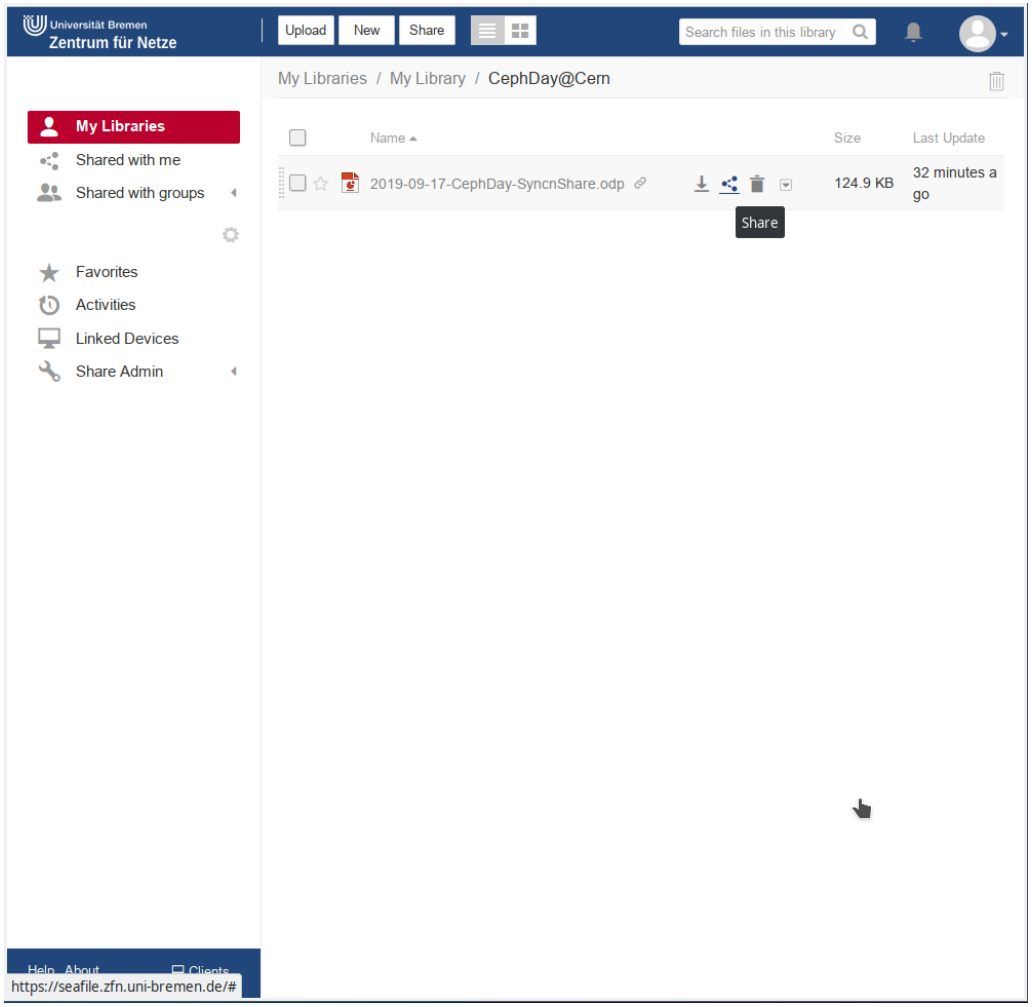
Scale out Sync & Share with Seafile on Ceph

Sönke Schippmann
Stefan Wichmann

Ceph Day @ CERN
17.9.2019

Just another sync & share service

- Private cloud service, like owncloud/ nextcloud and many others
- better performance
- lightweight daemon
- not a groupware
- community / pro edition



Seafile at University of Bremen

- 9.000 Seafile users
- 3.000 active users
- 21.000 libraries
- 34 TByte data
- 150 million files
- growing 25% per year

- 1 virtual machine
- 8 vCPUs (>50% idle)
- 32 GByte RAM
- file storage: ZFS on RBD
- migrating to S3 storage

Storage backends

- Backends supported (pro edition only):
  - file storage (default)
  - S3 (recommended for Ceph)
  - OpenStack Swift
  - Ceph / librados (faster, but no multiple backend support)
- multiple backends can be used in parallel (since 6.3)

Configure S3 storage

- 3 buckets needed:
  - seafile-commits
  - seafile-fs
  - seafile-blocks
- configure S3 user
- consider placement target by buckets or user
  - separate data and index pool gives better flexibility

Enable multiple storage backends

/seafile/conf/seafile.conf:

```
[storage]
enable_storage_classes = true
storage_classes_file = /seafile/conf/storage_classes.json
```

/seafile/conf/seahub_settings.py:

```
ENABLE_STORAGE_CLASSES = True
STORAGE_CLASS_MAPPING_POLICY = 'ROLE_BASED'
                             # or 'USER_SELECT'
```

storage_classes.json

```
[{"storage_id": "s3_storage",
"name": "S3",
"is_default": false,
"commits": {
  "backend": "s3",
  "use_https": true,
  "path_style_request": true,
  "bucket": "seafile-commits",
  "key": "yourkey",
  "key_id": “yourid",
  "host":"radosgw.example.com:7480" },
“blocks”: { ... }, “fs”: { ... }},
```

```
{"storage_id": "filesystem",
"name": "Filesystem",
"is_default": true,
"commits": {
  "backend": "fs",
  "dir": "/seafile/seafile-data"},
“blocks”: { ... }, “fs”: { ... } }]
```

Seafile caveats

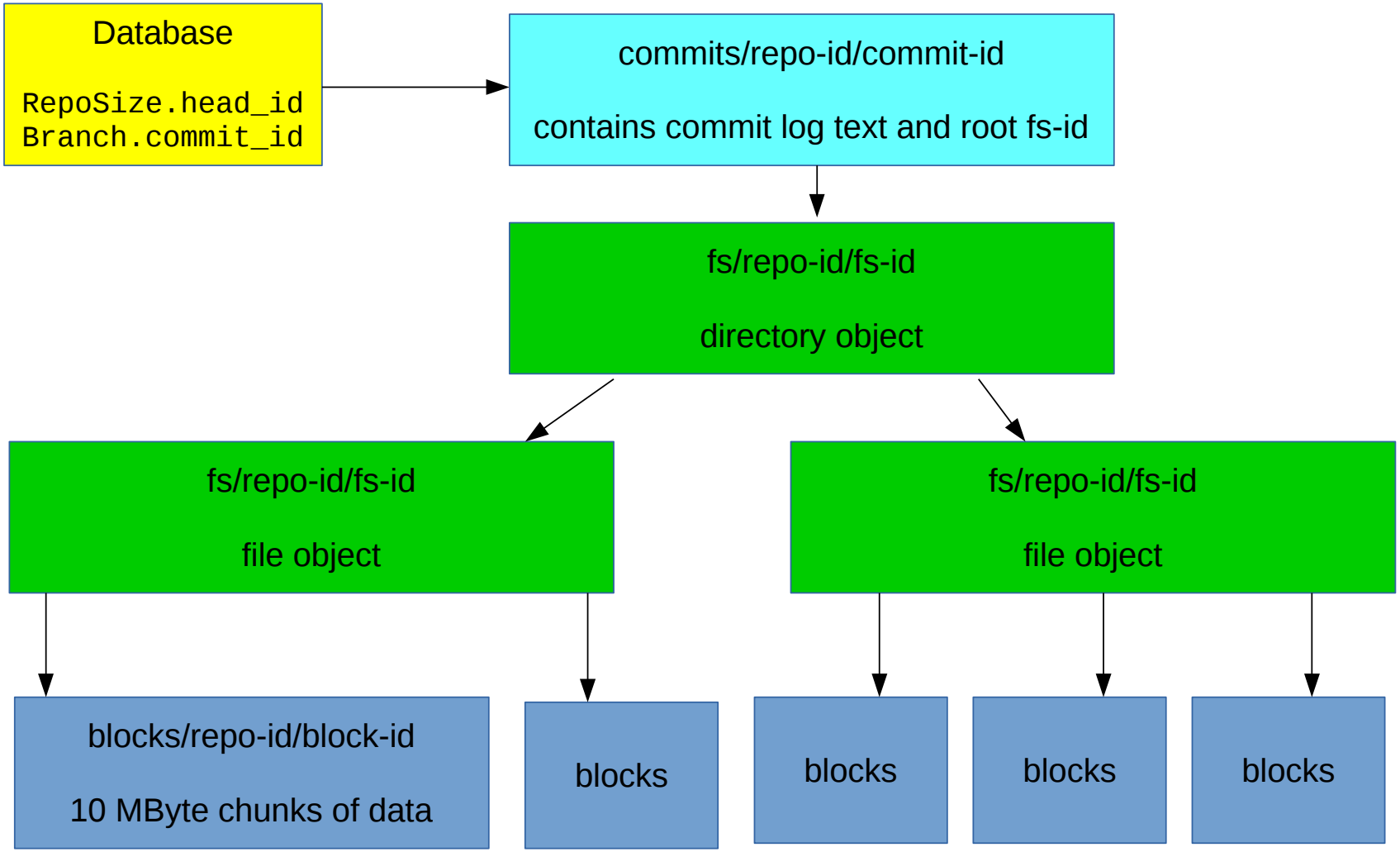
- Before changing default storage to S3, repos on file storage need to be configured in database:

```
insert into RepoStorageId (repo_id,storage_id)
select r.repo_id, 'filesystem' from Repo r left join RepoStorageId s

on r.repo_id=s.repo_id where s.repo_id is null
```

  Also with GarbageRepos, RepoTrash and (just to be sure) VirtualRepo
- In Seafile version 6.3: Bug in configuration code for S3 storage backend (fixed in 7.0.3, patch backport available)

Data model derived from Git



On file storage, object ID is split: 2 chars subfolder, 38 filename

Migrating data to S3

- Seafile offers a script for offline migration
- downtime not acceptable (2-4 weeks)
- storage file names translate well to S3 object names:

```
storage/commits/36125405-bba8-45d1-80f1-c4aeb351ecc1/
19/1cdd9f5372bd5ee186d8bc5a1ea23236061ccd

s3://seafile-commits/36125405-bba8-45d1-80f1-c4aeb351ecc1/
191cdd9f5372bd5ee186d8bc5a1ea23236061ccd
```

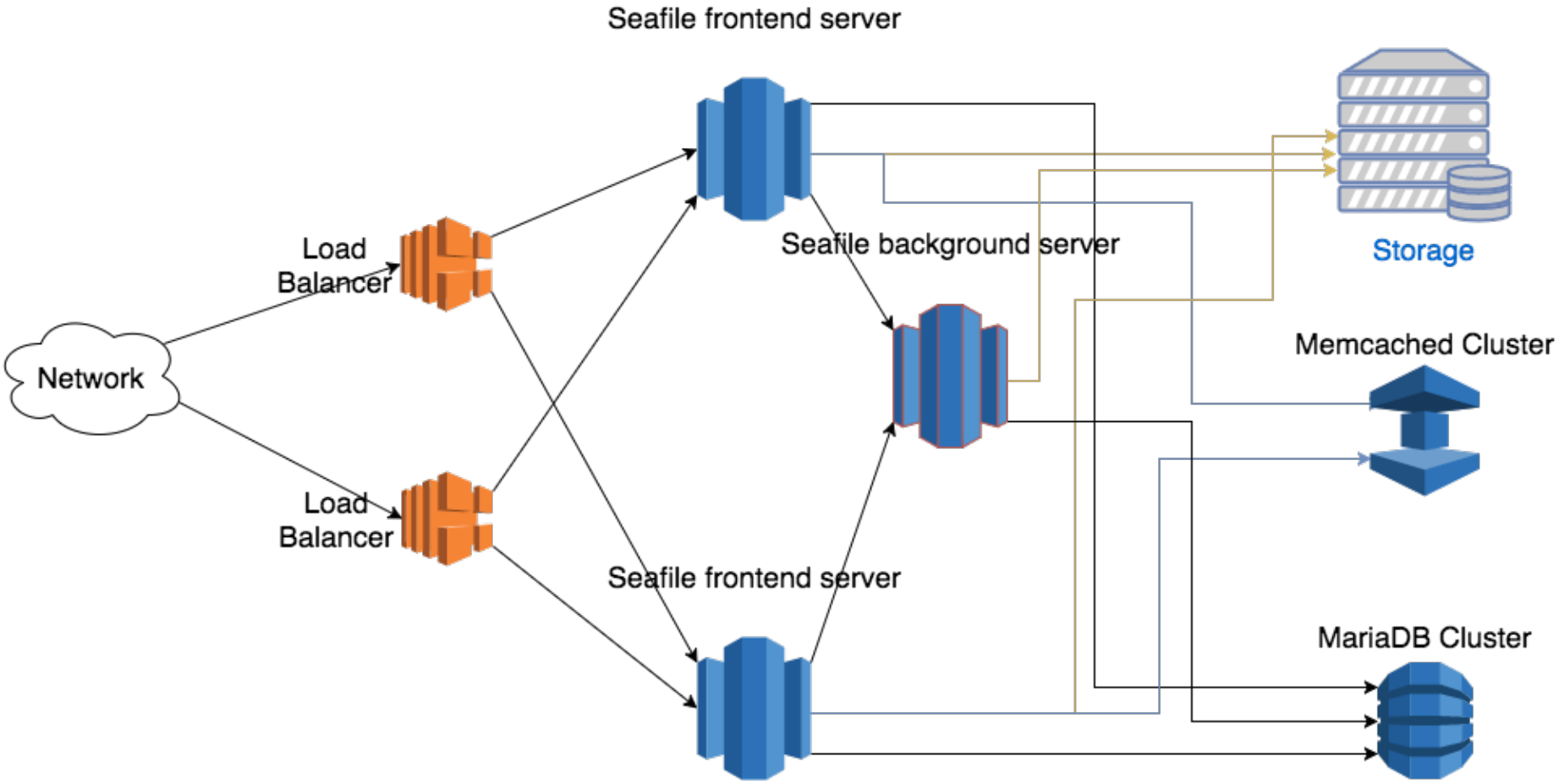
- online migration scenario (scripts available):
  - copy all files, repo by repo
  - change repo's Storage-ID in Database

Backup and Restore

- make database dump
- copy Branch.commit_id
- copy objects from S3 to a backup filesystem
  - use filestorage layout
- Seafile could directly use backup file storage

- restore per repository:
  - create new repo with WebAPI
  - copy all files with new repo ID
  - change head ID/commit ID to value of Branch.commit_id for that backup

Running a Seafile cluster



Thank you

- More in Seafile's manual: https://manual.seafile.com
- Our slides, patch, migration and backup scripts: http://unihb.eu/4eNoRHGd

Questions?

schippmann@uni-bremen.de