Computação na nuvem
ISEL – LEIRT / LEIC / LEIM

Armazenamento distribuído de ficheiros

José Simão jsimao@cc.isel.ipl.pt ; jose.simao@isel.pt

Luís Assunção lass@isel.ipl.pt ; luis.assuncao@isel.pt

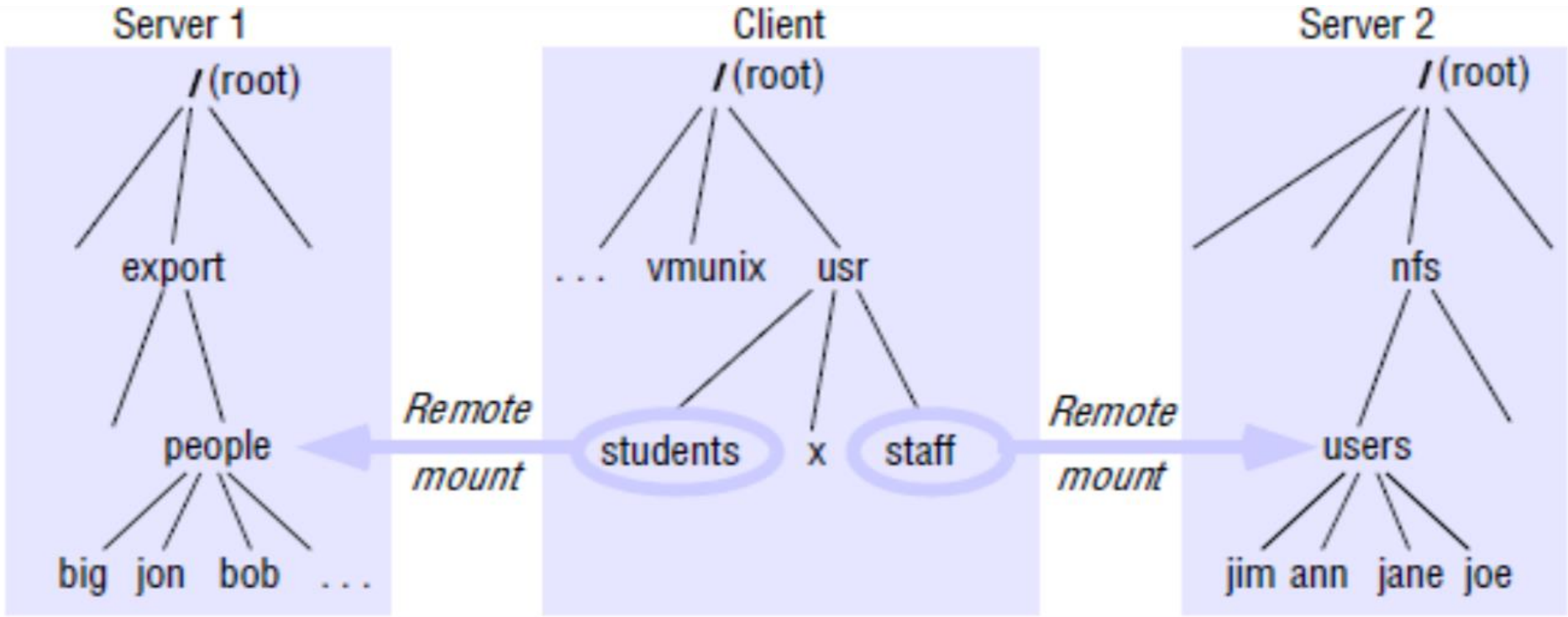
Sumário

- Armazenamento distribuído de dados em ficheiros
  - Um pouco de evolução histórica
- Requisitos de escalabilidade, disponibilidade
  - Dimensões BigData
  - Google File Sytem
- Armazenamento Flat com agrupamentos (*buckets*) de objetos
- Serviço Google Cloud *Storage* (GCS)
  - Classes de armazenamento no GCS
  - Criação e controlo de acesso a *buckets*
  - Espaço de nomes dos *buckets*
  - Metadados dos objetos
  - Consistência, boas práticas e limites

Armazenamento de Dados: Anos 90 – 2010

- Necessidade de armazenar dados em ficheiros de vários formatos, partilhados entre computadores das redes locais (*intranets*) das organizações;
- A partir dos File System dos sistemas operativos (tipicamente organizados num espaço de nomes hierárquico em árvore) foram desenvolvidos mecanismos para suportar, de forma transparente, a extensão do espaço de nomes integrando múltiplos computadores, criando um espaço de nomes global baseado em subárvores:
  - NFS (Network File System)
  - SAN (Storage Area Network - high speed network that makes connections between storage devices and servers) & NAS (Network Attached Storage - shares files over the network)
  - HDFS (Hadoop Distributed File System)

Exemplo do NFS

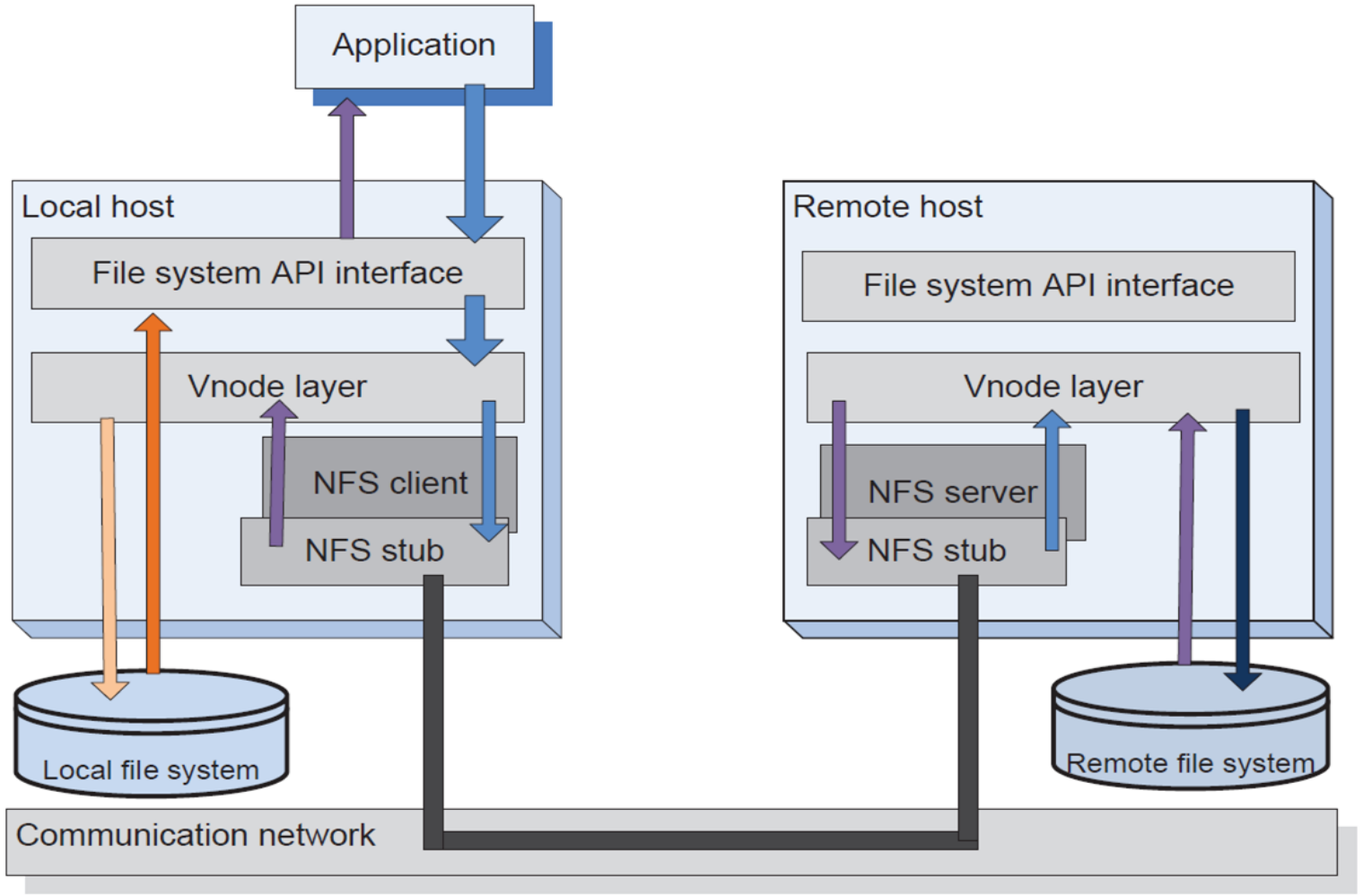


- Um utilizador na máquina *Client* vê, de forma transparente, uma árvore global que envolve o acesso a 3 *file systems*, permitindo operações sobre ficheiros remotos com *pathnames* hierárquicos
- Por exemplo o *pathname* `/usr/students/bob` permite nomear ficheiros que existam na diretoria `/export/people/bob` do servidor 1

adaptado de "Distributed Systems: Concepts and Design", George Coulouris

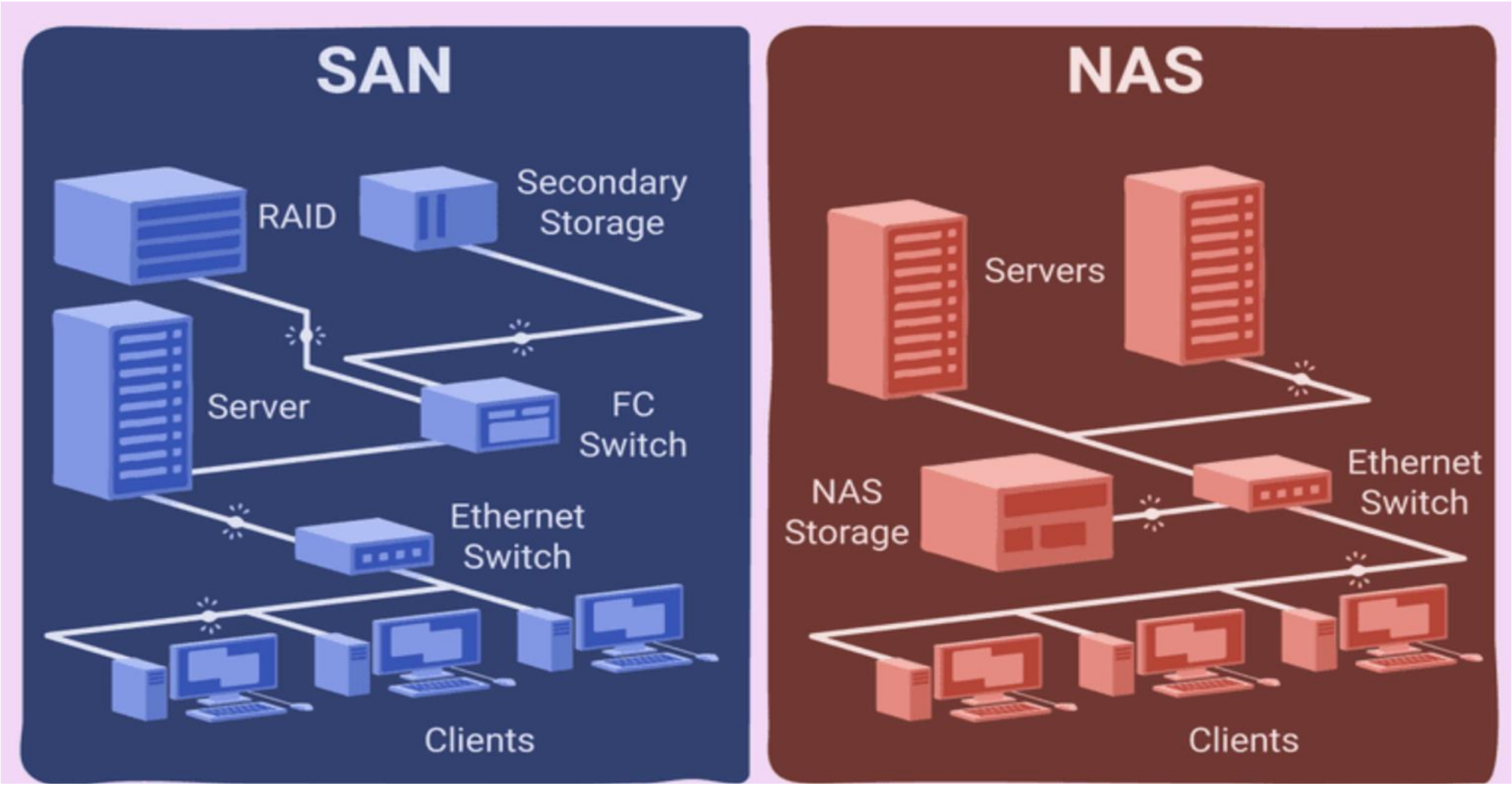
Network File System (NFS, 1984)

- Integração com file system UNIX, mantendo compatibilidade das aplicações



adaptado de "Cloud Computing: Theory and Practice: Dan C. Marinescu, 2012"

(*Storage Area Network*) vs (*Network Attached Storage*)



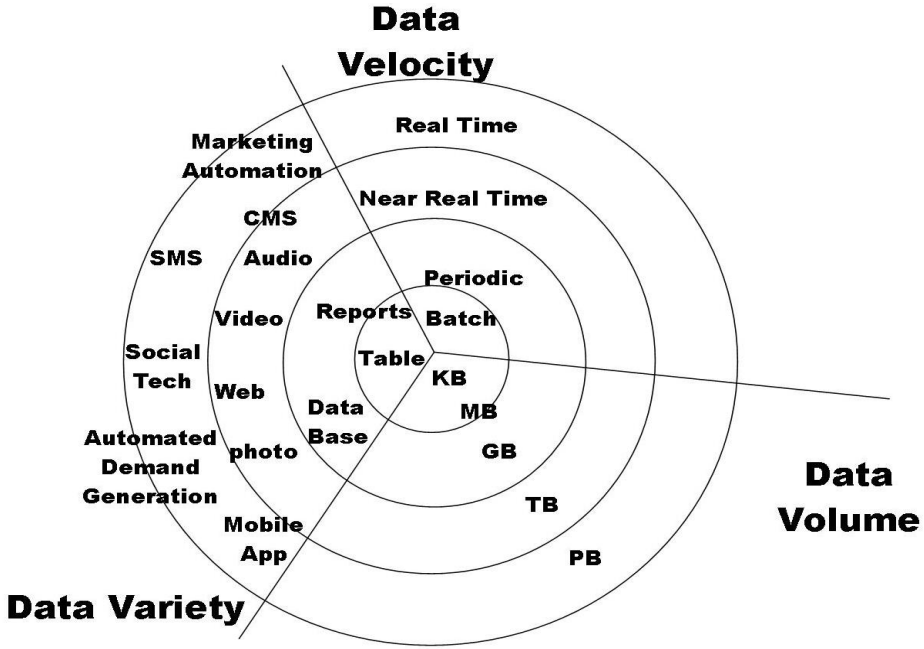
https://www.lifewire.com/san-vs-nas-818005

Dificuldades dos *File System* hierárquicos

- Múltiplos acessos a disco para localizar os blocos do ficheiro
- Tempos de acesso Read/Write altos quando a profundidade da árvore aumenta
- Dificuldades de manter réplicas consistentes
- Dificuldades de manter versões de ficheiros (uma versão é tipicamente uma cópia integral do ficheiro)
- Em NFS, SAN e NAS:
  - *Bottlenecks* no lado dos servidores de controlo do storage;
  - Utilização de *cache* em memória, com perda de desempenho perante cenários de escritas frequentes

Dimensões *Big Data*

- Grande capacidade de processamento dos dados
- Grande diversidade de fontes e tipos de dados
- Grande dimensão de dados



https://velvetchainsaw.com/2012/07/20/three-vs-of-big-data-as-applied-conferences/

- Redução do tempo de acesso aos dados, suportando distribuição de conteúdos em *streaming*;
- Replicação em larga escala para aumentar disponibilidade e tolerância a falhas, assumindo cenários em que os *updates* não são frequentes e portanto não degrada o desempenho;

Google File System (GFS'03): requisitos

- Escalabilidade e tolerância a falhas
- Ficheiros de grandes dimensões (GB até TB)
- A leitura sequencial dos ficheiros é o mais frequente
- Escrita aleatória do conteúdo dos ficheiros é rara. Quando muito é sempre uma operação *append*
- Privilegiar o processamento de grande quantidade de dados sem preocupação do tempo de *read*/*write* de ficheiros individuais
- Relaxar a consistência, privilegiando a replicação para aumentar disponibilidade

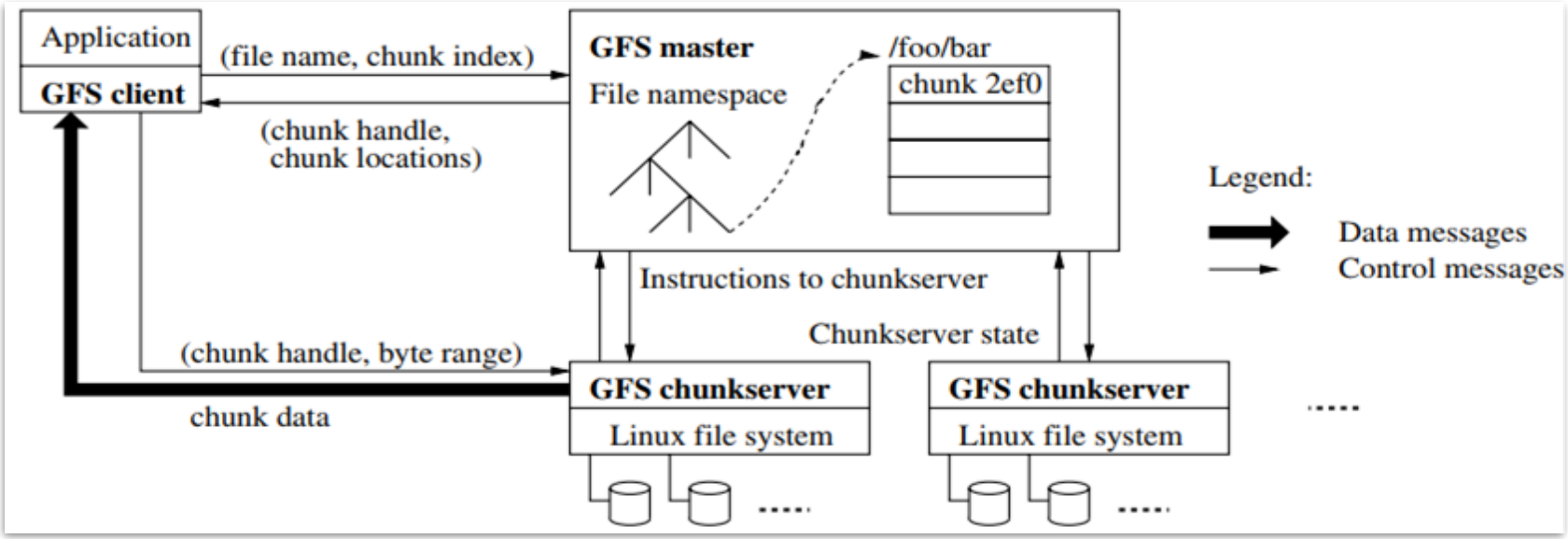
Arquitetura do GFS

- Um ficheiro é segmentado em *chunks*
- Só a operação *append* no mesmo ficheiro é atómica suportando concorrência
- Eliminar *cache* do lado do cliente para facilitar a garantia de consistência;
- Assegurar consistência de operações críticas (*write*, *delete*, etc.) através de um componente centralizado (*master*) que controla todo o sistema;
- Suporte para *checkpoints*, mecanismos de recuperação e *garbage collection* eficientes;

Em GFS um ficheiro é uma coleção de *chunks* de tamanho fixo, armazenados no *file system* UNIX e replicados em múltiplos servidores

https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf

Google File System (GFS, 2003)



- O *master* controla um conjunto de *chunk servers*, mantendo de forma persistente a *metadata* do ficheiro (nome, ACL; localização das réplicas dos *chunks* de cada ficheiro, e estado de cada *chunk server*);
- Para recuperar de falhas o *master* mantém um *Log* com *checkpoints* e as operações, podendo repetir as operações desde o *checkpoint* anterior;
- Os *chunk servers* são Linux. Cada *chunk* é um ficheiro com um *handle*;
- Uma aplicação para fazer *Read/Write* acede ao *master* que lhe devolve um *handle* para o *chunk* e a localização do mesmo num *chunk server*.