SC19 BoF: HPC System Testing: Procedures, Acceptance, Regression Testing, and Automation

KAUST Regression Testing

Bilel Hadri
KAUST Supercomputing Laboratory (KSL)
bilel.hadri@kaust.edu.sa

Shaheen2 Supercomputer

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

COMPUTE	Node	Processor type: Intel Haswell	2 CPU sockets per node, 16 processors cores per CPU,2.3GHz
		6174 Nodes	197,568 cores
		128 GB of memory per node	Over 790 TB total memory
	Prg-Env	Cray PrgEnv	KSL staff installation of 3rd party packages (about 200)
	Speed	7.2 Pflop/s speak theoretical performance	5.53 Pflop/s sustained LINPACK and ranked 7th in July 2015 Top500 list
	Network	Cray Aries interconnect with Dragonfly topology	57% of the maximum global bandwidth between the 18 groups of two cabinets.
STORE	Disk	Sonexion 2000	17.6 Petabytes of usable storage. Over 500 GB/s bandwidth
	Burst Buffer	DataWarp	Intel Solid Sate Devices (SSD) fast data cache. 1.5 Petabytes of capacity Over 1.5 TB/s bandwidth.

Since July 2015, Shaheen is used by as of today by:

- 80 KAUST faculty (Over 50% of KAUST Faculty)
- 14 Saudi Institutions(gvt, industry, academia)
- 999 users. Who will be the 1000th ?
- 440 projects lead by 140PIs
- 4.2 Billion core-hours consumed

Need to deliver the best computing environment to our users !

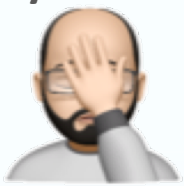
System performance and software assessments are critical !

REGRESSION TESTING is needed !

Motivations

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- On previous HPC systems at KAUST since 2009.
  - Acceptance test were run only once with basic tests.
  - Simple and basic functionality were checked only before releasing the system back to the users as soon as possible.
  - Issues were resolved after users complaints



- With Shaheen2 installation in April 2015,
  - Set detailed acceptance tests with expected functionality and performance
    - Identify potential hardware or software issues in a more rational & methodical way
    - Around 100 tests of functionalities and performance

Motivations

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Following Acceptance, a regression procedure has been adopted
  - Developed SRT: Shaheen Regression Testing
  - Gathered a set of well-defined tests from acceptance tests to systematically assess the actual state of the system.
  - Designed to run after each maintenance session or unscheduled downtime
  - Main tests are done manually and occasionally using Jenkins
  - Keep adding additional tests on new features or new workload of users

- **GOAL: Have Zero ticket/complaints about SW/HW by users following maintenance.**

Objective and Design

- **Objectives:**
  - Provide performance similar or beyond acceptance results.
  - Run the tests with no special privileges.
  - Analysis of the results by KSL team on whether or not to release the system to the users, based on the criticality of any issues detected
  - Enabling 'on-the-fly' performance evaluation and even earlier detection of potential issues.

- **Testing protocol :**
  - Component Tests:
    - Test the regular and basic of functionality of the system including the scheduler and programming environments
  - Synthetic Tests
    - Extremely well-localized performance runs: compute nodes, interconnect, filesystem
  - Typical Shaheen2 workload
    - Run real applications in short jobs

Component Tests

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Category	Purpose	How to test?
General	Connection	Try to login via ssh (do this test with each login node), tunneling…
	promptness of command line	How long for a regular shell command to return?
	check X-Windows	Does an X11 window open correctly when spawned from Shaheen front-end?
	check files	Are files accessible in /home, /lustre, /project, /scratch?
Licenses	Cray compiler	Can we compile a toy program with these compilers?
	Intel compiler	
	Commercial software	Can we run Totalview, DDT, Ansys ?
Scheduler	Availability	Check that all queues are up and running and record the number of nodes down
	Nominal use	Submit (1, 4-512, 510-1000, > 1000) -node jobs
		Submit from /project, from /scratch
	Stress	Measure the time needed to submit a job-array of 500 jobs. When running, cancel all of them.
	Scheduling	Check policies and QoS, partitions
	policies	
	Accounting	Check if the accounting is working
Programming Environment	Compilers	Compile a toy code with Cray, Intel and GNU compiler
	Libraries, modules	Link toy codes against petsc, perftools, hdf5 and netcdf libraries
	Monitoring	Check that the previous compilations have been recorded in the xalt database. Check that a toy program's IO behavior is tracked in Darshan.
Burst Buffer	Availability	Submit a job using the burst-buffer and check the queue status and all the functionalities

Performance Tests

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Category	Purpose	How to test?
Synthetic benchmark	Node performance	Short HPL per node wrapped by MPI. Full scale test
	Network	Test links. Full scale test
	IO	IOR performance
Burst Buffer	Performance	IOR performance
Applications	Performance and variability	VASP, WRF, SPECFEM, NEK5000

In average, when no issue detected, SRT last in average 1h30min.

Benefits

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

In the last 4 years, our acceptance and regression procedure has provided essential benefits:

1. **No hardware or software tickets/complaints related to the system for the next 24 hours after it is released to users.**
2. **An improved reproducibility of user experiments since the installation of the systems**
3. **Better collaboration between CS team, sys-admin and Cray on site team**
4. **More detailed history of observed hardware and software problems**
   → Allowing us to provide more accurate data to vendors about any performance degradation
5. **With the new functionalities, new users, we add new regression testing and adapt it for acceptance for new HPC acquisition**
6. **Compared results with other similar systems and detected issues (Hadri,Parsani CUG 2019 Paper)**
7. **Many success by detecting HW (faulty CPU, SSD) and SW bugs (SLURM, PrgEnv, Patches …)**

What is next ?

- Coordinate the effort with the community
- Make some benchmark test available to the community