
CERN

EOS-FUSE in CERN's DevOps Infrastructure

Dan van der Ster, CERN IT Storage Group
daniel.vanderster@cern.ch

EOS Workshop
2 February 2017

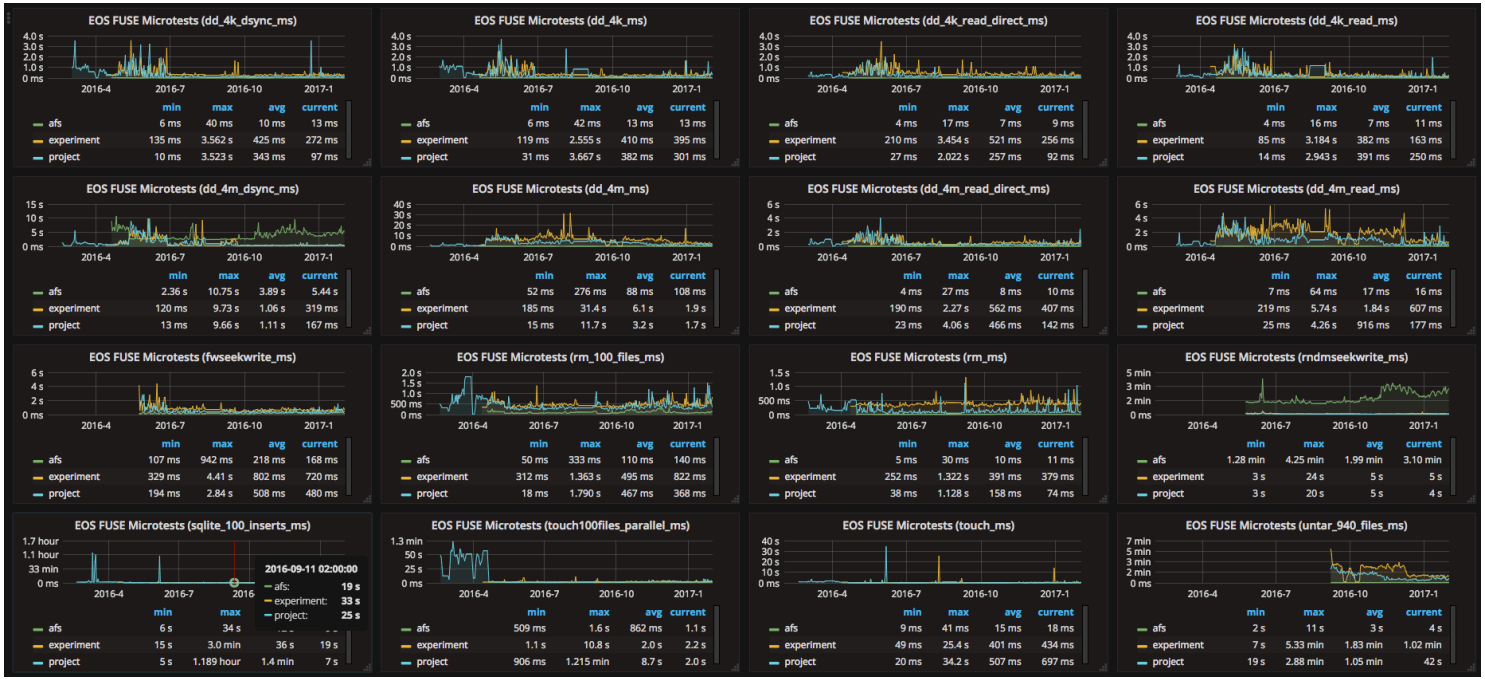
Putting /eos in production at CERN

- This is a (short) story about how we prepared eos-fuse to be widely mounted at CERN
  - *lxplus*: our interactive login workstations
  - *lxbatch*: our very large LSF/HTCondor batch system
- We started in early 2016:
  - Quantifying its performance
  - Packaging for deployment at scale
  - QA testing and deployment

Quantifying /eos Performance

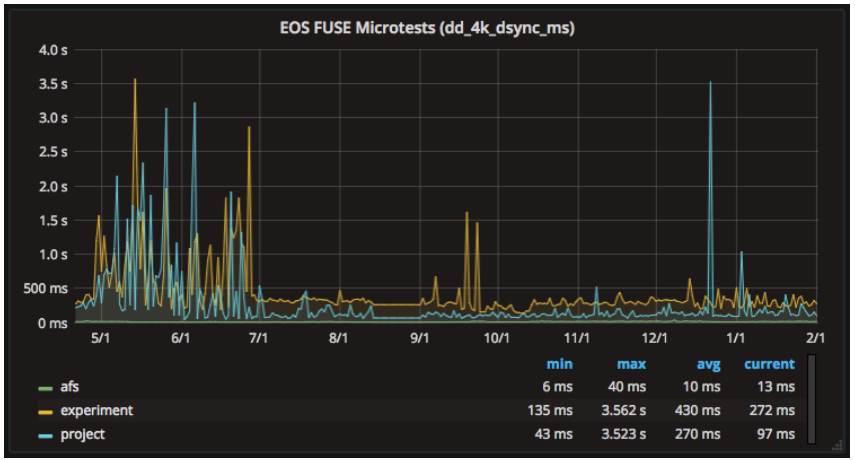
- Mounting /eos widely adds new “small file”, “small IO” requirements
  - Previously eosd had been used for streaming IO and large files
- Success of /eos depends on fast interactive performance:
  - Create files quickly (tar xf …), unlink files quickly (rm -rf …)
  - 4k random IO (dsync, odirect, buffered, …)
- We created a small test suite (so called “microtests”) to evaluate /eos vs our current $HOME service (OpenAFS)
  - https://filer-carbon.cern.ch/grafana/dashboard/db/eos-fuse-microtests (sorry, this is probably an internal CERN website)

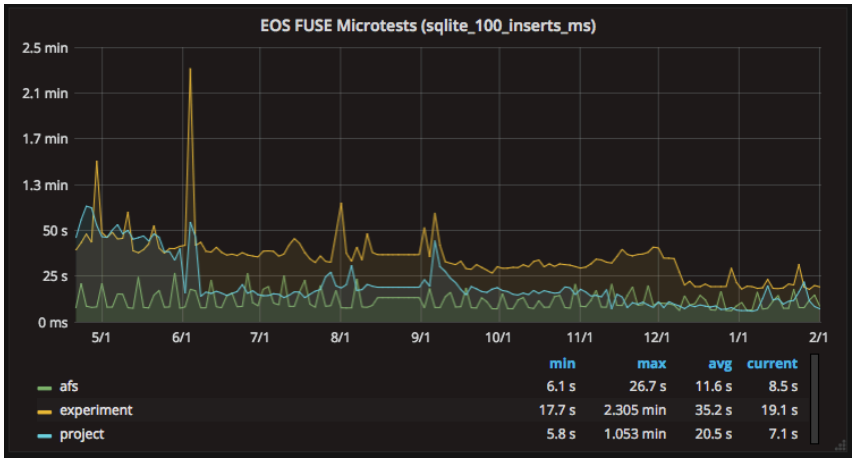
EOS-FUSE Microtests




```
dd_4k_dsync_ms
dd_4k_ms
dd_4k_read_direct_ms
dd_4k_read_ms
dd_4m_dsync_ms
dd_4m_ms
dd_4m_read_direct_ms
dd_4m_read_ms
fwseekwrite_ms
rm_100_files_ms
rm_ms
rndmseekwrite_ms
sqlite_100_inserts_ms
touch100files_parallel_ms
touch_ms
untar_940_files_ms
untar_ms
```

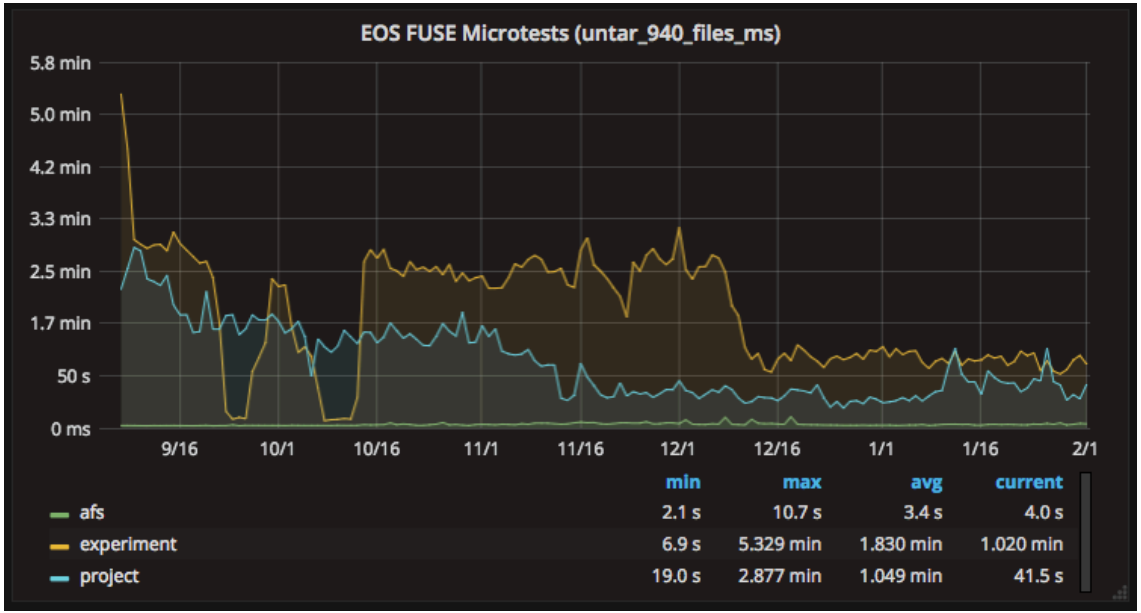
Microtests: Example speed up





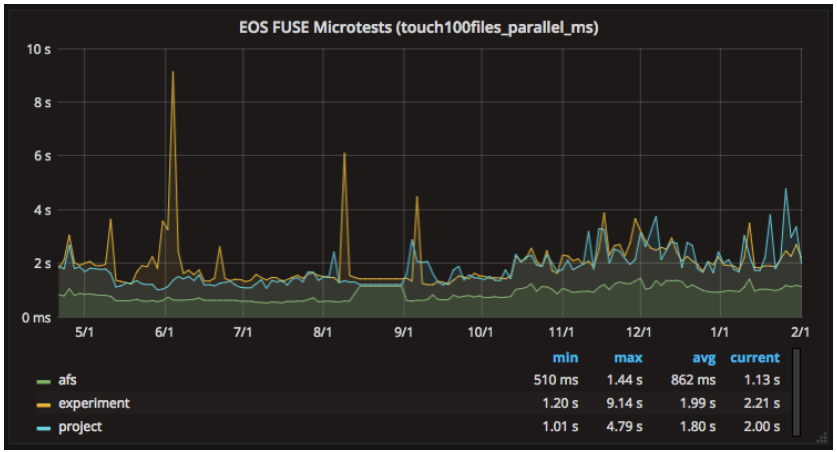
- Small sync writes sped up by up to 10x

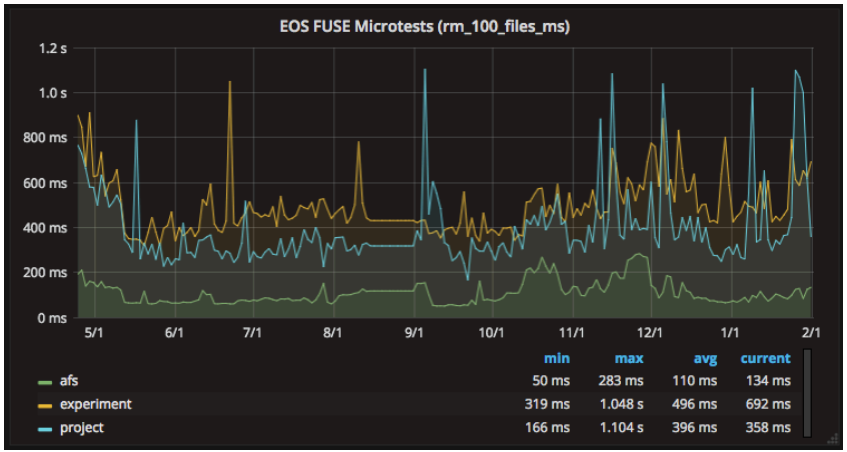
Microtests: Example speed up



- tar xf (940 small files) has improved by ~2-5x

Microtests: Needs Improvement





- Small (empty) file create/rm still needs work

Packaging and Configuration

- CERN uses Puppet for config management:
  - Wrote an *eosclient* module to configure our lxplus/lxbatch nodes

- Requirements:
  - Unique config for each /eos instance (user vs exp, etc…)
  - PAM integration for eos/kerberos
  - Use autofs instead of sysv service to allow smoother upgrades
  - selinux: and eosfuse selinux module compatible with autofs

Puppet eosclient usage

- Basic usage

```
include eosclient
```

```
---
eosclient::enable_fuse: true
eosclient::enable_pam_hook: true
eosclient::mounts:
  - user
  - cms
eosclient_custom_config:
  EOS_FUSE_CACHE_SIZE: 134217728
```

- Auto-mount config

```
include eosclient
```

```
---
eosclient::enable_fuse: true
eosclient::enable_pam_hook: true
eosclient::enable_autofs: true

eosclient::mounts:
  - ams
  - experiment
  - project
  - user
  - workspace
```

https://gitlab.cern.ch/ai/it-puppet-module-eosclient/blob/master/code/README.md

Full instance configurations are in the appendix of this talk.

QA Testing

- The last requirement was a qa process for staging out new releases

1. Upstream proposes an RC via their dss-ci repo [1]:
2. We build the RC in koji for our eos6/7-testing repo [2]
3. Quick regression testing: Build new el6/el7 nodes using sysv and autofs, run interactive microtests.
4. Upstream tags the RC as a new release (e.g. 4.1.15)
5. We build the new release in koji (still eos6-testing / eos7-testing), and check again for regressions.
6. We tag the build for our eos6-qa and eos7-qa repositories:
   - deploys to ~1-10% of the infrastructure
7. After 1 week in qa, the build is tagged into eos6/7-stable
   - Deploys to 100% the infrastructure

[1] http://dss-ci-repo.web.cern.ch/dss-ci-repo/eos/citrine/commit/

[2] http://linuxsoft.cern.ch/internal/repos/eos6-testing/

Publishing eos-fuse to the desktops

- Following the previous procedure, we optionally deploy stable versions to our Linux desktops
- RPMs are published for one week in our desktop “testing” repo, followed by publication in our official CentOS 7 “CERN” repo: http://linuxsoft.cern.ch/cern/centos/7/cern/
- The desktops have the most tested/stable eos-fuse release.

TODO

- More microtests, including more regression testing in preparation for rewritten eos-fuse
- Client log collection and analysis to find error patterns.
- Process automation:
  - Upstream jenkins could trigger builds in koji
  - Continuous integration testing


CERN

/eos Puppet Common Config

```
eosclient_common_config:
  XRD_LOGLEVEL: Info
  EOS_FUSE_USER_KRB5CC: 1
  EOS_FUSE_CACHE_SIZE: 268435456
  EOS_FUSE_CACHE_PAGE_SIZE: 32768
  EOS_FUSE_SHOW_SPECIAL_FILES: 0
  EOS_FUSE_RDAHEAD: 1
  EOS_FUSE_RDAHEAD_WINDOW: 262144
  EOS_FUSE_NOPIO: 1
  EOS_FUSE_PIDMAP: 1
  EOS_FUSE_DEBUG: 0
  EOS_FUSE_LOGLEVEL: 4
  EOS_LOG_SYSLOG: 0
  EOS_FUSE_NEG_ENTRY_CACHE_TIME: 0.1
  XRD_CONNECTIONWINDOW: 10
  XRD_CONNECTIONRETRY: 4096
  XRD_REQUESTTIMEOUT: 60
  XRD_STREAMTIMEOUT: 60
  XRD_TIMEOUTRESOLUTION: 1
  XRD_STREAMERRORWINDOW: 60
  XRD_REDIRECTLIMIT: 5
  XRD_WORKERTHREADS: 16
  XRD_DATASERVERTTL: 300
  XRD_LOADBALANCERTTL: 1800
  XRD_APPNAME: eos-fuse
```

https://gitlab.cern.ch/ai/it-puppet-module-eosclient/blob/master/data/eosclient.yaml

/eos Puppet Per-Instance Config

```
eosclient_instance_config:
  ams:
    EOS_FUSE_MGM_ALIAS: eosams.cern.ch
    EOS_FUSE_MOUNTDIR: /eos/ams/
  atlas:
    EOS_FUSE_MGM_ALIAS: eosatlas.cern.ch
    EOS_FUSE_MOUNTDIR: /eos/atlas/
  australia:
    EOS_FUSE_MGM_ALIAS: p05151113837349.cern.ch
    EOS_FUSE_MOUNTDIR: /eos/australia/
    EOS_FUSE_USER_KRB5CC: 0
    EOS_FUSE_RMLVL_PROTECT: 4
    EOS_FUSE_EXEC: 1
  cms:
    EOS_FUSE_MGM_ALIAS: eoscms.cern.ch
    EOS_FUSE_MOUNTDIR: /eos/cms/
```

```
experiment:
  EOS_FUSE_MGM_ALIAS: eospublic.cern.ch
  EOS_FUSE_MOUNTDIR: /eos/experiment/
  XRD_REQUESTTIMEOUT: 300
  XRD_STREAMTIMEOUT: 300
  XRD_STREAMERRORWINDOW: 300
lhcb:
  EOS_FUSE_MGM_ALIAS: eoslhcb.cern.ch
  EOS_FUSE_MOUNTDIR: /eos/lhcb/
pps:
  EOS_FUSE_MGM_ALIAS: eospps.cern.ch
  EOS_FUSE_MOUNTDIR: /eos/pps/
project:
  EOS_FUSE_MGM_ALIAS: eosuser.cern.ch
  EOS_FUSE_MOUNTDIR: /eos/project/
  EOS_FUSE_RMLVL_PROTECT: 1
user:
  EOS_FUSE_MGM_ALIAS: eosuser.cern.ch
  EOS_FUSE_MOUNTDIR: /eos/user/
  EOS_FUSE_RMLVL_PROTECT: 2
```

https://gitlab.cern.ch/ai/it-puppet-module-eosclient/blob/master/data/eosclient.yaml