

# Capítulo 1

## Caracterización de los Sistemas Distribuidos



### *Sistemas Distribuidos*

Universidad Nacional de Asunción  
Facultad Politécnica  
Ingeniería Informática

***Profesor: Ing. Fernando Mancía***

## Antecedentes

---

El avance de las TIC's produjo que los sistemas de información evolucionen rápidamente en el contexto del trabajo coordinado separados geográficamente.

La utilización de la Programación Orientada a Objetos, cambió la forma diseñar en implementar Sistemas Distribuidos en las últimas décadas. Se crearon nuevos modelos a partir de la P.O.O y nuevas tecnologías.

La utilización de Internet en casi todas las organizaciones de todo tipo y tamaño, y la disponibilidad de información, causan la emergente demanda de un gran número de tecnologías para implementar Sistemas Distribuidos. Esto da origen a lo que se denomina: "Service Oriented Computing".

# Sistema Distribuido

---

*Un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora*

Tanenbaum

*Un sistema distribuido es aquel en el que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes*

Coulouris

## Recursos

---

Compartir recursos es uno de los motivos principales para construir sistemas distribuidos.

Los recursos son administrados por servidores y accedidos por clientes. O encapsulados como objetos y accedidos por otros objetos clientes.

Recursos pueden ser componentes hardware como discos, impresoras, o entidades software como archivos, bases de datos y objetos. Secuencia de imágenes de una cámara de video digital, conexión de audio que representa una llamada de un teléfono móvil.

## Características de los Sistemas Distribuidos

---

***Concurrencia:*** en una red de computadores, la ejecución de programas concurrentes es la norma.

La coordinación de programas que comparten recursos y se ejecutan de forma concurrente es un tema importante y recurrente.

***Inexistencia de reloj global:*** la cooperación entre programas se realiza mediante el intercambio de mensajes. La coordinación depende de una idea compartida del instante en el que ocurren las “acciones” de los programas.

Existen límites de precisión en los computadores de una red para sincronizar sus relojes. No existe una única noción global del tiempo correcto.

**Fallos independientes:** todos los sistemas pueden fallar. Los S.D pueden fallar de formas diferentes:

- ❑ Fallos en la red. Producen aislamiento de los computadores conectados a él, pero eso no significa que detengan su ejecución. Pueden no ser capaces de detectar cuando la red ha fallado o está excesivamente lenta.
- ❑ Parada de un computador o terminación inesperada de un programa en alguna parte del sistema (crash). No se da a conocer inmediatamente a lo demás componentes con los que se comunica.

## Ventajas de los Sistemas Distribuidos

---

Elemento	Descripción
Economía	Los microprocesadores ofrecen mejor proporción precio/rendimiento que los mainframes
Velocidad	Un sistema distribuido puede tener mayor poder de cómputo que un mainframe
Distribución inherente	Algunas aplicaciones utilizan máquinas que están separadas a cierta distancia
Confiabilidad	Si una máquina se descompone, el sistema puede sobrevivir como un todo
Crecimiento por incrementos	Se puede añadir poder de cómputo en pequeños incrementos



## Ejemplos de Sistemas Distribuidos INTERNET

---

Programas ejecutándose en los computadores conectados a Internet interactúan mediante paso de mensajes, empleando un medio común de comunicación. El diseño y la construcción de los mecanismos de comunicación Internet (los protocolos Internet) es una realización técnica fundamental, que permite que un programa que se está ejecutando en cualquier parte dirija mensajes a programas en cualquier otra parte.

Internet es también un sistema distribuido muy grande. Permite a los usuarios, donde quiera que estén, hacer uso de servicios como el World Wide Web, el correo electrónico, y la transferencia de archivos.



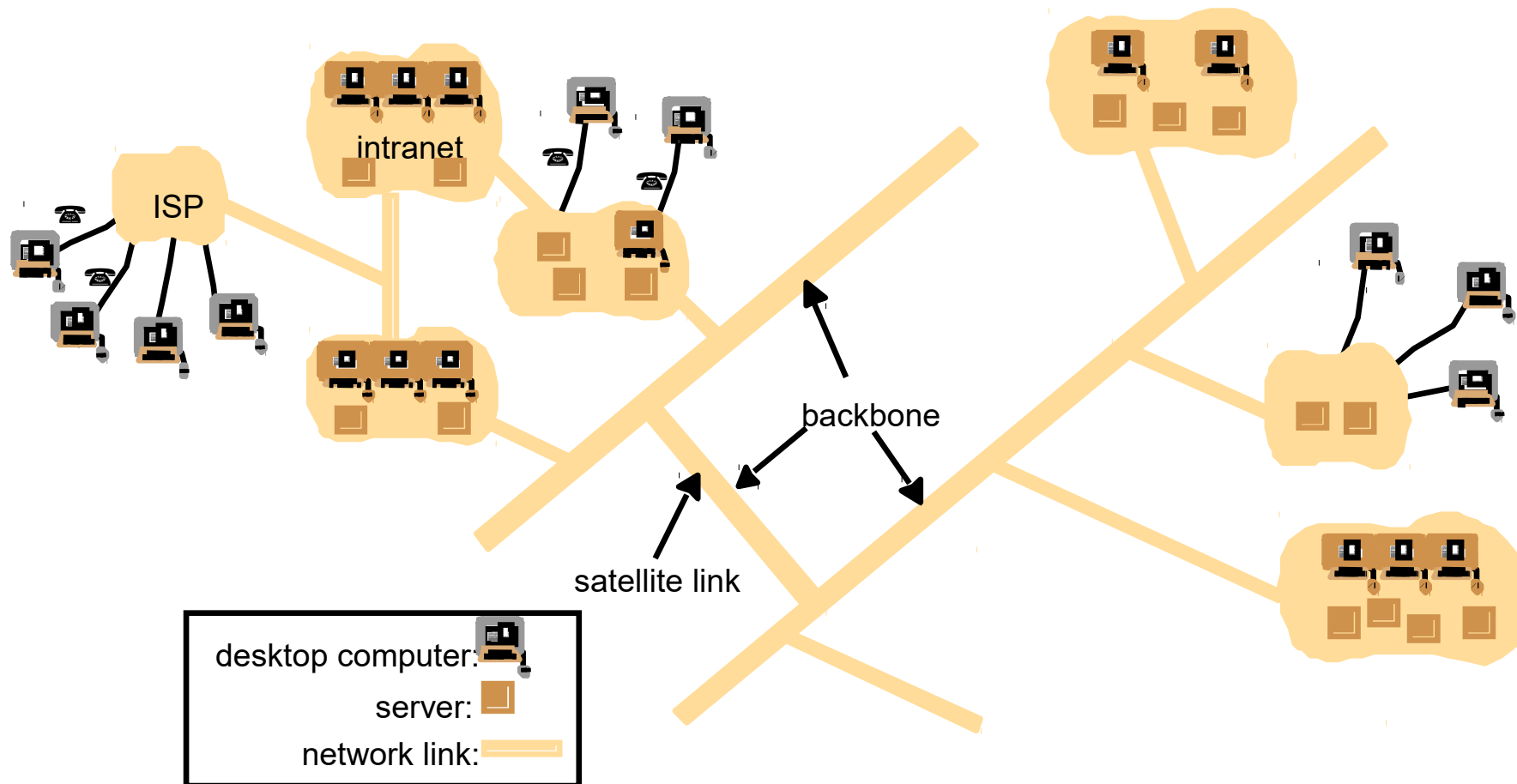
## Ejemplos de Sistemas Distribuidos INTERNET

---

El conjunto de servicios es abierto, puede ser extendido por la adición de servidores y nuevos tipos de servicios.

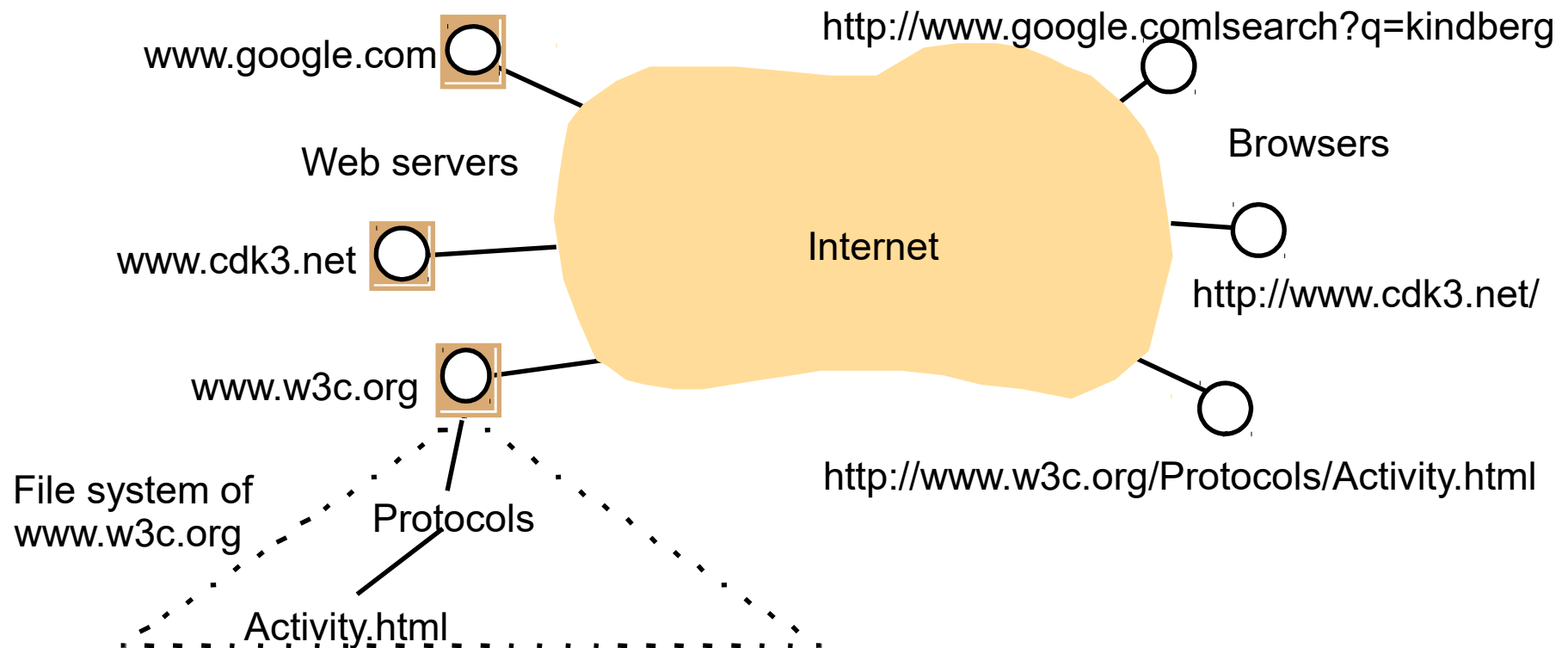
La figura nos muestra una colección de intranets, subredes gestionadas por compañías y otras organizaciones. Los proveedores de servicios de Internet (ISP's) son empresas que proporcionan enlaces de módem y otros tipos de conexión a usuarios individuales y pequeñas organizaciones, permitiéndolas el acceso a servicios desde cualquier parte de Internet, así como proporcionando servicios como correo electrónico y páginas web. Las intranets están enlazadas conjuntamente por conexiones troncales (*backbones*). Una conexión o red troncal es un enlace de red con una gran capacidad de transmisión, que puede emplear conexiones de satélite, cables de fibra óptica y otros circuitos de gran ancho de banda.

## Una topología típica de Internet



## Servidores y visualizadores web

---



## Ejemplos de Sistemas Distribuidos

### BÚSCADORES WEB

---

El número global de búsquedas ha aumentado a más de 10 mil millones por mes calendario.

La tarea de un motor de búsqueda web es *indexar el contenido* completo de la World Wide Web, abarcando una amplia gama de estilos de información incluyendo páginas web, fuentes multimedia y libros (escaneados).

Esta es una tarea muy compleja, ya que las estimaciones actuales indican que la Web consiste en más de **63 mil millones de páginas y un trillón de direcciones web únicas**.

La mayoría de los motores de búsqueda analizan todo el contenido de la web y luego realizan sofisticados procesamientos en esta enorme base de datos, esta tarea representa un desafío importante para el diseño de sistemas distribuidos.



Google, líder del mercado en tecnología de búsqueda web, ha puesto un esfuerzo significativo en el diseño de una sofisticada infraestructura de sistemas distribuidos para apoyar la búsqueda (y de hecho otras aplicaciones y servicios de Google, como Google Earth).

Esto representa una de las mayores y más complejas instalaciones de sistemas distribuidos en la historia de la informática y por lo tanto requiere un examen detallado. Los puntos destacados de esta infraestructura incluyen:



## Ejemplos de Sistemas Distribuidos

### BÚSCADORES WEB

---

- **Infraestructura física** subyacente que consta de un gran número de ordenadores en red ubicados en centros de datos de todo el mundo;
- **Sistema de archivos distribuido** diseñado para soportar archivos muy grandes y muy optimizado para el estilo de uso requerido por la búsqueda y otras aplicaciones de Google (especialmente lectura de archivos a tasas altas y sostenidas)
- **Sistema de almacenamiento distribuido** estructurado asociado que ofrece acceso rápido a conjuntos de datos grandes;
- Servicio de **bloqueo** que ofrece funciones de sistema distribuidas tales como **bloqueo distribuido y acuerdo**;
- Modelo de **programación** que soporta la gestión de cálculos paralelos y distribuidos muy grandes a través de la infraestructura física subyacente.

# INTRANET

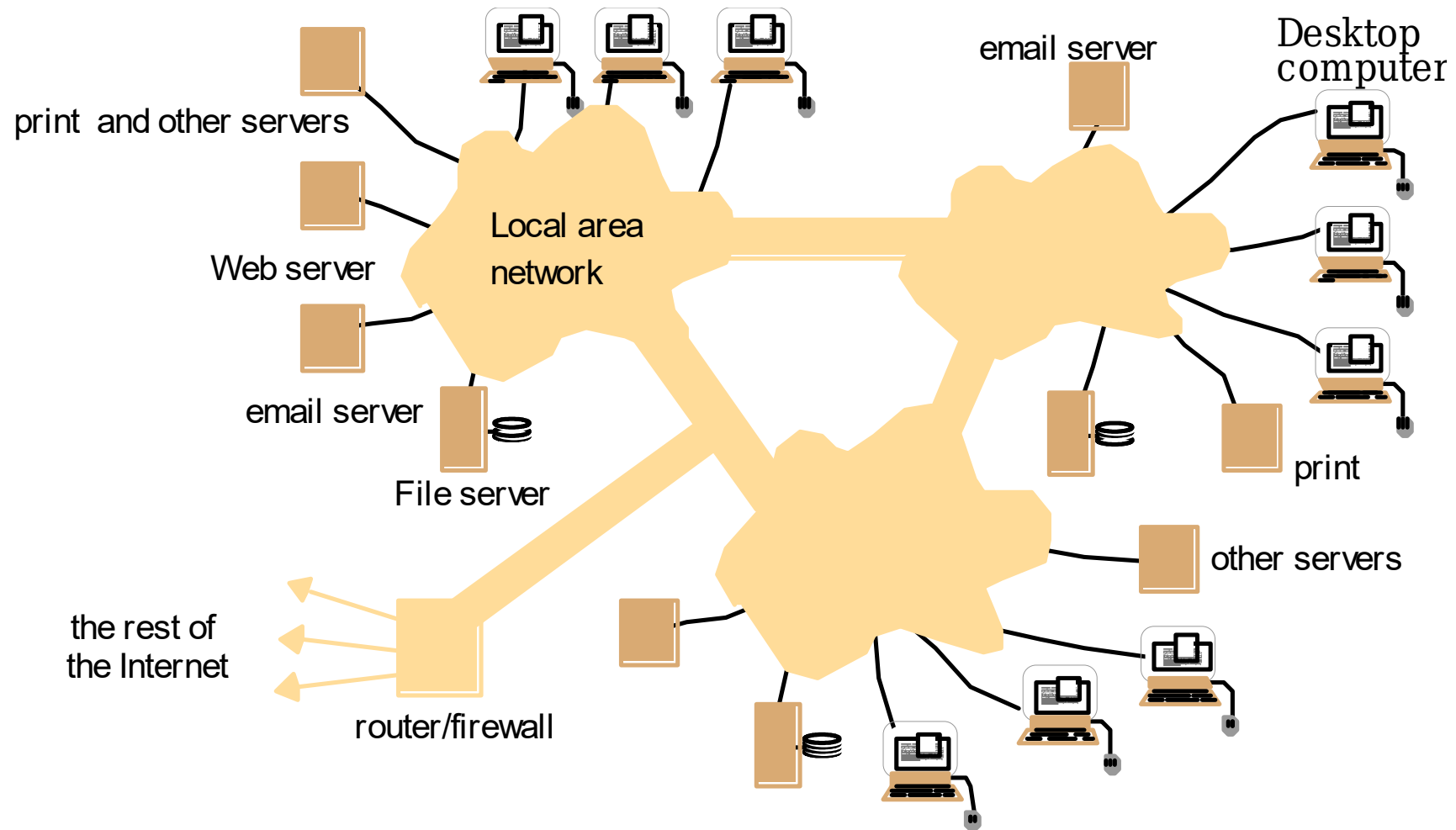
---

Una intranet es una porción de Internet que es, administrada separadamente y que tiene un límite que puede ser configurado para hacer cumplir políticas de seguridad local.

La Figura 1.2 muestra una intranet típica. Está compuesta de varias redes de área local (LAN s) enlazadas por conexiones backbone. La configuración de red de una intranet particular es responsabilidad de la organización que la administra y puede variar ampliamente, desde una LAN en un único sitio a un conjunto de LANs conectadas perteneciendo a ramas de la empresa u otra organización en diferentes países. Una intranet está conectada a Internet por medio de un encaminador (*router*), lo que permite a los usuarios hacer uso de servicios de otro sitio como el Web o el correo electrónico.



## Una Intranet típica



## Tendencias

---

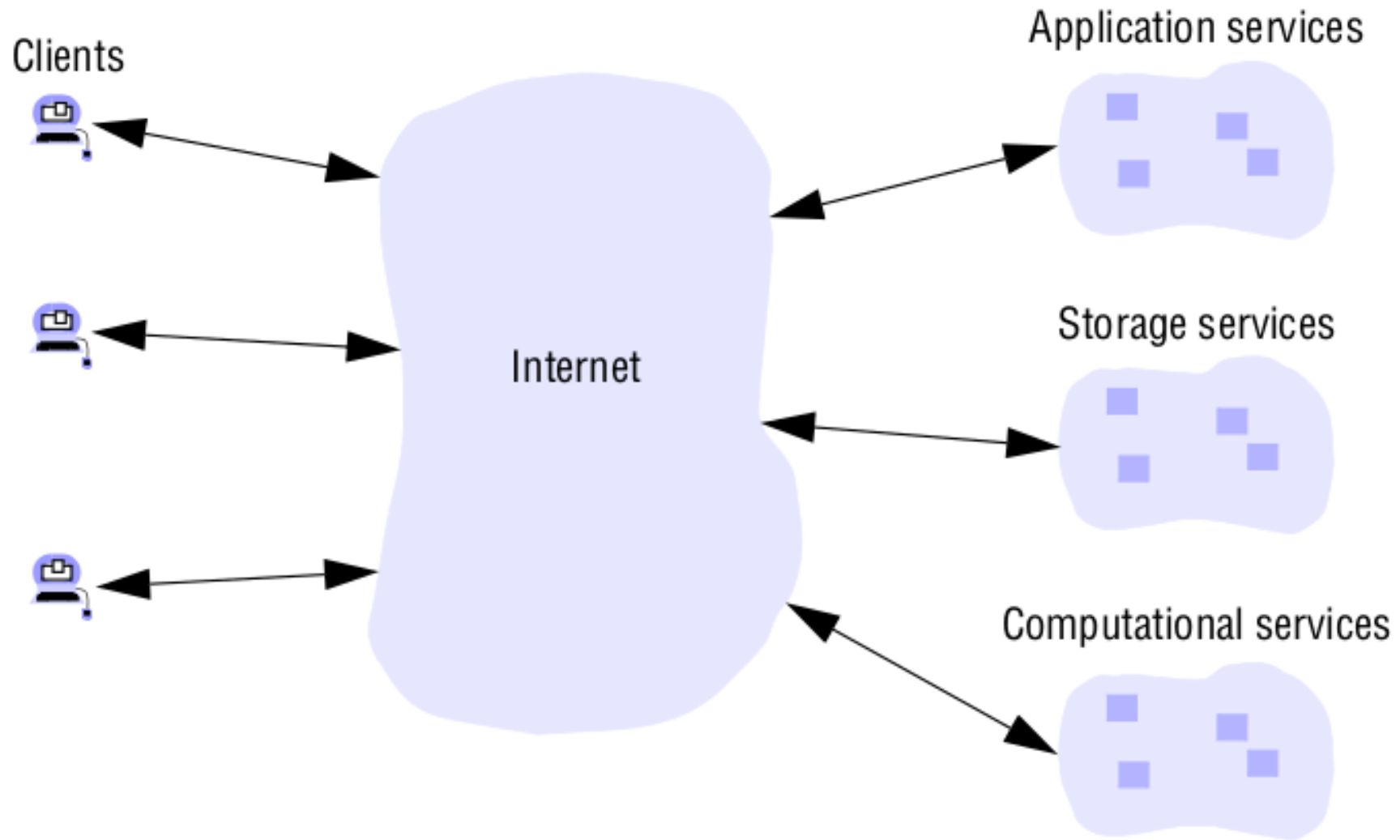
### **Cloud Computing. Computación en la nube.**

El término se utiliza para capturar esta visión de la computación como una utilidad.

Una nube se define como un conjunto de servicios de aplicaciones y almacenamiento basados en Internet suficientes para soportar la mayoría de las necesidades de los usuarios, lo que les permite activar mayor o totalmente el almacenamiento de datos local y el software de aplicación.

El término también promueve una visión de todo como un servicio, desde la infraestructura física o virtual hasta el software, a menudo pagado en base al uso en lugar de comprado. Tenga en cuenta que la computación en la nube reduce los requisitos de los dispositivos de los usuarios, permitiendo que dispositivos de escritorio o portátiles muy sencillos accedan a una amplia variedad de recursos y servicios.

## Cloud computing



## Tendencias

---

### **Computación móvil y ubicua.**

Los avances tecnológicos en la miniaturización de dispositivos, redes inalámbricas y redes móviles han llevado cada vez más a la integración de dispositivos de computación pequeños y portátiles en sistemas distribuidos.

Estos dispositivos incluyen:

- ☐ Computadores portátiles.
- ☐ Dispositivos de mano (handheld), entre los que se incluyen asistentes digitales personales (PDA), teléfonos móviles, buscapersonas y videocámaras o cámaras digitales.
- ☐ Dispositivos que se pueden llevar puestos, como relojes inteligentes.
- ☐ Dispositivos insertados en aparatos, como lavadoras, sistemas de alta fidelidad, automóviles y frigoríficos

# COMPUTACIÓN MÓVIL Y UBICUA

---

Facilidad de transporte y capacidad para conectarse adecuadamente a redes en diferentes lugares, hace posible la computación móvil.

Computación móvil: se denomina a la realización de tareas de cómputo mientras el usuario está en movimiento.

Los usuarios que están fuera de su hogar intranet (la intranet de su trabajo o su casa) disponen de la posibilidad de acceder a los recursos mediante los dispositivos que llevan con ellos. Pueden continuar accediendo a Internet; pueden acceder a los recursos de su intranet; y se está incrementando la posibilidad de que utilicen recursos, como impresoras, que están suficientemente próximos a donde se encuentren. Esto último se conoce como computación independiente de posición .

## COMPUTACIÓN MÓVIL Y UBICUA

---

**Computación ubicua:** utilización concertada de dispositivos de computación pequeños y baratos que están presentes en los entornos físicos de los usuarios, incluyendo la casa, la oficina y otros.

El término ubicuo está pensado para sugerir que los pequeños dispositivos llegarán a estar tan extendidos en los objetos de cada día que apenas nos daremos cuenta de ellos. O sea, su comportamiento computacional estará ligado con su función física de forma íntima y transparente.

# COMPUTACIÓN MÓVIL Y UBICUA

---

**Computación ubicua:** la integración de la computación en el entorno de la persona, de forma que los ordenadores no se perciban como objetos diferenciados.

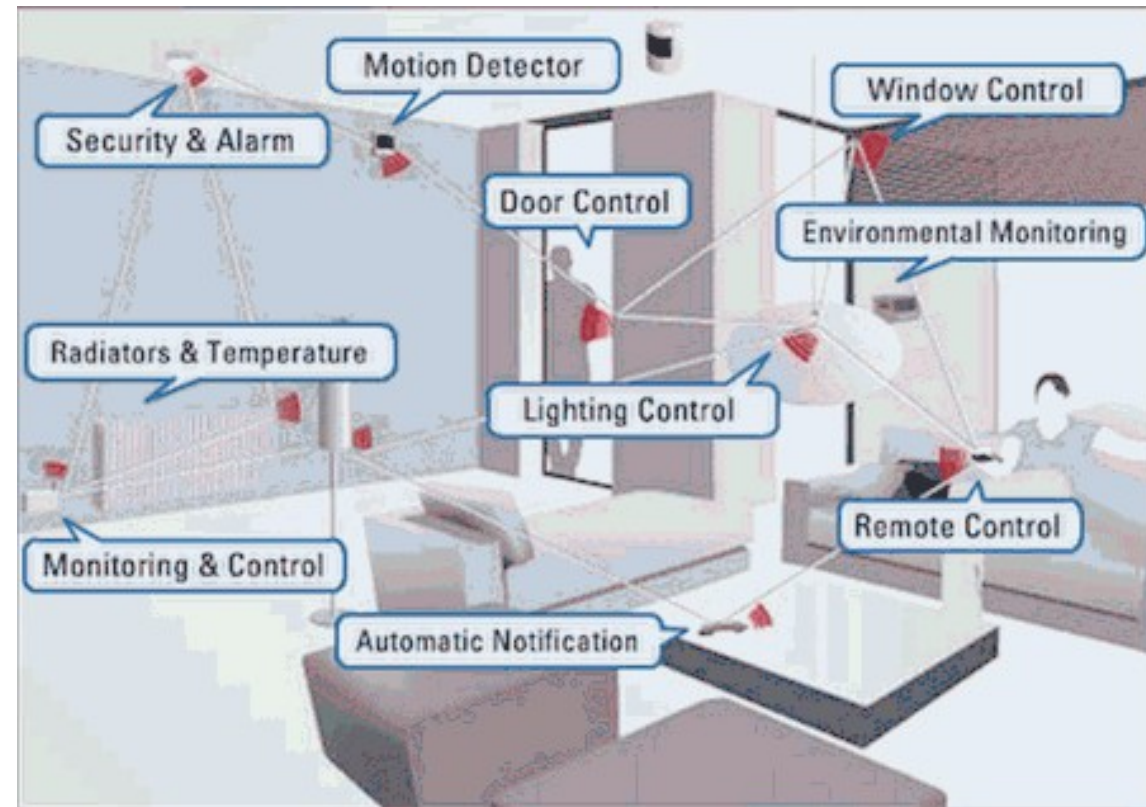
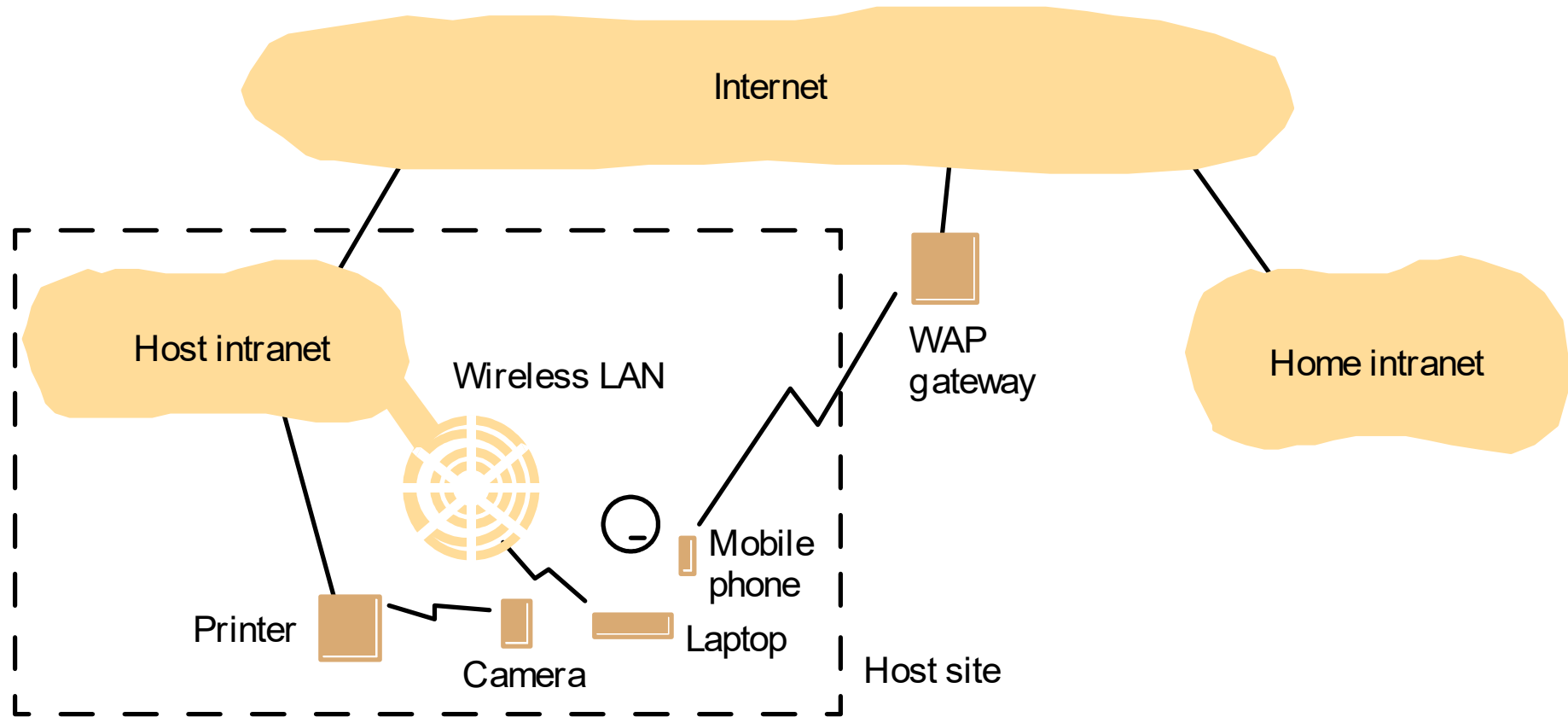




Figura 1.3  
Dispositivos portátiles y de mano en un sistema distribuido



# Desafios de los Sistemas Distribuidos

## HETEROGENEIDAD

---

Internet permite que los usuarios accedan a servicios y ejecuten aplicaciones sobre un conjunto heterogéneo de redes y computadores. Esta heterogeneidad (es decir, variedad y diferencia) se aplica a todos los siguientes elementos:

- . Redes.
- . Hardware de computadores.
- . Sistemas operativos.
- . Lenguajes de programación.
- . Implementaciones de diferentes desarrolladores.

## Desafios de los Sistemas Distribuidos

### HETEROGENEIDAD (2)

---

**Middleware:** estrato software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación. CORBA, es un ejemplo de ello.

Generalmente se implementa sobre protocolos de Internet, enmascarando éstos la diversidad de redes existentes. Aun así cualquier middleware trata con las diferencias de sistema operativo y hardware.

Proporciona un modelo computacional uniforme al alcance de los programadores de servidores y aplicaciones distribuidas. Los posibles modelos incluyen invocación sobre objetos remotos, notificación de eventos remotos, acceso remoto mediante SQL y procesamiento distribuido de transacciones.

## Desafios de los Sistemas Distribuidos

### HETEROGENEIDAD (3)

---

**Código móvil:** código que puede ser enviado desde un computador a otro y ejecutarse en éste. Ejemplo: applets de Java.

El conjunto de instrucciones de un computador depende del hardware, el código de nivel de máquina adecuado para correr en un computador no es adecuado para ejecutarse en otro tipo. Ejemplo, ejecutables para PC no pueden ser ejecutados en un computador Mac.

La aproximación de máquina virtual provee un modo de crear código ejecutable sobre cualquier hardware: el compilador de un lenguaje concreto generará código para una máquina virtual

## Desafios de los Sistemas Distribuidos

### EXTENSIBILIDAD

---

La extensibilidad de un sistema de cómputo es la característica que determina si el sistema puede ser extendido y reimplementado en diversos aspectos. La extensibilidad de los sistemas distribuidos se determina en primer lugar por el grado en el cual se pueden añadir nuevos servicios de compartición de recursos y ponerlos a disposición para el uso por una variedad de programas cliente.

- . Los sistemas abiertos se caracterizan porque sus interfaces están publicadas.
- . Los sistemas distribuidos abiertos se basan en la providencia de un mecanismo de comunicación uniforme e interfaces públicas para acceder a recursos compartidos.
- . Los sistemas distribuidos abiertos pueden construirse con hardware y software heterogéneo, posiblemente de diferentes proveedores. Sin embargo, la conformidad con el estándar publicado de cada componente debe contrastarse y verificarse cuidadosamente si se desea que el sistema trabaje correctamente.

## Desafios de los Sistemas Distribuidos

### SEGURIDAD

---

Entre los recursos de información que se ofrecen y se mantienen en los sistemas distribuidos, muchos tienen un alto valor intrínseco para sus usuarios. Por esto su seguridad es de considerable importancia.

La seguridad de los recursos de información tiene tres componentes: confidencialidad (protección contra el descubrimiento por individuos no autorizados); integridad (protección contra la alteración o corrupción); y disponibilidad (protección contra interferencia con los procedimientos de acceso a los recursos).

## Desafios de los Sistemas Distribuidos

### ESCALABILIDAD

---

Los sistemas distribuidos operan efectiva y eficientemente en muchas escalas diferentes, desde pequeñas intranets a Internet. Se dice que un sistema es *escalable* si conserva su efectividad cuando ocurre un incremento significativo en el número de recursos y el número de usuarios.

El diseño de los sistemas distribuidos escalables presenta los siguientes retos:

*Control del costo de los recursos físicos:* según crece la demanda de un recurso, debiera ser posible extender el sistema, a un coste razonable, para satisfacerla.

*Control de las pérdidas de prestaciones:* considere la administración de un conjunto de datos cuyo tamaño es proporcional al número de usuarios o recursos del sistema



## Desafios de los Sistemas Distribuidos

### ESCALABILIDAD

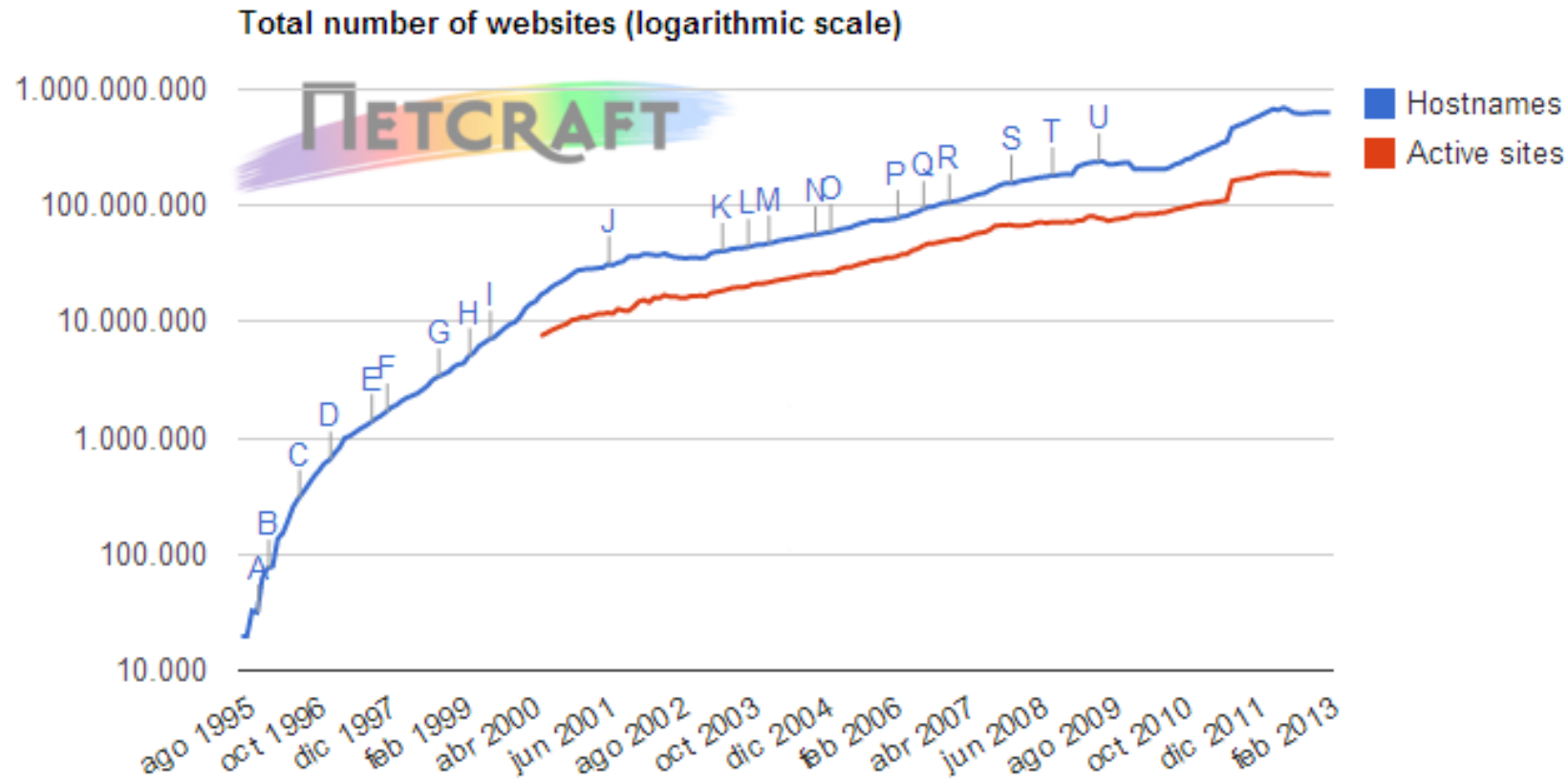
---

*Prevención de desbordamiento de recursos software:* un ejemplo de pérdida de escalabilidad se muestra en el tipo de número usado para las direcciones Internet.

*Evitar cuellos de botella de prestaciones:* en general, para evitar cuellos de botella de prestaciones, los algoritmos deberían ser descentralizados.

# ESCALABILIDAD

## Estadísticas de número de sitios web



***Fuente:** netcraft.com*

## Desafios de los Sistemas Distribuidos

### TRATAMIENTO DE FALLOS

---

***Detección de fallos:*** algunos fallos son detectables. Por ejemplo, se pueden utilizar sumas de comprobación (*checksums*) para detectar datos corruptos en un mensaje o un archivo.

***Enmascaramiento de fallos:*** algunos fallos que han sido detectados pueden ocultarse o atenuarse.

***Tolerancia de fallos:*** la mayoría de los servicios en Internet exhiben fallos; es posible que no sea práctico para ellos pretender detectar y ocultar todos los fallos que pudieran aparecer en una red tan grande y con tantos componentes. Sus clientes pueden diseñarse para tolerar ciertos fallos, lo que implica que también los usuarios tendrán que tolerarlos generalmente. Por ejemplo, cuando un visualizador web no puede contactar con un servidor web no hará que el cliente tenga que esperar indefinidamente mientras hace sucesivos intentos; informará al usuario del problema, dándole la libertad de intentarlo más tarde.

***Recuperación frente a fallos:*** la recuperación implica el diseño de software en el que, tras una caída del servidor, el estado de los datos pueda reponerse o retractarse (*roll back*) a una situación anterior.

***Redundancia:*** puede lograrse que los servicios toleren fallos mediante el empleo redundante de componentes.

## CONCURRENCIA

---

Tanto los servicios como las aplicaciones proporcionan recursos que pueden compartirse entre los clientes en un sistema distribuido. Existe por lo tanto una posibilidad de que varios clientes intenten acceder a un recurso compartido a la vez.

El proceso que administra un recurso compartido puede atender las peticiones de cliente una por una en cada momento, pero esta aproximación limita el ritmo de producción del sistema (*throughput*). Por esto los servicios y aplicaciones permiten, usualmente, procesar concurrentemente múltiples peticiones de los clientes. Más concretamente, suponga que cada recurso se encapsula en un objeto y que las invocaciones se ejecutan en hilos de ejecución concurrentes (*threads*).

## Desafíos de los Sistemas Distribuidos

### TRANSPARENCIA

---

*Transparencia de acceso* que permite acceder a los recursos locales y remotos empleando operaciones idénticas.

*Transparencia de ubicación* que permite acceder a los recursos sin conocer su localización.

*Transparencia de concurrencia* que permite que varios procesos operen concurrentemente sobre recursos compartidos sin interferencia mutua.

*Transparencia de replicación* que permite utilizar múltiples ejemplares de cada recurso para aumentar la fiabilidad y las prestaciones sin que los usuarios y los programadores de aplicaciones necesiten su conocimiento.

## Desafíos de los Sistemas Distribuidos

### TRANSPARENCIA

---

***Transparencia frente a fallos*** que permite ocultar los fallos, dejando que los usuarios y programas de aplicación completen sus tareas a pesar de fallos del hardware o de los componentes software.

***Transparencia de movilidad*** que permite la reubicación de recursos y clientes en un sistema sin afectar la operación de los usuarios y los programas.

***Transparencia de prestaciones:*** que permite reconfigurar el sistema para mejorar las prestaciones según varía su carga.

***Transparencia al escalado:*** que permite al sistema y a las aplicaciones expandirse en tamaño sin cambiar la estructura del sistema o los algoritmos de aplicación.