CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it

Prototyping a file sharing and synchronisation platform with ownCloud

Jakub T. Moscicki
Massimo Lamanna

CERN IT-DSS

CHEP 2013 - Amsterdam

Content

- Background & context of the cernbox project
- Details on core functionality of owncloud
- Testing owncloud
- Outlook

The origins of the **cernbox** project

- We need a competitive alternative to Dropbox for CERN users
  - Reasons
    - SLAs: availability, confidentiality
    - integration into IT infrastructure
    - archival & backup policies
  - The scale of the problem is unknown but we have some indications
    - 4500 distinct IPs in DNS from cern.ch to *.dropbox.com (daily...)
  - We also want to adapt to user expectations
  - We manage large-scale online-storage systems
    - …and we can leverage on them

EOS (RAW)	CEPH (RAW)	Other services
~62 PB	~3.5 PB	~30 PB

Bulk of disk storage operated by IT/DSS

Gateway to the future

- A **unified platform** integrated with physics data storage
- **Federated** "dropbox" service for HEP community
  - … and possibly in wider science
- Novel ways for supporting specialized **scientific workflows**
  - based on a common sharing and syncing platform
- Novel ways of **delivering home directories** in the virtualized IT environment
  - local folder replica lives within the VM snapshot

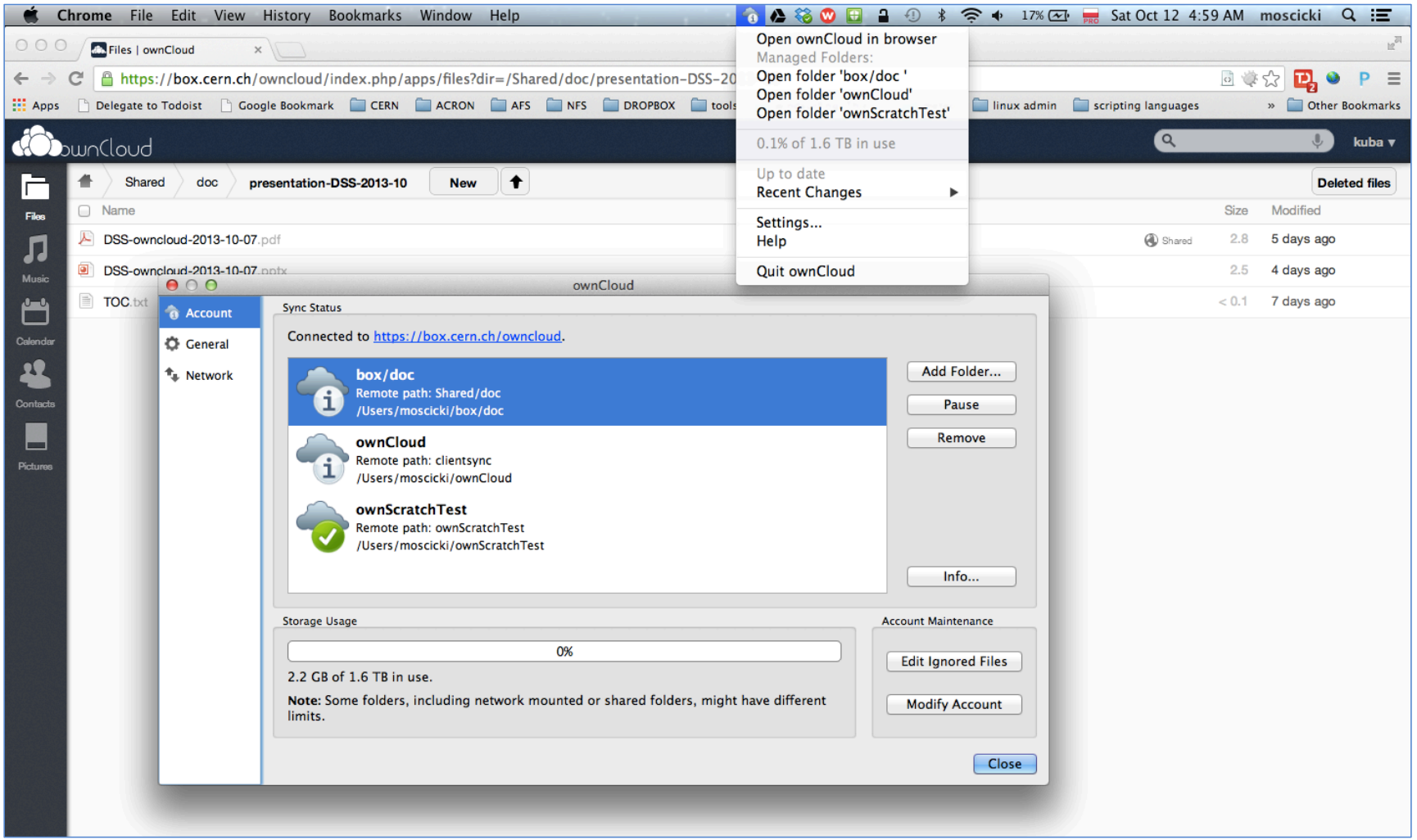
…however, first we need to positively address the classic Dropbox use-case…

OwnCloud Evaluation

- OSS “Dropbox” Market
  - not-yet-mature
  - but products ramp up with quality and features
- Why ownCloud?
  - It has a vision matching our needs
  - It has the required functionality
  - It has an extensible architecture for future use-cases
  - It is open-source
- After a market survey we decided to seriously test ownCloud and provide feedback to the company and to the community

What OwnCloud provides?

- Interface
  - cross platform web/mobile/desktop access
  - desktop integration (drag/drop, notifications,…)
  - web 2.0
- Core functionality
  - Syncing of folders
  - Folder/file sharing with ACLs, expiry date
  - Public (hashed) links with optional password protection
  - Anonymous upload folders (hashed links)
  - Currently NO support for user-defined groups
    - Admin creates groups
    - Needed: integration with IT infrastructure (external groups)


Chrome File Edit View History Bookmarks Window Help
Sat Oct 12 4:59 AM moscicki
Open ownCloud in browser
Managed Folders:
Open folder 'box/doc '
Open folder 'ownCloud'
Open folder 'ownScratchTest'
0.1% of 1.6 TB in use
Up to date
Recent Changes
Settings...
Help
Quit ownCloud
ownCloud
Shared
doc
presentation-DSS-2013-10
New
Deleted files
Name
Size
Modified
DSS-owncloud-2013-10-07.pdf
DSS-owncloud-2013-10-07.pptx
TOC.txt
5 days ago
4 days ago
7 days ago
Account
General
Network
Sync Status
Connected to https://box.cern.ch/owncloud.
box/doc
Remote path: Shared/doc
/Users/moscicki/box/doc
ownCloud
Remote path: clientsync
/Users/moscicki/ownCloud
ownScratchTest
Remote path: ownScratchTest
/Users/moscicki/ownScratchTest
Add Folder...
Pause
Remove
Info...
Storage Usage
0%
2.2 GB of 1.6 TB in use.
Note: Some folders, including network mounted or shared folders, might have different limits.
Account Maintenance
Edit Ignored Files
Modify Account
Close

File transport protocol

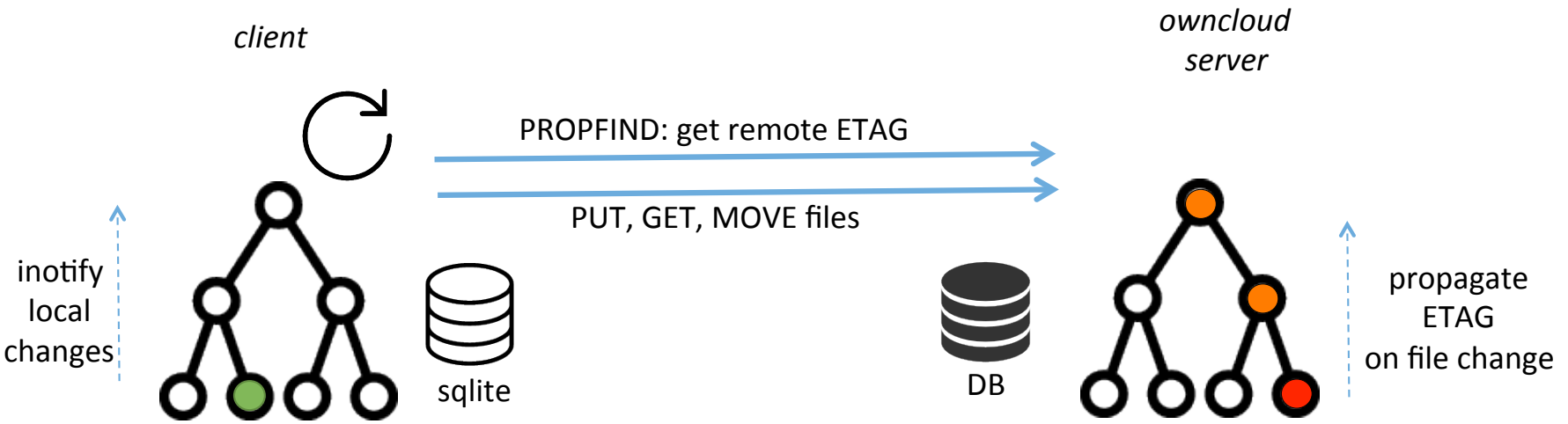
- WebDAV (extension to HTTP with XML body)
  - OwnCloud Server is RFC 2518 compliant
  - Protocol is HTTP with XML body so it is bloated
    - Basic metadata query for a file ~0.5KB
    - Compresses well: metadata for 1000 files ~16KB
  - Some good points
    - Integration with other web-services
    - Desktop browsers: OSX/Finder, …
    - Simply curl/wget to GET, PROPFIND, PUT, DELETE, MOVE
    - Fuse mount (davfs2)
- HTTP POST/GET
  - 2GB file limit for upload currently but it is removed in newer PHP version
- Sync client
  - Chunked upload (10MB chunks)
  - Extension: OC-CHUNKED header attribute

Server storage layout

- Simple and transparent, including trashbin and versions

```
<user>
    |-cache
    |
    |-files
    |---dirA
    |---dirB
    |------dirC: hello.txt
    |
    |-files_versions
    |------dirC: hello.txt.v1380894998
    |
    |-files_trashbin
    |---files
    |------dirA: byebye.txt.d1381078676
    |---versions
    |------dirA: byebye.txt.v1380891350.d1381078676
    |            byebye.txt.v1380891050.d1381078676
```

How sync works



icons: http://www.visualpharm.com

- Notes:
  - ETAG is a standard HTTP header for cache control
    - ETAG is a unique identifier generated by the server
  - No file diffs over the wire

Testing principles

- **Community edition** latest server (5.0.11) and client (1.4.1)
- Automatic testing of critical functionality
  - testing of new releases
  - testing in local/changed environment
  - product-agnostic collection of test cases and test ideas
- Share our work
  - We plan to move our test toolkit to github.com/opensmashbox
- Test plan
  - ***Core logics testing***
  - Stress testing
  - Scale testing
  - Operational scenarios

Core logics testing

Fundamental checks: *break it!*

- check functionality, trigger file conflicts, check consistency of behaviour
- symlinks, hardlinks, device files, characters: | \ : " < > % ? ' <space> *
- "extreme" conditions
  - How many files can we put in a single directory
  - How deep may be the directory tree
  - In parallel: Client A: remove directory, Client B: add files to this directory

Field testing: *see if system is reasonable under "normal" conditions*

- "realistic" actions
  - e.g. keep on updating a file somewhere in a directory tree
- "realistic" file size distributions
  - from CERN AFS home directories but random content
- "realistic" directory tree
  - /afs/cern.ch/user/m/moscicki (30K files, 8K directories)
- automatic verification of integrity of files and directories (md5)
- check propagation of changes between different clients of the same user

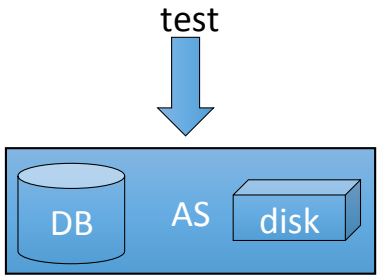
Interactive analysis: *tap into the server framework, protocol, db, apache, …*

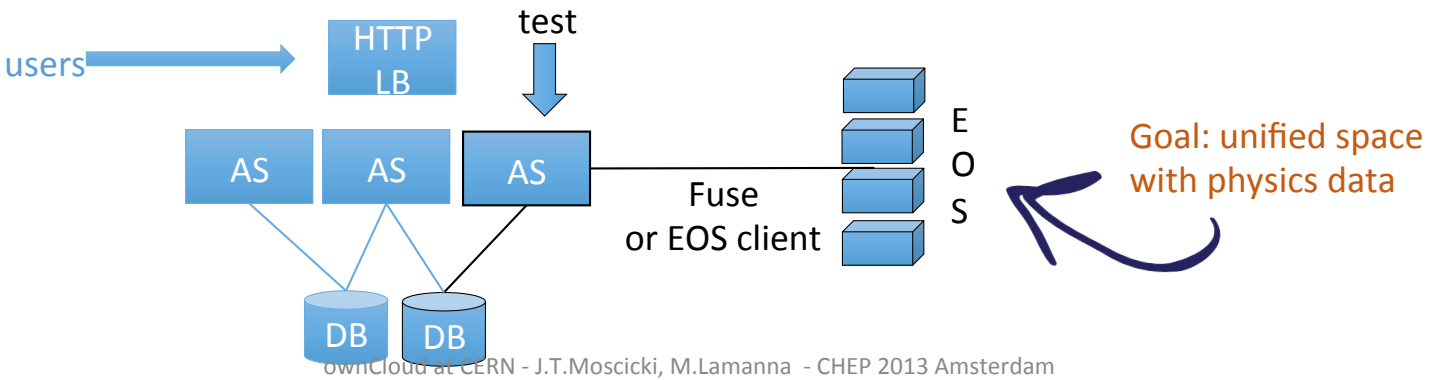
Scale / stress testing *(in progress)*

- Horizontal scaling
  - How many concurrent clients may a single server support?
    - Idle users, Active users
- System scaling
  - How many files/directories/users?
    - "10% prototype": 1 K active users, 10K dormant users, 100TB data
- Performance testing and tuning
  - How efficient we are in sending and receiving files
  - Many tuning opportunities
    - MySQL/Innodb/indexes/cluster, memcached
    - PHP version/accelerators,
    - Apache tuning, …
- Operational scenarios
  - Network access cut
  - Server reboot
  - Client killed
  - Database lost,…

Test infrastructure

- Baseline server setup
  - RHEL 6, Apache, MySQL, local disk
  - should work out-of-the-box
  - “ultimate reference”
- Intended final configuration





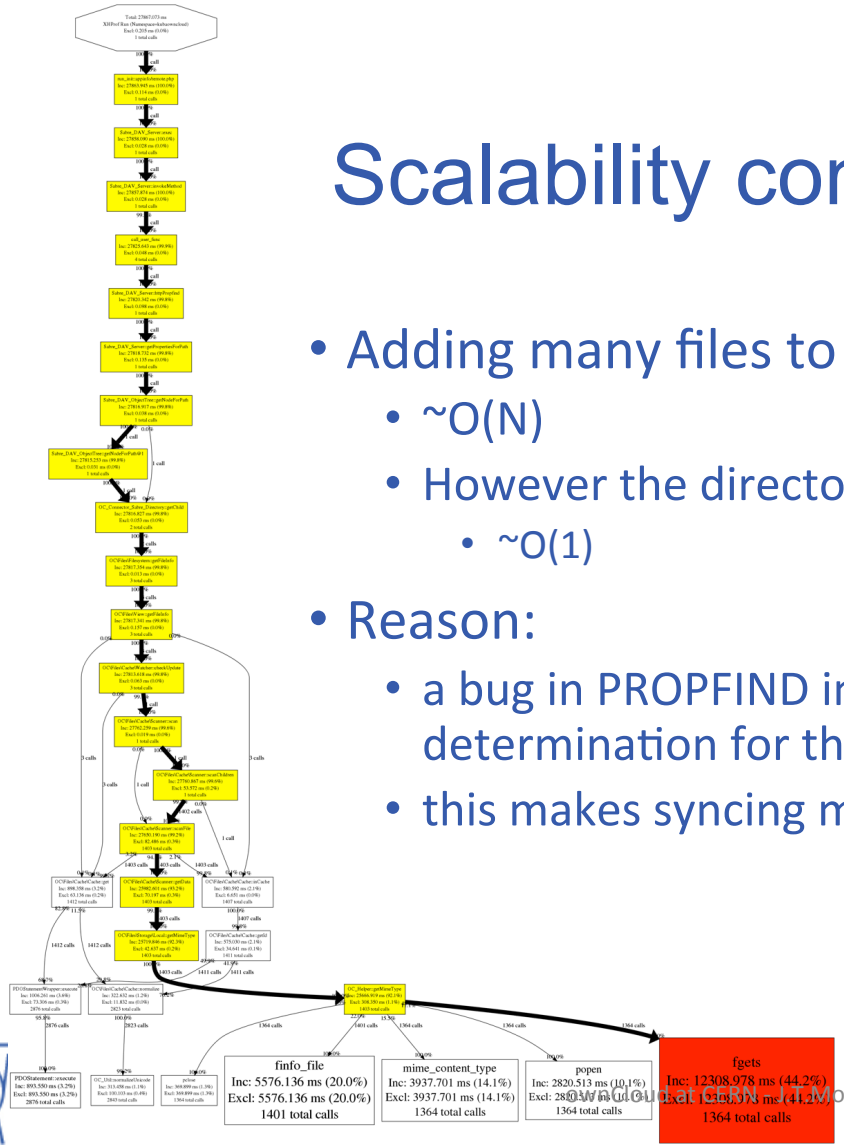
Upsides

- Integrates smoothly with our LDAP(S)
- ETAG propagation works as expected
  - Parallel directory trees are really independent
  - Large number of files in the ETAG propagation chain does not affect efficiency
  - ~O(log N)
- In field testing we did not manage to see corrupted files
- Conflict files are(mostly) created in expected ways
  - However it may be problematic for some applications
- Decent WebDAV streaming for large files
  - Example: 400MB file on my desktop
    - http/upload: ~40MB/s, http/download: ~100MB/s
    - https/upload: ~25MB/s, https/download: ~60MB/s

Downsides

- Parallel sync with chunking could corrupt data ✓
  - now fixed in devel: temporary chunk files should encode session ids
- Client tends to traverse directory tree out of order ✓
  - now fixed in devel: asymptotically correct but very suboptimal
- Problems with more than 50 levels of directories
  - should give a clear feedback to the user if limits exceeded
- Shared folders:
  - ETAG propagation problems if not at top-level
- Syncing rates are low (~1-5 files/s)
  - callgraph analysis of the framework shows opportunities for large improvements
  - additionaly this may also be due to server tuning or older PHP version
- Random glitches of the desktop clients
  - Sometimes hangs (wireless networks)
  - Updates of client versions not smooth

Scalability concerns for large folders



- Adding many files to a single directory does *not* scale
  - ~O(N)
  - However the directory tree hierarchy *does* scale
    - ~O(1)
- Reason:
  - a bug in PROPFIND implementation triggers the mime-type determination for the entire folder
  - this makes syncing many files in a single directory very slow

Bug reporting

- We report all bugs on github
  - …and discuss them with owncloud CEO directly…
  - some issues are fixed already (not release yet, needs retesting)

Expectations?

open #1069 request for enterprise use: possibility to manually recover locally deleted files 4 days ago
closed #5099 IE10 support in owncloud 5? 4 days ago
open #1068 RC of the ocsync is generally unreliable (which is not helpful for testing) 4 days ago
open #1067 problem syncing deep directory trees 4 days ago
open #1066 out of order sync of directories causes errors when removing directories 4 days ago
open #1065 out of order sync of directories causes errors when adding new files 4 days ago
open #5098 syncing of shared folders is broken 4 days ago
open #5089 PUT of chunked file inefficient if many chunks 5 days ago
open #5084 PROPFIND bug prevents larger number of files per directory (and causes connection timeouts) 5 days ago
closed #1032 data corruption: parallel upload of the same file may lead to corrupt
closed #1014 data loss: local files deleted if sync upload is only partially success
open #1013 own cloud client lags behind on MacOSX (last sync log time ago) 1

ownCloud

- Fix an issue that caused endless syncing when encountering permissions issues.
- Other minor fixes

2.2.3 6/13/2013 comments

- Fix an issue on Mac OS X 10.4 which causes files to disappear after uploading.
- Fix a rare issue on Windows which causes Dropbox to endlessly sync files with long paths

2.2.2 6/12/2013 comments

- Fix a rare issue that causes Dropbox to change directory names to all lowercase.
- Use a consistent image for photo notifications.

2.2.1 6/7/2013 comments

- Fix an issue preventing Dropbox from starting up on 10.4 and 10.5

2.2.0 6/5/2013

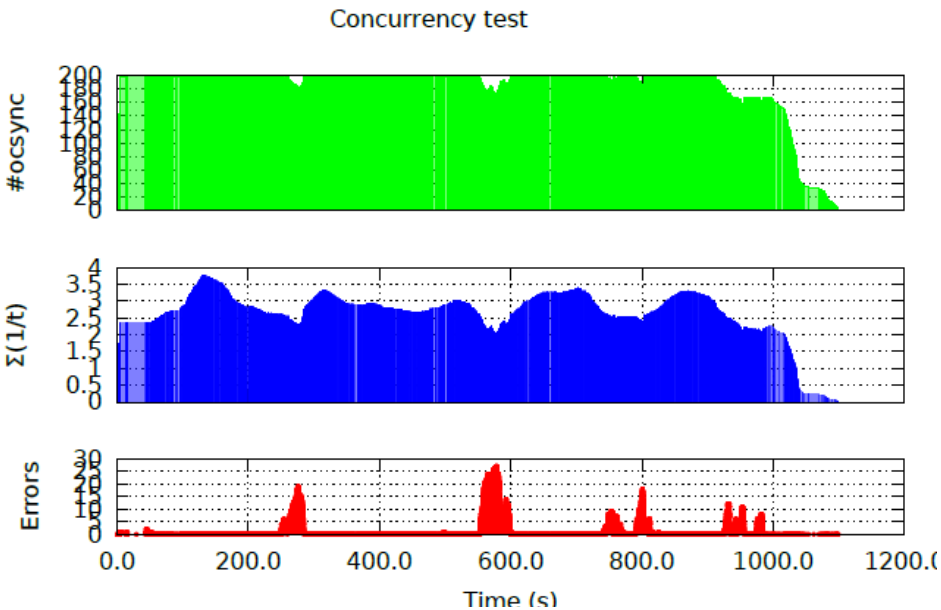
- New notifications badge

Feedback on architecture

- Generic framework server-side
  - Open and extensible, API for adding new apps
  - A challenge for core business performance
  - Decouple view from data model
    - Whether current coupling incidental or purposeful
  - We see many opportunities for performance improvement and better scalability
- Documentation of sync model is badly needed
- ETAG: calculate it ?
  - concerns about hashing content (mounting secondary storage)
  - what about hashing metadata (size+mtime)?
  - requires careful thinking of client-side mtimes, ACLs, …

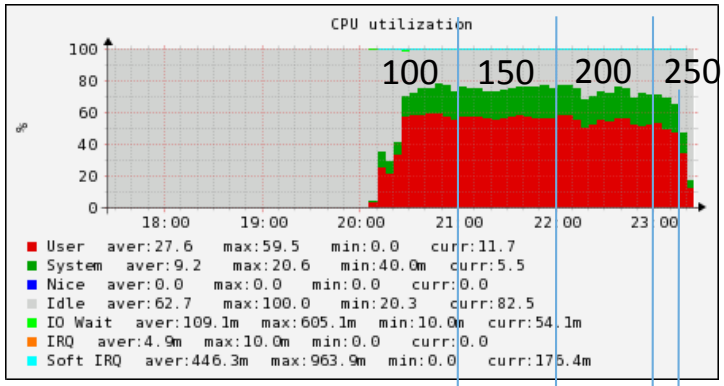
Horizontal scalability *(preliminary)*

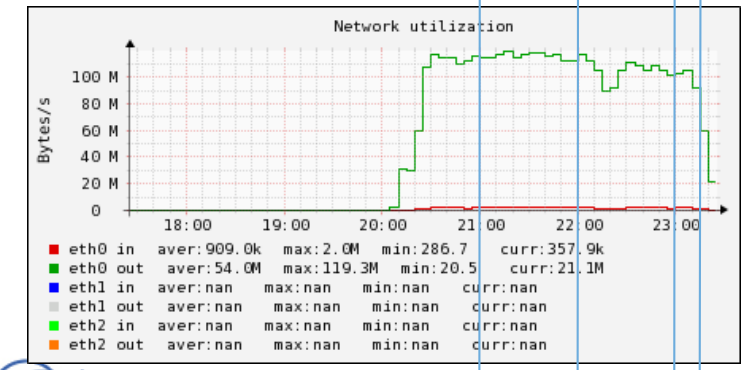
- Test configuration (field testing)
  - 100 users account, 50 VM clients
  - sync sessions run in parallel
  - data: 1 directory with 100 files
- Scenarios
  - Dormant accounts (“idle” system footprint)
  - Download, upload files
  - Modify and sync
- Measurement
  - number of parallel sync sessions
  - speed of the system
  - errors: incomplete transfers
    - Not ciritical by design
      (will catch up on the next sync session)

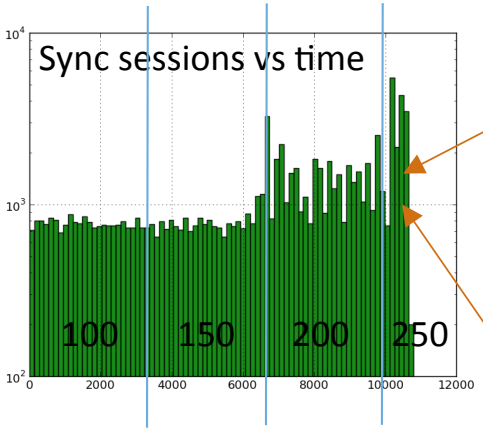


~100 MB/s measured on the server NIC

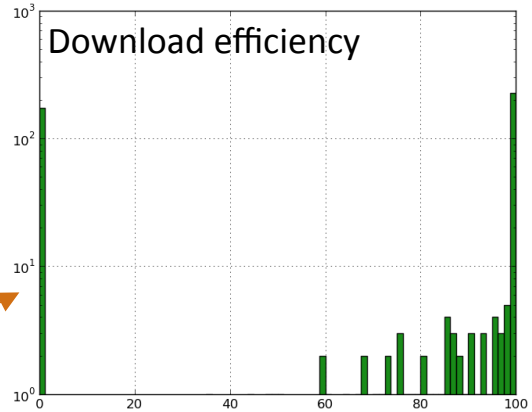
Server load *(preliminary)*

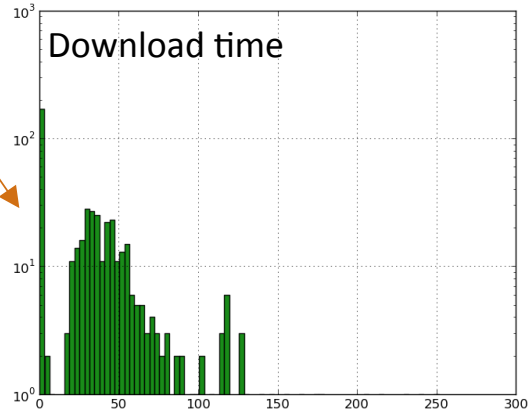






overload → refuse client





Testing EOS storage for cernbox

- 1000 test user accounts
  - … and ramping up → 1K “active”, 10K “idle”
- 5 million test files
  - … and ramping up
- 2 TB of test data
  - … and ramping up → 100TB

What others do?

- ETHZ: public beta service (using owncloud enterprise)
  - limited use-case: memory stick replacement
  - 5GB user quota
  - 1900 users, 500GB data since 28 June
  - Backend storage: SONAS EMC
- Universities starting beta service
  - TUB
  - MPI
- owncloud.com:
  - scaleout tests with commercial vendors (~250K users)
  - CEPH integration under discussion

Evaluation

- OwnCloud
  - Excellent vision and great functionality
  - Young product in (very) active development
  - Challenging task: support different semantics of local OSes and FSes
- Strong points
  - Platform integration, web interface, mobile clients rapidly improving
  - Open architecture
  - LAMP stack: well known and used in industry
  - Developers are responsive and give attention to our feedback
- Weak points
  - Maturity of the product – rapid development and many changes at all levels
  - Challenges to address in the core framework and sync client

What comes next

- Beta service at CERN
  - we are confident that our concerns will be addressed in due time
- Your feedback is important
  - on the beta service at CERN (as a user)
  - on your local deployment experience (as a service manager)
- We report all issues on github
  - and discuss our concerns with owncloud.com
  - we have positive feedback
- We continue to get user feedback on the product within HEP
- We will share our test toolkit on github

Summary

- Dropbox-like service on premise
  - solving an important “corporate” issue for our Organization
  - exciting new ideas and opportunities for the future
- Dropbox-like service has a huge potential for HEP community
  - we want to see it fly at CERN
- We have a testing framework for QA assessment
  - Collaboration with owncloud
  - Contribution to/from the community
  - Extend the test coverage
  - Learn from others
- Your feedback?
  - Beta service at CERN
  - Your deployment experience

Questions? Comments?