

Diseño de Sistemas Operativos

Capítulo 10 Introducción a los Sistemas Operativos Distribuidos

Extraído de "Sistemas operativos: una visión aplicada"
© J. Carretero, F. García, P. de Miguel, F. Pérez

Introducción a los Sistemas Operativos Distribuidos

1. Sistemas distribuidos (conceptos)
2. Sistemas operativos distribuidos
3. Comunicación de procesos
4. Otras necesidades de los sistemas distribuidos

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

2

© J. Carretero, F. García, P. de Miguel, F. Pérez

Sistemas Distribuidos

Un *sistema distribuido (SD)*

- **Modelo físico:** conjunto de nodos (procesadores sin memoria ni reloj común) conectados por una red.
- **Modelo lógico:** conjunto de procesos que ejecutan *concurrentemente* en uno o más computadores que colaboran y comunican intercambiando *mensajes*.

VENTAJAS

- Compartir recursos (HW, SW, datos).
 - Acceso a recursos remotos.
- Buena relación coste/rendimiento
- Capacidad de crecimiento
- Tolerancia a fallos, disponibilidad:
 - Replicación
- Concurrencia
- Velocidad: Paralelismo

DESVENTAJAS

- Necesidad de software más complejos
- Problemas de fiabilidad: Red
- Problemas de seguridad y confidencialidad

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

3

© J. Carretero, F. García, P. de Miguel, F. Pérez

Comunicación en un Sistema Distribuido

- **Paquete:** tipo de mensaje que se intercambia entre dos *dispositivos de comunicación*. Tamaño limitado por el hardware
- **Mensaje:** objeto lógico que se intercambian entre dos o más procesos. Su tamaño puede ser bastante grande. Un mensaje se descompone en paquetes.
- **Protocolo:** conjunto de reglas e instrucciones que gobiernan el intercambio de paquetes y mensajes
- **Tasa de transferencia:** velocidad de transferencia
- **Latencia:** tiempo necesario para transferir un mensaje vacío
- **Tiempo de transferencia:** latencia + tamaño/tasa de transferencia
- **Paquetes/segundo**
- **Capacidad de crecimiento.** Aumento en el n° de nodos
- **Calidad de servicio** Importante en aplicaciones multimedia y de tiempo real
- **Fiabilidad del subsistema** Mecanismos de detección de errores

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

4

© J. Carretero, F. García, P. de Miguel, F. Pérez

Papel del SO en los Protocolos de Comunicaciones

- El SW de comunicación de un SO:
 - Se organiza como un conjunto de componentes con tareas concretas
 - Subsistema de almacenamiento: *buffers* donde almacenar los paquetes que llegan y se envían
- En implementaciones UNIX típicas
 - TCP para cada puerto un buffer de 8 KB
 - UDP 2 *buffers* de 8KB (incrementable hasta 64 KB).
 - Los mensajes a enviar se copian a estos *buffers*.
 - IP tiene otro espacio de *buffers* (fragmentación).

Diferentes Sistemas Operativos Distribuidos

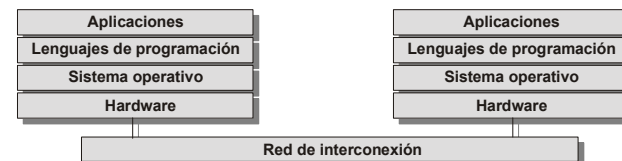
- Sistemas operativos para multiprocesadores con memoria compartida (SMP):
 - Software **fuertemente** acoplado
 - sobre Hardware **fuertemente** acoplado
- Sistema operativo distribuido (SOD):
 - Software **fuertemente** acoplado
 - sobre Hardware **débilmente** acoplado
- Sistema operativo en red:
 - Software **débilmente** acoplado
 - sobre Hardware **débilmente** acoplado

Sistemas Operativos para SMPs

- Características:
- “*Ligeras*” variaciones sobre versiones tradicionales.
 - Sólo hay una copia del sistema operativo.
 - Concurrencia se traduce en paralelismo real (\neq tiempo compartido).
 - Comercialmente aceptados (Linux, Win2K, AIX, ...).
 - Plantea retos para: la ejecución del núcleo en varios procesadores (llamadas al sistema concurrentes), los mecanismos de sincronización (*spin-locks*), optimización y planificación (*afinidad al procesador*), ...

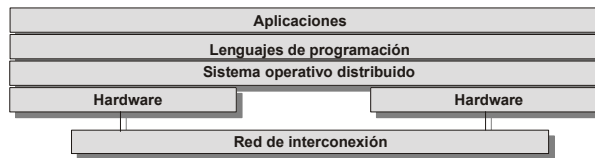
Sistemas Operativos en Red

- El usuario ve un conjunto de máquinas independientes
 - No hay transparencia
- Se debe acceder de forma explícita a los recursos de otras máquinas
- Dificiles de utilizar para desarrollar aplicaciones distribuidas



Sistemas Operativos Distribuidos

- Se comporta como un SO único (visión única)
 - Distribución. **Transparencia**
- Se construyen normalmente como micronúcleos que ofrecen servicios básicos de comunicación
 - Mach, Amoeba, Chorus.
- Todos los computadores deben ejecutar el mismo SOD



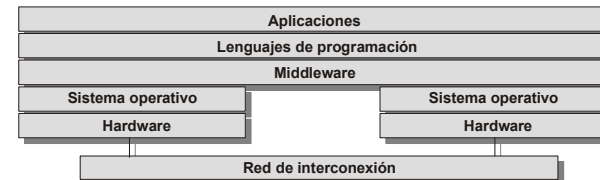
Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

9

© J. Carretero, F. García, P. de Miguel, F. Pérez

Middlewares y Entornos Distribuidos

- Servicios y protocolos estandarizados: Sistemas abiertos
- Ofrecen servicios no incluidos en el SO
- Facilitan el desarrollo de aplicaciones distribuidas
- Independientes del HW y del SO subyacente.
- DCE, CORBA, COM+, Legion, Globe, Globus



Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

10

© J. Carretero, F. García, P. de Miguel, F. Pérez

Comunicación en Sistemas Distribuidos

Mecanismos de bajo nivel: el programador debe preocuparse de establecer los protocolos de comunicación, representación de datos, etc.

- Colas de mensajes
- Sockets

Mecanismo de alto nivel: ofrecen abstracciones donde el programador no debe preocuparse de establecer protocolos

- RPCs
- RMIs (entornos orientados a objetos)

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

11

© J. Carretero, F. García, P. de Miguel, F. Pérez

Necesidades de los Sistemas Distribuidos

El desarrollo de un sistema distribuido complejo requiere el uso de las siguientes funciones y servicios:

- Servicios de comunicación.
- Sistemas de ficheros y nombrado distribuido.
- Servicios de sincronización y coordinación.
- Memoria compartida distribuida.
- Gestión de procesos.
- Servicio de seguridad.

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

12

© J. Carretero, F. García, P. de Miguel, F. Pérez

Servicios de Comunicaciones (I)

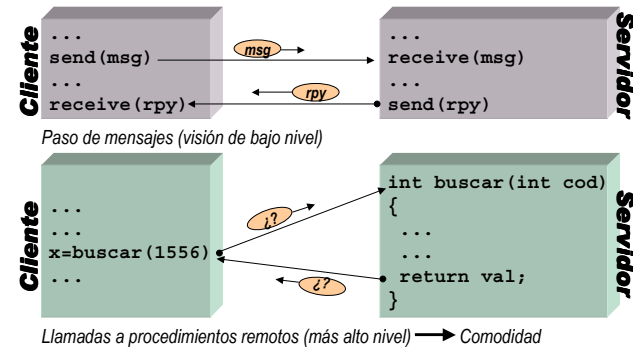
- Modelos de interacción:
 - Cliente/servidor: (2-niveles, 3-niveles o n -niveles)
 - *Peer-to-peer*: Equilibrio de roles.
 - Intermediarios: *Proxy, Dispatcher, Caches, ...*
 - *Unicast vs Multicast*
 - Fiabilidad.
 - Síncronos vs Asíncronos
- Tecnologías de comunicación:
 - Paso de mensajes: *Berkeley sockets*.
 - Llamada a procedimientos remotos: *RPC*.
 - Tecnologías de objetos distribuidos: *CORBA, COM+, EJB*
 - Código móvil: Entornos de agentes.

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

13

© J. Carretero, F. García, P. de Miguel, F. Pérez

Servicios de Comunicaciones (II)



Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

14

© J. Carretero, F. García, P. de Miguel, F. Pérez

Servicios de Comunicaciones (III)

La extensión de los mecanismos de *RPC* a una programación orientada a objetos dio lugar a los **modelos de objetos distribuidos**.

Ventajas:

- Los métodos remotos están asociados a objetos remotos.
- Más natural para desarrollo orientado a objetos.
- Admite modelos de programación orientada a eventos.

Problemas:

- El concepto de referencia a objeto es fundamental.
- Objetos volátiles y objetos persistentes.

Soluciones: *DCOM/COM+* de Microsoft, *CORBA* de OMG y Tecnologías Java de Sun Microsystems

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

15

© J. Carretero, F. García, P. de Miguel, F. Pérez

Sistemas de Ficheros Distribuidos (I)

Identificación, localización y acceso a elementos del entorno distribuido.

Comprende:

- Sistemas de ficheros distribuidos (SFD): *NFS, AFS*.
- Servicios de nombres: *DNS, COS-Naming (CORBA)*.
- Servicios de directorio: *X.500, LDAP, JNDI*.

• Cuestiones:

- Arquitectura de los servicios.
- Almacenamiento intermedio: *caching*.
- Replicación y coherencia.

Sistemas operativos: una visión aplicada
Versión seleccionada y ampliada por © J.M. Peña

16

© J. Carretero, F. García, P. de Miguel, F. Pérez

Sistemas de Ficheros Distribuidos (II)

- **Modelo carga/descarga**
 - Transferencias completas del fichero
 - Normalmente utilizan semántica de sesión
- **Modelo de servicios remotos**
 - El servidor debe proporcionar todas las operaciones sobre el fichero.
 - Acceso por bloques
 - Modelo cliente/servidor
- **Empleo de cache en el cliente**
 - Combina los dos modelos anteriores

Sistemas de Ficheros Distribuidos (III)

- **Servidores con estado**
 - Cuando se abre un fichero, el servidor almacena información y da al cliente un identificador único a utilizar en las posteriores llamadas
 - Cuando se cierra un fichero se libera la información
- **Servidores sin estado**
 - Cada petición está autocontenida (fichero y posición)

El empleo de cache de bloques permite mejorar el rendimiento

Servicios de Sincronización y Coordinación (I)

- Comprende los conceptos de:
 - Tiempo en entornos distribuidos: Sincronización de relojes y relojes lógicos.
 - Concurrencia y Paralelismo: Exclusión mutua e interbloqueos.
 - Algoritmos distribuidos: Elección de líder, coordinación, ...
 - Transacciones: Propiedades ACID, modelos de commit/rollback.
- Afecta a otros servicios:
 - Nombrado e identificación.
 - Seguridad y fiabilidad.
 - Comunicaciones.
 - ...

Servicios de Sincronización y Coordinación (II)

Relojes Físicos:

- Sincronización de relojes hardware.

Relojes Lógicos:

- Ordenación de eventos.

UTC: *Universal Coordinated Time*

- Transmisión de señal desde centros terrestres o satélites.
- Una o más máquinas del sistema distribuido son receptoras de señal UTC.

Servicios de Sincronización y Coordinación (III)

Exclusión mútua: Mecanismo de coordinación entre varios procesos concurrentes a la hora de acceder a recursos/secciones compartidas.

Las soluciones definidas para estos problemas son:

- Algoritmos centralizados.
- Algoritmos distribuidos.
- Algoritmos basados en marcas de tiempo.

Problemática:

- No existen variables compartidas
- Riesgo de interbloqueos
- Riesgo de inanición

Servicios de Sincronización y Coordinación (IV)

Elección de líder:

- Determinar un único proceso/nodo para realizar una tarea determinada.

Coordinación:

- Problemas de consenso.

Transacciones:

- Propiedades ACID.
- *Commit* de 2 fases (2PC).

Memoria Compartida Distribuida (I)

Hardware:

- Memoria físicamente compartida.
- Memoria distribuida (lógicamente compartida).
- Acceso uniforme vs. acceso no uniforme.

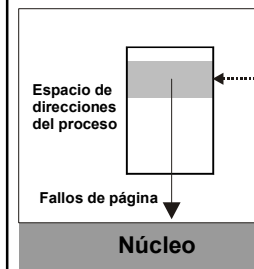
Distributed Shared Memory:

- Basada en páginas.
- Basada en variables compartidas.
- Basada en objetos.

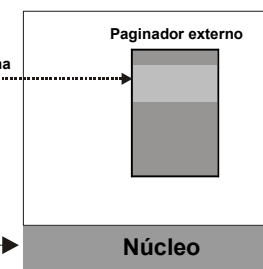
Modelos de consistencia

Memoria Compartida Distribuida (II)

Nodo A



Nodo B



Transferir página

Mensajes

Gestión de Procesos (I)

- Taxonomía de los procesos:
 - Niveles de granularidad.
 - Congelación de procesos (persistencia).
 - Migración de procesos (estado/código).
- Planificación de procesos:
 - Planificación interna: Procesos y *threads*.
 - Planificación global.
 - Migración y equilibrado de carga.
 - Aprovechamiento de máquinas inactivas.

Gestión de Procesos (II)

- **Conjunto de estaciones de trabajo**
 - El sistema consta de estaciones de trabajo a las que tienen acceso los usuarios.
- **Pool de procesadores**
 - Los usuarios con terminales.
 - Los procesos se envían a procesadores de un *pool*.
- **Modelo híbridos**
 - Trabajos interactivos en las estaciones de trabajo.
 - Trabajos no interactivos en el *pool* de procesadores.

Servicio de Seguridad

- Tipología de los ataques:
- Privacidad y confidencialidad.
 - Autenticación (*spoofing*).
 - Denegación de servicio.
- Modelos y herramientas de seguridad:
- Cifrado: clave pública (RSA) y privada (DES).
 - Protocolos de seguridad: IPsec, SSL.
 - Certificados y firmas digitales: X.509.
 - Elementos de seguridad: Firewalls.
- Entornos seguros: e.g. Kerberos.