TACC
TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

XALT Changes to support Sampling and why Signaling didn't save us

Robert McLay

June 16, 2022

XALT: Outline

- ▶ XALT 2 Can track everything
- ▶ This is bad unless you want to throw a Supercomputer to collect data.
- ▶ XALT has several tricks to manage the firehose of data
- ▶ 1st Level Defense: Path Filtering
- ▶ 2nd Level Defense: Sampling
- ▶ 3rd Level Defense: Signals and why they don't work

History

- XALT 1 only tracked MPI programs by modifying mpirun, ibrun etc.
- XALT 2 can track everything via libelf trick
- See xalt.readthedocs.io and libelf_trick in source repo on how this works.
- 1st line of defense: path filtering

Path Filtering

- ▶ XALT allows site to have paths to KEEP or SKIP
- ▶ Your site `Config.py` has a section like this:
- ▶ This allows sites to pick some program like `sed, perl` to track and ignore others.

```
path_patterns = [
    ['PKGS',  r'.*\/R'],
    ['PKGS',  r'.*\/MATLAB'],
    ['PKGS',  r'.*\/python[0-9.]*'],
    ['KEEP',  r'^\/usr\/bin\/sed'],
    ['KEEP',  r'^\/usr\/bin\/perl'],
    ['SKIP',  r'^\/usr\/.*'],
    ['SKIP',  r'^\/sbin\/.*'],
    ['SKIP',  r'^\/bin\/.*'],
```

But that is not sufficient!!

Not Sufficent!

- A two hour image processing generated millions of json records.
- It took over 4 days to process this one day's result.
- Obviously, this is not sustainable.

Not Sufficient, part 2

▶ I thought that MPI programs were safe from thru-put computing, But...

▶ At least one group is using short time 4 task MPI prgms to train a neural network.

▶ So both MPI and NON-MPI executions must be sampled.

Non-MPI Programming Sampling

- ▶ As I have mention before
- ▶ Scalar (Non-MPI) execution can be configured to be sampled
- ▶ TACC's Scalar rules are below.
- ▶ Longer execution increases change of being tracked

```
interval_array = [
    [     0.0,             0.0001 ],
    [ 1800.0,             0.01    ],
    [ 7200.0,             1.0     ],
    [ sys.float_info.max, 1.0     ]
]
```

MPI Programming Sampling

- ► TACC's MPI rules are below.
- ► Longer execution increases change of being tracked

```
mpi_interval_array = [
    [     0.0,                0.0001 ],
    [   900.0,                0.01   ],
    [  1800.0,                1.0    ],
    [ sys.float_info.max,     1.0    ]
]
```

Consequences of Sampling

- ▶ Before Sampling XALT generated both a start and end json record.
- ▶ XALT only produces records on the "wire" that are meant to be saved.
- ▶ ⇒ no start record for sampled data
- ▶ Can't know how long something will run at the beginning
- ▶ So only an end record is produced
- ▶ But this means that the program must end normally.
- ▶ Means that segfault etc runs will be ignored

Consequences of Sampling: Package Records

- ▶ A package record can be generated for python, R, MATLAB at anytime.
- ▶ A package record needs an execution record to "hang-on" to.
- ▶ Instead of forcing all python and R execution to produce a start record.
- ▶ ALL package json records are written to /dev/shm
- ▶ If chosen then json records are copied to "wire"
- ▶ If not then deleted at end of execution
- ▶ Or left to hang if python job doesn't complete.

What about long running MPI simulations?

- ▶ Some Large MPI execution never terminate
- ▶ Long running simulations like Weather or other calculations
- ▶ These run as many timesteps as they can in 24 or 48 hours job window.
- ▶ They restart from the last timestep or similar results file.

Long running MPI Simulations (II)

- ▶ XALT has a configuration option MPI_ALWAYS_RECORD
- ▶ Typically set to 128 MPI Tasks (Site configurable)
- ▶ Hopefully no-one is training Neural Networks with 128 or more Tasks.
- ▶ This means that XALT does NOT sample any executions 128 tasks or more.
- ▶ Sites can use the endtime from "SLURM" accounting to set the endtime that doesn't exist.

TACC

What about using Signals to capture scalar execution?

► XALT has support for capturing end records for segfault etc jobs.

► It is off by default now.

► This works well for almost all executions

► But occasionally it doesn't with some python programs

► And I don't understand why.

► XALT sets the signal handler first

► This allows a user program to overwrite XALT's handler

► But this sometime causes problems.

► But since XALT has to sample scalar jobs anyway

► This is not problem worth solving

What about using Signals to capture Long running MPI execution?

- ▶ SLURM sends a signal to a job that it is about to timeout.
- ▶ This works with scalar executions
- ▶ This could be used by XALT to write the end json record
- ▶ If this worked, XALT could stop writing start json records
- ▶ But it doesn't work the way I want it to.

Signals and MPI executions (II)

► However the controlling MPI executable doesn't pass this signal on to the user program.

► Our resident MPI expert Amit Ruhela tried to get this to work with Intel MPI and Mvapich2 but couldn't

► This means that large MPI execution > 127 tasks are not sampled.

Signals and MPI executions (III)

- ▶ I not sure that this would always work anyway
- ▶ The timeout signal would have to be captured by task 0.
- ▶ Remember XALT does nothing on all other tasks.

Conclusions

▶ Sampling saved XALT from the firehose of data from thru-put computing

▶ Sampling works for Scalar and Small MPI executions

▶ Signals won't save us.

Future Topics?

▶ How XALT "hijack" the link process to watermark user programs.

▶ This hijacking is not required for execution tracking

▶ Next Meeting will be on July 22, 2022 at 10:00 am U.S. Central (15:00 UTC)

▶ This talk will be given by my colleague: Amit Ruhela.