CS 744: BIG DATA SYSTEMS

Shivaram Venkataraman

Fall 2018

HISTORY OF DISTRIBUTED FILE SYSTEMS

SUN NFS

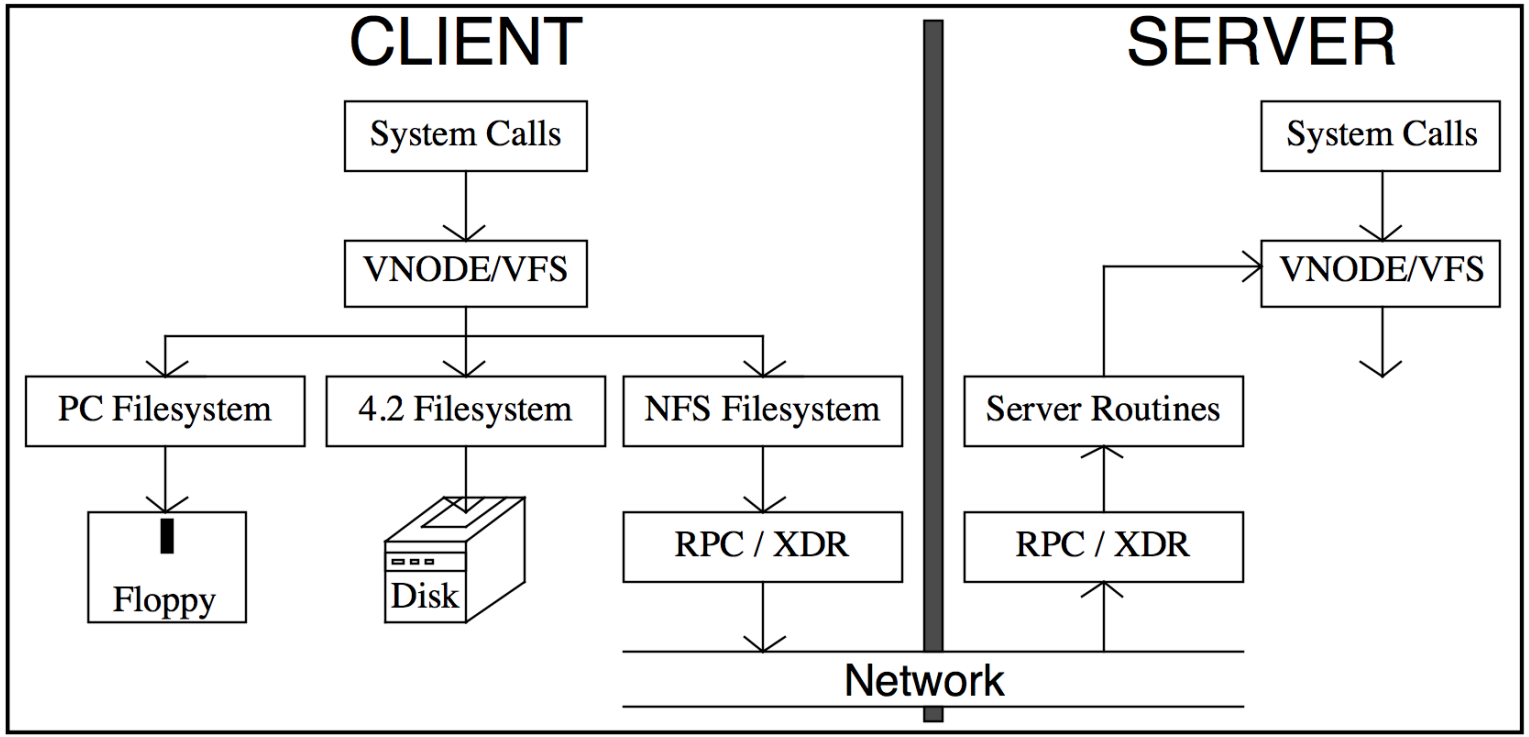


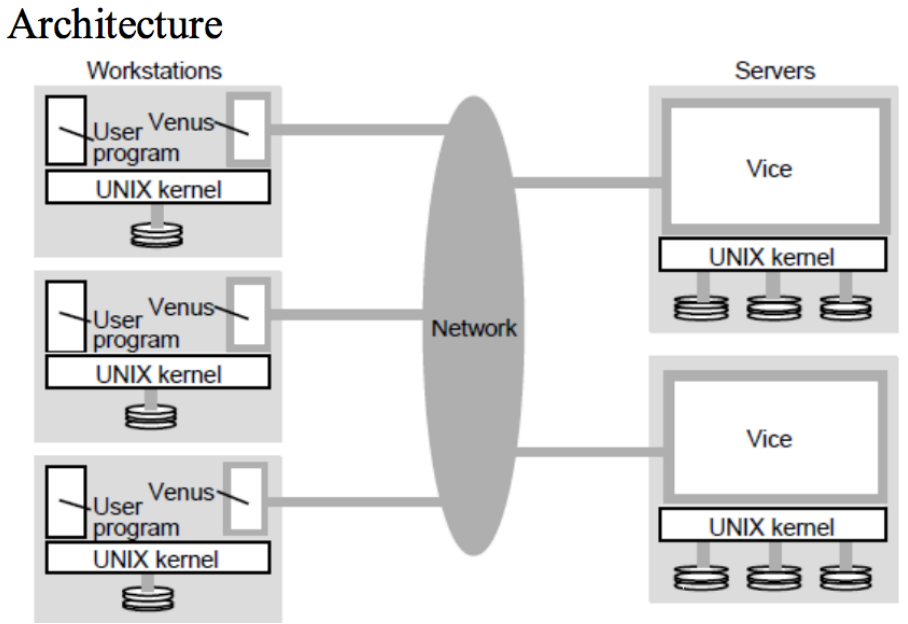
Figure 1

ANDREW FILE SYSTEM

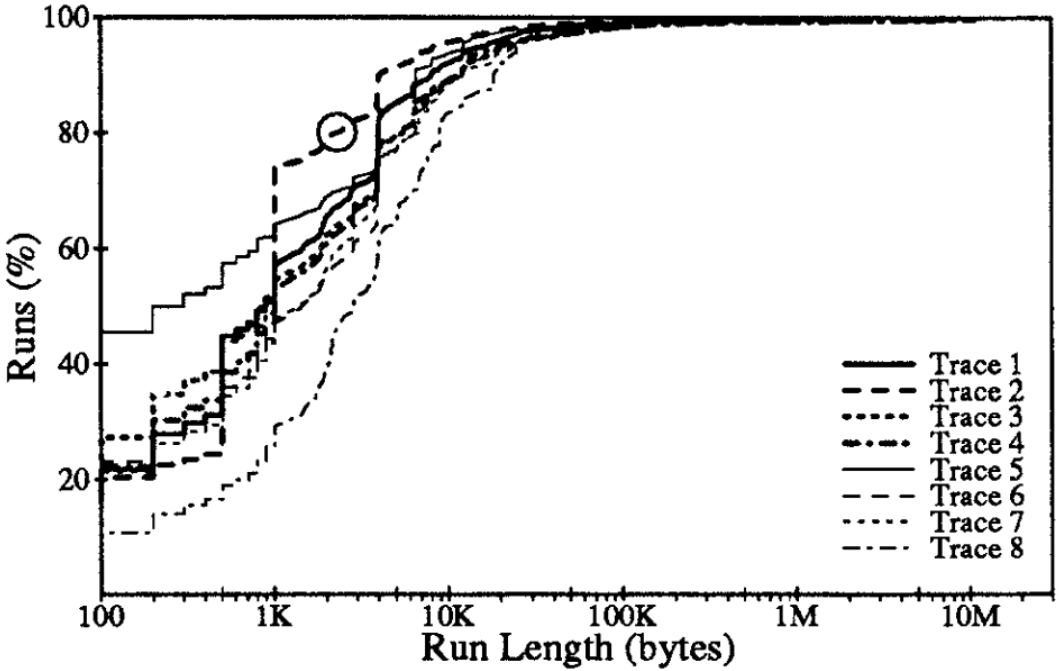
- Design for scale

- Whole-file caching

- Callbacks from server



WORKLOAD PATTERNS (1991)



Mary G. Baker, John H. Hartman, Michael D. Kupfer, Ken W. Shirriff, and John K. Ousterhout

WORKLOAD PATTERNS (1991)

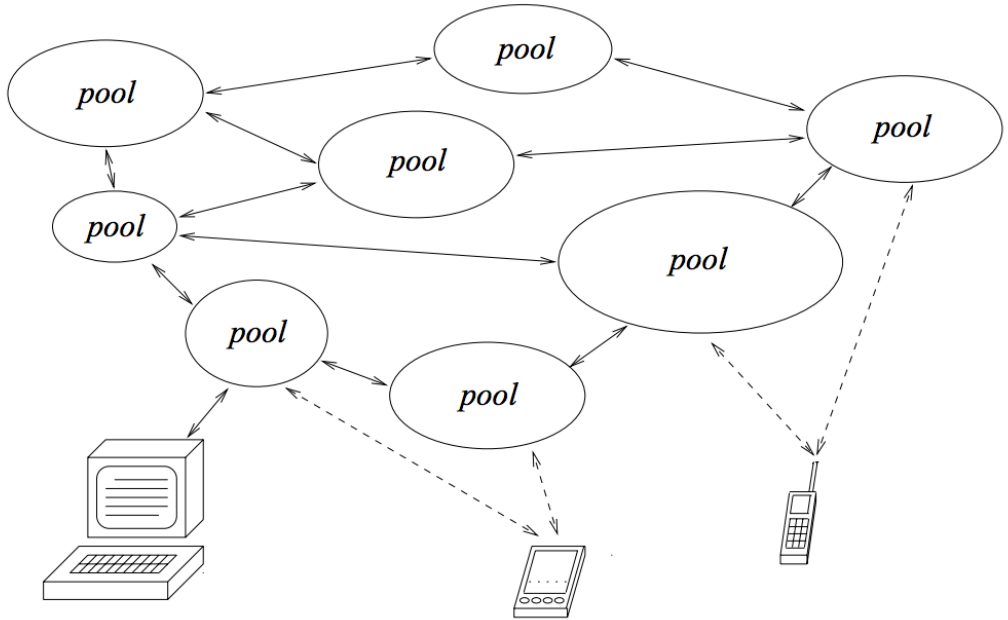
File Usage	Type of Transfer	Accesses (%)		Bytes (%)	
Read-only	Whole-file	78	(64-91)	89	(46-96)
	Other sequential	19	(7-33)	5	(2-29)
	Random	3	(1-5)	7	(2-37)
Write-only	Whole-file	67	(50-79)	69	(56-76)
	Other sequential	29	(18-47)	19	(4-27)
	Random	4	(2-8)	11	(4-41)
Read/write	Whole-file	0	(0-0)	0	(0-0)
	Other sequential	0	(0-0)	0	(0-0)
	Random	100	(100-100)	100	(100-100)

OCEANSTORE/PAST

Wide area storage systems

Fully decentralized

Built on distributed hash tables (DHT)



Components with failures

Files are huge !

GFS: WHY ?

Applications are different

GFS: WORKLOAD ASSUMPTIONS

Two kinds of reads: Large Streaming and small random

Writes: Many large, sequential writes. No random

High bandwidth more important than low latency

GFS: WHAT ?

- Single Master for metadata

- Chunkservers for storing data

- No POSIX API !
- No Caches!

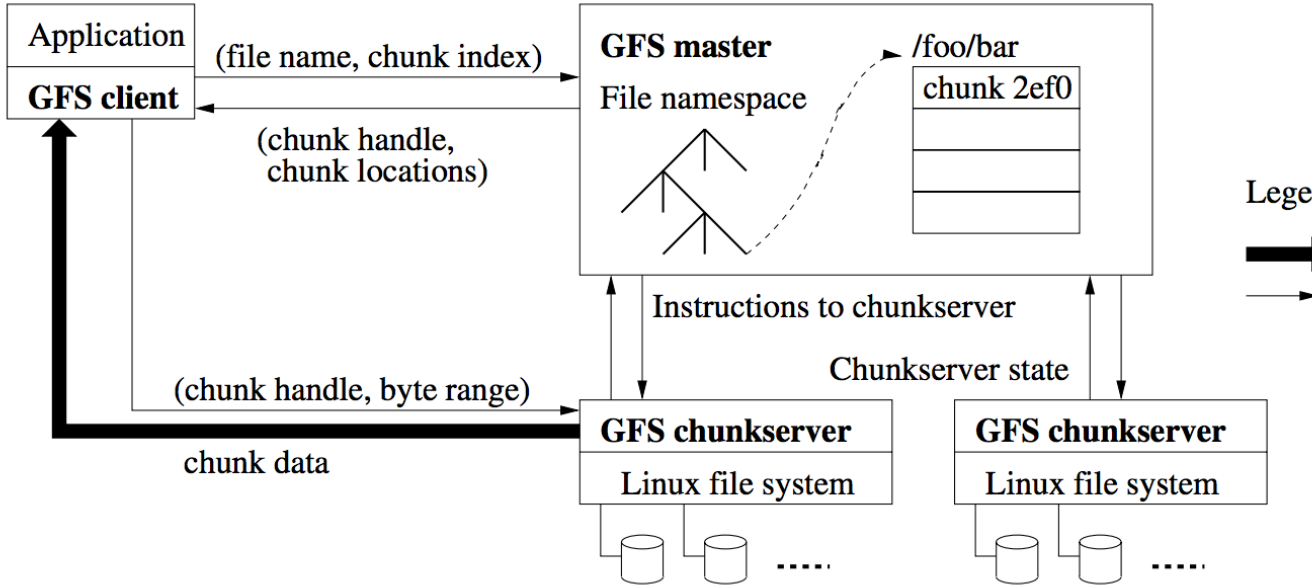
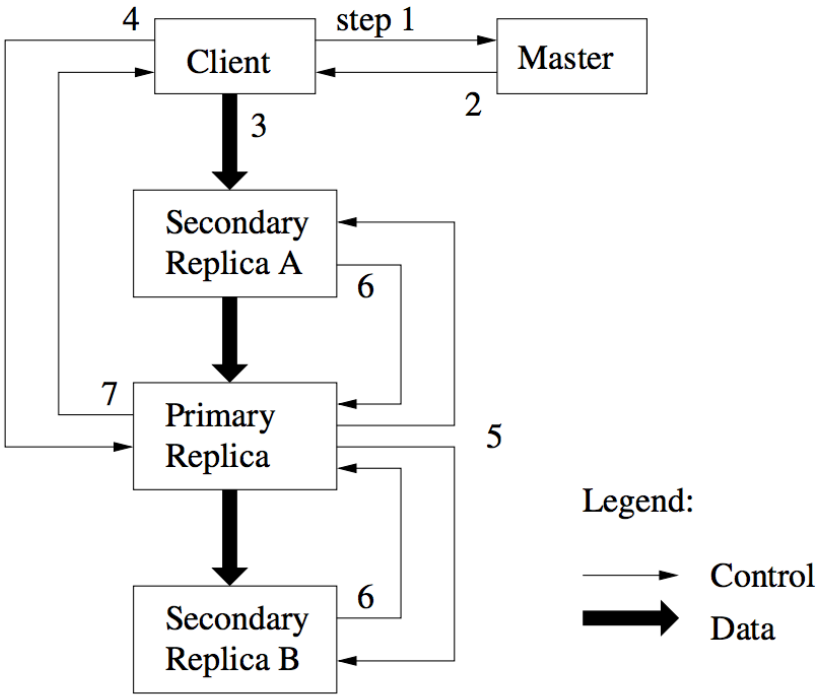


Figure 1: GFS Architecture

GFS: WHAT ?



- Replication to handle faults

- Primary replica for each chunk

- Chain replication (consistency)

WHAT HAPPENED NEXT

Cluster-Level Storage @ Google

How we use *Colossus* to improve storage efficiency

Denis Serenyi
Senior Staff Software Engineer
dserenyi@google.com

Keynote at PDSW-DISCS 2017: 2nd Joint International Workshop On Parallel Data Storage & Data Intensive Scalable Computing Systems

GFS EVOLUTION

Motivation:

- GFS Master
  - One machine not large enough for large FS
  - Single bottleneck for metadata operations (data path offloaded)
  - Fault tolerant, but not HA

- Lack of predictable performance
  - No guarantees of latency
  - (GFS problems: one slow chunkserver -> slow writes)

GFS EVOLUTION

GFS master replaced by Colossus

Metadata stored in BigTable [next class !]

Recursive structure ? If Metadata is ~1/10000 the size of data

100 PB data → 10 TB metadata

10TB metadata → 1GB metametadata

1GB metametadata → 100KB meta...

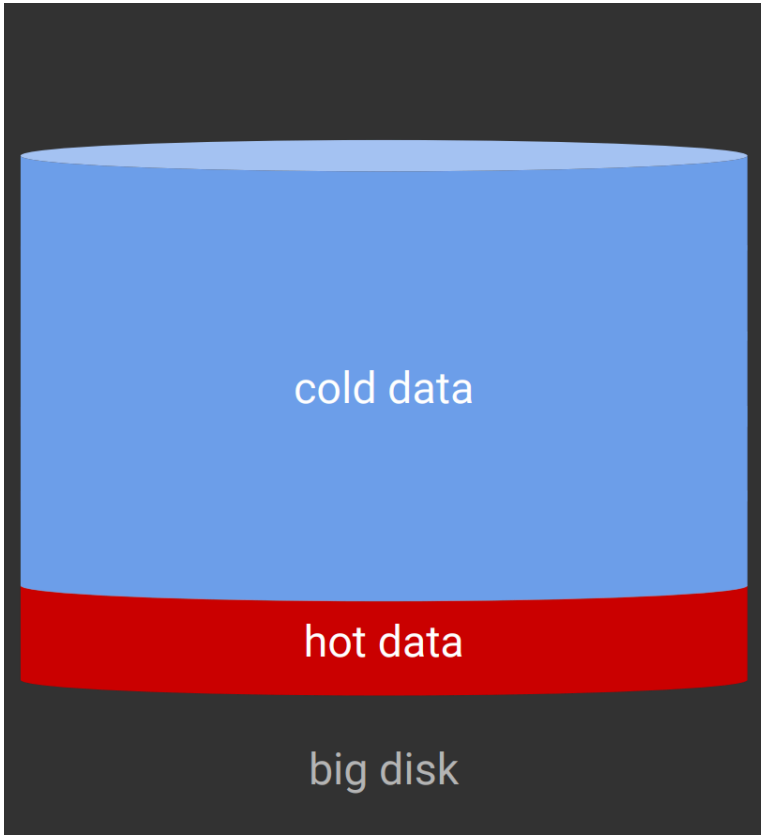
GFS EVOLUTION

Need for Efficient Storage

Rebalance old, cold data

Distributes newly written data evenly across disk

Manage both SSD and hard disks



HETEROGENEOUS STORAGE

F4: Facebook (This class !)

Blob stores

Key Value Stores