TALES FROM THE TRENCHES: CAN USER SUPPORT TOOLS MAKE A DIFFERENCE?

Robert McLay

Doug James, Si Liu, Todd Evans, Bill Barth, Antia Lamas-Linares, Reuben Budiardja, Mark Fahey

1Texas Advanced Computing Center (TACC)
2National Institute for Computational Sciences (NICS)
3Argonne National Laboratory (ANL)
REFLECTS OUR COLLECTIVE EXPERIENCE SUPPORTING STAMPEDE USERS

Stampede: TACC flagship and XSEDE workhorse

- Delivering 2 million core-hours a day to thousands of active users from hundreds of institutions
- 6 million jobs and 2 billion core-hours in its first 33 months of production service
- Both high impact and representative of larger community's needs and experience

One of 6,400 compute nodes on the 9+ petaflop Stampede system at TACC
OVERVIEW

Support Tools
Case Studies
Ad Hoc Observations
XALT (Fahey, McLay) – Job-Level Usage Data

- Job-Level: build-time and run-time data on essentially every job
- Usage: what software and libraries do researchers actually use?
- Of particular value to decision-makers and support staff
- Separate HUST 2015 talk

Download: github.com/Fahey-McLay/xalt

<table>
<thead>
<tr>
<th>Software</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>intel/13.0.2.146</td>
<td>3520152</td>
</tr>
<tr>
<td>mvapich2/1.9a2(intel/13.0.079)</td>
<td>1170548</td>
</tr>
<tr>
<td>intel/14.0.1.106</td>
<td>611698</td>
</tr>
<tr>
<td>intel/15.0.2</td>
<td>543938</td>
</tr>
<tr>
<td>impi/4.1.3.049(intel/13.0.079)</td>
<td>167063</td>
</tr>
<tr>
<td>gc/4.9.1</td>
<td>133638</td>
</tr>
<tr>
<td>impi/4.1.0.030(intel/13.0.079)</td>
<td>103845</td>
</tr>
<tr>
<td>mvapich2/2.1(intel/15.0.2)</td>
<td>98403</td>
</tr>
<tr>
<td>phdf5/1.8.9(intel/13.0.079:mvapich2/1.9a2)</td>
<td>65302</td>
</tr>
<tr>
<td>cuda/5.5</td>
<td>52962</td>
</tr>
<tr>
<td>mvapich2/2.1(gcc/4.9.1)</td>
<td>48891</td>
</tr>
<tr>
<td>fftw3/3.3.2(intel/13.0.079:mvapich2/1.9a2)</td>
<td>48802</td>
</tr>
<tr>
<td>boost/1.55.0(intel/13.0.079)</td>
<td>45351</td>
</tr>
<tr>
<td>boost/1.55.0(gcc/4.7.1)</td>
<td>41794</td>
</tr>
<tr>
<td>opencv/2.4.6.1(intel/13.0.079)</td>
<td>39791</td>
</tr>
<tr>
<td>qt/4.8.4</td>
<td>33427</td>
</tr>
</tbody>
</table>
TACC Stats (Barth, Evans, Hammond) – Job-Level Performance Data

- Job-Level: performance snapshots on essentially every job
- Automatic filters generate daily reports on jobs that may require attention
- Web access to integrated dashboard view of any job

Download: github.com/TACC/tacc_stats
Lmod (McLay) -- User Ownership of the Environment

- Control: load, manage, use the software I need to do my research
- Repeatability: robust mechanisms to replicate software environment
- Visibility: know what's available under what conditions
- Protection: detect, prevent, eliminate common mistakes

Download: sourceforge.net/projects/lmod
Sanity Tool (Liu, McLAY, Wilson) – Account-Level Diagnostics

- Push-button assessment of account configuration
- Saves time and serves as general health screening: repeatable checks for common problems that are often notoriously hard to diagnose
- Evolves to reflect lessons learned

Download: sourceforge.net/projects/sanitytool

```bash
login1 62$ sanitycheck
Sanitytool Version: 1.1

1: Check SSH permissions:
   Failed
   Error: group permission on $HOME will cause RSA to fail!
   Error: other permission on $HOME will cause RSA to fail!

   Make sure you have a .ssh directory under your $HOME directory. You can use the following commands to set the proper permissions:
   $ chmod 700 $HOME
   $ chmod 700 $HOME/.ssh
   $ chmod 600 $HOME/.ssh/authorized_keys
   $ chmod 600 $HOME/.ssh/id_rsa
   $ chmod 644 $HOME/.ssh/id_rsa.pub

2: Check SSH Key:
   Passed

3: Check environment variables (e.g. HOME, WORK) and file system access:
   Passed

4: Check user’s queue accessibility (Stampede Only):
   Passed

5: Check allocation balance:
   Passed

6: Check quota for $HOME and $WORK:
   Passed

7: Check module environment:
   Passed

8: Check compilers:
   Passed

9: Check scheduler commands:
   Passed

-----------------------------
1/9 tests failed
-----------------------------
login1 63$
```
A single random “tip of the day” appears on each user’s console at the end of the login sequence.

Tips include command-line basics (Bash, Linux, editors, modules, etc.), best practices (change management, backups, job submission, etc.), system-specific info (e.g. Xeon Phi basics).

Tip 197 (See "module help tacc_tips" for features or how to disable)

If you don't need 48 hrs for your job, don't request that much: the job scheduler will have an easier time finding a slot for the 1 hour you really need.

...or...

Tip 4 (See "module help tacc_tips" for features or how to disable)

Use "!!" to repeat the most recent command. Use "!mpi" to repeat the most recent command that started with "mpi" and "!?mpi?" for the most recent one that contained "mpi".

Email TACC for access to source code.
Synergy through Integration

- **Optional Interfaces**
  - XALT supplies meta-data for TACC Stats
  - XALT prototype supplies usage data to XSEDE XDMoD dashboards
  - Lmod supplies module data to XALT

- **Related Functionality**
  - Lmod produces usage data (as in XALT)
  - XALT will soon check build-vs-run consistency (as in Lmod)

- **Cross-Pollination**
  - XALT syslog implementation informing Lmod migration
  - Tips database highlights other tools
Queue Wait Times: Targeted Outreach

- Dramatic increase in demand during 2015
  - Normal queue wait time went from 2-3 hrs to 24+ hrs
  - Largemem queue wait time reached 4+ days
- Scheduler tuning helped, but usage data was revealing
  - High percentage of jobs specified substantially more time than actually needed
  - XALT suggested opportunities to move some largemem users to normal queues by reducing tasks/node
- Interventions: steady, targeted outreach (email and tickets), tips database
- Results: demand’s still up, but queue times are down from their peaks (normal queue down 50%; largemem 75%)
Compiler Migration: Managing Change

- Original plan: remove four very old compilers and associated software stacks; install new one
- Usage data: three of four had light usage; oldest had 2000+ users
- Interventions: change in plan, targeted outreach, nag messages

```
login1 69$ module load intel/13.0.079
------------------------------------------------------------
There are messages associated with the following module(s):
------------------------------------------------------------
intel/13.0.079: This module is deprecated. It is marked for removal in Dec 2015. Please transition to a newer Intel compiler (intel/15.0.2 recommended) before this occurs.
```
Help Desk: Recurring Issues

- Recurring ticket: jobs fail to launch (mpi_spawn errors)
- ssh-related root causes difficult to diagnose unless you know what to look for
- Intervention: automation (job submission, account diagnostics)
- Recurring errors are the motivation behind both Sanity Tool and the build-vs-runtime consistency checks in Lmod/XALT.

```bash
login1 62$ sanitycheck
Sanitytool Version: 1.1

1: Check SSH permissions:
   Failed
   Error: group permission on $HOME will cause RSA to fail!
   Error: other permission on $HOME will cause RSA to fail!

   Make sure you have a .ssh directory under your $HOME directory.
   You can use the following commands to set the proper permissions:
   $ chmod 700 $HOME #(750 and 755 are also acceptable)
   $ chmod 700 $HOME/.ssh
   $ chmod 600 $HOME/.ssh/authorized_keys
   $ chmod 600 $HOME/.ssh/id_rsa
   $ chmod 644 $HOME/.ssh/id_rsa.pub

2: Check SSH Key:
   Passed
```
Job-Level Performance Data: Open-Ended Discovery

- Collect first and see what we learn
- Certain recurring patterns jumped off the page
- Led to auto-detection filters and daily reports: common problems, need for intervention
- Also learned to turn to job-level and user-level integrated dashboards before first response to a ticket

Sanitized (user names hidden)
Data-Driven Design: Benchmarking

- Analyzed 55,000 jobs with XALT and TACC Stats looking for general characteristics that could affect design decisions for next-generation systems
- Top codes cluster naturally in ways that amount to performance signatures
- Selected four representative codes as cornerstone of real-world benchmark suite; will begin to validate using Lonestar 5
- Issues remain: e.g. licenses that make it difficult to publish performance results
AD HOC OBSERVATIONS
OUR WORLDVIEW

- Labor hours are more precious than core-hours
- Support needs are not proportional to core-hours
- Formal training may not move the needle
- There’s too much actionable data
- Urgent need: put tools and data in the hands of users
- Integration is power
- Targeted outreach gets users’ attention
SUMMARY

- User support tools can and do make a difference
- Support at scale requires...
  - Targeting small allocations, inexperienced users
  - Putting data and tools in the hands of users
  - Putting a priority on labor costs
- We gratefully acknowledge NSF support through grants ACI 10-53575 (XSEDE), ACI-1134872 (Stampede), 1339708 (XALT), and ACI-1203560 (TACC Stats)

Ref paper: dx.doi.org/10.1145/2834996.2834998
LICENSE

© The University of Texas at Austin, 2015

This work is licensed under the Creative Commons Attribution Non-Commercial 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/

When attributing this work, please use the following text: "Tales from the Trenches: Can User Support Tools Make a Difference?", Texas Advanced Computing Center, 2015. Available under a Creative Commons Attribution Non-Commercial 3.0 Unported License"