

# The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung  
*Google*

Niek Linnenbank

Faculty of Science  
Vrije universiteit  
nlk800@few.vu.nl

March 17, 2010

1 Introduction

2 Architecture

3 Measurements

4 Latest Work

5 Conclusion

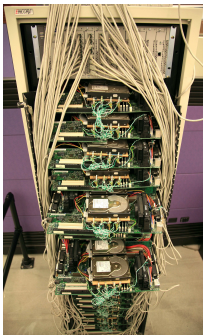
# Size of the Internet

- 6,767,805,208 people on earth
- 1,733,993,741 people on the internet
- 5,000,000 terabytes of data (Eric Schmidt, 2005)

# Top 10 Search Provider in US, January 2010

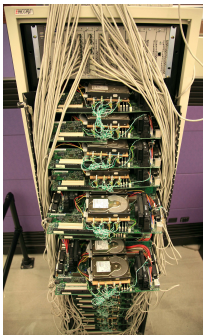
RANK	PROVIDER	SEARCHES (000)	SHARE
-	ALL SEARCH	10,272,099	100.0
1	GOOGLE SEARCH	6,805,424	66.3
2	YAHOO SEARCH	1,488,476	14.5
3	MSN SEARCH	1,116,546	10.9
4	AOL SEARCH	251,762	2.5
5	ASK.COM SEARCH	194,161	1.9
6	MY WEB SEARCH SEARCH	112,356	1.1
7	COMCAST SEARCH	59,608	0.6
8	YELLOW PAGES SEARCH	35,101	0.3
9	NEXTAG SEARCH	34,736	0.3
10	BIZRATE SEARCH	20,123	0.2

# The Google Way



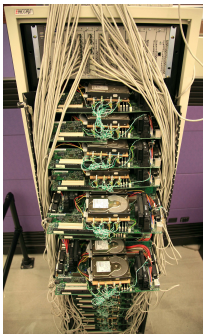
- Google does web indexing (and more)
- Cheap commodity hardware
- Patented PageRank(tm) technology

# Google Filesystem



- Scalable distributed filesystem
- Designed for cheap clusters
- Capable of storing hundreds of terabytes

# Assumptions

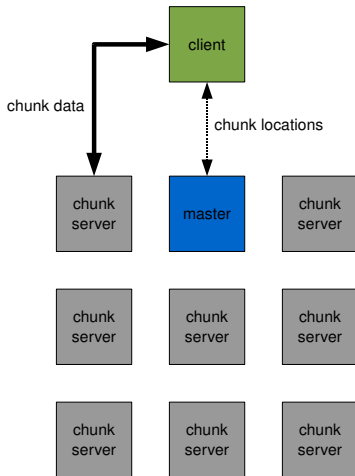


- Component failures are the norm
- Inexpensive commodity hardware
- Large files
- Files mutated with appends
- Workload typically large streaming reads and appends

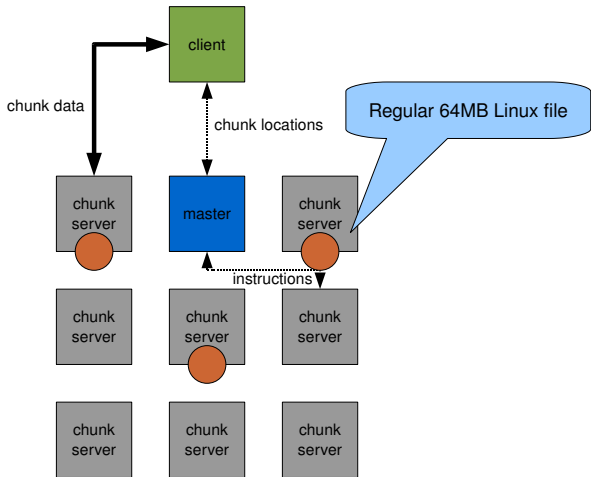
# Design

- One master process keeps file metadata.
- Files are split into chunks.
- Multiple chunkservers to store chunks.
- Multiple clients may access concurrently.
- POSIX-a-like API (create, read, write, append, delete)

# Design

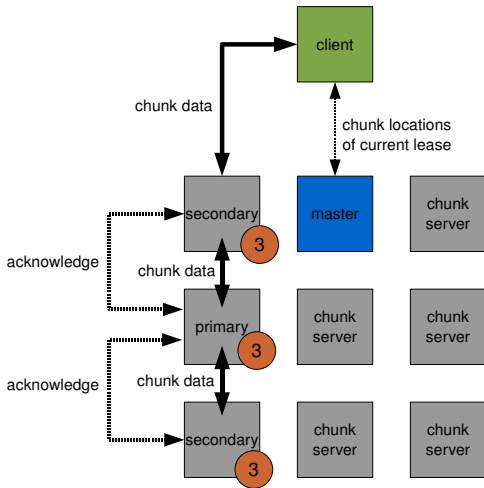


# Chunk Replicas

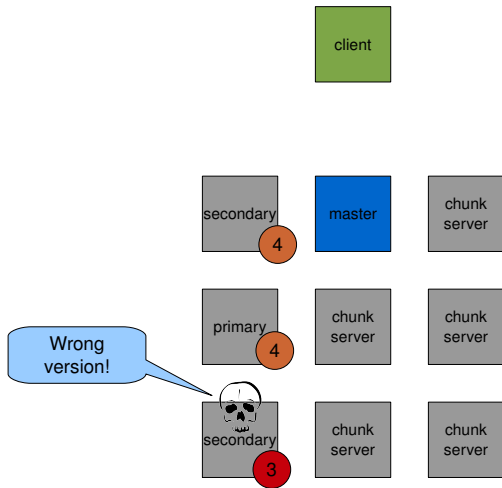




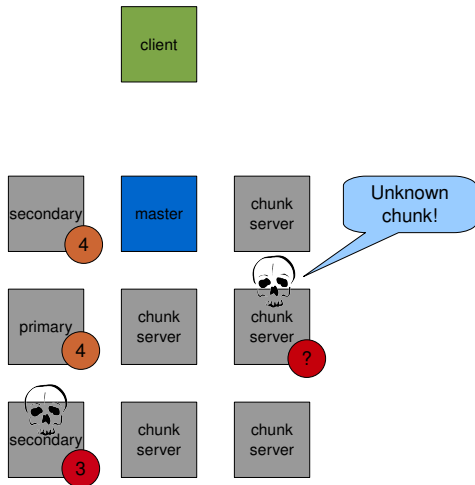
# Chunk Versions



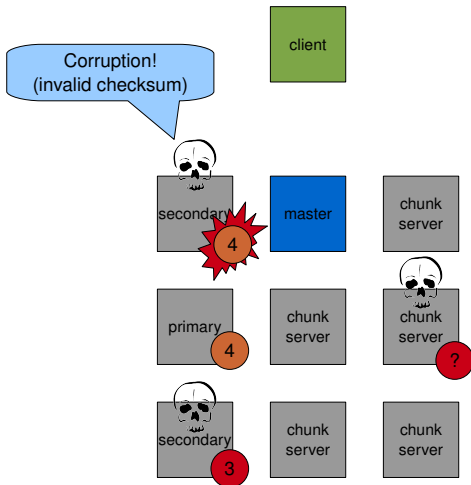
# Stale Replica



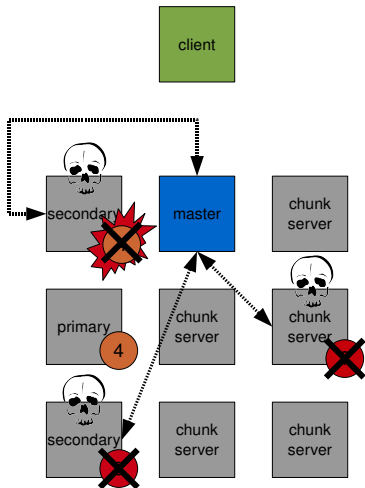
# Chunk Orphans



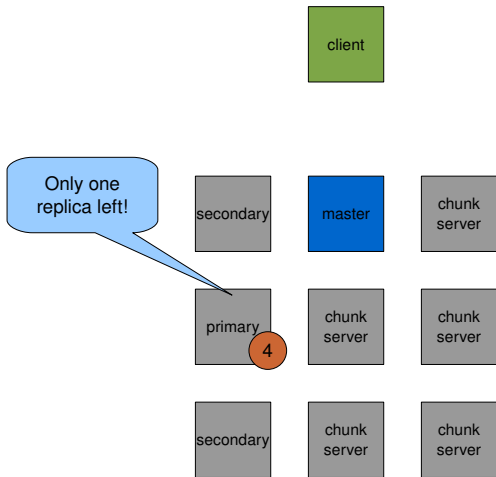
# Chunk Corruption



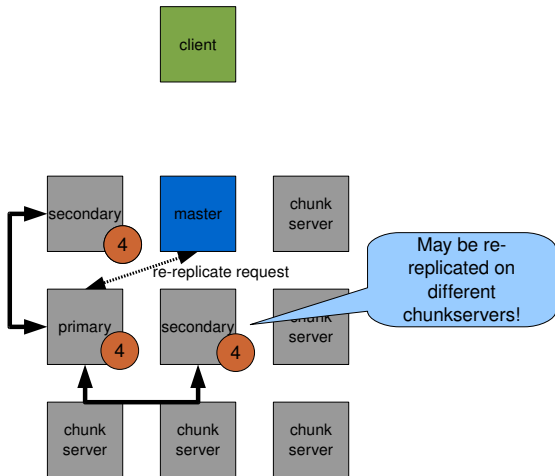
# Garbage Collection



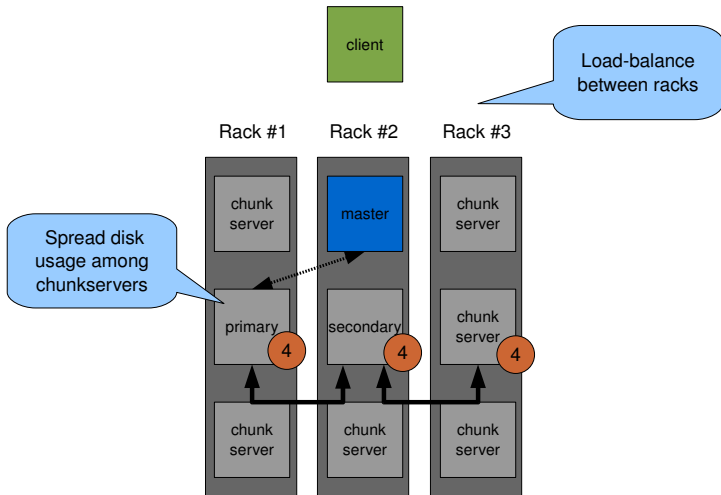
# Garbage Collection



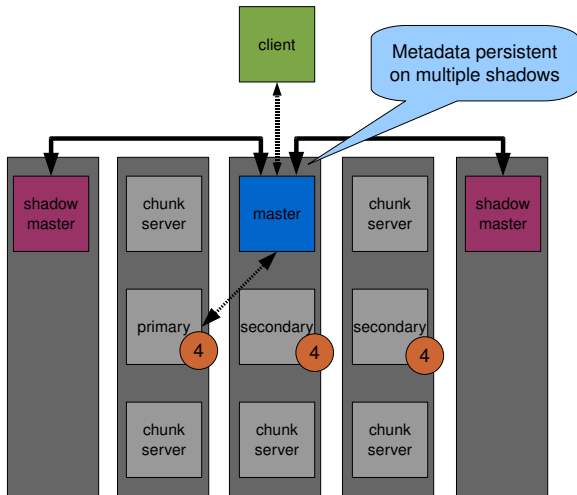
# Chunk Re-replication



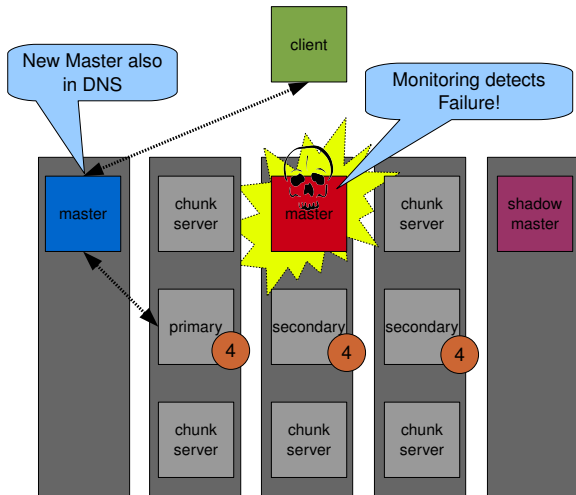
# Chunk Rebalancing



# Master State Replication

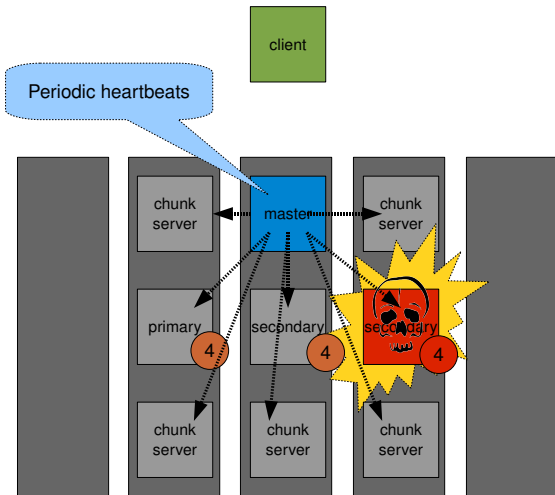


# Master High Availability

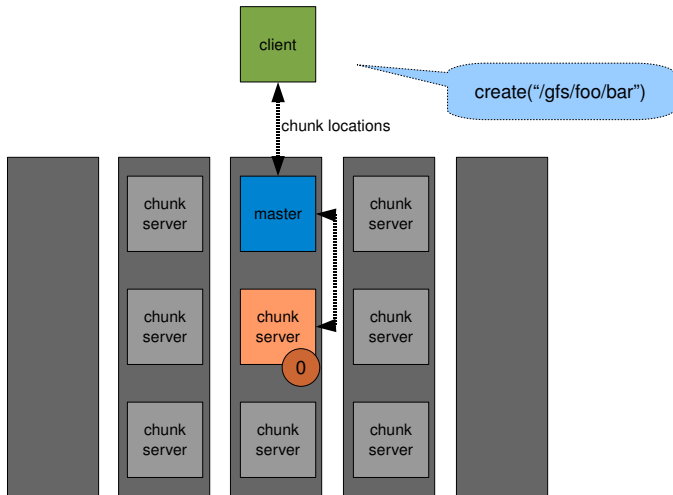




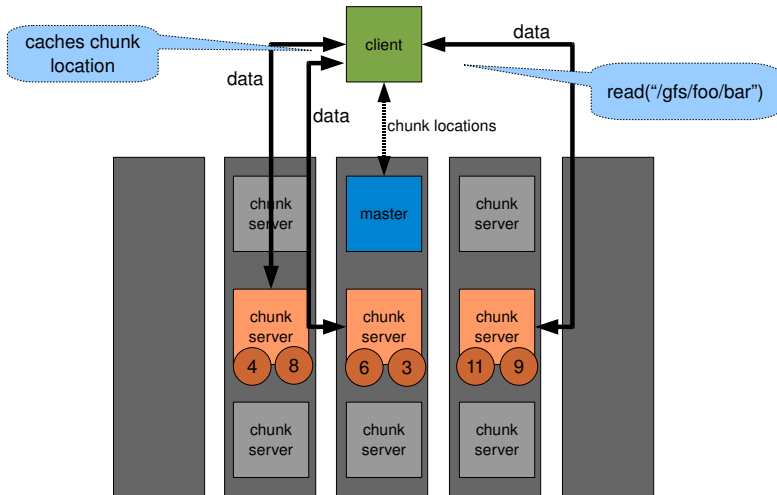
# Chunkserver High Availability



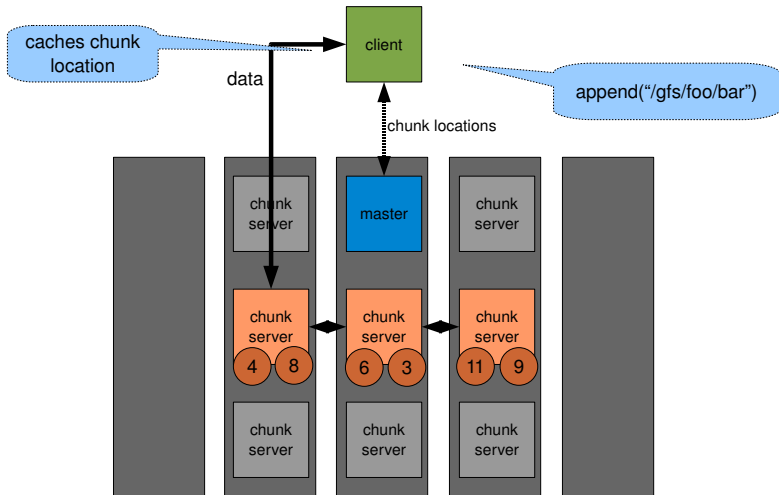
# Client API: Creates



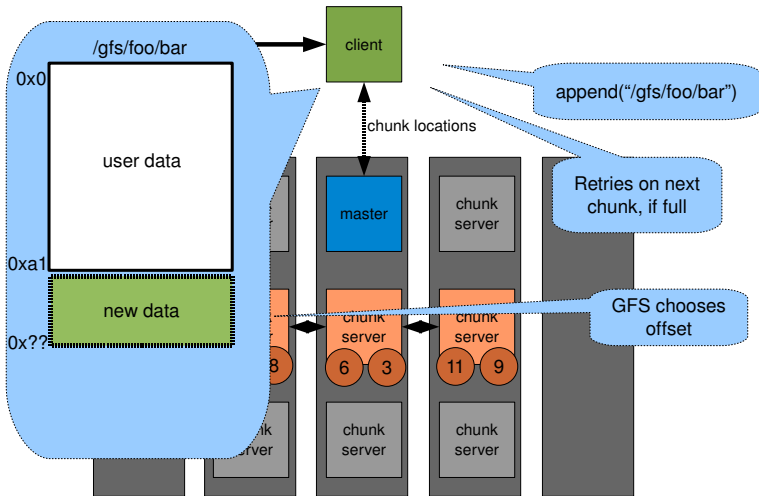
# Client API: Reads



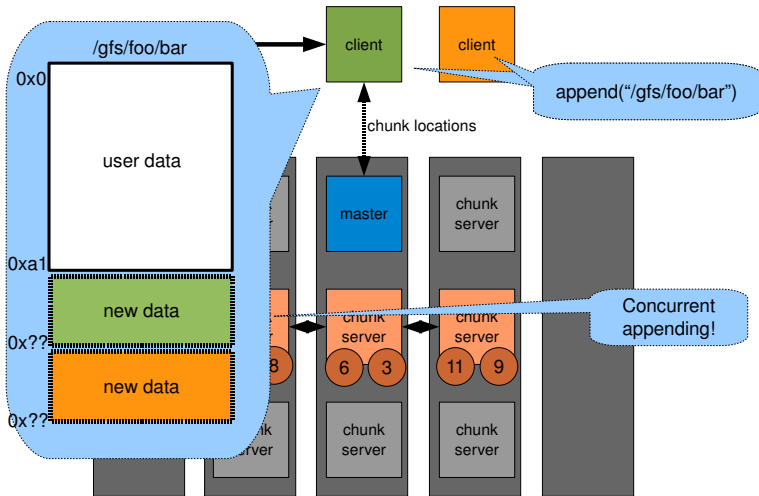
# Client API: Record Appends



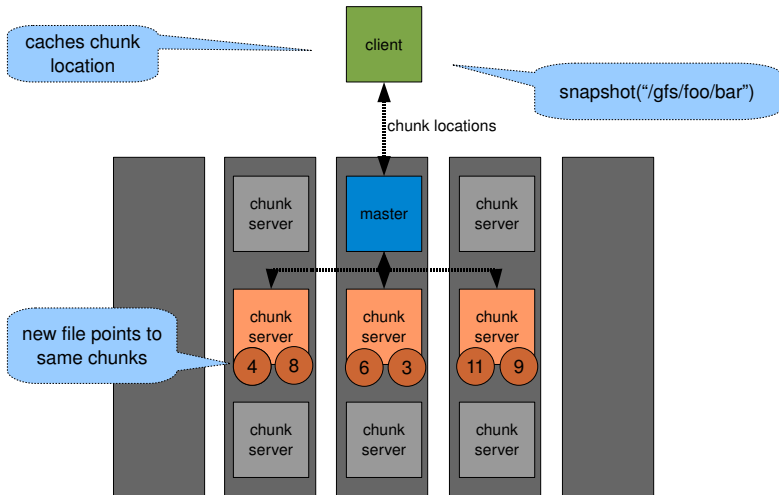
## Client API: Record Appends



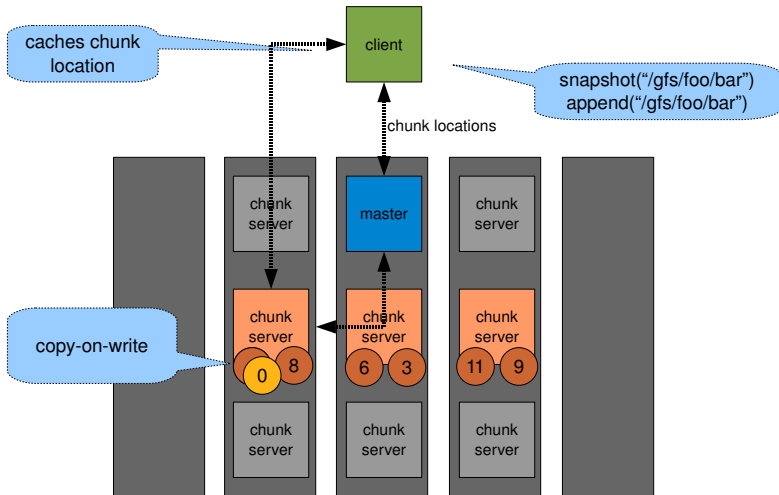
# Client API: Record Appends



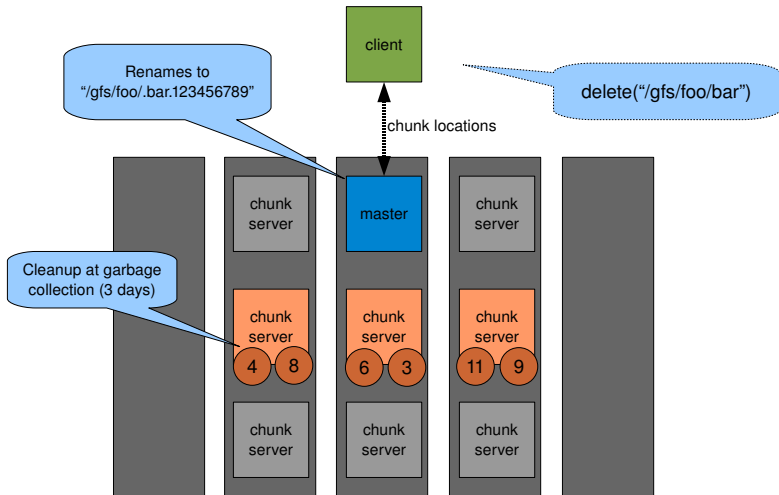
# Client API: Snapshot



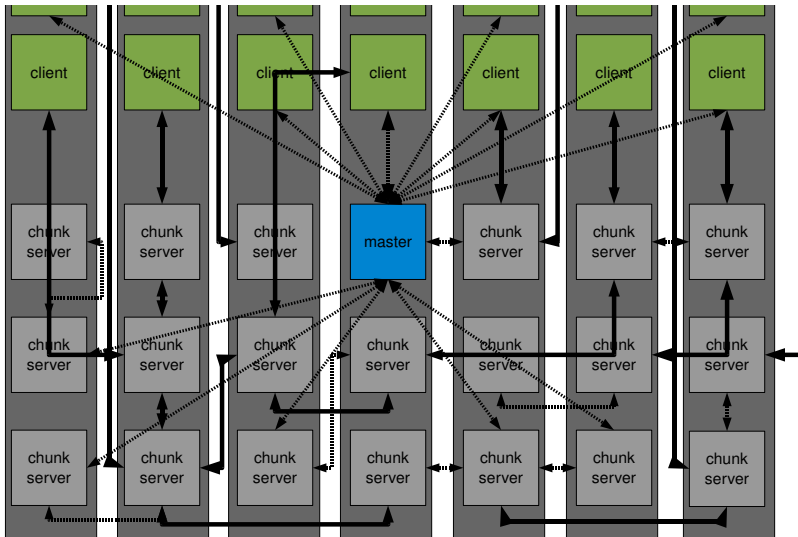
# Client API: Snapshot



## Client API: Deletes



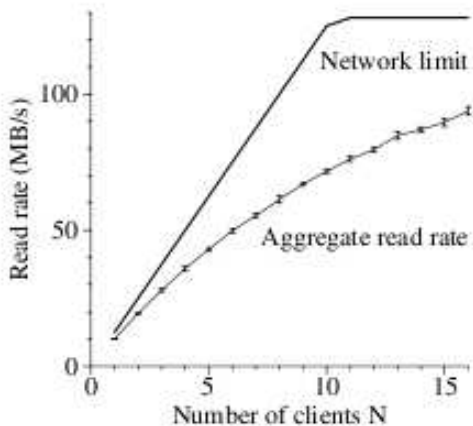
# Real World Clusters



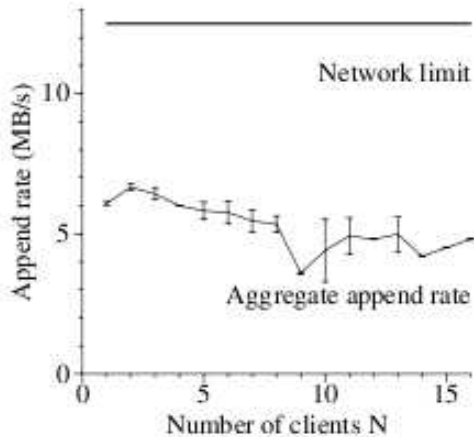
# Performance Test

- Dual 1.4 GHZ PIII, 2G memory, 2x 80GB 5400rpm disks
- One master, with two replicas
- 16 chunkservers
- 16 clients

# Read Performance



# Append Performance



# Latest Work on GFS

- GFS now supports at least tens of petabytes.
- Large filecount filled up master memory.
  - Solution: Filecount quotas.
  - Solution: clients use multiple GFS clusters for many files.
  - Long-Term Solution: multiple masters.

# Observations

- No clear description of real clients.
- How does google use multiple clusters?
- I want to work at google! :D

# Conclusion

- GFS is a single-master distributed large-scale filesystem.
- Designed for large cheap clusters.
- Assumes failure at any time.
- Used now by Google services.

# Questions

Questions + Answers.