Lab 1

Distributed Software Systems

UNIBO - 2024/25

Davide De Rosa - 1186536

Distributed File Storage System

Goal: enable users to interact with files in a distributed environment as if they are stored locally, hiding the complexity of the infrastructure.

Challenge: achieve *location transparency*, where users shouldn’t know where files are physically stored.

Location Transparency

Definition: *location transparency* in a distributed system means users can access resources without needing to know the physical location of those resources.

Users can interact with **logical names** or **paths** that represent files.

The system handles the mapping from logical names to the physical storage locations.

Goals of Location Transparency

Seamless Access: users should access files just like they would on a local disk, without needing to be aware of file distribution across servers.

Scalability: the system must scale horizontally across servers and data centres without impacting user experience.

Resilience: files should remain accessible even in case of server or network failures.

One way to achieve *Location Transparency* is to use **Distributed File Naming System** combined with **Name Resolution and Mapping**.

Distributed File Naming System

Purpose: provides an easy and consistent way for users to refer to files without knowing their physical location.

Abstracts the complexity of where files are stored. Users interact with a simple logical name ("*/user/documents/doc.pdf*") without needing to care if the file is on *Server A* or *Server B*.

Benefits:

- **Consistency**: name remains unchanged even if the file is moved among servers.
- **User simplicity**: users don't have to deal with complexities related to where their files are stored.

Name Resolution and Mapping

Purpose: maps the logical file names to the actual physical locations of files on different servers.

When a user requests a file using the logical name, the system needs to resolve the logical name to the actual physical location where the file is stored. This is done by using a **metadata server** (or a distributed set of them).

The *metadata server* maintains an internal mapping of logical file names to the physical addresses of files stored on various servers.

The information stored is the **logical name**, the **physical location** (server ID and the exact storage path) and the **redundancy information** (details of any replicas of the file in case of server failures).

Name Resolution and Mapping

Benefits:

- **Dynamic Mapping**: if the file is moved or replicated across different servers, the metadata server updates the mapping, ensuring the user's logical file name remains the same.
- **Efficient Access**: the metadata server optimizes access by pointing users to the closest or least-loaded copy of the file.
- **Resilience**: if a server fails, the metadata server can redirect the request to a replica stored elsewhere.

Why do we need both?

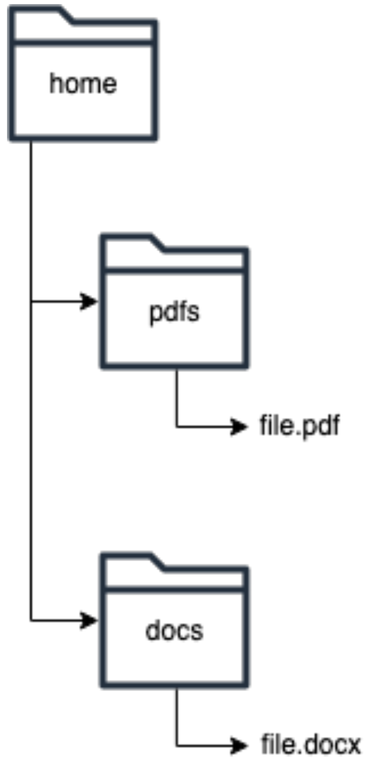
Logical names alone are **not sufficient** because they don’t tell the system where the file is actually stored. This is where *name resolution* comes in.

Without a *name resolution mechanism*, the system wouldn’t know where to find the file associated with a logical name, so even though users are abstracted from the complexity, the system itself **wouldn't be able to function effectively**.

Both components are **required** to achieve *location transparency* in a distributed system. **Logical file names** provide a user-friendly way to access files. **Name resolution and mapping** ensure the system can find those files across the distributed network.

Let’s see an example

Logical path



Physical path

Server A

/files/file.pdf

Server B

/extra/file.docx

Indexing

/home/pdfs/file.pdf -> serverA@/files/file.pdf

/home/docs/file.docx -> serverB@/extra/file.docx

Thank you!