file systems

file service architecture

- requirements
  1. most common for any file systems:

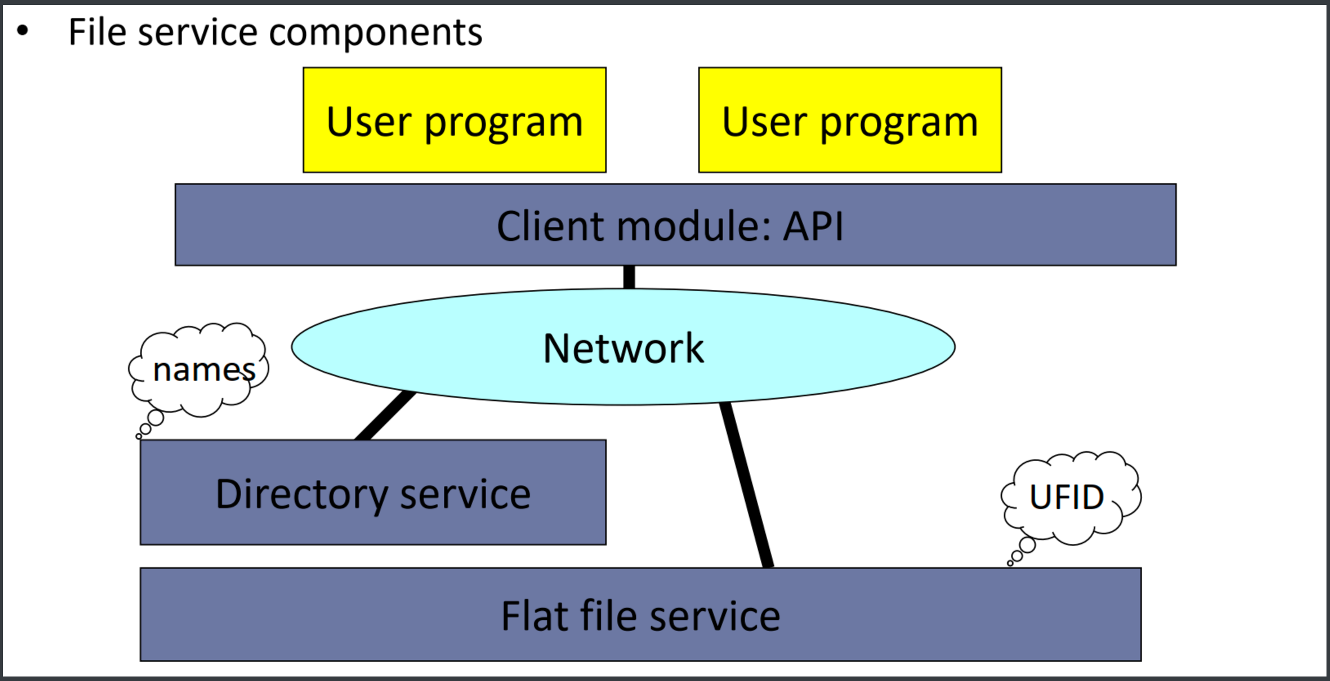
     access / location ... transparency
  2. distribution related

     concurrent updates, heterogeneity, scalability
  3. scalable to a very large # nodes

     replication / migration transparency
  4. future

     support for fine-grained distribution of data

     tolerance to network partitioning
- flat file service



- File service components

User program

User program

Client module: API

Network

names

Directory service

UFID

Flat file service

  - file = data + attributes （文件=数据+元数据）
  - data = sequence of items
  - attributes 就是metadata

    既存在flat file service中，也存在directory service中以便更高层次的服务使用

e.g. access right / time-stamps

- UFID: Unique File ID 全局唯一标识符
- operations

  read, write, create (a file), delete (a file), getAttributed, setAttributes

- directory service:
  - **translate** file name into UFID
  - substitute for open()

    传统文件系统用open()调用路径解析，但distributed情况下文件的物理存储可能在多个节点上，直接用open()无法解析出一个valid的路径

  - **responsible for access control**
  - operations

    lookup, addName, unName, reName, getNames

- flat file service - **fault tolerance**
  - straightforward for simple servers

    提供文件的基础读写服务，不涉及高层次的管理或复杂逻辑，非常适合简单的服务器架构

  - idempotent operations

    幂等操作（执行一次和执行多次结果相同），适合fault tolerance的重试操作；

  - stateless servers

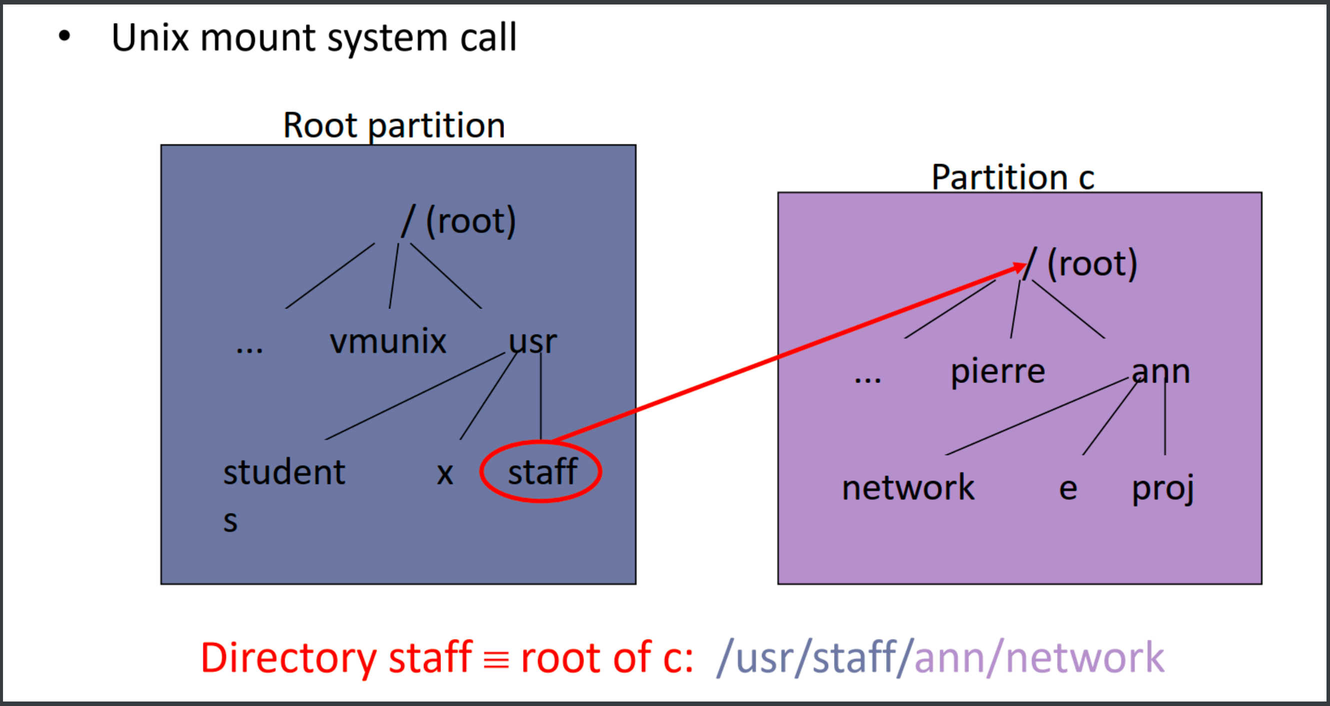
    stateless设计，不会保留关于client操作的信息，使其故障后易于恢复（不会丢失未完成的会话状态 / 任务）

implementation techniques

- file groups: unit of administration
- space leaks
- capabilities (a "digital key")
- file location
- caching

NFS (network file system)

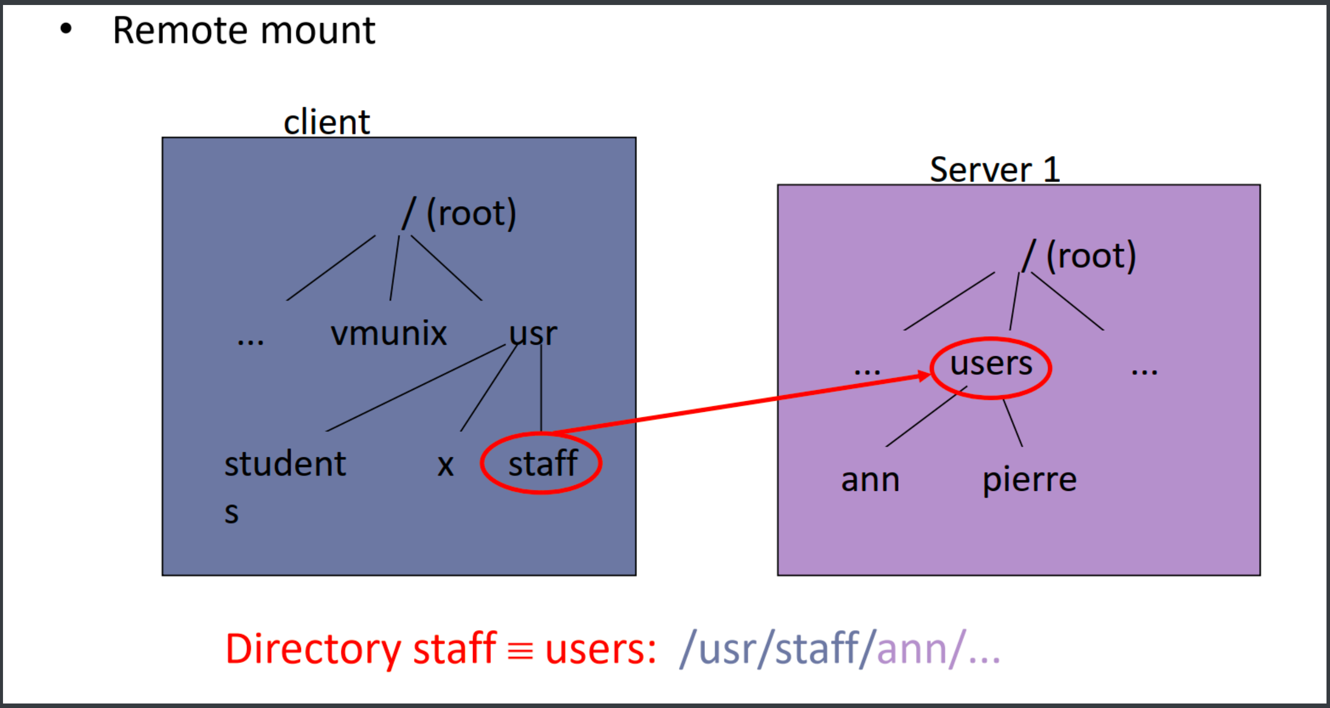
- unix mount system call



- Unix mount system call

Directory staff ≡ root of c: /usr/staff/ann/network

- remote mount



- Remote mount

Directory staff ≡ users: /usr/staff/ann/...

- mounting semantics

  hard - client waits until request for a remote file succeeds (could be forever)

  soft - failure returned if request does not succeed after n retries

- automounter: 动态挂载远程文件系统，仅在需要访问时挂载，从而避免不必要的资源消耗

AFS (andrew file system)

comparison NFS <> AFS