Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

interacción y cooperación

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Los procesos distribuidos necesitan a menudo coordinar sus actividades
  - sistemas síncronos y asíncronos
  - tolerancia a fallos
  - exclusión mutua de los procesos distribuidos
  - ejemplo: reservas de billetes de avión
- En los SD para solucionar el problema de la exclusión mutua, no se pueden utilizar:
  - ni variables compartidas
  - ni facilidades dadas por un único núcleo central

→ Solución basada en el paso de mensajes

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Algunos servidores implementan sus propios cerrojos para sincronizar los accesos a los recursos que gestionan
- Otros servidores no incluyen sincronización (p.e. Sun NFS):
  - necesitan servicio de exclusión mutua (p.ej. *daemon lockd*)
  - para este caso se requiere un mecanismo de exclusión mutua distribuida:
    - dar a un único proceso el derecho de acceder temporalmente a los recursos compartidos
- En otros casos se necesita elegir a un único proceso de un conjunto para que desarrolle un papel privilegiado durante un largo tiempo (Redes ethernet o inalámbricas)
  - necesario un algoritmo de elección

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Requisitos en la exclusión mutua

- EM1: Seguridad → en todo momento, como máximo hay un solo proceso ejecutando la región crítica
- EM2: Vitalidad → a todo proceso que lo solicita se le concede la entrada/salida en la región crítica en algún momento:
  - evita el abrazo mortal (*deadlock)* e inanición (*starvation*)
- EM3: Ordenación → la entrada en la región crítica debe concederse según la relación sucedió - antes

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Algoritmo basado en servidor central

- El servidor central concede permisos en forma de testigo que concede acceso a la sección crítica (SC)
  - al salir de la SC, el proceso devuelve el testigo al servidor
- Suponiendo que no hay caídas y no se pierden mensajes:
  - se cumplen E1 y E2
  - E3 está asegurada en el orden de llegada de los mensajes al servidor
- Rendimiento del algoritmo
  - 2 mensajes para la entrada en la sección crítica
  - 1 mensaje para salir de la sección crítica

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Problemas:
  - todas las solicitudes se envían al servidor → cuello de botella
  - caída o fallo del servidor → elección de nuevo servidor → E3 no asegurada
  - caída o fallo del proceso en la SC

interacción y cooperación

coordinación distribuida

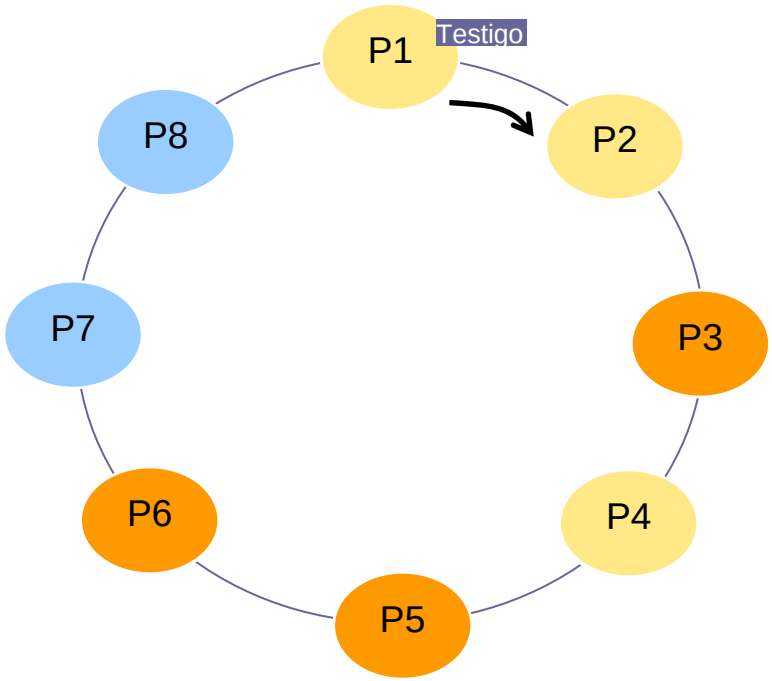
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Algoritmo basado en Anillo
  - La exclusión se logra por la obtención de un testigo
  - Anillo lógico → se crea dando a cada proceso la dirección de su vecino
    - El testigo está siempre circulando por el anillo
    - Cuando un proceso recibe el testigo:
      - si no quiere entrar en la SC → lo envía a su vecino
      - si quiere entrar en la SC → lo retiene
    - Al salir de la SC: lo envía a su vecino
- Se verifican E1 y E2, pero no se asegura E3
- Obtención del recurso necesita entre *1 y (n-1)* mensajes

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Algoritmo basado en Anillo



Anillo de procesos que transfieren un testigo de exclusión mutua

interacción y cooperación

coordinación distribuida

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Problemas:
  - se carga la red aun cuando ningún proceso quiera entrar en la SC
  - si un proceso cae necesita reconfiguración
    - si además tenía el testigo: elección para regenerar el testigo
  - asegurarse de que el proceso ha caído →varios testigos
  - desconexión o ruptura de la red

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Algoritmos basados en relojes lógicos
  - Premisas:
    - cada proceso conoce la dirección de los demás
    - cada proceso posee un reloj lógico

- Algoritmos:
  - Ricart y Agrawala
  - Lamport

interacción y cooperación

coordinación distribuida

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Basados en relojes lógicos: RICART y AGRAWALA
  - Idea básica: cuando un proceso quiere entrar en la sección crítica (SC) → les pregunta a los demás si puede entrar
  - Cuando **todos** los demás le contesten → entra
- El acceso se obtiene a través de un testigo
  - cada proceso guarda el estado en relación a la SC: liberada, buscada o tomada
- Cola de solicitudes en cada proceso
- Mensaje → Tupla <Ti, Pi, SCi>

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Algoritmo de RICART y AGRAWALA

- **En la inicialización**
  estado := LIBERADA;

- **Para entrar en la sección crítica**
  estado := BUSCADA;
  Multitransmite *petición* a todos los procesos;
  T := marca temporal de la petición;
  Espera hasta que (número de respuestas recibidas = (*N* – 1));
  estado := TOMADA;

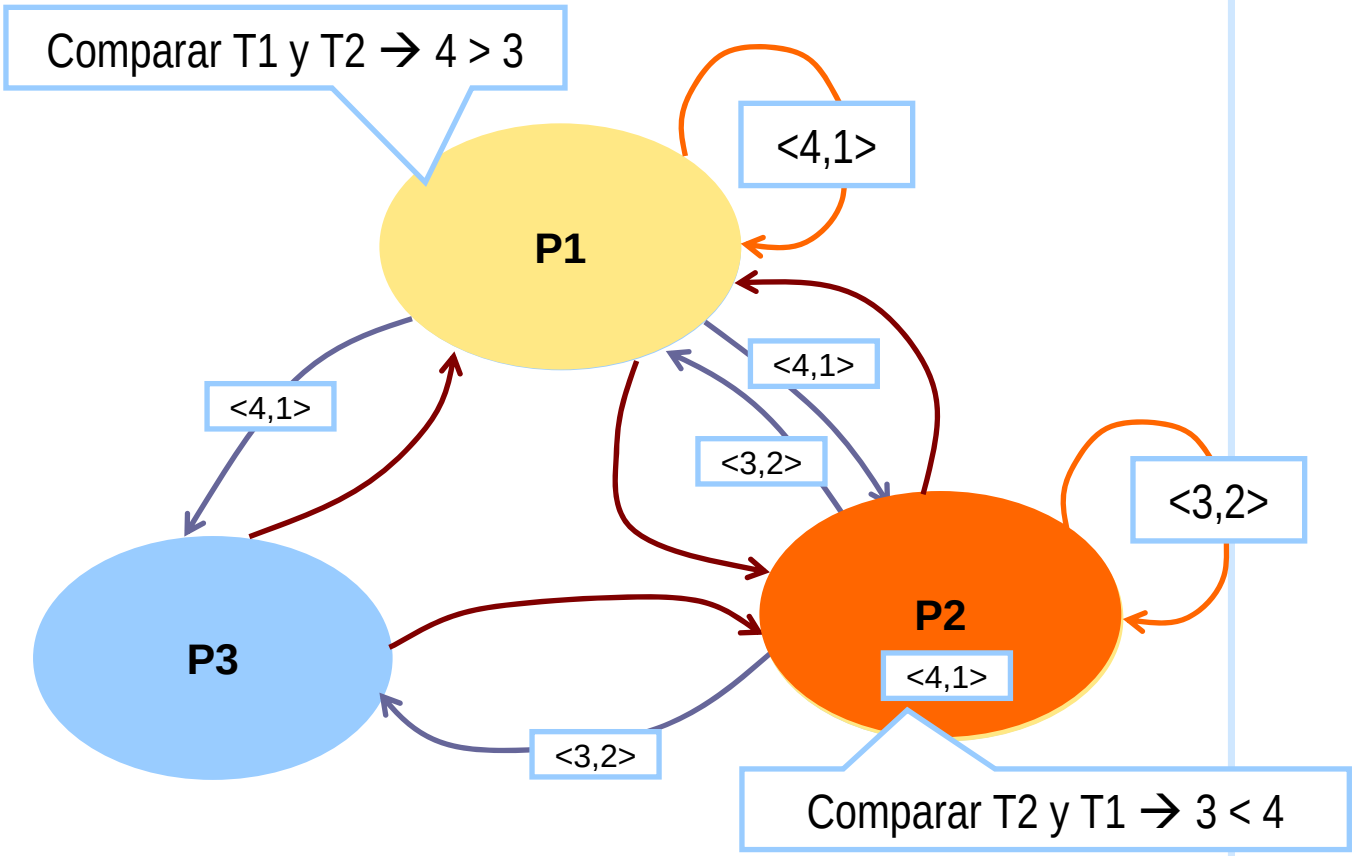
  ⇨ Se aplaza el procesamiento de peticiones

- **Al recibir una petición $<T_i, p_i>$ en el proceso p_j ($i \neq j$)**
  si (*estado* = TOMADA o (*estado* = BUSCADA *y* $(T, p_j) < (T_i, p_i)$))
  entonces
  pone en la cola la *petición* por parte de p_i sin responder;
  sino
  responde inmediatamente a p_i;
  fin si

- **Para salir de la sección crítica**
  estado := LIBERADA;
  responde a cualquiera de las peticiones en la cola;

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Basados en relojes lógicos: RICART y AGRAWALA



Algoritmo de multidifusión y relojes lógicos: Ricart y Agrawala

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Número de mensajes necesarios para obtener el recurso:
  - sin soporte *multicast*: *2(n-1)*
  - con soporte *multicast*: *n*
  - el algoritmo fue refinado hasta *n* mensajes sin soporte *multicast* (Raynal, 1988)
- Problemas:
  - Algoritmo más costoso que el del servidor central
  - Pese a ser algoritmos distribuidos, el fallo de cualquier proceso bloquea el sistema
  - Los procesos implicados reciben y procesan cada solicitud:
    - igual o peor congestión que el servidor central

interacción y cooperación

coordinación distribuida

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Discusión de los algoritmos estudiados
  - Ninguno puede tratar el problema de la caída de un computador o proceso
  - El algoritmo de servidor central es el que tiene menor número de mensajes, pero supone un cuello de botella
- Conclusión:
  - es preferible que el servidor que gestiona el recurso implemente también la exclusión mutua

interacción y cooperación

algoritmos de elección

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Procedimiento para elegir a un proceso dentro de un grupo
  - ejemplo: elegir a un proceso que sustituya a uno especial (coordinador, maestro, ...) cuando éste cae
- Principal exigencia: elección única incluso si varios procesos lanzan el algoritmo de elección de forma concurrente
- E1 → Seguridad
- E2 → Vivacidad
- Dos algoritmos:
  - algoritmo basado en anillo: Chang y Roberts
  - algoritmo del matón (bully): Silberschatz

interacción y cooperación

algoritmos de elección: anillo

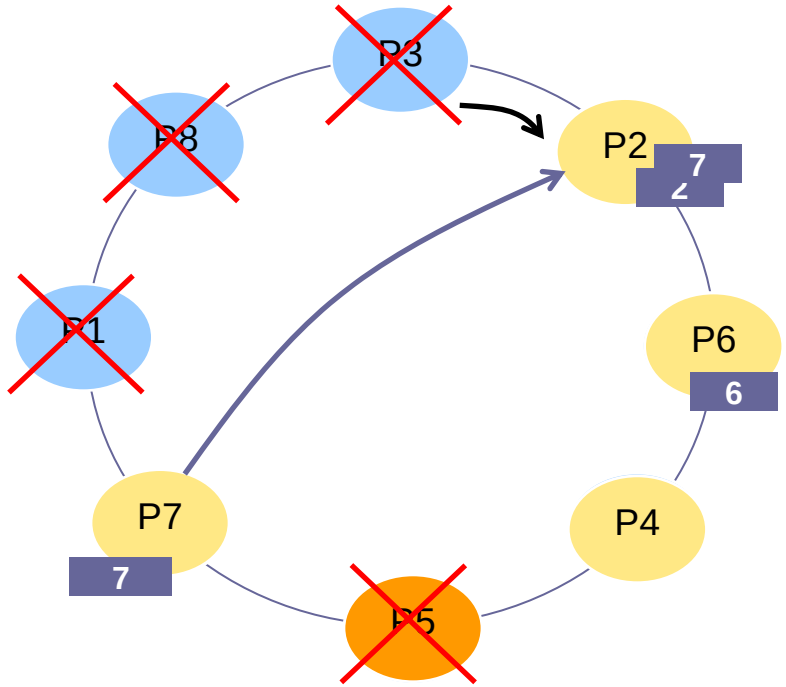
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Inicialmente todos los procesos son no-candidatos: cualquiera puede empezar una elección:
  - se marca como candidato
  - envía mensaje de **elección** con su identificador
- Cuando un proceso recibe un mensaje de **elección**:
  - si identificador del mensaje es mayor que el suyo → envía mensaje a sus vecinos
  - si es menor:
    - si es no-candidato → sustituye el identificador y envía mensaje al vecino y se marca como candidato
  - si es el suyo
    - se marca como no-candidato
    - envía mensaje de **elegido** a su vecino añadiendo su identidad
- Cuando un proceso recibe un mensaje de **elegido**:
  - se marca como no-candidato
  - lo envía a su vecino

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Algoritmo basado en anillo: ejemplo



Anillo de procesos que transfieren un testigo de exclusión mutua

interacción y cooperación

algoritmos de elección: anillo

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Anillo lógico: cada proceso sólo sabe comunicarse con su vecino
- Se elige al proceso con identificador más alto
- Se supone procesos estables durante la elección
- Tanenbaum (1992):
  - variante donde los procesos pueden caer
- Número de mensajes para elegir coordinador:
  - peor caso: lanza elección sólo el siguiente al futuro coordinador → *(3n-1)* mensajes
  - mejor caso: lanza elección el futuro coordinador → *2n* mensajes
- No detecta fallos

interacción y cooperación

algoritmos de elección: bully

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Requisitos:
  - todos los miembros del grupo deben conocer las identidades y direcciones de los demás miembros
  - se supone comunicación fiable
- El algoritmo selecciona al miembro superviviente con mayor identificador
- Los procesos pueden caer durante la elección
- Hay 3 tipos de mensajes:
  - mensaje de elección: para anunciar una elección
  - mensaje de respuesta a un mensaje de elección
  - mensaje de coordinador: anuncia identidad de nuevo coordinador
- Número de mensajes para elegir coordinador:
  - caso mejor: se da cuenta el segundo más alto → *(n-2) m*ensajes
  - caso peor: se da cuenta el más bajo → $O(n^2)$ mensajes

interacción y cooperación

algoritmos de elección: bully

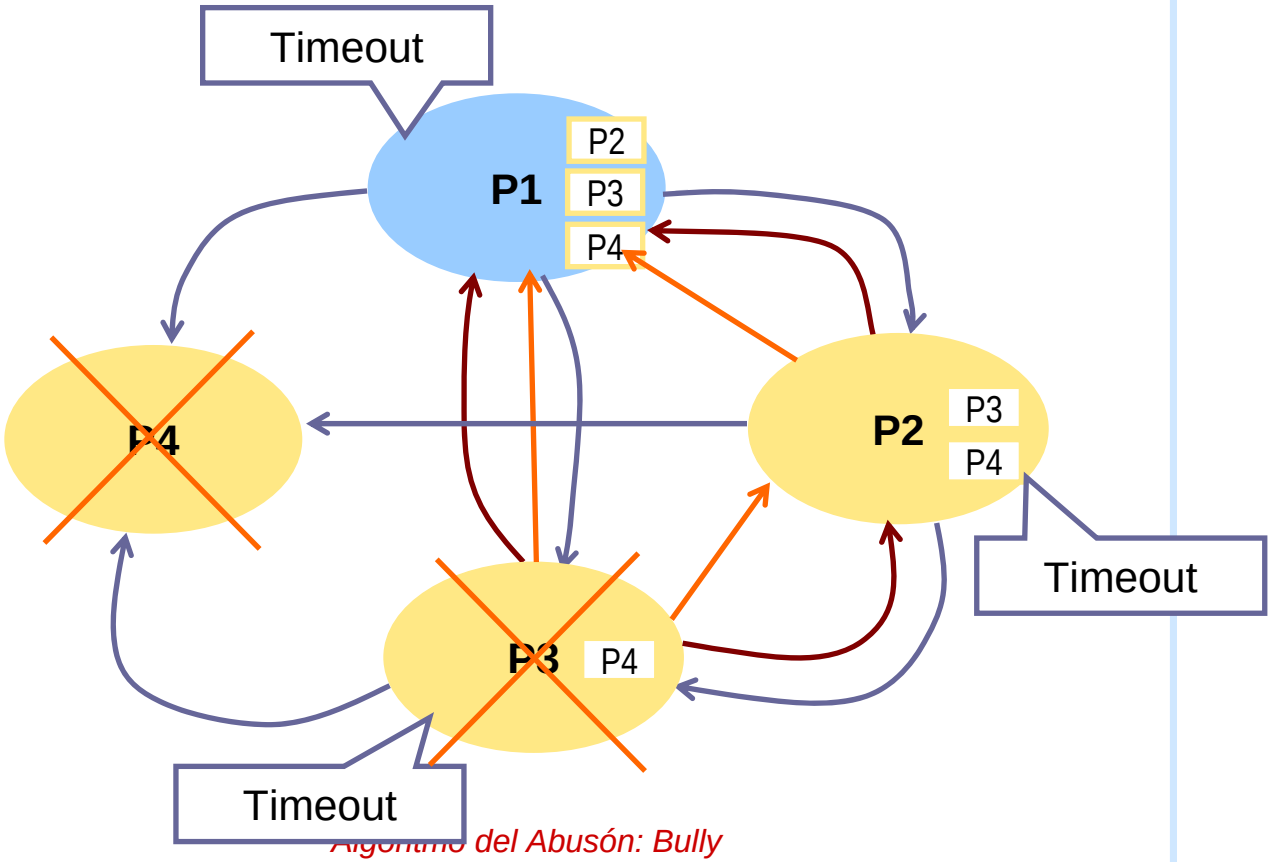
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:
  - envía mensaje de **elección** a los procesos con identificador mayor que el suyo
  - espera algún mensaje de **respuesta**:
    - si vence temporizador →el proceso se erige como coordinador y envía mensaje de **coordinador** a todos los procesos con identificadores más bajos
  - si recibe alguna **respuesta** → espera mensaje de **coordinador**.
    - si vence temporizador, lanza una nueva elección
- Si un proceso recibe un mensaje de **coordinador**:
  - guarda el identificador y trata a ese proceso como nuevo coordinador
- Si un proceso recibe un mensaje de **elección**:
  - contesta con un mensaje de **respuesta** y lanza una elección, (si no ha lanzado ya antes una)
- Cuando un proceso se reinicia:
  - lanza una elección a menos que sea el de identificador más alto (en cuyo caso se erigiría como nuevo coordinador)

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Algoritmo “bully”: ejemplo



Algoritmo del Abusón: Bully

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

sistemas de archivo distribuido

introducción

funciones del FS

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Un sistema de ficheros distribuido (DFS):

- Tiene las mismas funciones que el sistema de ficheros de un SO convencional, pero más complejo
- Los usuarios y los dispositivos de almacenamiento se encuentran dispersos por la red
- Adicionalmente, un SFD debe permitir:
  - compartir información remotamente
  - movilidad de los usuarios
  - disponibilidad

introducción

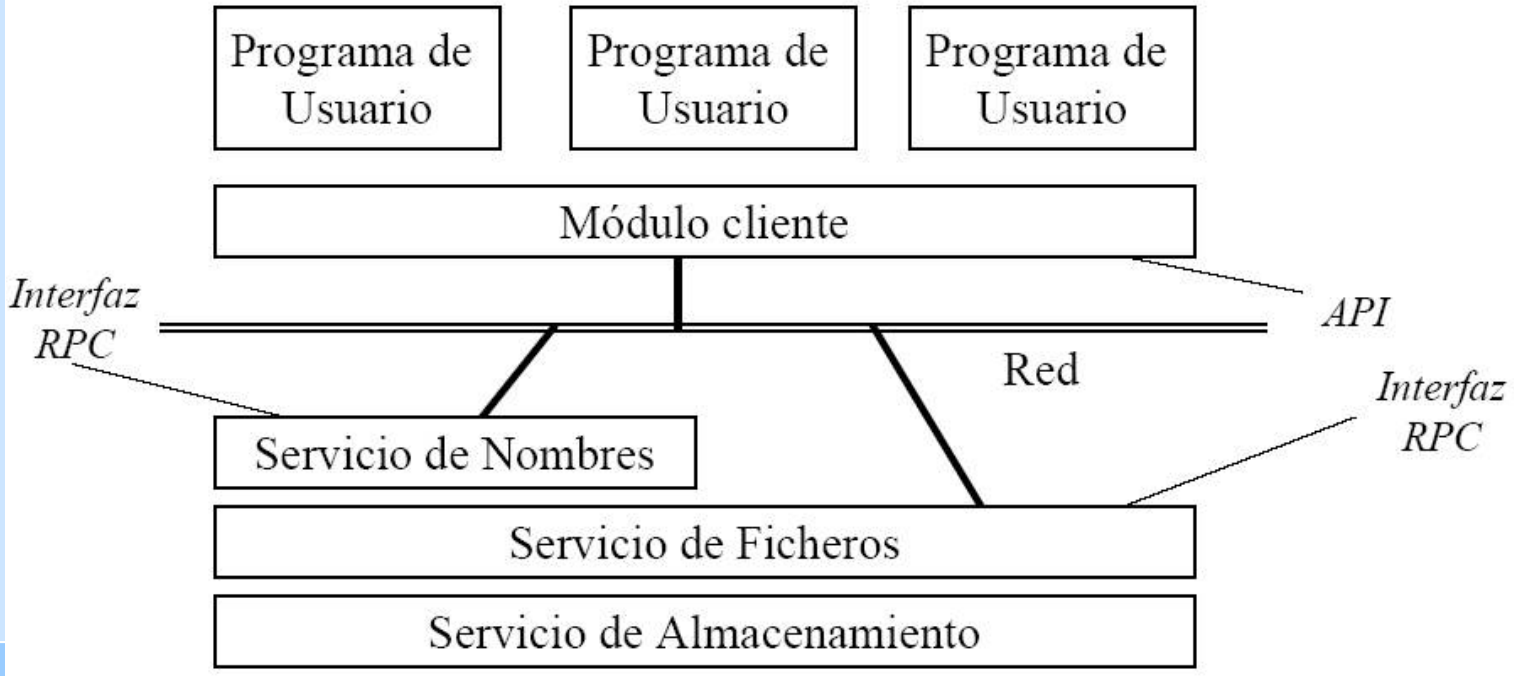
componentes de un SFD

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Un SFD se puede estructurar en cuatro componentes:

- servicio de almacenamiento
- servicio de ficheros
- servicio de nombres o de directorio
- módulo cliente (biblioteca de funciones o API)



introducción

objetivos de diseño

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Transparencia
  - de acceso
  - de ubicación
  - de movilidad
  - de prestaciones, aunque la carga varíe
  - de escala
- Movilidad de los usuarios
- Rendimiento
- Alta disponibilidad y tolerancia a fallos
- Concurrencia
- Fiabilidad e integridad de la información
- Seguridad
- Heterogeneidad del HW's y SO's

servicio de ficheros

modelos de acceso a ficheros remotos

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Existen dos modelos básicos:
  - modelo de servicio remoto
    - las operaciones se realizan en los servidores
    - problemas de eficiencia
  - Modelo de caché de datos
    - los ficheros se acceden de forma local
    - aumento del rendimiento
    - problemas de consistencia
- Se pueden combinar ambos modelos
  - ejemplo: NFS, Dropbox, OneDrive, ...

técnicas de implementación

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Según los tipos de servidores

Implementación de los DFS:

- **Caché**: copia asociada a cliente, a nivel de bloque

- **Replicación de ficheros**: copia asociada a servidor, a nivel fichero completo

técnicas de implementación

tipos de servidores

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Servidores sin estado (*stateless*): no almacenan información entre solicitudes de un mismo cliente
  - tolerancia a fallos
  - no requieren llamadas para abrir y cerrar ficheros
  - no se desperdicia memoria en tablas
  - no existe límite para el número de ficheros en uso (ficheros abiertos)
  - no se producen problemas si cae un cliente
- Servidores con estado (*stateful*): almacenan información entre solicitudes de un mismo cliente
  - mensajes de solicitud de servicio más cortos
  - mejor rendimiento
  - es posible realizar operaciones de lectura anticipada
  - permiten el bloqueo de ficheros

técnicas de implementación

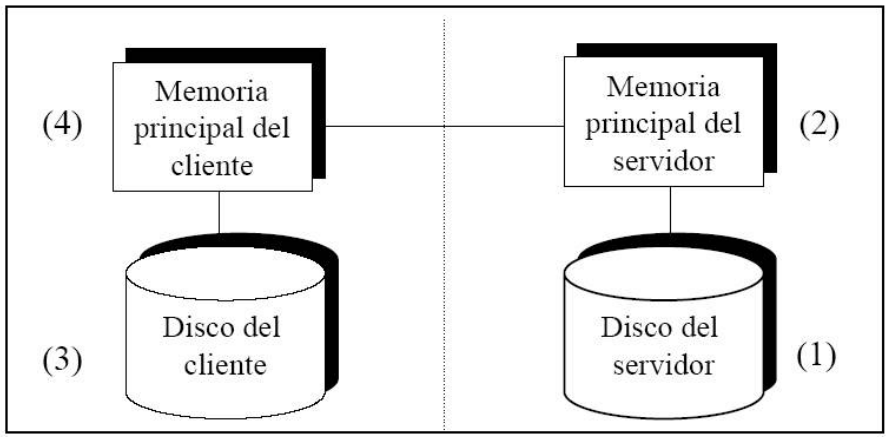
técnicas de caché

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Retener en memoria principal aquellos datos que han sido usados más recientemente

Localización de la caché

- Hay cuatro lugares posibles en los que almacenar la información
  1. el disco del servidor
  2. la memoria del servidor
  3. el disco del cliente (el cliente puede no tener disco)
  4. la memoria del cliente



técnicas de implementación

técnicas de caché

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Localización de la caché en el disco del servidor

- No se aconseja la caché en disco, demasiado costoso

Localización de la caché en la memoria del servidor

- Ventajas
  - transparente para los clientes
  - los sistemas operativos tradicionales suelen utilizar este esquema (NFS)
- Inconvenientes
  - coste de las transferencias por la red

Localización de la caché en el disco del cliente

- Ventajas
  - fiabilidad
  - gran tamaño de la caché
- Inconveniente
  - hay que acceder a disco

Localización de la caché en la memoria del cliente

- Ventajas
  - se eliminan los costes de transmisión por la red y de acceso a disco
  - máximo rendimiento en caso de acierto
- Inconvenientes
  - problema de inconsistencia de la información en la caché

técnicas de implementación

tratamiento de la caché

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- Cuando se utiliza caché en los clientes se pueden producir inconsistencias
- Aspectos a considerar:
  - Cuándo propagar las modificaciones hechas en la caché al fichero en el servidor
  - Cómo verificar la validez de los datos en las cachés
- Dos esquemas básicos para cuándo propagar las modificaciones:
  - Escritura inmediata (*Write-through*)
  - Posponer las escrituras (*Delayed –Write* o *Writeback*)

técnicas de implementación

tratamiento de la caché: validación

Contenido
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

- La política de propagación de modificaciones
  - especifica cuándo se actualiza la copia principal de un fichero cuando una de las copias en caché es modificada
  - pero no establece cuándo actualizar las cachés
- El contenido de una caché se vuelve inválido cuando otro cliente modifica la copia principal
  - es necesario comprobar si la caché de un cliente es consistente con la copia principal
  - en caso contrario es necesario invalidar la caché y actualizar los datos
- Dos estrategias básicas
  - iniciadas por los clientes
  - iniciadas por los servidores

técnicas de implementación

replicación de ficheros

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Un **fichero replicado** es aquél del que existen varias copias, cada una de las cuales está en un servidor diferente

- La replicación de ficheros aporta las siguientes ventajas
  - aumenta la disponibilidad
  - aumenta la fiabilidad
  - mejora el tiempo de respuesta
  - reduce el tráfico en la red
  - mejora el rendimiento (*throughput*)
  - beneficia la escalabilidad
  - permite trabajar en modo de operación desconectada

- Cuestiones a considerar
  - conseguir que la replicación sea transparente a los usuarios
  - cómo actualizar las copias en el caso de modificaciones en una réplica

técnicas de implementación

replicación de ficheros

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Transparencia de replicación:

- Aspectos a considerar
  - denominación de las réplicas
  - control de la replicación
- Denominación de las réplicas
  - ¿cómo distinguir a una réplica de otra si ambas tienen el mismo identificador?
  - un servidor de nombres debería hacer corresponder al identificador la réplica más conveniente
- Control de la replicación
  - explícita
  - implícita
    - replicación perezosa
    - comunicación a grupos

técnicas de implementación

replicación de ficheros

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

Protocolos de actualización de réplicas

- Replicación de sólo lectura
  - se aplica a ficheros inmutables
- Protocolo *Escribir en todos-Leer de cualquiera*
  - las escrituras son costosas
  - problemas si un servidor cae
- Protocolo basado en la disponibilidad de copias
  - problemas de inconsistencias en caso de partición de la red
- Protocolo de copia primaria
  - se lee de cualquiera, se escribe en una
  - problemas si cae el servidor primario
- Protocolo basado en quorum
  - los clientes requieren un quorum de Nr servidores para leer y Nw para escribir
  - Nr + Nw > N (N = número de réplicas)

casos de ejemplo

oneDrive

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación

OneDrive for Business Synchronization

How does OneDrive for Business synchronize files?

The business of synchronization

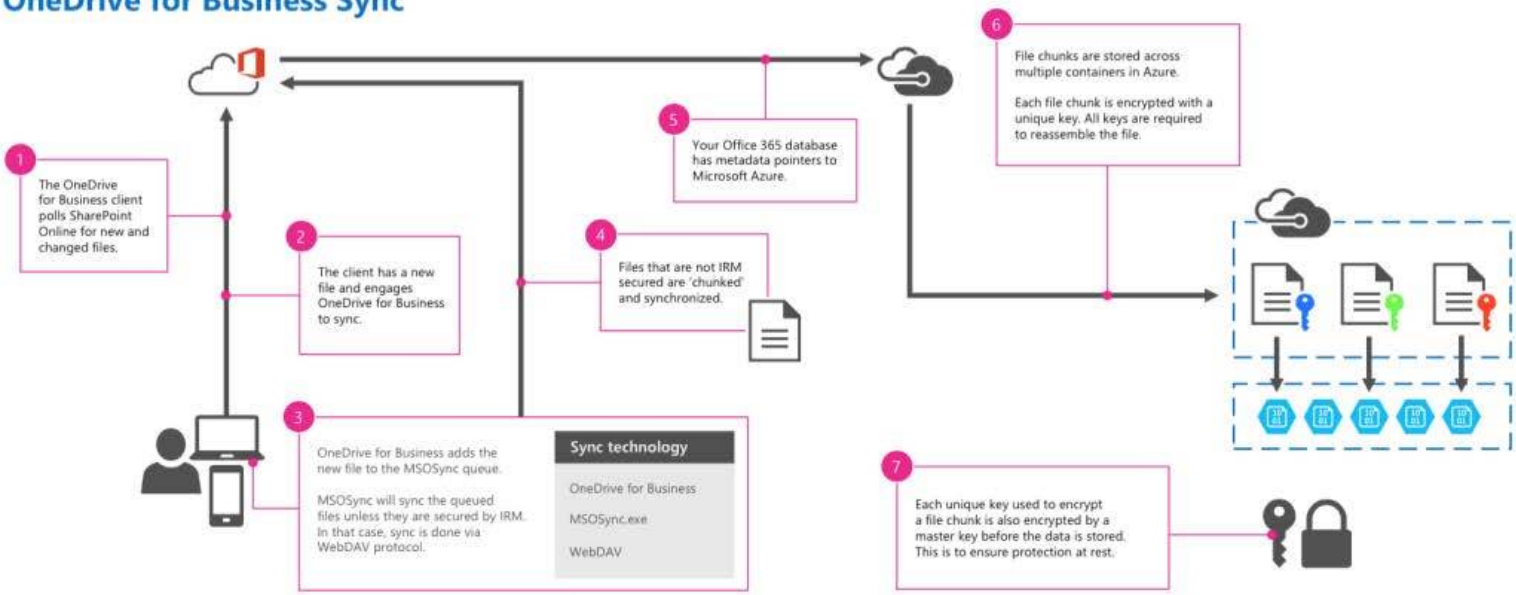
OneDrive for business is constantly checking for new and changed files to synchronize. When a change is detected, such as a new file on a client machine, client synchronization determines the technology to use to sync. Web services are used to bridge the gap between the client computer and Office 365.

Web Services used in synchronization

- **Cellstorage.svc** - used for file synchronization.
- **Sites.asmx** - returns site collection information.
- **Webs.asmx** - used to work with sites and webs.
- **Lists.asmx** - used to work with lists and list data.
- **Version.asmx** - used to work with file versions.

OneDrive

OneDrive for Business Sync



casos de ejemplo

IPFS

Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad
- coordinación