Seafile - Scalable Cloud Storage System

Johnathan Xu
Seafile Ltd.

Agenda

- Seafile Introduction
- Feature Overview
- System Design & Performance
- Roadmap

What is Seafile?

VS

Seafile is a FAST, SCALABLE, and PRIVATE file sync & share solution

What can Seafile do?

- Fast and reliable file sync between cloud and devices
- Scales to millions of files, PB class storage
- High performance, light weight
- Productive file collaboration
  - Groups
  - File prview, discussion
  - Message and notification

Who are using Seafile?

- https://github.com/haiwen/seafile

  2400+ stars
- Estimated at least 100K users worldwide, most in Europe

JG|U

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

museum

KASPERSKY lab

Universities in Rhineland-Palatine (Germany)

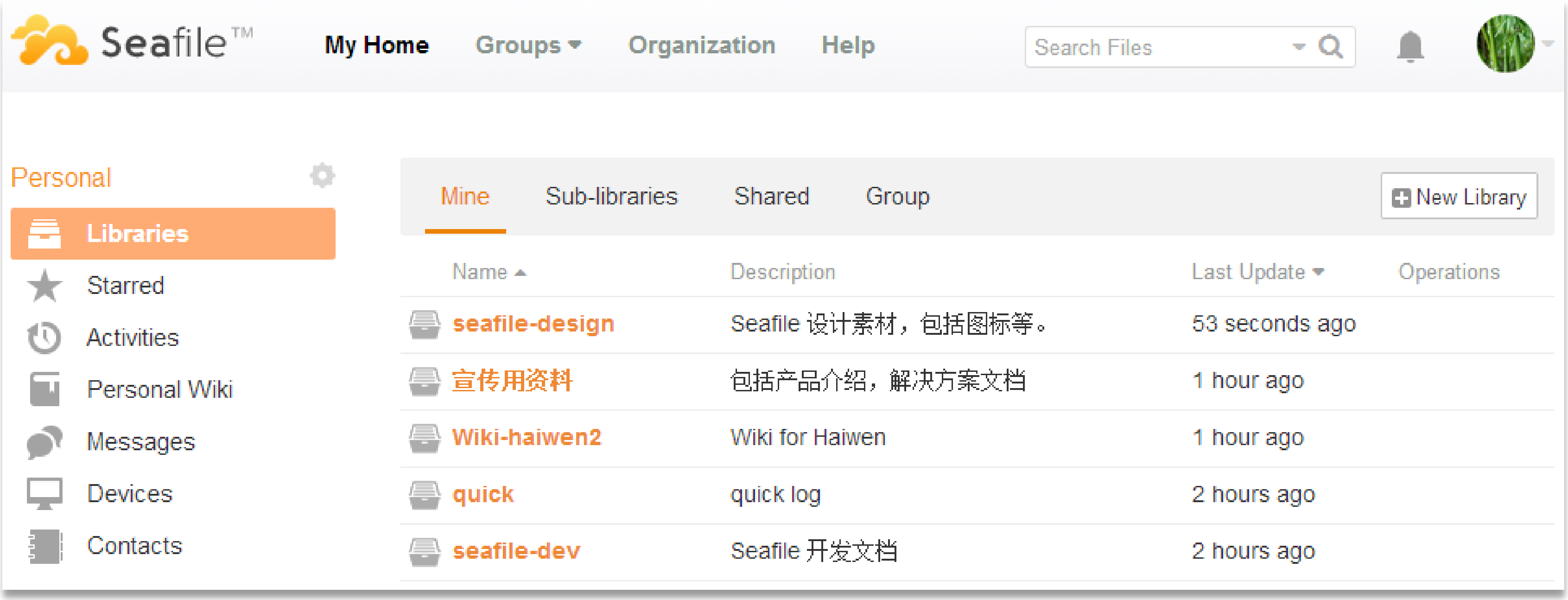
belgian royal institute of natural sciences

Agenda

- Seafile Introduction
- Feature Overview
- System Design & Performance
- Roadmap

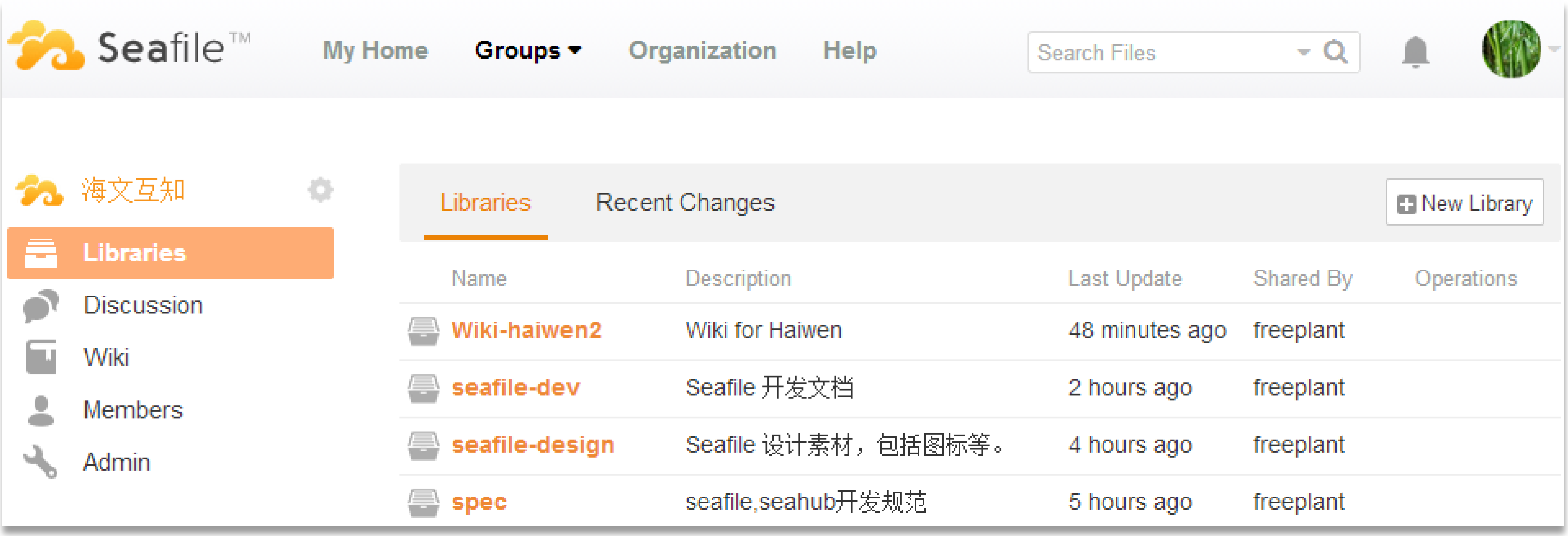
File Sync and Share

- Files are organized into Libraries
- Selective sync library to devices
- Sync with existing folder
- Client-side end-to-end data encryption
- Full platform support: Win, OSX, Linux, mobile
- Share to a person or a group
- Read-write and read-only share
- LDAP/AD integration



Name ▲	Description	Last Update ▾	Operations
seafile-design	Seafile 设计素材，包括图标等。	53 seconds ago	
宣传用资料	包括产品介绍，解决方案文档	1 hour ago	
Wiki-haiwen2	Wiki for Haiwen	1 hour ago	
quick	quick log	2 hours ago	
seafile-dev	Seafile 开发文档	2 hours ago	

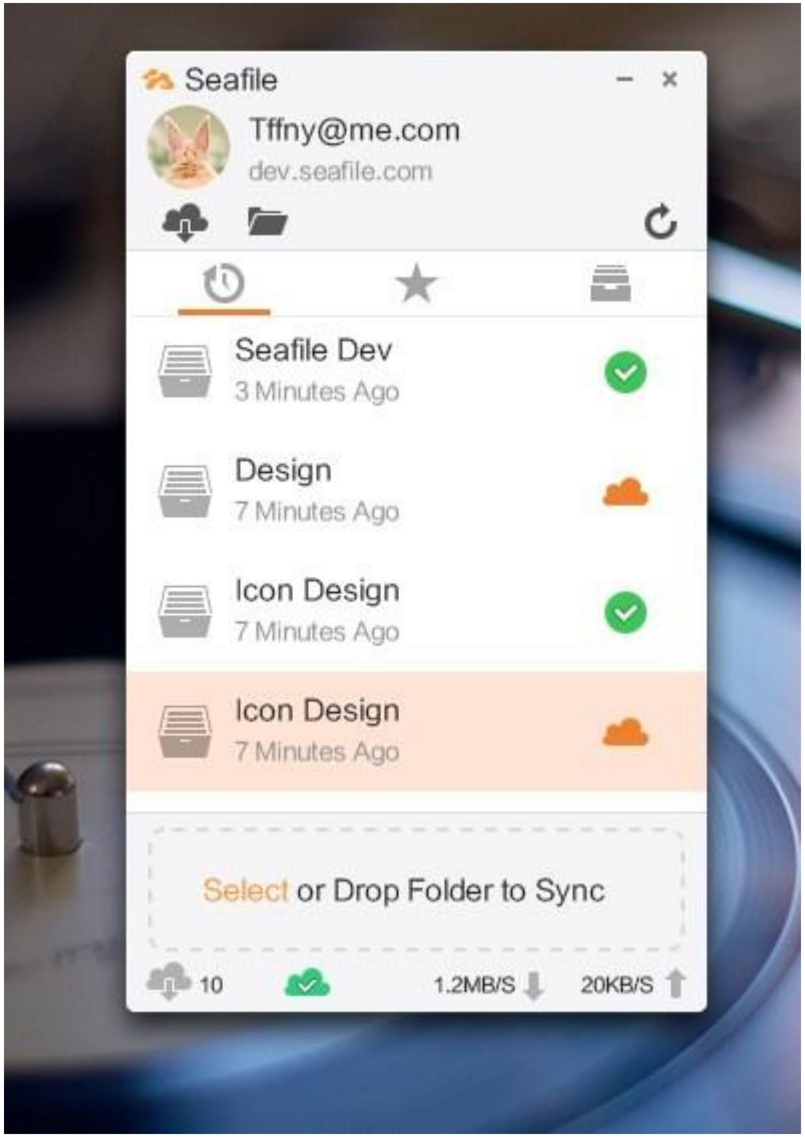
View all your libraries in the home page



Name	Description	Last Update	Shared By	Operations
Wiki-haiwen2	Wiki for Haiwen	48 minutes ago	freeplant	
seafile-dev	Seafile 开发文档	2 hours ago	freeplant	
seafile-design	Seafile 设计素材，包括图标等。	4 hours ago	freeplant	
spec	seafile,seahub开发规范	5 hours ago	freeplant	

All libraries shared to a group

Desktop Client



Seafile

Tffny@me.com

dev.seafile.com

Seafile Dev
3 Minutes Ago

Design
7 Minutes Ago

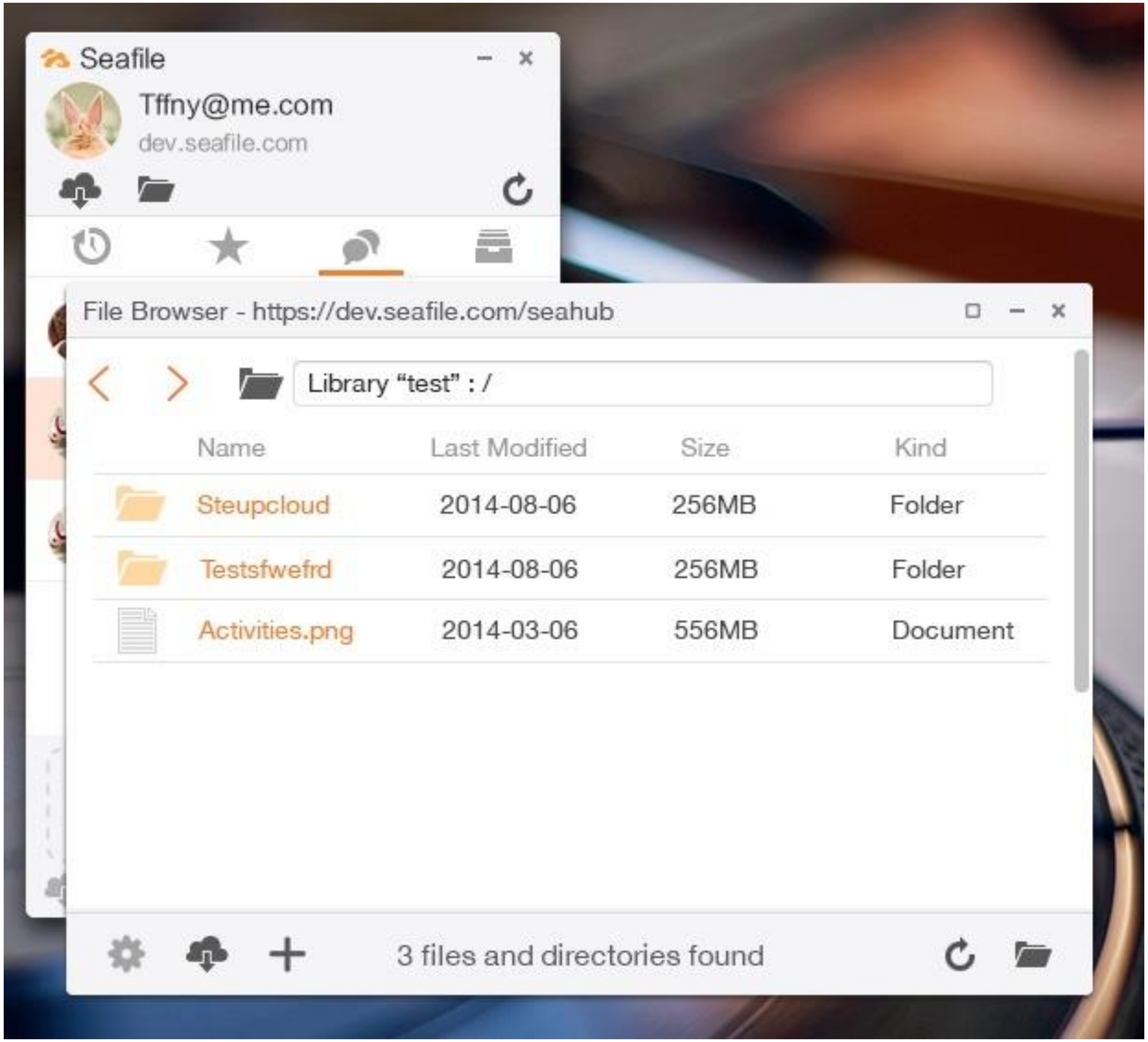
Icon Design
7 Minutes Ago

Icon Design
7 Minutes Ago

Select or Drop Folder to Sync

✓Selective sync library
✓Cloud file browser
✓Starred files
✓Notifications

Desktop Client



Seafile

Tffny@me.com

dev.seafile.com

File Browser - https://dev.seafile.com/seahub

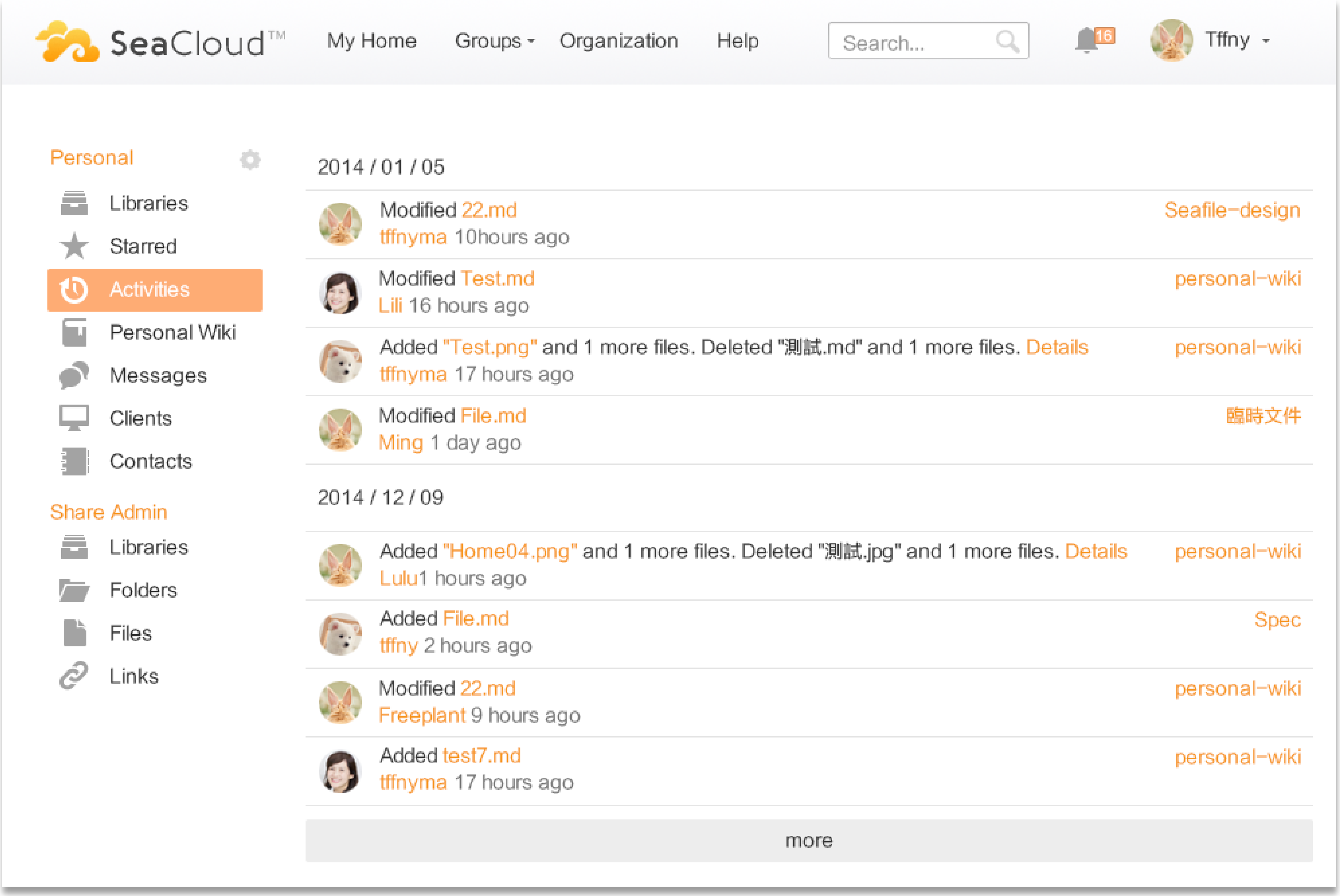
Library “test” : /

Name	Last Modified	Size	Kind
Steupcloud	2014-08-06	256MB	Folder
Testsfwefrd	2014-08-06	256MB	Folder
Activities.png	2014-03-06	556MB	Document

3 files and directories found

Collaboration

- File activities
- Group discussion
- File discussion
- Message notifications



2014 / 01 / 05

Modified 22.md — Seafile-design
tffnyma 10hours ago

Modified Test.md — personal-wiki
Lili 16 hours ago

Added "Test.png" and 1 more files. Deleted "測試.md" and 1 more files. Details — personal-wiki
tffnyma 17 hours ago

Modified File.md — 臨時文件
Ming 1 day ago

2014 / 12 / 09

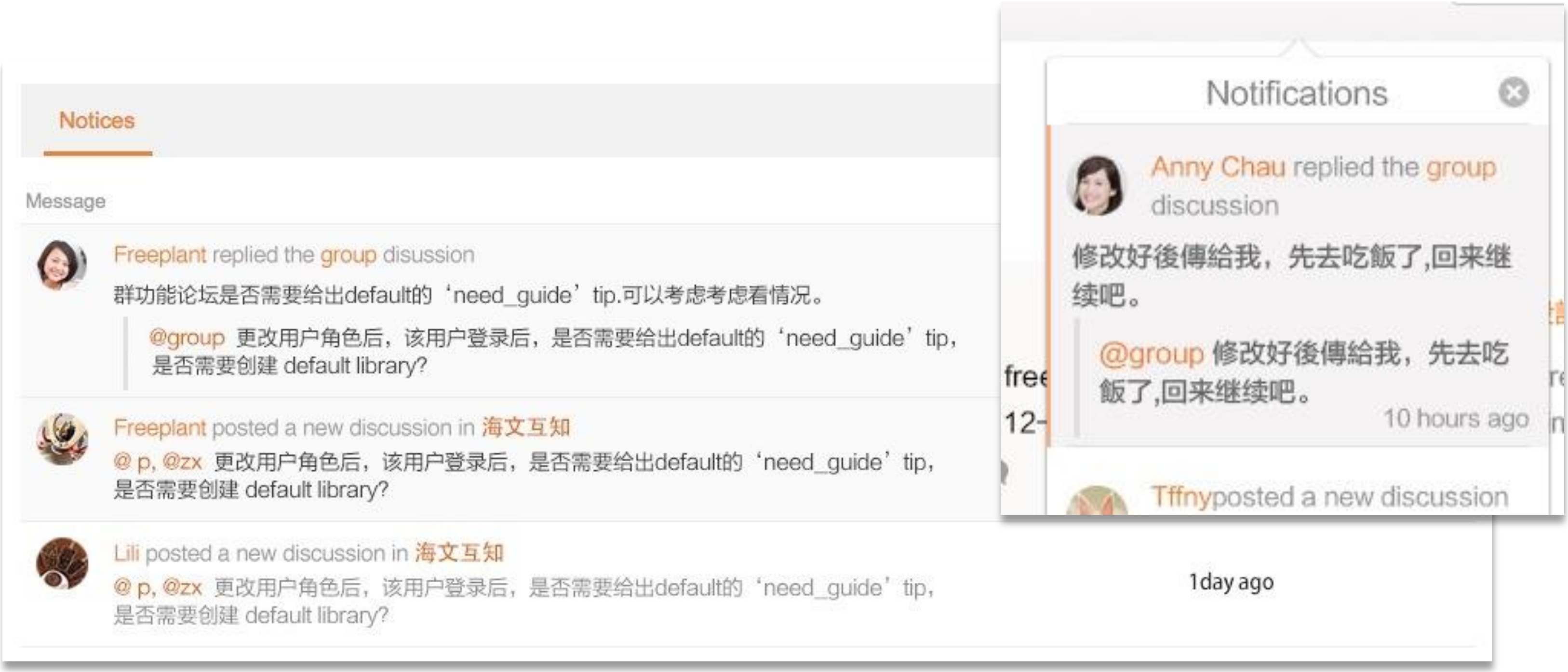
Added "Home04.png" and 1 more files. Deleted "測試.jpg" and 1 more files. Details — personal-wiki
Lulu1 hours ago

Added File.md — Spec
tffny 2 hours ago

Modified 22.md — personal-wiki
Freeplant 9 hours ago

Added test7.md — personal-wiki
tffnyma 17 hours ago

File Activities



Freeplant replied the group disussion

群功能论坛是否需要给出default的‘need_guide’tip.可以考虑考虑看情况。

> @group 更改用户角色后，该用户登录后，是否需要给出default的‘need_guide’tip，是否需要创建 default library?

Freeplant posted a new discussion in 海文互知

@ p, @zx 更改用户角色后，该用户登录后，是否需要给出default的‘need_guide’tip，是否需要创建 default library?

Lili posted a new discussion in 海文互知

@ p, @zx 更改用户角色后，该用户登录后，是否需要给出default的‘need_guide’tip，是否需要创建 default library?

1day ago

Anny Chau replied the group discussion

修改好後傳給我，先去吃飯了,回来继续吧。

> @group 修改好後傳給我，先去吃飯了,回来继续吧。

10 hours ago

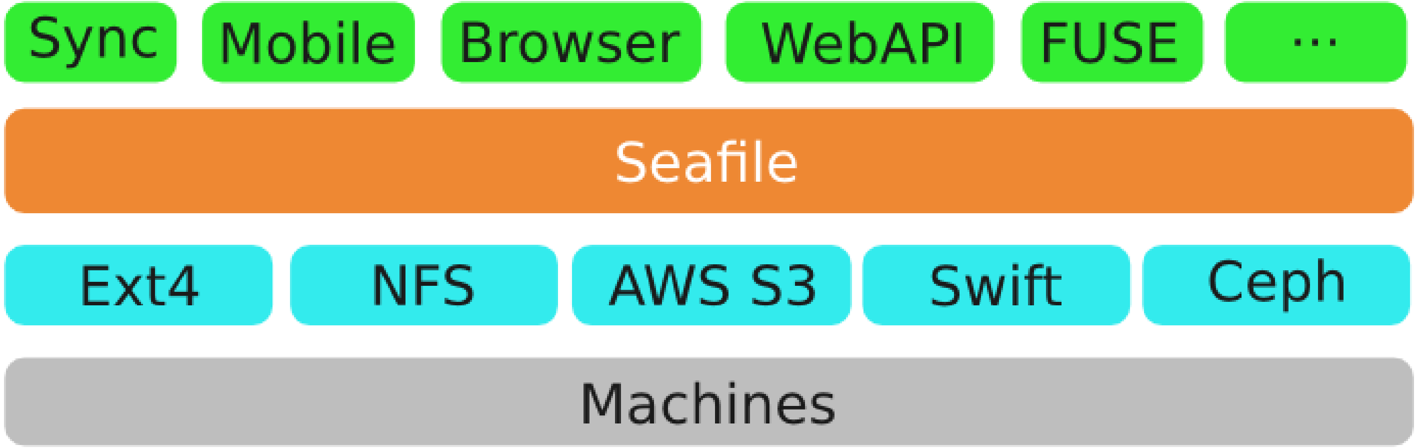
Tffnyposted a new discussion

Message Notifications

Agenda

- Seafile Introduction
- Feature Overview
- System Design & Performance
- Roadmap

Server Architecture



Sync Mobile Browser WebAPI FUSE …

Seafile

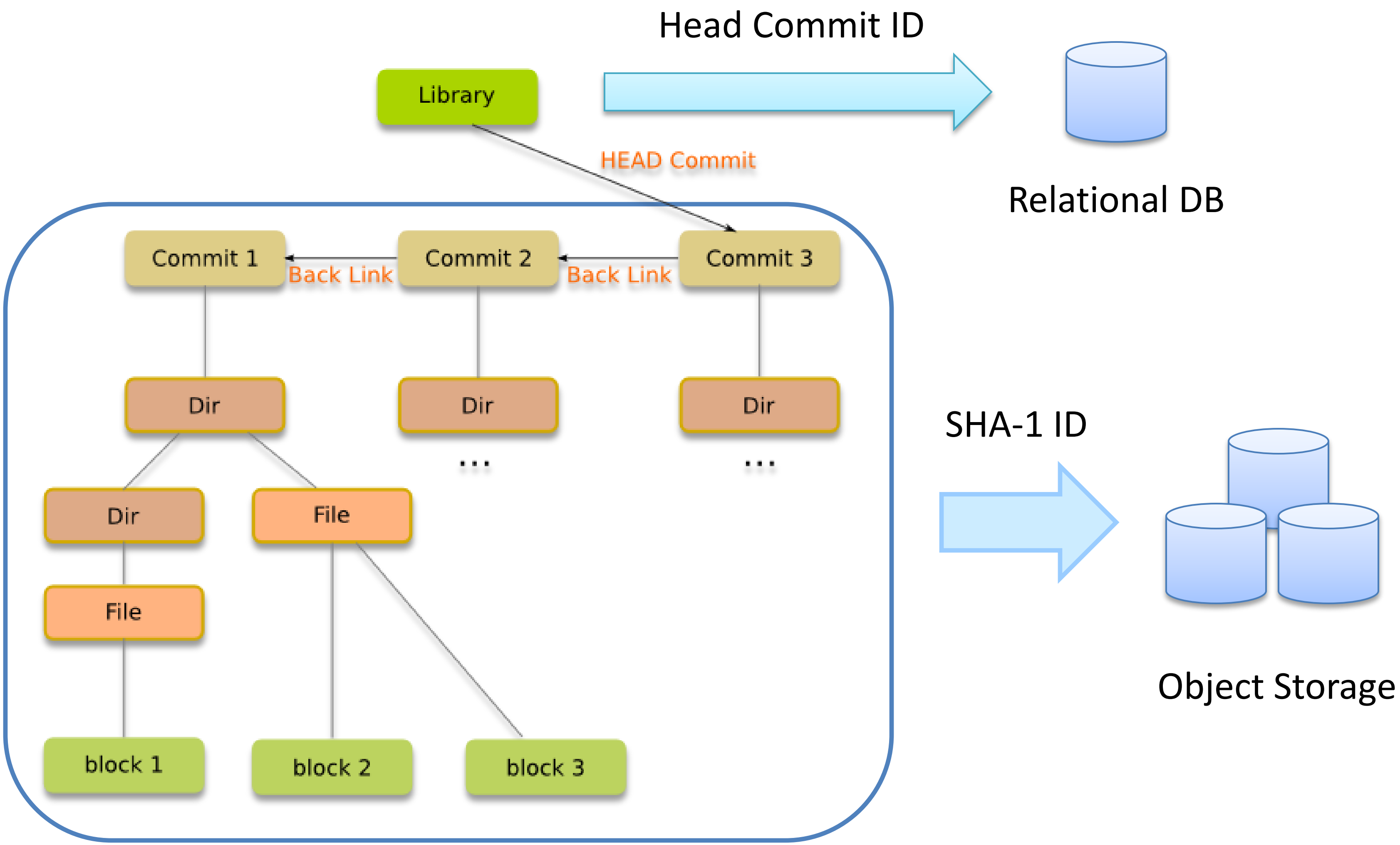
Ext4 NFS AWS S3 Swift Ceph

Machines

Seafile is a “file system” built on top of object storage

Non-POSIX, User space, Light weight

File System Design



Data model similar to Git

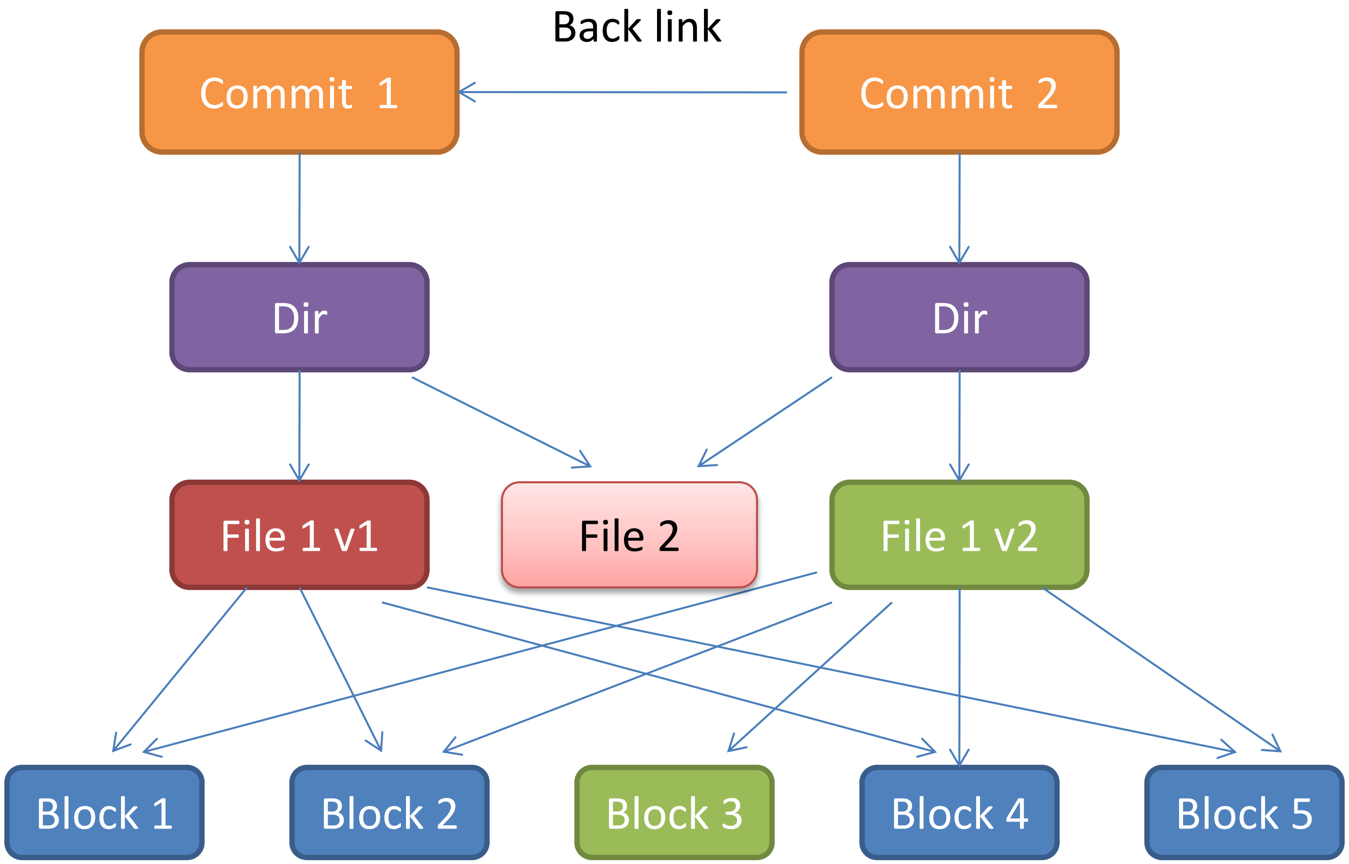
Design Advantage

- Object storage is more scalable than file system
  - Heavy DB + Filesystem v.s. Light DB + Object Storage
- No database bottleneck
  - Metadata is in object storage
  - Filesystem level versioning v.s. File-level versioning
- File system designed for syncing
  - Storage/Network deduplication
  - No upload/download limit, fast upload
- Backend daemons implemented in C

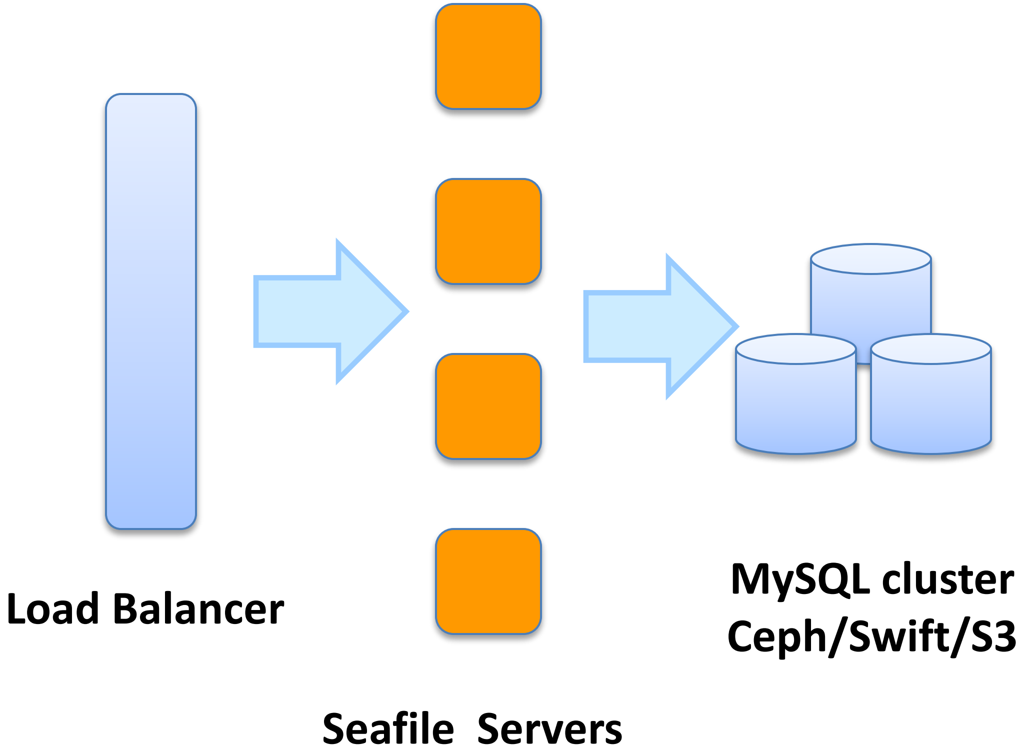
Deduplication

Dedup with Content Defined Chunking (CDC) algorithm

Only store/send delta between file system snapshots



Cluster Architecture



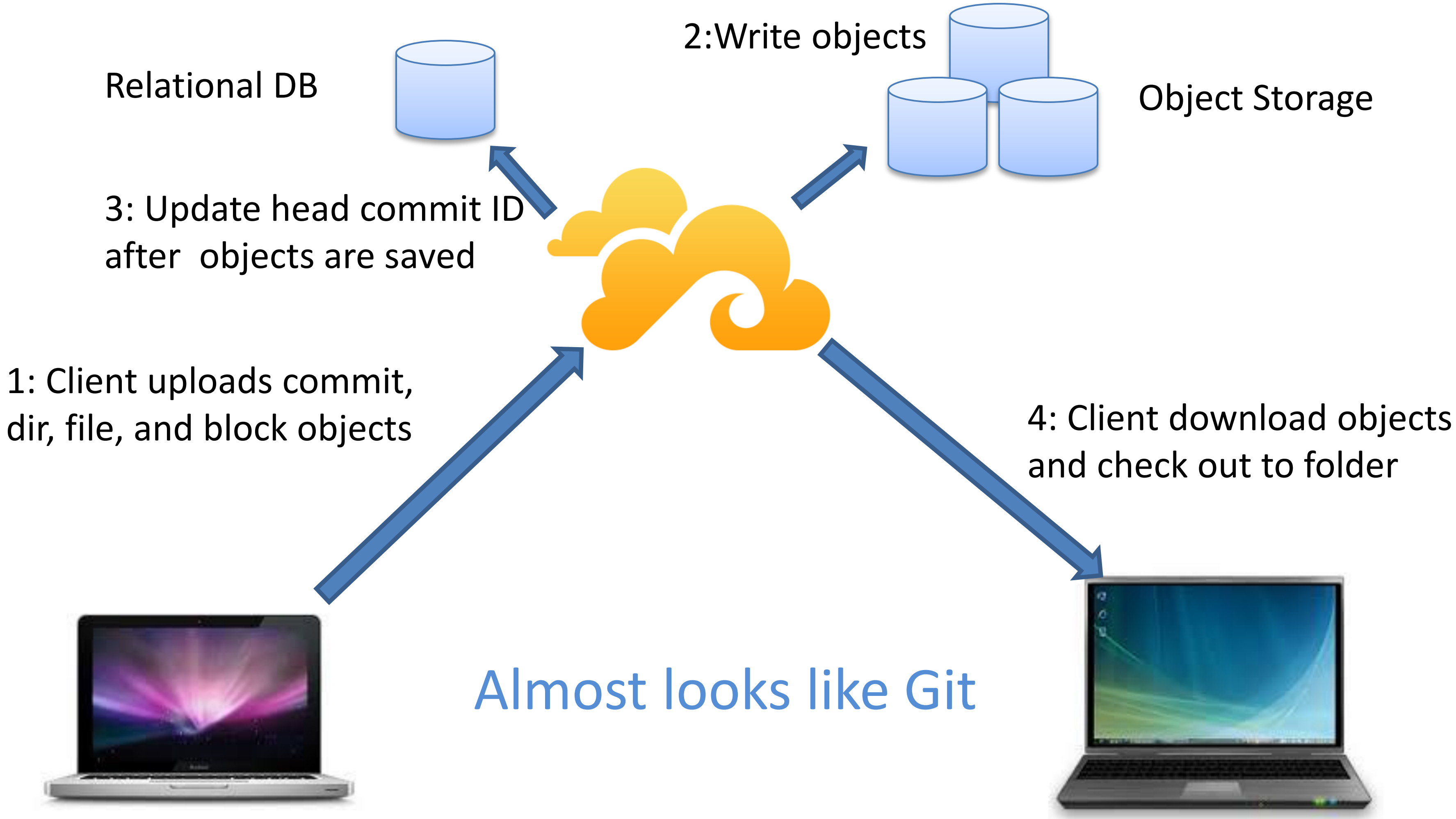
- Seafile server is stateless, scales horizontally
- Head commit ID and user-library mapping in MySQL cluster
- All data and metadata in object storage

Fast and Reliable File Syncing

- Detect file changes with OS mechanisms
- Low CPU usage on client and server side
- Sync 100K files easily and quickly
- No data transfer after rename/move
- Don’t send duplicate files. Delta dection.
- Handles conflicts
  - Concurrent updates
  - Case conflict: sync ABC.txt and abc.txt to Windows
- Never remove a file unless user does

Devil is in the details

How Syncing Works



Relational DB

2:Write objects

Object Storage

3: Update head commit ID
after objects are saved

1: Client uploads commit,
dir, file, and block objects

4: Client download objects
and check out to folder

Almost looks like Git

Syncing Performance

- Keep version info for the whole fs tree
  - Combine many file updates into 1 commit
  - A few database writes for a few K files
- Results
  - 1 core, 1GB memory VM server
  - 40K small files, ~20 files/s upload and download; single TCP connection; server CPU 2% - 5%
  - Big file, ~8MB/s upload and download in 100bps network; server CPU 50%

Agenda

- Seafile Introduction
- Feature Overview
- System Design & Performance
- Roadmap

Roadmap

- Sync & Share
  - File locking for better collaboration
  - Hierarchical access control within a library
- Auth integration
  - OAuth
  - Shibboleth
- Improve GUI responsibility with backbone.js

Conclusion

- Do one thing and do one thing well
  - Reliablity
  - Scalability
  - Performance

Choose any three ;-)

- Lightweight DB + Object Storage
- Git like data model, no client-side history
- Syncing model similar to Git, redesigned for auto syncing

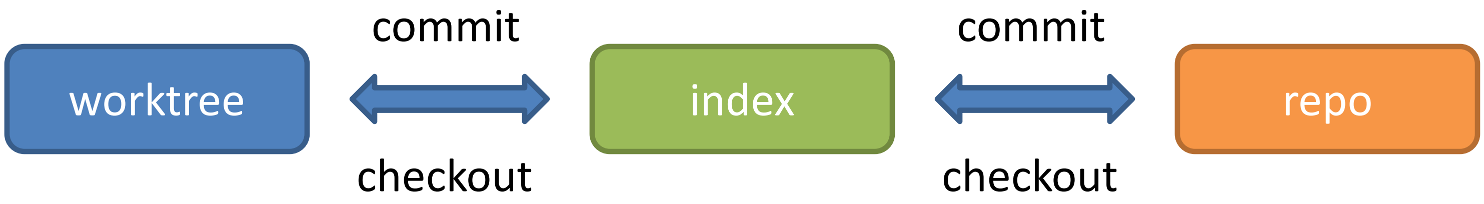
Seafile

Sharing & Collaboration

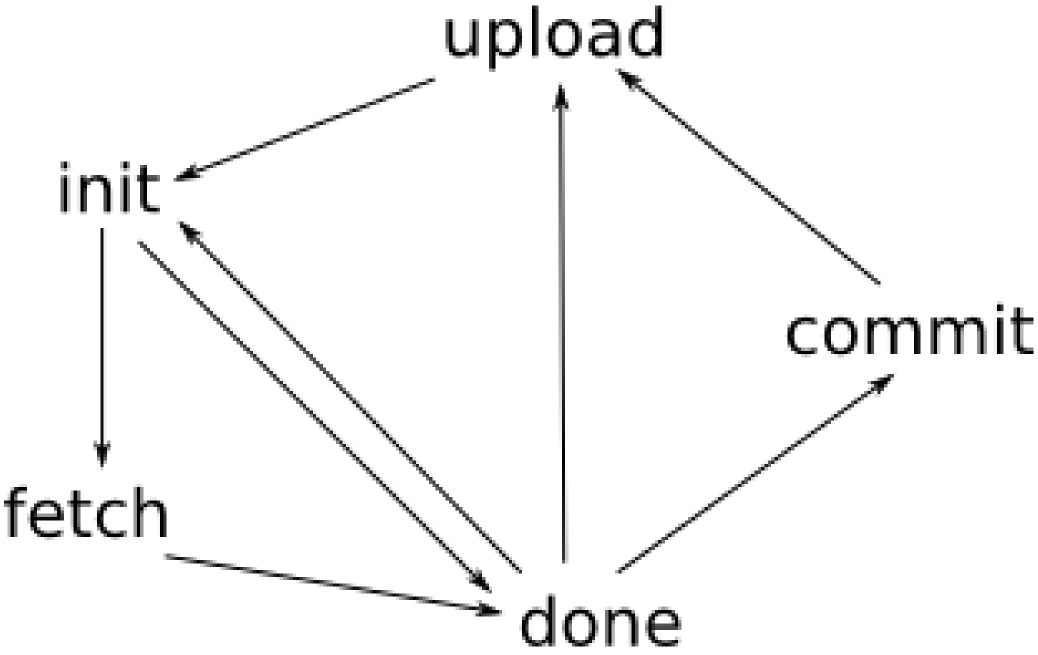
Thanks！

File Syncing Algorithm

- Client data 3 stages: worktree, index, repo
  - Worktree: user visible folder, one worktree per library
  - Index file: last modification time of each file in worktree
  - Repo: Internal representation of the latest fs tree for the library. Only have delta blocks.



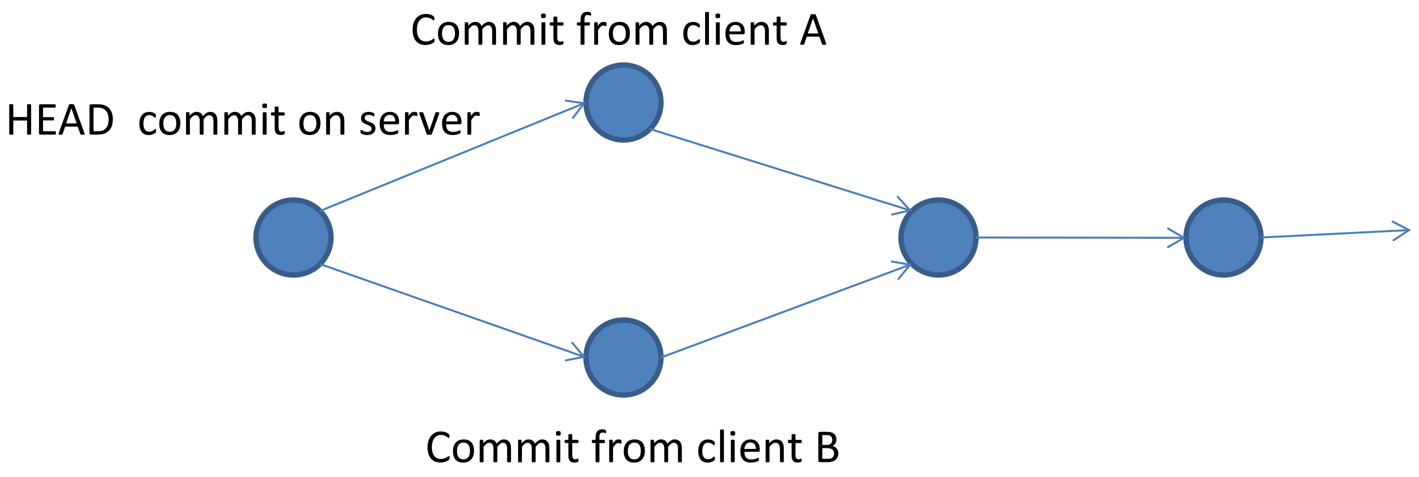
File Syncing Algorithm



Sync State Machine

File Syncing Algorithm

- Upload
  - Client creates new commit from batch of local changes
  - Diff between local repo and the cached server fs tree
  - After objects are uploaded, update server head commit ID in database
  - Server do merge on concurrent updates, resolve conflicts



File Syncing Algorithm

- Version Check（init）
  - Client caches server’s head commit ID
  - Compare with server every 30s, if not the same trigger download
- download
  - Server calculate update list with diff
  - Client download and apply the update to worktree
  - Update cached server head commit ID