definition, the response time for interactive users in a timesharing system would be the interval between striking the last return (or enter) key and the receipt of the *last* character of the system's response.

For a batch stream, responsiveness is measured by **turnaround time**, which is the time between the

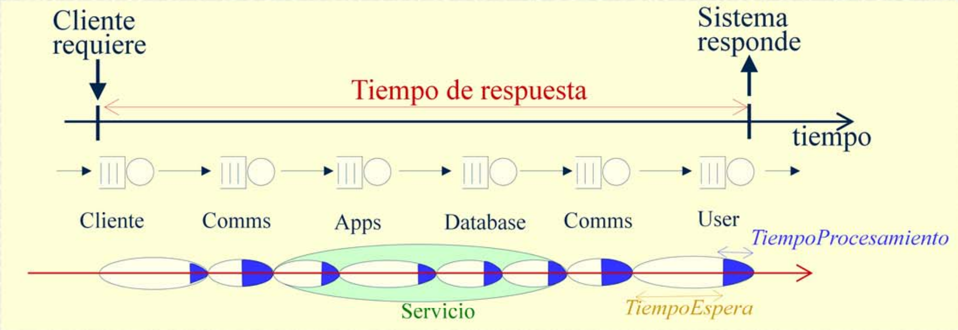
submission of a batch job and the completion of its output. Notice that the time to read the input is included in

the turnaround time.

The time between submission of a request and the beginning of its execution by the system is called the **reaction time**. To measure the reaction time, one has to able to monitor the actions inside a system since the beginning of the execution may not correspond to any externally visible event. For example, in timesharing systems, the interval between a user's last key stroke and the user's process receiving the first CPU quantum

would be called reaction time.

Tiempo de respuesta y cuellos de botella



- Un servicio distribuido esta compuesto por una cadena de actividades que se ejecutan en diferentes procesos lógicos mapeados sobre diferentes recursos. Cada etapa del servicio tiene dos fases: la espera a que el procesos esté disponible (*awaiting time*), el tiempo de procesamiento que requiere la actividad (*processing time*).
- El tiempo de espera (directamente) y el tiempo de procesamiento (indirectamente) dependen de que otros servicios, de los que también usan el mismo recurso, se están ejecutando. Por ello, el tiempo de espera end-to-end se estima haciendo uso de un modelo de colas.

Notas:

Distributed computer systems are composed of logical processes mapped onto a number oo d physical computing resources. A request tipically requires the uses of a sequence of these logica processes. The time it takes at each stage to process the transaction adds up to the response time observed by the user that initiated the transaction. Users are often concerned with end-to-end-response time. The process with the longest rocessing time is a key determinant of the response time. It is given a special name of the bottleneck.

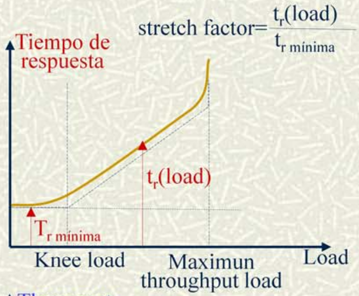
The view of the user is not far removed from that of the computer performnce analyst in that both aim to reconcile the transaction response time with the presence of system bottlenecks. In general, both the users and the analyst like to know the processing times for each of the sequential software component that process the database transaction during it proccess through the system. On average, the sum of this processing times should be equal to the measured end-to-end response time for the transaction type, within some prescribed tolerance.

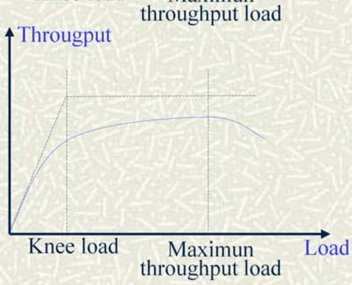
Since many transactions can be processed simultaneosly at each stage, there exists the possibility of contention among these transaction at each queueing center. The processing time at each stage is then given by the sum of the time the transaction request waits to obtain the necessary processing resources plus the time it actually takes to get processed when it owns the resource. Cleary, the more heavily loaded the system, the longer the queues wil tehnd to be and therefore the longer time spente each stage.

The end-to-end response time is the sumof all these processing time (more formally, residence times) and depends of the queue length at each stage.

Dependencia del tiempo de respuesta de la carga

- El tiempo de respuesta de un sistema se incrementa (empeora) cuando se incrementa la carga. La dependencia entre el tiempo de respuesta y la carga suele ser la que se muestra en la gráfica:
  - **Carga óptima** (*knee load*) es la mayor carga en la que el tiempo de respuesta es próximo el óptimo (mínimo).
  - **Factor de esfuerzo** (*stretch factor*) para una carga dada es la relación entre el tiempo de respuesta del sistema a esa carga y el que corresponde a la mínima carga.
  - **Carga de colapso** (maximun throughput load). Es la carga que colapsa la capacidad del sistema, esto es, un incremento de la carga no incrementa el throughput del sistema.



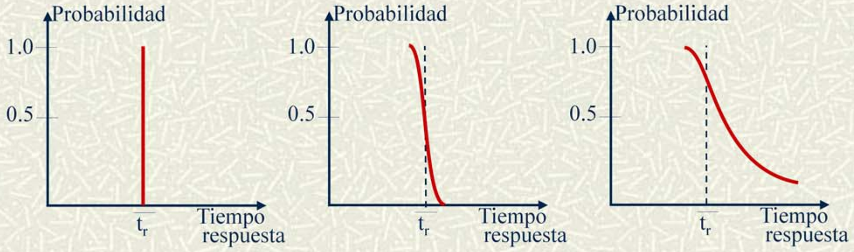


Notas:

The response time of a system generally increases as the load on the system increases. The ratio of response time at a particular load to that at the minimum load is called the **stretch factor**. For a timesharing system, for example, the stretch factor is defined as the ratio of the response time with multiprogramming to that without multiprogramming.

Variabilidad del tiempo de respuesta (Jitter)

- La variabilidad del tiempo de respuesta es una magnitud que tiene gran influencia en el comportamiento de un sistema:
  - Si la variabilidad es nula el sistema es predecible
  - Determina la longitud de los buffers que se requieren.
- El termino jitter tiene un significado diferentes en cada dominio en que se emplea. Es mejor hablar de variabilidad del tiempo de respuesta.
- Se mide como cualquier variabilidad: Rango, varianza, percentiles, etc.



Caso **ideal** (Jitter=0) | Caso **bueno** (Jitter bajo) | Caso **malo** (Jitter alto)

Notas:

Throughput

- Throughput es la frecuencia con la que los requerimientos del cliente pueden ser ejecutados por el sistema.
- Corresponde a una métrica de capacidad del sistema.
- Se mide en:
  - Ops o TTS:Operaciones/segundo o transacciones/s
  - Mips o MFLOPS: Millones de instrucciones/s o Millones de operaciones de punto flotantes/s
  - pps, bps: Paquetes por segundo. Bit por segundos.
- El througput nominal de un procesador, red de comunicación o dispositivo de almacenamiento de datos se denomina anchura de banda (*Bandwidth*)

Notas:

Throughput is defined as the rate (requests per unit of time) at which the requests can be serviced by the

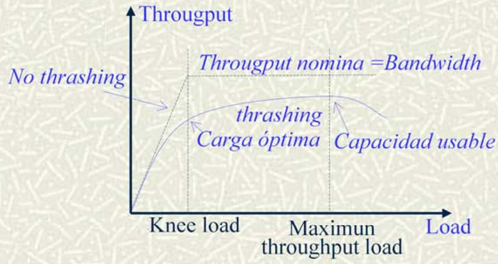
system. For batch streams, the throughput is measured in jobs per second. For interactive systems, the

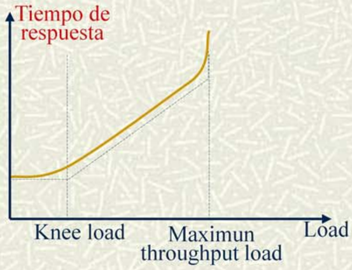
throughput is measured in requests per second. For CPUs, the throughput is measured in Millions of Instructions Per Second **(MIPS)**, or Millions of Floating-Point Operations Per Second **(MFLOPS)**. For networks, the throughput is measured in packets per second **(pps)** or bits per second **(bps)**. For transactions processing systems, the throughput is measured in Transactions Per Second **(TPS)**.

For computer networks, the nominal capacity is called the **bandwidth** and is usually expressed in bits per second.

Dependencia del throughput con la carga

- Para bajas cargas el througput se incrementa linealmente con la carga.
- Cuando la carga se acerca a la máxima capacidad, la pendiente del througput con el dibujo disminuye (*thrashing*).
- Knee load representa el punto de operación óptimo: un throughput alto con un tiempo de respuesta próximo al mínimo.
- El sistema no es capaz de alcanzar el throughput nominal (*Bandwidth*)





DEC-2013 II- Magnitudes, métricas y tiempo José M.Drake 17

Notas:

The throughput of a system generally increases as the load on the system initially increases. After a certain load, the throughput stops increasing; in most cases, it may even start decreasing, as shown in teh figure. The maximum achievable throughput under ideal workload conditions is called **nominal capacity** of the system. For computer networks, the nominal capacity is called the **bandwidth** and is usually expressed in bits per

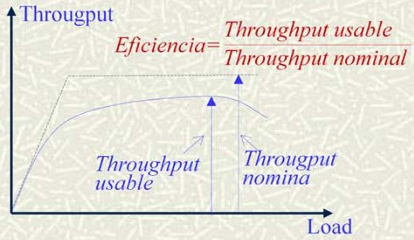
second. Often the response time at maximum throughput is too high to be acceptable. In such cases, it is more interesting to know the maximum throughput achievable without exceeding a prespecified response time limit. This may be called the **usable capacity** of the system. In many applications, the knee of the throughput or the response-time curve is considered the optimal operating point. As shown in Figure 3.3, this is the point

beyond which the response time increases rapidly as a function of the load but the gain in throughput is small.

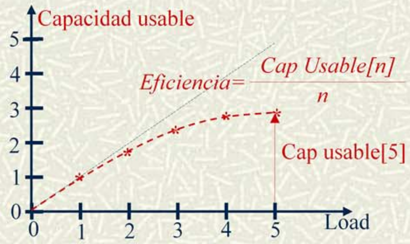
Before the knee, the response time does not increase significantly but the throughput rises as the load increases. The throughput at the knee is called the **knee capacity** of the system. It is also common to measure capacity in terms of load, for example, the number of users rather than the throughput. Once again, it is a good idea to precisely define the metrics and their units before using them in a performance evaluation project.

Eficiencia (*Efficiency*)

- Es la relación entre el máximo throughput usable en el sistema y su throughput nominal.



- También se utiliza en el caso de sistema multiprocesadores, al que corresponde la curva típica de la figura adjunta.



Notas:

The ratio of maximum achievable throughput (usable capacity) to nominal capacity is called the **efficiency**. For example, if the maximum throughput from a 100-Mbps (megabits per second) Local Area Network (LAN) is only 85 Mbps, its efficiency is 85%. The term efficiency is also used for multiprocessor systems. The ratio of the performance of an *n*-processor system to that of a one-processor system is its efficiency, as

shown in the figure. The performance is usually measured in terms of MIPS or MFLOPS.

Utilización (*Utilization*)

- La utilización de un recurso es el tanto por ciento del recurso que se encuentra utilizado en procesar requerimientos.
  - Hay recursos que sólo pueden estar en dos estados ocupado y libre (por ejemplo la CPU). En este caso la utilización se mide como el tiempo en que se encuentra ocupado frente al tiempo en que se realiza la medida.
  - Los recursos que pueden usarse parcialmente (por ejemplo la memoria), la utilización se mide como el tanto por ciento del recurso que se está utilizando en cada instante, y la utilización en un intervalo de tiempo como el promedio de las utilizaciones instantáneas en ese intervalo.
- La utilización es la métrica básica para detectar los recursos que constituyen los cuellos de botella que limitan la velocidad de respuesta del sistema.
- Una estrategia típica de optimización del sistema consiste en balancear la carga entre los diferentes recursos, de forma que se ecualice la utilización de los recursos.

Notas:

The **utilization** of a resource is measured as the fraction of time the resource is busy servicing requests. Thus this is the ratio of busy time and total elapsed time over a given period. The period during which a resource is not being used is called the **idle time**. System managers are often interested in balancing the load so that no one resource is utilized more than others. Of course, this is not always possible.

Some resources, such as processors, are always either busy or idle, so their utilization in terms of ratio of busy time to total time makes sense. For other resources, such as memory, only a fraction of the resource may be used at a given time; their utilization is measured as the average fraction used over an interval.

Averías, errores y fallos

- Son términos usados en la vida cotidiana, y por consiguiente muy sobrecargados de diferentes significados: En los estudios de confiabilidad deben estar definidos de forma precisa ya que se requiere cuantificarlos:
  - Avería, falta (*fault*): Es una desviación del comportamiento del sistema respecto del comportamiento especificado como correcto. Puede ser: Una avería hardware, un código erróneo, una equivocación de un operador, una caída de potencia, etc.
  - Error: Es un estado incorrecto del sistema como consecuencia de una avería o falta. Un error puede manifestarse como fallo, o permanecer oculto y sin consecuencias.
  - Fallo (*Failure*): Es la manifestación externa de un error que es detectable por un operador humano, u otro sistema que interacciona con él.
- Las métricas claves relativas a las fiabilidad son:
  - Tiempo medio entre fallos (MTBF)
  - Tiempo medio para el fallo (MTTF)
  - Tiempo medio de reparación (MTTR)

Notas:

The words fault, error and failure have a plethora of definitions in the literature:

A fault is a deviation of the behavior of a system from the authoritative specificaction of its behavior. A hardware fault is a physical change in hardware that cause the system to change ist behavior in an undesirable way. A software fault is a mistake (bug) in the code. A procedural fault consist of a mistake by a person in carrying out some procedure. An environmental fault is a desviation from expected behavior of the world outside the computer system, electric power interruption is an example.

An error is an incorrect state of hardware, software or data resulting from a component failure, a software bug, physical interference from the enviroment, an operator mistake, or incorrect design. An error is, therefore, that part of the system state which is liable to lead to failure. Upon ocurrence, a fault creates a latent error, which becomes effective when it is activated, leading a failure. If never activated, the latent error never becomes effective.

A failure is the external manifestation of an error within a program or dato structure. That is , a failure is the external efect of the error, as seen by a (human or physical device) user, or by another program.

Mean time between failures MTBF (θ) is the sum of the operational periods divided by the number of observed failures. If the "Down time" refers to the start of "downtime" and "up time" refers to the start of "uptime", the formula will be:

MTBF= ∑(Start of downtime – Start of uptime)/Number of failure

Mean Time To Failure (MTTF): An estimate of the average, or mean time until a design's or component's first failure, or disruption in the operation of the system. Mean time until a failure assumes that the product CAN NOT be repaired and the product CAN NOT resume any of it’s normal operations.

Mean time to repair (MTTR): It represents the average time required to repair a failed system.

Métricas relacionadas con la confiabilidad (1)

- La fiabilidad (*Realibility*) R(t) de un sistema es la probabilidad de que el sistema se haya mantenido sin fallo en el intervalo [0,t], supuesto que en el instante 0 estaba operativo.
- La fiabilidad R(t) se formula en función de la frecuencia de fallo λ:

$$\lambda(t) = -\frac{d(\ln R(t))}{dt}$$

- o del tiempo medio para fallo MTTF:

$$MTTF = \frac{\int_0^\infty t\, R(t)\, dt}{\int_0^\infty R(t)\, dt}$$

- Si la distribución de tiempos para el fallo es de tipo exponencial:

$$R(t) = R_o \in^{-\lambda t} = R_o \in^{-\frac{t}{MTTF}} \qquad \Rightarrow \qquad MTTF = \frac{1}{\lambda}$$

Notas:

The reliability R(t) of a device or system is the conditional probability that the devive or system has survived the interval [0,t], given that it was operating at time 0. reliability is often given in terms of the failure rate λ(t) or the mean time to failure mttf.

If the failure time is constant, mttf= 1/ λ.

Métricas relacionadas con la confiabilidad (2)

- La disponibilidad (*Availability*) A(t) de un sistema es la probabilidad de que el sistema esté operacional en el instante t si en el instante inicial estaba operativo.
- Para sistemas sin reparaciones la disponibilidad y la fiabilidad de un sistema coinciden. Si embargo si los sistemas fallan y se recuperan, ambos conceptos son diferentes.
  - Por ejemplo un sistema que se caiga cada 1 segundo, pero se recupere cada 1 μs, tiene una alta disponibilidad, pero una baja fiabilidad.
- La mantenibilidad (*maintainability*)de un sistema M(t) es la probabilidad condicional de que en un tiempo t este operativo, si en el instante 0 estaba averiado.
- La mantenibilidad se puede expresar en función de la frecuencia de reparación μ(t) y del tiempo medio de reparación MTTR:

$$\mu(t) = -\frac{d(\ln(1-M(t))}{dt} \qquad MTTR = \int_0^\infty (1-M(t))dt$$

$$Si\ la\ distrubción\ es\ exponencial \Rightarrow \quad MTTR = \frac{1}{\mu}$$

Notas:

The availability A(t) of a system is the probability that the system is operational at the instant t. For no repairable systems , availability and reability are equal. For repairable systems, they are not.

The mantainnability M(t) of a suystem is the conditional probability that the system wilbe restored to operational effectiveness by time t, given that it was not functioning at time 0. Maintaibility if often given in terms of the repair rate μ(t).

The saety S(t) of a system is te conditional probability that the system has not encountered a catastrophic failure by time t, given that threre was no catastrophic failure at time 0.

Unfortunately, the term reliability has both a specific and a broad meaning. The specific meaning is given earlier in this secction. The broader use is a general term for (specific) reliability, availability maintainability and safety, as defined above.

Disponibilidad (*Availability*)

- La disponibilidad de un sistema mide la probabilidad de que el sistema este operando en el tiempo t.
- La disponibilidad de un sistema se mide como la fracción de tiempo en que el sistema esta disponible para atender requerimientos de servicio.

$$A = \frac{T_{Up}}{T_{Up} + T_{Down}}$$

T_{up} => El tiempo con sistema disponible.
T_{down} =>Tiempo con sistema no disponible.

Si X(t) es la función status $X(t) = \begin{vmatrix} 0 & \textit{si sistema operativo} \\ 1 & \textit{Sisistema no operativo} \end{vmatrix}$

La disponibilidad A es $A = \lim_{T\to\infty} \frac{1}{T}\int_0^T X(t)\, dt$

- La no disponibilidad de un sistema pueden ser por:
  - Avería que requiere mantenimiento.
  - Inicialización o configuración programada
  - Congestión por sobrecarga (se resuelve con gestión de accesos).

Notas:

the term **availability** has the following meanings:

•The degree to which a system, subsystem, or equipment is in a specified operable and committable state at the start of a mission, when the mission is called for at an unknown, *i.e.*,a random, time. Simply put, availability is the proportion of time a system is in a functioning condition. This is often described as a **mission capable rate**. Mathematically, this is expressed as 1 minus unavailability.

•The ratio of (a) the total time a functional unit is capable of being used during a given interval to (b) the length of the interval.

For example, a unit that is capable of being used 100 hours per week (168 hours) would have an availability of 100/168. However, typical availability values are specified in decimal (such as 0.9998). In high availability applications, a metric known as nines, corresponding to the number of nines following the decimal point, is used. In this system, "five nines" equals 0.99999 (or 99.999%) availability.

Metricas de la disponibilidad

- Las métricas que suelen utilizarse son:
  - Tanto por ciento disponibilidad

    %Disponibilidad= $T_{up}/(T_{up}+ T_{down})$.

    No es útil si T_{up} y T_{down} son pequeños.
  - MTBF: Tiempo medio entre fallos.
  - MTTF: Tiempo medio de fallo.
  - MTTR: Tiempo medio de reparación: Incluye tiempo de respuesta, de evaluación del fallo y de reparación

$$A = \%Disponibilidad = \frac{MTBF}{MTBF + MTTF}$$

Notas:

Ejemplos: Disponibilidad

- Caso 1:

  Un sistema con un 99.99% de disponibilidad en una ventana de 30 días tiene un tiempo de fallo de

  (1-0.9999) x 30días x 24 horas/dia x 60 min/hora=4.32 minutos

- Caso 2:

  Si utilizamos un equipo con un tiempo medio entre fallo de 81.5 años y un tiempo de reparación de 1 hora.

  - MTBF en horas = 81.5*365*24=713940 hr
  - Availability= MTBF/(MTBF+MTTR) = 713940/713941 =99.999859%
  - Unavailability = 0.000141%
  - Fallos debidos al equipo en hr/año   U=0.01235 hours per year.

Notas:

Cálculo de la disponibilidad: tirania de los 9

- Tiempo de fallos con una disponibilidad de 99.99..9%:
  - 1 año ≈ 32 10^6 s

%Disponibilidad	Fracción de año	Tiempo caído
99.9	0.001	~9 h/año
99.99	0.0001	~ 1 h/año
99.9999	0.000001	~ 32 s/año

Notas:

Modelos de fiabilidad y disponibilidad

- Hay diferentes técnicas para describir la fiabilidad y disponibilidad de sistemas complejos compuestos por muchos subsistemas, cada uno de ellos con su propia disponibilidad.
- Es mas fácil modelar la fiabilidad y la seguridad (no tienen reparación) que la disponibilidad (con reparación)
- Aquí mencionamos 3 modelos:
  - Modelos basados en diagramas de bloque de fiabilidad.
  - Modelos basados en árboles de fallos
  - Modelos basados en modelos de Markov.

Notas:

Cuando se considera reliability, availability and safety are used a wider variety of models. There are two major reasons for this: differences in kind and in degree among faults.reliability, availability and safety; and differences in kind an degree among the varieties of faults.

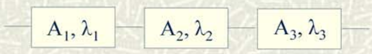
The operational reliability of a nonrepairable system is much easier than the case os a repairable system.

There are many overlaps in modeling techniques as well as many differences. Consequently, the analyst needs to be familiar with several techniques. Three are described here: Reliability block diagrams, fault trees, and Markov models. The description are necessary simplifieds.

Reliability modeling generally requires computer assistence.

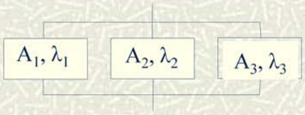
Diagramas de bloques de fiabilidad

- Se considera que un sistema S está compuesto de diferentes subsistemas S_1, S_2, S_3... La fiabilidad de cada uno está especificada por su frecuencia de fallo λ_i.
- En el diagrama de bloques de fiabilidad cada bloque representa un componente, y su conexión representa la dependencia de fallo:
  - Conexión serie: El sistema falla, si falla uno de los subsistemas



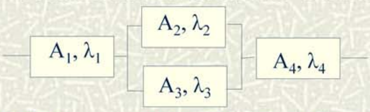
$$\lambda_S = \sum_{\forall i} \lambda_i \qquad A_s = \prod_{\forall i} A_i$$

  - Conexión paralela: Para que el sistema falle deben fallar todos los subsistemas



$$\frac{1}{\lambda_S} = \sum_{\forall i} \frac{1}{\lambda_i} \qquad A_s = 1 - \prod_{\forall i} (1 - A_i)$$

  - Conexiones en escalera: Combinaciones de conexiones serie y paralelas:



$$A_s = A_1 \left(1 - (1 - A2)(1 - A3)\right) A4$$

Notas:

Reliability Block diagrams are easy to construct and analyze, and are completely adequate for many cases involving the operational realiability of simple systems.

We consider a system S composed of n component S1,S2, Sn. Each component will continue to operate until it fails; reapair is not carried out. The question is: what is the reliability of the entire system?.

Let us suppose that component Cj has a constant failure rate λ_j (and therefore a MTTF=1/ λ_j)

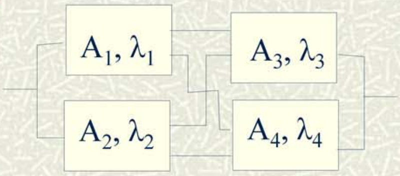
In the reliabilty block diagram, block represent components. These are connected together to represent failure dependencies.

If the failure of any of a set of components will cause the system to fail, a series connection is appropiated.

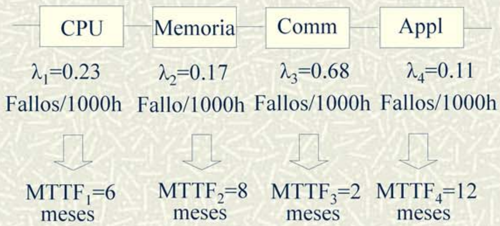
If the system will fail only if all component fail, a parallel connection is appropiated. More complex topologies are olso possible.

Diagramas de bloques de fiabilidad (2)

- Diagramas que no tiene estructura descomponible en serie/paralelo: No se pueden analizar



- Ejemplo:



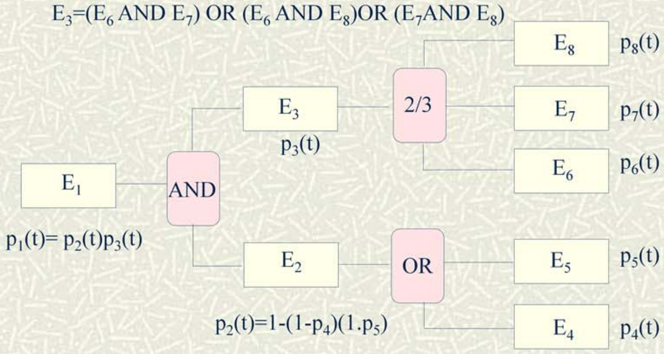
$$\lambda_S = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1.19\ (fallos/1000h)$$
$$\Rightarrow MTTF_S = 1.15\ meses$$

Notas:

Modelo de árbol de fallos

- Los fallos de un sistema se jerarquizan desde fallos mas próximos a los subsistemas a fallos mas próximos a la funcionalidad del sistema y de mayor relevancia.
- Entre los fallos se establecen relaciones AND, OR y R de N.
- Cada fallo tiene asociada su probabilidad y distribución de tiempos entre fallos.

E_3=(E_6 AND E_7) OR (E_6 AND E_8)OR (E_7AND E_8)



DEC-2013 II- Magnitudes, métricas y tiempo José M.Drake 30

Notas:

The fault tree is developed by successively break down events into lower-level events that generate the upper level event, the tree is an extended form o and-or tree. In the diagram, the Event E1 occurs only if both of events E2 and E3 occur. E2 can occur if either of E4 or E5 occur. Event E3 will occur if any two E4 orE5 occur. Event E3 will occur if any two of E6,E7 and E8 occur.

The faut tree is expanded to righ until events are reached whose probability can be given directly. Note the assumption that the occurrence of the events at the righ side of the tree are mutuallly independent. In may cases, the actual probabilities of these events are estimated.

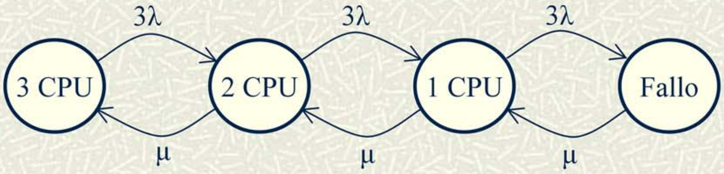
The fault tree can be evaluated from righ to left. Consider the tree in the figure, y which the righ events are not replicated. Suppose pj(t) denotes te probability that event Ej will occur by time t, and we wish to evaluate and AND node. Here, p1(t)=p2(t)p3(t); probabilities at AND nodes multiply.

If we have an OR node, s shown the figure, we have p2(t)=1-(1-p5(t))(1-p4(t).

Generalization to AND and OR node represents a boolean combination of AND and OR nodes, with more than two events should ne clear, so its evaluation is straightfoward, through tedious.

Modelo basada en máquinas de estado de Markov

- Las situaciones de fallo del sistema se representan mediante estados de la máquina de estados. Las transiciones entre estados se corresponde con la posibilidad de ocurrencia de un nuevo evento de fallo.
- En la figura se muestra el modelo de fallo de un sistema con tres CPU, de las que 2 son redundantes, yque tiene con sólo una capacidad de operar.
- λ es la frecuencia de fallo de la CPU y μ es la frecuencia de reparación.



Notas:

In the reliability models based on Markov Models, the state represents knowledge of which components are operational an which are being repaired (if any).

Systems ehose components are repairable, and systems where component failures have interactions, are usually modeled directly by Markov models cith cycles.

Let us begin with a system which contains three CPUs. Only one is required for operation, the other two provide redundancy. Only one repair station is available, so even if more than one CPU is down, repairs happen one at a time. If state k is used to mean “k CPU are operating” the Markov model is shown in the Figure.

Seguridad (*Security*)

- Hace referencia a los mecanismos de encriptación que son necesarios para garantizar la integridad de los datos que se gestionan en los servicios del sistema. Incluye tres aspectos:
  - Confidencialidad: Sólo los elementos autorizados tienen acceso a los datos.
  - Integridad de los datos: Los datos no pueden ser modificados por terceros.
  - Repudiación: No se admiten datos que no procedan de fuentes autorizadas.
- La encriptación puede llevar a un gasto de procesamiento que comprometa otras métricas de performance.

Notas:

Escalabilidad (Scalability)

- Un sistema se dice escalable si admite que se pueda aumentar sus recursos para soportar una mayor demanda del rendimiento, y/o admite disminuir sus recursos para reducir costes.
- La escalabilidad implica diferentes aspectos:
  - Escalabilidad del rendimiento: La capacidad del sistema debe incrementarse de forma proporcional al incremento de los recursos. De forma ideal, debería producirse un aumento de la capacidad próximo a un facto α cuando se incrementan los recursos en el mismo factor.
  - Escalabilidad del costo: El coste de escalado debe ser razonable. Típicamente si el escalado se produce en un factor α, se debe esperar que su coste se incremente en un factor entre α y $\alpha \log \alpha$.
  - Compatibilidad del escalado: :Cuando se produce un escalado los elementos existentes deben poderse reutilizar.

Notas:

Métricas de escalabilidad para sistemas distribuidos

- Speedup S : mide la relación entre el trabajo que realiza con el número de procesadores, y el caso de que solo haya un procesador:
  - Métrica ideal: crecimiento lineal S(K)=k
  - Un ejemplo sencillo es medir S(k)=T(1)/T(k) el tiempo en relizar un determinado trabajo.
- Efficiency E: mide el trabajo realizado por procesador, esto es E(k)=S(k)/K.
  - El valor ideal de la métrica es 1.
- Scalability ψ(k1, k2): la relación de eficiencia con k1 y k2 procesadores. ψ(k1, k2)= E(k2)/E(k1)
  - El valor ideal de la métrica es 1.

Notas:

A variety of scalability metrics have been developed for massively parallel computation, to evaluate the effectiveness of a given algorithm running on different sized platforms, and to compare the scalability of algorithms. These metrics assume that the program runs by itself, on a set of k processors with a given architecture, and that the completion time T measures the performance.

Three related kinds of metrics have been reported: speedup metrics, efficiency metrics, and scalability metrics.

The following definitions give the flavor of the proposed metrics, although there are variations in detail among different authors:

. Speedup S measures how the rate of doing work increases with the number of processors k, compared to one processor, and has an ideal linear speedup value of S(k)=k.

. Efficiency E measures the work rate per processor (that is, E(k)=S(k)/k), and has an ideal value of unity.

. Scalability ψ(k1; k2). from one scale k1 to another scale k2 is the ratio of the efficiency figures for the two cases, ψ(k1; k2)=E(k2).=E(k1). It also has an ideal value of unity.

A typical metric is the fixed size speedup, in which the scaled-up base case has the same total computational work, and the speedup S is the ratio of the completion times (i.e., S.k. . T.1.=T.k.).

Como establecer los requisitos de performance

- Una especificación como la siguientes no es aceptable:
  - *"El sistema debe se eficiente tanto en el uso del procesador y de la memoria, y no debe dar lugar a excesivo overhead".*
  - *"La probabilidad de que la red duplique un paquete, envíe uno a dirección errónea o cambie los datos , debe ser extremadamente baja".*
- La especificación de los requisitos de performance deben ser SMART:
  - Specifics (concretos): Los requisitos deben ser formulados de forma precisa. Deben eludirse palabras como "*baja probabilidad*", "*raro*", etc.
  - Measurables(medibles): Los requisitos deben ser magnitudes medibles directa o indirectamente.
  - Acceptables (Aceptables): Deben ser compatibles con la arquitectura y naturaleza del sistema.
  - Repeatable (Reproducibles): Pueden ser medidos por diferentes analistas y con resultados equivalentes.
  - Thoroughs (meticuloso): Deben incluir todos los posibles resultados y modos de fallos.

Notas:

One problem performance analysts are faced with repeatedly is that of specifying performance requirements for a system to be acquired or designed. A general method to specify such requirements is presented in this section and is illustrated with a case study.To begin, consider these typical requirement statements:

> "The system should be both processing and memory efficient. It should not create excessive overhead."
>
> "There should be an extremely low probability that the network will duplicate a packet, deliver a packet to the wrong destination, or change the data in a packet".

These requirement statements are unacceptable since they suffer from one or more of the following problems:

1. *Nonspecific:* No clear numbers are specified. Qualitative words such as low, high, rare, and extremely small are used instead.

2. *Nonmeasurable:* There is no way to measure a system and verify that it meets the requirement.

3. *Nonacceptable:* Numerical values of requirements, if specified, are set based upon what can be achieved or what looks good. If an attempt is made to set the requirements realistically, they turn out to be so low that they become unacceptable.

4. *Nonrealizable:* Often, requirements are set high so that they look good. However, such requirements may not be realizable.

5. *Nonthorough:* No attempt is made to specify a possible outcomes.

Ejemplo: Requisitos de performance de un sistema LAN

- Se quiere especificar los requisitos de performance de un sistema de comunicación LAN de alta velocidad (1 Gbit/s). Un servicio LAN provee servicios para transmitir paquetes de información a un determinado destino. Cuando un cliente invoca el envío de un mensaje a un destino D, pueden ocurrir tres cosas:
  - El mensaje es correctamente entregado en el destino D.
  - El mensaje es entregado con error en D.
  - El mensaje no alcanza el destino.

En las siguientes transparencia se muestran unos requisitos de performance que son apropiados apara este sistema:

Todos ellos deberían haber sido chequeados previamente sobre un modelo para garantizar que son compatibles con la naturaleza del sistema.

Notas:

Consider the problem of specifying the performance requirements for a high-speed LAN system. A LAN basically provides the service of transporting frames (or packets) to the specified destination station. Given a user request to send a frame to destination station D, there are three

categories of outcomes: the frame is correctly delivered to D, incorrectly delivered (delivered to a wrong destination or with an error indication to D), or not delivered at all.

Requisitos relativos a velocidad del sistema LAN

- Speed: Hace referencia a los mensjaes que han sido entregados de forma correcta. Especifica los límites temporales con los que se entregan los mensajes en D y la máxima frecuencia con los que se pueden enviar:
  - Requisito A1: El máximo tiempo de acceso desde cualquier terminal debe ser inferior a 1 segundo.
  - Requisito A2: Por la red se puede mantener un flujo (*througput*) sostenido de 80 Mbits/s
  - Requisito A3: La red debe garantizar el 12% de su anchura de banda para soporta los picos de carga que ocasionalmente se requieren.

Notas:

1. *Speed:* If the packet is correctly delivered, the time taken to deliver it and the rate at which it is delivered are important. This leads to the following two requirements:

   (a) The access delay at any station should be less than 1 second.

   (b) Sustained throughput must be at least 80 Mbits/sec.

Requisitos relativos a fiabilidad del sistema LAN

- Reliability: Se consideran 5 tipos de errores que se pueden presentar en la transmisión de un mensajes. Los requisitos relativos a las probabilidades de estos errores son:
  - Requisito B1: Error en el dato: La probabilidad de que en un mensaje se modifique un bit debe ser inferior a 10^{-7}.
  - Requisito B2: Error en paquete notificado: la probabilidad de que se introduzca un error en un paquete, siendo este notificado por el bit de error de la cabecera, debe ser inferior al 1%.
  - Requisito B3: Error en paquete no notificado: la probabilidad de que se introduzca un error en un paquete, sin que el mismo sea notificado en el bit de error de cabecera debe ser menor de 10^{-15}.
  - Requisito B4: Perdida de paquete: La probabilidad de que un paquete sea perdido como consecuencia de un error no detectado en el campo de dirección de destino debe ser inferior a 10^{-18}.
  - Requisito B5: Duplicado de paquete: La probabilidad de que un mismo paquete sea entregado mas de una vez en destino debe ser inferior a 10^{-5}.
  - Requisito B6: Error de perdida acumulado. La probabilidad de que un paquete sea perdido por cualquier causa debe ser inferior al 1%.

Notas:

2. *Reliability:* Five different error modes were considered important. Each of these error modes causes a different amount of damage and, hence, has a different level of acceptability. The probability requirements for each of these error modes and their combined effect are specified as follows:

(a) The probability of any bit being in error must be less than 10-7.

(b) The probability of any frame being in error (with error indication set) must be less than 1%.

(c) The probability of a frame in error being delivered without error indication must be less than 10-15.

(d) The probability of a frame being misdelivered due to an undetected error in the destination address must be less than 10-18.

(e) The probability of a frame being delivered more than once (duplicate) must be less than 10-5.

(f) The probability of losing a frame on the LAN (due to all sorts of errors) must be less than 1%.

Requisitos relativos a disponibilidad del sistema LAN

- Availability: Se consideran dos modos de fallo:
  - Fallos debidos a inicialización de la red por necesidades de reconfiguración planificada.
  - Fallos debidos a caída de la red que requieren se solucionados por el servicio de mantenimiento.
- Los requisitos relativos a estos fallos son:
  - Requisito C1: El tiempo medio para inicializar la LAN debe ser menor de 15 ms.
  - Requisito C2: El tiempo medio entre inicializaciones de la LAN debe ser superior a 60 segundos.
  - Requisito C3: El tiempo máximo de detección y reparación del fallo en una LAN debe ser inferior a 1 hora.
  - Requisito C4: El tiempo máximo entre fallos debe ser superior a 72 horas.

Notas:

3. *Availability:* Two fault modes were considered significant. The first was the time lost due to the network reinitializations, and the second was time lost due to permanent failures requiring field service calls. The requirements for frequency and duration of these fault modes were specified as follows:

(a) The mean time to initialize the LAN must be less than 15 milliseconds.

(b) The mean time between LAN initializations must be at least 1 minute.

(c) The mean time to repair a LAN must be less than 1 hour. (LAN partitions may be operational during this period.)

(d) The mean time between LAN partitioning must be at least half a week.