EOS and fixes & low level changes for OpenSSL, XRootD & eosxd

David Smith - CERN IT Storage Group
15 March 2024

Overview: looking at some fixes that are relevant to EOS

As usual, this year some bug fixes, scaling improvements or other changes were needed; for EOS service stability with Centos7, and for issues discovered when moving to Alma9 at scale. I look at:

- **XRootD**
  - A server framework, the ‘xroot’ client-server protocol and a component that works as an HTTP server.
- **OpenSSL**
  - Typically used for cryptographic operations, TLS.
- **eosxd**
  - The EOS component that runs as the fuse user space service

Will mention the work of several people from the EOS and XRootD teams.

XRootD

- **Xrootd - server**
  - Daylight saving time change, rare incorrect filehandle returned after opening file, CRL file handling, rare connection problems. (#1955, #1998, #2065, #1928, #2021)
  - Other developments
    - features like SciTags, or refactoring in XrdHttp concerning getting file byte ranges ranges (#2100, #1976 and others)
    - Other XrdHttp improvements or fixes, digest handling (several tickets)
- **Xrootd - client**
  - Stream timeout, crash on early connection close (#2042, #1934)

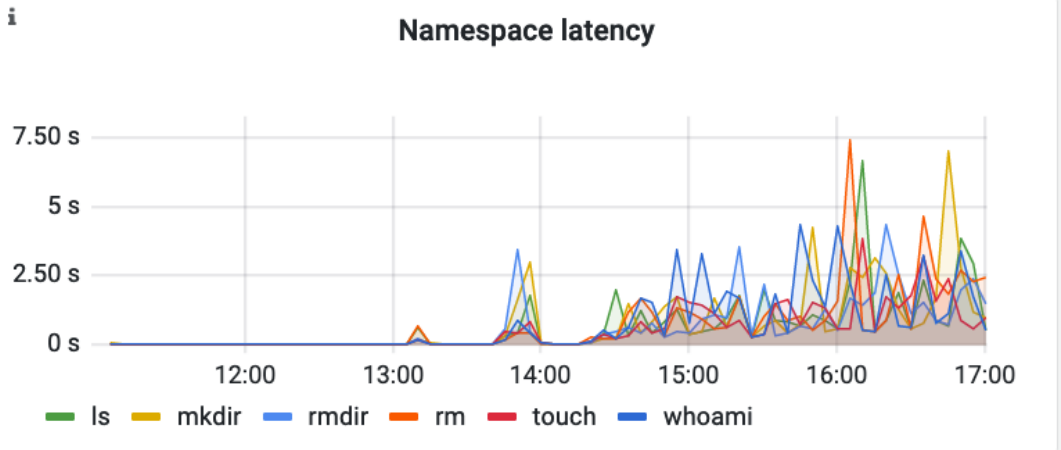
OpenSSL

- **Issue with OpenSSL 1.0.2k (centos 7):**
  - Occasional spiking number of threads on MGM; stack trace show many ongoing openssl calls
  - Improve generation of thread ID used by openssl (XROOTD #2084)

- **Slow with Alma9 (openssl 3.0.7)**
  - Seen to be much slow for gsi connections (0.15s to ~1.9s for an xrdcp of a tiny file)
  - Cause was that a prime test became slower (XROOTD #2162)

OpenSSL 1.0.2k: slow ERR_get_state

- **EOS-5743**
  - Increase in namespace lstency (CMS and ATLAS instances); stack traces showed many threads in EER_clear_error/ int_thread_get.



OpenSSL 1.0.2k: slow ERR_get_state

- Poor distribution of entries the interal openssl hash table containing per-thread last error status.
  - Hash key based on ”thread ID”, which can be computed with a user supplied function
  - Addressed the problem by changing the thread ID definition so that the number of bits that typically change between threads is larger

```
uint64_t x = (uint64_t)XrdSysThread::ID();
x ^= x >> 30;
x *= 0xbf58476d1ce4e5b9ULL;
x ^= x >> 27;
x *= 0x94d049bb133111ebULL;
x ^= x >> 31;
tid_ = (unsigned long)x;
```

OpenSSL 3.0.7: slow DH_check

- **Saw gsi connections with xrdcp or other clients taking much longer on alma9 compared to centos 7**
  - Call in OpenSSL 3 (DH_check) seen to be ~10 times slower than in OpenSSL 1
  - Purpose of DH_check is to validate some Diffie-Hellman exchange parameters used during the gsi authenitcation handshake; the heaviest part computationally is checking if a number is prime. (Typically a 3072 bit number *p* and also (*p*-1)/2 are checked).
- **OpenSSL3 has adjusted it prime checking to functions (see openssl #9272) to be easier to use & harder to misuse.**
  - Two aspects to the check: trial division with small primes, followed by application of Miller-Rabin (probabilistic primality test) of *n* rounds, which the caller had to set.
  - Motivated by reference in #9272, trial division is reduced and rounds of Miller-Rabin are rasied; to correctly establish confidence even for potentially adversarially selected test number *p*.

OpenSSL 3.0.7: slow DH_check

- **Approach taken (xrootd #2162)**
  - Recent xrootd-based server will use the same DH parameters. If the client detects an exact match no specific DH_check is done.

Eosxd (the EOS fuse service)

- **Ongoing effort to address problems as they are noticed and to improve stability**
  - There was set of issues of similar type, variation of threading issues
  - Thread not-safe:
    - Concurrent update of std::strings
    - Update of members of a protobuf structure
  - Deadlock
    - Lock order violation
    - Wrong lock acquired due to temporary release and then reacquire of lock

Summary

As well as developing new features there is a development effort around tracing of failures (e.g. crashes or deadlocks) and effort to improve performance when needed.

Sometimes these are rare effects, but which can nevertheless introduce important problems.

Sometimes appear in because of new work or deployment at scale on a new OS, e.g. because the version of dependencies change.

Testing and careful deployment but still there are issues

Sometimes in the core part of EOS but important problems can also enter via dependencies

Thank you!

home.cern