Extracting Useful Data from XALT DB

Robert McLay

Jan. 20, 2022

XALT: Outline

- ▶ XALT Parts: Generating and Storing
- ▶ Quick Discussion about what is stored
- ▶ Caveats about XALT Records
- ▶ xalt_usage_report.py
- ▶ Kinds of reports available
- ▶ What kinds of reports would sites like?
- ▶ Future Topics for XALT zoom mtg.

XALT Generating and Storing

- ▶ ALL programs on system have XALT via LD_PRELOAD
- ▶ Some prgms generate *.json records
- ▶ These records can be written to a MySQL DB (not required)
- ▶ Afterward *.json records are deleted.

Caveats about XALT records

- ► Core-Hours not Node-Hours
- ► XALT knows about mpi tasks and threads
- ► It doesn't know if a node is shared.
- ► User might run multiple single core prgms on a node
- ► User might run ingle core prgms on one or more nodes
- ► XALT runs inside each program.
- ► It is blind to what happens outside of user prgm.
- ► It would require a daemon on each node to know.

XALT time records won't match Accounting

- ▶ XALT will filter out or sample programs
- ▶ It won't catch them all.
- ▶ XALT can double count (rarely)
- ▶ If user forks off another prgm $\Rightarrow$ double counting
- ▶ If user runs > 1 prgm per core.
- ▶ XALT uses real time not cpu time.

Python prgm:
sbin/xalt_usage_report.py

- ▶ An attempt to be system agnostic.
- ▶ It provides a name mapping file.
- ▶ It reports data that TACC has found useful:
    - ▶ Overall Job counts
    - ▶ Self Built vs. Not.
    - ▶ It reports Top Execs by Core-Hours, Number of Runs, Number of users for All, MPI Only, Scalar
    - ▶ Top Module usage
    - ▶ Compiler usage
    - ▶ Library Usage

Overall Job counts

- ► Core hours for "system" prgms vs. user prgms.
- ► System prgms come from a module, User prgms don't.
- ► We see about 5% for system prgms.
- ► Still have to teach how to build program.

Self-Built vs. Not.

▶ Beside system supplied program we have groups that share prgms

▶ Like to track that.

▶ Prgms built under XALT know build user.

▶ XALT knows the run user.

▶ We report 2 to 4 % non self-built prgm runs.

xalt_name_mapping.py

```
equiv_patternA = [
    [ r'^pmemd'              , 'Amber*'      ],
    [ r'^sander'             , 'Amber*'      ],
    [ r'^absorptionrealx'    , 'BerkeleyGW*' ],
    [ r'^absorptioncplxx'    , 'BerkeleyGW*' ],
    [ r'^kernelrealx'        , 'BerkeleyGW*' ],
```

▶ Map prgms to projects.

▶ Sites may need to modify to match their site.

Future Topics?

- Package tracking
- Extracting results from the DB.
- Others?