

Sistemas distribuidos

Grado en Ingeniería Informática

Tema 05-01: Sistemas de Ficheros Distribuidos

Departamento de Ingeniería Informática
Universidad de Cádiz



Escuela Superior de Ingeniería
Dpto. de Ingeniería Informática



Curso 2019 – 2020

Indice

- 1 Introducción
- 2 Arquitectura de Servicio de Ficheros
- 3 Network File System
- 4 Otros

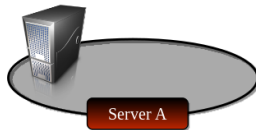
Sección 1 | Introducción

Introducción

- **Objetivo** Compartir información almacenada en localizaciones distribuidas
- **Enfoques de almacenamiento**
 - **No persistente:**
 - Almacenamiento en memoria compartida
 - Almacenamiento de objetos: CORBA, EJB, etc.
 - **Persistente:**
 - Datos no estructurados (Ficheros) (NFS, xFS..gNutella, Peer-to-Peer)
 - Datos Estructurados: BBDD distribuidas/federadas/paralelas, Datawarehousing

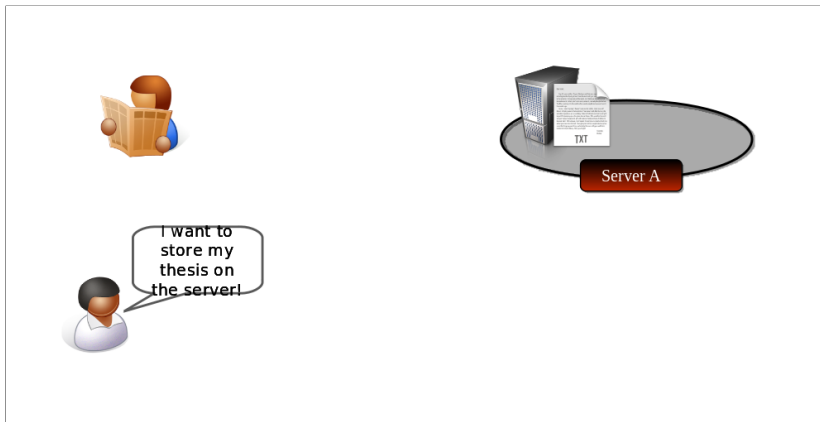
Introducción

Un caso de SD



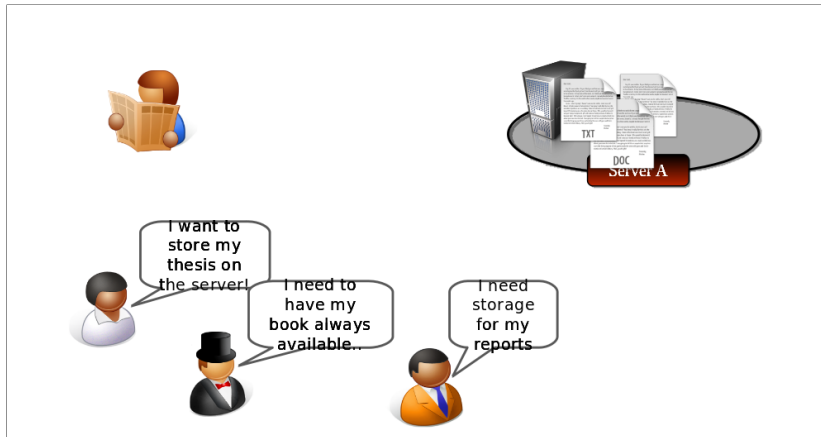
Introducción

Un caso de SD



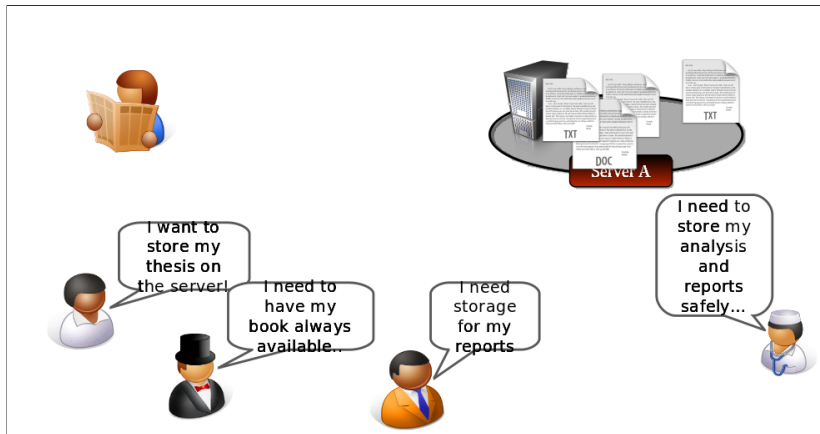
Introducción

Un caso de SD



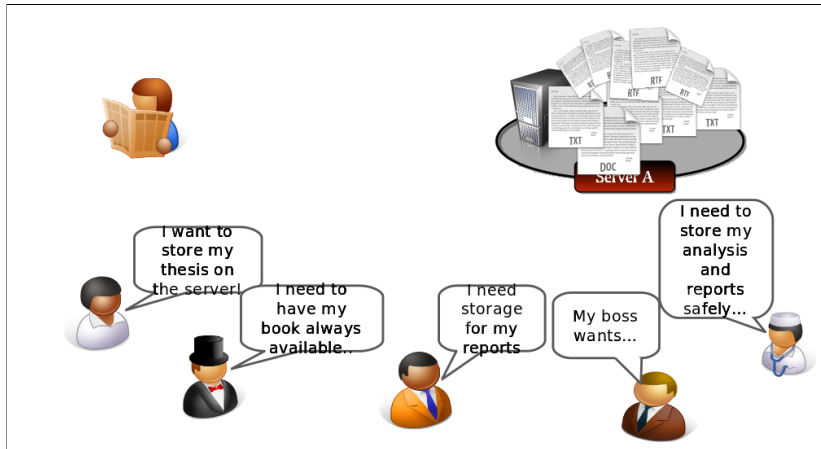
Introducción

Un caso de SD



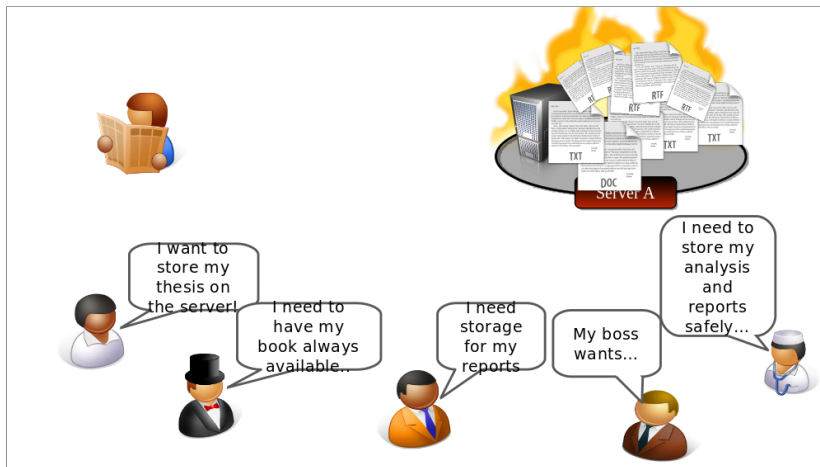
Introducción

Un caso de SD



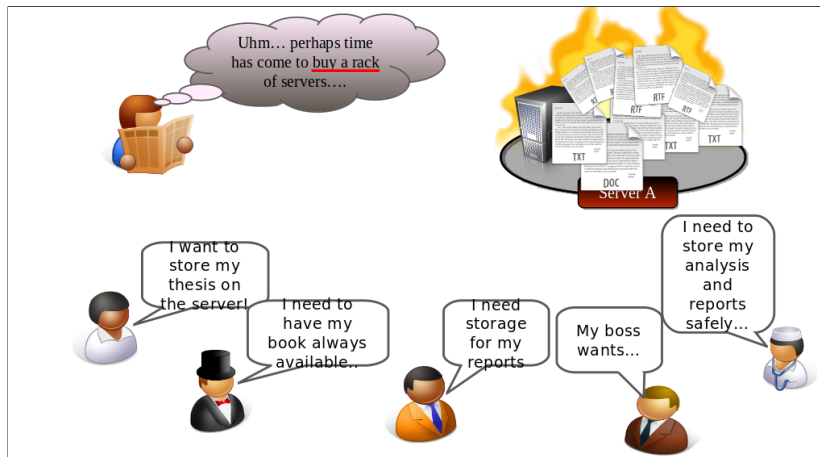
Introducción

Un caso de SD



Introducción

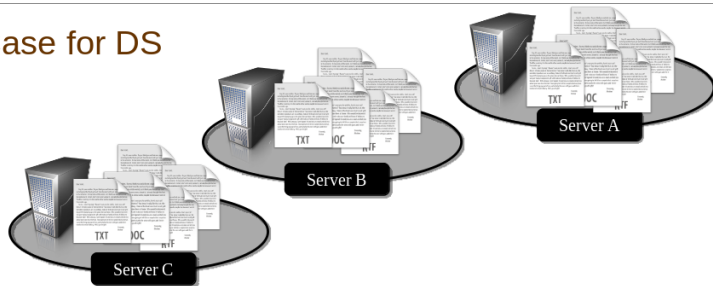
Un caso de SD



Introducción

Un caso de SD

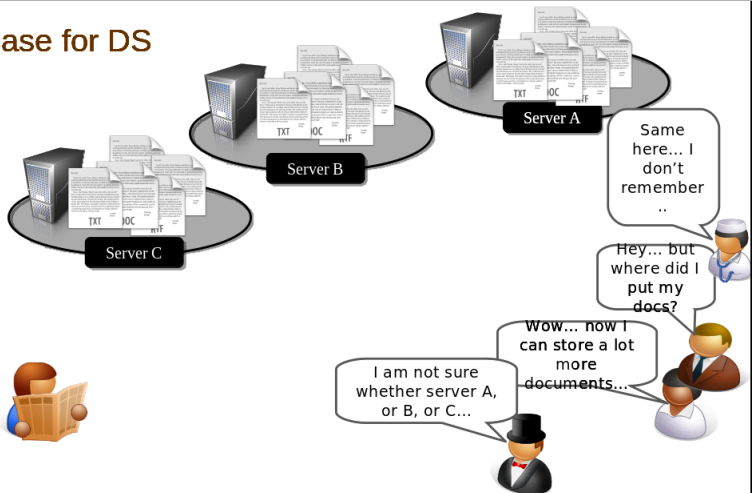
A Case for DS



Introducción

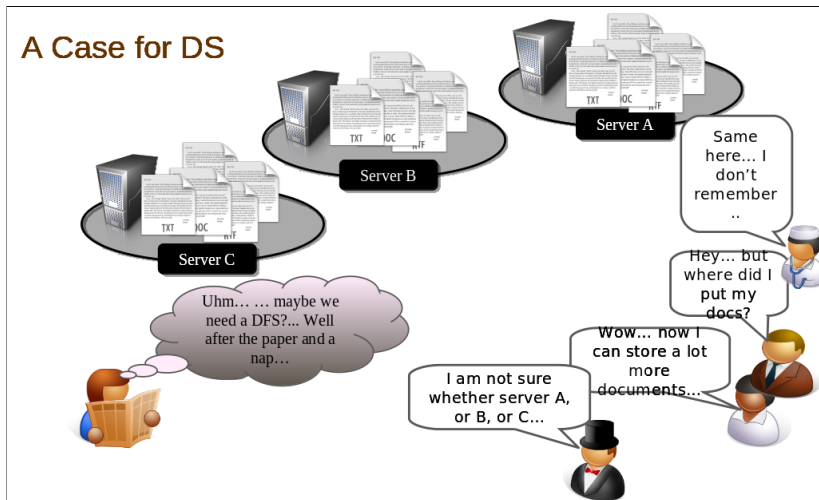
Un caso de SD

A Case for DS



Introducción

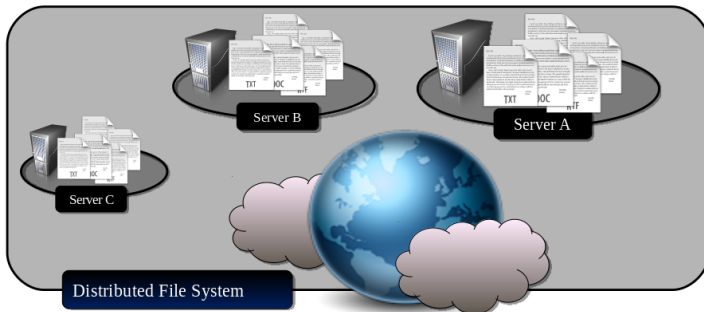
Un caso de SD



Introducción

Un caso de SD

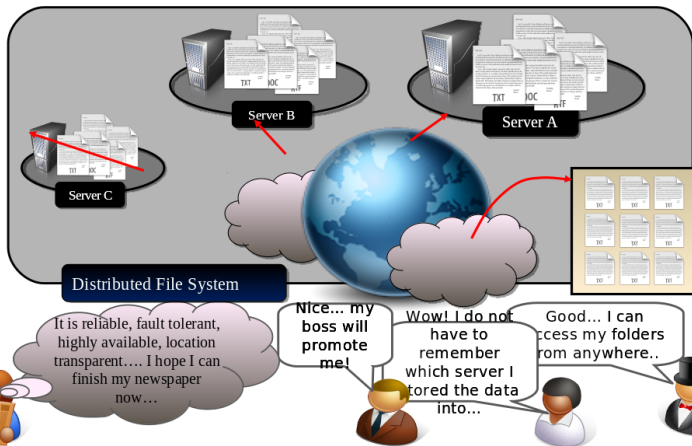
- A Case for DFS



Introducción

Un caso de SD

- A Case for DFS



Introducción

Conceptos

- **Caracterización**

- **Sistema o servicio de ficheros distribuido:** software cuyo propósito es emular la funcionalidad de un sistema de ficheros no distribuido
- **Objetivo:** permitir a programas clientes acceder a los ficheros como si estuvieran en su nodo local

- **Conceptos implicados**

- **Ficheros** elementos almacenados formados por “datos” (secuencias de bits) y “atributos” (longitud, tipo, timestamp, propietario, ACL, etc.)
- **Directorios** ficheros especiales que contienen nombres (id) de ficheros dependientes de él
- **Metadatos:** información usada para la gestión del sistema de ficheros
- **Operaciones:** crear, abrir, cerrar, leer escribir, posicionar, eliminar, renombrar, asignar atributos.

Introducción

Requisitos

- Transparencia: equilibrio entre flexibilidad y escalabilidad de la transparencia frente a complejidad y rendimiento del SW
 - De acceso: los clientes no deben ser conscientes de la distribución de los ficheros → conjunto único de operaciones
 - De localización: los clientes deben ver un único espacio de nombres de ficheros uniforme → transferencia de ficheros sin que cambie el "pathname"
 - De movilidad: información en los clientes acerca de los ficheros no debe variar cuando se migra un fichero
 - De rendimiento: los clientes deben seguir con su rendimiento aun cuando la carga del servicio de ficheros distribuidos aumenta
 - De escalabilidad: el sistema de ficheros debe ser capaz de escalar sin que mermen sus capacidades

Introducción

Requisitos

- Concurrencia: Los cambios hechos a un fichero por parte de un cliente no deben afectar al acceso a ese fichero por parte de otros clientes
- Replicación: un mismo nombre de fichero puede referirse a varios ficheros replicados.
 - Ventaja: reparto de carga y tolerancia a fallos
 - Desventaja: actualizaciones y mantenimiento de copias
- Heterogeneidad
 - El servicio puede ser accedido por clientes que utilizan plataformas o sistemas operativos diferentes
 - El diseño debe ser compatible con los sistemas de ficheros de los diferentes sistemas operativos
 - Se necesita saber la especificación de API para acceder a los servicios

Introducción

Requisitos

- Tolerancia a fallos: recuperación ante fallos del cliente o del servidor
- Consistencia: debido a la existencia de replicación, todas las copias de un fichero deben ser iguales → propagación de cambios
- Seguridad: mecanismos de control de acceso a los ficheros
- Eficiencia: un sistema de ficheros distribuidos debe ofrecer, al menos, la misma eficiencia que uno local

Evolución

1974-1995

En la primera generación de los sistemas distribuidos, los sistemas de ficheros (como NFS), eran solo sistemas de almacenamiento en red.

1995-hoy

Con los avances de los sistemas de objetos distribuidos (como CORBA, Java) y la web han creado sistemas más complejos

Actualmente

Los sistemas a gran escala y con almacenamiento escalable

- Google File System
- Amazon S3
- Cloud Storage (e.g., Dropbox)

Introducción

Sistemas de almacenamiento y sus propiedades

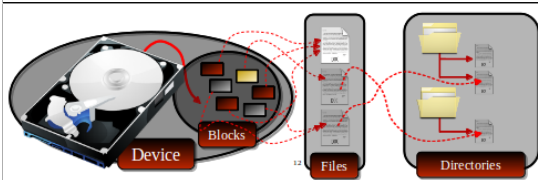
	<i>Sharing</i>	<i>Persistence</i>	<i>Distributed cache/replicas</i>	<i>Consistency maintenance</i>	<i>Example</i>
Main memory	×	×	×	1	RAM
File system	×	✓	×	1	UNIX file system
Distributed file system	✓	✓	✓	✓	Sun NFS
Web	✓	✓	✓	×	Web server
Distributed shared memory	✓	×	✓	✓	Ivy (DSM, Ch. 6)
Remote objects (RMI/ORB)	✓	×	×	1	CORBA
<u>Persistent object store</u>	✓	✓	×	1	CORBA Persistent State Service
<u>Peer-to-peer storage system</u>	✓	✓	✓	2	OceanStore (Ch. 10)

Types of consistency:
 1 strict one-copy ✓: slightly weaker guarantees 2: considerably weaker guarantees

Sistema de ficheros

Módulos típicos para un sistema no Distribuido

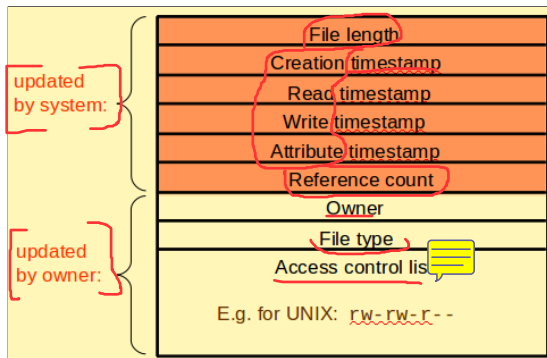
Directory module:	relates <u>file names to file IDs</u>
File module:	relates <u>file IDs to particular files</u>
Access control module:	<u>checks permission</u> for operation requested
File access module:	<u>reads or writes</u> file data or attributes
Block module:	<u>accesses and allocates disk blocks</u>
Device module:	<u>performs disk I/O and buffering</u>



Sistema de ficheros

Estructura típica del registro de atributos

Los sistemas de archivos son responsables de la organización, almacenamiento, recuperación, nominación, compartición y protección de los archivos. Los archivos contienen **datos** y **atributos**



Ejemplos de sistemas de ficheros distribuidos

- Arquitectura de servicio de ficheros:

- Modelo arquitectónico genérico sobre el que se asientan NFS o AFS
- Basado en la división de responsabilidades entre el cliente y el servidor

- NFS: Network File System

- Idea de Sun Microsystems (1985)
- Protocolo NFS: RFC 1813 (1995)
- Relación simétrica entre clientes y servidores (multiplataforma)

- AFS: Andrew File System

- Entorno de computación distribuido de la CMU (1986)
- Basado en la transferencia de grandes bloques de ficheros y el uso de cachés

para aumentar la
eficiencia a la hora de
pasar leer, etc ficheros


Sección 2 | Arquitectura de Servicio de Ficheros

Arq. de Servicio de Ficheros

Componentes

El alcance de la extensibilidad y configurabilidad se mejora si el servicio se estructura en tres componentes encargados de lo concerniente al acceso a ficheros

- Servicios de Ficheros Planos

- Implementa las operaciones en los  tenidos de los ficheros para su acceso
- Crear **UFIDs** (Unique File identifiers) para representar cada fichero físico

- Servicios de directorio

- Mapea nombres (texto) a UFIDs Mapping (clave(id unico), valor)
- Gestiona ficheros y directorios

- Módulo cliente

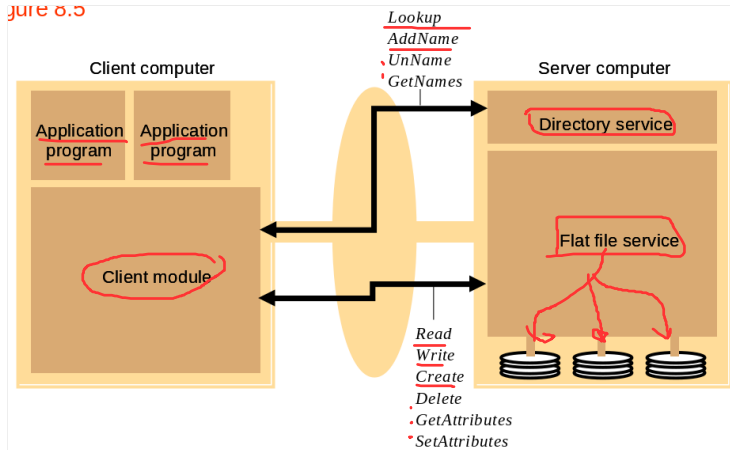
- Integra y extiende las operaciones del servicio de ficheros planos y del servicio de directorio como un simple API.
- Crea un interfaz común de acceso para aplicaciones cliente.

Sistema de ficheros

Estructura típica del registro de atributos

Los módulos y sus relaciones.

Figure 8.5



Sistema de ficheros

Interfaces/operaciones del Servicio

- Interfaz del servicio de ficheros planos
 - Conjunto de operaciones usadas por un cliente para operar con los ficheros
- Interfaz del servicio de directorio
 - Propósito: ofrecer una forma de trasladar nombres de ficheros planos a UFIDs
 - Ofrece también operaciones sobre directorios.

Sistema de ficheros

Interfaces/operaciones del Servicio

Flat file service

id fich
position of first byte
`Read(FileId, i n) -> Data`
position of first byte
`Write(FileId, i Data)`
`Create() -> FileId`
`Delete(FileId)`
`GetAttributes(FileId) -> Attr`
`SetAttributes(FileId, Attr)`

Directory service

`Lookup(Dir, Name) -> FileId`
`AddName(Dir, Name, FileId)`
`UnName(Dir, Name)`
`GetNames(Dir, Pattern) -> NameSeq`
busqueda en un directorio

Sistema de ficheros

¿Suficiente?

¿Asunto zanjado? No todo está resuelto:

- Resolución de nombre de fichero:

Cliente y varios servidores involucrados. ¿Cómo se reparten trabajo?

- Acceso a los datos:

- ¿Se transfiere sólo lo pedido? ¿más cantidad? ¿todo el fichero?
- Uso de cache en el cliente. Coherencia entre múltiples caches

- Otros

- Migración
- Replicación

Sección 3

Network File System

Sun Network File System

Introducción

- Desarrollado por Sun Microsystems in 1985
- Muy popular, abierto y ampliamente usado
- Estandarizado: Versión 3 IETF (RFC 1813), Versión 4 RFC-3010

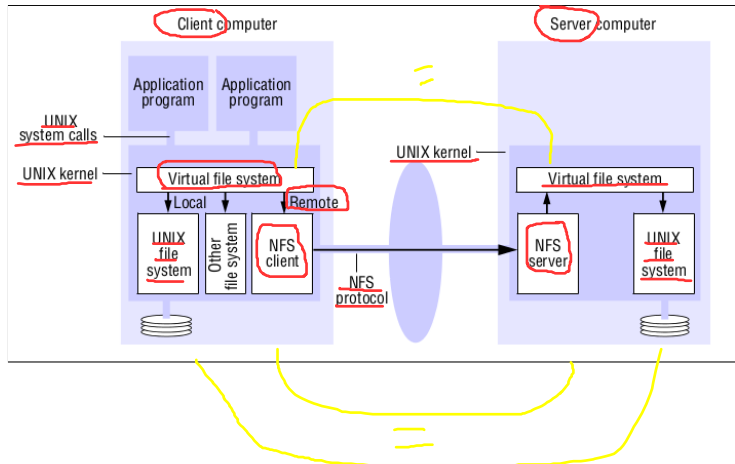
Características

- Soporta mucho de los requerimientos de diseño mencionados
 - Transparencia, Heterogeneidad, Eficiencia, Tolerancia a fallos
- Comportamiento limitado respecto a
 - Concurrencia, Replicación, Consistencia, Seguridad
- El protocolo no ofrece mecanismos de control de concurrencia
(Protocolo independiente de NFS: Network Lock Manager)

Sun Network File System

Arquitectura NFS

el cliente a través del módulo nfs cliente hace uso del protocolo nfs, y acceder al sistema de ficheros virtual de ese servidor. Todo esto es transparente al usuario



Sun Network File System

Operaciones NFS

- read(fh, offset, count) -> attr, data
- write(fh, offset, count, data) -> attr
- create(dirfh, name, attr) -> newfh, attr
- remove(dirfh, name) status
- getattr(fh) -> attr
- setattr(fh, attr) -> attr
- lookup(dirfh, name) -> fh, attr
- rename(dirfh, name, todirfh, toname)
- link(newdirfh, newname, dirfh, name)
- readdir(dirfh, cookie, count) -> entries
- symlink(newdirfh, newname, string) -> status
- readlink(fh) -> string
- mkdir(dirfh, name, attr) -> newfh, attr
- rmdir(dirfh, name) -> status
- statfs(fh) -> fsstats

fh = file handle:

Filesystem identifier	i-node number	i-node generation
-----------------------	---------------	-------------------



Sun Network File System

Conceptos Básicos

- Protocolo NFS: conjunto de llamadas a procedimiento remoto que ofrecen las capacidades necesarias para que los clientes puedan realizar operaciones sobre un almacenamiento de ficheros remoto.
- Clientes y servidores NFS:
 - Se comunican por RPCs siguiendo el protocolo NFS
 - Comunicación con operaciones síncronas
 - Con TCP o UDP por debajo
- Únicas restricciones de uso: que la petición esté bien formulada y que el usuario tenga las credenciales adecuada

Sun Network File System

Resumen

- Un excelente ejemplo de sistema distribuido sencillo, robusto y alto rendimiento.
- **Acceso:** Excelente. El API se esta integrado en el infefaz de llamadas locales y remotas de UNIX.
- **Localización** No se garantiza pero normalmente funciona. El nombre de sistemas de ficheros es controlado por los  ntes durante las operaciones de montaje, pero la transparencia se puede asegurar mediante una configuración apropiada del sistema
- **Concurrencia** Limitada, pero adecuada para la mayoría de los propositos. Cuando ficheros de lectura-escritura son compartidos de manera concurrente entre cliente la consistencia no es perfecta 
- **Réplica** Limitada a los ficheros de solo lectura, para los de escritura, el SUN Network Informaticon Service (NIS) corre sobre NFS y es usado para sistema esencial de réplica

Sun Network File System

Resumen

- **Tolerancia a fallos** Limitado pero eficaz. El servicio es suspendido si un servidor falla.
- **Movilidad** Apenas conseguido. La reubicación entre servidores se puede realizar pero la actualización de las tablas de montaje remoto en cada cliente debe hacerse de manera individual.
- **Rendimiento** Bueno. Los servidores multiprocesador permiten un alto rendimiento.
- **Escalabilidad** Buena. Se puede aumentar las prestaciones de un servidor mediante la adición de procesadores. Los grupos de ficheros pueden ser subdivididos y realojados en otros servidores.

Sección 4 | Otros

Google File System (GFS)

Desarrollo

Google File System (GFS)

- Características
- Arquitectura
- Funciones del servidor Master y Chunk
- Cómo se resuelve una consulta de un cliente
- GFS API
- Replicaciones
- Consistencia

Hadoop Distributed File System (HDFS)

Desarrollo

Hadoop Distributed File System (HDFS)

- Características
- Arquitectura
- Funciones de los servidores
- Cómo se resuelve una consulta de un cliente
- HDFS API
- Replicaciones
- Consistencia