Chaos of Architecture

jingmi@gmail.com

2012-01-13

Why we need better architecture

- Ticket System

Google & Facebook

- Goal
- Bottleneck

Facebook - Capacity

- http://www.slideshare.net/mysqlops/facebook-architecture

facebook **At a Glance**

The Social Graph

120M+ active users

50B+ PVs per month

10B+ Photos

1B+ connections

50K+ Platform Apps

400K+ App Developers

General Design Principles

- Use open source where possible
  - Explore making optimizations where needed
- Unix Philosophy
  - Keep individual components simple yet performant
  - Combine as necessary
  - Concentrate on clean interface points
- Build everything for scale
- Try to minimize failure points
- Simplicity, Simplicity, Simplicity!

Architecture Overview

memcached

LAMP

PHP

Memcache

MySQL

php!

\+

Services

AdServer
Search
Network Selector
News Feed
Blogfeeds
CSSParser
Mobile
ShareScraper

!php

Thrift
Scribe
ODS
Tools

PHP: What we Learnt

- Tough to scale for large code bases
  - Weak typing
  - Limited opportunities for static analysis, code optimizations
- Not necessarily optimized for large website use case
  - E.g. No dynamic reloading of files on web server
- Linearly increasing cost per included file
- Extension framework is difficult to use

PHP: Customizations

- Op-code optimization
- APC improvements
  - Lazy loading
  - Cache priming
  - More efficient locking semantics for variable cache data
- Custom extensions
  - Memcache client extension
  - Serialization format
  - Logging, Stats collection, Monitoring
  - Asynchronous event-handling mechanism

HipHop

- HipHop for PHP transforms PHP source code into highly optimized C++. It was developed by Facebook and was released as open source in early 2010.
- Facebook sees about a 50% reduction in CPU usage when serving equal amounts of Web traffic when compared to Apache and PHP. Facebook'sAPI tier can serve twice the traffic using 30% less CPU.
- https://developers.facebook.com/blog/post/358
- http://wiki.github.com/facebook/hiphop-php/

PHPEmbed

- PHPEmbed enables developers writing applications in C++ and other languages to easily embed PHP in their code, making development times faster as developers don't have to abandon existing development patterns to get the job done. Facebook wrote this to speed up their development processes and has released the code to the public.
- https://github.com/facebook/phpembed

Varnish

- http://varnish-cache.org/
- https://www.varnish-software.com/references/social-media/facebook
- **Varnish** serves billions of requests every day to Facebook users around the world. Whenever you load photos and profile pictures of your friends, there's a very good chance that Varnish is involved.

MySQL

- Fast, reliable
- Used primarily as ‹key,value› store
  - Data randomly distributed amongst large set of logical instances
  - Most data access based on global id
- Large number of logical instances spread out across physical nodes
  - Load balancing at physical node level
- No read replication

MySQL: What We Learnt (ing)

- Logical migration of data is *very* difficult
- Create a large number of logical dbs, load balance them over varying number of physical nodes
- No joins in production
  - Logically difficult (because data is distributed randomly)
- Easier to scale CPU on web tier

How To JOIN?

- NOSQL - Bigtable – Leveldb
- NOSQL - MongoDB
- Access Model & Distributed Model

MySQL: What we Learnt (ing)

- Most data access is for recent data
  - Optimize table layout for recency
  - Archive older data
- Don't ever store non-static data in a central db
  - CDB makes it easier to perform certain aggregated queries
  - Will not scale
- Use services or memcache for global queries
  - E.g.: What are the most popular groups in my network

MySQL: Customizations

- No extensive native MySQL modifications
- Custom partitioning scheme
  - Global id assigned to all data
- Custom archiving scheme
  - Based on frequency and recency of data on a per-user basis
- Extended Query Engine for cross-data center replication, cache consistency

MySQL: Customizations

- Graph based data-access libraries
  - Loosely typed objects (nodes) with limited datatypes (int, varchar, text)
  - Replicated connections (edges)
  - Analogous to distributed foreign keys
- Some data collocated
  - Example: User profile data and all of user's connections
- Most data distributed randomly

mysqlatfacebook

- **Online Schema Change for MySQL** lets you alter large database tables without taking your cluster offline.
- https://www.facebook.com/notes/mysql-at-facebook/online-schema-change-for-mysql/430801045932

Flashcache

- Flashcache is a simple write back persistent block cache designed to accelerate reads and writes from slower rotational media by caching data in SSD's.
- https://www.facebook.com/note.php?note_id=388112370932
- https://github.com/facebook/flashcache/tree/

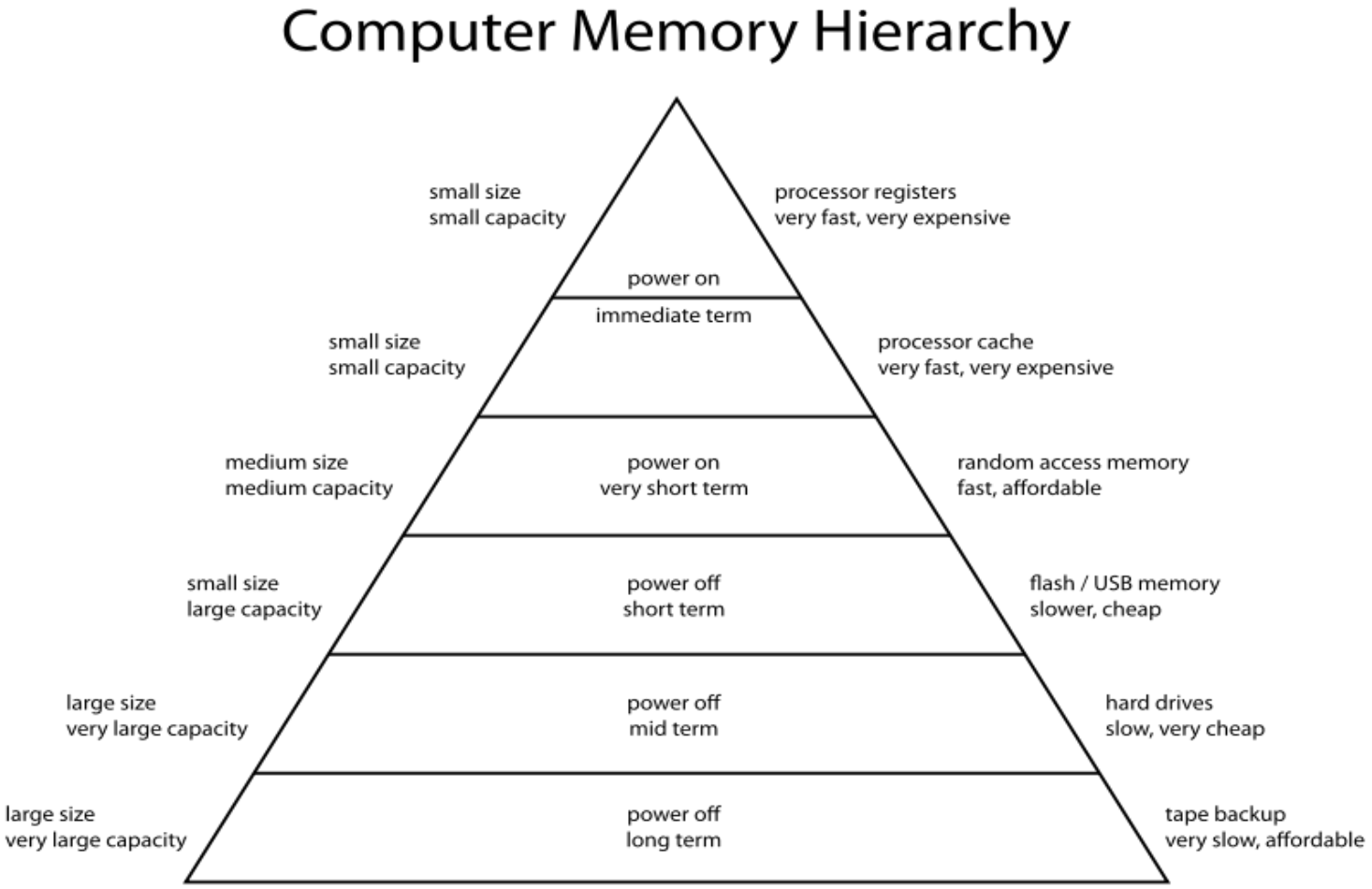
Hardware - Storage

- http://blog.yufeng.info/archives/1991

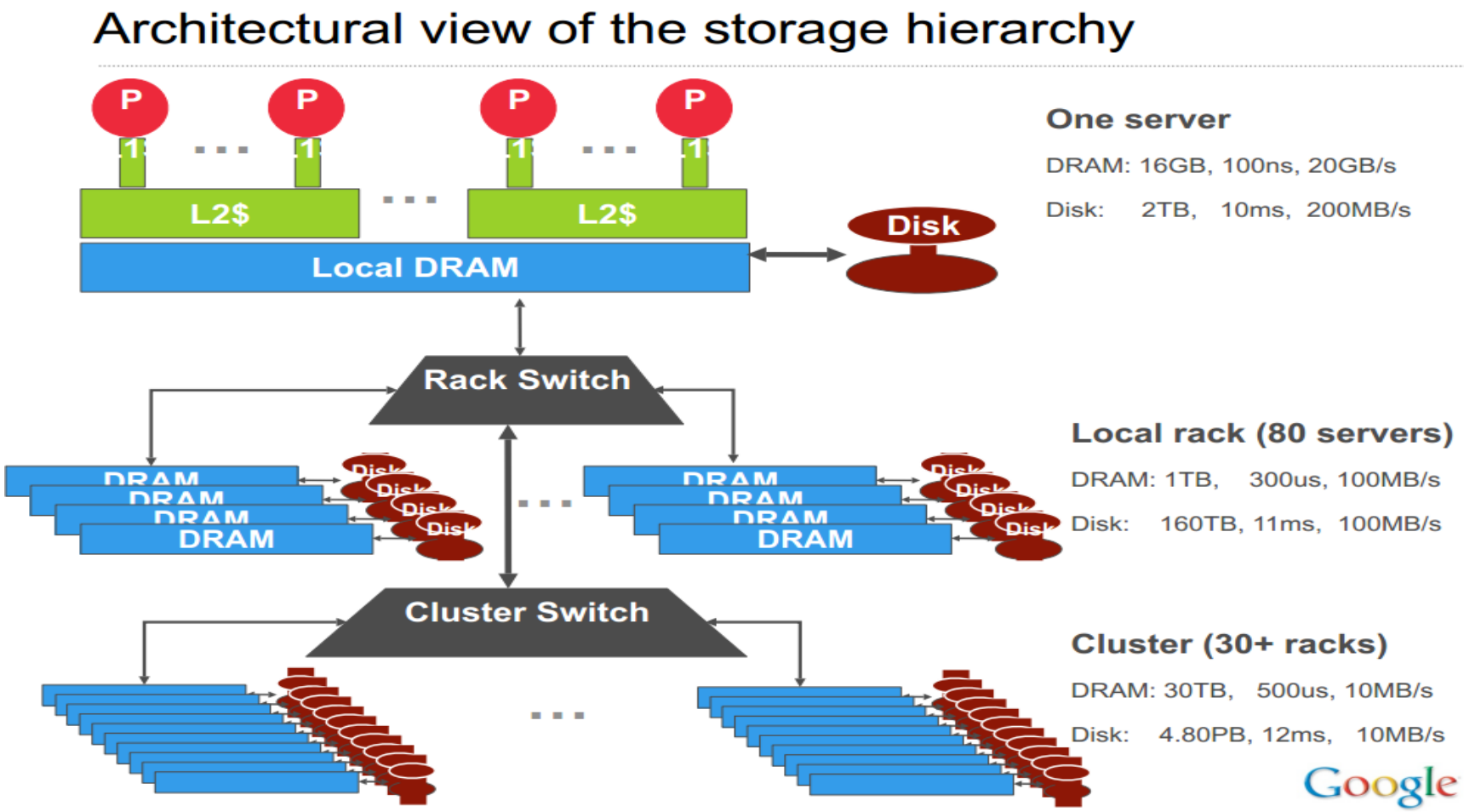
存储设备 IOPS 演变史

设备	IOPS	接口
7200 RPM SATA drives	~90 IOPS	SATA II
15k RPM SCSI drives	~180 IOPS	SAS
Intel X25-M G2 (MLC)	~8,600 IOPS	SATA II
ioDrive, a PCI-Express card with Flash	with Flash 140,000 Read IOPS, 135,000 Write IOPS	PCIe
Fusion-io ioDrive Octal	1,180,000+ Random Read/Write IOPS	PCIe

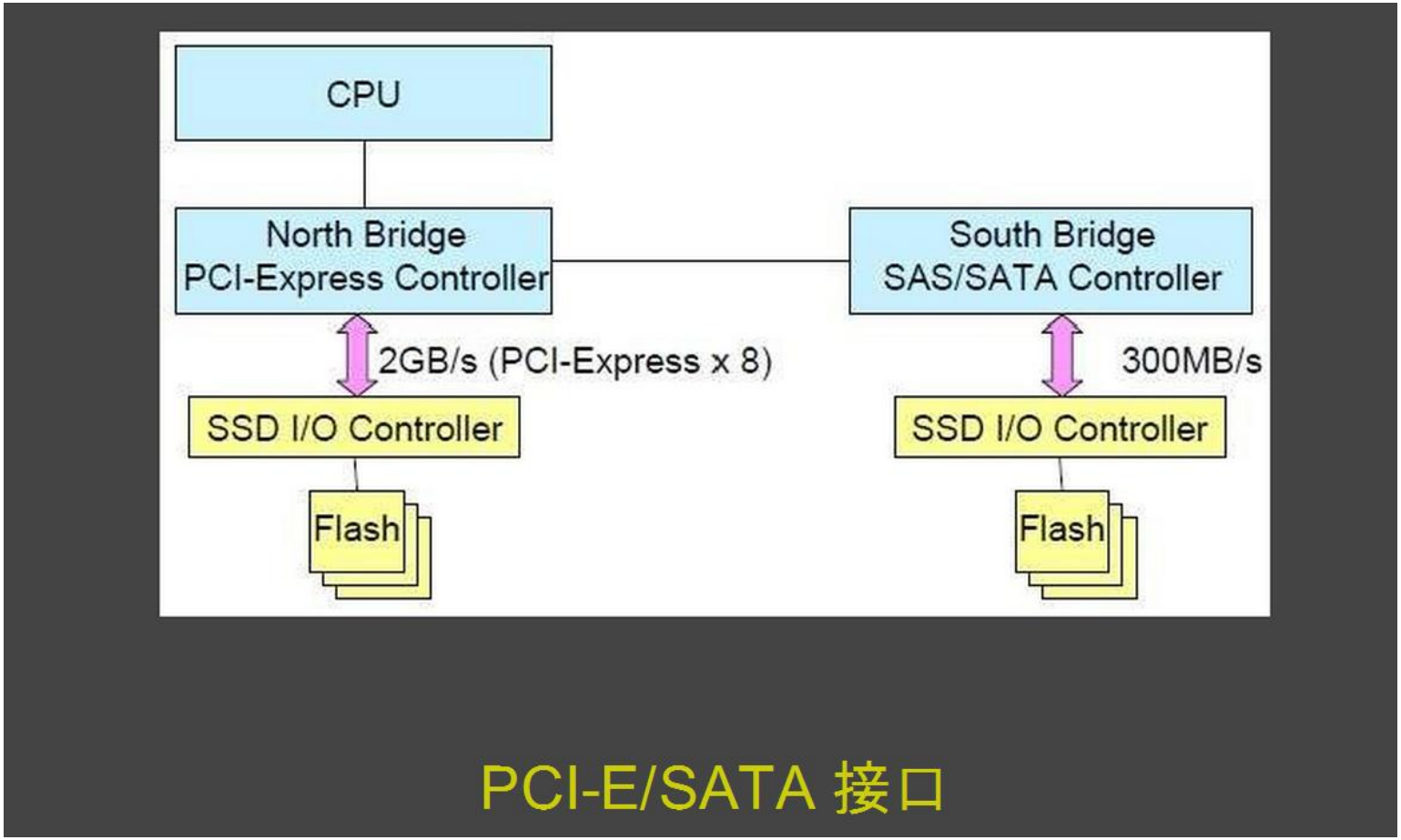
Storage Hierarchy



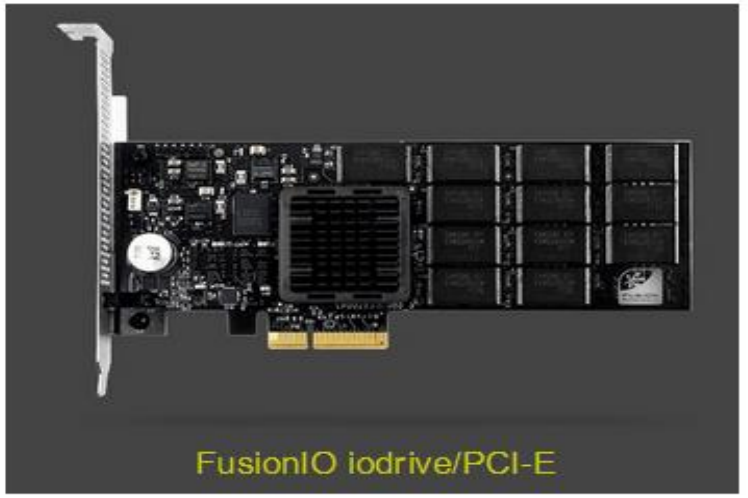
Storage Hierarchy(2)



PCI-E VS SATA


CPU
North Bridge
PCI-Express Controller
South Bridge
SAS/SATA Controller
2GB/s (PCI-Express x 8)
300MB/s
SSD I/O Controller
SSD I/O Controller
Flash
Flash
PCI-E/SATA 接口

Intel X25-M/SATA



FusionIO iodrive/PCI-E

Virident tachion/PCI-E

memcached

Memcache

- High-Performance, distributed in-memory hash table
- Used to alleviate database load
- Primary form of caching
- Over 25TB of in-memory cache
- Average latency < 200 micro-seconds
- Cache serialized PHP data structures
- Lots and lots of multi-gets to retrieve data spanning across graph edges

Memcache: Customizations

- https://www.facebook.com/note.php?note_id=39391378919

memcached

Memache: Customizations

- Memache over UDP
  - Reduce memory overhead of thousands of TCP connection buffers
  - Application-level flow control (optimization for multi-gets)
- On demand aggregation of per-thread stats
  - Reduces global lock contention
- Multiple Kernel changes to optimize for Memcache usage
  - Distributing network interrupt handling over multiple cores
  - Opportunistic polling of network interface

Consistency

- https://www.facebook.com/note.php?note_id=23844338919&id=9445547199&index=0

- I update my first name from "Jason" to "Monkey"
- We write "Monkey" in to the master database in California and delete my first name from memcache in California and Virginia
- Someone goes to my profile in Virginia
- We don't find my first name in memcache so we read from the Virginia slave database and get "Jason" because of replication lag
- We update Virginia memcache with my first name as "Jason"
- Replication catches up and we update the slave database with my first name as "Monkey"
- Someone else goes to my profile in Virginia
- We find my first name in memcache and return "Jason"

- Solutions?

Consistency

- I update my first name from "Jason" to "Monkey"
- We write "Monkey" in to the master database in California and delete my first name from memcache in California **but not Virginia**
- Someone goes to my profile in Virginia
- **We find my first name in memcache and return "Jason"**
- Replication catches up and we update the slave database with my first name as "Monkey." **We also delete my first name from Virginia memcache because that cache object showed up in the replication stream**
- Someone else goes to my profile in Virginia
- **We don't find my first name in memcache so we read from the slave and get "Monkey"**

Under the Covers

- Get my profile data
  - Fetch from cache, potentially go to my DB (based on user-id)
- Get friend connections
  - Cache, if not DB (based on user-id)
- In parallel, fetch last 10 photo album ids for each of my friends
  - Multi-get; individual cache misses fetches data from db (based on photo-album id)
- Fetch data for most recent photo albums in parallel
- Execute page-specific rendering logic in PHP
- Return data, make user happy

LAMP is not Perfect

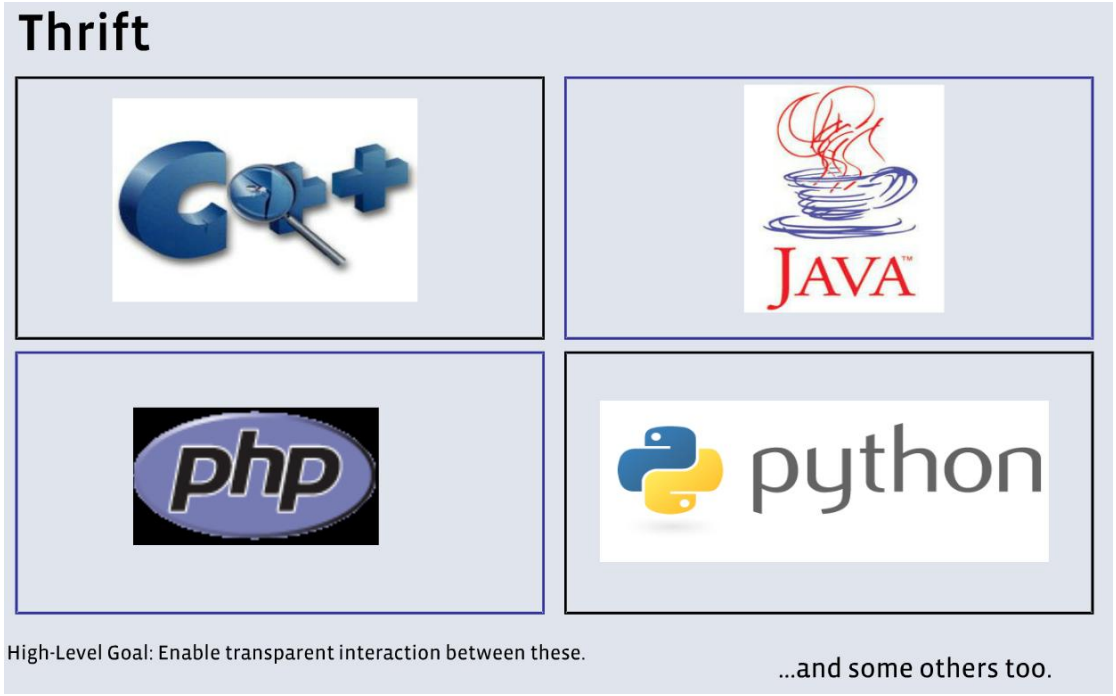
- PHP+MySQL+Memcache works for a large class of problems but not for everything
  - PHP is stateless
  - PHP not the fastest executing language
  - All data is remote
- Reasons why services are written
  - Store code closer to data
  - Compiled environment is more efficient
  - Certain functionality only present in other languages

Services Philosophy

- Create a service iff required
  - Real overhead for deployment, maintenance, separate code-base
  - Another failure point
- Create a common framework and toolset that will allow for easier creation of services
  - Thrift
  - Scribe
  - ODS, Alerting service, Monitoring service
- Use the right language, library and tool for the task

Thrift

- http://thrift.apache.org/



Thrift

- Lightweight software framework for cross-language development
- Provide IDL, statically generate code
- Supported bindings: C++, PHP, Python, Java, Ruby, Erlang, Perl, Haskell etc.
- Transports: Simple Interface to I/O
  - Tsocket, TFileTransport, TMemoryBuffer
- Protocols: Serialization Format
  - TBinaryProtocol, TJSONProtocol
- Servers
  - Non-Blocking, Async, Single Threaded, Multi-threaded

Thrift: Why?

- It's quick. Really quick.
- Less time wasted by individual developers
  - No duplicated networking and protocol code
  - Less time dealing with boilerplate stuff
  - Write your client and server in about 5 minutes
- Division of labor
  - Work on high-performance servers separate from applications
- Common toolkit
  - Fosters code reuse and shared tools

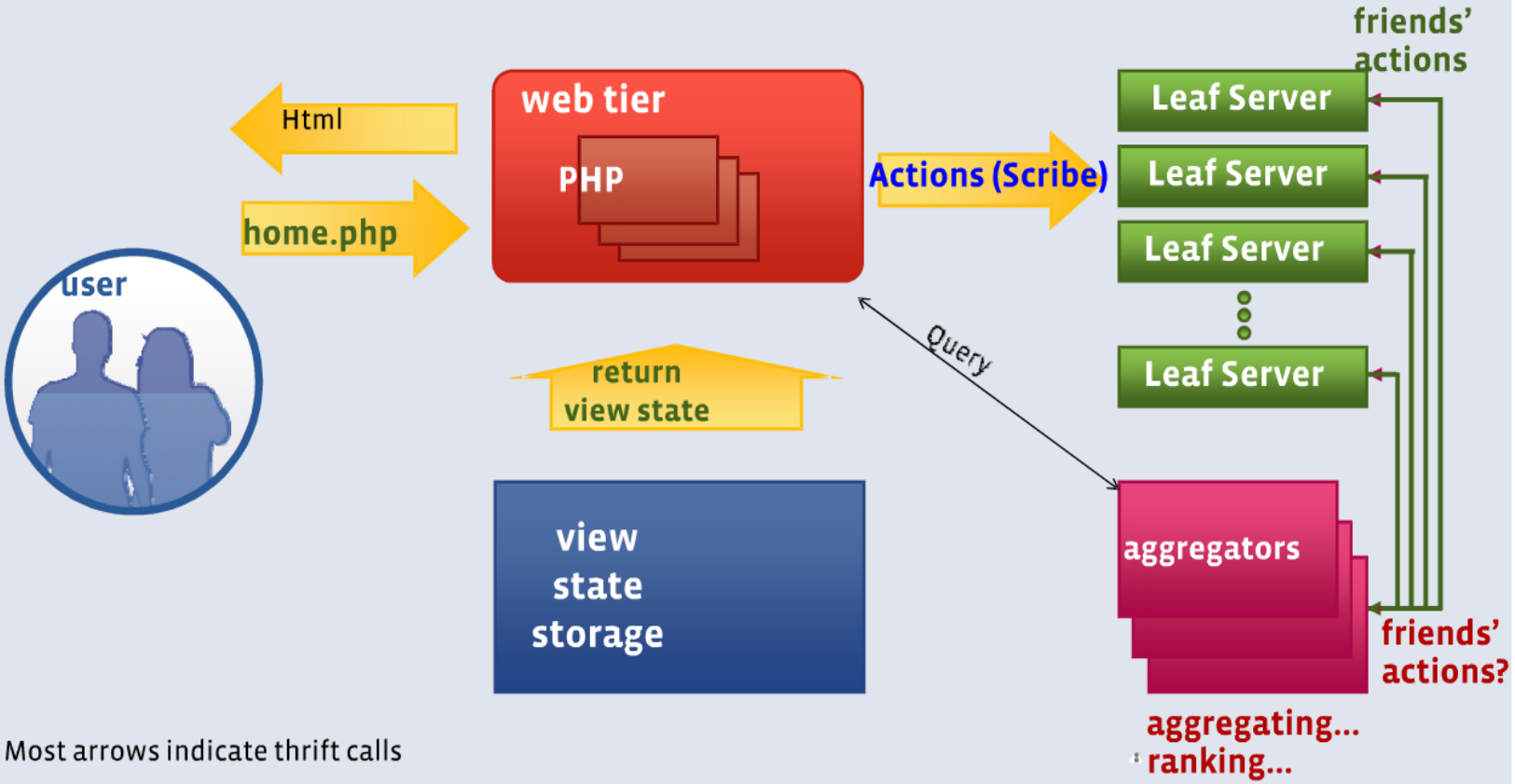
Scribe

- https://github.com/facebook/scribe

Scribe

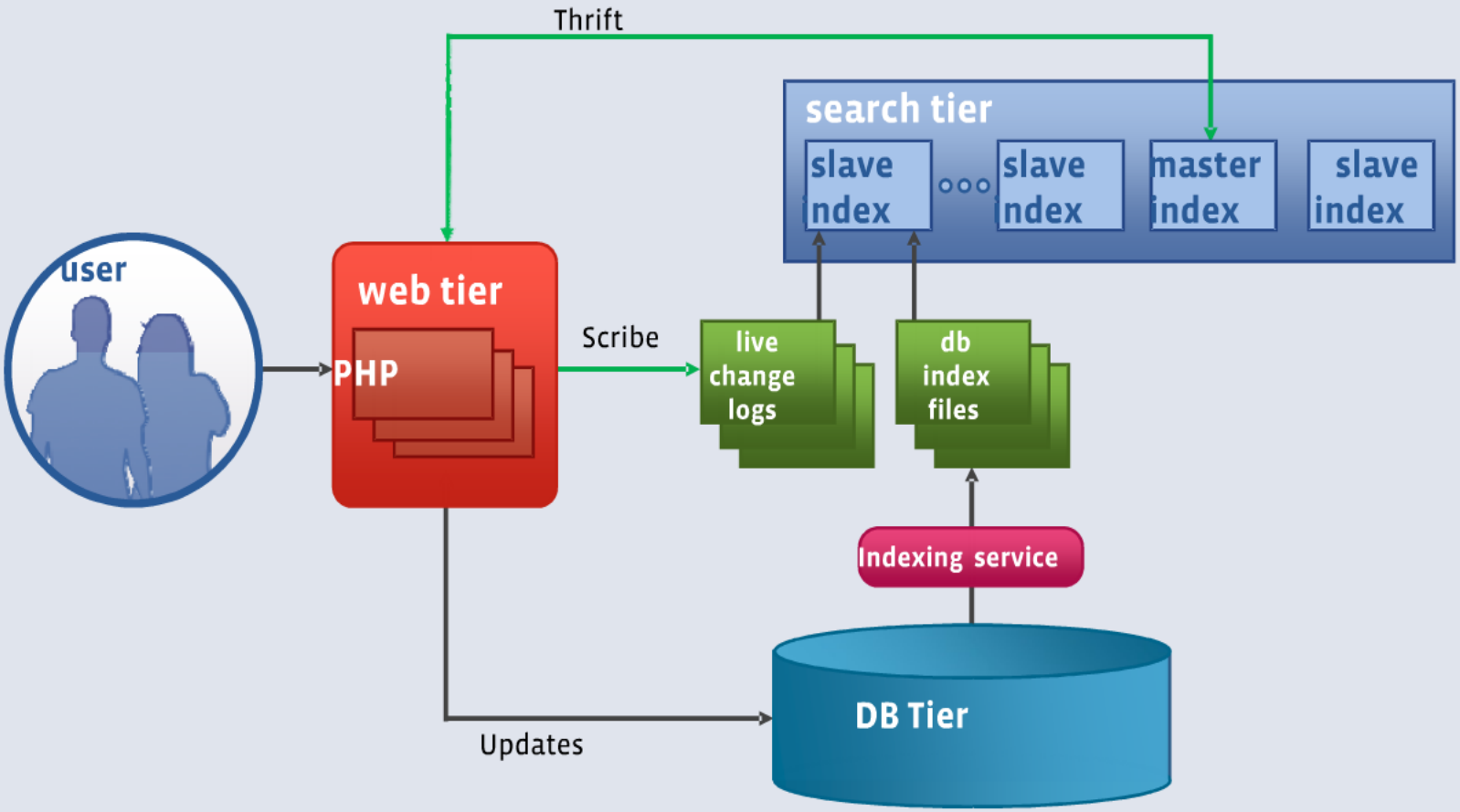
- Scalable distributed logging framework
- Useful for logging a wide array of data
  - Search Redologs
  - Powers news feed publishing
  - A/B testing data
- Weak Reliability
  - More reliable than traditional logging but not suitable for database transactions.
- Simple data model
- Built on top of Thrift

NewsFeed – The Work



- Most arrows indicate thrift calls

Search – The Work



Projects in facebook

- **phpsh** provides an interactive shell for PHP that features readline history, tab completion, and quick access to documentation. It is ironically written mostly in Python.
- **Three20** is an Objective-C library for iPhone developers which provides many UI elements and data helpers behind our iPhone application.
- **Facebook Animation** is a JavaScript library for creating customizable animations using DOM and CSS manipulation.
- **Tornado** is a relatively simple, non-blocking web server framework written in Python. It is designed to handle thousands of simultaneous connections, making it ideal for real-time Web services.
- **jemalloc** is a memory allocator which is fast, consistant, and supports heap profiling. Facebook engineers added heap profiling and made many optimizations.
- people.freebsd.org/~jasone/**jemalloc**/bsdcan2006/**jemalloc**.pdf

Platform

- http://opencompute.org/
- We started a project at Facebook a little over a year ago with a pretty big goal: to build one of the most efficient computing infrastructures at the lowest possible cost. We decided to honor our hacker roots and challenge convention by custom designing and building our software, servers and data centers from the ground up – and then share these technologies as they evolve.
- The result is a data center full of vanity free servers which is 38% more efficient and 24% less expensive to build and run than other state-of-the-art data centers.

NOSQL in Facebook

- **Apache Cassandra** is a distributed storage system for managing structured data that is designed to scale to a very large size across many commodity servers, with no single point of failure.
- **Apache Hive** is data warehouse infrastructure built on top of Hadoop that provides tools to enable easy data summarization, adhoc querying and analysis of large datasets.
- **Apache Hadoop** provides reliable, scalable, distributed computing infrastructure which we use for data analysis.
- **Apache HBase** is a distributed, versioned, column-oriented data store built on top of the Hadoop Distributed Filesystem.

Others

- **Scaling the Messages Application Back End**
- https://www.facebook.com/note.php?note_id=10150148835363920

- **Typeahead Search**
- https://www.facebook.com/video/video.php?v=432864835468

What is Facebook's architecture?

- http://www.quora.com/Micha%C3%ABl-Figui%C3%A8re/answers/Facebook-Engineering

- **Scaling Facebook to 500 Million Users and Beyond** https://www.facebook.com/note.php?note_id=409881258919

Image Service in Facebook

- http://www.usenix.org/event/osdi10/tech/full_papers/Beaver.pdf

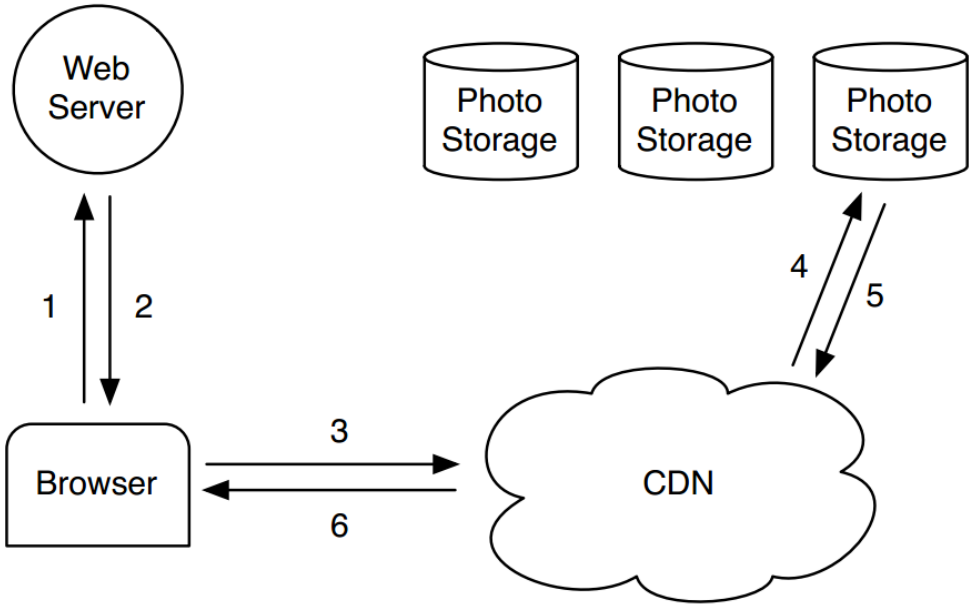


Figure 1: Typical Design

NFS Based Storage

- Bottleneck?

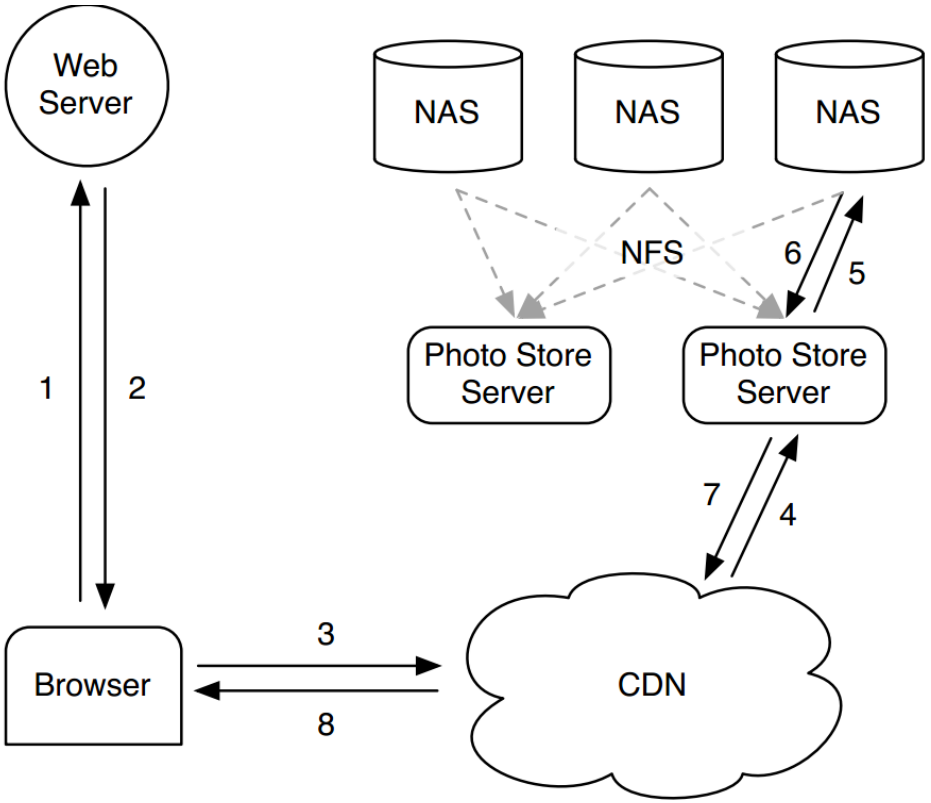


Figure 2: NFS-based Design

Haystack

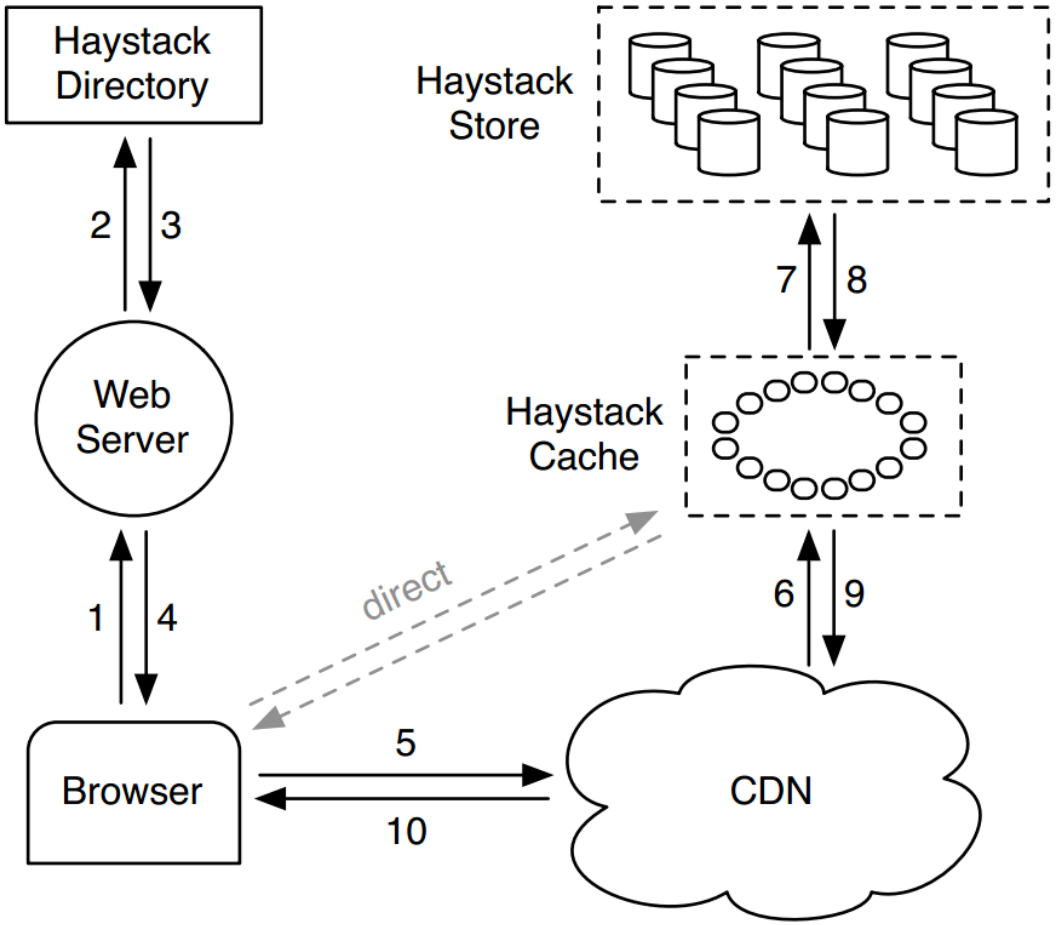


Figure 3: Serving a photo

Uploading Photos

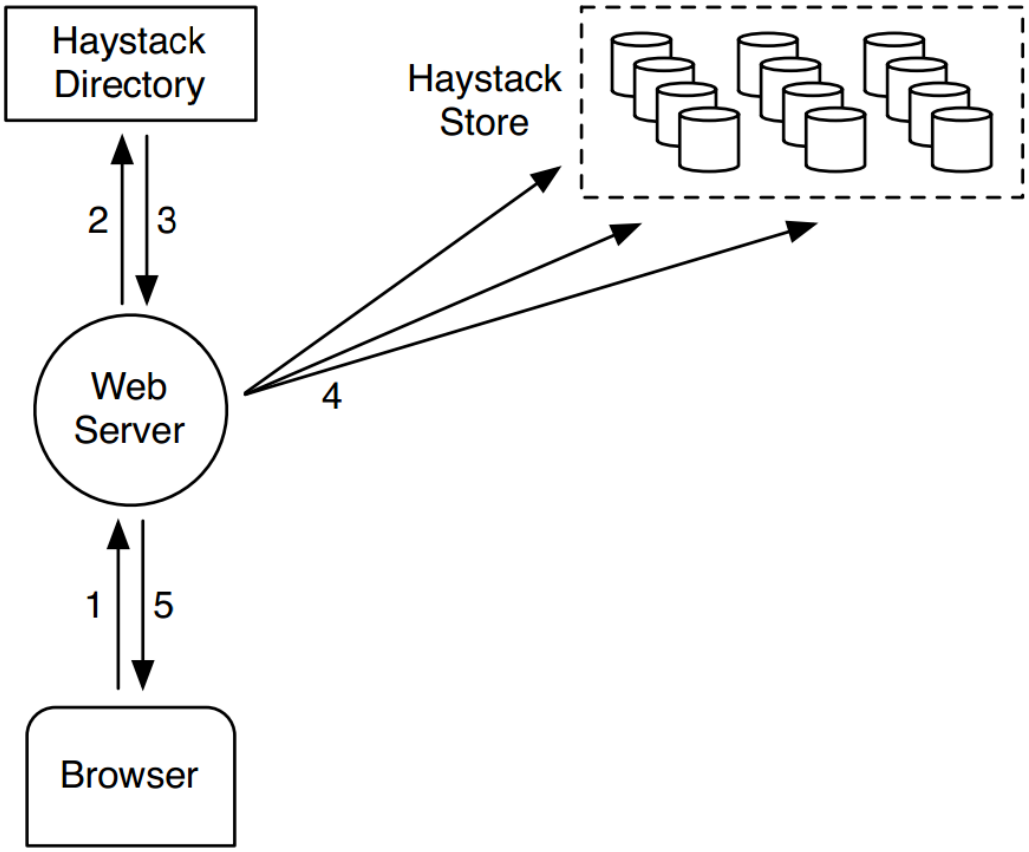


Figure 4: Uploading a photo

Operations	**Daily Counts**
Photos Uploaded	~120 Million
Haystack Photos Written	~1.44 Billion
Photos Viewed	80-100 Billion
[*Thumbnails*]	10.2 %
[*Small*]	84.4 %
[*Medium*]	0.2 %
[*Large*]	5.2 %
Haystack Photos Read	10 Billion

Table 3: Volume of daily photo traffic.

Photos @ Facebook

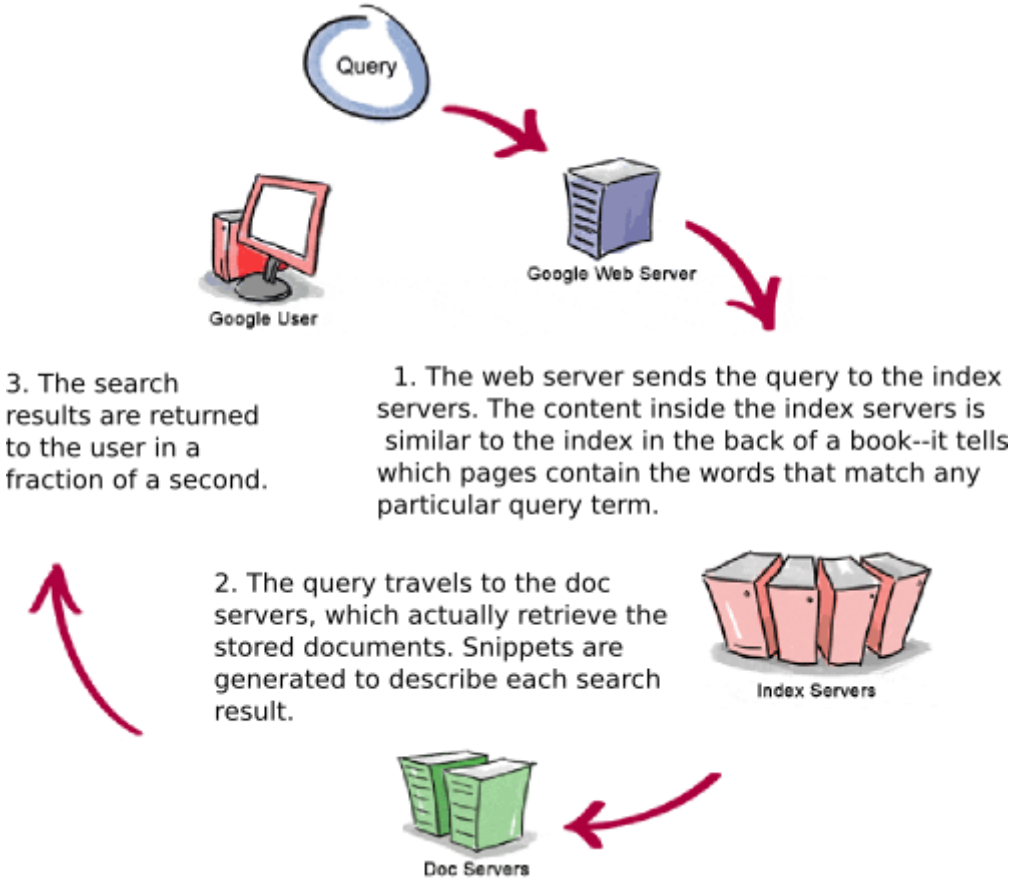
	April 2009	Current
Total	15 billion photos 60 billion images 1.5 petabytes	65 billion photos 260 billion images 20 petabytes
Upload Rate	220 million photos / week 25 terabytes	1 billion photos / week 60 terabytes
Serving Rate	550,000 images / sec	1 million images / sec

Conclusion

- Haystack – simple and effective storage system
  - Optimized for random reads (~1 I/O per object read)
  - Cheap commodity storage
  - 8,500 LOC (C++)
  - 2 engineers 4 months from inception to initial deployment
- Future work
  - Software RAID6
  - Limit dependency on external CDN
  - Index on flash

How Google Works?

- http://www.googleguide.com/google_works.html



WEB SEARCH FOR A PLANET: THE GOOGLE CLUSTER ARCHITECTURE

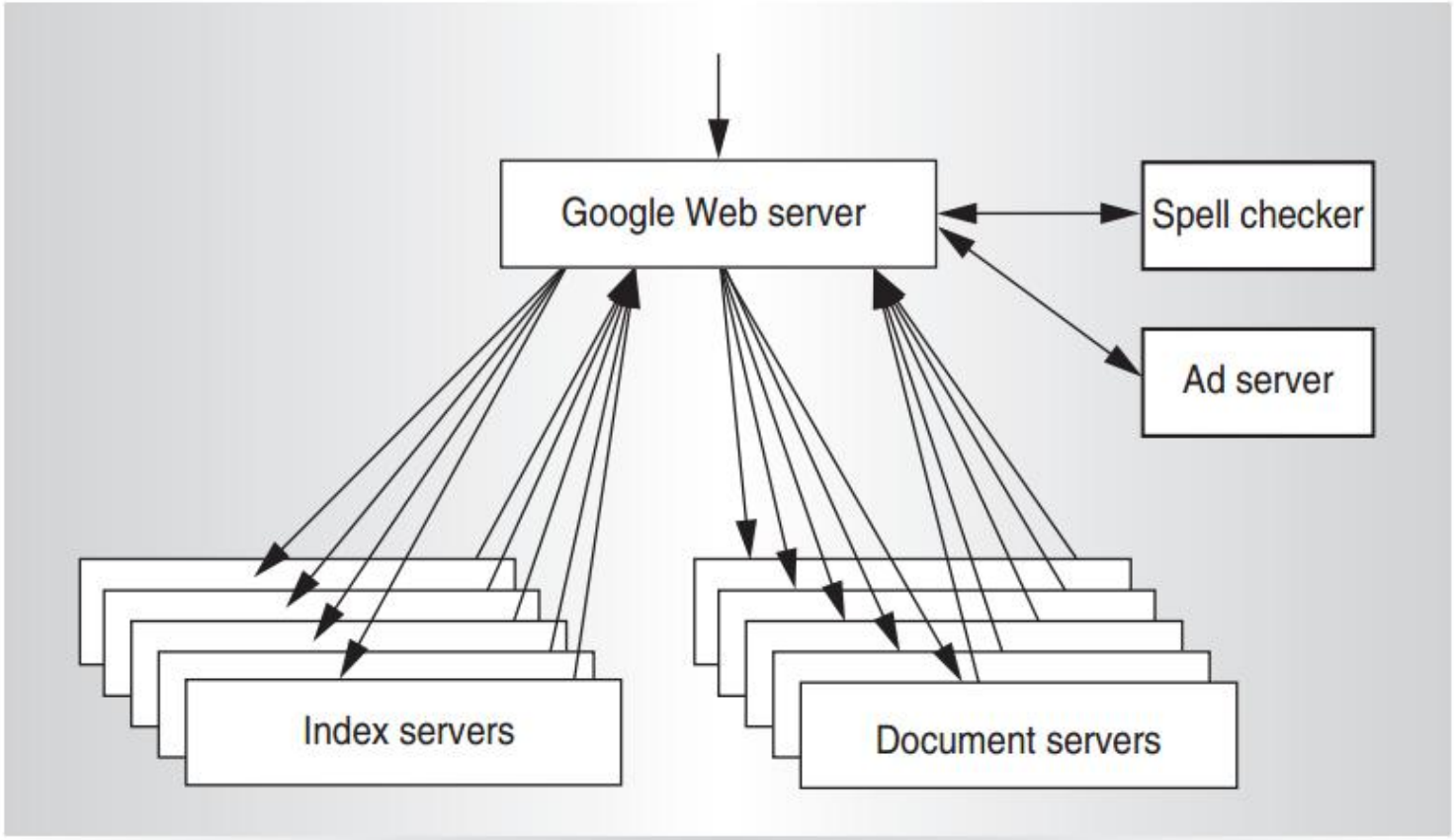
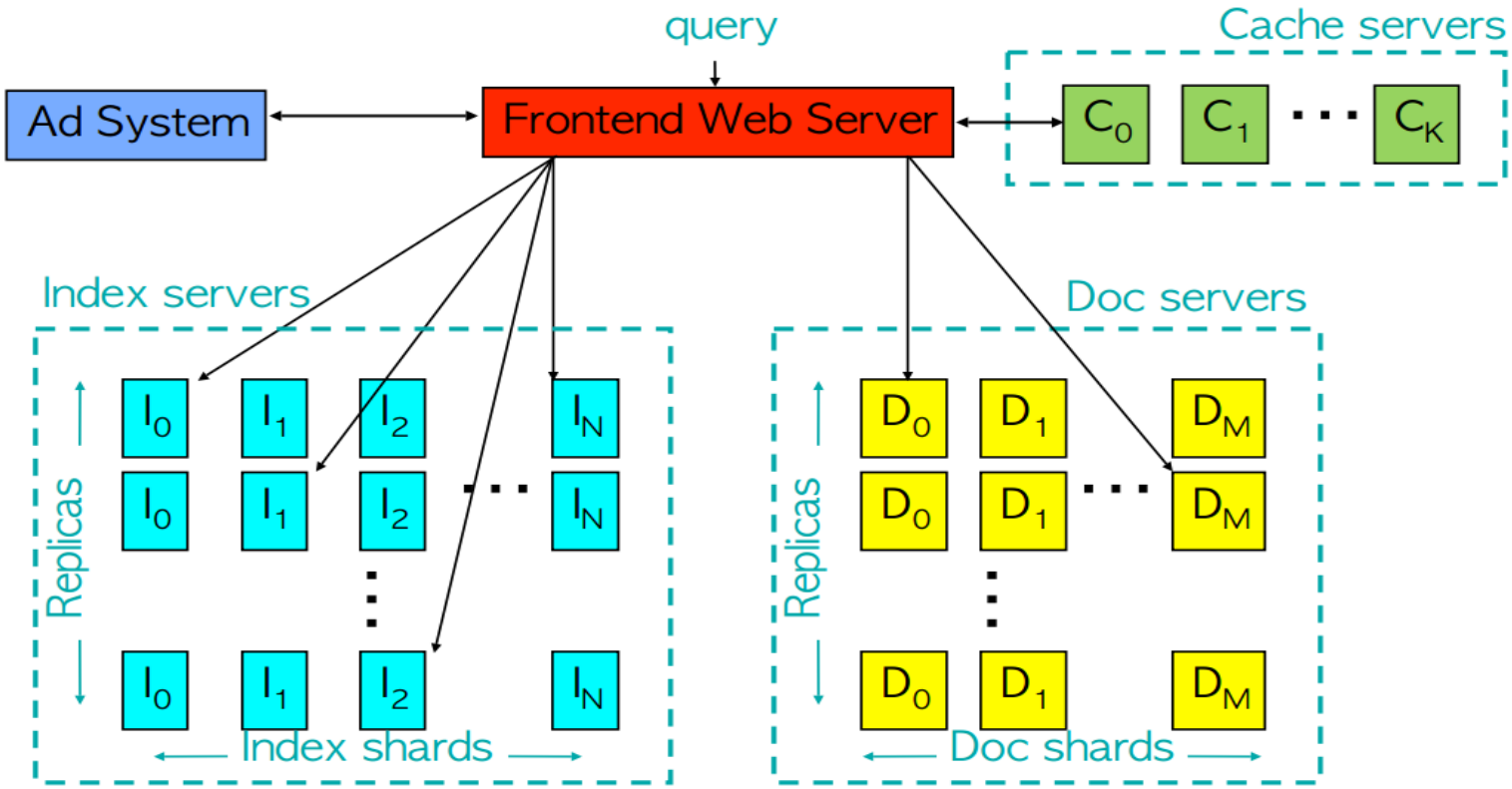


Figure 1. Google query-serving architecture.

Design Principles

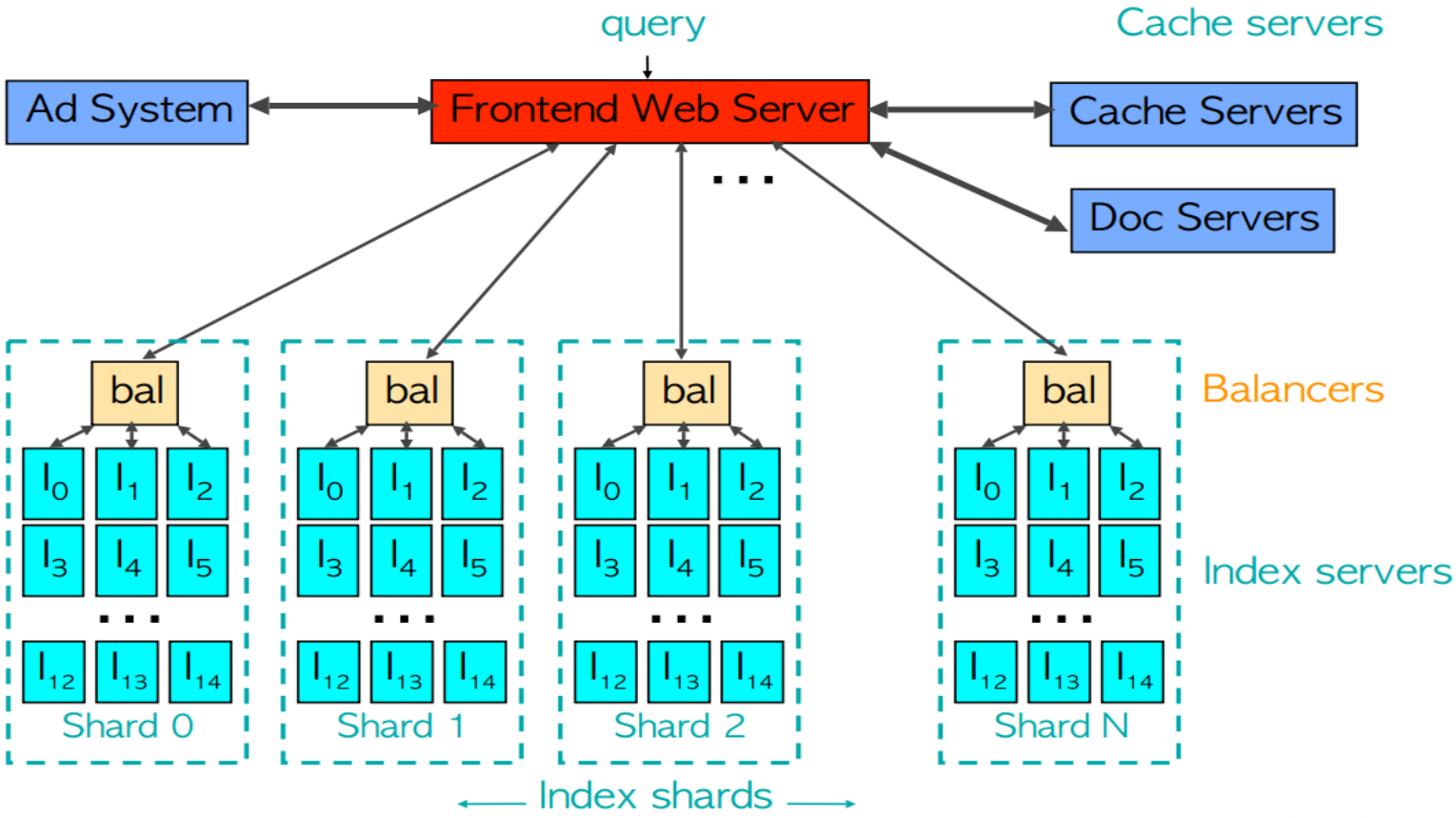
- Using replication for capacity and fault-tolerance
- Leveraging commodity parts
- Memory system
- Large-scale multiprocessing

Serving System, circa 1999



Google

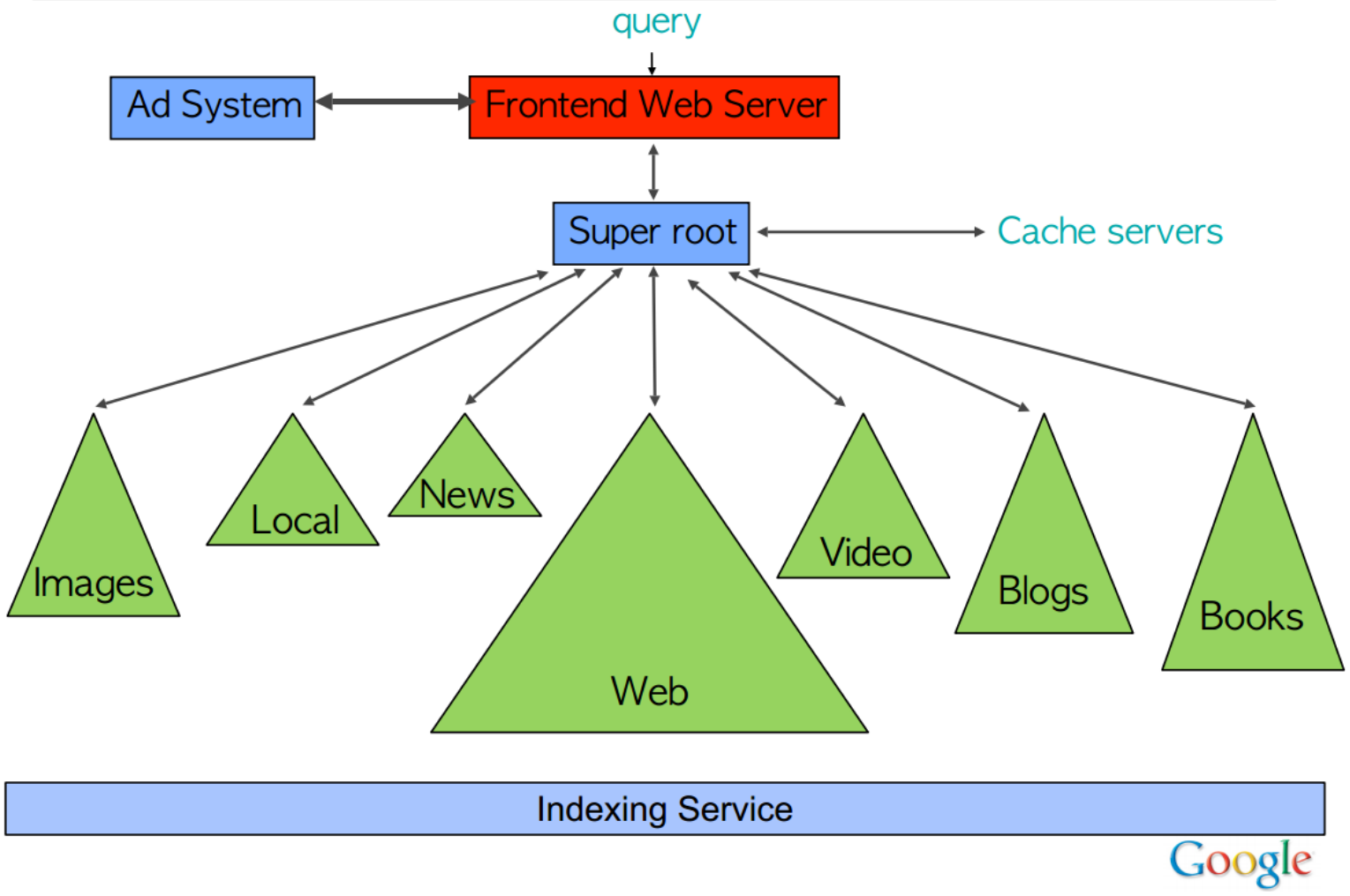
Early 2001: In-Memory Index



Google


+jing Search Images Videos Maps News Shopping Gmail More
jingm@gmail.com
Share
Google
notebook
Search
7 personal results and 94,000,000 other results (0.29 seconds)
Everything
Images
Maps
Videos
News
Shopping
More
Mountain View, CA
Change location
Any time
Past hour
Past 24 hours
Past 3 days
Past week
Past month
Past year
Custom range...
More search tools
Ads - Why these ads?
Dell Notebook Sale | dell.com
www.dell.com/Notebooks - 6,182 seller reviews
Save 35% on Notebooks w/ Intel® Core™ During Dell's Last Call Event
Check Out Best-Selling Notebooks
Dell Notebook Accessories
Amazingly Thin And Powerful XPS 14z
Dell Notebook Deals
New MacBook Air Notebook - Thin and light. Yet rock solid.
www.apple.com/macbookair
The ultimate everyday notebook.
Why you'll love a Mac - Which Mac is right for you? - OS X Lion - Great Mac apps
Notebook Shopping Guides | microsoft.com
www.microsoft.com/Laptops
Stream Movies & Music On Microsoft® Windows 7 PCs. Watch A Demo Today!
Browse PCs - Fit Finder
Ads - Why these ads?
Sony VAIO Notebooks
store.sony.com/VAIO
Portable High-Performance Notebooks
Smart performance w/Intel® Core™ i5
HP® Official Store
www.shopping.hp.com
shopping.hp.com is rated
Up to $285 off PCs w/ Visibly Smart
2nd Gen Intel® Core™ Processors!
Slim & Light Notebooks
www.acer.com/Notebook
Ultra-light, ultra-slim, with
Instant On and Instant Connect!
2nd Gen Intel® Core™ i5
www.intel.com/Corei5
Find a New 2nd Gen Intel® Core™ i5
Laptop Today! It's Visibly Smart.
Notebook - Low Prices
www.nextag.com/Notebook
Get Incredible Bargains on a Top
Brand Notebook. Find Many Models!
Target® Official Site
www.target.com
Shop Target.com & Get Free Shipping
on 500,000 Items With $50 Purchase.
Top 10 Cheap Laptops
www.priceseekers.net/Cheap+laptops
All Laptops Rated & Priced
Find Lowest Prices on PriceSeekers!
Computer Notebook
www.walmart.com/Computers
walmart.com is rated
Save on Computer Notebook with
Low Prices & Ship for Less!
See your ad here »
Welcome to Search plus Your World
Your photos, your friends, your stuff...
These results are just for you. Learn more
Related searches for notebook:
Stores: PCWorld Amazon Best Buy Staples Walmart
Brands: Dell HP Toshiba Acer Sony
Google Notebook
www.google.com/notebook
Notebook. Google recently stopped development on Notebook, which means it's no longer open to sign-ups by new users or being improved. Don't worry if ...
The Notebook (2004) - IMDb
www.imdb.com/title/tt0332280/
Rating: 7.9/10 - 142,259 votes
A poor and passionate young man falls in love with a rich young woman and gives her a sense of freedom. They soon are separated by their social differences.
Directed by Nick Cassavetes. Starring Gena Rowlands, James Garner.
Laptops & Notebooks - New Dell Laptop Computers for Sale | Dell
www.dell.com/us/p/laptops
Laptop & Notebook Computer Sales from the Official Dell Site. Dell offers everyday, performance, ultra-thin and gaming ready laptops. Build and ship yours ...
Laptop Reviews - Notebook Reviews and Netbook Computer News ...
www.notebookreview.com/
29 minutes ago – NotebookReview.com offers the latest notebook and laptop reviews, as well as price comparisons, support forums and news.
Images for notebook - Report images
Yuancheng Yang
Yuancheng Yang
Yuancheng Yang
Yuancheng Yang
Circus Ponies NoteBook - Award-Winning Mac and iPad Application ...
www.circusponies.com/
Circus Ponies NoteBook is the award-winning Mac application for getting organized on OS X. De-clutter your Desktop, track your tasks, manage your projects, ...
Best Prices on Notebooks - Shop & Compare | PCWorld
www.pcworld.com › ... › Computers › Notebooks & Accessories
Apple MacBook Pro 13.3" Silver Notebook (2.4 GHz Intel Core i5, 4 GB DDR3, 500 GB HDD, DVDRW DL, Intel HD 3000 Graphics, Mac OS X 10.7 Lion, LED ...
Notebook - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Notebook
A notebook (notepad, writing pad, drawing pad, legal pad) is a book or binder composed of pages of notes, often ruled, made out of paper, used for purposes ...
Amazon.com: Notebooks & Writing Pads: Memo & Scratch Pads ...
www.amazon.com/Notebooks-Writing-Pads/b?ie=UTF8&node...
Results 1 - 24 of 13432 – Online shopping for Notebooks & Writing Pads from a great selection of Office Products, Memo & Scratch Pads, Self-Stick Notes, ...
Moleskine ® - Legendary notebooks
www.moleskine.com/
Moleskine® is a brand that identifies a family of notebooks, diaries, and city guides flexible and brilliantly simple tools for use both in everyday and extraordinary ...
Laptops & Notebook Computers | HP Laptop Computers
welcome.hp.com/country/us/en/prodserv/laptops.html
Find the HP laptop just for you. As the #1 provider of laptops in the world, HP offers affordable and powerful laptop and notebook computers. Compare and shop ...
Shopping results for notebook
Apple MacBook Pro - Core i5 2.3 GHz - 4 GB Ram
553 reviews - $700 - 25 stores - Nearby stores - In stock
119 people +1'd this
Apple MacBook Air - Core i5 1.7 GHz - 4 GB Ram
128 reviews - $1,225 - 43 stores - Nearby stores - In stock
Apple MacBook Air - Core i5 1.6 GHz - 2 GB Ram
73 reviews - $910 - 36 stores - Nearby stores - In stock
Searches related to notebook
notebook vs laptop
notebook quotes
notebook paper
notebook wiki
notebook software
notebook laptop
notebook movie
dell notebook
Goooooooooogle
1 2 3 4 5 6 7 8 9 10 Next
Advanced search Search Help Give us feedback
Google Home Advertising Programs Business Solutions Privacy
About Google

2007: Universal Search

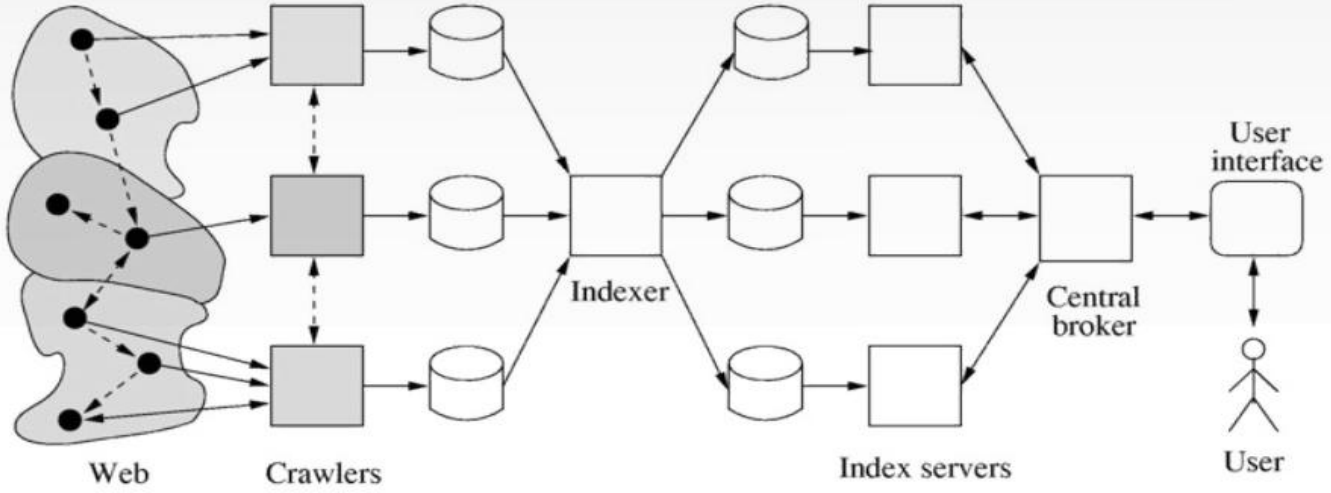


Jiepang System

- Render
- AD
- Product

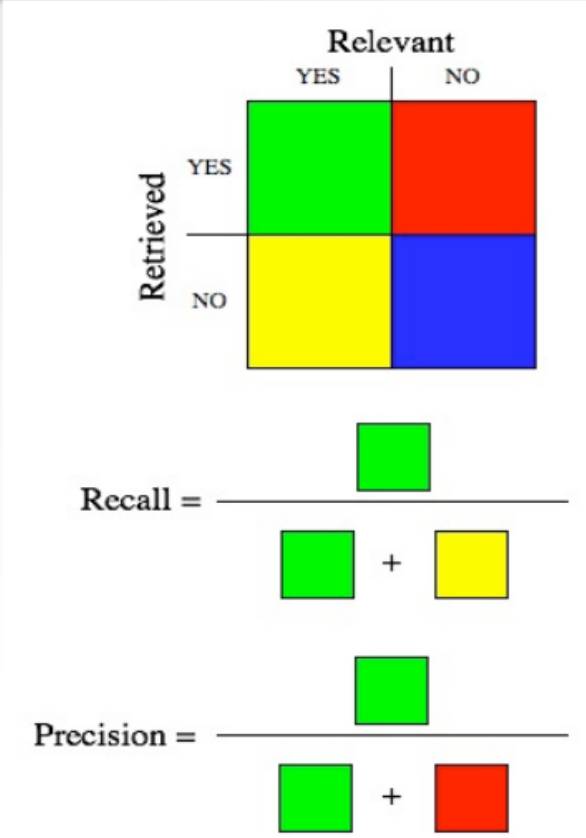
Components of a Search Engine

- Three main components in a search engine
  - crawling
  - indexing
  - query processing

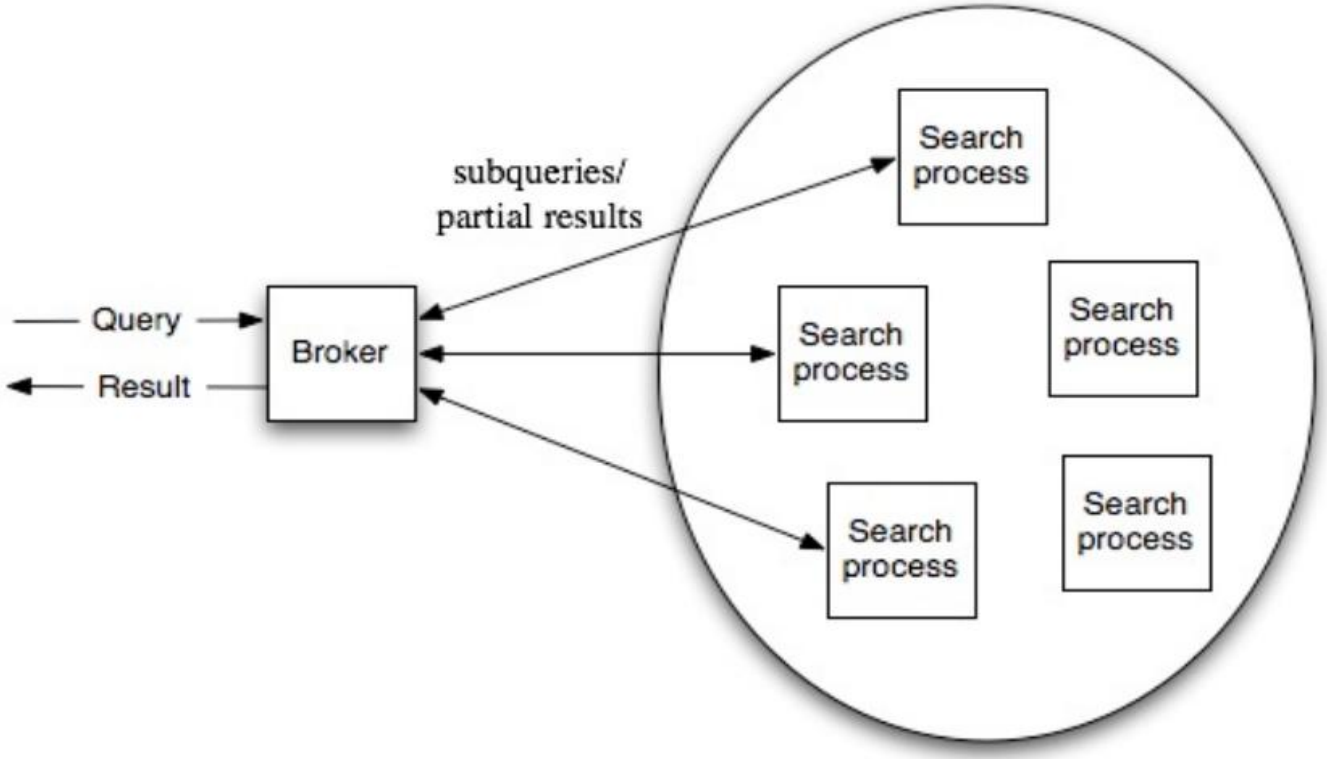


Ranking

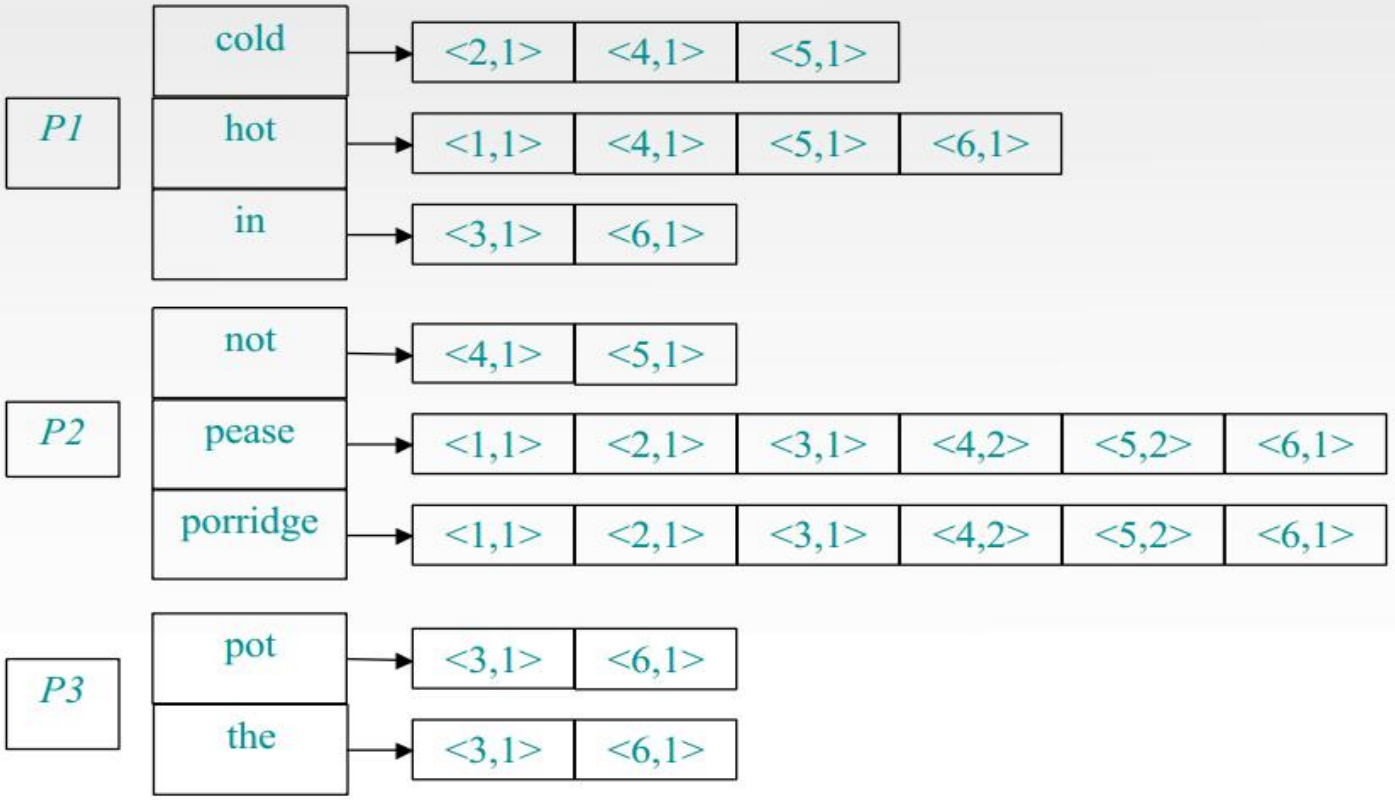
- Two important measures
  - recall
  - precision
- Most important features
  - Content (e.g., tf-idf)
  - URL (e.g., site importance)
  - Link (e.g., PageRank)
  - Spam (e.g., porn)
  - Click



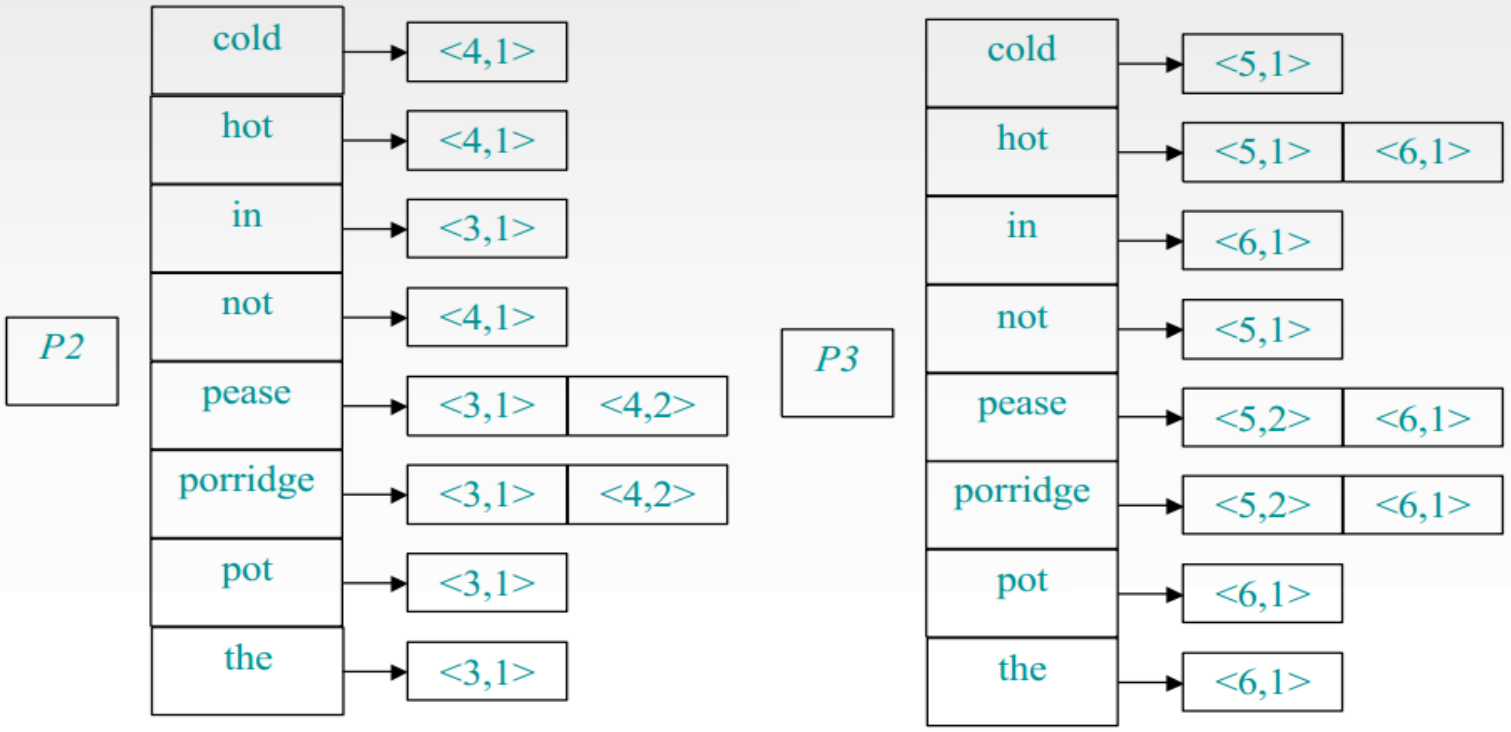
Intra-query Parallelism



Term-Based Partitioning

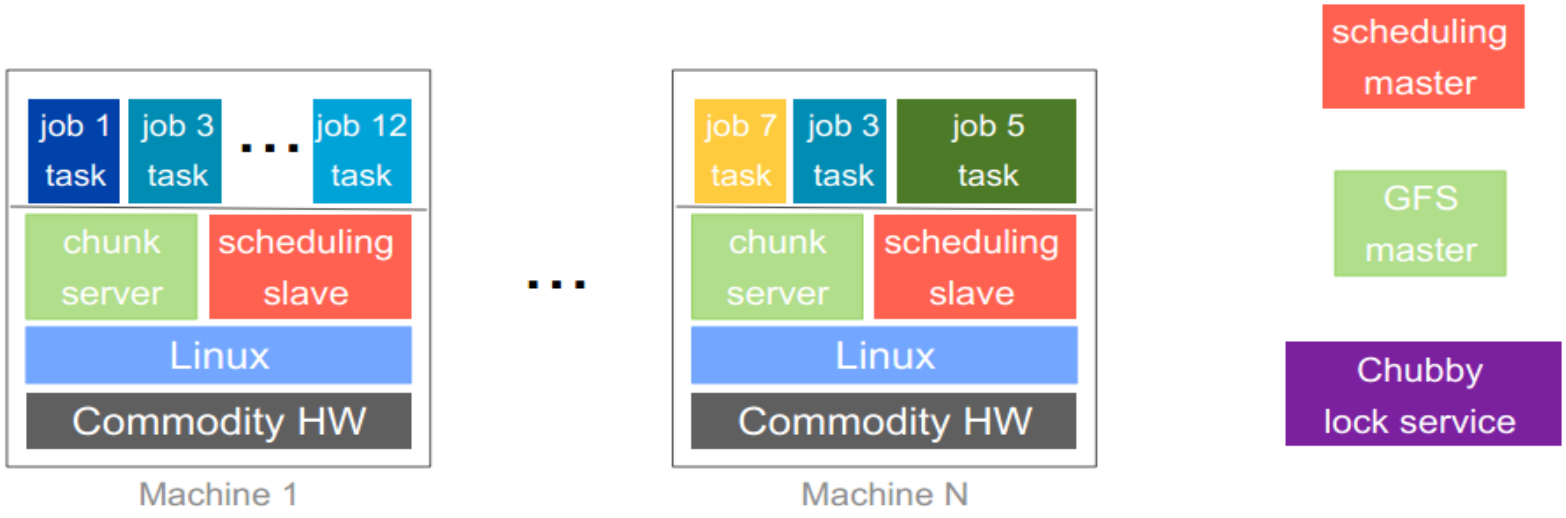


Document-Based Partitioning



Google Cluster Environment

- Cluster is 1000s of machines, typically one or handful of configurations
- File system (GFS) + Cluster scheduling system are core services
- Typically 100s to 1000s of active jobs (some w/1 task, some w/1000s)
  - mix of batch and low-latency, user-facing production jobs



Google

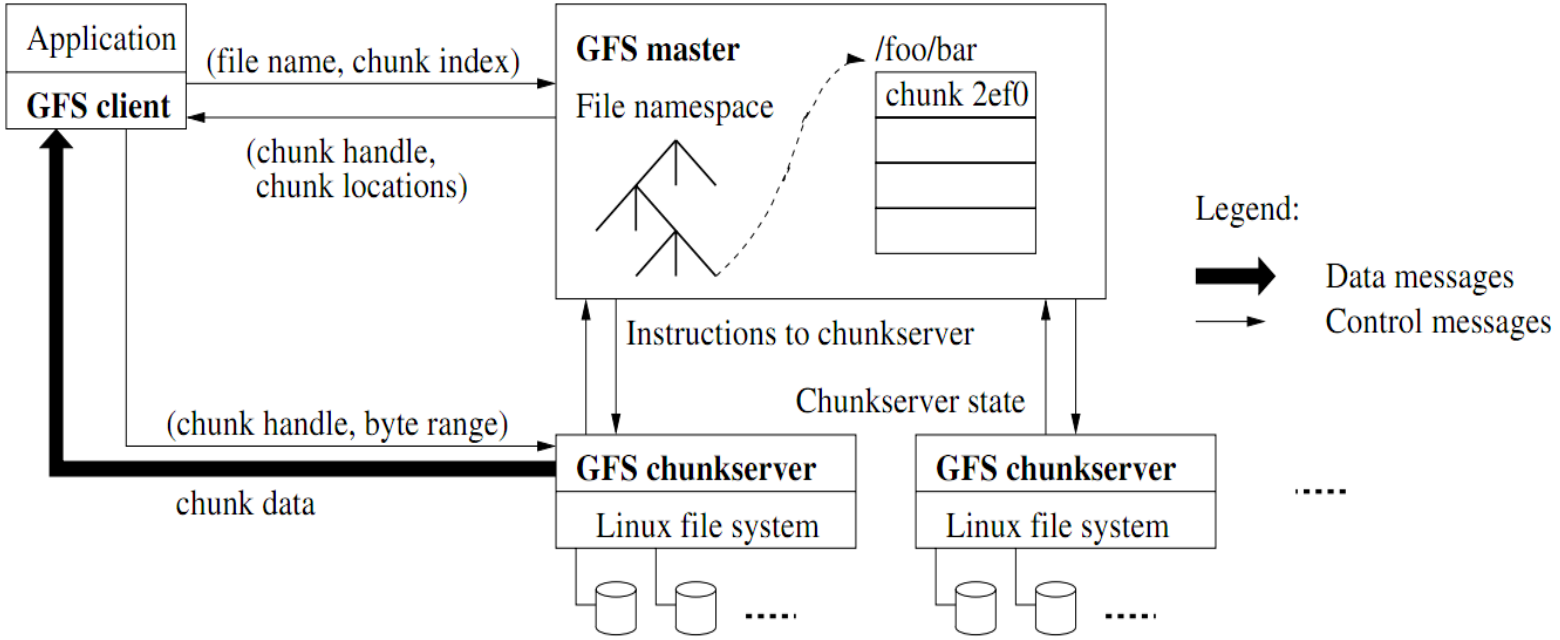
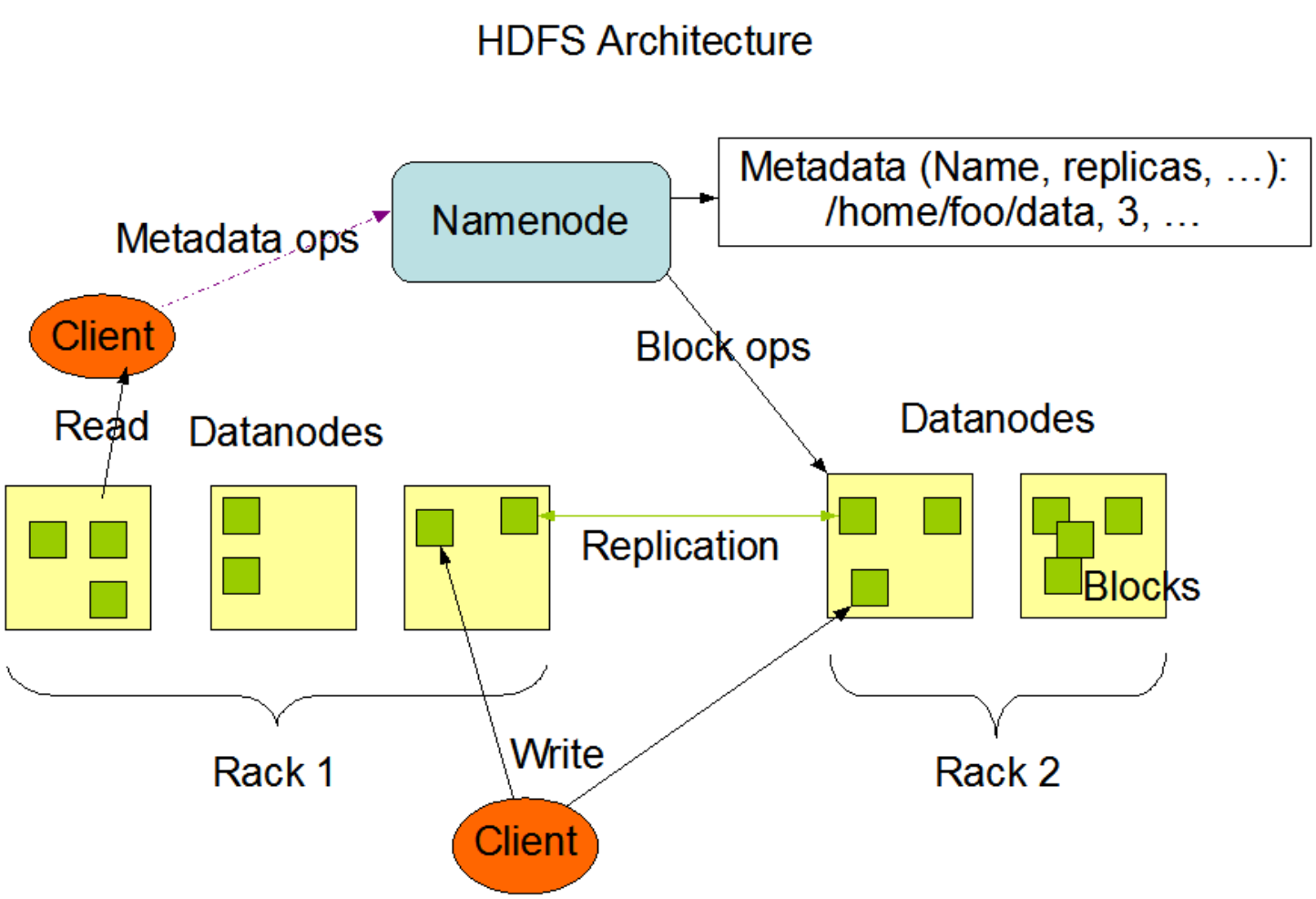
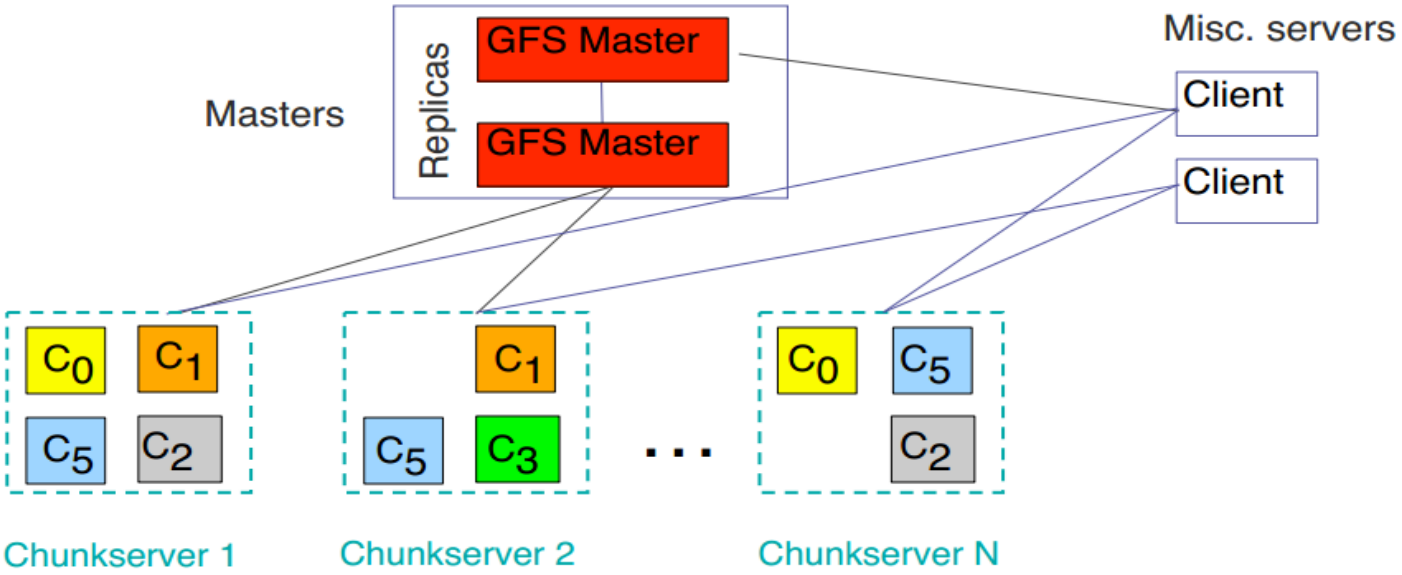


Figure 1: GFS Architecture


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Client
Rack 2

GFS Design



- Master manages metadata
- Data transfers happen directly between clients/ chunkservers
- Files broken into chunks (typically 64 MB)

Google

GFS Usage @ Google

- 200+ clusters
- Many clusters of 1000s of machines
- Pools of 1000s of clients
- 4+ PB Filesystems
- 40 GB/s read/write load
  - (in the presence of frequent HW failures)

Google

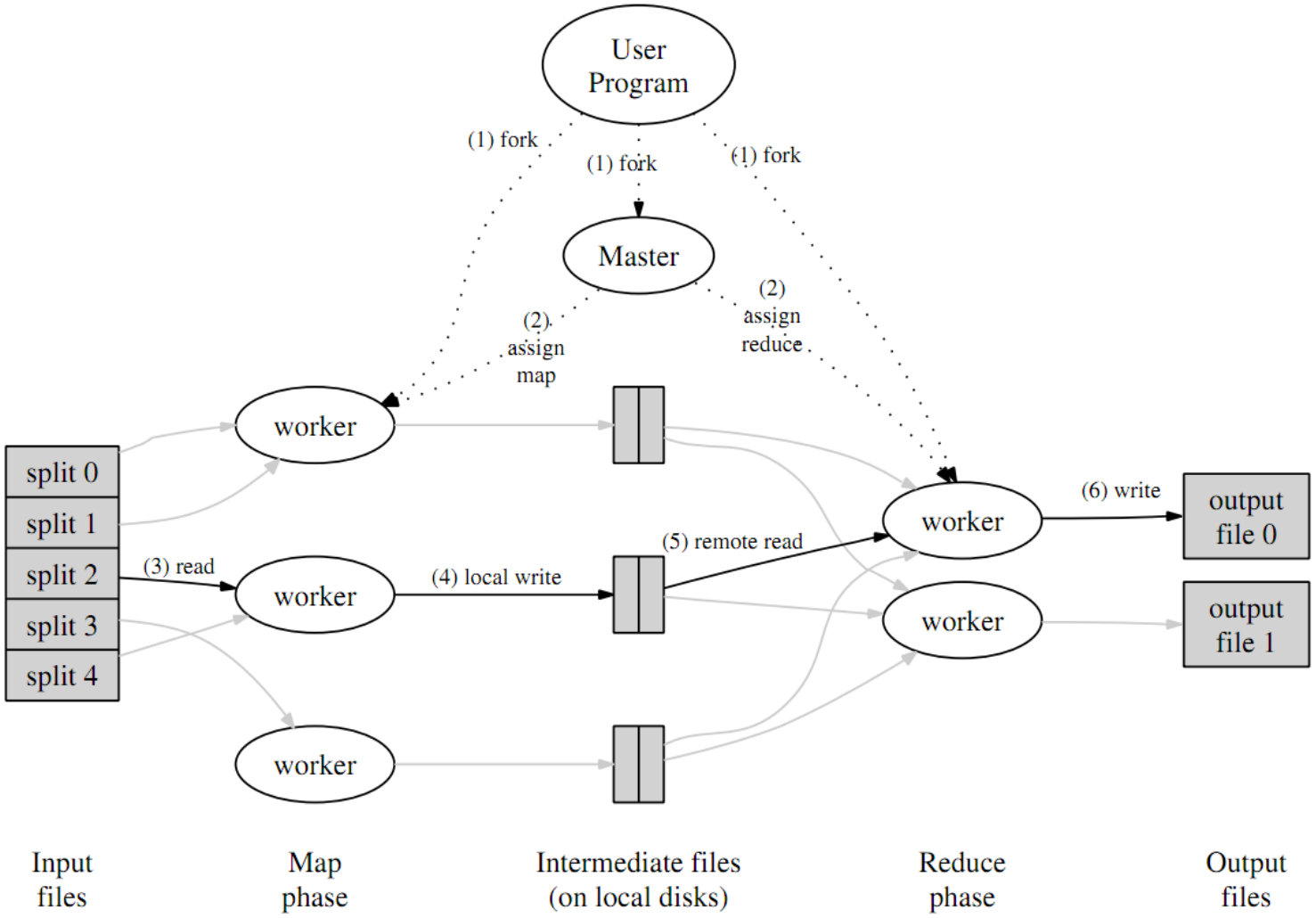
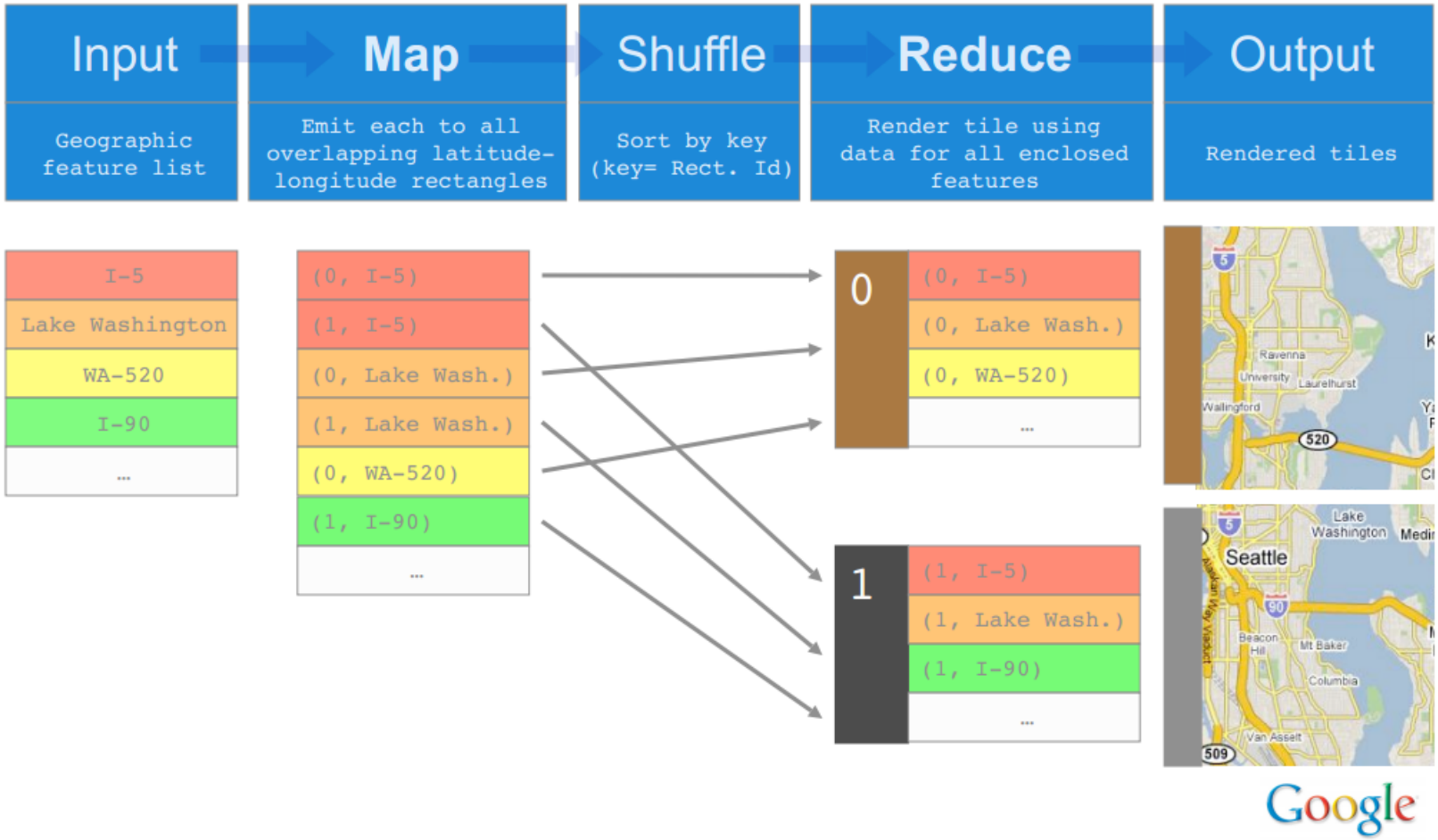


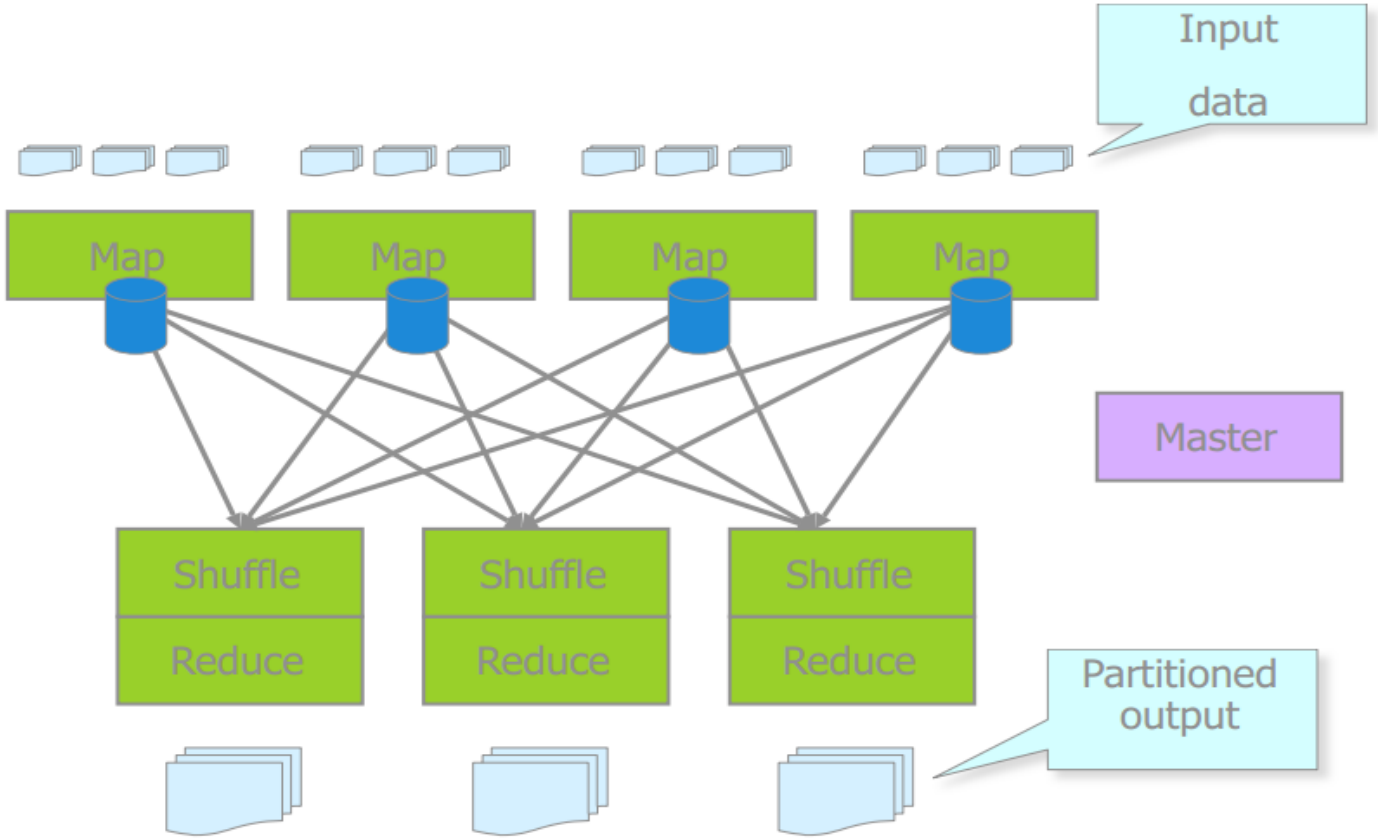
Figure 1: Execution overview

Example: Rendering Map Tiles

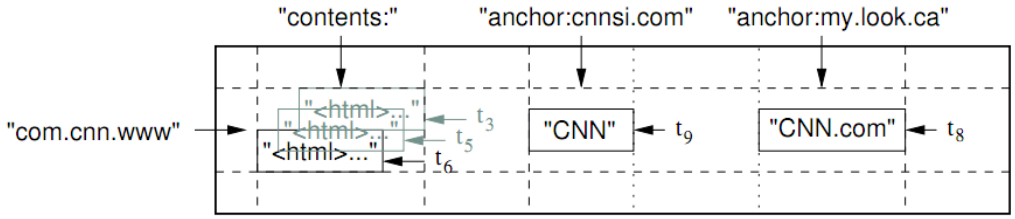


Google

Parallel MapReduce

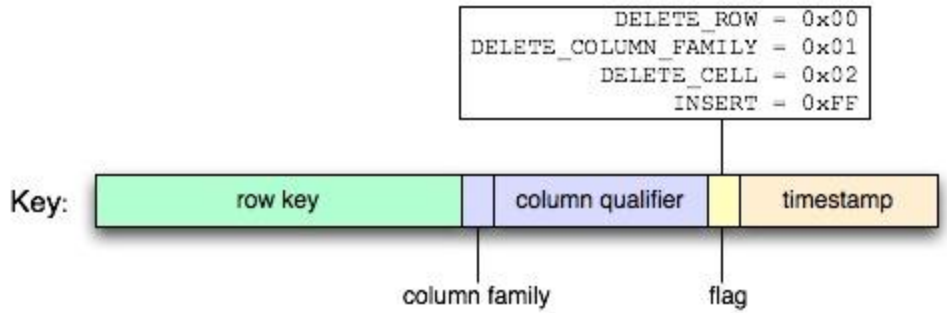


Google



crawldb Table

	title	content	anchor	
			anchor:com.apple.www/	**anchor**:com.redherring.www/
com.facebook.www	Facebook I Home *2008-02-11 15:14:01*	<!DOCTYPE html PUBLIC "-//W3C... *2008-02-11 15:14:01*	Facebook *2008-02-11 15:14:01* *2008-02-03 19:27:57* *2008-01-22 08:46:28*	Facebook *2008-02-11 15:14:01* *2008-02-03 19:27:57*
com.yahoo.www	Yahoo! *2008-02-10 21:12:09*	<html><head> <meta http-equiv="Content-... *2008-02-10 21:12:09*		
com.zvents.www	Discover Things To Do - Zvents *2008-02-07 08:32:22*	<html xmlns="http://www.w3.org/1999/xhtml"> ... *2008-02-07 08:32:22*	**anchor**:org.slashdot.www/ Zvents *2008-02-07 08:32:22* *2008-02-01 23:06:35* *2008-01-23 11:19:36*	
org.hypertable.www	Hypertable: An Open Source, High Performance, ... *2008-02-11 13:41:53*	<!DOCTYPE html PUBLIC "-//W3C// DTD XHTML 1.0... *2008-02-11 13:41:53*		
	2008-02-02 09:17:41	*2008-02-02 09:17:41*		
	2008-01-25 17:44:13	*2008-01-25 17:44:13*		



crawldb Table

key	value
com.facebook.www title *2008-02-11 15:14:01*	Facebook | Home
com.facebook.www title *2008-02-03 19:27:57*	Facebook | Home
com.facebook.www title *2008-01-22 08:46:28*	Facebook | Home
com.facebook.www content *2008-02-11 15:14:01*	<!DOCTYPE html PUBLIC "-//W3C//DTD...
com.facebook.www content *2008-02-03 19:27:57*	<!DOCTYPE html PUBLIC "-//W3C//DTD...
com.facebook.www content *2008-01-22 08:46:28*	<!DOCTYPE html PUBLIC "-//W3C//DTD...
com.facebook.www anchor:com.apple.www/ *2008-02-11 15:14:01*	Facebook
com.facebook.www anchor:com.apple.www/ *2008-02-03 19:27:57*	Facebook
com.facebook.www anchor:com.apple.www/ *2008-01-22 08:46:28*	Facebook
com.facebook.www anchor:com.redherring.www/ *2008-02-11 15:14:01*	Facebook
com.facebook.www anchor:com.redherring.www/ *2008-02-03 19:27:57*	Facebook
com.yahoo.www title *2008-02-10 21:12:09*	Yahoo!
com.yahoo.www title *2008-02-04 03:46:22*	Yahoo!
com.yahoo.www title *2008-01-22 08:46:28*	Yahoo!
com.yahoo.www content *2008-02-10 21:12:09*	<html><head><meta http-equiv="Content-...
com.yahoo.www content *2008-02-04 03:46:22*	<html><head><meta http-equiv="Content-...
...	...

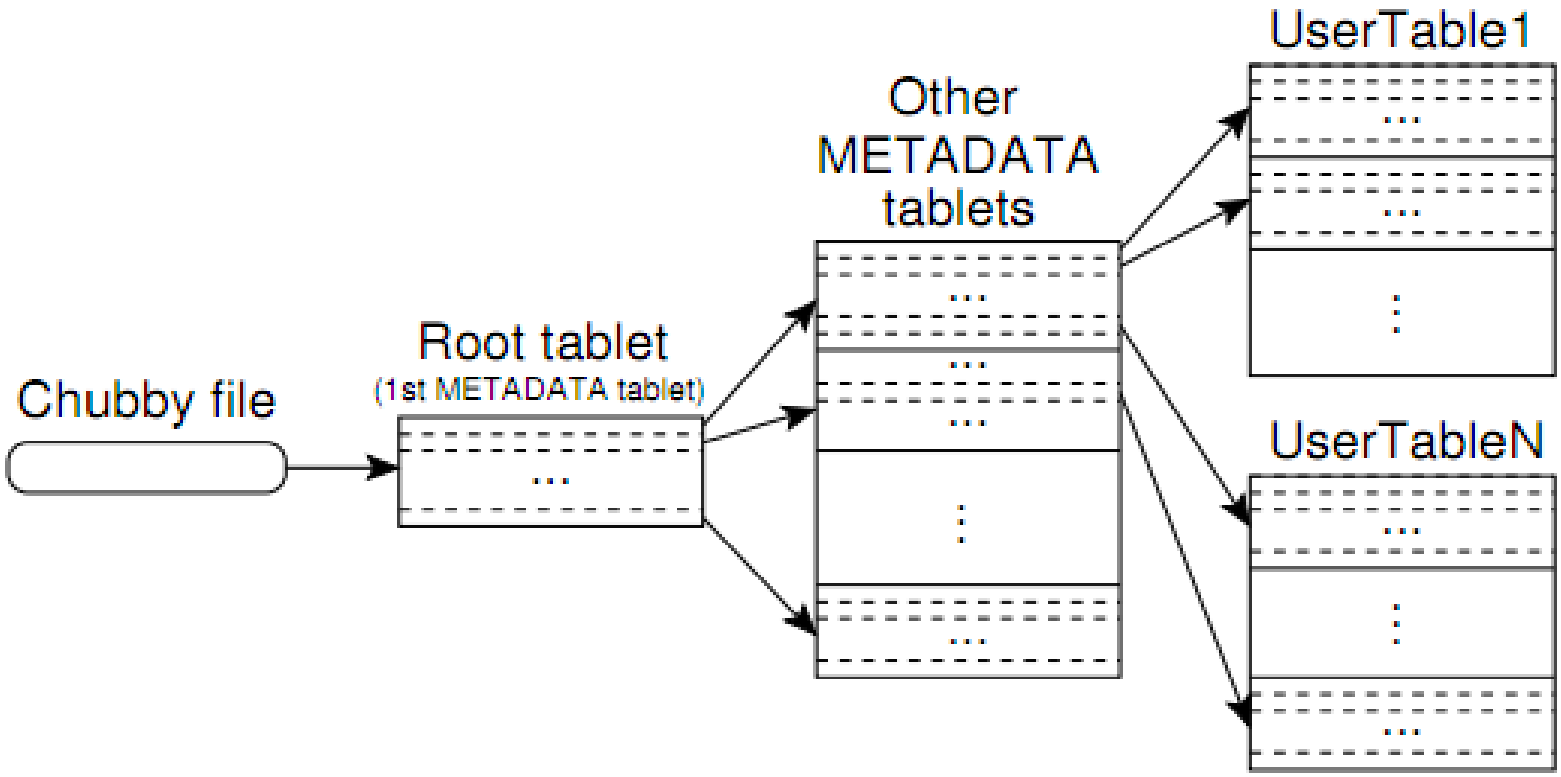
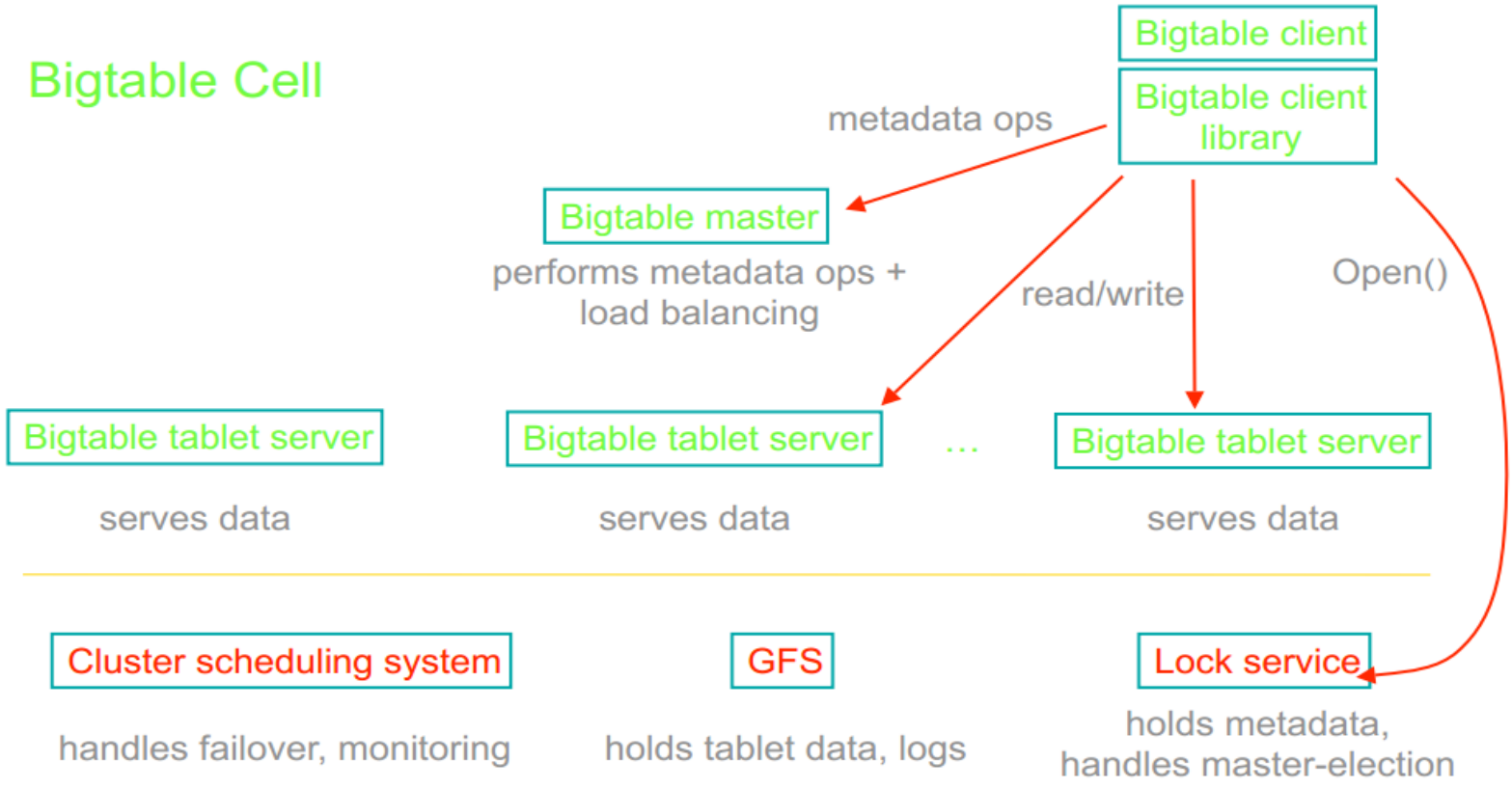


Figure 4: Tablet location hierarchy.

BigTable System Structure



Google

Chubby

- The ***Chubby*** lock service for loosely-coupled distributed systems

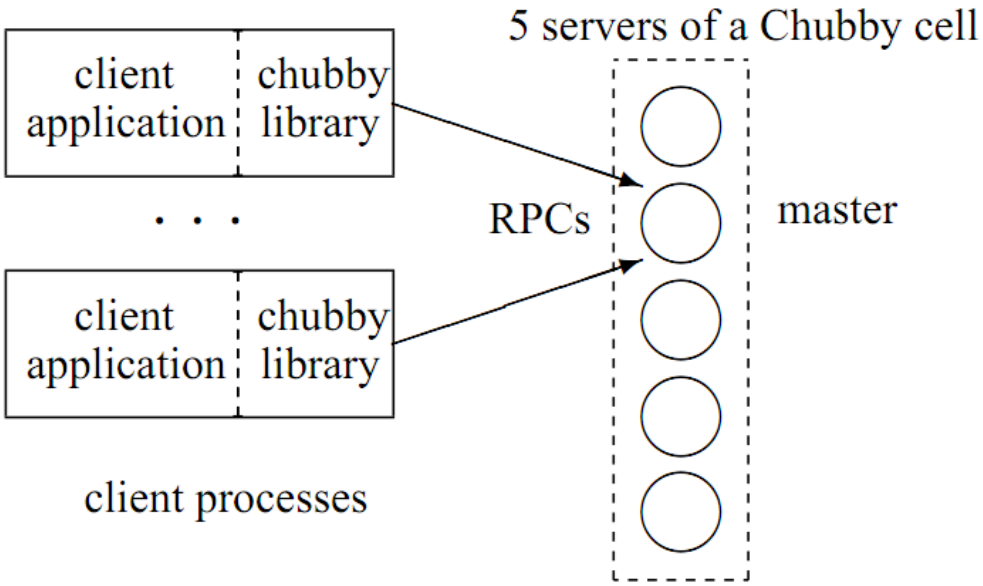
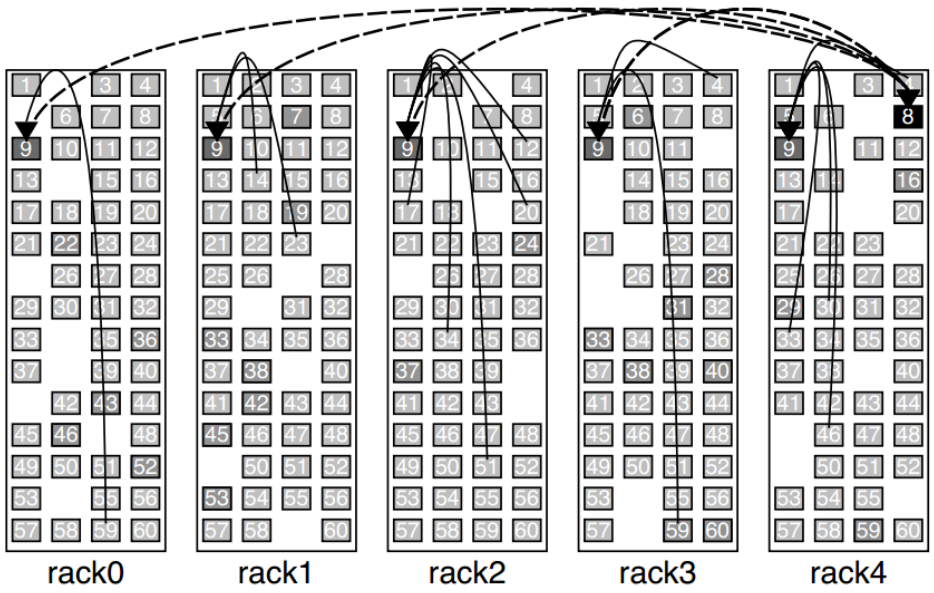
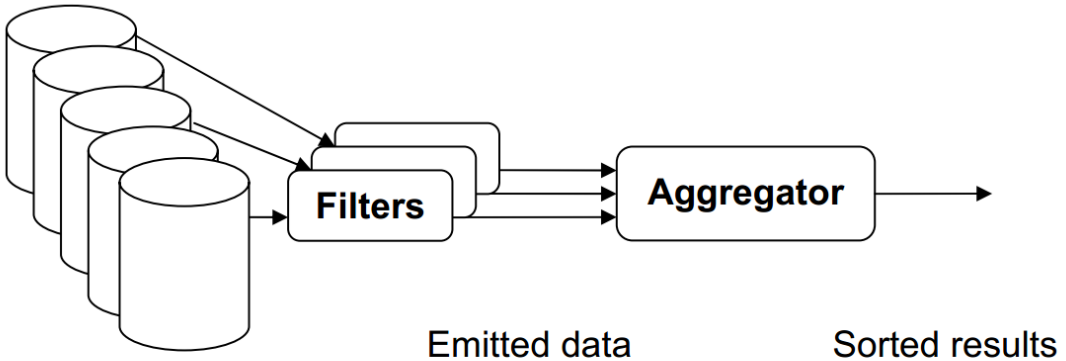


Figure 1: System structure

Sawzall


rack0
rack1
rack2
rack3
rack4


Filters
Aggregator
Raw data
Emitted data
Sorted results

Caffine

- Large-scale Incremental Processing Using Distributed Transactions and Notifications
-
- http://googleblog.blogspot.com/2010/06/our-new-search-index-caffeine.html