

## Drilling Down: Understanding User-Level Activity on Today's Supercomputers

### Presenters

Robert McLay, TACC

Mark Fahey, NICS

Bill Barth, TACC

Richard Gerber/Zhengji Zhao, NERSC

Tim Robinson, CSCS

### Moderator

John Lockman, TACC

### Abstract

Lets talk real, no-kiddin supercomputer analytics: drilling down to the level of individual batch submissions, users, and binaries. And we're not just targeting performance: were after everything from which libraries and functions are in demand to preventing the problems that get in the way of successful science. This BoF will bring together those with experience and interest in technologies that can provide this type of job-level insight, and will be the kickoff meeting for a new Special Interest Group.

The BoF was very successful. We had 47 people put their names on our sign-up sheet. Our primary goal was to form a community and we have a good list of people with which we can start. Since the meeting, we have made a project space at SourceForge and put our slides there. We have also emailed everyone that we want them to join our mailing list that we set up.

During the BoF, we had 5 speakers each give 3 minutes to talk. The moderator started the meeting and then Robert McLay, Bill Barth, Richard Gerber, Tim Robinson, and Mark Fahey each spoke.

Robert McLay mentioned his goals which are to protect users and leave veteran users alone, and make life easier for staff. He talked about use of Lmod to control modules. He then talked about Lariat and what it tracks - modules loaded, shared libraries linked in, check that compiled executable is in the correct batch environment. He also mentioned that ALTD tracks similar things at NICS and that Lariat and ALTD will be combined into XALT -- NSF funded.

Next, Bill Barth talked about TACC Stats: it collects job level performance, analyzes jobs every night to identify problem jobs, provided to support and admins, and eventually will be available to all users. He then showed some examples of high imbalance, low flops, idle flops, and catastrophic performance drop. He also mentioned that TACC\_Stats collects meta data from the job stater script via lariat (I.e., dependency.)

Richard Gerber/Zhengji Zhao followed representing NERSC and NERSC User Services. NERSC uses ALTD and has made some modifications. They also use some other tools as well: Torque, ALPS, Darshan, IPM, LMT -- lustre data. And they try to expose as much as possible to users; all the data collection is transparent to users. Users can look at their own or other jobs on the system and most of the data collected is made available to the users with no interaction from the user. List of data includes nodes used correlated with job id, Darshan includes I/O stats, and ALTD captures link lines and all libraries included.

Then Tim Robinson of CSCS spoke. He said staff want to know what compilers/libraries are being used. He said surveys and module logs are not enough, and that ALTD is their preferred solution. ALTD intercepts the linker and records the link line, the job launcher pulls this information, and this has been in production since 2011. ALTD stores results in mysql database; for example use this to find all the users who have built with a buggy library.

Mark Fahey (NICS) was the last planned speaker. He is the developer of ALTD. He mentioned that XALT is a new NSF funded project and that it is a combination of lariat and ALTD. It will collect job level and link level data and they want to build a community around analytics regarding HPC jobs. The goal--

- census of libraries in use
  - What can we detect and report automatically
  - Can we track function calls --- within the libraries
- e.g. For widely used libraries -- which functions are actually being used  
Would be useful for popular libraries to determine which functions aren't worth optimizing.

The plan is to develop XALT to be a portable flexible tool to be installed at multiple sites and attempt to simplify the install process for sysadmins. Will work with TACC\_stats.

Things it can do:

- Can catch users using the wrong library
- Catches users with dynamic libraries running in the wrong environment

The formal presentations then ended and a question and answer period started. It was a very lively discussion with a great deal of participation from the audience. The topics ranged from what about making this part of the resource managers job? (Answer: This might be possible at some later date once the technology was solid.) To how would XALT capture this data? (Answer: The current answer is to instrument the mpi launcher script and possibly the linker) There were many question on how light-weight was XALT going to be. (Answer: Both precursors were very light-weight, so XALT will be as well)

We, the organizers of this BoF found the hour long discussion to be quite informative. We learned about tools like system-tap to help track data as well as

getting a feel for how interested the community is for tools like XALT. We had about 50+ people in the room and almost all of them put their names and email address so we feel that we have the makings of a growing community in this area.

Comments with responses  
=====

Comment: Less opposition for changing the linker than changing the scheduler since scheduler's change

Comment: Have you considered making this the resource manager's job?

Bill:

No. We don't want the scheduler/resource manager responsible for this.  
(Removes the scheduler problem)

Comment: Would be natural for the resource manager to control this.

Comment: Could take snapshots from the system

Bill: This mechanism is lightweight and doesn't run during the execution  
We use our wrapper instead of the scheduler's job starter

Comment: What starts the job?

Scheduler.

Mark: We don't know every scheduler. But, there is a job launcher in every batch script. This can be wrapped. TACC has ibrun which wraps MPI executables. The only way to catch everything is to track everything in the batch script.

Comment:

If the RM had a tool that captured details of every child process.

Comment:

This is a barrier to adoption for smaller clusters.

This is already done at many sites(the mpirun wrapper).

Mark:

We want to make this work in the small shops. Make it easy to install and run.

Comment:

Could this work for data analysis workflows?

Robert:

The problem is for non-MPI jobs because they don't need a job starter(mpirun,..) and there is no easy way to wrap these executables.

Mark:

XALT won't be able to catch this info without a job launcher. But, tacc stats could get some data.

Bill:

Some apps that use MPI under the covers have information collected.

Comment:

Could this info be captured with accounting logs?

Bill:

Schedulers don't collect his data.

Comment:

What if you collect all process accounting?

Bill:

Could work but there would be a lot of chaff.

Comment:

Already examined using the system accounting to get data. Used filters to capture what was needed. Use gaussian which doesn't use MPI.

C

omment:

Have you spoken with gcc developers about tracking in the linker.

Bill:

Could be feasible for gcc. But, not for multiple compilers.

Mark:

We intercept at points the user's don't notice. Nothing is executed during the parallel run.

This is the difference between capturing at link stage and capturing during the run.

Comment:

What is minimal impact? seconds? milliseconds?

Mark:

less than 1 second -- not noticeable

Comment:

THis overhead will change if you're tracking function calls.

Mark:

Yes. It will change at link time and run time.

COmment:

You only want to track which functions are being used?

Robert:

We only want to know which functions were included in the binary.

Bill:

We think this will help us get to point of profiling.

Mark:

This doesn't mean a function is being used. It only means it was linked in. We want to avoid analysis during runtime.

Bill/RObert:

An example of retiring software at TACC is one of old PETSc versions.

Bill: We understand that including functions does not mean they were used.

Comment:

May see uncommon functions that aren't normally optimized.

Bill:

We have an example with VASP at TACC. VASP showed a lot of problems because they were using two uncommon routines. TACC's speaking with intel.

Comment:

Developer for HPCMP portal.

What scope is the tool for? Geared more for a site's POV.

Would like the user's to be able to see this in the web portal.

Would like the user's to see EVERYTHING they did on the system.

Want's to track the full user workflow.

Bill:

THis is a piece in that process, but not everything to track everything.

Mark:

THere is a user component. The environment checking is to help the user or support staff.

Library tracking may be useful for the users.

Comment:

Software is upgraded and user's build in new environment and can't reproduce results. Provided a command that queries env data from jobs.

Richard:

Nersc users can get that same info.

Luke:

If we have info, we can help users choose the appropriate library.

JohnC:

Information collected is valuable to users for provenance and reproducibility.

Comment:

Collect info in the binary and have a tool that allows users to dump info on the binary.(ldd?)

Limit user's choice of libraries by changing the dynamic libraries that are used.

Bill:

do you tell the user's no.

Comment:

No. The libraries are rpathed in.

Mark:

We put the linker interceptor on many machines without including the job launcher.

Comment:

If the data is collected and exposed with an API, the schedulers could provide some of it.

Comment:

If you publish an API slurm will adopt it.

We're done. :) hahaha

Comment:

Is there any automatic detection of problems?

Bill:

Yes for tacc stats. But we don't send info to users.

Mark:

Detection now is by hand.

Bill:

We will not automate notices to the user, but will automate notices to support.

Tim:

We could detect who was using the GPU via the linker info.

Bill:

When you find a query you like, you can turn it into a report.

Tim:

Can track the user's movement to a new compiler.

Comment:

Something to put into the wrappers is the ability to record certain env. variables.

Tim:

Catch many people running old scripts that don't use all the available cores.

Robert:

We know our system mainly runs chemistry codes. Does this change the way we setup or plan our systems?

Richard:

What about a.out? Could be anything.

Robert:

We take a sha1 hash of the executable.

Bill:

easy to change a sha1. BUt it's hard to recreate executables with the same sha1.

Richard:

Problems with many people running the same code in different places.

Robert:

We have a filter for code categories, e.g. vasp.

Richard:

Ask users to tell us what they're running.

Robert:

Can't make users tell us what they're running or use a wrapper when they don't have to.

Thought about hijacking exec. Which would collect everything.

Comment:

Tim from LLNL

20% of users are developers. 80% are users running the same code for 1-2 years.

Because of security concerns, would like to approach this on a user by user basis.

Can you do that?

Robert:

We can easily filter users. We can tell which are the top users in time, jobs, cores.

Bill:

We have 1600 projects/3200 users. 1600-2000 executables.

For top codes that have professional development, would we find problems? Or would they care if we find problems?

Comment:

Presentations will be attached to email list.

Robert:

Thanks

Comment:

Is anything public yet?

Mark:

ALTD is available

Robert:

Lariat is available.