Distributed Systems

COMP90015 2021 Semester 1
Tutorial 10

Today’s Agenda

- Distributed File System (DFS) questions (Overview)
- Case Study: Drop Box

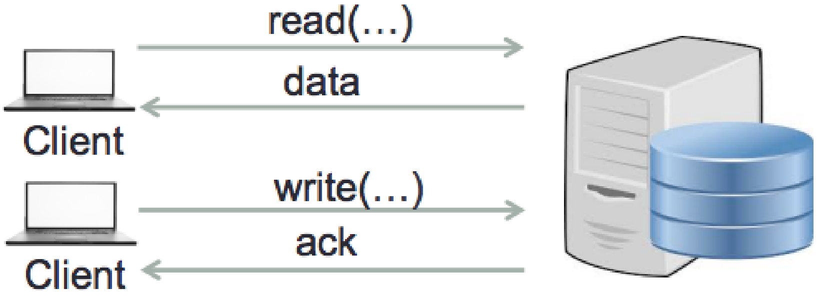
1. Name and explain transparencies that should be addressed by distributed file systems.

2. What are the advantages and disadvantages of using absolute names as a naming strategy?

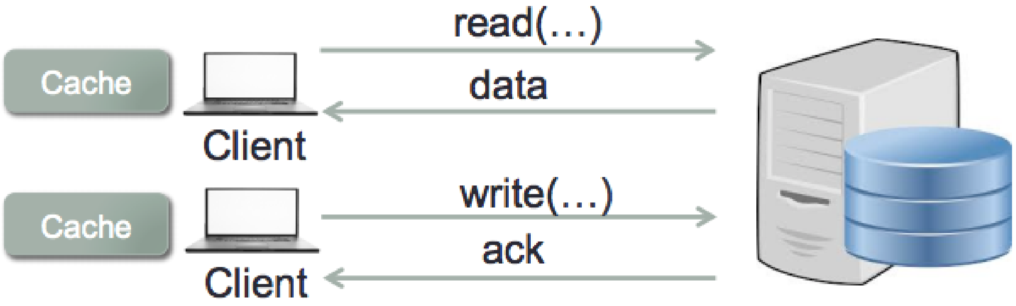
3. What are the advantages and disadvantages of a naming strategy based on mount points?

4. What are the advantages and disadvantages of the following two types of distributed file systems?

(A)

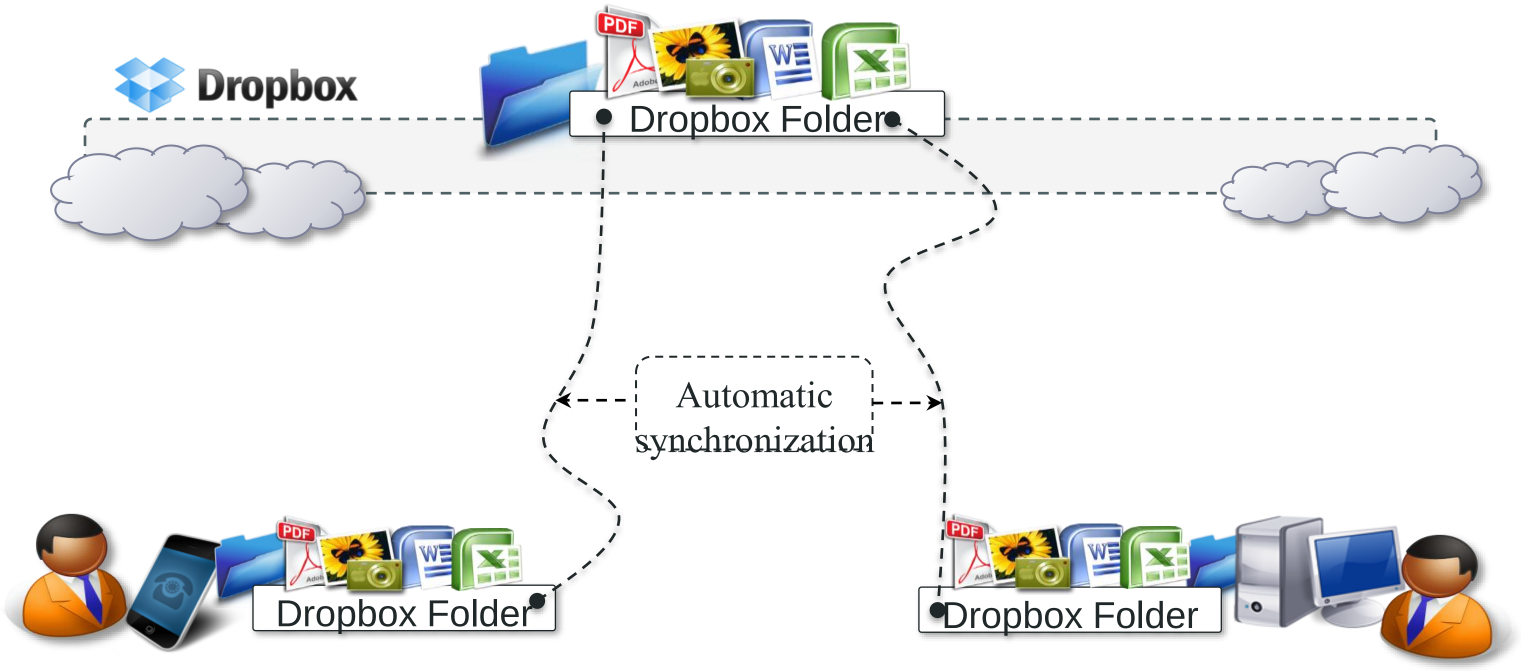


(B)



Case Study: Dropbox

Dropbox



Dropbox
Dropbox Folder
Automatic synchronization
Dropbox Folder
Dropbox Folder

Dropbox

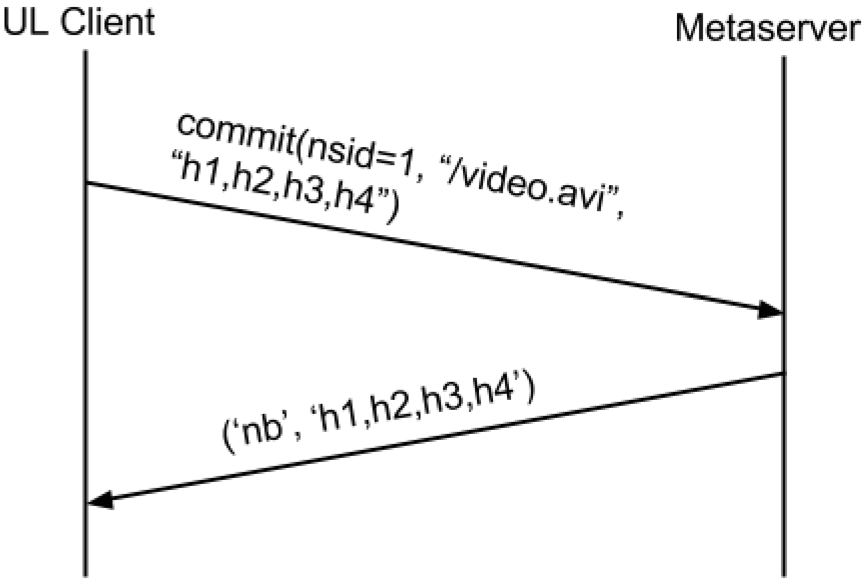
- Client runs on desktop
- Copies changes to local folder
  - Uploaded automatically
  - Downloads new versions automatically
- Huge scale – 100+ million users, 1 billion files/day
- Design
  - Small client, few resources
  - Possibility of low-capacity network to user
  - Scalable back-end
  - (99% of code in Python)

Dropbox

- Everyone’s computer has complete copy of Dropbox
    - Run daemon on computer to track “Sync” folder
- Traffic only when changes occur
    - Results in file upload : file download
    - Huge number of uploads compared to traditional service
    - Dropbox service’s read/write ratio is ***1:1***
- Uses compression to reduce traffic

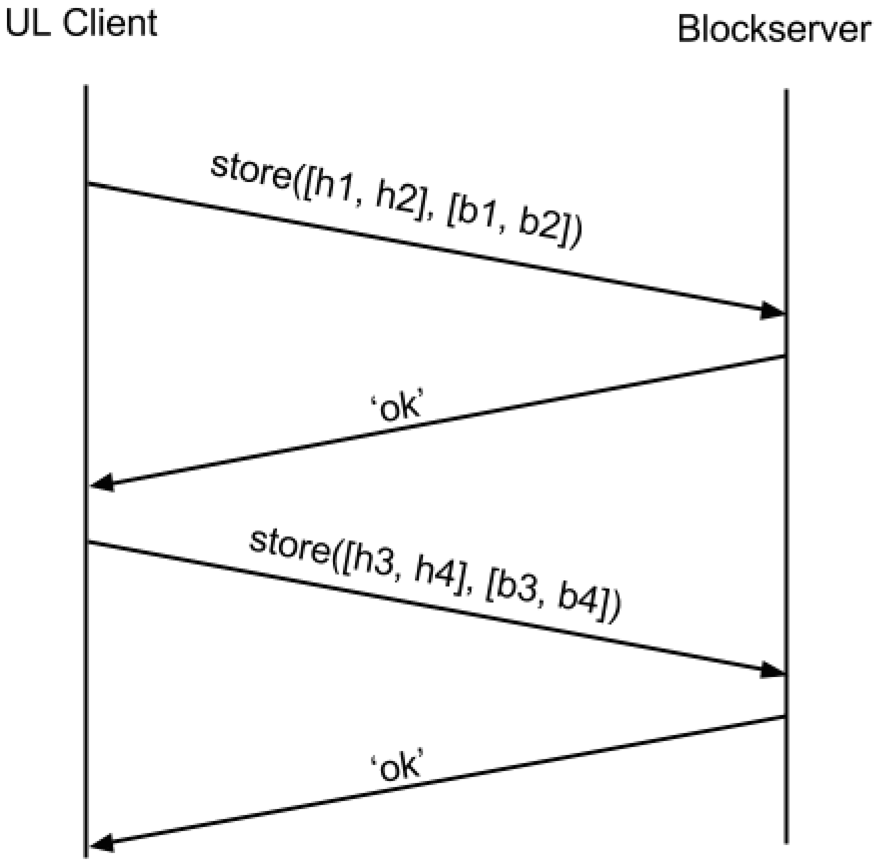
Dropbox - Upload

- Client attempts to “commit” new file
    - Breaks file into blocks, computes hashes
    - Contacts Metaserver
- Metaserver checks if hashes known
- If not, Metaserver returns that it “needs blocks” (nb)



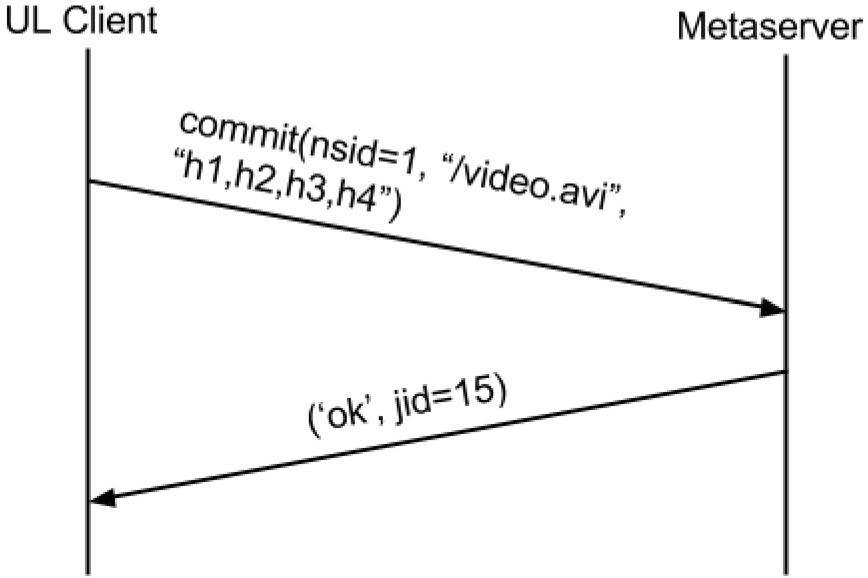
Dropbox - Upload

- Client talks to Blockserver to add needed blocks
- Limit bytes/request (typically 8 MB), so may be multiple requests



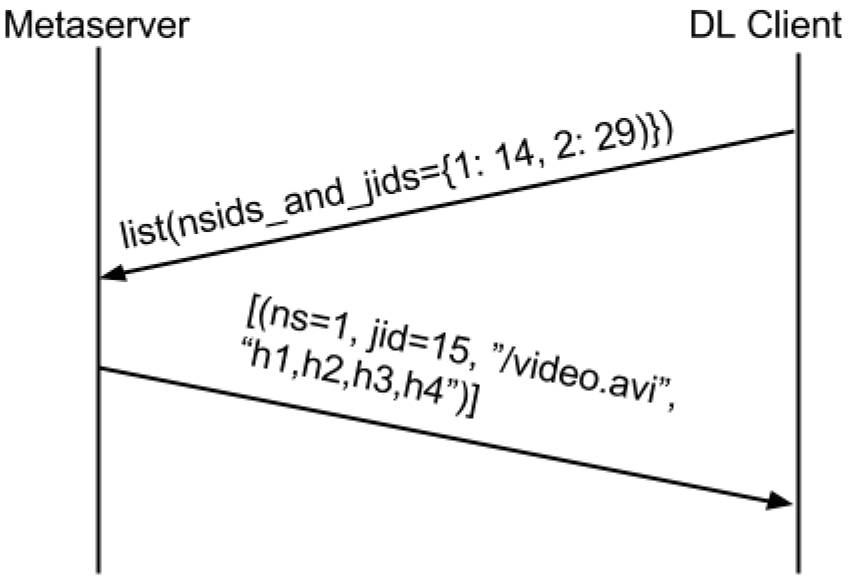
Dropbox - Upload

- Client commits again
  - Contacts Metaserver with same request
- This time, ok



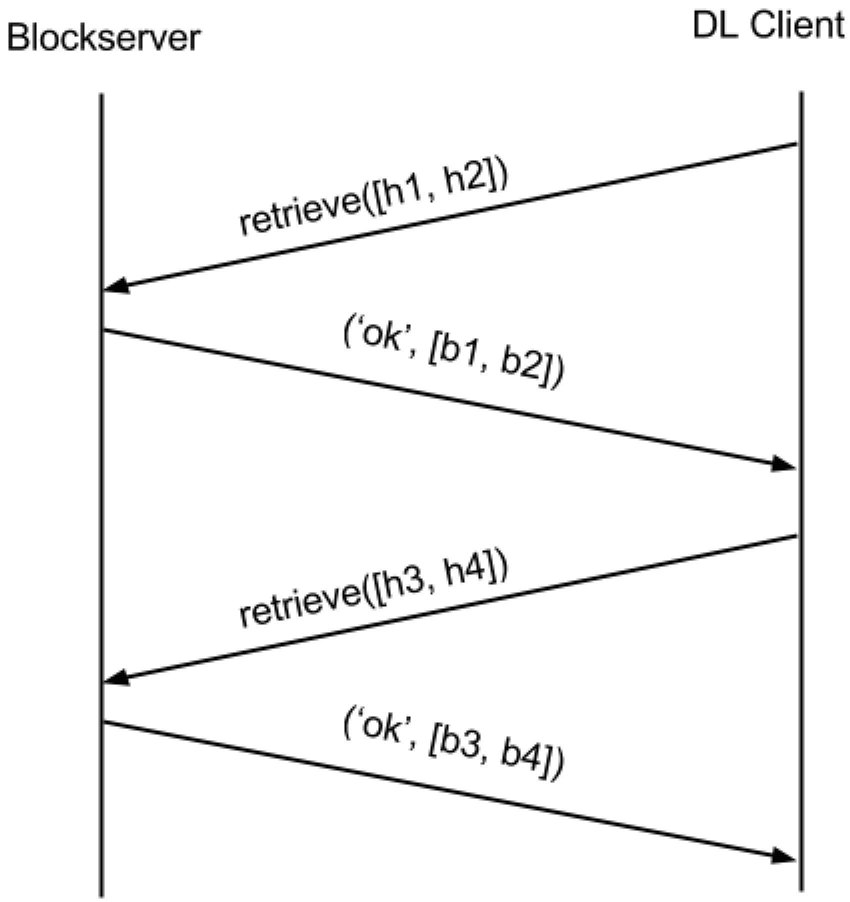
Dropbox - Download

- Client periodically polls Metaserver
  - Lists files it "knows about"
- Metaserver returns information on new files



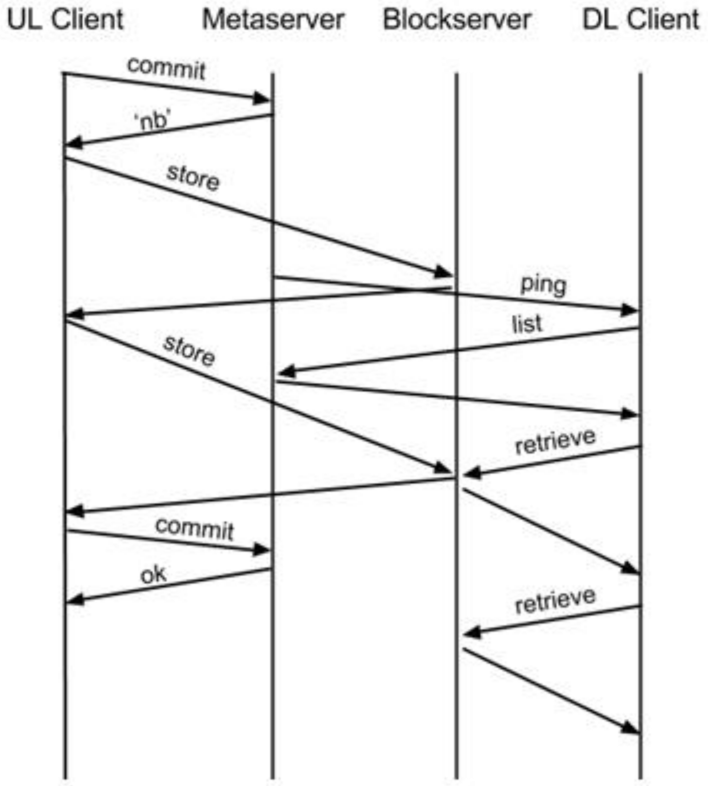
Dropbox - Download

- Client checks if blocks exist
    - For new file, this fails
- Retrieve blocks
- Limit bytes/request (typically 8 MB), so may be multiple requests
- When done, reconstruct and add to local file system
    - Using local filesystem system calls (e.g., open(), write()...)



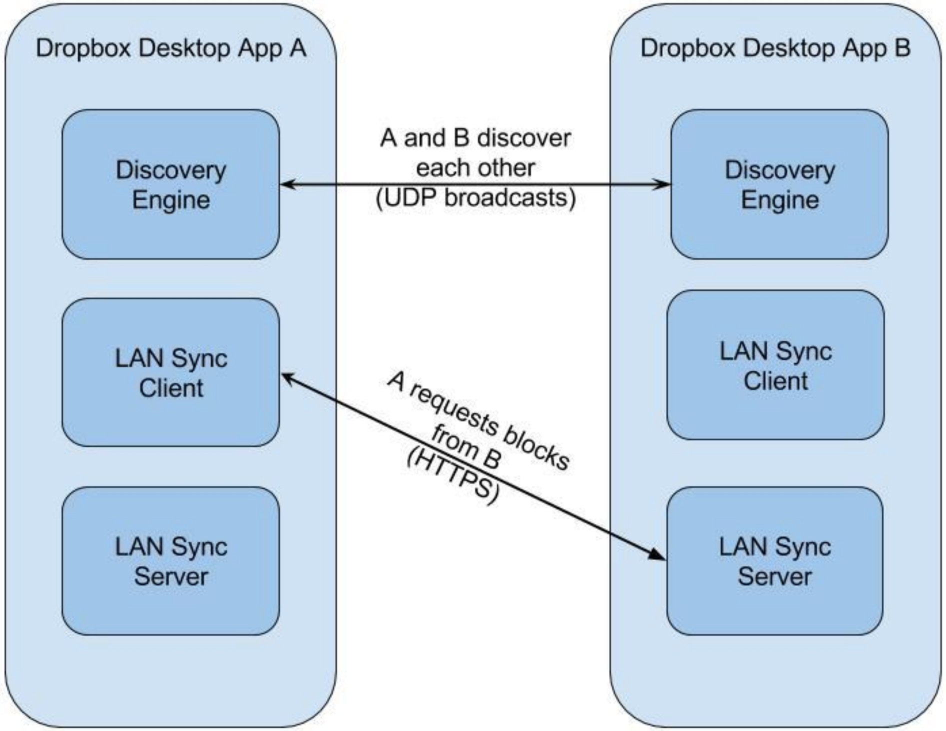
Dropbox – Streaming Sync

- Normally, cannot download to another until upload complete
  - For large files, takes time "sync"
- Instead, enable client to start download when some blocks arrive, before commit
  - Streaming Sync



Dropbox – LAN Sync

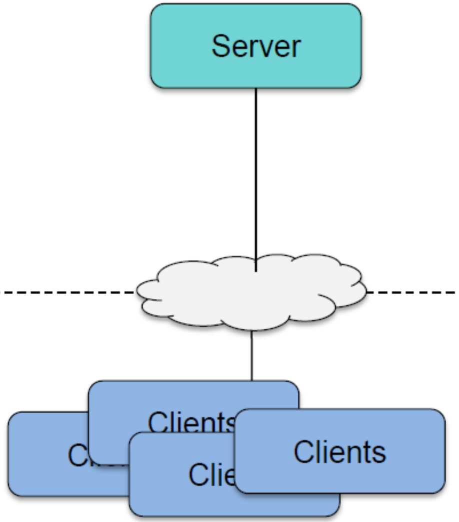
- LAN Sync – download from other clients
- Periodically broadcast on LAN (via UDP)
- Response to get TCP connection to other clients
- Pull blocks over HTTP



DropBox: Architecture Evolution

Dropbox Architecture – v1

– One server: web server, app server, mySQL database, sync server

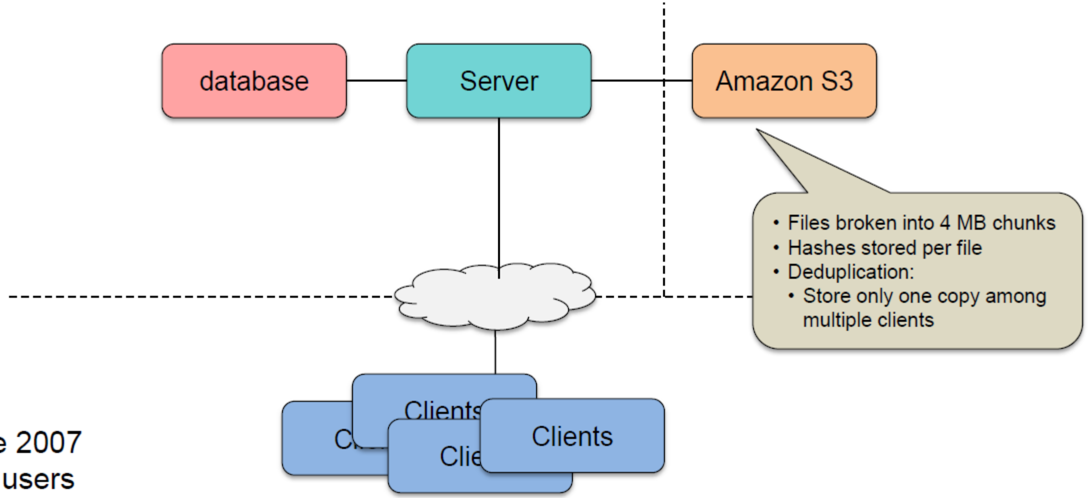


mid 2007
0 users

See http://youtu.be/PE4gwstWhmc

Dropbox Architecture – v2

- Server ran out of disk space:
  moved data to Amazon S3 service (key-value store)
- Servers became overloaded: moved mySQL DB to another machine
- Clients polled server for changes periodically

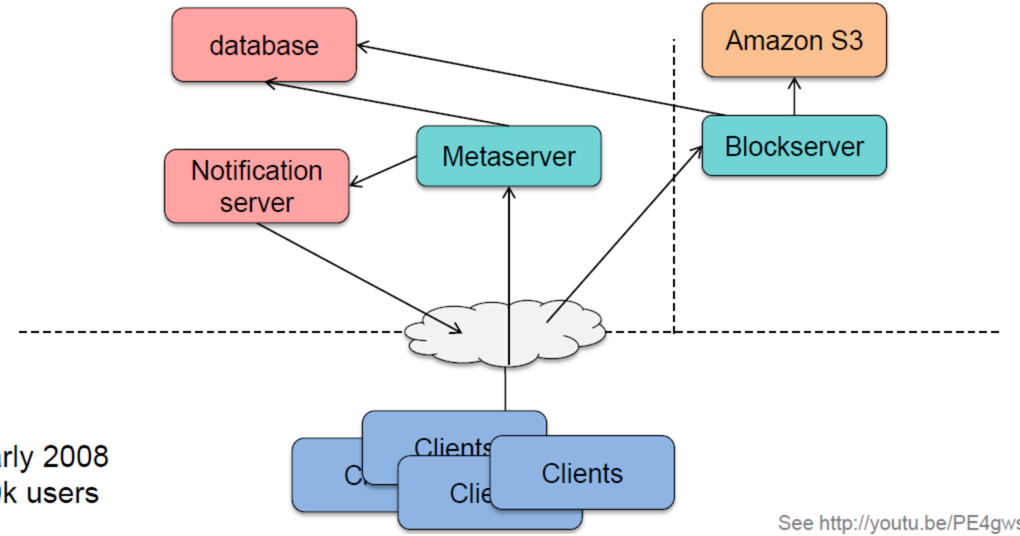


late 2007
~0 users

See http://youtu.be/PE4gwstWhmc

Dropbox Architecture – v3

- Move from polling to notifications: add notification server
- Split web server into two:
  - Amazon-hosted server hosts file content and accepts uploads (stored as blocks)
  - Locally-hosted server manages metadata

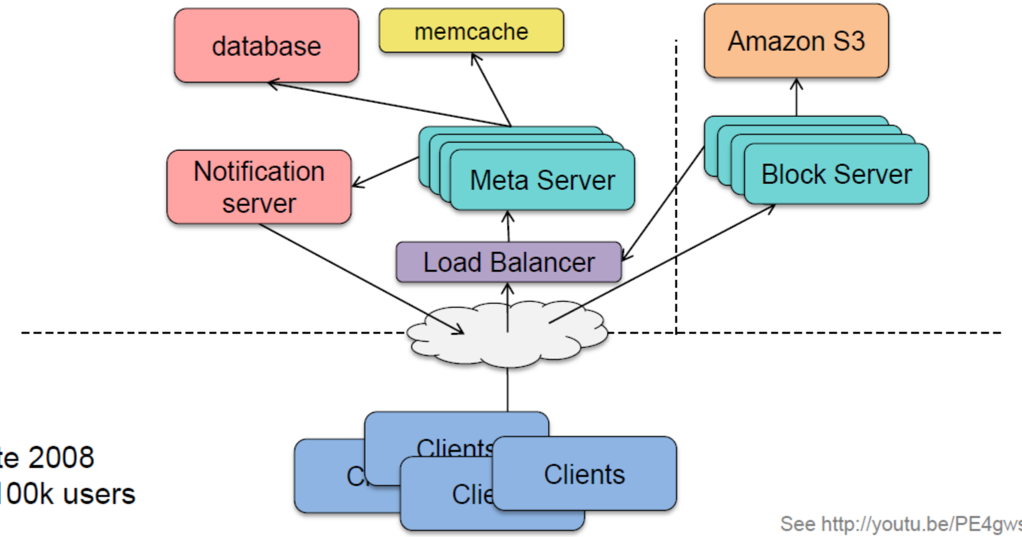


early 2008
50k users

See http://youtu.be/PE4gwstWhmc

Dropbox Architecture – v4

- Add more metaservers and blockservers
- Blockservers do not access DB directly; they send RPCs to metaservers
- Add a memory cache (memcache) in front of the database to avoid scaling

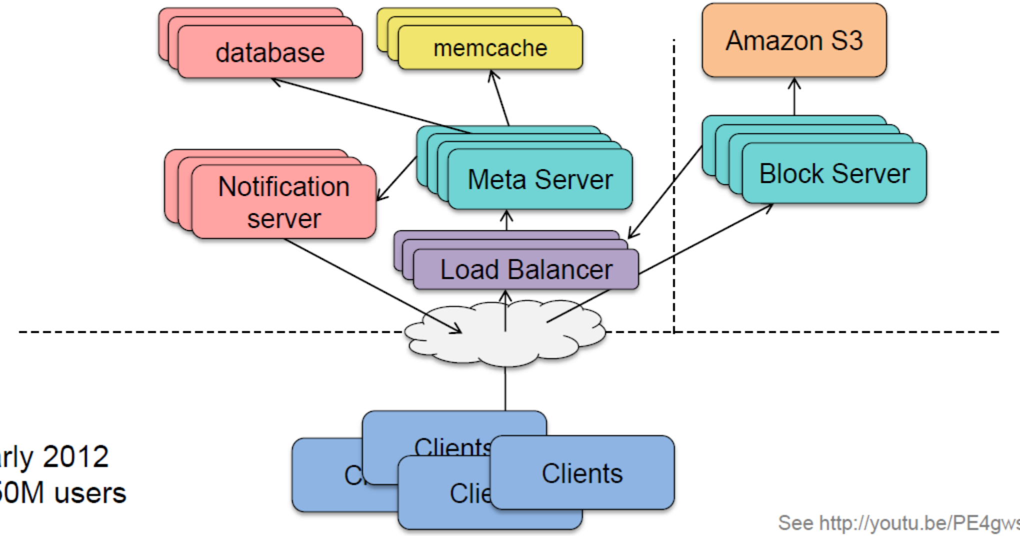


late 2008
~100k users

See http://youtu.be/PE4gwstWhmc

Dropbox Architecture – v5

- 10s of millions of clients – Clients have connect before getting notifications
- Add 2-level hierarchy to notification servers: ~1 million connections/server



early 2012
>50M users

See http://youtu.be/PE4gwstWhmc