

Sistemas de Operación III

(CI-4822)

Tema I: Introducción

Prof. Yudith Cardinale
Sept-dic 2011

Motivación

- La demanda de cómputo ha estado siempre por encima del *status quo* tecnológico
- Respuestas obvias
 - Diversos elementos de cómputo trabajando en armonía para resolver un único problema
 - Necesidad de tener recursos compartidos entre los diferentes elementos de computación

Evolución de los SD

- Desde el punto de vista del soporte de software
 - Inicialmente no existían Sistemas de Operación
 - Sistemas de operación monousuarios, monotareas
 - Sistemas de operación monousuarios, multitareas
 - Sistemas de operación multiusuarios, multitareas
 - Sistemas batch, sistemas de operación tiempo compartido
 - Sistemas de operación de redes, sistemas de operación paralelos, sistemas de operación distribuidos
 - Middlewares

Evolución de los SD

- Desde el punto de vista del soporte de la infraestructura
 - Arquitecturas mainframes (procesamiento batch) con terminales tontos (sin capacidades de procesamiento)
 - Minicomputadores alojados en departamentos pero persistía el modelo de computación maestro-esclavo
 - El modelo de microcomputador cambió significativamente, la potencia de procesamiento se “mudó”, primero a desktops, luego a laptops y ahora a dispositivos de bolsillo (celulares, PDAs, etc.)
 - Actualmente el poder de cómputo vuelve a un esquema de procesamiento centralizado con redes de servidores e Internet
 - Aplicaciones de información y clientes livianos realizan requerimiento
 - Los servicios Web continúan su evolución y desarrollo.

Clasificación de las arquitecturas de computación

- Simplificando el cómputo como “instrucciones operando sobre datos”, todas las arquitecturas de computación caen en la siguiente clasificación:
 - **SISD** (**S**ingle **I**nstructions **S**ingle **D**ata): computadores personales, uniprocesadores
 - **MISD** (**M**ultiple **I**nstructions **S**ingle **D**ata): no existe arquitectura bajo esta clasificación
 - **SIMD** (**S**ingle **I**nstructions **M**ultiple **D**ata): arquitecturas paralelas
 - **MIMD** (**M**ultiple **I**nstructions **M**ultiple **D**ata): arquitecturas paralelas, redes y sistemas distribuidos

Arquitecturas SIMD

- Originalmente se observó que si se tiene
for (i=0; i< 1000;i++)
 a[i]=b[i]*c[i];
se repite la misma instrucción sobre datos diferentes
- De allí nacen los procesadores vectoriales
- Los procesadores vectoriales toman una única instrucción de vectores que opera simultáneamente sobre una serie de datos almacenados en arreglos
- Arquitectura Cray-1 (1976) usó 11 estados de pipelines que podían ejecutarse concurrentemente sobre datos
- CDC Cyber 205 (4 unidades de punto flotante idénticas), IBM 3090, ILLIAC IV, GF11 y Connection Machine

Arquitecturas SIMD

- Ventajas y desventajas
 - Costo excesivamente alto
 - Se requiere de compiladores especializados
 - La programación debe ser hecha con lenguajes específicos
 - Aceleración del cálculo sólo para un tipo específico de problemas
 - Es difícil hacer uso simultáneo de toda la potencia de cómputo
 - Para problemas específicos, se obtienen velocidades de cómputo considerables.

Arquitecturas MIMD

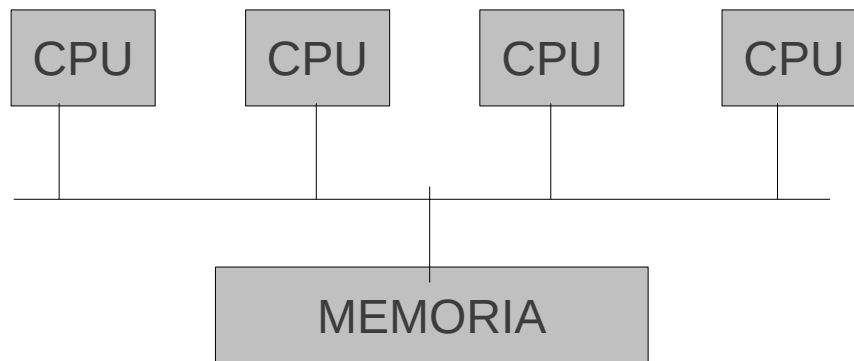
- Para reducir el precio se pueden conectar múltiples procesadores, cada uno marchando con su propio ritmo
- Muchos problemas de la vida real se pueden dividir en un gran número de pequeñas tareas que se pueden resolver individualmente
- La solución se puede recolectar a través de la comunicación entre los diferentes elementos de cómputo
 - Topologías de conexión se hacen más efectivas en aceleración del cómputo
 - Se usan protocolos de conexión específicos: bus del sistema, Myrinet, Infiniband, ...
- Sin embargo, para muchos problemas se requiere aún formas de compartimiento de recursos.

MIMD con Memoria Privada (distribuida)

- Diferentes elementos de computación con memoria privada se comunican a través de una red de conexión
- La comunicación se realiza a través de pase de mensajes
- Clusters de estaciones de trabajo (granjas) comunicándose a través de una LAN
 - Comunicación lenta
 - Comunicación no frecuente conduce a mejor desempeño
 - La escalabilidad es inherente a la arquitectura LAN
 - Componentes *commodities* muy baratos
- Primer computador de memoria distribuida fue la Cosmic Cube con 64 nodos, cada nodo tenía una conexión directa punto-a-punto a six nodos
- Desarrollos posteriores fueron hieprcubos, mallas y máquinas de flujos de datos

MIMD con Memoria Compartida

- Cuando los datos requieren ser accedidos frecuentemente por varios procesos, una alternativa común para obtener mejor desempeño es usar arquitecturas de memoria compartida
- Los elementos de cómputo comparten el bus de memoria
- Son conocidos como computadores SMP (**S**ymmetric **M**ulti-**P**rocessor)
- Hoy día se pueden adquirir por una fracción del costo de las arquitecturas paralelas



Arquitecturas MIMD

Ventajas

- Compartimiento
- Extensible
- Autonomía local
- Desempeño mejorado
- Confiabilidad y disponibilidad mejoradas
- Potencial reducción de costos

Desventajas

- Retardo de red
- Complejidad
- Seguridad
- Características específicas del SO requeridas

Actualidad MIMD

- Clusters Beowulf comenzaron en 1993 usando plataformas *commodities* y Ethernet tradicional.
- LANs con velocidades de Gigabits
- Máquinas SMP relativamente baratas
- Internet: millones de elementos MIMD conectados para compartir recursos
- Modelo cliente-servidor, Peer-to-Peer, Grid y Cloud computing
- Computación paralela masiva: Top 500
- Máquinas multiprocesadores se hacen populares
- Muchos APIs disponibles (PVM, MPI, RMI, ...)

Atributos de los SD

- El grado de distribución de un sistema está determinado por las respuestas a las siguientes preguntas:
 - Dónde se realiza el procesamiento
 - Cómo están interconectados los procesadores y los demás dispositivos
 - Dónde se almacena la información
 - Qué reglas y estándares son usados
 - Dónde están los usuarios

Dónde se realiza el procesamiento

- **Procesamiento distribuido** es la habilidad de que varios procesadores interconectados operen simultáneamente, típicamente para procesar una aplicación sobre más de un computador a la vez
- El objetivo es mover el procesamiento apropiado lo más cercano posible al usuario y dejar que otras máquinas realicen el trabajo que mejor saben hacer
- Permitir interoperabilidad o capacidad para que computadores con sistemas de operación diferentes en redes diferentes trabajen conjuntamente para resolver tareas

Cómo están interconectados los procesadores

- **Conectividad entre procesadores** significa que cada procesador en un sistema distribuido puede enviar datos y mensajes a los demás procesadores a través de enlaces de comunicación electrónicos
- Es deseable tener al menos dos caminos independientes entre dos nodos para proveer enrutamiento automático

Dónde se almacena la información

- **Bases de datos distribuidas** pueden:
 - Dividir una base de datos y distribuir las porciones en el sistema sin replicar los datos
 - Los usuarios no requieren conocer dónde se almacenan las piezas de datos para accederlas dado que el sistema es quien conoce esta información
 - Almacenar los mismos datos en diferentes localizaciones (un sitio contiene el archivo maestro)
 - La sincronización de datos es un problema significativo

Qué reglas y estándares son usados

- **Las reglas del sistema** gobiernan la comunicación entre los nodos, la seguridad, la accesibilidad de los datos, la transferencia de datos y programas y los procedimientos de operación comunes.
- Desde 1990, el concepto de sistemas abiertos dicta el uso de estándares abiertos. Sistemas abiertos e interfaces que permiten que diferentes productos inter-operen a través de redes, sistemas de operación y bases de datos heterogéneas.
- Las APIs definen la forma de presentar datos a otros componentes del sistemas. Hacen mucho más fácil la creación de sistemas distribuidos.

Dónde están los usuarios

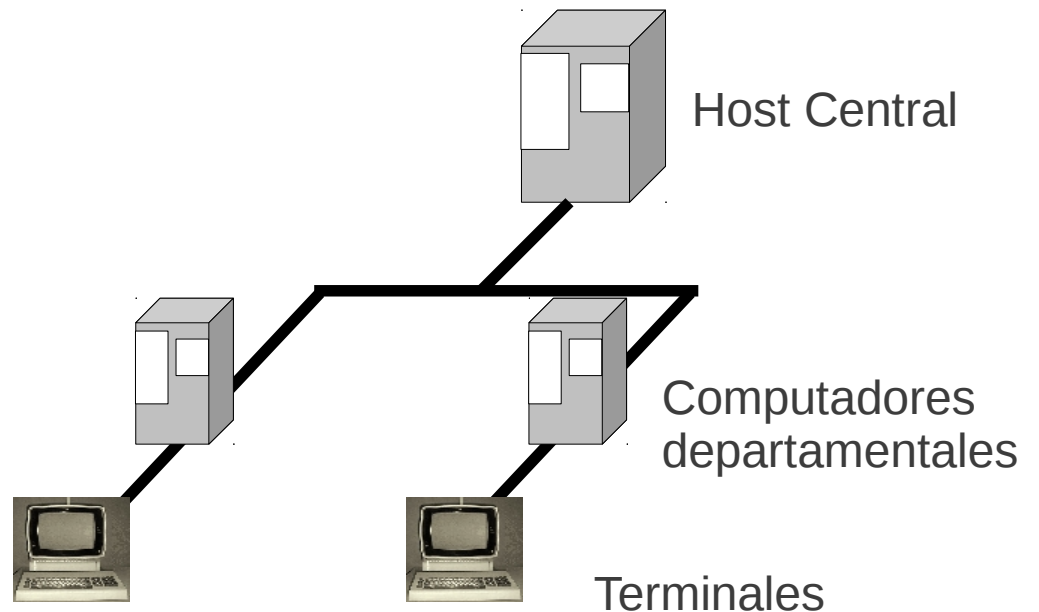
- Los usuarios pueden acceder el sistema desde cualquier punto de conexión de manera transparente
- Desarrollo de clientes con interfaces que apoyan a los usuarios a usar el sistema
- Clientes livianos para acceder el sistema desde dispositivos móviles

Tipos de Sistemas Distribuidos

- Jerarquía basada en hosts
- Sistemas *stand-alone* descentralizados
- Sistemas Peer-to-Peer basados en LAN
- Sistemas híbridos
- Sistemas cliente-servidor
- Computación basada en Internet
- Servicios Web

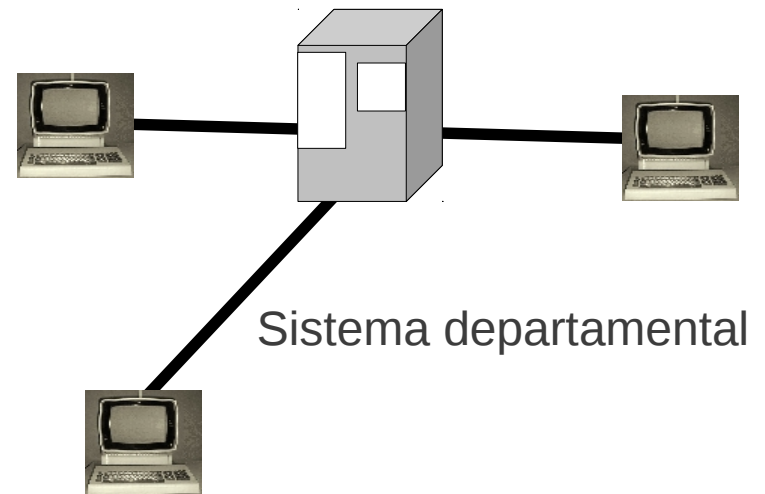
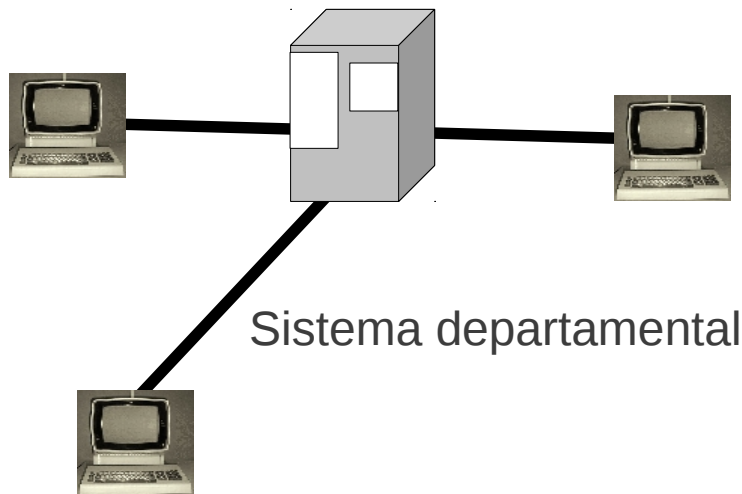
Jerarquía basada en hosts

- Primer sistema distribuido de procesamiento de datos. Un computador central controlando componentes; los terminales son los puntos de acceso al sistema
- Relación maestro-esclavo: un mainframe central en la raíz, PCs en las hojas y minicomputadores en el medio
- Los datos podían estar almacenados en cualquier nivel.



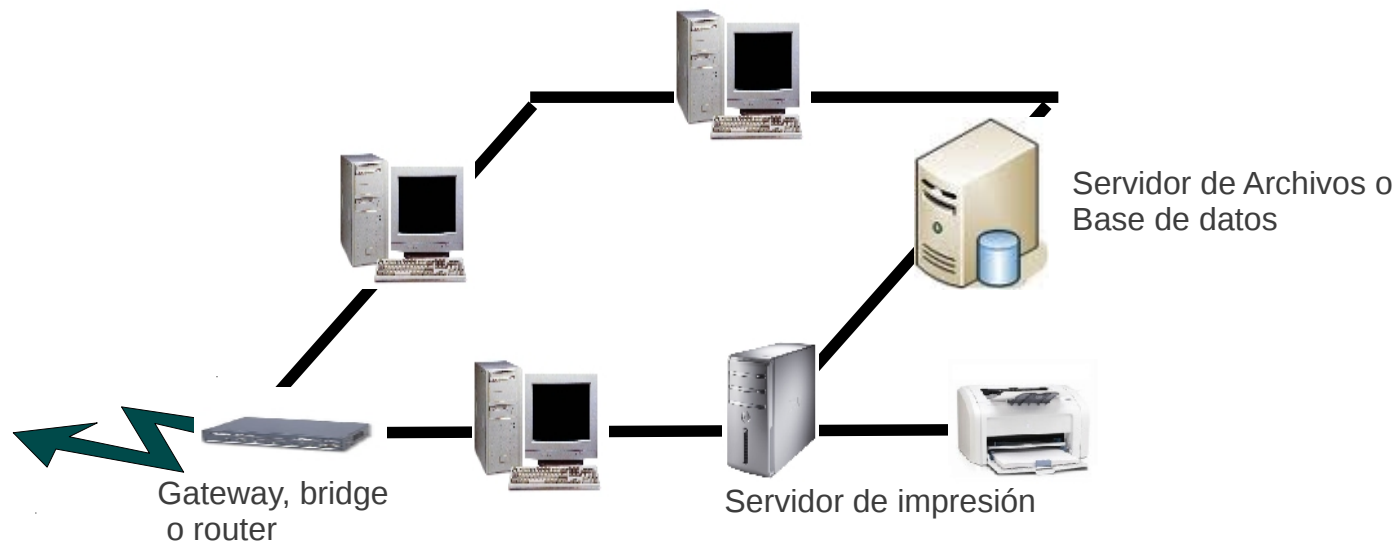
Sistemas *stand-alone* descentralizados

- Descentralizados pero no vistos exactamente como sistemas distribuidos
- Los departamentos tenían sus propios minicomputadores si la intención de interconectarlo con los demás



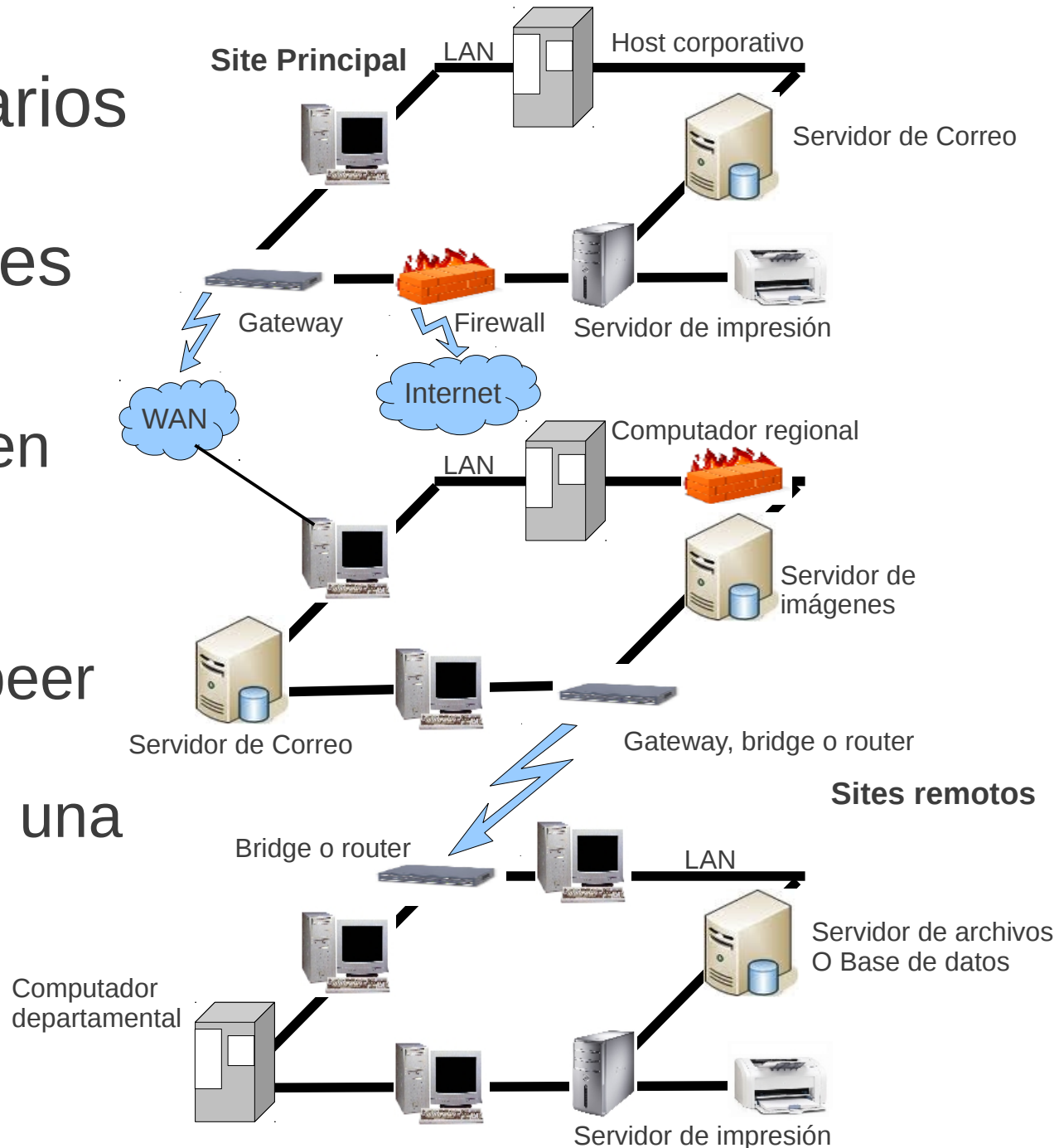
Sistemas peer-to-peer basados en LAN

- No hay jerarquía de computadores
- Comunicaciones peer-to-peer
 - LANs interconectadas a través de un hub central (no hay jerarquía de comunicación)
 - No hay un computador central



Sistemas híbridos

- Combinación de varios tipos de sistemas distribuidos via redes (e.g., WAN):
 - Jerarquía basada en host (basada en Mainframes)
 - Sistemas peer-to-peer basados en LAN (departamentos de una corporación)



Sistemas híbridos

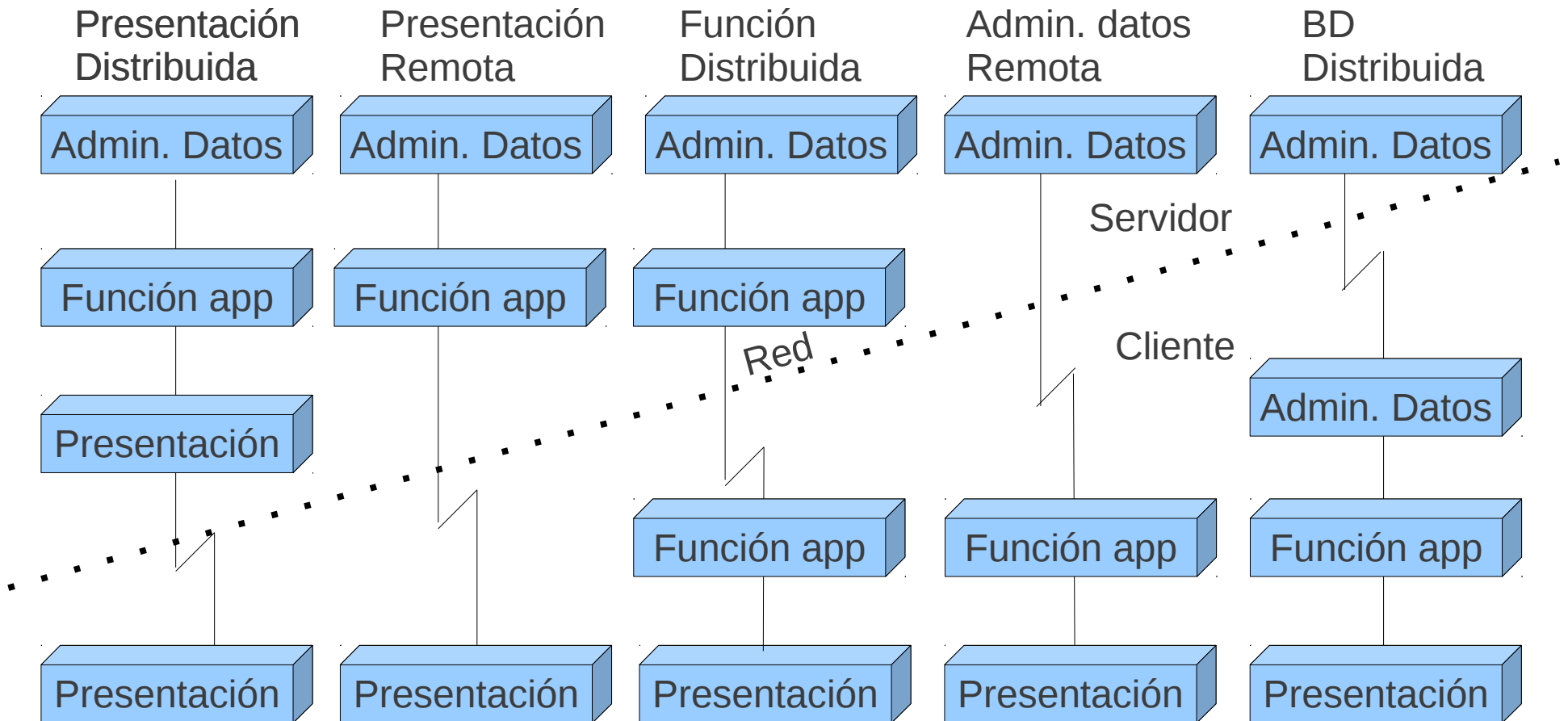
- Permite a las compañías enlazar las partes automatizadas y ahorrar en inversión de IT. Comienza la automatización de los procesos de negocio
- Los procesos cooperantes permiten a las compañías tomar ventajas de programas de computación especializados, además de extender la utilidad de los sistemas
 - El proceso de unir tales aplicaciones o componentes individuales se llama **Integración de Sistemas**

Sistemas cliente-servidor

- Toma ventajas de las capacidades de procesamiento de máquinas host y PCs en el mismo sistema
 - El flujo computacional de un sistema cliente-servidor puro es asimétrico
 - Se divide el trabajo entre clientes y servidores
- Limitaciones:
 - Limita el número de clientes que se pueden conectar a un servidor
 - Los servidores no pueden interactuar con otros servidores

Sistemas cliente-servidor

- Componentes de un sistema cliente-servidor:
 - Software de presentación
 - Aplicación
 - Software de administración de datos



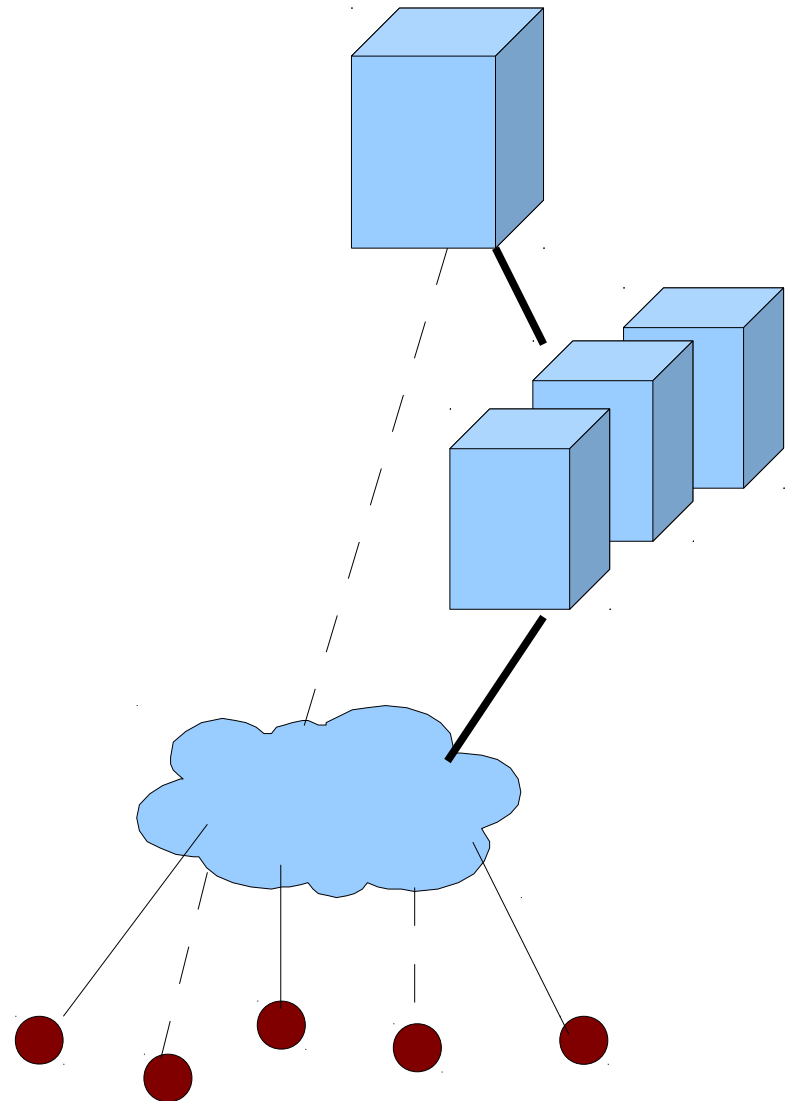
Sistemas cliente-servidor

- La arquitectura más usada ha sido la de tres niveles (three tier)

Nivel 3: Servidores: mainframe, granja de servidores

Nivel 2: Servidores múltiples especializados, middleware

Nivel 1: Clientes, clientes livianos, clientes portables



Sistemas cliente-servidor

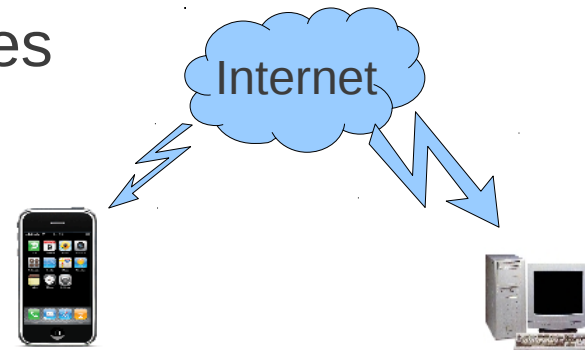
- Beneficios:
 - Mejor acceso a la información
 - Se torna el énfasis de la computación hacia usuarios
 - Incrementa la flexibilidad organizacional
 - Se agrega fácilmente nueva tecnología sin afectar el resto del sistema
 - Incrementa el trabajo en equipo vía redes
 - Soporta nuevas estructuras organizacionales a través de la conectividad
 - Separación clara del flujo de trabajo entre áreas funcionales.

Sistemas cliente-servidor

- Desventajas:
 - No necesariamente representa bajos costos
 - Más fácil para los usuarios pero más complicado para los Sistemas de Información
 - Lo que parecía simple conexión se torna en un gran, generalmente frágil y complejo sistema.

Computación basada en Internet

- A finales de los 90, la tendencia del modelo cliente-servidor aumenta debido a Internet
 - Permanecen los principios del modelo cliente-servidor
 - Internet se convierte en el *core* de estos sistemas
 - Computación Ubicua
 - Integración de plataformas diferentes
 - Nuevo ambiente de computación
 - Modelos irregulares de generación de requerimientos
 - Número de usuarios impredecibles



Computación basada en Internet

- Computación basada en servidores
 - Con más uso de laptops que no tienen fuertes características de seguridad
 - Las aplicaciones residen en servidores corporativos más que en laptops
 - Las aplicaciones se acceden de manera segura desde cualquier dispositivo, se actualizan directamente sobre los servidores y no es necesario adaptarlas a arquitecturas específicas
 - Mientras que la actualización masiva no es fácil

Computación basada en Internet

- Computación Peer-to-Peer
 - Las tareas se distribuyen entre un amplio número de computadores conectados a Internet
 - Se hizo famoso desde Seti y Napster. Redes P2P
 - Promovido por el movimiento de software abierto
 - Conformación de organizaciones virtuales (Computación Grid)
 - Actualmente el interés se volcó a cómo hacer dinero en este ambiente: suscripciones que permiten que los usuarios paguen por el acceso más que por la propiedad (Computación en Cloud)

Servicios Web

- 2da. Generación de sistemas distribuidos basados en Internet, provee módulos de software como URLs (direcciones de Internet) de manera que pueden invocarse para que realicen su función como un servicio vía Internet
- Permite extender el uso de Internet para comunicaciones computador-computador. Los servicios web se comunican entre si
 - **Wrapping:** encapsular la funcionalidad de una aplicación existente en un envoltorio como XML
 - **Exposing:** para que sea usado por otros

Servicios Web

- Algunos estándares de software:
 - XML (**eX**tensible **M**arkup **L**anguage)
 - WSDL (**W**eb **S**ervice **D**efinition **L**anguage)
 - UDDI (**U**niversal **D**iscovery, **D**escription, and **I**ntegration)
- Algunos estándares de comunicación:
 - SOAP (**S**imple **O**bject **A**ccess **P**rotocol)
 - HTTP (**H**yper**T**ext **T**ransfer **P**rotocol)
 - TCP/IP (**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol)

Servicios Web

- Beneficios
 - Considerando los Sistemas de Información como propietarios conduce a procesos rígidos de negocio, que son lentos para cambiar y responder a los cambios del mercado
 - Los Servicios Web ofrecen una arquitectura IT basada en la característica abierta de Internet. En lugar de construir sistemas propietarios, las compañías pueden obtener de Internet la funcionalidad que requieran
 - Esta modularidad permite manejar una amplia variedad de posibilidades de mezclar y hacer corresponder servicios, además de permitir fácil enlace de sistemas entre compañías
 - Las empresas sólo pagan por la funcionalidad que usan y cuando la usan, reduciendo la inversión en IT necesaria