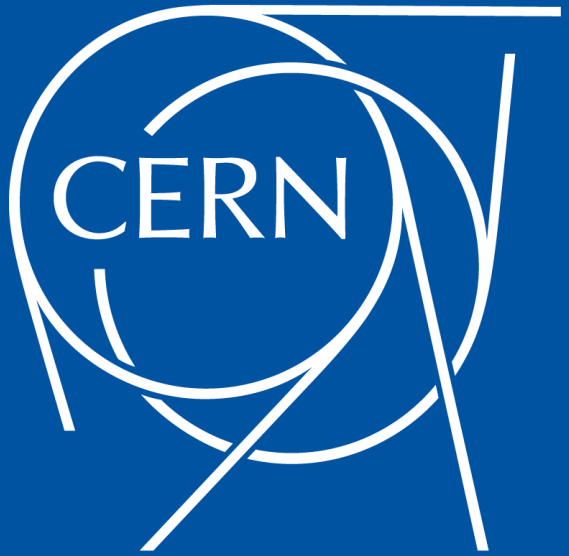

CERN

EOS @ CERN: The road to QuarkDB

Cristian Contescu, on behalf of the EOS Operations Team

Outline

- What is QuarkDB?
- Namespace implementations comparison
- Deployment of QuarkDB in production
  - Timeline
  - Different cluster setups
  - Conversion guide
- Conclusions

What is QuarkDB

- Key-value data store developed @ CERN (in the IT Storage group)
  - built on top of RocksDB
  - subset of redis command set
- Highly available
  - raft distributed consensus algorithm

EOS namespace implementations comparison

Native (In-Memory) Namespace

- Very fast and with low latency... but...
- Re-do log format fully loaded in-memory
- Limit on headnode RAM
- Restart time depends on #entries
- Compaction reduces loading time

QuarkDB Namespace

- Catalogue Stored in a HA K/V Store
- Resident on disk
- No more RAM limitation
- Restart time not depending on #entries
- Active caching of entries

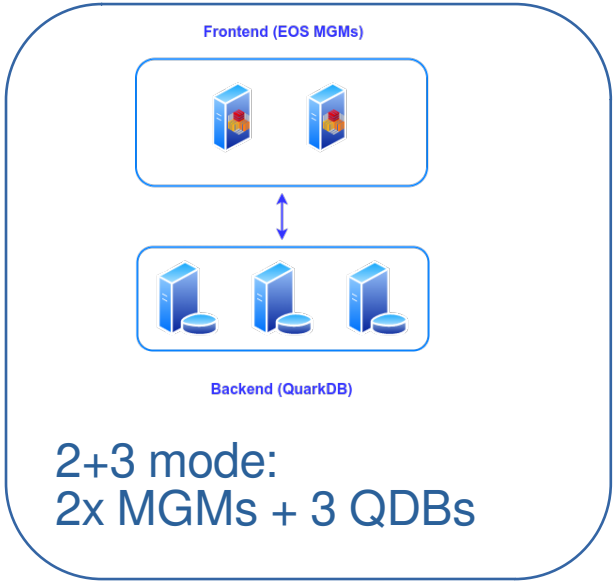
Deployment in production - timeline

EOSBACKUP
EOSHOME(s)
EOSPROJECT(s)

EOSALICE
EOSATLAS
EOSCMS
EOSLHCB
EOSPUBLIC

EOSALICEDAQ
EOSMEDIA (soon)

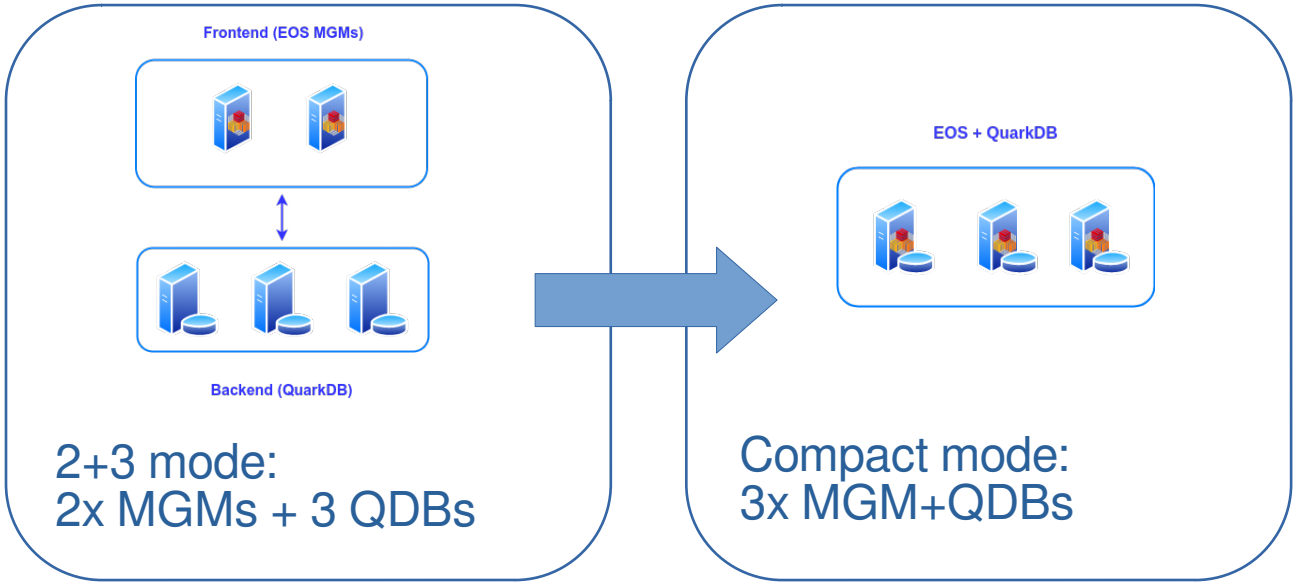
Deployment in production – different cluster setups



2+3 mode:
2x MGMs + 3 QDBs

Compact mode:
3x MGM+QDBs

Deployment in production – different cluster setups



Deployment in production – conversion guide (I)

- Conversion node requirements and advices:
  - CentOS 7 with `eos-server` and `quarkdb` packages installed
  - At least same amount of RAM as currently used by the in-memory NS
    - Disabling `transparent_hugepage` might help saving some memory
    - If in doubt: use a swap file/partition during the conversion
  - SSD disk for the QuarkDB namespace
    - with scheduler set to `noop`
  - Set `tuned` to `latency_performance` profile
    - e.g.: `tuned-adm profile latency_performance`

Deployment in production – conversion guide (II)

- Stop the MGM service (after taking note of the current # of files&dirs)
- Copy the most recent mdlog files from the master (in case you don't use the master for conversion)
- Do an offline compaction on the mdlog files

```
$ eos-log-compact files.<fqdn>.mdlog files_compacted.mdlog
$ eos-log-compact directories.<fqdn>.mdlog directories_compacted.mdlog
```

- Create QuarkDB folder for the conversion process
  - Make sure the owner/group are the same as the user running the service (in our case, daemon:daemon)

```
$ quarkdb-create --path /var/lib/quarkdb/convert/
$ chown -R daemon:daemon /var/lib/quarkdb/convert/
$ mkdir /var/spool/xrootd/quarkdb ; chown -R daemon:daemon /var/spool/xrootd/quarkdb
$ mkdir -p /var/run/xrootd/quarkdb ; chown -R daemon:daemon /var/run/xrootd/quarkdb
```

Deployment in production – conversion guide (III)

- Put QuarkDB node into bulk insertion mode

```
/etc/xrootd/xrootd-quarkdb.cfg:
 xrd.port 7777
 xrd.protocol redis:7777 /usr/lib64/libXrdQuarkDB.so
 redis.mode bulkload
 redis.database /var/lib/quarkdb/convert
# make sure redis.myself is not set / comment it out:
# redis.myself  <FQDN>:7777
```

- Restart the QDB service:

```
$ systemctl restart xrootd@quarkdb
```

- ...and start the conversion:

```
$ eos-ns-convert files_compacted.mdlog directories_compacted.mdlog localhost 7777
```

- Once the conversion is done, prepare QuarkDB for running in cluster mode

```
$ systemctl stop xrootd@quarkdb

$ quarkdb-create --path /var/lib/quarkdb/eosns --clusterID $(uuidgen) --nodes
$HOSTNAME:7777,node2.example.org:7777,node3.example.org:7777 --steal-state-machine /var/lib/quarkdb/convert/current/state-machine/

$ chown -R daemon:daemon /var/lib/quarkdb/eosns/

$ rsync -avxXHP /var/lib/quarkdb/eosns/ node2.example.org:/var/lib/quarkdb/eosns/

$ rsync -avxXHP /var/lib/quarkdb/eosns/ node3.example.org:/var/lib/quarkdb/eosns/
```

```
/etc/xrootd/xrootd-quarkdb.cfg:
 xrd.port 7777
 xrd.protocol redis:7777 /usr/lib64/libXrdQuarkDB.so
 redis.mode raft
 redis.database /var/lib/quarkdb/eosns
 redis.myself  <FQDN>:7777
 redis.password_file /etc/eos.keytab
```

- ...and start the `xrootd@quarkdb` services on all three nodes

Deployment in production – conversion guide (IV)

- Check the QDB cluster:

```
$ redis-cli -p 7777 raft-info
```

- Check #files and #directories:

```
$ redis-cli -p 7777 lhlen eos-file-md
$ redis-cli -p 7777 lhlen eos-container-md
```

- Adjust the MGM and FST configurations file:

/etc/xrd.cf.mgm:

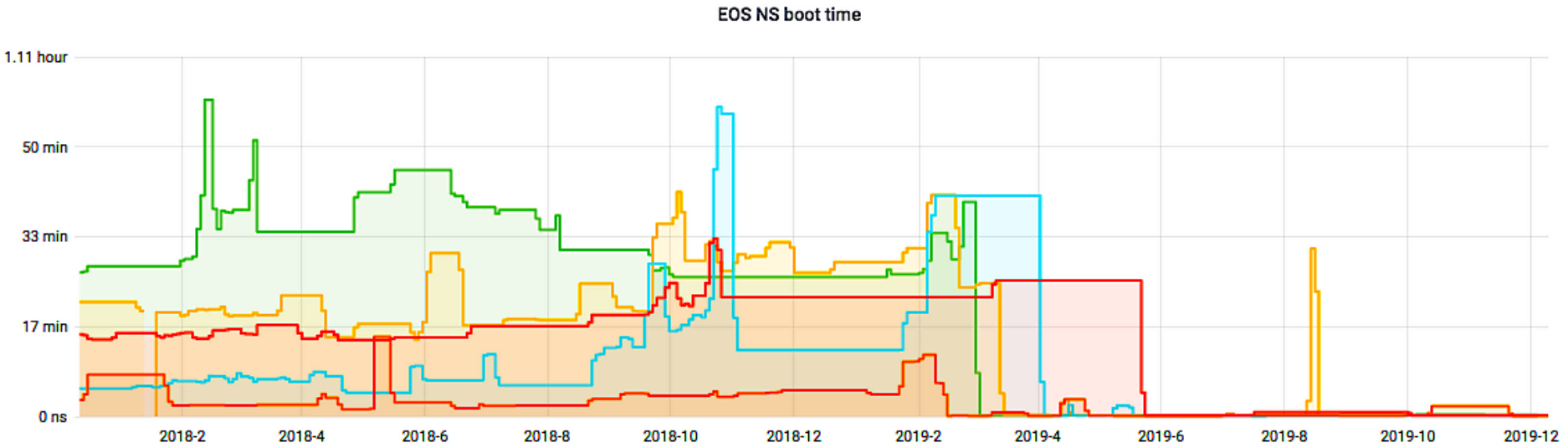
```
mgmofs.nslib /usr/lib64/libEosNsQuarkdb.so
mgmofs.qdbcluster eosXXXX-qdb.cern.ch:7777
mgmofs.qdbpassword_file  /etc/eos.keytab
```

/etc/xrd.cf.fst:

```
fstofs.qdbcluster eosXXXX-qdb.cern.ch:7777
fstofs.qdbpassword_file  /etc/eos.keytab
```

- ...and finally... start the MGM and restart the FSTs

Deployment in production: boot time



	min	max
alice.boot	10 s	58.7 min
atlas.boot	14 s	41.7 min
cms.boot	11 s	57.4 min
lhcb.boot	6 s	14.8 min
public.boot	10 s	33.0 min

Conclusions

- Greatly improved EOS startup time: from tens of minutes to seconds
- Service incidents no longer amplified by the long startup time
- With the QuarkDB backend, the memory capacity is no longer an issue
- (Almost) seamless interventions on the cluster members (running QDB in degraded mode has no visible impact)

EOS + QuarkDB = 

Thank you !

Any questions?

home.cern