Ethics Pledge

Consistent with the above statements, all homework exercises, tests and exams that are designated as individual assignments MUST contain the following signed statement before they can be accepted for grading.

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

Signature: Haodong Zhao Date: Feb 5th. 2019

Please note that assignments in this class may be submitted to

www.turnitin.com, a web-based anti-plagiarism system, for an evaluation of their originality.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reading review**

**The Google File System**

The Google file system (GFS) is an extensible distributed file system for large distributed data intensive applications. The file system is widely deployed in Google as a storage platform for generating and processing data used by services and for R&D work that requires a large data set.

GFS has many of the same goals as previous distributed file systems: performance, scalability, reliability, and availability. The article mentions 4 different views from the past:

1. Component failure is the norm rather than the exception. Therefore, continuous monitoring, error detection, fault tolerance and automatic response must be part of the system.
2. The documents of traditional standards are large. The size of the data set is growing rapidly, and design assumptions and parameters must be reconsidered.
3. Most files mutate by appending new data instead of overwriting existing data.
4. By increasing flexibility, co-designing applications and file system APIs benefits the entire system.

The design overview of GFS:

1. When designing file systems based on requirements, we always follow assumptions that provide both challenges and opportunities.
2. The file system interfaces provided by GFS are: normal operations such as create, delete, open, close, read and write, as well as snapshot and record append operations.
3. GFS cluster consists of a single primary server and multiple servers accessed by multiple clients.
4. Only one master device can greatly simplify our design, enabling the master device to use global knowledge for complex block placement and copy decisions.
5. Chunk size is a key design parameter. The GFS chooses a chunk that is much larger than a typical file system chunk size.
6. Main memory stores three main types of metadata: In-Memory Data Structures, Chunk Locations, Operation Log.
7. GFS has a loose consistency model that supports highly distributed applications and is relatively simple and efficient to implement.

System interaction: GFS minimizes the master’s involvement in all operations.

The role of the main server is: perform all namespace operations, manage chunk replication in the entire system.

GFS’s Fault tolerance and diagnostics: One of the biggest challenges in designing systems is dealing with frequent component failures.

1. GFS guarantees high availability in 2 ways: fast recovery and replication, where replication in turn includes chunk replication and master replication.
2. GFS has data integrity: each chunk server verifies the integrity of its own copy by using a checksum.
3. Extensive and detailed diagnostic logging is a significant aid in problem isolation, debugging, and performance analysis, and GFS keeps these logs as much as possible, allowing for continuous online monitoring.

From this paper, we learned that GFS has become an important tool that allows us to continue to innovate and attack the problems in entire network.