Pesquisas em MDS e CEPH

Tópicos em Redes de Computadores (INFO-7065)

Josiney de Souza (josiney.souza@ifc.edu.br)

UFPR / DInf

6 de Junho de 2023

Sumário

Sumário

Visão geral de sistemas de arquivos

- Computadores usam Sistemas Operacionais (SO's)
- Em nível físico, SO's estão em memória (RAM, HD, *flash*, SSD, memória persistente)
- Em nível lógico, SO's usam sistemas de arquivos (*file systems* - FS)
- Podem ser locais ...
  - EXT4
  - XFS
  - NTFS

Visão geral de sistemas de arquivos

- Computadores usam Sistemas Operacionais (SO's)
- Em nível físico, SO's estão em memória (RAM, HD, *flash*, SSD, memória persistente)
- Em nível lógico, SO's usam sistemas de arquivos (*file systems* - FS)
- Podem ser locais ...
  - EXT4
  - XFS
  - NTFS
- ou remotos:
  - NFS
  - CEPH
  - EOS
  - GlusterFS
  - Lustre

Sumário

Sistemas distribuídos

Sobre a arquitetura e organização:

Centralizado clientes e servidores;

Descentralizado P2P (pode ser cliente e servidor ao mesmo tempo);

Híbrido P2P com componentes centralizados.

Sistemas distribuídos

Sobre a arquitetura e organização:

Centralizado clientes e servidores;

Descentralizado P2P (pode ser cliente e servidor ao mesmo tempo);

Híbrido P2P com componentes centralizados.

Sobre o modelo temporal:

assíncrono não se fala em tempo;

parcialmente síncrono entre assíncrono e síncrono. Algumas asserções podem ser feitas;

síncrono sabe-se o tempo máximo para transmissão de mensagem e execução de tarefas.

Sistemas distribuídos

Sobre o modelo de falhas:

Crash	cessa a computação e para de responder
Omissão	falha em responder (não processa o resultado ou não o trata)
Bizantino	age diferente do normal ou esperado (podem ser nodos maliciosos)

Sistemas distribuídos

Sobre o modelo de falhas:

Crash cessa a computação e para de responder

Omissão falha em responder (não processa o resultado ou não o trata)

Bizantino age diferente do normal ou esperado (podem ser nodos maliciosos)

Sobre o número de nodos para um limiar "f" de falhas:

Tipo de falha	**Número de nodos**
Crash	$2f + 1$
Bizantina	$3f + 1$

O consenso

O consenso em sistemas distribuídos:

- é o problema de múltiplos nodos concordarem sobre uma sequência de valores
- algoritmo de consenso faz com que os nodos corretos executem o mesmo comando, na mesma ordem e cheguem ao mesmo estado

O consenso

O consenso em sistemas distribuídos:

- é o problema de múltiplos nodos concordarem sobre uma sequência de valores
- algoritmo de consenso faz com que os nodos corretos executem o mesmo comando, na mesma ordem e cheguem ao mesmo estado

Modelo de falhas	**Algoritmo de Consenso**	**Consenso de valor**
Crash	Paxos (1989, 1998)	Único
	Multi-Paxos	Geral
	Cheap Paxos	Geral
	Fast Paxos	Geral
	Raft (2014)	Geral
Bizantino	Byzantine Paxos	Único
	PBFT	Geral

Sumário

Por que metadados?

Por que se pensaria em metadados?

- *Big data* e *cloud* estão em alta
- Escala de TiB, PiB, EiB
- Aplicações demandam armazenamento
- É necessário gerenciar os dados e seus metadados

Os metadados

Em sistemas de arquivos, em relação aos metadados:

- são os dados responsáveis por manter o *namespace*, semântica de permissões e localização dos dados dos arquivos;
- operações com eles podem representar até 80% das operações totais;
- representam apenas 1% do dado em si;
- informações sobre tipo, tamanho, estado;
- informações sobre permissões, propriedade, tempos;
- informações sobre os blocos dos dados em si.

Os metadados

TABLE 1
File Read Timeline for a One-Block Sized File `/foo/bar` [11]
(Time Increasing Downward)

	Metadata			Data		
	root inode	foo inode	bar inode	root data	foo data	bar data
open (bar)	read	read	read	read	read	
read()			read write			read

Fonte: [Dai et al., 2022]

Os metadados: cargas de trabalho

Estudos observaram o seguinte sobre a carga de trabalho dos sistemas:

- para o refinamento de resultados, mais de 95% das buscas são para vários atributos de metadados
- 33% das buscas são limitadas para uma área relacionada do *namespace*
- quase 25% das buscas que os usuários pensam ser importantes são para várias versões do metadado

Os metadados: cargas de trabalho

Estudos observaram o seguinte sobre a carga de trabalho dos sistemas:

- para o refinamento de resultados, mais de 95% das buscas são para vários atributos de metadados
- 33% das buscas são limitadas para uma área relacionada do *namespace*
- quase 25% das buscas que os usuários pensam ser importantes são para várias versões do metadado
- a maioria dos valores dos atributos estão sob localidade espacial (ex.: /home/jsouza)
- há um grau de assimetria alto e valores predominantes ocupam mais do espaço de valor (ex.: 80% dos arquivos têm mais valores em comum que os outros 20% para os atributos `ext` e `size`)

Os metadados: cargas de trabalho

Estudos observaram o seguinte sobre a carga de trabalho dos sistemas:

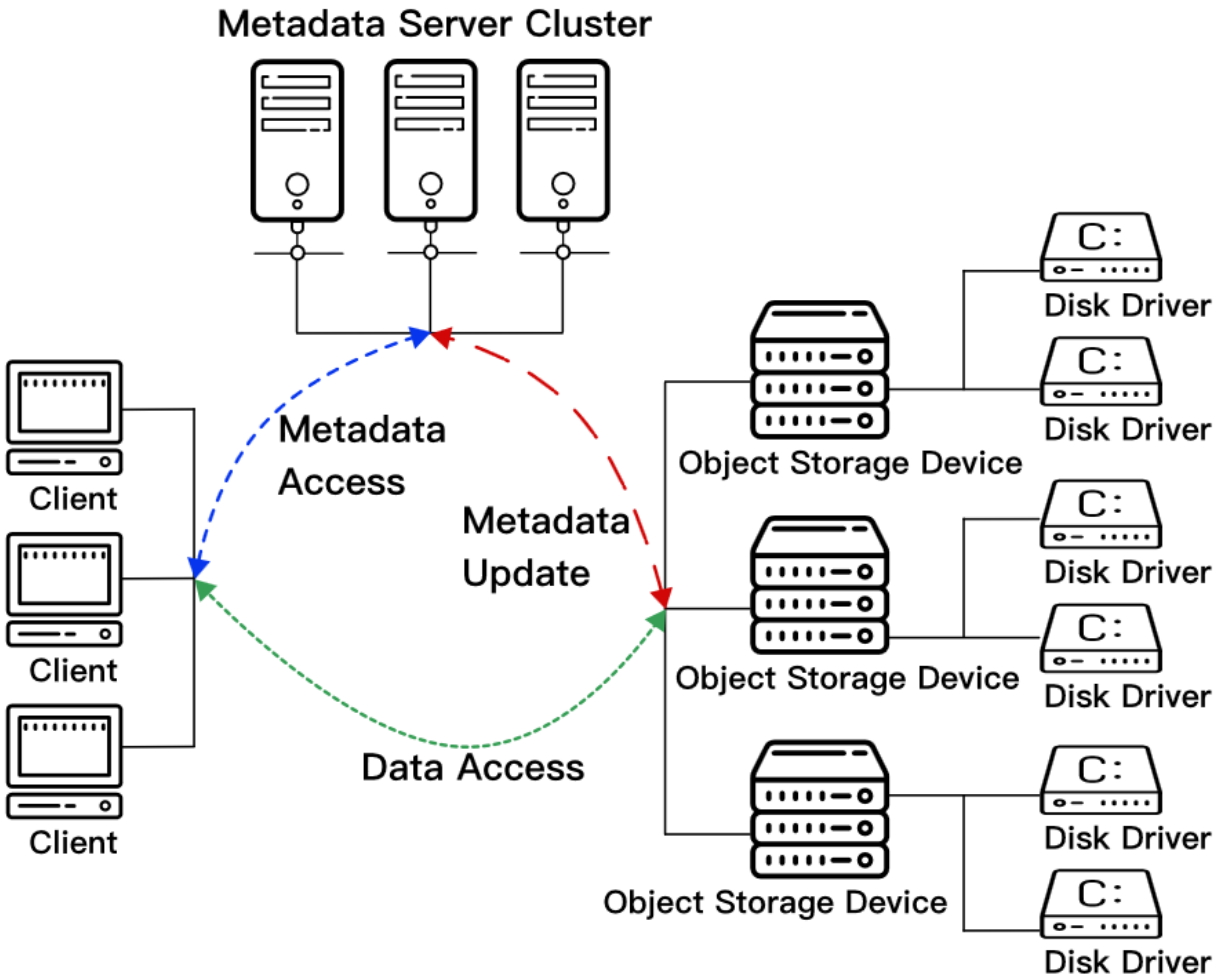
- de 60 sistemas de arquivos, 90% dos diretórios têm menos que 128 entradas de diretórios
- diretórios grandes continuam a crescer com o aumento da capacidade do armazenamento
- 64% das mídias têm menos que 64 KiB
- 90% das árvores de diretórios não passam de profundidade 16
- na manipulação, geralmente se executa `open` e/ou `readdir`

Os metadados: arquiteturas

Acerca de um MDS (MetaData Server), as arquiteturas de gerenciamento utilizadas para os metadados são:

- Sem MDS, no mesmo servidor dos dados
- Com um MDS único
- Com um MDS distribuído

Os metadados: arquiteturas



Fonte: [Dai et al., 2022]

Os metadados: campos de pesquisa

São campos de pesquisa atualmente:

- Alta escalabilidade
- Alto desempenho
- Alta disponibilidade

Alta escalabilidade

É feita através da divisão de espaços.

- A divisão dos metadados podem ser sobre algumas informações
  - nome do caminho
  - tamanho da subárvore
- Características:
  - Localidade
  - Balanceamento de carga
- Métodos usados:
  1. Estáticos
  2. Dinâmicos

Alta escalabilidade: (1) métodos estáticos

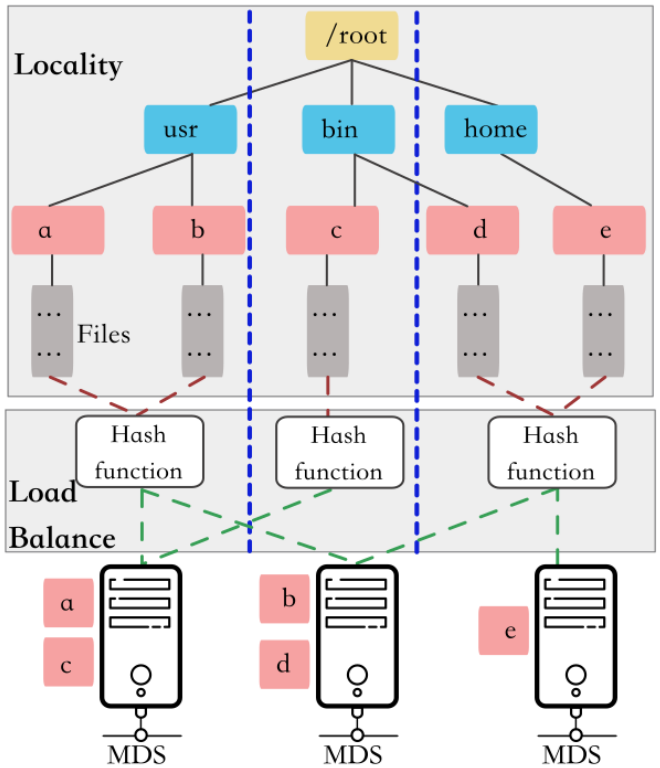
Modelos dos métodos estáticos:

- **Particionamento de subárvore**
  - NFS
  - administrador decide como dividir os fragmentos
  - aumenta a localidade
  - não consegue lidar bem com desbalanceamento
- **Mapeamento baseado em hash**
  - aplicar função de *hash* sobre o arquivo
  - descobre-se o MDS
  - aumenta o balanceamento
  - tende à baixa localidade

Alta escalabilidade: (1) métodos estáticos

Modelos dos métodos estáticos:

- **Híbrido (subárvore + hash)**
  - teoricamente, o melhor de dois mundos



  - mesmo assim, sobrecarga de rede na migração de metadados

Alta escalabilidade: (2) métodos dinâmicos

Modelos dos métodos dinâmicos:

- Divisão dinâmica dos espaços
- Na mudança da carga de trabalho, dados são rearranjados entre MDS;
- Modelos:
  1. Subárvore
  2. Subconjunto de itens de diretório
  3. *Hashing*

Alta escalabilidade: (2) métodos dinâmicos

Modelos dos métodos dinâmicos:

- **Subárvore**
  - Migrar cargas de trabalho dos nodos sobrecarregados
  - Decaimento de contador de tempo exponencial
  - O *load* é comparado frequentemente
  - Sofre com migração de dados não controlada
- **Subconjunto de itens de diretório**
  - Cada diretório em um servidor primário aleatório
  - Em um limiar, divide em subconjuntos em secundários aleatórios
  - LSM-tree, LevelDB, Sorted String Table (SSTable)
  - Depende fortemente da exploração de *cache*
- **Hashing**
  - Usa *hash* e as posições dos bits do *hash*
  - Balanceado em armazenamento mas com problemas com *hotspots*
  - Estima a frequência de acessos de dados
  - Existem trabalhos com *Software Defined Network* (SDN)

Alta escalabilidade: sumarização

Sumarizando os métodos de alta escalabilidade:

Métrica	Estático			Dinâmico		
	Subárvore	*Hash*	Híbrido	Subárvore	Diretório	*Hashing*
Hotspot	Sim	Não	Não	Sim	Sim	Sim
Bal. Carga	Ruim	Bom	Bom	Bom	Bom	Bom
Localidade	Bom	Ruim	Bom	Bom	Bom	Ruim
Escalab.	Custa	Custa	Custa	B. custo	Custa	Custa
Troca nome	Custa	Custa	Custa	Custa	Custa	Custa

Fonte: adaptado de [Dai et al., 2022]

Alto desempenho

Os servidores de metadados podem considerar o alto desempenho sob as seguintes óticas:

1. *Cache* e replicação
2. Recuperação de metadados
3. Metadados de valor agregado

Alto desempenho: (1) *cache* e replicação

Em relação a *cache* e replicação:

- *Caching*
  - cooperativo
  - no servidor ou no cliente
  - *write-through* ou *write-back*
- Nós de autoridade
- Busca pelo nome do caminho é o maior impacto
- Alguns estudos assumem que dados são idempotentes
- Em algum momento, metadados podem estar desbalanceados

Alto desempenho: (2) recuperação de metadados

Em relação a recuperação de metadados:

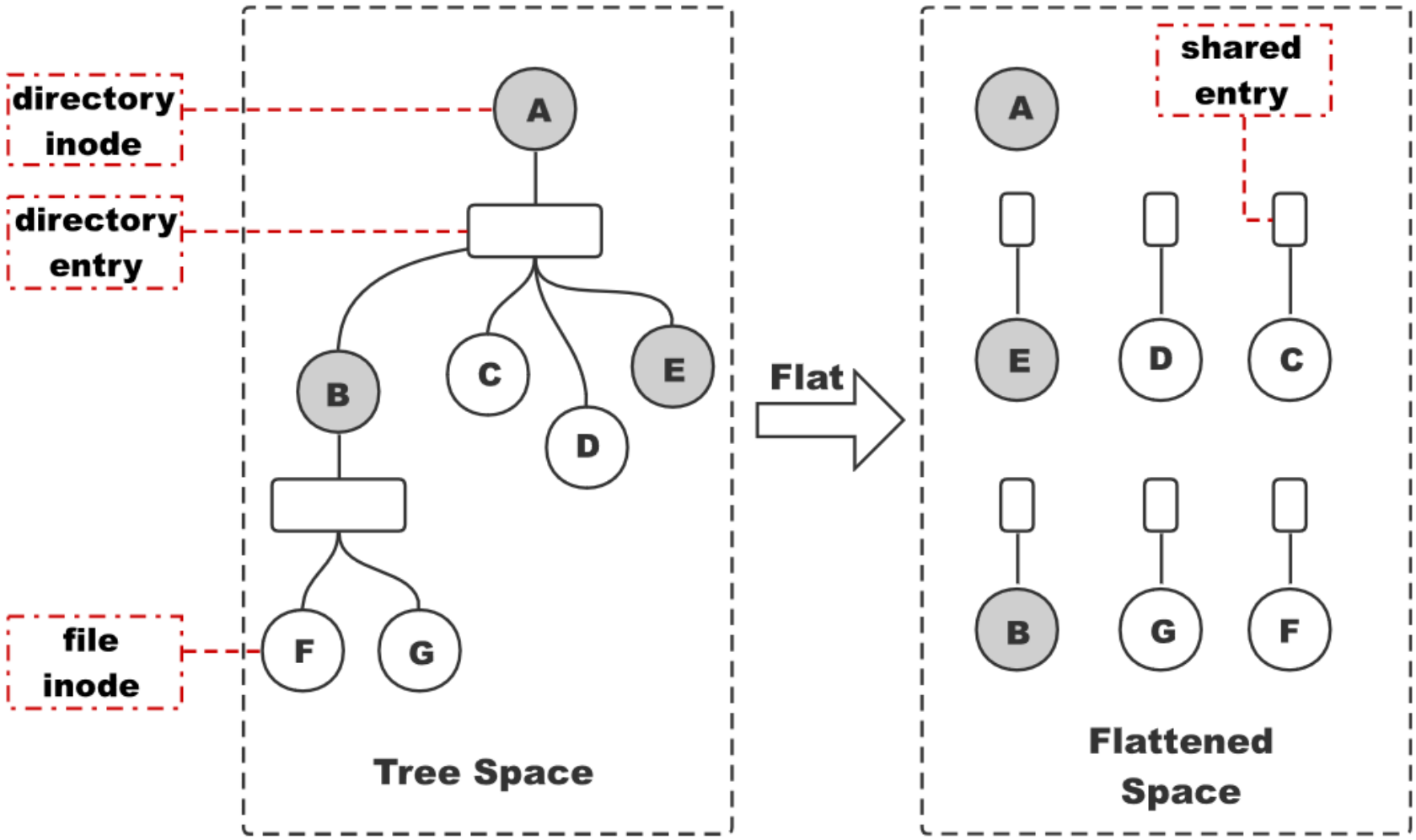
Árvore de indexação (*index tree*). Usa árvore para indexar os metadados e pode fazer controle de versão;

Filtros de Bloom (*Bloom filter*). Técnica para testar se um elemento é membro de um conjunto. Pode ter falsos positivos (possivelmente existe) mas não falsos negativos (definitivamente não existe). Exemplos: Chrome, Bing, Squid, Bitcoin, Ethereum;

Pré-busca (*pre-fetching*). *Provenance* (proveniência, histórico). Correlação entre dados, metadados e processos. Pode construir grafos e árvores para a navegação;

Banco de dados chave-valor (*key-value database*). Poucos o usam. LevelDB, RocksDB. Apesar dos benefícios, a dependência entre os diretórios é um empecilho. Espaço plano.

Alto desempenho: (2) recuperação de metadados



Fonte: [Dai et al., 2022]

Alto desempenho: (3) metadados de valor agregado

Em relação a metadados de valor agregado:

- Feito sobre o gerenciamento padrão de metadados
- Aproveitar metadados para outras aplicações e consultas
- Registrar resultados intermediários que seriam descartados
- Informações de dependências entre arquivos
- "Metadados ricos"
- Escalonador de *workflow* pode se beneficiar

Alta disponibilidade

A **alta disponibilidade** dos *clusters* de MDS podem ser entendidas sob os seguintes olhares:

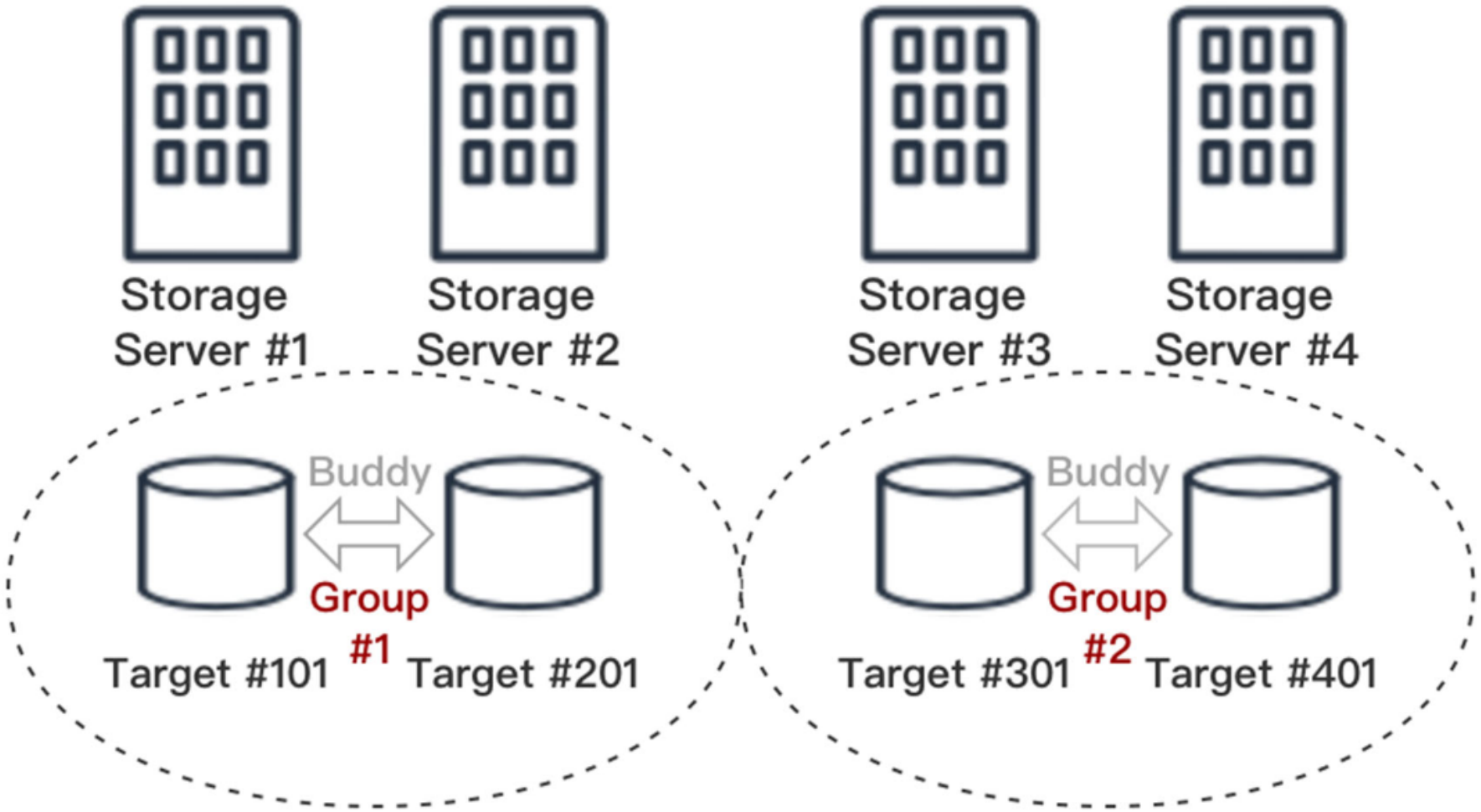
1. Baseada em cópia
2. Baseada em *log*

Alta disponibilidade: (1) baseada em cópia

O que é, prós e contras:

- Redundância
- Fazer cópia dos primários nos secundários
- Último ponto de salvamento
- Pode gerar inconsistências
- É possível adicionar técnicas de *logs* para complementar
- Fazer troca em tempo-real
- Necessário consenso entre as partes

Alta disponibilidade: (1) baseada em cópia



Fonte: [Dai et al., 2022]

Alta disponibilidade: (2) baseada em *log*

Para se evitar I/O, se usa *cache*.
Então *logs* podem ser alternativa para não se perder essas informações.
Sobre a disponibilidade baseada em *log*:

- *Logs* com travas/*locks* distribuídos
- Em sistemas com discos compartilhados, pode levar a "hot files"
- Em sistemas sem compartilhamentos (*shared-nothing*), operações devem entrar nas transações
- Também deve haver consenso para se ter consistência
- Paxos ou Multi-Paxos são complexos e difíceis de implementar
- Raft tem sido usado

Sumário

Visão geral do CEPH

O sistema CEPH é:

- sistema de armazenamento distribuído
- código-fonte aberto
- Sage A. Weil. **Ceph: Reliable, Scalable, and High-Performance Distributed Storage**. PhD thesis, University of California, Santa Cruz, 2007.
- `https://docs.huihoo.com/ceph/Ceph-Reliable-Scalable-and-High-Performance-Distributed-Storage.pdf`
- Suportado por Red Hat, Canonical, Intel
- Usado pelo CERN
  - Large Hadron Collider (LHC)
  - gera 25 PiB por ano

OFF-Topic: LHC e buraco negro

Fonte: `https://www.youtube.com/watch?v=9JYkMhQ9gf8`
`lhc-buraco-negro.mp4`

OFF-Topic: LHC e buraco negro

Em 2008 ...

Para Hawking, projeto que 'recria Big Bang' não ameaça a Terra

O físico britânico Stephen Hawking afirma que não há perigo de que, ao ser acionado nesta quarta-feira, um gigantesco acelerador de partículas instalado numa área subterrânea da fronteira franco-suíça, possa criar um buraco negro capaz de engolir o planeta (e o resto do sistema solar) em questão de minutos - como temem alguns cientistas.

`https://www.bbc.com/portuguese/reporterbbc/story/2008/09/080909_lhc_hawking_mv`

OFF-Topic: LHC e buraco negro

Em 2013 ...

Algum dia o acelerador de partícula da CERN poderá criar um buraco negro?

Embora seja possível criar um "micromini" buraquinho negro através do acelerador de partículas, nunca — jamais — a CERN poderá criar um buraco negro poderoso o suficiente para engolir a Terra. Segundo o site, o buraquinho seria tão pequenino que seria difícil forçá-lo a engolir um único elétron.

`https://www.megacurioso.com.br/experiencias/36332-algum-dia-o-acelerador-de-particula-da-cern-podera-criar-um-`
`htm`

OFF-Topic: LHC e buraco negro

Em 2019 ...

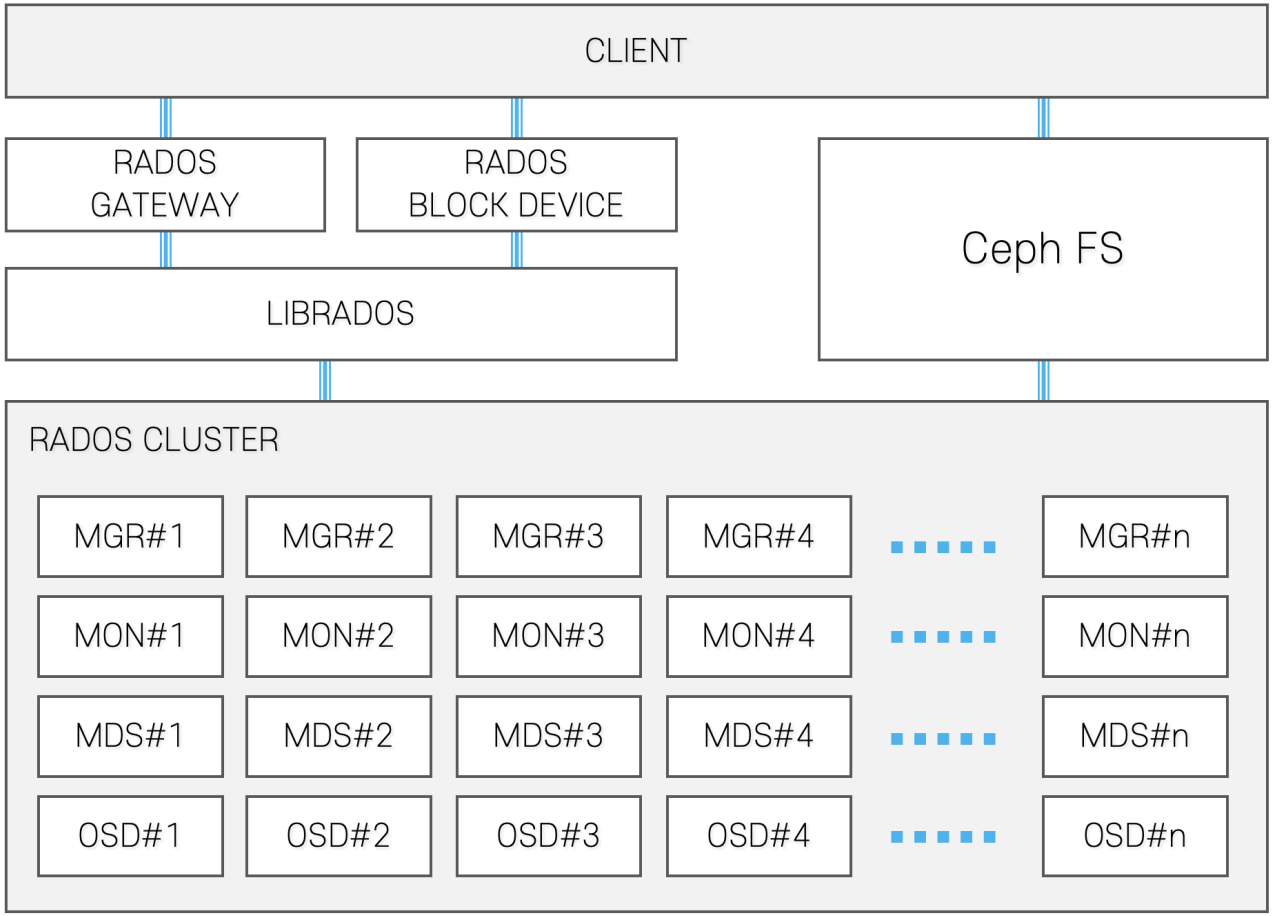
Fonte: `https://i.ytimg.com/vi/pw-56d2qfKM/maxresdefault.jpg`

`https://www.youtube.com/watch?v=pw-56d2qfKM`

OFF-Topic: LHC e buraco negro

Fonte: https://hypescience.com/wp-content/uploads/2019/04/meme-buraco-negro-publicidade-1.jpg

Arquitetura do CEPH



Fonte: [Lee et al. 2021]

Arquitetura do CEPH - Parte servidor

Fazem parte do *Cluster* RADOS (*Reliable Autonomous Distributed Object Storage*):

MGR Gerenciador (*Manager*). Monitora e orquestra sistema (exemplo, carga/*load* e utilização de armazenamento), mesmo externo via painel web (*dashboard*). Recomendados pelo menos 2;

Arquitetura do CEPH - Parte servidor

Fazem parte do *Cluster* RADOS (*Reliable Autonomous Distributed Object Storage*):

MGR Gerenciador (*Manager*). Monitora e orquestra sistema (exemplo, carga/*load* e utilização de armazenamento), mesmo externo via painel web (*dashboard*). Recomendados pelo menos 2;

MON Monitor. Mantém mapas dos recursos do cluster. Recomendado ter entre 3 e 7. Usa Multi-Paxos;

Arquitetura do CEPH - Parte servidor

Fazem parte do *Cluster* RADOS (*Reliable Autonomous Distributed Object Storage*):

MGR Gerenciador (*Manager*). Monitora e orquestra sistema (exemplo, carga/*load* e utilização de armazenamento), mesmo externo via painel web (*dashboard*). Recomendados pelo menos 2;

MON Monitor. Mantém mapas dos recursos do cluster. Recomendado ter entre 3 e 7. Usa Multi-Paxos;

OSD *Daemon* de *Object Storage Devices*. Acompanha o estado dos objetos, replicação, recuperação, rebalanceamento. Recomendado pelo menos 3;

Arquitetura do CEPH - Parte servidor

Fazem parte do *Cluster* RADOS (*Reliable Autonomous Distributed Object Storage*):

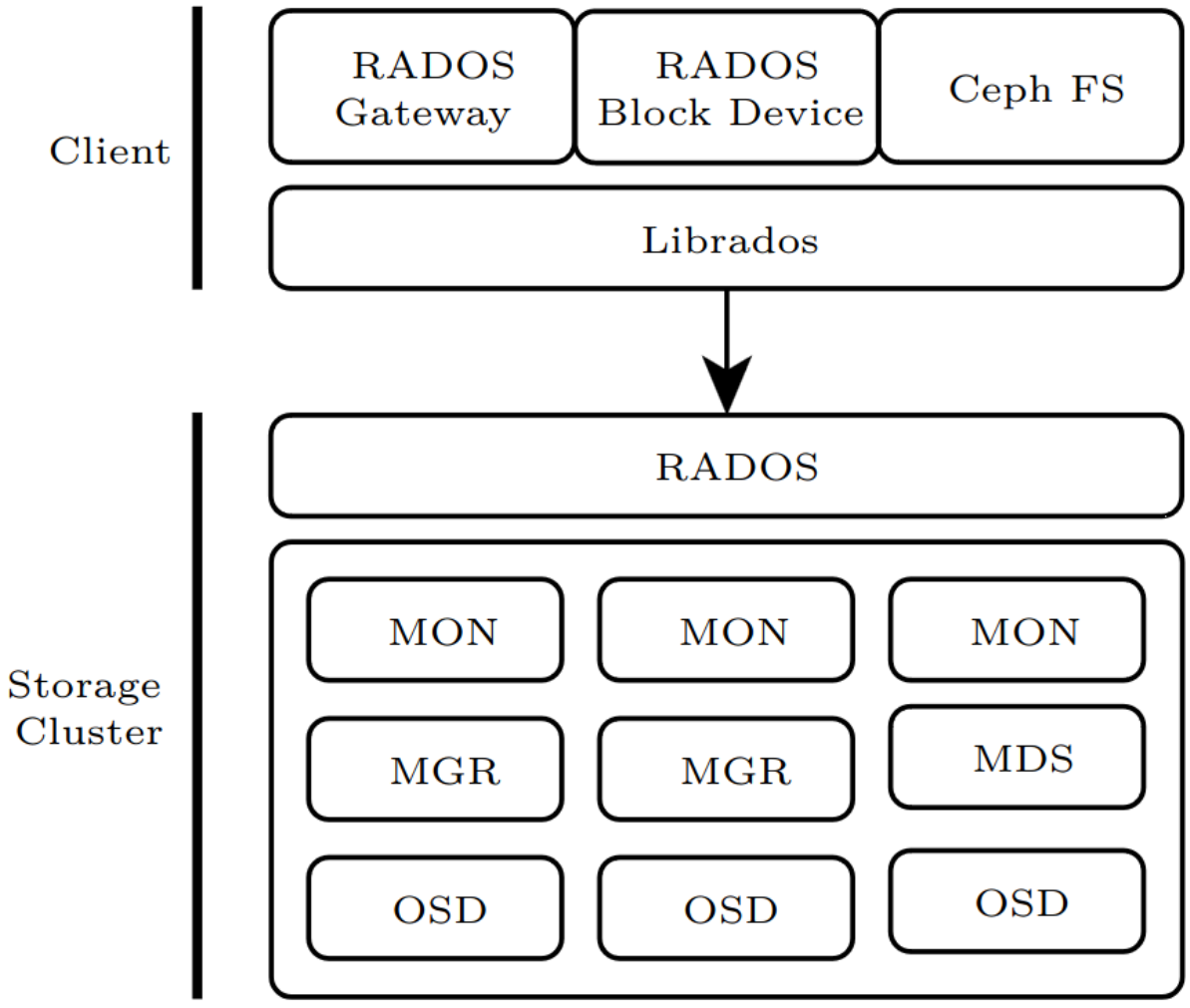
MGR Gerenciador (*Manager*). Monitora e orquestra sistema (exemplo, carga/*load* e utilização de armazenamento), mesmo externo via painel web (*dashboard*). Recomendados pelo menos 2;

MON Monitor. Mantém mapas dos recursos do cluster. Recomendado ter entre 3 e 7. Usa Multi-Paxos;

OSD *Daemon* de *Object Storage Devices*. Acompanha o estado dos objetos, replicação, recuperação, rebalanceamento. Recomendado pelo menos 3;

MDS Servidor de metadados. Gerencia metadados do CephFS, se estiver presente na instância. Aceita comando como ls(1) e find(1).

Arquitetura do CEPH



Fonte: [Fernandes, 2021]

Arquitetura do CEPH - Parte cliente

Parte cliente do sistema RADOS (*Reliable Autonomous Distributed Object Storage*):

Librados Biblioteca RADOS. Permite a comunicação com o *Cluster* RADOS. Pode se comunicar diretamente com os OSD's;

Arquitetura do CEPH - Parte cliente

Parte cliente do sistema RADOS (*Reliable Autonomous Distributed Object Storage*):

Librados Biblioteca RADOS. Permite a comunicação com o *Cluster* RADOS. Pode se comunicar diretamente com os OSD's;

RBD Dispositivo de Blocos de Dados (Rados Block Device). Provisiona dispositivos virtuais de blocos. Compatível com máquinas virtuais e Kubernates;

Arquitetura do CEPH - Parte cliente

Parte cliente do sistema RADOS (*Reliable Autonomous Distributed Object Storage*):

Librados Biblioteca RADOS. Permite a comunicação com o *Cluster* RADOS. Pode se comunicar diretamente com os OSD's;

RBD Dispositivo de Blocos de Dados (Rados Block Device). Provisiona dispositivos virtuais de blocos. Compatível com máquinas virtuais e Kubernates;

RGW Interface de armazenamento de objetos (Rados Gateway). Disponibiliza API para armazenar objetos e metadados. Compatível com Amazon S3 e OpenStack Swift;

Arquitetura do CEPH - Parte cliente

Parte cliente do sistema RADOS (*Reliable Autonomous Distributed Object Storage*):

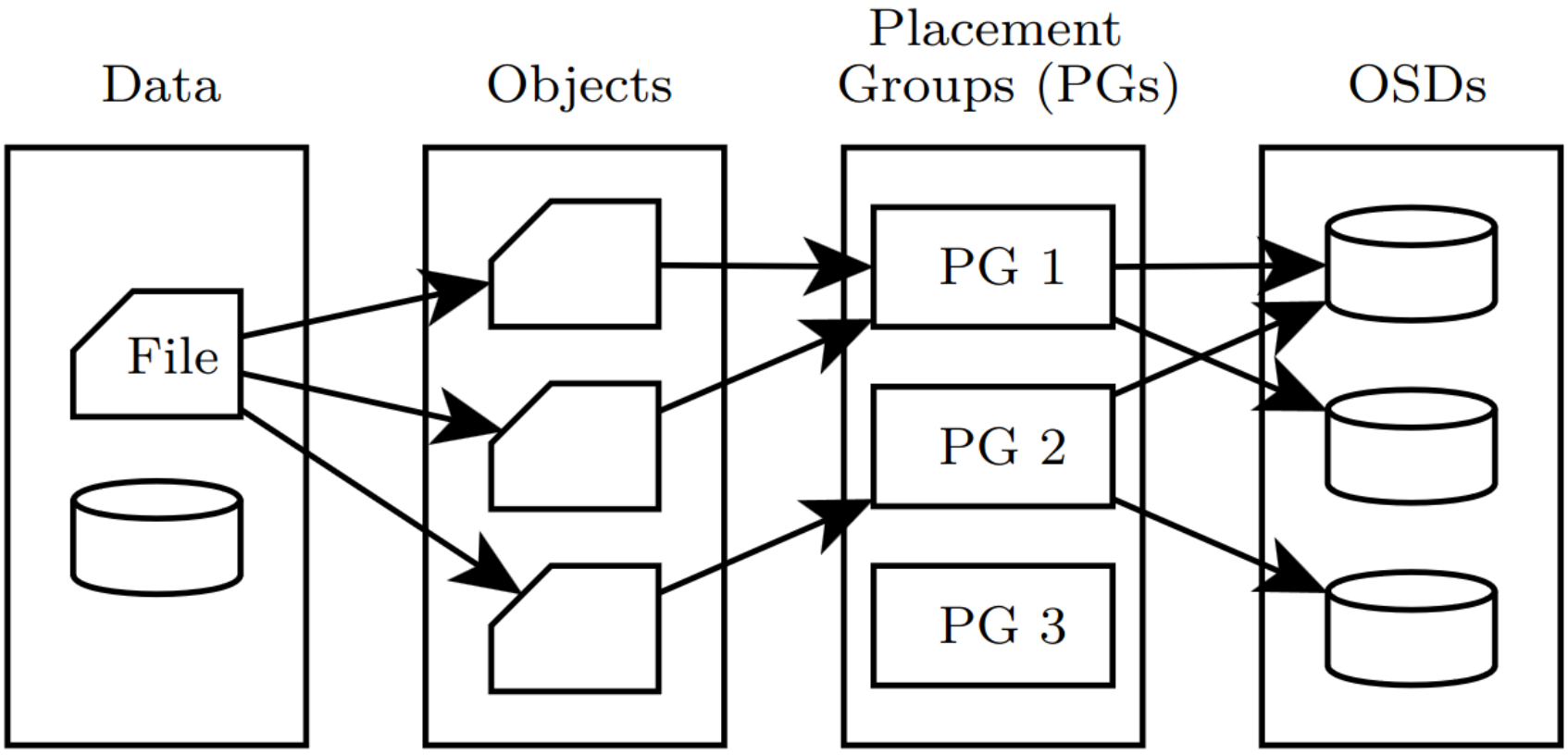
Librados Biblioteca RADOS. Permite a comunicação com o *Cluster* RADOS. Pode se comunicar diretamente com os OSD's;

RBD Dispositivo de Blocos de Dados (Rados Block Device). Provisiona dispositivos virtuais de blocos. Compatível com máquinas virtuais e Kubernates;

RGW Interface de armazenamento de objetos (Rados Gateway). Disponibiliza API para armazenar objetos e metadados. Compatível com Amazon S3 e OpenStack Swift;

CephFS Sistema de arquivos. Para armazenar dados. Usa o MDS. Compatível com POSIX: ls(1), find(1).

Armazenamento de dados



Fonte: [Fernandes, 2021]

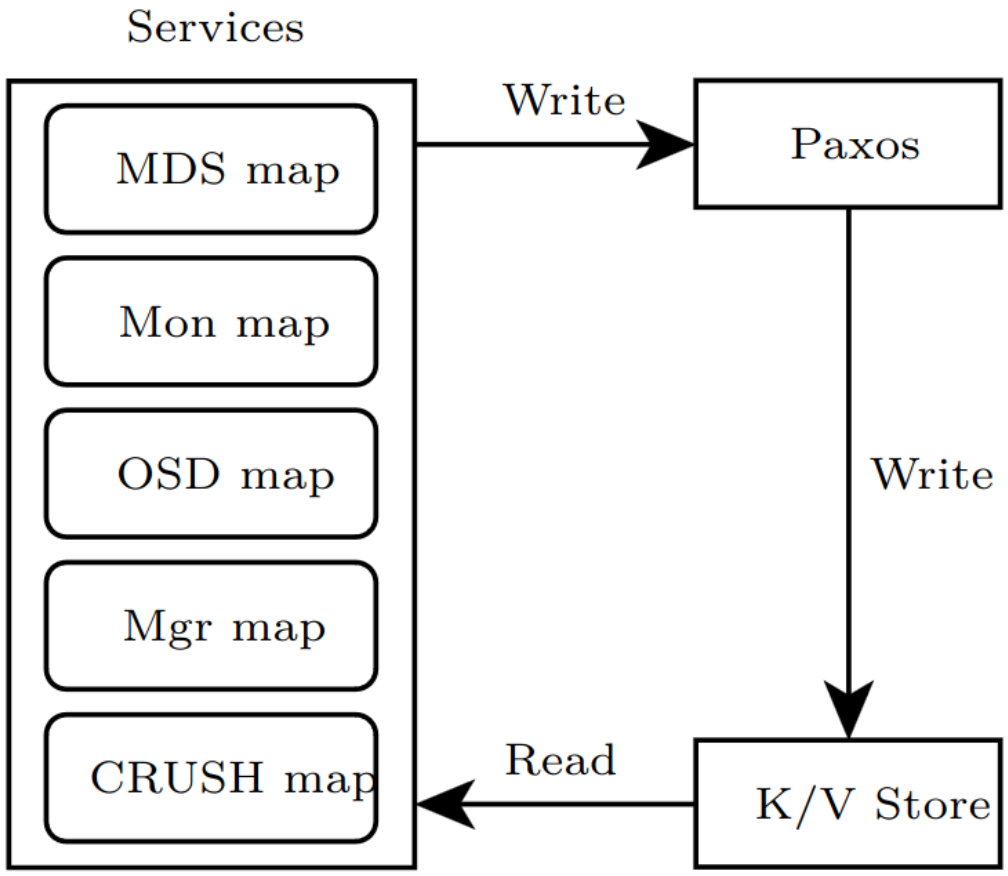
Armazenamento de dados

O armazenamento de dados assim ocorre:

1. Os dados são divididos em objetos menores;
2. Cada objeto é adicionado a um agrupamento (*pool* de armazenamento) para compor um Grupo de Colocação (*Placement Groups - PG*) disponível;
3. Os PG são atribuídos a um OSD ...
    - Sob supervisão do algoritmo CRUSH (*Controlled Replication Under Scalable Hashing*)
    - Domínio de falhas
    - Número de réplicas
    - Estratégia de replicação
    - Tipo de armazenamento (exemplo: HDD, SSD)

Consenso no CEPH

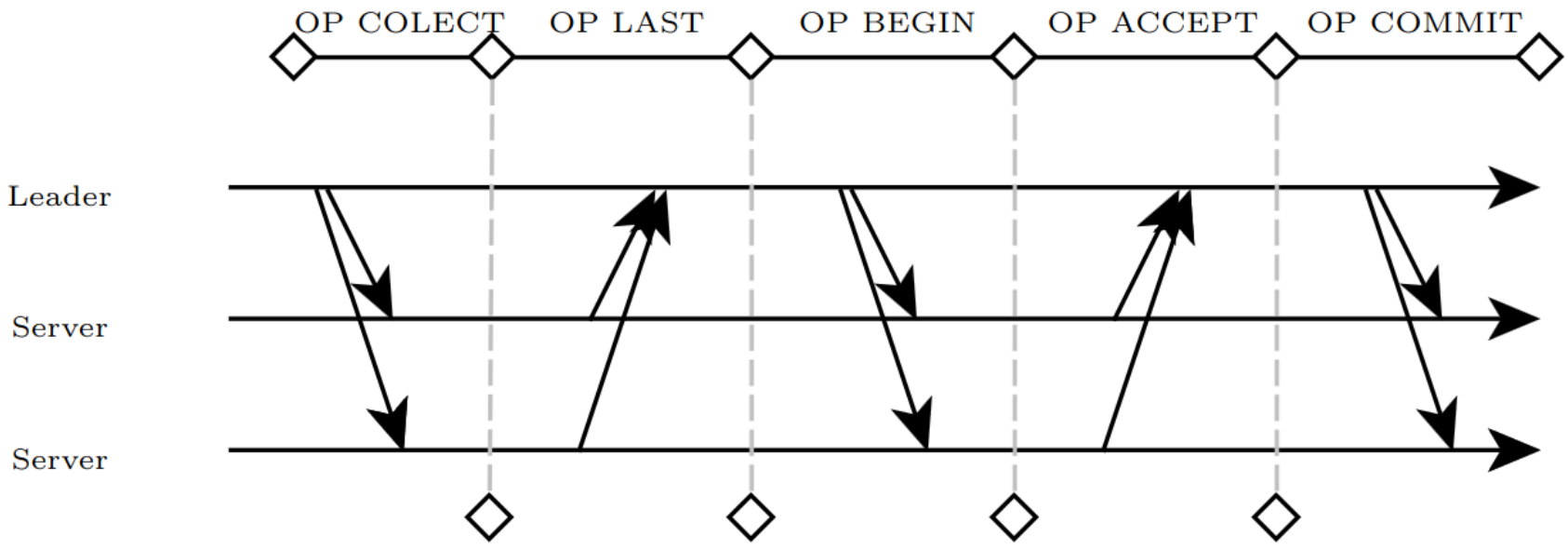
O consenso no CEPH ocorre nos monitores (MON) e usa Multi-Paxos:



Fonte: [Fernandes, 2021]

Consenso no CEPH

O consenso no CEPH ocorre nos monitores (MON) e usa Multi-Paxos:



Fonte: [Fernandes, 2021]

Sumário

Conclusão

Este trabalho apresentou ...

- uma introdução sobre sistemas de arquivos
- uma revisão sobre sistemas distribuídos
- um *survey* sobre MDS
- o sistema CEPH

Trabalhos futuros

Meus trabalhos futuros:

- Realizar mais pesquisa sobre o estado da arte
- Acompanhar as defesas de TCC e de qualificação
- Fazer testes iniciais com o CEPH e o MDS
  - MDS baseado em IA
  - Novas mídias de armazenamento
  - Serviços sem metadados

Trabalhos futuros

Meus trabalhos futuros:

- Realizar mais pesquisa sobre o estado da arte
- Acompanhar as defesas de TCC e de qualificação
- Fazer testes iniciais com o CEPH e o MDS
  - MDS baseado em IA
  - Novas mídias de armazenamento
  - Serviços sem metadados

Trabalhos futuros desejáveis:

- Substituir o NFS por CEPH no DInf/UFPR
- Contribuir com a comunidade acadêmica
- Escrever artigos e TCCs, dissertações, teses

Sumário

Referências

- Afonso das Neves Fernandes. **FORMAL VERIFICATION OF THE CEPH CONSENSUS ALGORITHM USING TLA$^+$**. Dissertação de mestrado, 2021. Universidade do Porto. URL: `https://repositorio-aberto.up.pt/bitstream/10216/139563/2/529181.pdf`
- Lee, J.-Y., Kim, M.-H., Shah, S. A. R., Ahn, S.-U., Yoon, H., and Noh, S.-Y. (2021). **Performance evaluations of distributed file systems for scientific big data in fuse environment**. In Electronics 2021, 10, 1471, pages 1–16. Eletronics.
- Dai, H; Wang, Y; Kent, K. B.; Zeng, L.; Xu, C. **The State of the Art of Metadata Managements in Large-Scale Distributed File Systems — Scalability, Performance and Availability**. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2022.

Pesquisas em MDS e CEPH

Tópicos em Redes de Computadores (INFO-7065)

Josiney de Souza (josiney.souza@ifc.edu.br)

UFPR / DInf

6 de Junho de 2023