Notes on Petabyte-Scale File Systems and Distributed Computing

1. Warehouse Scale Computers (WSCs)

Definition & Role

- **Warehouse Scale Computers (WSCs)** are **large-scale data centers** optimized for **massive computational workloads**.
- Unlike traditional **multi-tenant data centers**, WSCs are **owned by a single organization** and designed for **homogeneous hardware and system software**.

Structure & Challenges

- **Basic Building Blocks**:
  - **Rack units** with CPUs, GPUs/TPUs, memory, and disks.
- **Key Challenges**:
  - **Power Distribution**: High energy consumption requires strategic power sourcing.
  - **Cooling**: Large-scale operations generate **massive heat** that must be managed efficiently.
  - **Network Bottlenecks**: Moving petabytes of data across machines can be slow.
  - **System Failures**: Machines fail over time, so **fault tolerance is critical**.

Scaling Beyond a Single Machine

- **Why not a single node?**
  - **Example from Google**:
    - **10 billion web pages**, averaging **20KB each → 200TB total**.
    - **Disk read bandwidth = 50MB/s**, meaning a **single disk would take ~46 days** to read the data.
  - **Solution**:
    - **Data is split into chunks** and stored across multiple machines.
    - **Parallel processing** speeds up read and compute times.

---

2. Distributed File Systems (DFS)

Concept & Evolution

- **A Distributed File System (DFS) allows multiple nodes to store and process data as if it were on a single system**.
- **Evolution**:

Notes on Petabyte-Scale File Systems and Distributed Computing

- Traditional file systems had limitations such as:
    - **Physical storage constraints** (a single drive can't handle petabytes).
    - **Scalability issues** (millions of users accessing data simultaneously).
    - **Lack of redundancy** (single drive failures could lead to data loss).
- **DFS introduced solutions**:
    - **Uniform namespace**: No need to know physical file locations.
    - **Redundancy**: Files are **replicated across multiple nodes**.
    - **Atomic operations**: Some actions (file creation, deletion) must be consistent.

Key DFS Systems

DFS	Key Features	Impact
Google File System (GFS)	Large-scale DFS for Google's internal needs. Uses **chunk replication** and a **single master node**.	**Foundation for Hadoop HDFS**. Led to later innovations.
Hadoop Distributed File System (HDFS)	Open-source DFS inspired by GFS. Built for **big data processing**. **Optimized for batch jobs**.	**Powers Hadoop & Spark workloads**in big data analytics.
Cloud Storage (AWS, GCP, Azure)	Managed DFS-like services. **Scalable, redundant, automated storage**.	Used in **modern applications** without needing in-house infrastructure.

---

3. Google File System (GFS)

Why GFS Was Needed

- **Google needed a system to handle petabyte-scale data efficiently**.
- **Traditional file systems couldn't meet the requirements** of:
    - **High availability** despite hardware failures.
    - **Efficient storage for massive files** (multi-GB).
    - **Fast read speeds for large datasets**.

GFS Key Features

- **Files stored in large chunks** (typically **64MB**).
- **Master node manages metadata**, while **chunkservers store actual data**.
- **Replication for fault tolerance**: Each chunk is replicated **across 3 machines**.
- **Optimized for append-heavy workloads** (rather than random writes).

Notes on Petabyte-Scale File Systems and Distributed Computing

Impact

- GFS powered **Google Search, Gmail, YouTube, and other large-scale services**.
- Inspired **Hadoop Distributed File System (HDFS)**.

---

4. Hadoop Distributed File System (HDFS)

What is HDFS?

- **An open-source alternative to GFS**, used for **big data analytics** in the **Hadoop ecosystem**.
- **Designed to store & process petabyte-scale datasets** efficiently.

Key Components

Component	Function
NameNode	Stores **metadata** (file structure, chunk locations, replication factor).
DataNodes	Store **actual file blocks** (typically 128MB per block).
Clients	Communicate with the NameNode for metadata and retrieve file chunks from DataNodes.

Key Features

- **Write-once, read-many model**.
- **Replication for fault tolerance** (default **3 copies per block**).
- **Optimized for batch processing** (e.g., Hadoop MapReduce).
- **Rack-aware placement**:
  - Replicas **spread across racks** to minimize failure risks.
  - Balances **performance vs. disaster recovery**.

HDFS vs. GFS

Feature	GFS	HDFS
Owner	Google (internal)	Open-source (Apache)

Notes on Petabyte-Scale File Systems and Distributed Computing

Use Case	Google's internal services (Search, Gmail, etc.)	Hadoop-based analytics (big data processing)
File Access	Optimized for **Google workloads**	Optimized for **batch jobs**

---

5. MapReduce: Solving Large-Scale Processing Challenges

Challenges in Cluster Computing

- **Nodes can fail** (machines typically last ~3 years).
- **Network bottlenecks**: Moving **10TB over a standard network takes ~1 day**.
- **Distributed programming is difficult**.

MapReduce as a Solution

- Developed at **Google** to **automate parallel computation** on large datasets.
- **Hadoop implemented MapReduce on top of HDFS**.

How It Works

1. **Map Phase**: Data is **processed in parallel** across multiple nodes.
2. **Shuffle & Sort**: Data is reorganized for efficient processing.
3. **Reduce Phase**: Aggregated results are **combined into a final output**.

Connection to GFS & HDFS

- **MapReduce depends on distributed file systems** to process large datasets efficiently.
- **GFS & HDFS provide the necessary storage backend**.

---

6. Cloud Storage: The Evolution Beyond Traditional DFS

How Cloud Storage Builds on DFS

- **Traditional DFS (e.g., GFS, HDFS) requires on-premises infrastructure**.
- **Cloud Storage provides the same benefits as DFS but as a service**.
- Examples:

Notes on Petabyte-Scale File Systems and Distributed Computing

- **Amazon S3**
- **Google Cloud Storage**
- **Microsoft Azure Blob Storage**

Key Differences Between DFS and Cloud Storage

Feature	DFS (GFS, HDFS)	Cloud Storage (AWS S3, GCP)
Management	Self-managed (requires infrastructure)	Fully managed
Scalability	Limited to owned hardware	Virtually infinite scalability
Access Model	POSIX-like file system	Object storage (buckets, metadata-driven)

Impact of Cloud Storage

- **Removes the need for organizations to maintain their own DFS**.
- **Allows dynamic scaling** with built-in redundancy and replication.

Final Takeaways

1. **Warehouse Scale Computers (WSCs) provide the infrastructure for petabyte-scale storage and computation.**
2. **Distributed File Systems (DFS) like GFS & HDFS enable efficient storage across multiple nodes.**
3. **Google File System (GFS) revolutionized large-scale storage and inspired open-source alternatives like HDFS.**
4. **Hadoop Distributed File System (HDFS) is widely used in big data applications, optimizing batch processing.**
5. **MapReduce simplifies large-scale distributed computation, leveraging DFS for efficient data access.**
6. **Cloud Storage is the modern evolution of DFS, offering scalable, fully managed storage without infrastructure maintenance.**