Case Studies

Google File System

Google File System (GFS) is a special kind of file storage system that Google created to handle massive amounts of data efficiently. Here's a simple breakdown:

1. **Distributed System**: Instead of storing data on a single computer, GFS spreads data across thousands of regular computers. This way, even if some computers fail, the data is still safe and accessible.
2. **Large Files**: GFS is designed to handle really big files (like videos or search indexes). It splits these files into smaller parts called *chunks*, usually 64 megabytes each.
3. **Chunk Servers**: These smaller parts of files (chunks) are stored on different *chunk servers*. Each chunk is copied multiple times (usually three copies) on different servers to ensure data safety in case of failures.
4. **Master Server**: There’s a *master server* that keeps track of where all the chunks are located and manages important information about the system. When a user wants to access a file, the master server tells them which chunk servers to contact.
5. **Fault Tolerance**: Since hardware can fail, GFS is built to automatically handle these failures without losing data or interrupting service. If a chunk server fails, the system will create new copies of the lost chunks on other servers.
6. **Optimized for Performance**: GFS is designed to work efficiently with the kinds of operations Google does a lot, like reading data sequentially and appending new data to existing files, rather than constantly updating or deleting data.

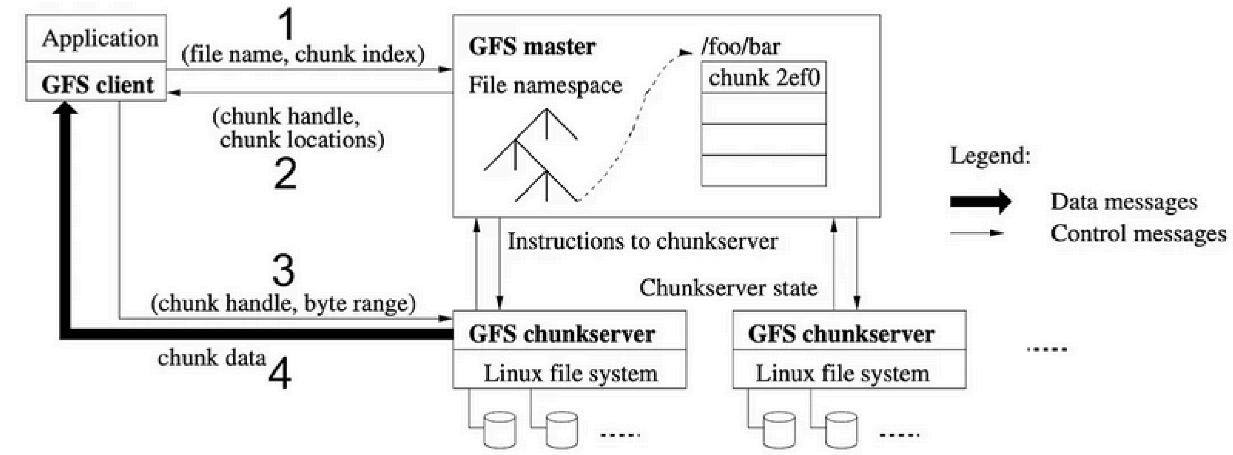


Figure 1: GFS Architecture

Google Spanner

Google Spanner is a globally distributed database service that provides strong consistency and horizontal scalability, bridging the gap between traditional relational databases and NoSQL systems. It achieves this by leveraging a combination of **TrueTime**, a synchronizing timekeeping service, and a **Paxos**-based globally distributed transaction infrastructure.

By using **TrueTime**, Google Spanner ensures external consistency across all its nodes. Despite its global distribution, it provides a synchronized and accurate timestamp for all operations.

- **Scalability**: Spanner is built to handle huge amounts of data and can grow easily as more data is added. It's used by companies that need their databases to handle millions of requests and lots of transactions without slowing down.
- **SQL Support**: Unlike some newer databases that only use custom query languages, Spanner supports familiar SQL queries. This makes it easier for developers who are used to working with traditional relational databases (like MySQL or PostgreSQL) to adopt.

In short, Google Spanner is like having a single, reliable database that can work efficiently across multiple continents while ensuring that all data is always accurate and up-to-date. It’s powerful for businesses that need to operate globally and can’t afford to have data inconsistencies or downtime.

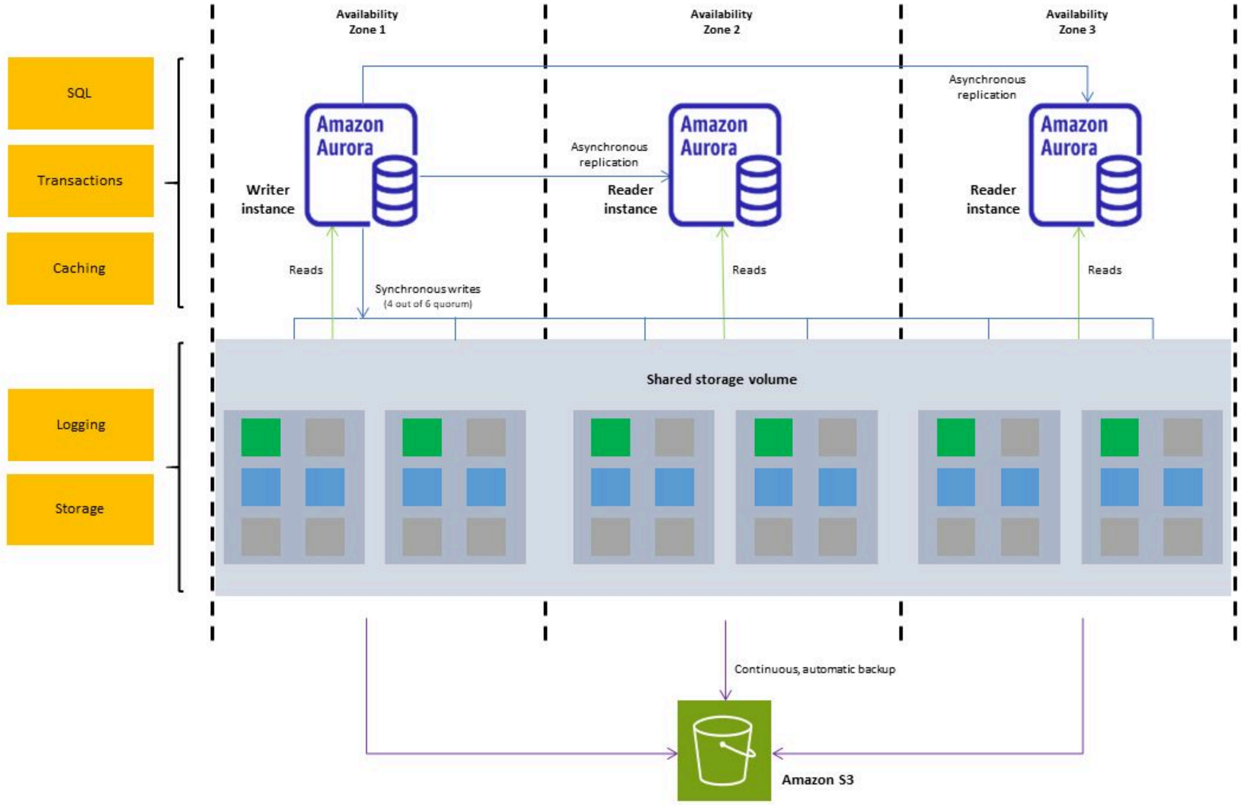
https://lh3.googleusercontent.com/I4aWsrbOMaPV_Q-vj02JkyWeY9x_4NBr01XeVe5s7Jk6F1tTFPRqfCRO9-7-1brkLJ6y1Yo1UtA=s472-w472

Amazon Aurora

Amazon Aurora is a cloud based relational database engine developed by AWS.
❖ It supports MySQL and PostgreSQL compatibility.
❖ Designed for high availability, scalability, and performance.
❖ Offers serverless deployment and automatic scaling.

❖ Aurora provides five times better performance than MySQL.
❖ It uses a distributed, fault-tolerant storage system.
❖ Ensures durability with data replication across multiple availability zones.
❖ Integrates with AWS services like Lambda, S3, and CloudWatch.

❖ Data is split and distributed across multiple nodes.
❖ Automated replication ensures high availability and fault tolerance.
❖ Uses quorum-based writes to ensure consistency.
❖ Decouples compute from storage for better scaling.



Apache Spark

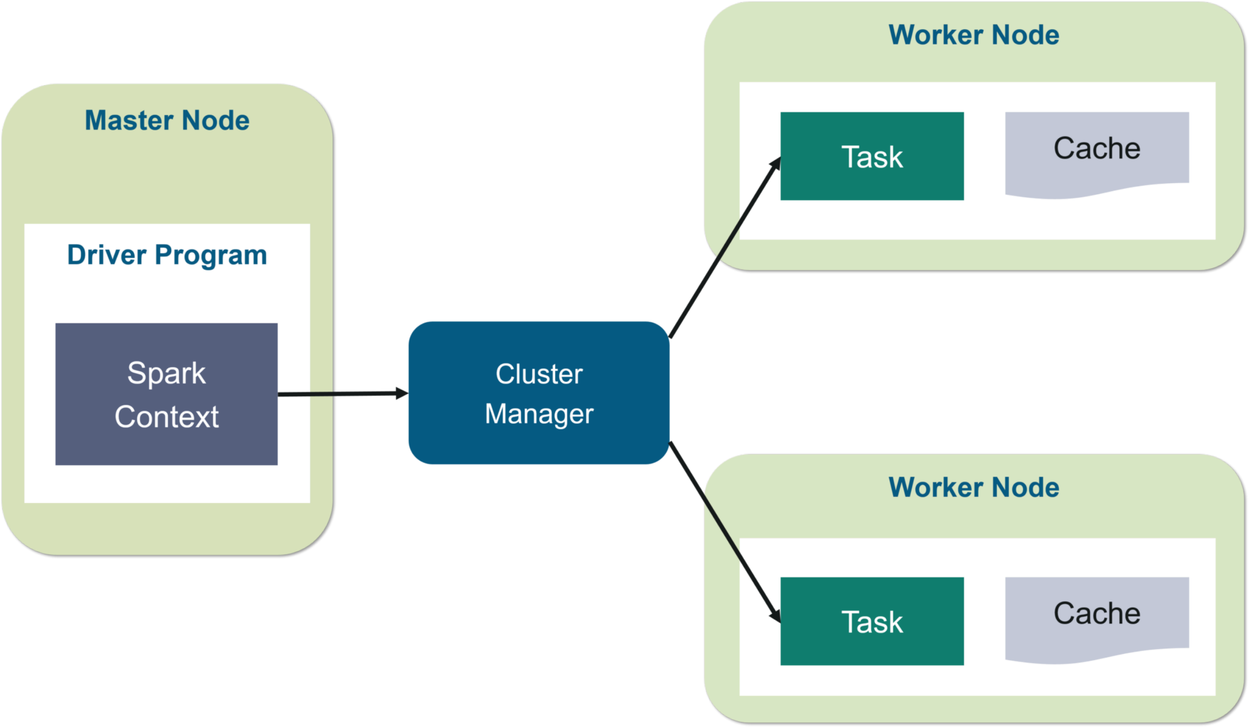
Apache Spark is a powerful tool used for **processing large amounts of data quickly**. Here’s a simple explanation:

1. **Big Data Processing**: Spark is designed to handle and analyze big data (large datasets) efficiently. It's commonly used when companies need to process data from millions or even billions of events, like social media interactions, web traffic logs, or sensor data from machines.

2. **Super Fast**: Spark is much faster than traditional data processing systems because it uses a clever technique called *in-memory computing*. This means it stores data in the computer's memory (RAM) instead of reading from or writing to slower disk storage repeatedly, which speeds up processing significantly.

3. **Scalable**: Spark can run on a single computer or be spread across many computers, working together to process data even faster. It's built to handle small or extremely large data sets, so you can scale up as your needs grow.
4. **Supports Multiple Languages**: Developers can write Spark programs using familiar programming languages like Python, Java, Scala, or R, making it easier for teams to work with it.
5. **Versatile**: Spark isn't just for one type of data processing. It can handle different types of tasks:
    - **Batch processing**: Processing large amounts of data at once.
    - **Stream processing**: Handling data in real-time, like monitoring live events.
    - **Machine learning**: Running algorithms to learn from data and make predictions.
    - **Graph processing**: Analyzing relationships in data, like social networks.

In simple terms, Apache Spark is like a super-fast engine for big data analysis that can run on a single computer or many computers working together. It's used by companies to process and analyze data quickly, whether for real-time monitoring, running machine learning models, or understanding trends in large datasets.



Dropbox

Dropbox is a **cloud storage service** that makes it easy to **store, share, and access files online**. Here's a simple breakdown of how it works:

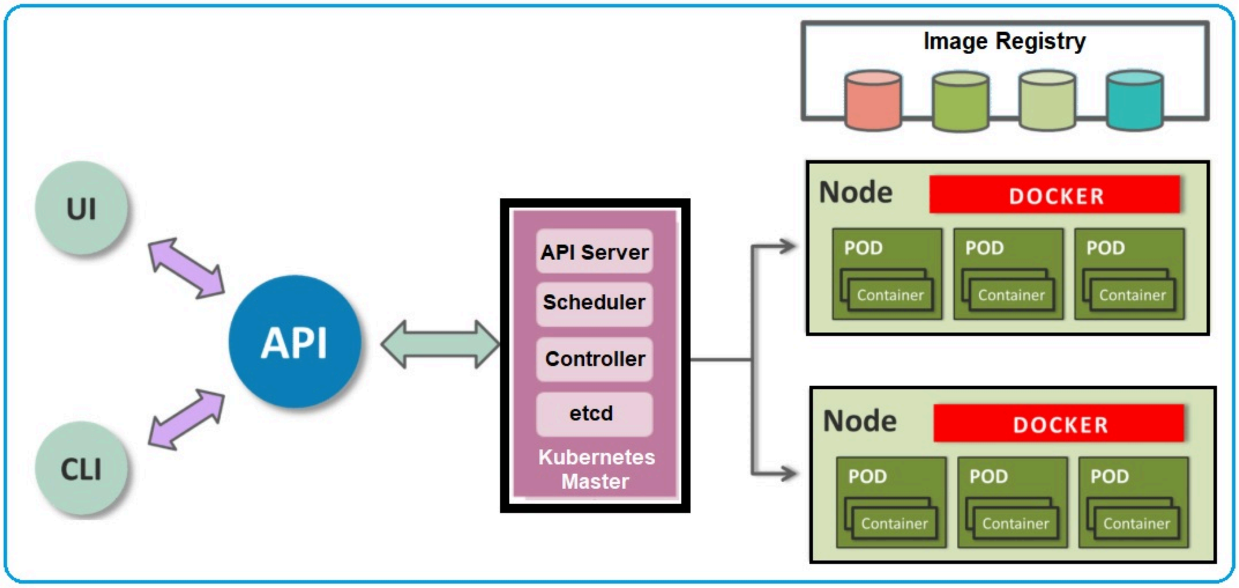
1. **Store Your Files**: Instead of saving everything on your computer's hard drive, you can upload your files (like documents, photos, or videos) to Dropbox. This frees up space on your device and keeps your data safe in the cloud.
2. **Access From Anywhere**: Once your files are in Dropbox, you can access them from any device, like your phone, tablet, or another computer, as long as you're connected to the internet. It's great for people who need their files on the go.
3. **Sync Across Devices**: Dropbox automatically synchronizes (or syncs) your files across all your devices. If you update a file on your computer, the latest version will be available on your phone or any other device with Dropbox.
4. **Share Files Easily**: You can easily share files or folders with other people. For example, if you have photos or a project document to share, you can send a

link to someone, and they can view or download the files even if they don't have a Dropbox account.

5. **Backup and Safety**: Dropbox keeps your files safe and secure. It also creates backups of your files, so if something happens to your device, your data is still safe and accessible.

In simple terms, Dropbox is like a virtual locker for your files that you can access and share from anywhere. It keeps everything organized, safe, and synced across your devices.

Docker and Kubernetes



HBase

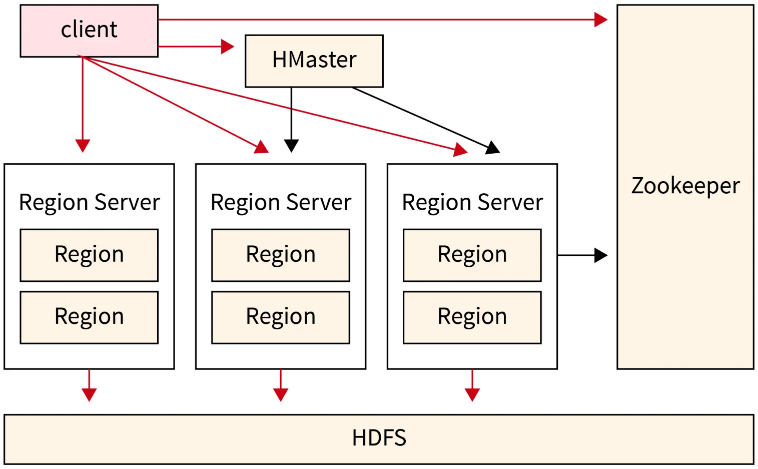
HBase is a **big data database** that's built to handle huge amounts of data across many servers. It's often used when you need to store and access very large datasets quickly. Here's a simple explanation:

1. **Built on Hadoop**: HBase is part of the Hadoop ecosystem. While Hadoop is great at storing large amounts of data, it’s slow at retrieving specific pieces. HBase fixes this by allowing for faster, random access to data in large datasets.
2. **NoSQL Database**: Unlike traditional SQL databases, HBase is a NoSQL database, meaning it doesn’t use tables with rows and columns in the same way. Instead, it stores data in a flexible, distributed way, making it ideal for handling unstructured or semi-structured data.
3. **Column-Based Storage**: HBase stores data in columns instead of rows, which makes it very fast for certain kinds of queries, especially when you’re only interested in specific pieces of data. This structure is useful for analytical applications and big data scenarios.
4. **Handles Big Data Effortlessly**: HBase can store billions of rows and millions of columns. It’s designed to scale across many machines, which means it can handle much more data than traditional databases and process it quickly.
5. **Real-Time Data Access**: HBase allows for real-time read and write access to large amounts of data. This is useful for applications like social networks, where data needs to be updated and accessed immediately.

When to Use HBase

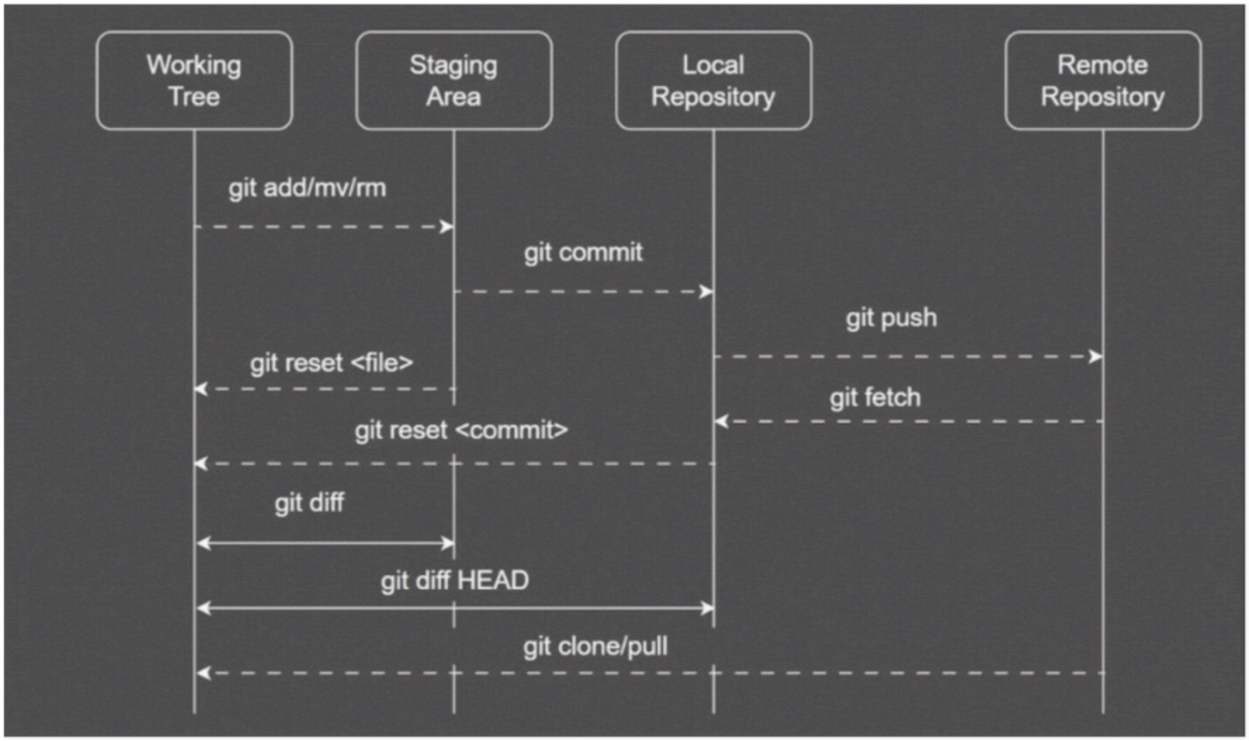
- HBase is a good choice if you have **very large datasets** that you need to access quickly and frequently.
- It’s commonly used in industries that rely on fast data access and analysis, such as finance, telecommunications, and web services.

In simple terms, HBase is like a giant, fast-access spreadsheet that can store and retrieve massive amounts of data across many servers, making it useful for big data applications that require real-time access.



Apache HBase Architecture

Git


Working Tree
Staging Area
Local Repository
Remote Repository
git add/mv/rm
git commit
git push
git reset <file>
git fetch
git reset <commit>
git diff
git diff HEAD
git clone/pull