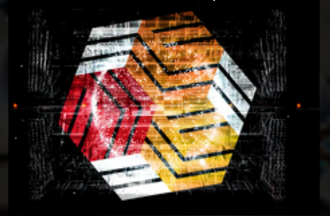
EOS workshop 2022

Taiming batch access in EOS

IT-ST-PDS

Andreas-Joachim Peters
CERN IT-ST for the EOS team

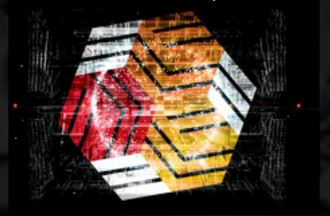


Introduction

- **What is the problem?**
  - EOS provides file access via redirection from a central namespace service [MGM]
  - the MGM is a multithreaded application and a typical production scenario is
    `n(clients) >> n(threads@mgm)`
    e.g. 30k clients >> 4096 threads

  - when an individual **user launches batch jobs**, it happens often that several **thousand jobs start** at the same time for this user
    - some jobs act like a DOS attack on the namespace service even if they access only few files via /eos
    - this is particular **problematic in** non-physics instances like **CERNBOX**, where people rely on interactive usability

- **What can we do?**
  - rate-limit meta-data access by user
  - limit the number of worker threads per user



Namespace Statistics

eos ns stat displays rates for all MGM operations - can be broken down by user [-a]

who	command	sum	5s	1min	5min	1h	exec(ms)	sigma(ms)	99p(ms)	max(ms)
all	Access	229.06 M	335.75	304.76	361.76	349.10	-NA-	-NA-	-NA-	-NA-
all	AccessControl	295	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	AdjustReplica	121.99 K	0.25	0.12	0.08	0.09	2.47	11.93	89.30	76.92
all	AttrGet	5.26 M	10.00	9.47	8.36	9.22	0.21	1.01	9.84	3.03
all	AttrLs	399.95 M	509.75	488.20	611.89	649.06	0.04	0.01	0.08	0.07
all	AttrRm	549	0.00	0.00	0.00	0.00	0.47	2.41	24.32	2.64
all	AttrSet	100.29 K	0.00	0.02	0.03	0.15	1.28	4.07	22.16	19.70
all	Cd	1	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	Checksum	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	Chmod	60.51 K	0.00	0.17	0.10	0.19	0.50	1.27	11.64	5.57
all	Chown	3.55 K	0.00	0.02	0.01	0.01	-NA-	-NA-	-NA-	-NA-
all	Commit	32.90 M	118.75	93.51	94.69	62.74	0.72	0.63	4.27	2.98
all	CommitFailedFid	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	CommitFailedNamespace	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	CommitFailedParameters	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	CommitFailedUnlinked	512.84 K	0.00	0.00	0.00	0.42	-NA-	-NA-	-NA-	-NA-
all	ConversionDone	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	ConversionFailed	0	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	CopyStripe	6.65 K	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	DrainCentralFailed	1.89 K	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	DrainCentralStarted	2.81 M	0.00	0.00	0.00	0.04	-NA-	-NA-	-NA-	-NA-
all	DrainCentralSuccessful	2.81 M	0.00	0.00	0.00	0.04	-NA-	-NA-	-NA-	-NA-
all	Drop	20.76 M	12.75	7.95	18.15	9.93	0.34	0.34	2.16	1.89
all	DropStripe	16	0.00	0.00	0.00	0.00	0.40	0.50	2.16	1.10
all	DumpMd	827	0.00	0.00	0.00	0.00	-NA-	-NA-	-NA-	-NA-
all	EAccess	31.45 M	49.00	46.05	49.43	29.92	-NA-	-NA-	-NA-	-NA-

…

Access Limits

eos access allows to view and set access rate limits [Hz]

```
[ 01 ]                  rate:user:*:Chmod => 500
[ 02 ]                  rate:user:*:Chown => 500
[ 03 ] rate:user:*:Eosxd::ext::LS-Entry => 10000
[ 04 ]      rate:user:*:Eosxd::ext::SET => 50
[ 05 ] rate:user:*:Eosxd::int::FillFileMD => 8000
[ 06 ]              rate:user:*:OpSetFile => 300
[ 07 ]                   rate:user:*:Open => 500
[ 08 ]         rate:user:*:OpenDir-Entry => 20000
[ 09 ]               rate:user:*:OpenProc => 200
[ 10 ]               rate:user:*:OpenRead => 500
[ 11 ]              rate:user:*:OpenWrite => 300
[ 12 ]                     rate:user:*:Rm => 100
[ 13 ]                   rate:user:*:Stat => 2000
[ 14 ]     rate:user:foo:Eosxd::ext::LS => 1
[ 15 ] rate:user:foo:Eosxd::ext::LS-Entry => 10
[ 16 ]            rate:user:bar:AttrLs => 50
[ 17 ]           rate:user:bar:Stat => 50
```

Application of Limits

eos ns stat | grep Stall shows functions where limits are applied and their rates

```
[root@eoshome-i01 (mgm:master mq:master) ~]$ eos ns stat | grep Stall
 all OpenStalled                            0     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall                                  0     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall::AttrLs                         44     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall::Eosxd::ext::LS-Entry    80.57 K     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall::Open                     1.74 M    10.00     0.00     0.00     0.23     -NA-     -NA-     -NA-     -NA-
 all Stall::OpenProc               100.59 K     0.00     0.00     0.00     0.68     -NA-     -NA-     -NA-     -NA-
 all Stall::OpenRead               474.33 K     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall::Rm                      41.17 K     0.00     0.00     0.00     0.01     -NA-     -NA-     -NA-     -NA-
 all Stall::Stat                     7.06 K     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
 all Stall::threads::77965              119     0.00     0.00     0.00     0.00     -NA-     -NA-     -NA-     -NA-
```

To figure out who is being stalled, one can add the -a option:

eos ns stat -a | grep Stall

Thread Limits

eos ns stat shows the current status of the thread pool, its limits and who is using it

uid	threads	sessions	limit	stalls	stalltime	status
0	1	3	500	0	22	user-OK
2	1	0	500	0	1	user-OK
83***	1	0	500	0	1	user-OK
120***	1	0	500	0	1	user-OK

The thread pool limit is configured as a rate limit using the access interface

eos access ls l grep threads

```
[ 16 ]                          threads:* => 500
```

Thread Limits

There are three types of thread limit rules:

1. **wild-card** rules for all users
2. **specific** user rules
3. **global** thread limit for user requests

```
threads:* => 500
threads:cmsprod => 2000
threads:max => 3900
```

The global thread limit is useful to reserve a given amount of threads to EOS components. The global limit only applies for **uid>3** and excludes the restic backup

Thread Limits

eos ns stat

uid	threads	sessions	limit	stalls	stalltime	status
0	1	3	500	0	22	user-OK
2	1	0	500	0	1	user-OK
63***	1	2500	500	1023	1	pool-OL
110***	1	6000	500	3034	1	pool-OL

possible status values:

user-OK *: threads are not limited*
user-LIMIT : available thread limit is >90% used
user-OL : available thread limit is reached
pool-OL : the global thread limit has been reached

Summary

How well does this work?

- service **degradation** has been significantly **decreased**
- we still saw **few episodes** where the configured limits where not sufficient to avoid degradation

- in these cases the following always works:

```
eos access ban user overloadingusername
```


CERN storage technology
used at the Large Hadron Collider (LHC)
EOS Open Storage
Thank you!
Question or Comments?
eos.web.cern.ch