TACC

TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS

The University of Texas at Austin

XALT 3.0: What Got Us Here

Robert McLay

June 15, 2023

XALT: Outline

- Releasing XALT 3.0 today
- Why change from XALT 2.10.*?
- What is new here XALT 3.0?
- A summary of XALT 2.0 and now

XALT 3.0 release

- I have been waiting for a good reason to bump XALT to 3.0
- It has been a long trip from the first release of XALT 2.0
- First xalt check-in was in 2014.
- The first release of XALT 2.0.0-devel was in April 17, 2018.
- XALT 3.0 is a evolution, not the revolution that 2.0 was.

XALT 1.0 ⇒ XALT 2.0

- XALT 1.0 use mpirun/ibrun to capture runtime and info about mpi executions (Only MPI executions tracked)
- XALT 2.0 introduced the ELF trick to attach to *every* execution to track all programs executions
- The journey to XALT 3.0 is how to deal with this fire hose of data

Major changes to XALT since 2.0

- Improved transport methods: SYSLOG, File, Curl
- Better debugging of XALT operations
- Filtering and Sampling
- Improved Performance
- Better Container Support.
- Support when UUID aren't

Better Debugging of XALT Operations

- Track what XALT does during execution
- XALT_TRACING=run try.exe
- Track what XALT does during Linking:
- XALT_TRACING=link gcc -o try.exe try.c
- Track what XALT does during ingestion:
- xalt_file_to_db.py -D ...

XALT Filtering and Sampling

- XALT generates a fire hose worth of data.
- Choice 1: Use big computer(s) to collect the data
- Choice 2: Filter and Sample
- XALT can filter based on path
- (NEW) filtering based on command line arguments
- XALT does both filtering and sampling (Site configurable)

XALT Sampling

- Sampling based on execution time
- Different sampling rules possible for MPI and NON-MPI executions
- Next meeting: Design changes for Sampling and Signal handling support and failure

Improved Performance

- Stopped taking SHA1 of shared libraries
- Full Tracking takes less than 0.01 seconds in my tests
- Timing are included in run.*.json files in the Q/A table.

Better Container support

- This has been the biggest area of change.
- All required libraries for XALT are copied to XALT installation directories.
- XALT uses dlopen to dynamically link in libuuid etc.
- There is support for Alpine containers (non-glibc based)

Two support programs

- `xalt_configuration_report`: How XALT is configured at your site
- `xalt_extract_record`: How, who and when an executable file was built.

Protecing XALT from bad user programs

- Better protection of malloc use in XALT
- XALT doesn’t free memory to avoid user memory errors

Some UUID implementations aren’t

- Older version of libuuid can produce duplicates
- XALT depends on UUID
- XALT now adds a CRC to the record to avoid issues when using Syslog transmission

Separate C++ programs replaced for Run record

- XALT used C based *.so to attach to each executable
- It used C++ programs to build json record (Hash Tables!!!)
- A version of MPI libraries prevented any execution after MPI_finalize.
- This caused a re-write build the json run record in C (Not C++)
- I found a hash table implementation in C (uthash etc) (Not connnected to U Texas)

Last Changes

- Support for package filtering based on command line arguments
- Support for ARM based linux computers
- There is one piece of assembly code (the watermark!)
- Fortunately the gnu assembler is the same for X86 and ARM.
- This is in terms of how strings are stored.

Conclusions

- Many changes between XALT 2.0 and 3.0
- Very few bug reports in the last year.

Future Topics?

- Next Meeting will be on July 20, 2023 at 10:00 am U.S. Central (15:00 UTC)
- We will be discussing the design changes to support sampling and why signal won't save us.