

## Contenido

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

# sistemas de archivo distribuido

#### Contenido

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

### Un sistema de ficheros distribuido (DFS):

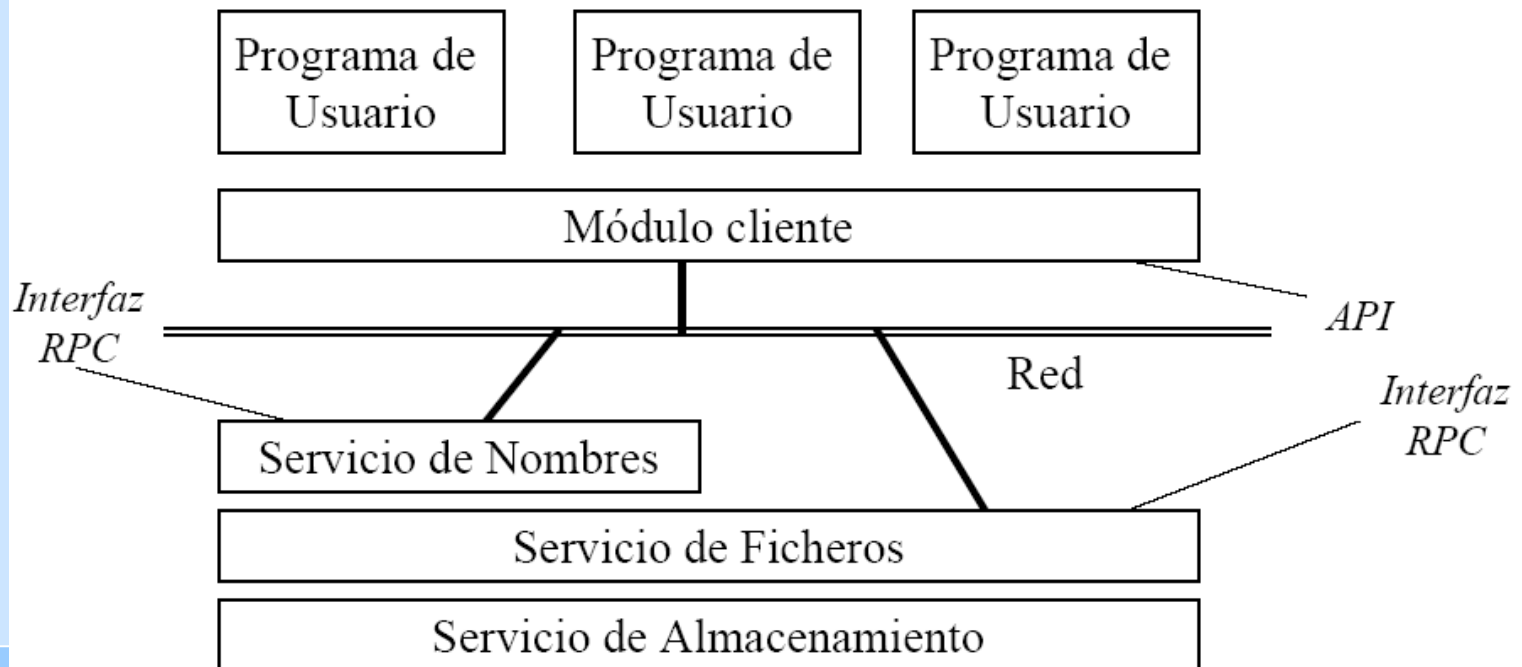
- Tiene las mismas funciones que el sistema de ficheros de un SO convencional, pero más complejo
- Los usuarios y los dispositivos de almacenamiento se encuentran dispersos por la red
- Adicionalmente, un SFD debe permitir:
  - compartir información remotamente
  - movilidad de los usuarios
  - disponibilidad

#### 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)



#### 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, ...

## 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

### Contenido

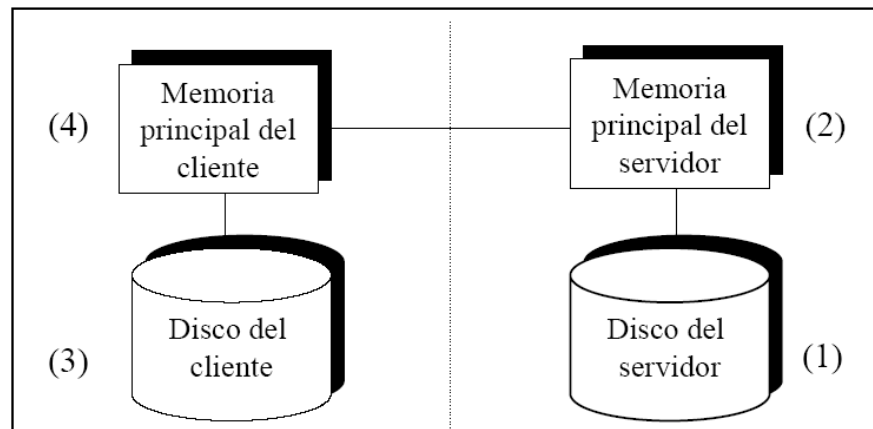
- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad
- coordinación

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

## 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

#### 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

## 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  $N_r$  servidores para leer y  $N_w$  para escribir
- $N_r + N_w > N$  ( $N$  = número de réplicas)

## Contenido

### OneDrive for Business Synchronization

#### How does OneDrive for Business synchronize files?

##### The business of synchronization

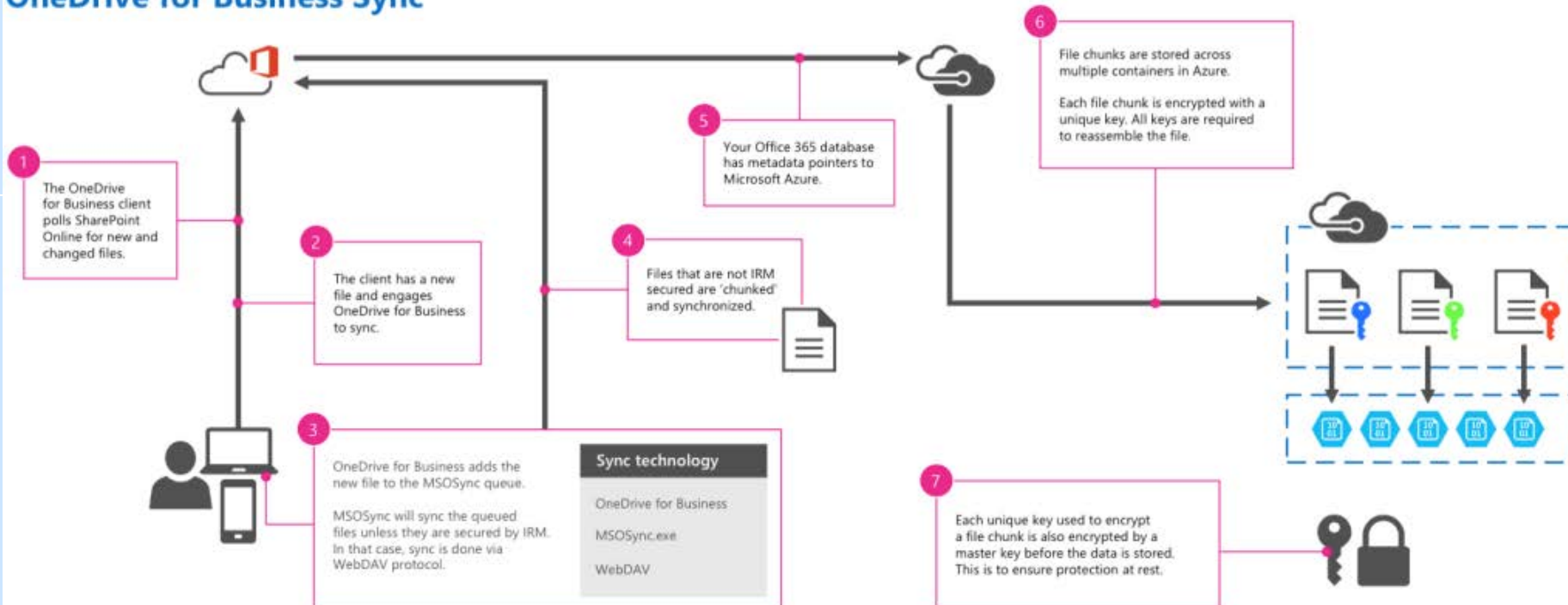
OneDrive for business is constantly checking for new and changed files to synchronize. When a change is detected, such as a new file on a client machine, client synchronization determines the technology to use to sync. Web services are used to bridge the gap between the client computer and Office 365.

##### 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 for Business Sync



### Contenido

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

