Tutorial 10: Distributed File Systems

Question Set:

Q1. What are the desirable features of Distributed File System (DFS)?

ANSWER:

Transparency:

- ***Access transparency***: clients are unaware that files are distributed and can access them in the same way as local files are accessed.
- ***Location transparency***: a consistent namespace exists encompassing local as well as remote files. The name of a file does not give its location.
- ***Concurrency transparency***: all clients have the same view of the state of the file system. This means that if one process is modifying a file, any other processes on the same system or remote systems that are accessing the files will see the modifications in a coherent manner.
- ***Failure transparency***: the client and client programs should operate correctly after a server failure.
- ***Replication transparency***: clients should be unaware of the file replication performed across multiple servers to support scalability.
- ***Migration transparency***: files should be able to move between different servers without the client's knowledge.

Heterogeneity: the file service should be provided across different hardware and operating system platforms.
Scalability: the file system should work well in small environments (e.g. 1 machine, a dozen machines) and also scale gracefully to bigger ones (e.g. hundreds through tens of thousands of systems).

Q2. What is a Clustered File System? Please use one sentence to answer this question.

ANSWER:

A clustered file system (CFS) is not a single server with a set of clients, but instead a cluster of servers that all work together to provide high-performance service to their clients.

Q3. Read the following materials [1] and describe each step in READ and WRITE operations in Google File System (GFS).

ANSWER:

Read:

- When the application is reading a file, the application needs to go through the GFS client. The client can send the **file name** and **chunk index** to the GFS master.
- After receiving the request, the GFS master will check its table and return the **chunk handle** and all the **chunk server IP addresses** (locations) where that chunk has the replicas.

- The client **caches** this information using the file name and chunk index as the key. The client then sends a request to one of the replicas, most likely the closest one. The request specifies the **chunk handle** and a **byte range** within that chunk.
- The chunkserver will send the **requested data** to the client thus requiring no more client-master interactions until the cached information expires or the file is reopened.
- The client forwards the data to the application

Write:

- Client asks master which chunk server holds **current lease of chunk** and **locations of other replicas**.
- Master replies with identity of **primary** and locations of **secondary replicas**.
- Client pushes data to all replicas.
- Once all replicas have acknowledged receiving the data, client sends write request to primary. The primary assigns **consecutive serial numbers** to all the mutations it receives, providing serialization. It applies mutations in serial number order.
- Primary **forwards write request** to all secondary replicas. They apply **mutations** in the same serial number order.
- Secondary replicas **reply** to primary indicating they have **completed** operation.
- Primary replies to the client with **success** or **error message**.

Q4. Please compare the differences between Google File System (GFS) and Hadoop Distributed File System (HDFS) [2].

Hadoop Distributed File System (HDFS) was inspired by Google File System (GFS). They have the similar design motivations and architecture (Clustered File System). However, there are some differences as follows:

	Hadoop Distributed File System (HDFS)	**Google File System (GFS)**
Platform	Cross Platform (Linux, Mac, Windows)	Linux
Development	Developed in Java environment	Developed in C, C++ environment
Chunk Size	128 MB	64 MB
Node	Name/Data Nodes	Master node & Chunk Server
Log	Editlog	Operational log
Write Operation	No more than one writer at one time	Can have multiple writers to one file at one time.
File Deletion	Deleted files are renamed into a particular folder and then it will be removed via garbage	Deleted files are not reclaimed immediately and are renamed in a hidden namespace and they will deleted after three days if it's not in use

References

[1] Ghemawat, S.; Gobioff, H.; Leung, S. T. (2003). The Google file system, Proceedings of the nineteenth ACM Symposium on Operating Systems Principles - SOSP'03, October 19–22, Bolton Landing, New York, USA.

(http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf)

[2] Shvachko, Konstantin, et al. "The hadoop distributed file system." 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE, 2010.