PAWSEY supercomputing centre

Systems Monitoring & Benchmarking at Pawsey

Government of Western Australia | Australian Government | CSIRO | Curtin University | ECU Edith Cowan University | Murdoch University | The University of Western Australia

Ashley Chew – HPC Systems Administrator

INTRO TO LMOD

PAWSEY supercomputing centre

What is Lmod?

- LUA-based system that dynamically changes / manages the user environment though module files
  - replaces traditional TCL environment modules system
- Developed at Argonne National Lab and Texas Advanced Computing Centre
- Available in current versions of SLES, RHEL, …
- Freely available on GitHub
  https://github.com/TACC/Lmod

Lmod Gotchas

- Versions 6.x and earlier contain a flaw where the minor version of module couldn't be discerned
  - Recommend using version 7.5 or later
- When loading / unloading modules:
  - Lmod will want to know what files are there
  - Lmod will spider the module system creating a noticeable delay
    - Delay is accumulative.

	module load a b c *(only 1 loop)*
	module load a; module load b; module load c *(3 loops)*

Switching to Lmod

- Can bootstrap Lmod to overlay over default Modules environment
- Recommend only targeting selected users initially
  - Add Lmod to their start-up profile of shells
  - Test for a condition to bootstrap lmod
    - usually in /etc/profile.d/zz-lmod.{sh|csh}
  - Good for testing in production (Can toggle Lmod on / off for users)
- Existing TCL module files can be used
- Expect significant longer login times
  - Login now a 2-step process: native shell start-up + Lmod overlay

Switching to Lmod

- Recommend re-writing modules if you want to use extra features like module weightings or execute command
- Lua and TCL-based module files can co-exist
  - only one module must exist for a particular version

	gcc/4.4.3.lua gcc/5.4.0.tcl
	gcc/4.4.3.lua gcc/4.4.3.tcl

Lmod Caching

- Lmod will generate a cache to speed up successive logins
  - default cache validity is 24 hours
  - cache is generated typically $HOME/.lmod.d/cache
- global Lmod caching of all modules can increase speed *however*
  - a public cache won't allow users to see their own personal modules
- Lmod caching is not tiered, recommend doing it on the user level
- Be careful when $HOME is mounted on different clusters
  - cache contamination can occur
  - avoid by setting $LMOD_SYSTEM_NAME

Ashley Chew – HPC Systems Administrator

LESSONS LEARNT WITH XALT

What is XALT?

- job activity collector for a cluster
- tied implicitly to Lmod
- released and maintained by the same group as Lmod
- Open source and freely available
  - https://github.com/Fahey-McLay/xalt

Overview

- essentially a wrapper intercepting the linker and job launcher
- relies on Lmod
  - Uses module weightings feature so that its wrappers have 1st precedence
- scheduler independent
- Information captured as JSON files that can be saved in SQL database
- Injection into SQL database can be direct or indirect
  - Direct injection can create a DDOS event!
  - Recommend indirect injection
- Currently there is no public web front end

Mapping of binaries & libraries

- XALT uses a reverse map file to correctly map modules being used
- This reverse map file needs to be:
  - cluster aware
  - updated when new public software is added
- An orderly layout of your modules will help collating this map file
  - Users' private modules will not be picked up in the refresh of this map file
- Some programs are sensitive to Xalt wrappers
  - debuggers (Arm Forge, Intel Vtune)
  - MPI based programs which cannot be spawn by the scheduler and require a nodelist file to spawn job

Examples of stored information

SQL query of all modules with a linking to libmpi

```
SELECT distinct xalt_run.run_id, xalt_run.job_id, xalt_run.date, xalt_run.syshost, xalt_run.user,
xalt_run.exec_path
FROM xalt_run, xalt_object, join_run_object
WHERE xalt_object.object_path LIKE '%libmpi%' AND xalt_object.obj_id=join_run_object.obj_id
AND join_run_object.run_id=xalt_run.run_id
```

```
+--------+---------+--------------------+---------+-----------+-------------------------------------+
| run_id | job_id  | date               | syshost | user      | exec_path                           |
+--------+---------+--------------------+---------+-----------+-------------------------------------+
| 334 | 2562511 | 2018-07-23 16:43:07 | zeus| bskjerven | /group/pawsey0001/…/osu_bw |
| 340 | 2562541 | 2018-07-23 17:15:24 | zeus| bskjerven | /group/pawsey0001/../osu_bw |
| 341 | 2562542 | 2018-07-23 17:15:48 | zeus| bskjerven | /group/pawsey0001/…/osu_bw   |
+--------+---------+--------------------+---------+-----------+-------------------------------------+
```

PAWSEY supercomputing centre

SQL query of all library numbers

```
SELECT object_path, module_name, COUNT(date) AS cnt
FROM xalt_link, join_link_object, xalt_object
WHERE build_syshost='zeus' AND xalt_link.link_id = join_link_object.link_id AND
join_link_object.obj_id = xalt_object.obj_id GROUP BY object_path ORDER BY cnt DESC
```

object_path	module_name	cnt
/lib64/ld-2.22.so	NULL	276
/lib64/libc-2.22.so	NULL	276
…		
/lib64/librt-2.22.so	NULL	137
/pawsey/intel/17.0.5/…/linux/mpi/intel64/lib/libmpifort.so.12.0	intel-mpi/2017.0.4	137
/pawsey/intel/17.0.5/…/linux/mpi/intel64/lib/debug_mt/libmpi.so.12.0	intel-mpi/2017.0.4	124
/pawsey/intel/17.0.5/…/linux/mpi/intel64/lib/libmpicxx.so.12.0	intel-mpi/2017.0.4	121
/lib64/libgcc_s.so.1	NULL	103
/pawsey/intel/17.0.5/…/linux/compiler/lib/intel64_lin/libcilkrts.so.5	intel/17.0.5	103
/usr/lib64/libstdc++.so.6.0.24	NULL	103
/pawsey/sles12sp3/devel/gcc/4.8.5/gcc/7.2.0/lib64/libgcc_s.so.1	gcc/7.2.0	88

PAWSEY supercomputing centre

SQL query showing modules used

```
SELECT xalt_run.module_name, count(date) AS jobs,
ROUND(SUM(run_time*num_cores)/3600) as TotalSUs FROM xalt_run
GROUP BY xalt_run.module_name ORDER BY jobs DESC
```

module_name	jobs	TotalSUs
NULL	251	55
shifter/18.06.00	72	0
python/2.7.14	47	0

SQL query of how many times a binary has been run

```
SELECT user, exec_path, syshost, count(exec_path)
FROM xalt_run
GROUP BY exec_path
```

achew	/bin/bash	zeus	16
mcytowski	/group/pawsey0001/…/beam2016test	zeus	122
mdelapierre	/group/…/Develop-with-CUDA/query/Zeus/query	zeus	2
mdelapierre	/group/…/Develop-with-CUDA/query/Zeus/query2	zeus	3
mdelapierre	/group/pawsey0001/…/binary/cuda/9.0/bin/nvcc	zeus	1

Mohsin Ahmed Shaikh – Supercomputing Specialist

USING XDMOD AT PAWSEY

What is XDMod?

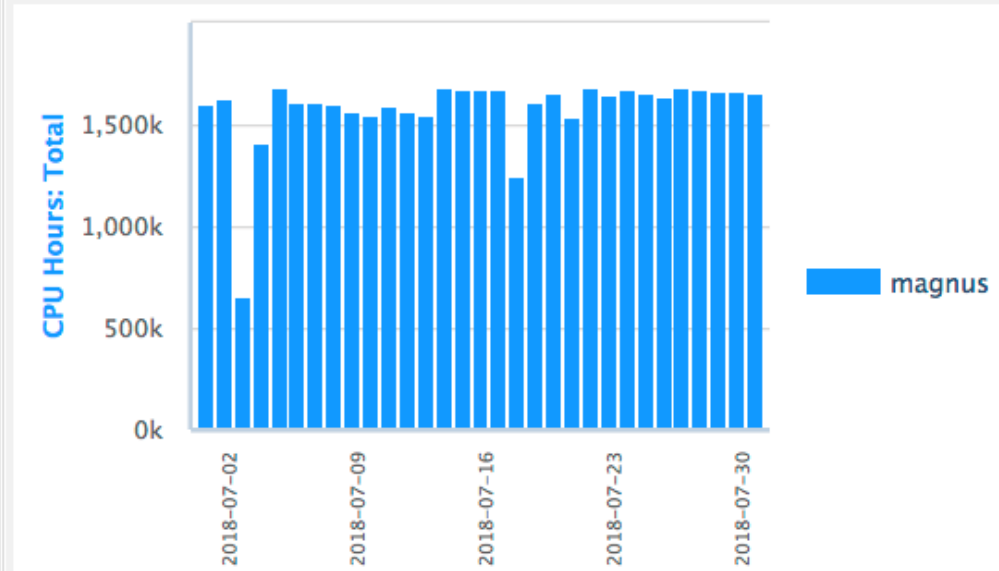
An open source tool to facilitate the management of HPC resources

http://open.xdmod.org

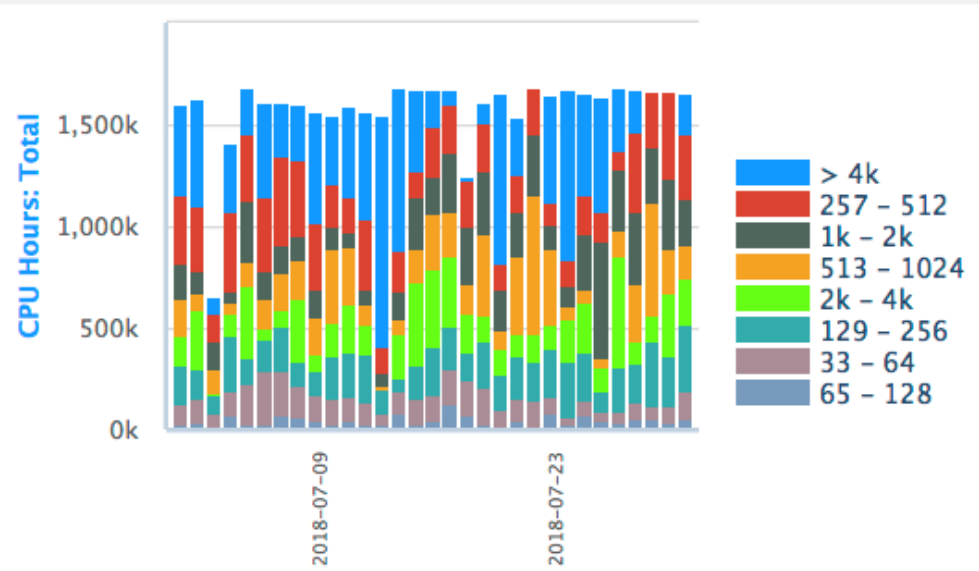
Integrates with various Resource Managers (Slurm, SGE, PBS, LSF)

Collects usage data from different clusters

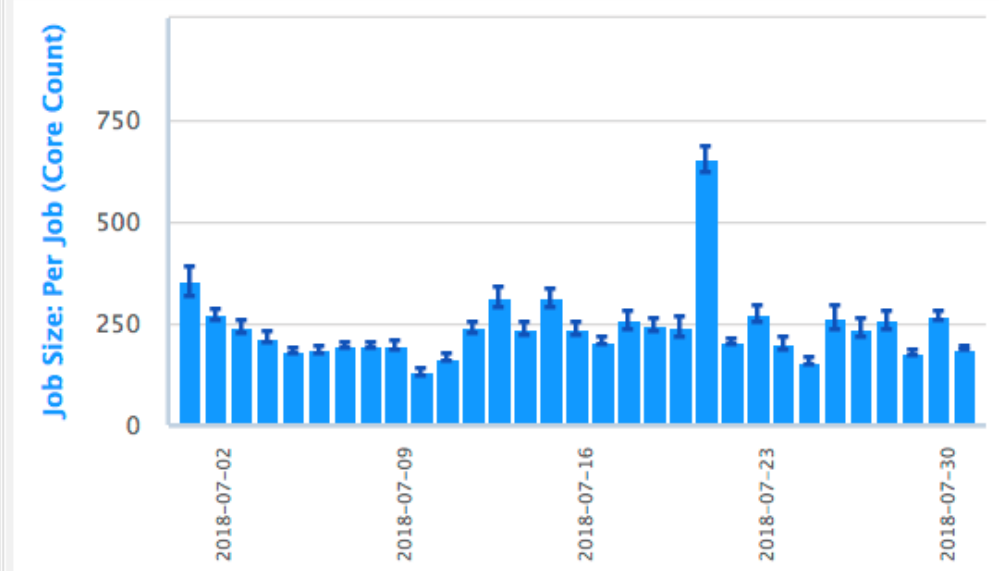
Total CPU Hours By Resource (Top 10)



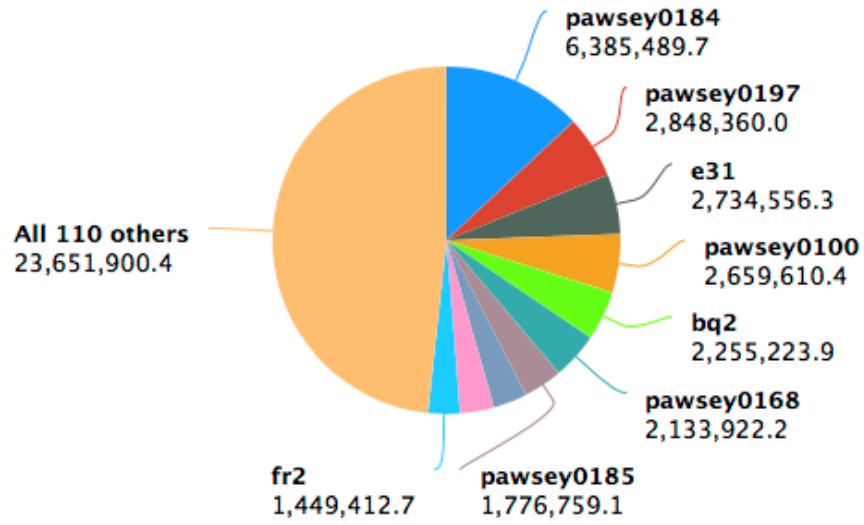
Total CPU Hours by Job Size



Avg Job Size (Core Count)



Total CPU Hours by PI



PAWSEY supercomputing centre

Use cases

- Inefficient resource consumption by users
  - user submitting 500-node jobs for small computational problem
- Identification of use policies
  - incorrect use of debug queues
- Creation of weighted service units for using different architectures
  - GPU, many-core (Xeon Phi), high memory

Mohsin Ahmed Shaikh – Supercomputing Specialist

APPLICATION TESTING

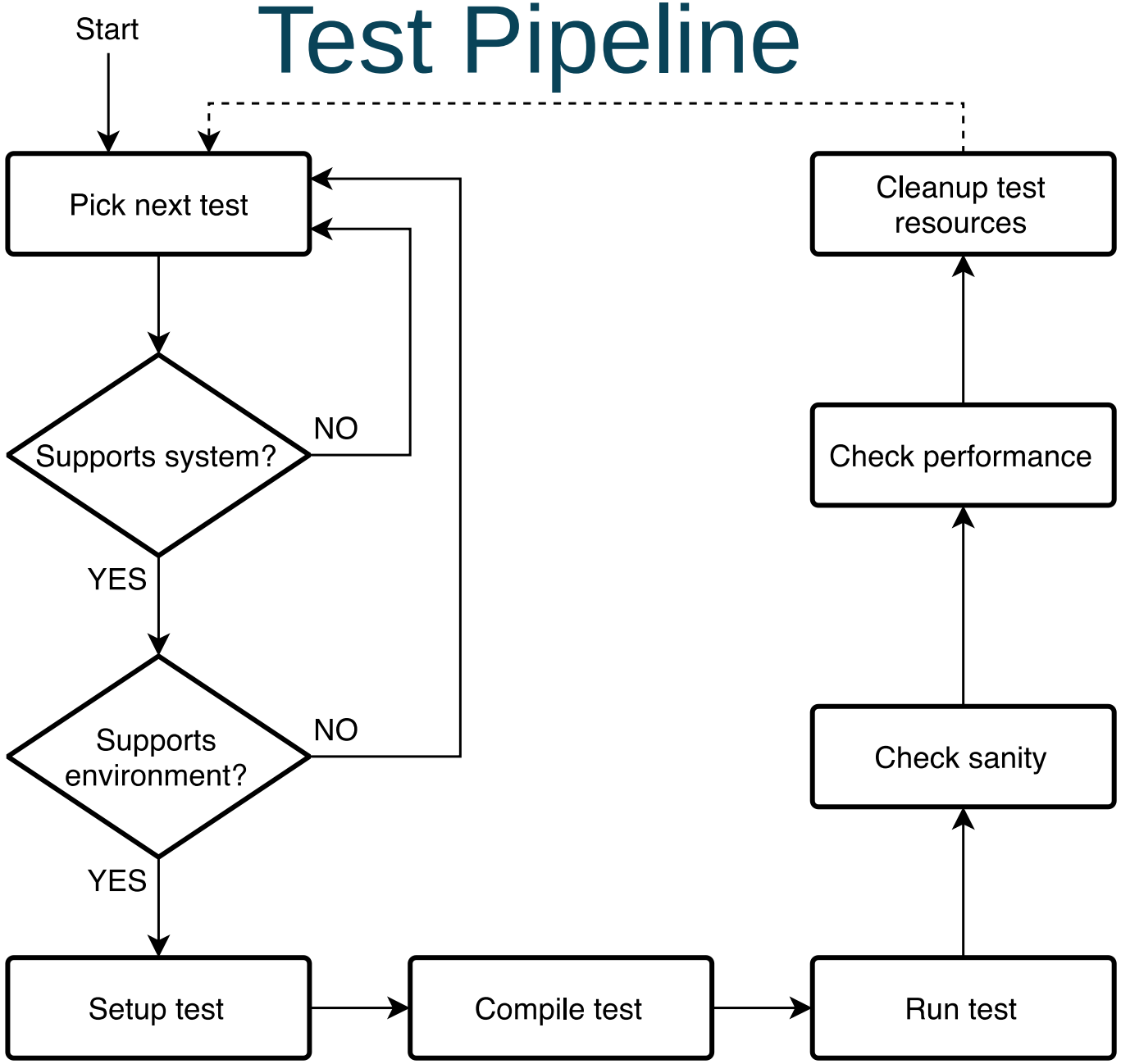
When to test?

- After a maintenance session
  - Sanity check of critical infrastructure
    - Job scheduler, Accounting, Interconnects, Filesystems ...
  - In *known* applications, check for changes in:
    - result accuracy, performance, reproducible
  - Has performance of specific subsystem(s) regressed?
- Frequent testing
  - Early detection of problems with compute/memory/IO
    - Daily tests

ReFrame

- Python 3-based regression test framework developed at CSCS
- Tests can be configured to be:
  - Ran on multiple systems and/or architectures
  - Run asynchronously
  - Include large jobs/workflows (it scales!)
- Allows sanity and performance checking
- Supports SLURM and PBS
- Well documented!

Test Pipeline



Sample output

```
Command line: ./bin/reframe -c pawseyapplicationtestsuite/pawsey_checks
                -t regression --exec-policy async -r --keep-stage-files
[==========] Running 16 check(s)
[----------] started processing lammps_small_scale (LAMMPS default system module check)
[ RUN      ] lammps_small_scale on magnus:workq using PrgEnv-gnu
[ RUN      ] lammps_small_scale on magnus:workq using PrgEnv-intel
[----------] finished processing lammps_small_scale (LAMMPS default system module check)
         .
         .
[----------] waiting for spawned checks to finish
[       OK ] lammps_small_scale on magnus:workq using PrgEnv-gnu
         .
         .
[     FAIL ] group on magnus:workq using PrgEnv-gnu
         .
         .
[       OK ] namd_cpu_check on magnus:workq using PrgEnv-gnu
[----------] all spawned checks have finished
[  FAILED  ] Ran 17 test case(s) from 16 check(s) (4 failure(s))
[==========] Finished on Mon Aug 27 13:42:28 2018
```

PAWSEY supercomputing centre

Sample output (contd …)

```
==============================================================================
SUMMARY OF FAILURES
------------------------------------------------------------------------------
FAILURE INFO for group
  * System partition: magnus:workq
  * Environment: PrgEnv-gnu
  * Stage directory:
/group/pawsey0001/mshaikh/Application_testsuite/reframe/stage/workq/group/PrgEnv-gnu
  * Job type: batch job (id=3060019)
  * Maintainers: ['mohsin.shaikh@pawsey.org.au']
  * Failing phase: sanity
  * Reason: sanity error: 1377.02 < 2000.0
------------------------------------------------------------------------------
```

Pawsey checks

Checks	Intent
Regression checks	Validating and checking performance of top applications at different scales
Performance checks	Synthetic benchmarks to test performance of subsystems e.g. microbenchmarks, IO benchmarks, mini apps
System checks	Testing standard user environment, scheduler, policy, and functionality (e.g. MPI, OpenMP, CUDA, OpenACC) on all systems and their nodes
Stress checks	Benchmark applications to stress subsystems to check bounds e.g. HPCC, Large IOR, MDTest, STREAM
Acceptance checks	Quick checks to run after routine maintenance before handing them back to users

Future Work

- Add continuous integration component
- HTML dashboard for quick system health summary