Universidad EAFIT

Escuela de Ingeniería

Departamento de Informática y Sistemas

Proyecto 1

Presentado a

Edwin Nelson Montoya Munera

Elaborado por

Jorge Alfredo Villarreal Márquez

Daniel Gonzalez Bernal

Martin Villegas Aristizábal

Fecha de Entrada

06/10/2023

https://github.com/jovillarrealm/jovillarrealm-st0263/tree/main/p1

Marco Teórico

Sistema de archivos

Un sistema de archivos FS (en inglés File System) es un componente esencial de cualquier sistema informático, pues es el encargado de gestionar cómo se almacenan, organizan y acceden los datos(*File System | What is a File System - javatpoint*, s. f.). Este concepto fundamental es crucial para comprender la informática moderna y su funcionamiento. En este marco teórico, explicaremos los diferentes tipos de sistemas de archivos.

Tipos de Sistemas de Archivos(«Sistema de archivos», 2023) (*File System | What is a File System - javatpoint*, s. f.)

Sistemas de Archivos Locales: Estos sistemas almacenan datos en un solo computador y son los más comunes en computadores personales y portátiles. Utilizan estructuras de datos y algoritmos específicos para organizar y recuperar información de manera eficiente en el disco duro local.

Sistemas de Archivos de Red: Los sistemas de archivos de red almacenan datos en una red de computadores. Esto permite a los usuarios acceder a archivos desde cualquier dispositivo conectado a la red. El acceso remoto y la gestión centralizada de datos son características esenciales de estos sistemas.

Sistemas de Archivos Distribuidos: Los sistemas de archivos distribuidos almacenan datos en múltiples computadores interconectados. Su diseño se centra en proporcionar alta disponibilidad y escalabilidad. Estos sistemas suelen utilizarse en entornos empresariales y en la nube, donde la redundancia y la tolerancia a fallos son críticas.

Para comprender en profundidad los sistemas de archivos, es esencial recurrir a las siguientes teorías fundamentales:

Teoría del Almacenamiento y Recuperación de Información: Esta teoría proporciona el fundamento para entender cómo se almacenan y recuperan datos en sistemas informáticos. Aborda cuestiones como la gestión de espacio en disco, la indexación eficiente y la recuperación de datos.

Teoría de las Estructuras de Datos: La organización eficiente de datos es esencial en cualquier sistema de archivos. La teoría de las estructuras de datos ofrece un conjunto de principios y técnicas para organizar información de manera que sea fácil de acceder y gestionar. Esto incluye árboles de directorios, tablas de asignación y métodos de indexación.

Teoría de los Sistemas Distribuidos: Cuando se trata de sistemas de archivos distribuidos, la teoría de los sistemas distribuidos es crucial. Ofrece una comprensión de cómo múltiples

computadoras pueden colaborar en la gestión de datos de manera coherente y efectiva, abordando desafíos como la sincronización y la consistencia.

Sistema de archivos distribuido

> Un sistema de archivos distribuido (DFS) es un sistema de archivos que abarca varios servidores de archivos o múltiples ubicaciones, como servidores de archivos que están situados en diferentes lugares físicos. Los archivos son accesibles como si se almacenarán localmente, desde cualquier dispositivo y desde cualquier lugar de la red. Un DFS hace conveniente compartir información y archivos entre los usuarios en una red de una manera controlada y autorizada
>
> (*Distributed File System*, s. f.).

Para satisfacer las necesidad a escala de sistemas que requieran manejar grandes volúmenes de datos o analítica sobre grandes cantidades de datos, se requiere una escalabilidad horizontal que amplía los requisitos del sistema. Estos requisitos incluyen: persistir y acceder grandes volúmenes de datos, que estos datos estén disponibles incluso en vista de fallos de nodos, que esto ocurra de manera eficiente, concurrente, y transparente a un cliente.

Esto, a su vez, introduce nuevos problemas. Por ejemplo, se introduce la necesidad de replicar datos para tolerancia a fallos en nodos y evitar cuellos de botella en un nodo que contenga datos en alta demanda; y particionamiento, para poder dividir los datos según las necesidades del software que vaya a interactuar con ellos. En un software como un sistema operativo, tenemos datos que usualmente son un archivo completo, por tanto muchos de los primeros DFSs están orientados a entregar datos que estén estructurados como un sistema de ficheros («What Is DFS (Distributed File System)?», 2020).

Sin embargo, con el auge de sistemas de Big Data, e incremento en la cantidad de datos que las organizaciones manejan en general, nacen necesidades de manejar los datos que sean más escalables y simplificadas que manejar directorios y archivos para acceder a los datos. Ahora, existen dos formas diferentes de manejo de datos que se relacionan con los populares sistemas de almacenamiento basados en particionamiento a nivel de bloque y objeto.

Bloques

Por un lado, el particionamiento por bloques en sistemas distribuidos, al no optimizar el uso para una sola máquina (como lo haría un FS como ext4), un bloque en un DFS es más grande (4KB (ext4) vs 128MB (HDFS) (*Configuración HDFS - Amazon EMR*, s. f.)) más apto para lecturas grandes y pensado para manejado a través de una red, lo que permite facilidad y eficiencia a la hora de replicar estos bloques. Por ejemplo, si un cliente quiere pedir un *.parquet* de varios Gigabytes a un HDFS, este sistemáticamente debe ir recopilando los bloques necesarios para reconstruir ese archivo para el usuario, lo que usualmente requiere un identificador para cada bloque, y muy pocos metadatos que

relacionen varios bloques como parte de un mismo archivo. Esto permite baja latencia y fácil mutación sobre los datos, por lo que se usan bastante para Storage Area Networks y sistemas transaccionales con datos estructurados (*Almacenamiento de objetos, bloques y archivos*, s. f.).

Objetos

Por otro lado, el almacenamiento basado en objetos, adiciona a sus particiones más metadatos para permitir diferentes formas de acceso a una misma partición en comparación que un bloque que solo tendría un simple identificador, estas particiones tienen tamaños dinámicos (*¿Qué son el almacenamiento de objetos y los almacenes de objetos?*, s. f.), más acoplados con los datos a los que apuntan. Este diseño está pensado para datos cuyo uso es ser ser creados y accedidos, pero no mutados (WORM), por lo que tienen datos inmutables, con formas flexibles de acceso. Esto permite arquitecturas de nada compartido (en inglés NSA)(*Almacenamiento de objetos, bloques y archivos*, s. f.), lo que permite la escalabilidad con grandes cantidades de datos no estructurados como los videos de Facebook, Netflix, audios de Spotify, o archivos en Dropbox. Estas ventajas para datos inmutables también funcionan como limitaciones para sistemas transaccionales (*Almacenamiento de objetos, bloques y archivos*, s. f.).

Sistema de archivos de red

El sistema de archivos de red (en inglés NFS) es un protocolo que permite compartir archivos entre sistemas de una red. Este protocolo de transferencia de archivos es de alto rendimiento y proporciona acceso a los archivos como si estuvieran almacenados localmente («Network File System», 2023). NFS fue desarrollado por Sun Microsystems en 1984. La arquitectura de NFS es relativamente simple. El sistema cliente NFS se conecta a un sistema servidor NFS a través de una red. El sistema servidor NFS proporciona acceso a los archivos almacenados en su sistema de archivos (JasonGerend, 2023).

Este protocolo se divide en dos componentes principales:

- El protocolo de servidor NFS: Este protocolo es responsable de proporcionar acceso a los archivos en el sistema servidor NFS.
- El protocolo de cliente NFS: Este protocolo es responsable de solicitar acceso a los archivos en el sistema servidor NFS.

Cuando un sistema cliente NFS se conecta a un sistema servidor NFS, el sistema cliente envía una solicitud al sistema servidor. La solicitud indica el archivo o directorio que el sistema cliente desea acceder, el sistema servidor responde a la solicitud del sistema cliente, esta respuesta incluye información sobre el archivo o directorio, como el tamaño del archivo, la fecha de modificación y los permisos de acceso, más adelante, el sistema cliente utiliza la información proporcionada por el sistema servidor NFS para acceder al archivo o directorio. El sistema cliente puede leer, escribir, crear y eliminar archivos y directorios(«Network File System», 2023).

Entre los beneficios de usar NFS tenemos (Lemus, 2022):

- Acceso a archivos compartidos: NFS permite a los usuarios acceder a archivos compartidos en otros sistemas informáticos. Esto facilita el trabajo en equipo y la colaboración.

- Escalabilidad: Es un protocolo escalable que puede soportar una gran cantidad de usuarios y tráfico.

- Portabilidad: Puede ser implementado en una amplia gama de sistemas operativos.

NFS también tiene algunas desventajas, como por ejemplo (*Protocolo NFS*, s. f.):

- Seguridad: NFS no es un protocolo de seguridad por sí mismo. Es necesario implementar medidas de seguridad adicionales para proteger los archivos compartidos.

- Rendimiento: NFS puede tener un impacto negativo en el rendimiento del sistema servidor NFS, especialmente si hay una gran cantidad de usuarios accediendo a archivos compartidos.

NFS es un protocolo maduro que ha sido ampliamente adoptado por la industria. Sin embargo, también está evolucionando para satisfacer las necesidades de las nuevas tecnologías, como la computación en la nube y el big data, una de las tendencias más importantes en el desarrollo de NFS es la virtualización. La virtualización permite a los usuarios crear sistemas de archivos virtuales transparentes, lo que los hace más flexibles y escalables.

Especificaciones

1. Arquitectura:

El sistema tiene una arquitectura distribuida y se basa en el concepto de WORM (Write Once - Read Many), lo que significa que los datos son escritos una vez y leídos muchas veces. La arquitectura consta de los siguientes elementos:

Dos NameNodes: Uno principal y otro secundario (backup).
Tres DataNodes: Estos almacenan archivos y se comunican con los NameNodes.
Cliente: Interactúa con el sistema a través de HTTP.
Protocolo de Comunicación: Utiliza gRPC para la replicación de registros y archivos entre NameNodes y DataNodes, y HTTP para las solicitudes del cliente.

2. Componentes:

NameNode Primario y Secundario: Estos nodos son responsables de la gestión de los datos y la asignación de DataNodes para almacenar archivos. Hay un NameNode que actúa como el principal encargado y el otro está ahí en caso de que se encuentre algún tipo de falla para ser el respaldo.

DataNodes: Almacenan los archivos y se encargan de replicar archivos y registros, así como de registrar su disponibilidad y contenido con los NameNodes.

Cliente: Interactúa con el sistema para leer o escribir archivos utilizando HTTP.

3. Relaciones:

RPC: Se utiliza gRPC para la replicación de registros entre NameNodes y la replicación de archivos entre DataNodes. También se utiliza para las solicitudes de lectura y escritura de archivos desde el NameNode al DataNode.

HTTP <-> Archivos: El cliente se comunica con los NameNodes a través de HTTP para las solicitudes de lectura y escritura de archivos. Los archivos se almacenan y gestionan internamente a través de gRPC entre DataNodes y NameNodes.

Para el tema de arquitectura se dispondrá de dos “NameNode”, uno que actuará como Nodo líder primario y otro nodo que actuará como backup o nodo secundario, estos NameNode a su vez estarán conectados con 3 “DataNodes” los cuales contendrán los archivos del servidor utilizando como unidad mínima de almacenamiento: “archivos”, cada vez que un archivo nuevo se quiera ingresar al sistema, el “NameNode” primario deberá

designar a un DataNode para recibir el archivo y el “DataNode” al recibir el archivo, designará a uno de sus pares para enviar una copia o “BackUp”. Del mismo modo, al eliminar un archivo, el “NameNode” enviará la solicitud a los “DataNodes” que contengan una copia de archivos y los deberán eliminar. Desde el punto de vista del cliente, todo el sistema se verá como una sola unidad de almacenamiento.

Se utilizará el protocolo gRPC para todas las conexiones, salvo aquellas iniciadas por el cliente, que serán con HTTP.

- Cliente <-> NameNode: El cliente va a hacer una petición con API REST para hacer una escritura o lectura de datos.

- NameNode <-> NameNode: gRPC para replicación de registros de DataNodes y archivos.

- DataNode <-> DataNode: gRPC para la replicación de archivos (escritura).

- NameNode -> DataNode: gRPC para las solicitudes de lectura y escritura de archivos.

- DataNode -> NameNode: Los “DataNodes” podrían registrarse con los “NameNodes”, para hablarle al “Leader” y notifiquen que están disponibles y si tienen datos, y luego este haga replicación pero de los registros de “DataNodes”. También tienen que decirle al “NameNode” si el cliente les escribió cosas para que los “NameNodes” sepan donde están los archivos cuando el cliente se los pida la próxima vez.

Bibliografía

Almacenamiento de objetos, bloques y archivos: ¿cuál es mejor para las aplicaciones en la nube? | *Computer Weekly*. (s. f.). ComputerWeekly.es. Recuperado 7 de septiembre de 2023, de https://www.computerweekly.com/es/consejo/Almacenamiento-de-objetos-bloques-y-archivos-cual-es-mejor-para-las-aplicaciones-en-la-nube

Configuración HDFS - Amazon EMR. (s. f.). Recuperado 4 de octubre de 2023, de https://docs.aws.amazon.com/es_es/emr/latest/ReleaseGuide/emr-hdfs-config.html

Distributed file system. (s. f.). Cohesity. Recuperado 4 de octubre de 2023, de

https://www.cohesity.com/glossary/distributed-file-system/

File System | What is a File System—Javatpoint. (s. f.). Recuperado 4 de octubre de 2023, de https://www.javatpoint.com/file-system

JasonGerend. (2023, abril 12). *Introducción a Network File System*. https://learn.microsoft.com/es-es/windows-server/storage/nfs/nfs-overview

Lemus, I. (2022, julio 23). Sistemas de archivos de red (NFS). *Conocimiento Libre*. https://conocimientolibre.mx/protocolonfs/

Network File System. (2023). En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Network_File_System&oldid=151904389

Protocolo NFS: Cómo funciona y cómo configurarlo en Windows. (s. f.). RedesZone. Recuperado 4 de octubre de 2023, de https://www.redeszone.net/tutoriales/internet/protocolo-archivos-nfs/

¿Qué son el almacenamiento de objetos y los almacenes de objetos? | Glosario. (s. f.). Recuperado 7 de septiembre de 2023, de https://www.hpe.com/lamerica/es/what-is/object-storage.html

Sistema de archivos. (2023). En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Sistema_de_archivos&oldid=151667854

What Is a File System? Types of Computer File Systems and How they Work – Explained with Examples. (2022, enero 11). freeCodeCamp.Org. https://www.freecodecamp.org/news/file-systems-architecture-explained/

What is DFS (Distributed File System)? (2020, julio 5). *GeeksforGeeks*. https://www.geeksforgeeks.org/what-is-dfsdistributed-file-system/