**Data Engineering   
Assignment 1**

**Fahad Zubair  
24280079**

**Hadoop Distributed File System**

**Summary:**

Hadoop Distributed File System is a popular choice for many organizations, including Yahoo, because of its ability to save copies of data across multiple nodes to ensure data safety and is highly scalable so it can process up to petabytes of data.

It can handle a very large amount of data up to petabytes and is easy to expand by adding more computers.

HDFS works differently than regular file systems. It processes data near where it's stored (called data locality) and handles many tasks at the same time (parallel processing). It runs on cheap, easily available hardware, making it cost-effective.

NameNode acts as the dictionary of HDFS because it contains all the information on which Data Node what data is stored. Request are generated from HDFS client for

**Pros:**

Scalable Storage: It can handle huge amounts of data and you can grow it by just adding more computers.

Data Locality: It processes data near where it's stored, making things faster.

Write-Once Model: Data is written once and read many times. This is good for analysis tasks.

Fault tolerance: HDFS replicates data across multiple nodes, so if one fails, the data isn’t lost. It’s built for reliability using cheap hardware.

Scalable storage: It easily handles petabytes of data and grows as needed just by adding more nodes.

**Cons:**

Single Point of Failure: If the NameNode fails, the whole system can stop. Even with backups, fixing it can take time.

Small File Problem: HDFS is not good at handling lots of tiny files because each one takes memory.

Not for Real-Time Use: HDFS is made for batch processing, not for real-time or fast tasks.

Hard to Manage: Running an HDFS system needs effort. Cloud services like Amazon S3 are easier to use and manage.

**My critique:**

HDFS is really good for storing and processing huge amounts of data. It also keeps your data safe if hardware fails. But it has problems too. The NameNode can cause issues if it fails, and recovery takes time.

Also, cloud systems today are easier to scale and better for real-time work. So while HDFS is still useful for big offline jobs, I wouldn’t use it for modern fast-moving data projects.