

# **The Google File System**

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google\*

Name: QiaoDong

# CONTENTS

- 1 Introduction
- 2 How to save a file
- 3 Main problems
- 4 Write and read a file
- 5 Conclusion

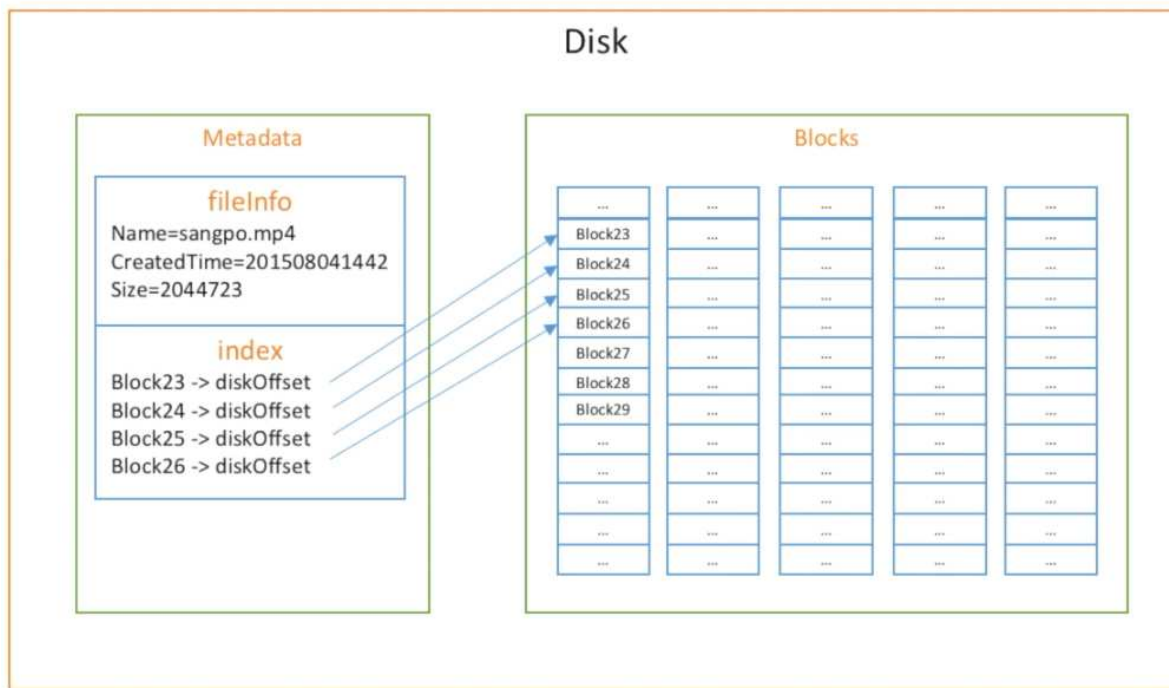
# 1. Introduction

- ❑ Google File System is a scalable *distributed* file system for large distributed data-intensive applications.
- ❑ It provides *fault tolerance* while running on inexpensive commodity hardware
- ❑ It delivers *high aggregate performance* to a large number of clients.



# 2. How to save a file

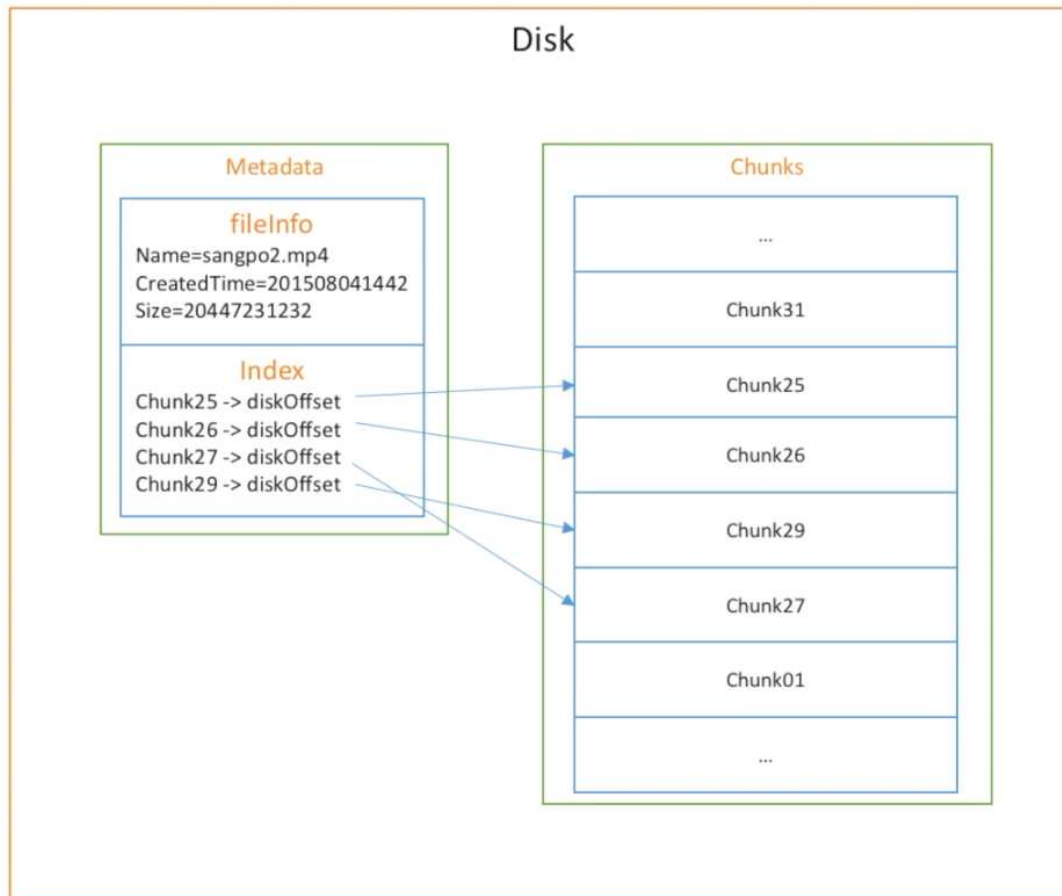
## 2.1 How to save a file?



1 Block=1024 Byte

# 2. How to save a file

## 2.2 How to save a big file?



1 Chunk=64 MB

pros :

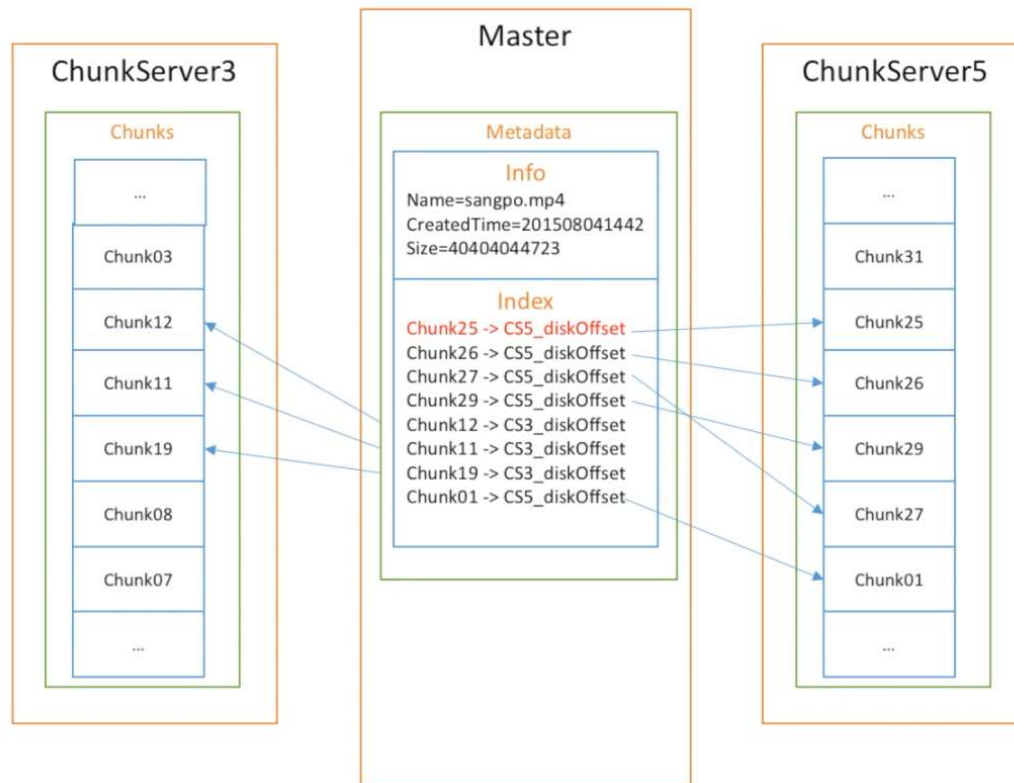
- Reduce Metadata
- Reduce flow

cons:

- Waste space for small data

## 2. How to save a file

### 2.3 How to save a super big file?



Key points:

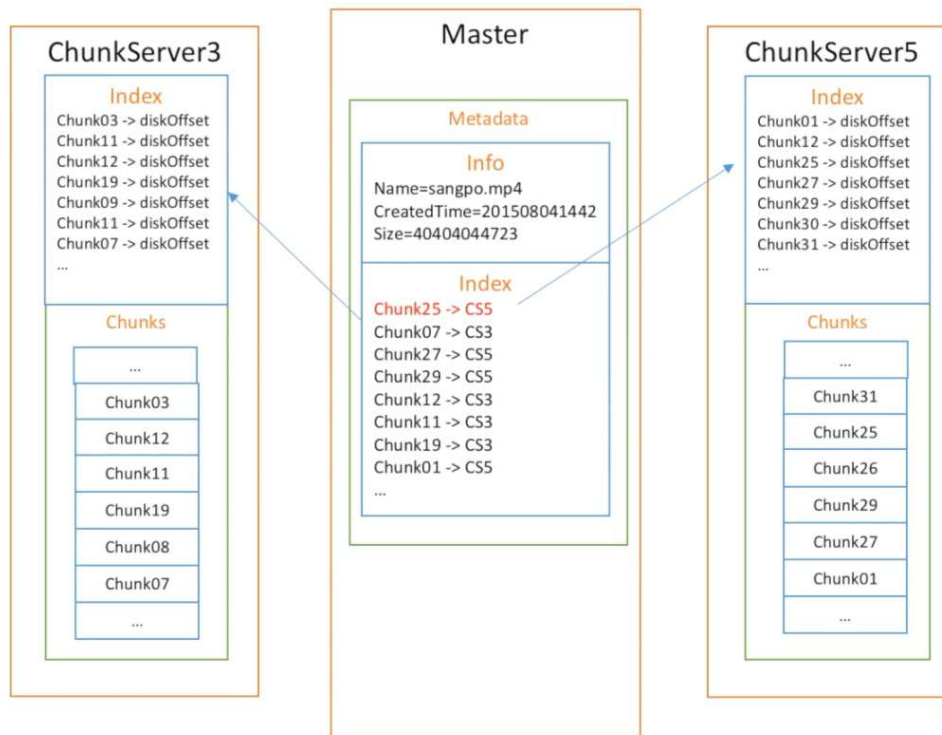
- Master + many ChunkServers

cons:

- Any change in ChunkServers must be sent to Master.

## 2. How to save a file

### 2.4 How to save flow?



Key points:

- Master just store ChunkServer number but not diskoffset.

pros:

- Reduce Metadata in Master.
- Reduce flow

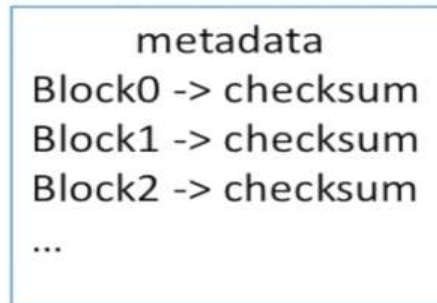
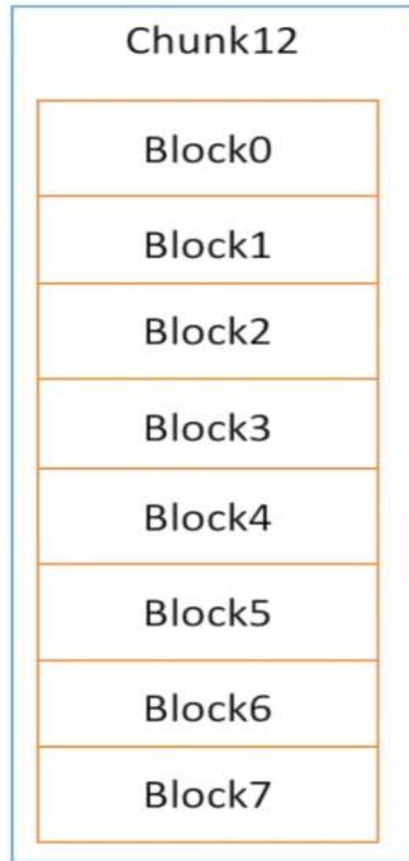
## 3. Main problems

- 3.1 How to discovery damaged data?
- 3.2 How to reduce losses when  
Chunkservers crash?
- 3.3 How recovery damaged Chunk ?
- 3.4 How to location it when chunkserver  
is dead?
- 3.5 How to handle Hotspot?



# 3. Main problems

## 3.1 How to discovery damaged data?



*Checksum !!!*

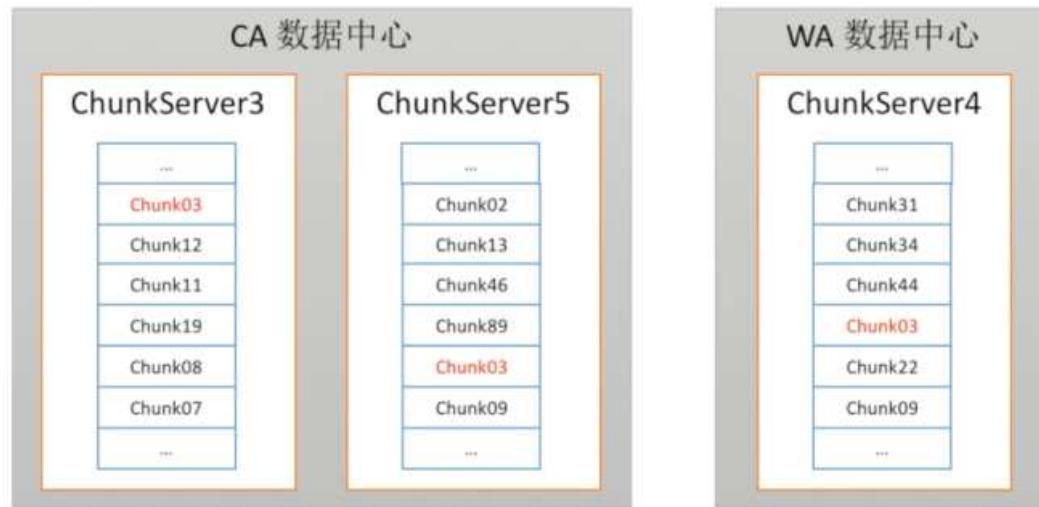
# 3. Main problems

## 3.2 How to reduce losses when Chunkservers crash?



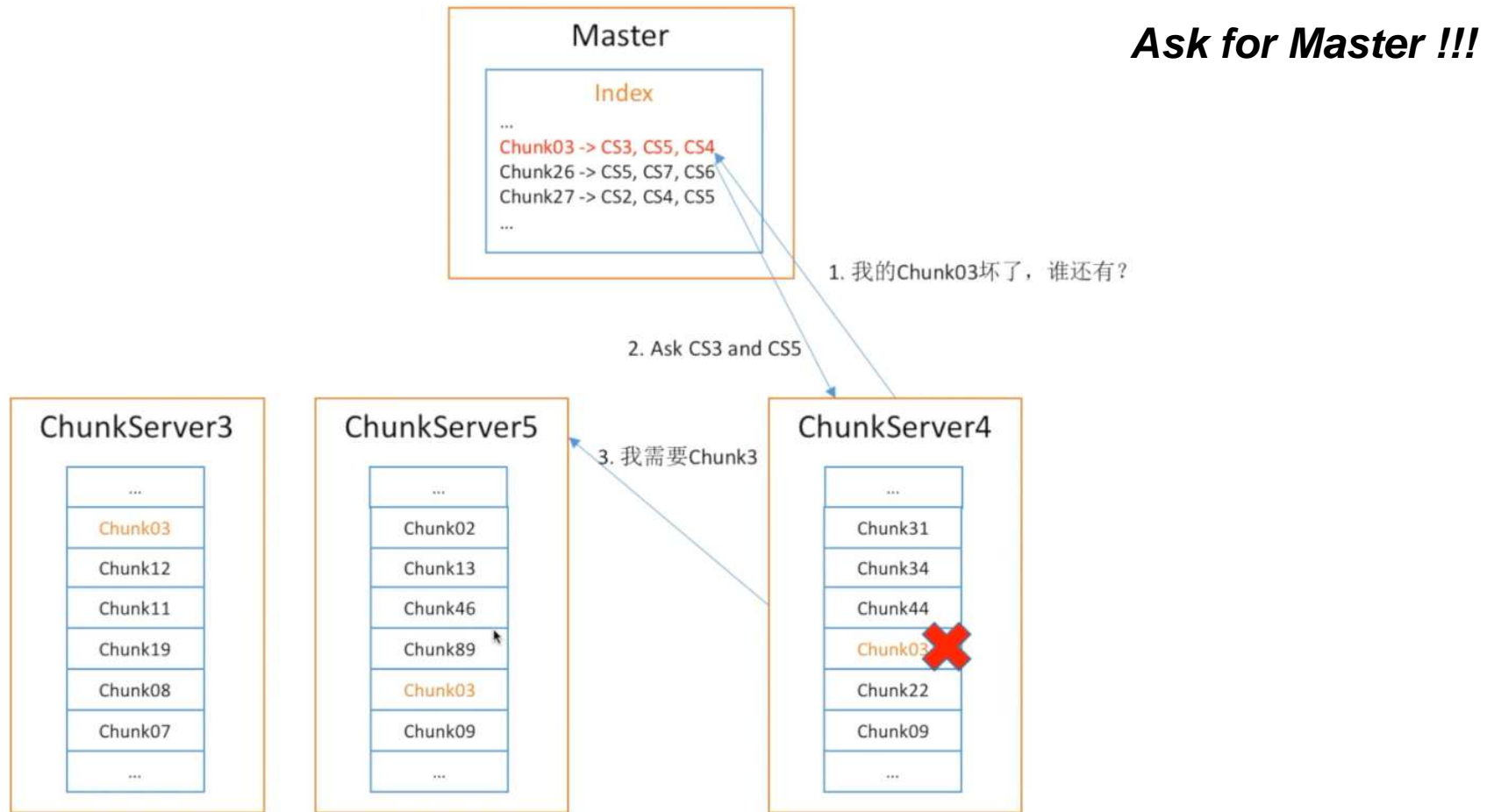
Key points:

- Copy Chunks
- The numbers of copy:  
**3**



# 3. Main problems

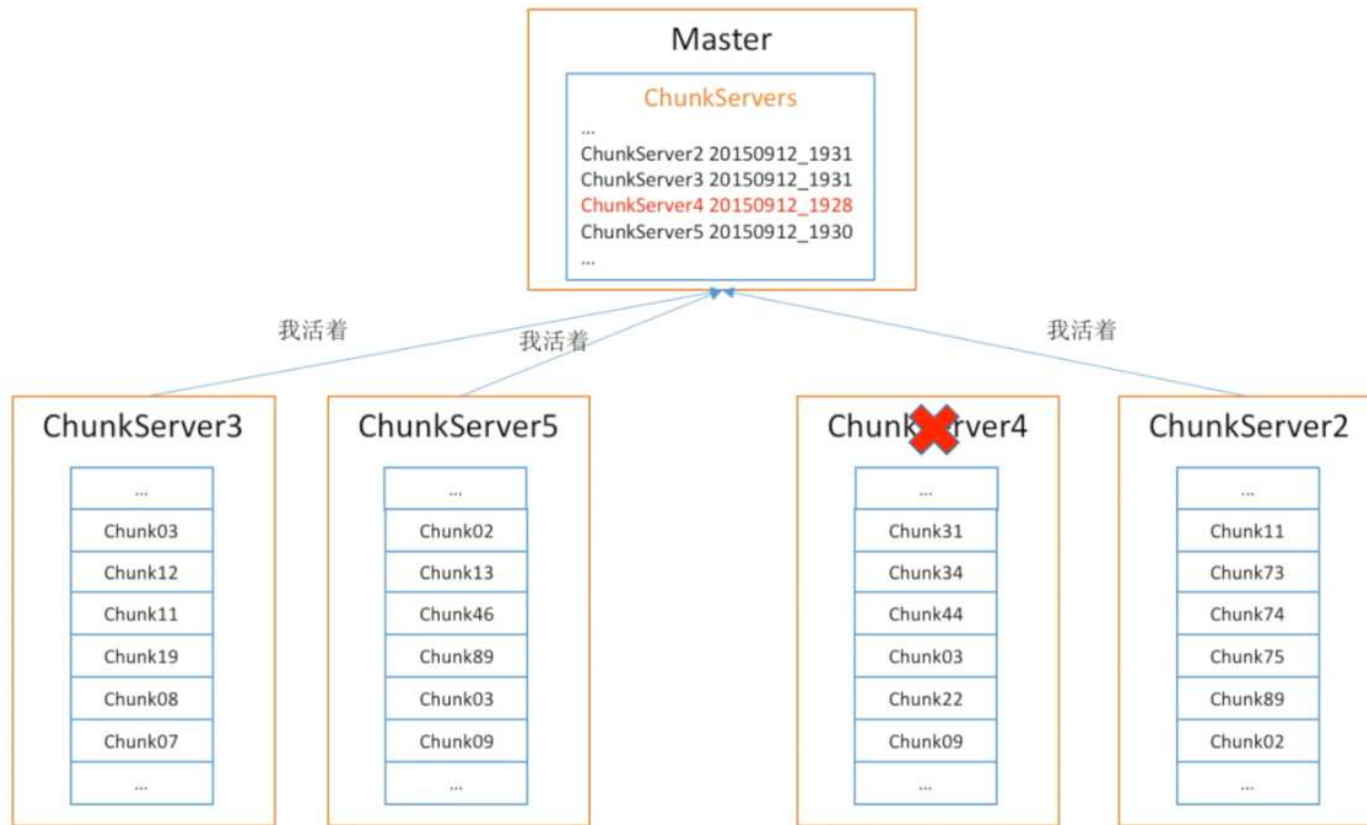
## 3.3 How recovery damaged Chunk



# 3. Main problems

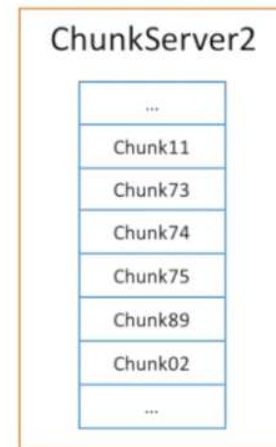
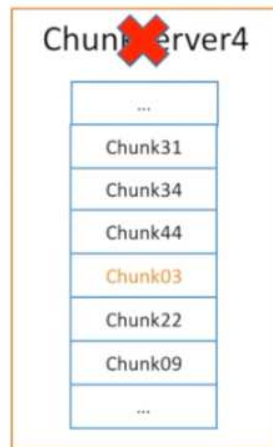
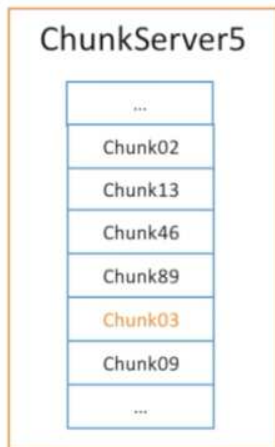
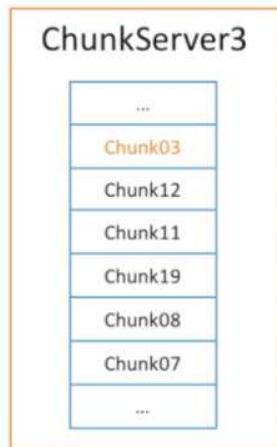
## 3.4 How to location it when chunkserver is dead?

***heartbeat !!!***



# 3. Main problems

## 3.4 How to location it when chunkserver is dead?



# 4. Read and Write

## 4.1 How to read a file?

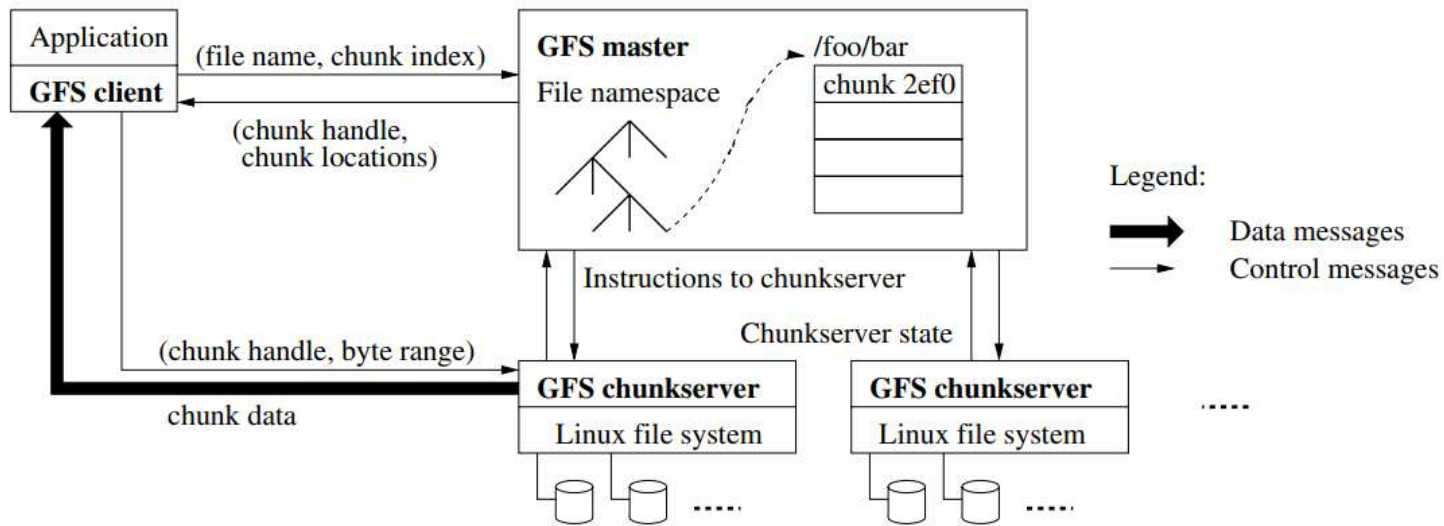
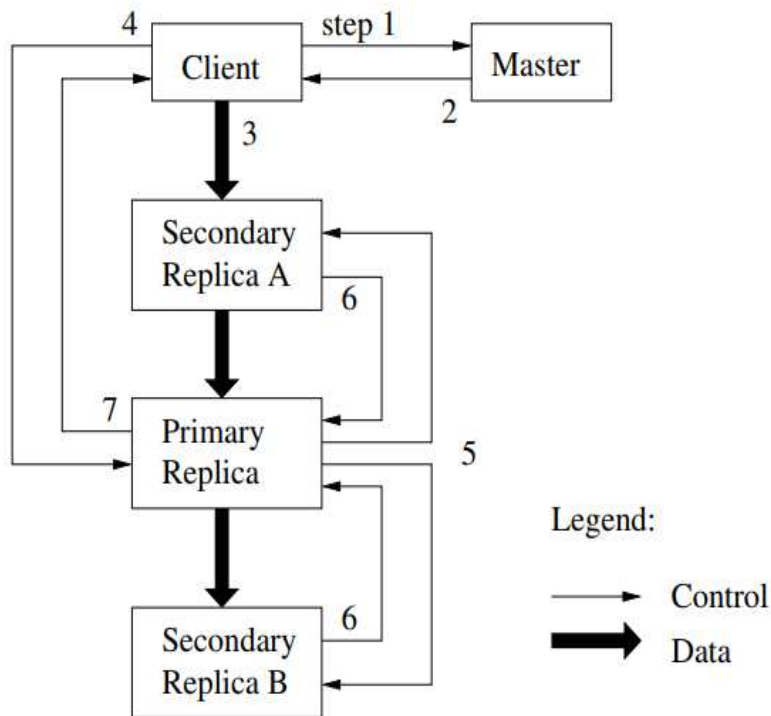


Figure 1: GFS Architecture

# 4. Read and Write

## 4.2 How to write a file?



1. Ask for write.
2. Get the primary and replica positions.
3. Send data to the nearest replica computer.
4. The Client sends write order.
5. Write order to replica computers
6. Reply the status of write.
7. Reply to Client.

# 5. Conclusion