Research Paper: The Google File System

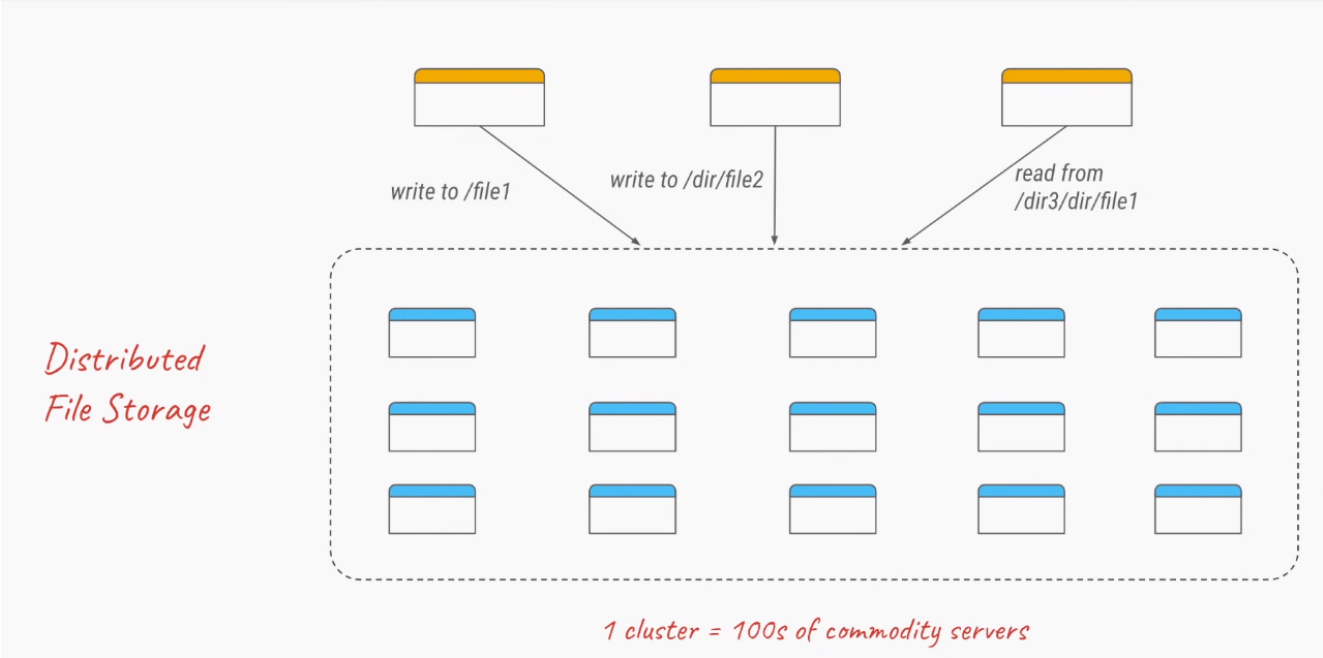
Source: https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf

Introduction

- Published in 2003
- Describes the distributed File System used by Google internally.
- This paper and corresponding architecture was the basis for the creation of Hadoop. The corresponding file system was called Hadoop Distributed FIle System.

What is Google File System?

- Google file system is essentially a distributed file storage.
- Any given cluster can contain anywhere from 100s to 1000s of Comodity servers.
- This cluster provides an interface for n number of clients to either a read a file or write a file.
- Essentially, a file system, distributed over many servers.



Design Considerations (Tradeoffs made when designing this particular architecture)

1. **It uses commodity hardware**

- Built by Google when it was still a startup, they opted for off-the-shelf commodity hardware instead of expensive servers.
- This commodity hardware was cost-effective.

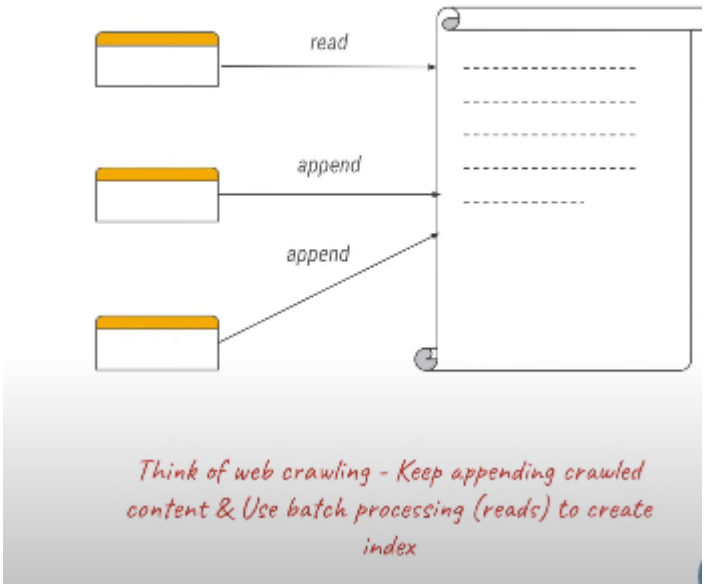
- With the right software layer, scalability could be achieved through horizontal scaling.
- Commodity hardware frequently fails, including disk failures, network issues, and server crashes.
- OS bugs and human errors are also factors.
- The challenge was to create a file system capable of enduring the aforementioned issues and still perform all functions in a fault-tolerant manner.

2. Large Files

- The file system is tailored for storing and reading large files.
- Typical files in GFS range from 100 MB to multi-GB files.
- Suitable for crawled web documents and batch processing.

3. File Operations

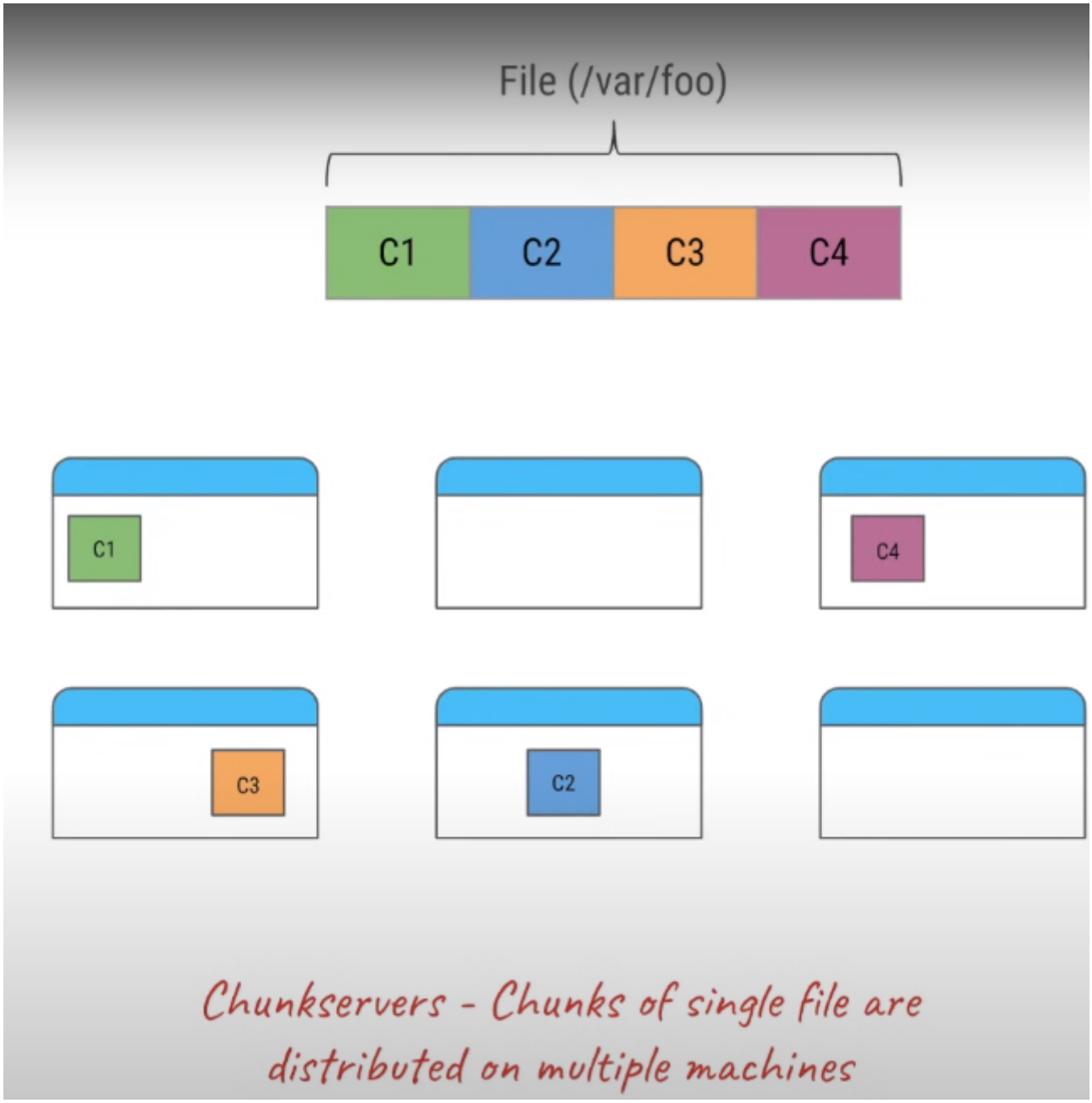
- GFS is optimized for two types of file operations:
  - Read and append-only operations.
  - No random writes, primarily sequential reads.



4. Chunks

- A single file isn't stored on one server; it's divided into multiple 64MB chunks.
- Each chunk has a 64-bit ID.

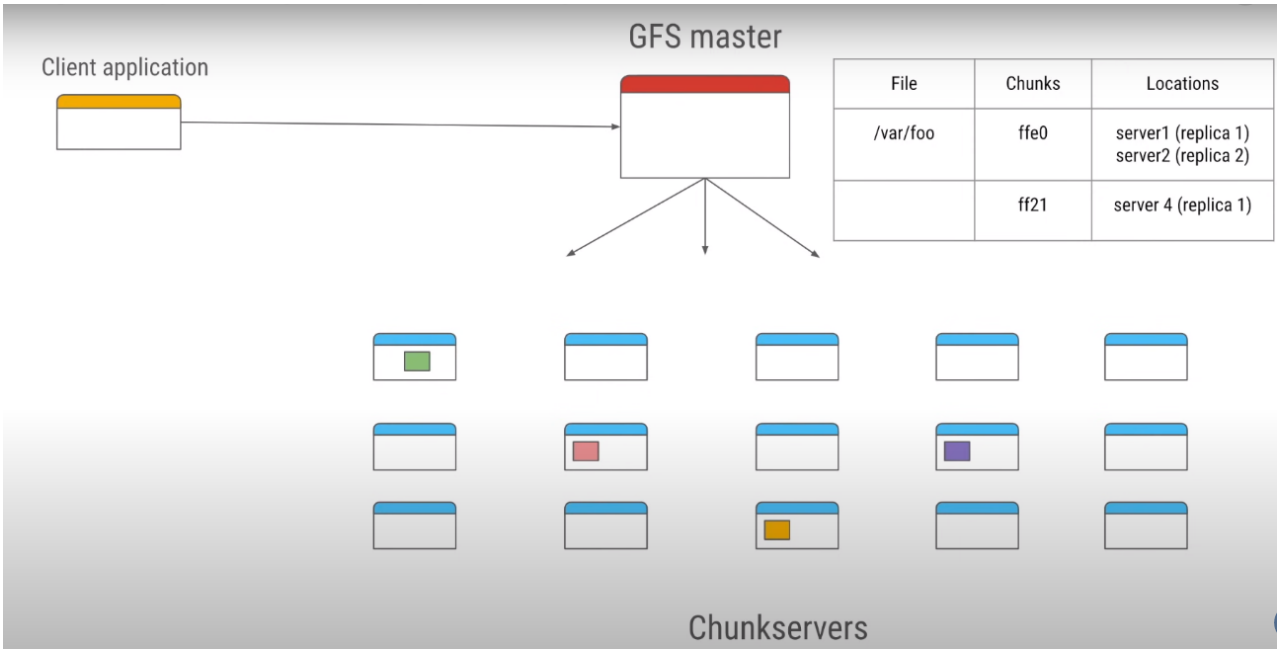
- These chunks are distributed across multiple commodity machines, also known as chunk servers.



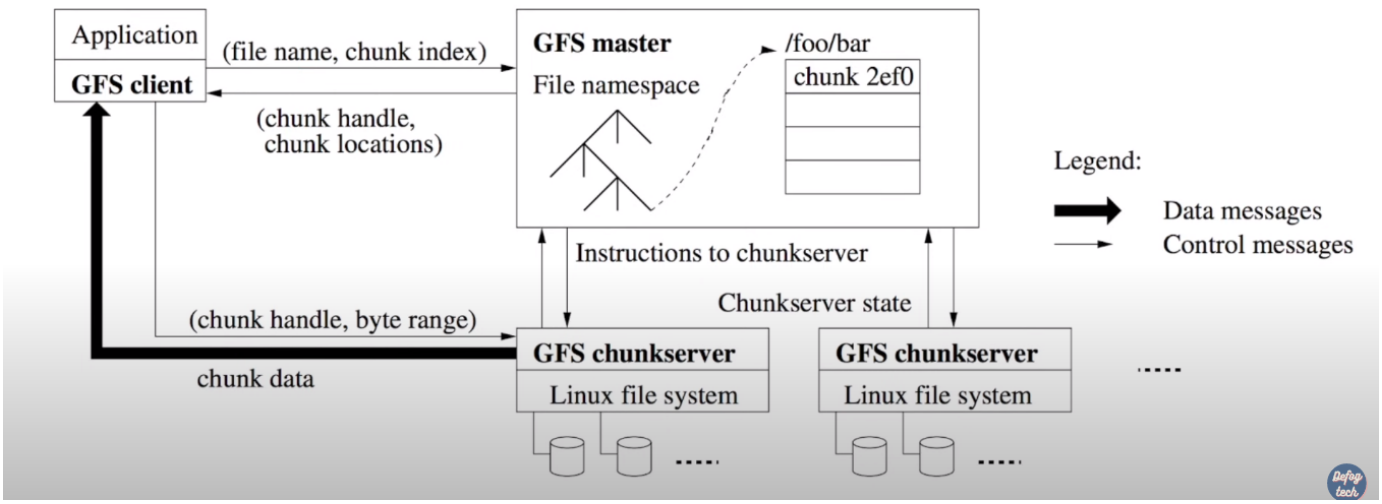
5. Replicas

- GFS ensures that each file chunk has at least three replicas on three different servers, providing fault tolerance.
- The default replica count is three but can be modified by the client.
- Instead of storing all chunk-related data on GFS or the client application, a separate component called the GFS master stores:
  - Metadata of the entire cluster
  - File names, chunk IDs, and chunk locations

- Access control details

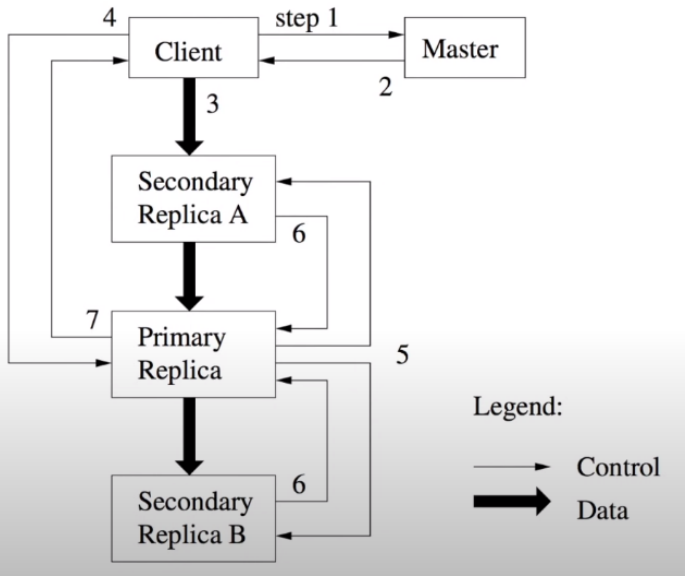


6. Reads



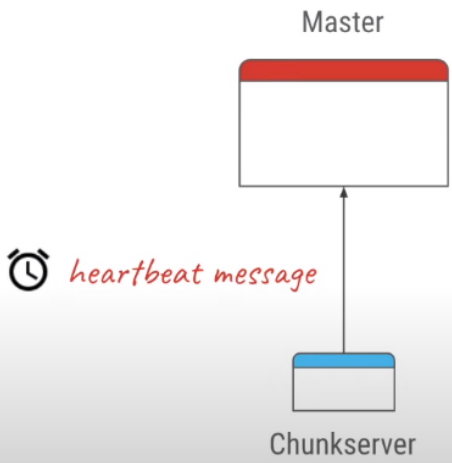
7. Writes

1. Ask for locations to write
2. Get replicate locations
3. Write data to closest replica.
4. Request commit to primary
5. Primary instructs order of writes to secondaries
6. Secondaries acknowledge
7. Primary ack to client



8. Heartbeats

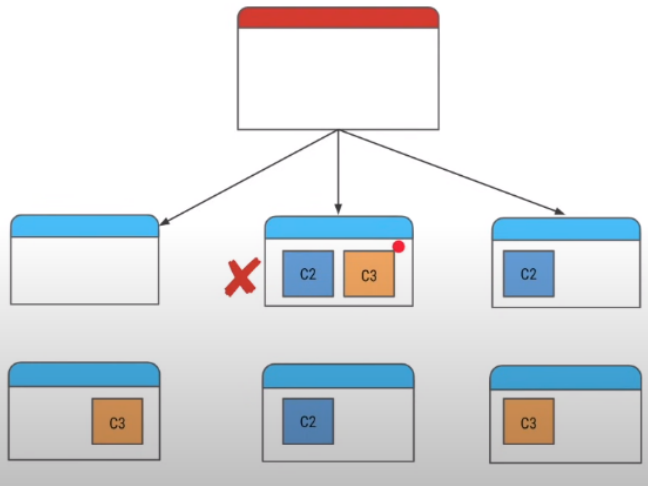
Regular heartbeats to ensure chunkservers are alive



9. Ensure Chunk Replica Count

If chunkserver is down, master ensures all chunks that were on it are copied on other servers.

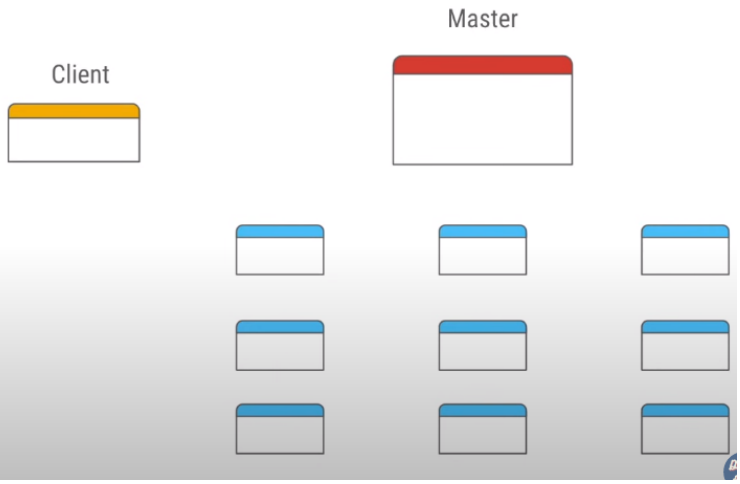
Ensures replica counts remains same.



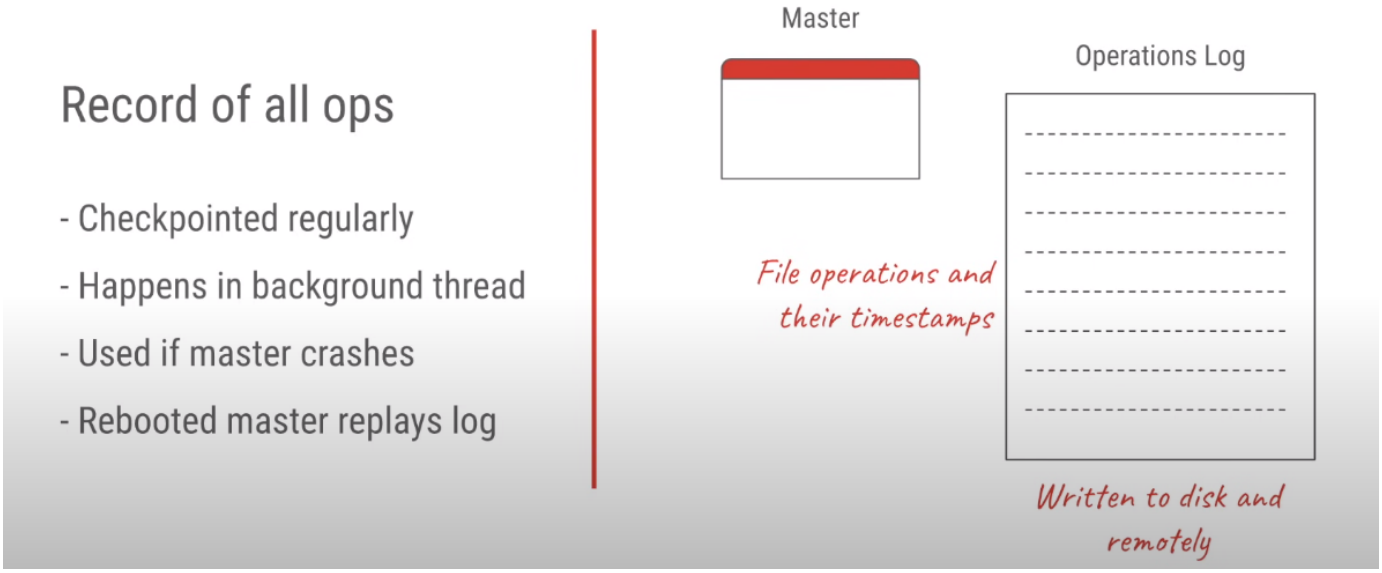
10. Single master for multi-TB Cluster

Large chunk size

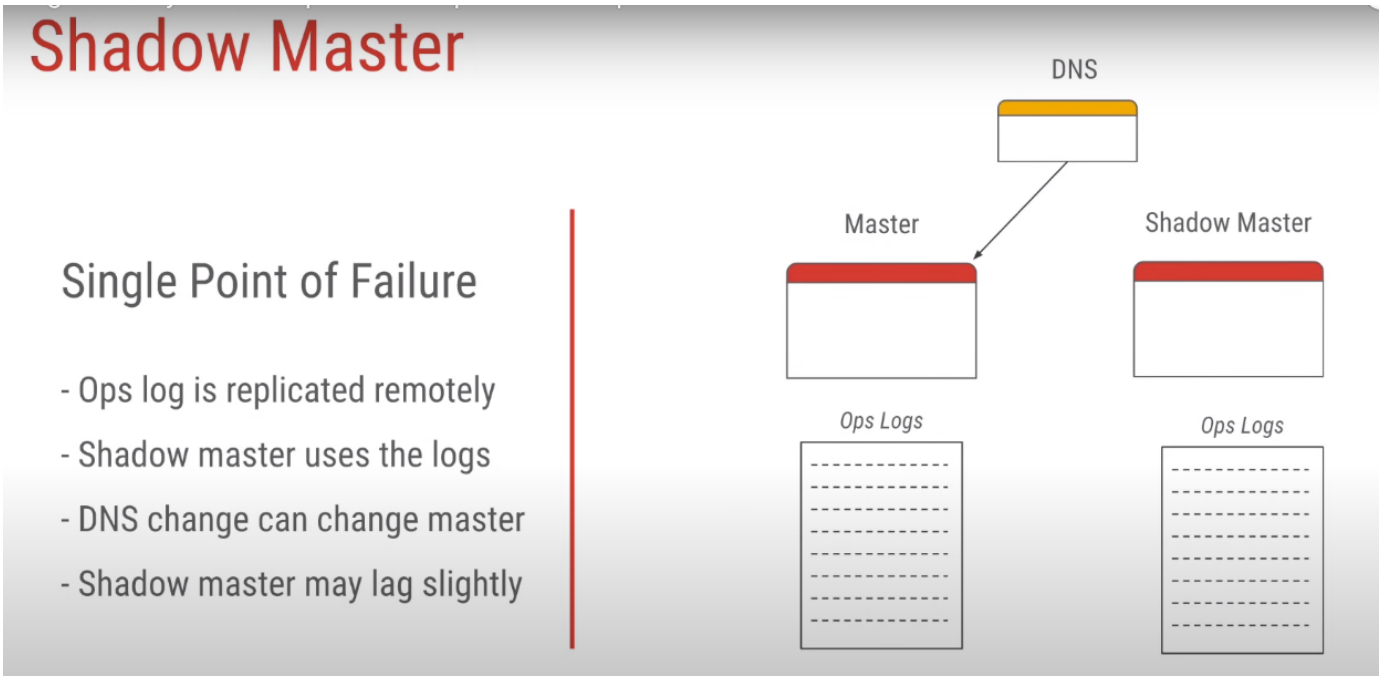
- 64MB chunk
- Reduced meta-data
- Reduces client interactions
- Client caches location data



11. Operations Log



12. Shadow Master



Other Important Points

- Chunk servers store and re-check checksums for all chunks (in 64KB blocks)
- Master on reboot asks chunkservers for the chunks they have.
- Checksum failures are reported to master.
- Master can rebalance replicas to distribute load more evenly.
- File locks for read/write
- Record Appends
- Snapshots
- Write leases provided to clients (60secs, which can be extended)
- File deletions
- Chunk Garbage Collection

Source:

- https://www.youtube.com/watch?v=eRgFNW4QFDc&t=134s
- https://medium.com/@roshan3munjal/google-file-system-gfs-overview-eed15f3e6f6e