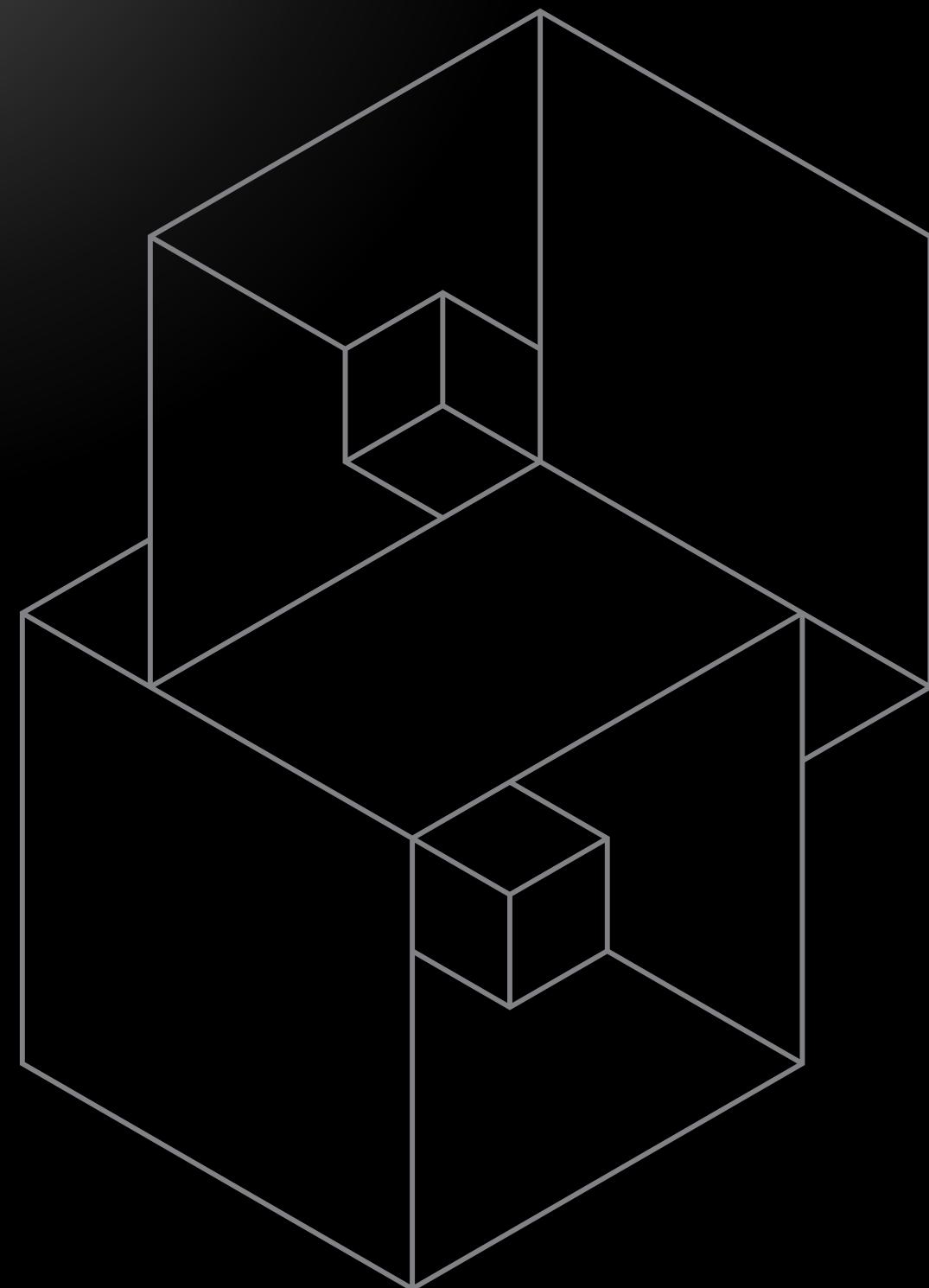


SISTEMAS DE ARCHIVOS DISTRIBUIDOS

ÍNDICE

- Introducción
- Arquitectura de un DFS
- Ejemplo de un DFS
- Aplicaciones del DFS



¿QUE ES UN DFS?

Un sistema de archivos distribuido, o por sus siglas DFS (Distributed File System), es un método para almacenar y acceder a archivos basado en una arquitectura cliente - servidor; uno o más servidores centrales almacenan archivos que pueden ser accedidos, con los derechos de autorización adecuados, por cualquier número de clientes remotos en la red [1].

CARACTERISTICAS

- Transparencia de acceso.
- Transparencia de ubicación.
- Actualizaciones concurrentes.
- Cifrado de datos.
- Heterogeneidad.
- Replicación.

¿COMO FUNCIONA UN DFS?

DFS se conecta con los terminales y servidores en un sistema de archivos paralelo con un espacio de nombres único, permitiendo el acceso rápido a datos almacenados en diversos dispositivos. Puede funcionar como un espacio de nombres autónomo o basado en dominios. Al acceder a un archivo, DFS consulta varios servidores para ofrecer la primera copia disponible, evitando la sobrecarga y asegurando la accesibilidad en caso de fallos. La replicación garantiza que las modificaciones se reflejen en todas las instancias del archivo.

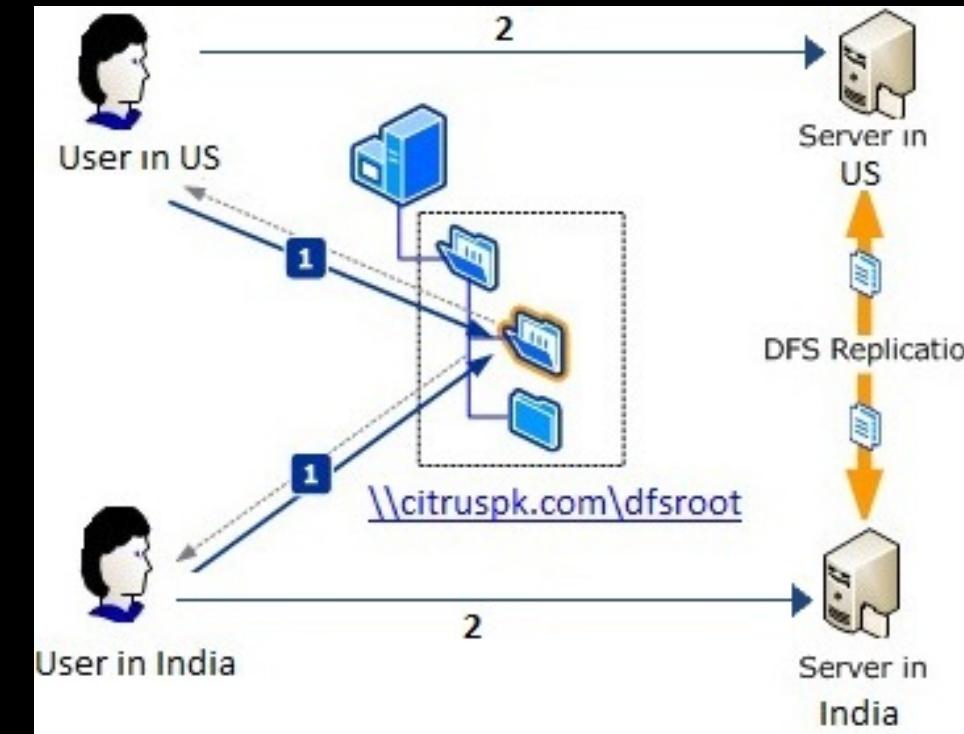


Image from (DFS. Hello World!, 2013).

VENTAJAS

- Diseñado para grandes conjuntos de datos.
- Tolerancia a fallos.
- Escalabilidad.
- Simplicidad.

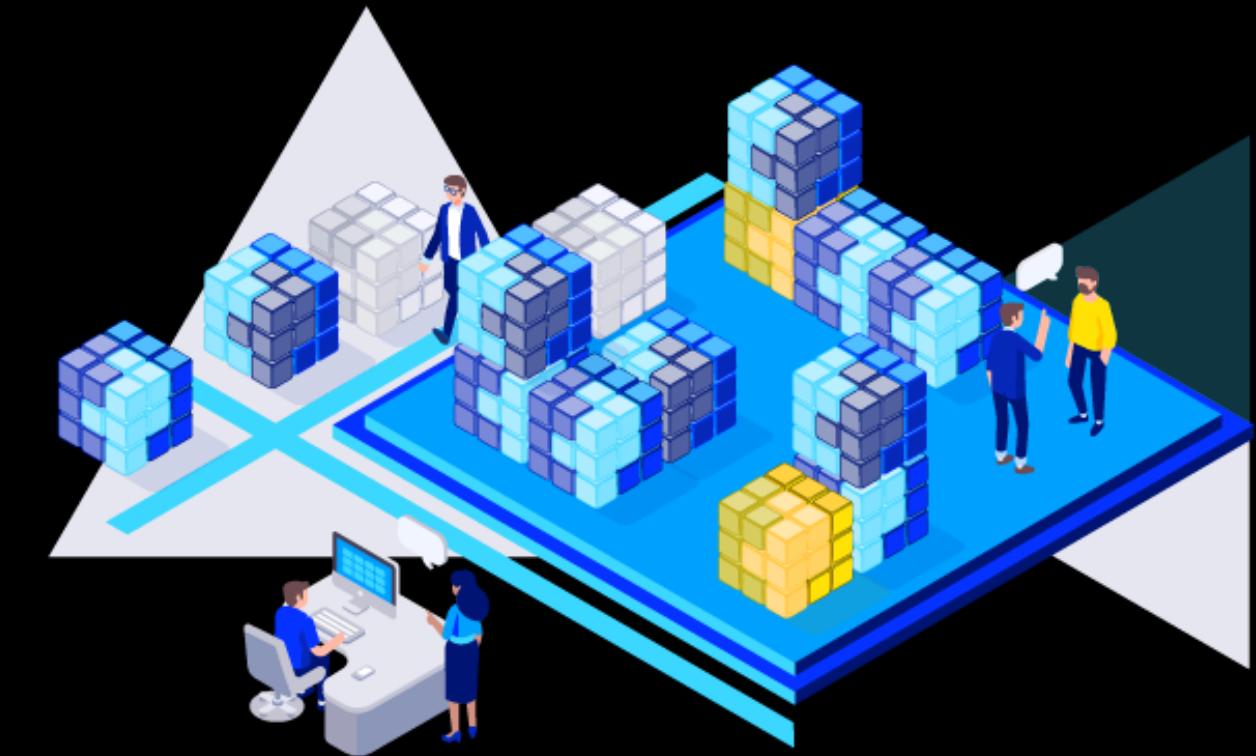


Image from (Large Data Set Storage, 2024).

DESVENTAJAS

- Más lento que los sistemas de archivos locales.
- Actualizaciones o cambios de archivos.
- Costos.



ARQUITECTURA DFS

La arquitectura DFS proporciona una separación clara y eficiente de las principales responsabilidades en la gestión del acceso a archivos y esto se puede lograr estructurando el servicio de archivos en 3 componentes clave:

- Servicio de archivos planos.
- Servicio de directorios.
- Modulo cliente.

SERVICIO DE ARCHIVOS PLANOS

El servicio de archivos planos maneja operaciones sobre archivos usando identificadores únicos (UFIDs) para referirse a ellos en todas las solicitudes. Cuando se crea un archivo, el servicio genera un UFID y lo devuelve al solicitante.

El servicio de archivos planos no requiere operaciones de apertura y cierre; los archivos se acceden directamente mediante su UFID, y esta diferencia se debe principalmente a la necesidad de tolerancia a fallos, como :

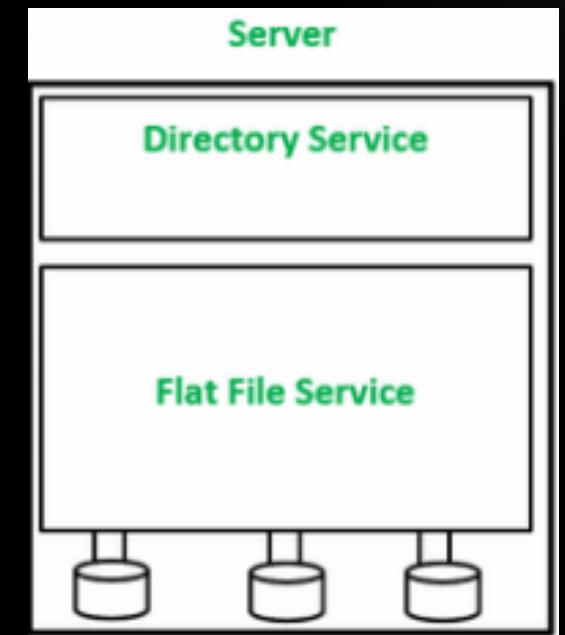


Image from (GeeksforGeeks, 2022).

Operaciones repetibles



Servidores sin estado



OPERACIONES DEL SERVICIO DE ARCHIVOS PLANO

<i>Read(FileId, i, n) → Data</i> — throws <i>BadPosition</i>	If $1 \leq i \leq Length(File)$: Reads a sequence of up to n items from a file starting at item i and returns it in $Data$.
<i>Write(FileId, i, Data)</i> — throws <i>BadPosition</i>	If $1 \leq i \leq Length(File)+1$: Writes a sequence of $Data$ to a file, starting at item i , extending the file if necessary.
<i>Create() → FileId</i>	Creates a new file of length 0 and delivers a UFID for it.
<i>Delete(FileId)</i>	Removes the file from the file store.
<i>GetAttributes(FileId) → Attr</i>	Returns the file attributes for the file.
<i>SetAttributes(FileId, Attr)</i>	Sets the file attributes (only those attributes that are not shaded in Figure 12.3).

Image from (Jisy Raju Assistant Professor, Distributed File Systems: Introduction, 2004).

SERVICIO DE DIRECTORIOS

El servicio de directorios realiza la correspondencia entre los nombres de archivo en texto y sus Identificadores Únicos de Archivos (UFIDs). Los clientes pueden obtener el UFID de un archivo al suministrar su nombre en texto al servicio de directorios [5].

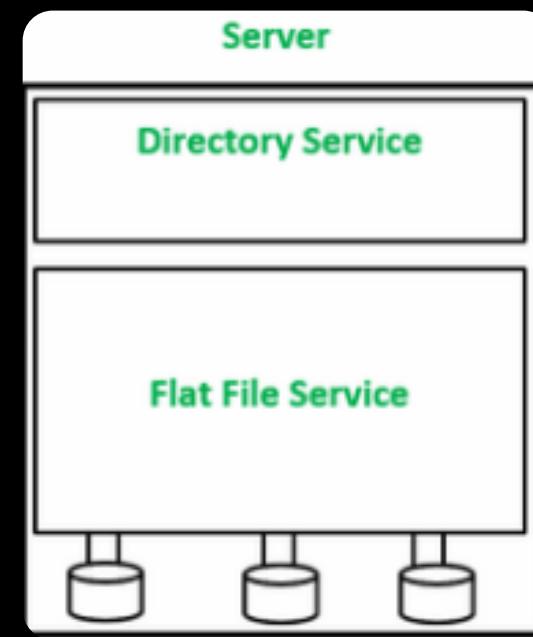


Image from (GeeksforGeeks, 2022).

OPERACIONES DEL SERVICIO DE DIRECTORIOS

Lookup(Dir, Name) → fileId

— throws *NotFound*

Locates the text name in the directory and returns the relevant UFID. If *Name* is not in the directory, throws an exception.

AddName(Dir, Name, fileId)

— throws *NameDuplicate*

If *Name* is not in the directory, adds *(Name, File)* to the directory and updates the file's attribute record.

If *Name* is already in the directory, throws an exception.

UnName(Dir, Name)

— throws *NotFound*

If *Name* is in the directory, removes the entry containing *Name* from the directory.

If *Name* is not in the directory, throws an exception.

GetNames(Dir, Pattern) → NameSeq

Returns all the text names in the directory that match the regular expression *Pattern*.

Image from (Jisy Raju Assistant Professor, Distributed File Systems: Introduction, 2004).

MODULO CLIENTE

Un módulo cliente se ejecuta en cada computadora cliente, combinando las operaciones del servicio de archivos planos y del servicio de directorios en una sola API, accesible para los programas de usuario en las computadoras cliente [5].

Para que un archivo pueda ser enviado al servidor primero debe pasar por un filtro llamado: Control de acceso.



Image from (Patricia, 2022).

DIAGRAMA DE LA ARQUITECTURA

La imagen muestra de manera completa la arquitectura del DFS, donde el módulo cliente en la computadora cliente interactúa con los servicios de directorios y archivos planos en la computadora servidora.

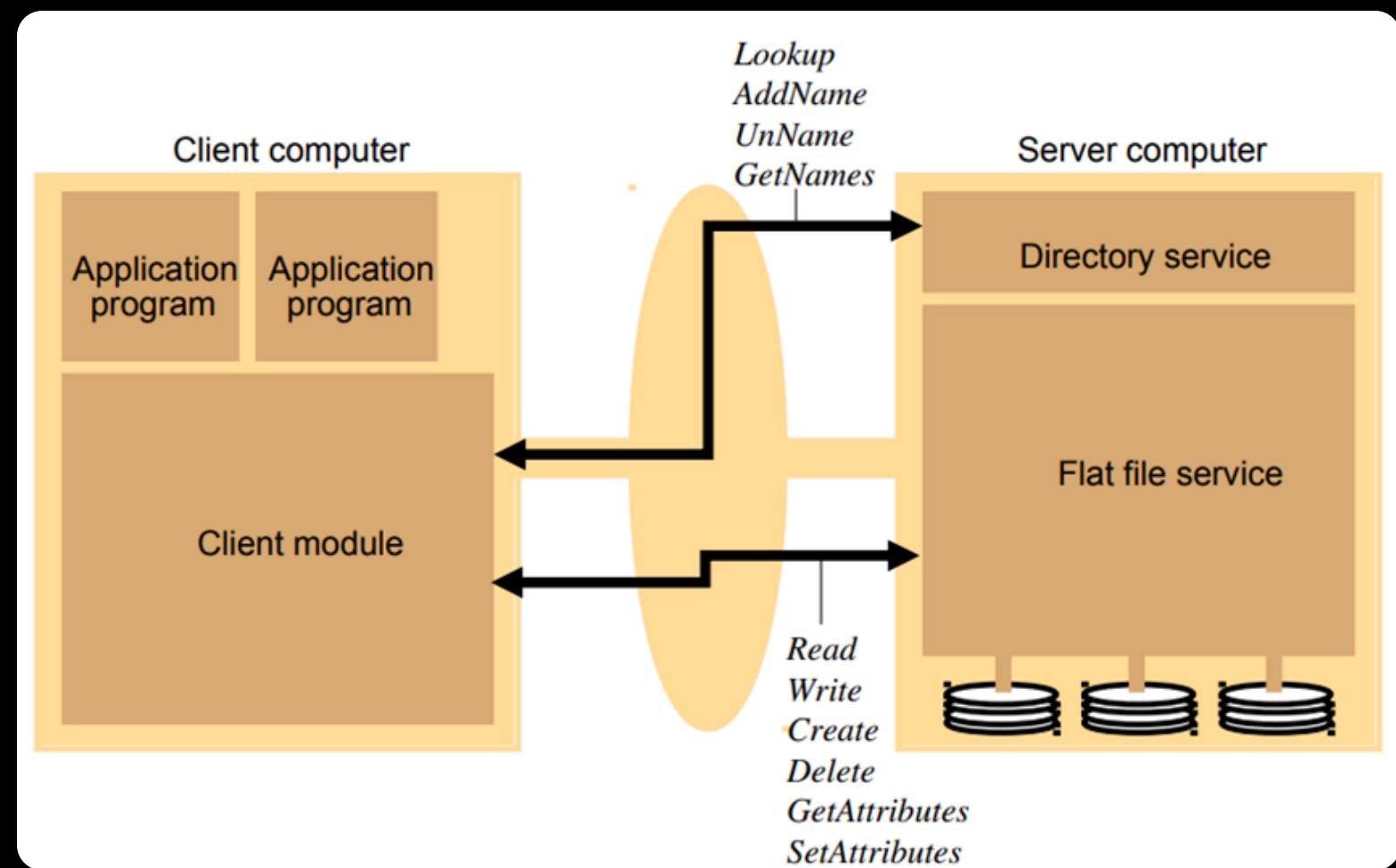


Image from (Jisy Raju Assistant Professor, Distributed File Systems: Introduction, 2004).

¿QUE ES CEPHFS?

CephFS es un sistema de archivos compatible con POSIX que se construye sobre el almacenamiento distribuido de objetos de Ceph (RADOS). Proporciona acceso a archivos en un clúster Red Hat Ceph y mantiene una fuerte coherencia de caché entre los clientes. Su objetivo es que los procesos funcionen de manera similar en diferentes hosts como lo harían en un solo host, aunque en algunos casos puede desviarse de las estrictas semánticas de POSIX [8].



Image from (Ceph File System – Ceph Documentation, 2016).

¿COMO FUNCIONA CEPHFS?

Ceph se basa en RADOS, un sistema de almacenamiento distribuido donde los archivos se dividen en objetos que se distribuyen entre múltiples OSDs mediante un algoritmo de hashing. Los objetos se replican en varios OSDs para garantizar durabilidad [8].

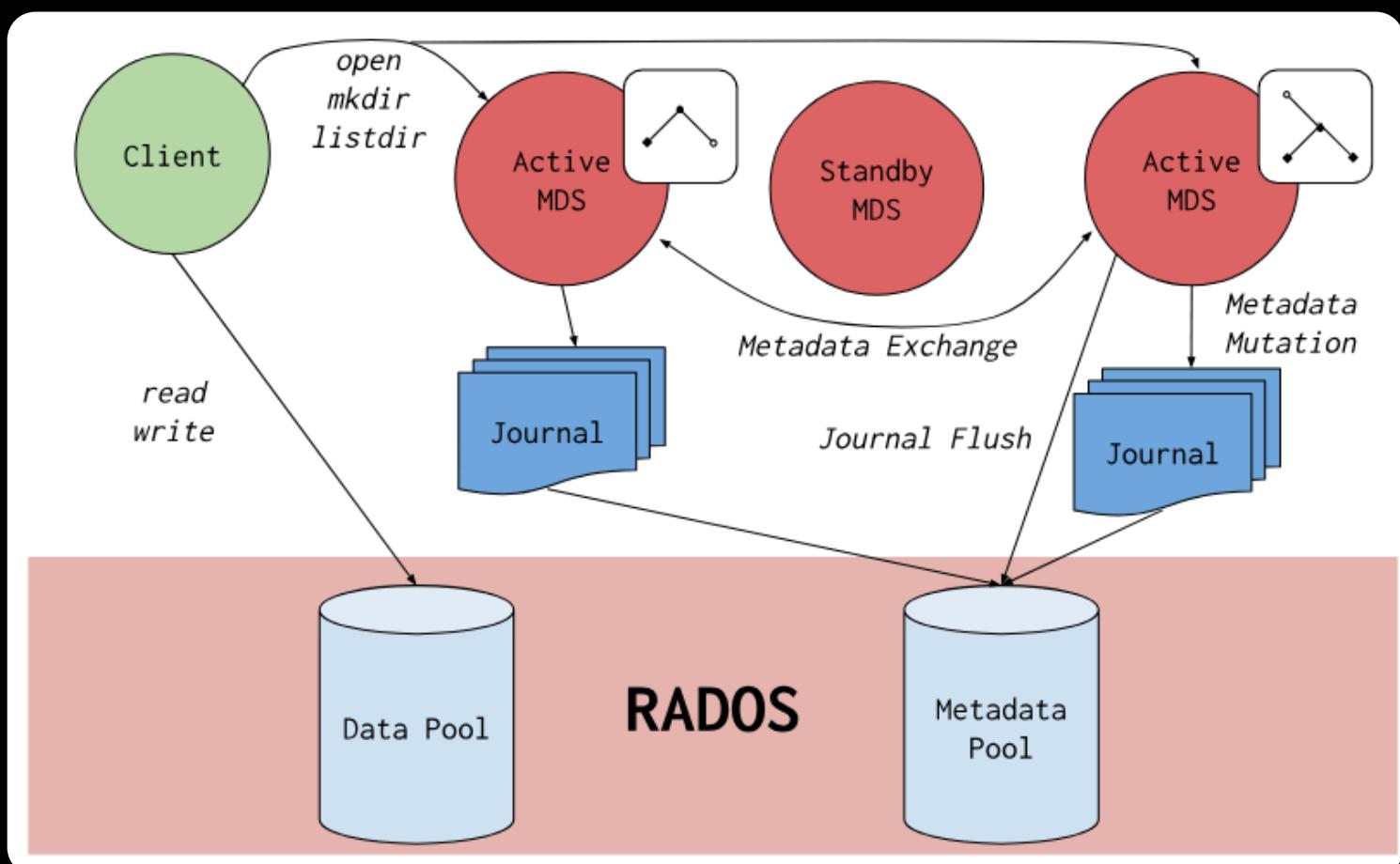


Image from (Ceph Is a Distributed Object, Block, and File Storage Platform, 2024).

ARQUITECTURA DE CEPHFS

Como ya habíamos visto con anterioridad, los componentes de los sistemas de archivos distribuido tienen como base un servicio de archivos planos, un servicio de directorios y un módulo cliente, pero en CephFS encontramos nuevos componentes que son:

- RADOS.
- CEPH OSD DAEMON (OSD).
- CEPH MONITORS (MON).
- CEPH MANAGER (MGR).
- CEPH META DATA SERVERS (MDS)

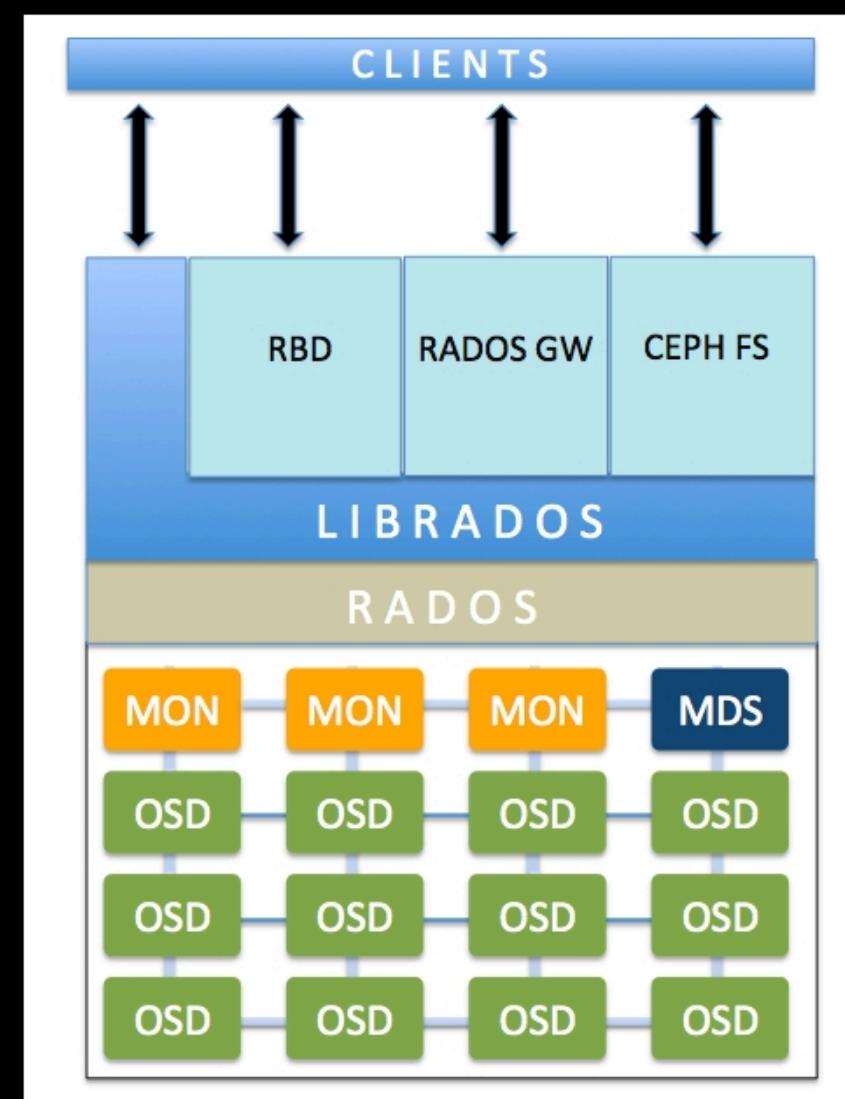


Image from (Red Hat Ceph Storage 5 File System Guide Configuring and Mounting Ceph File Systems).

CARACTERISTICAS ADICIONALES

Debido a estas características, Ceph ha escalado a un punto donde es ampliamente valorado como una de las soluciones de almacenamiento más robustas y flexibles del mercado:



Image from (Events - Ceph, 2024).

- Escalabilidad Horizontal.
- Alta Disponibilidad.
- Autorreparación.
- Multitenencia.

APLICACIONES

Los DFS tienen una amplia gama de aplicaciones en diferentes áreas tecnológicas, ya que permiten acceder a archivos y datos de forma remota, facilitando la colaboración, el almacenamiento masivo y el procesamiento distribuido. Algunas de las aplicaciones clave de los sistemas de archivos distribuidos incluyen:

- Entornos de virtualización y contenedores.
- Almacenamiento en la nube.
- Big data.



TABLA DE COMPARACIONES

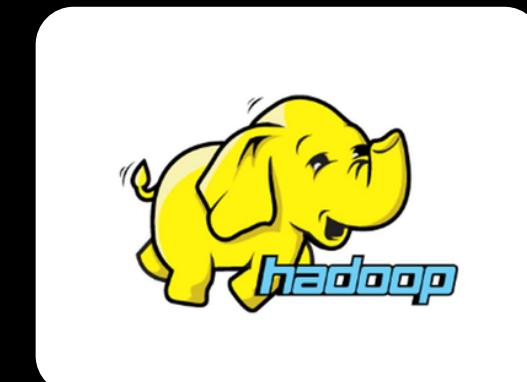
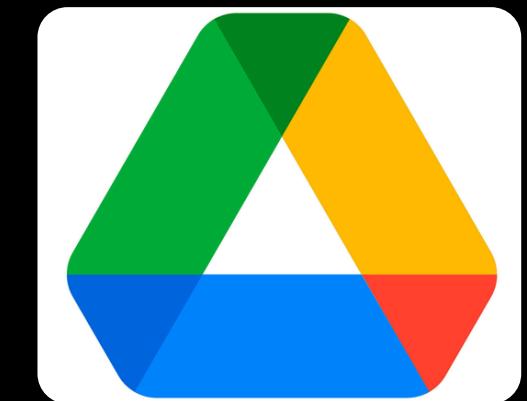


Amazon S3



Dropbox

DFS	Fortalezas	Debilidades
S3 (Amazon)	Su mayor fortaleza está en su escalabilidad y alta disponibilidad, con un enfoque claro en grandes volúmenes de datos para aplicaciones empresariales.	Puede ser complejo de usar y más costoso a largo plazo.
Google Drive	Muy fácil de usar y se integra bien con otros servicios de Google.	Limitaciones cuando se trata de manejar grandes volúmenes de datos y su escalabilidad está más enfocada en uso personal o empresarial pequeño.
Dropbox	Facilidad de uso y aplicabilidad.	Limitado para grandes necesidades empresariales o big data.
Hadoop (HDFS)	Excelencia en procesamiento y almacenamiento distribuido de big data e ideal para manejar enormes volúmenes de datos.	Requiere experiencia técnica para implementar y gestionar.
CephFS	Ofrece flexibilidad y rendimiento ajustable para infraestructuras empresariales complejas, con soporte tanto para almacenamiento de bloques como objetos.	Su configuración puede ser técnica y demandante.



CONCLUSIONES

REFERENCIAS BIBLIOGRÁFICAS

- 1.Jisy Raju Assistant Professor, CE Cherthala Module 4 Distributed file system: File service architecture -Network file system-Andrew file system- Name Service 4.1
Distributed file Systems: Introduction. (n.d.). http://www.ectl.ac.in/images/pdf_docs/studymaterial/cse/s7/DC4.pdf
- 2. ¿Qué es DFS? – Sistemas de archivos distribuidos. (2023). <https://www.nutanix.com/es/info/distributed-file-systems>
- 3.micronimics. (2022, December 24). Advantages and Disadvantages of HDFS And Traditional File Systems – Data Recovery in Ahmedabad, Best Data Recovery in India. Data Recovery in Ahmedabad, Best Data Recovery in India. <https://www.micronicsindia.com/advantages-and-disadvantages-of-hdfs-and-traditional-file-systems/>
- 4.Jisy Raju Assistant Professor, CE Cherthala Module 4 Distributed file system: File service architecture -Network file system-Andrew file system- Name Service 4.1
Distributed file Systems: Introduction. (n.d.). http://www.ectl.ac.in/images/pdf_docs/studymaterial/cse/s7/DC4.pdf
- 5.Kozlowski, S. W., & Bell, B. S. (2007). A theory-based approach for designing distributed learning systems.
- 6. ~~Distributed Systems~~ Distributed File System [2] Distributed File System. (n.d.). https://www.ia.pw.edu.pl/~tkruk/edu/rso.b/lecture/pre/rso08_pre.pdf
- 7. Murugesan, P. (n.d.). ~~Distributed File Systems~~. Retrieved October 12, 2024, from
<https://www.engr.colostate.edu/ECE658/2013/onlinepresentation/Prabhakaran/Prabhakaran.pdf>
- 8.Red Hat Ceph Storage 5 File System Guide Configuring and Mounting Ceph File Systems. (n.d.). Retrieved October 12, 2024, from
https://docs.redhat.com/ja/documentation/red_hat_ceph_storage/5/pdf/file_system_guide/red_hat_ceph_storage-5-file_system_guide-ja-jp.pdf

MUCHAS GRACIAS

POR SU ATENCIÓN