Regression Testing on Shaheen Cray XC40: Implementation and Lessons Learned

Bilel Hadri[1], Samuel Kortas[1], Robert Fiedler[2], George S. Markomanolis[1]

[1]KAUST Supercomputing Laboratory (KSL)
[2]Cray Inc,

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

SHAHEEN
SUPERCOMPUTING LABORATORY

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

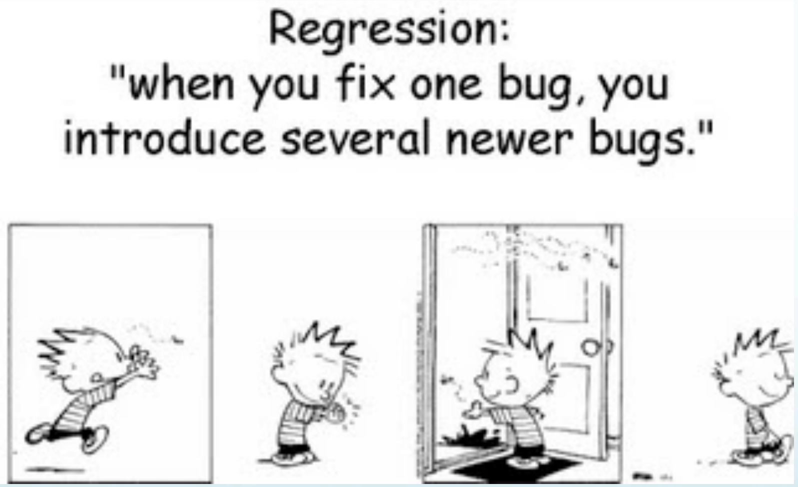
Shaheen Supercomputer

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
	Workload Manager	SLURM	Native
	Prg-Env	Cray PrgEnv	KSL staff installation of 3rd party packages (about 150)
	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 Lustre appliance	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.
	Archive	Tiered Adaptive Storage (TAS)	Hierarchical storage with 200 TB disk cache and 20 PB of tape storage, using a spectra logic tape library. (Upgradable to 100 PB)

Need to deliver the best computing environment to our users !
System performance assessments are critical ! REGRESSION TESTING is needed !

Regression Testing

- **What is Regression testing ? From Wiki:**
  - **Regression testing is a type of software testing that verifies that software previously developed and tested still performs correctly even after it was changed or interfaced with other software.** Changes may include software enhancements, patches, configuration changes, etc. […].
  - **The purpose of regression testing is to ensure that changes such as those mentioned above have not introduced new faults.**



Motivations

King Abdullah University of Science and Technology

- On previous HPC systems at KAUST(Blue Gene P/ 16 racks) since 2009.
  - Basic functionality of some system components was checked only before releasing the system back to the users as soon as possible.
- Since Shaheen2 installation in April 2015, a clear regression procedure has been adopted.
  - To identify potential hardware or software issues in a more rational & methodical way.
  - Set expected performance from the acceptance tests.
  - Gathered a set of well-defined tests to systematically assess the actual state of the system.
  - Designed to run after each maintenance session or unscheduled downtime
  - 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

Objective and Design

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

- **Objectives:**
  - No hardware or software tickets related to the system for the next 24 hours after it is released to users.
  - Provide performance similar or beyond acceptance results.
  - Run the tests with no special privileges.

- **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

→Guarantee good integration of all components (file system, compute nodes, and interconnect) while corroborating the synthetic test results.

Components Tests

Category	Purpose	How to test?
General	Connection	Try to login via ssh (do this test with each login node)
	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	It should not be possible to submit more than 800 jobs per user, more than 512 nodes occupied with jobs of 72 hours duration.
	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

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

Synthetic Tests

- Synthetic tests validating each crucial component of the system:
  - Compute nodes:
    - Checking the health and decent performance of any compute node
    - Submitting a one-node LINPACK test, wrapped into an MPI job to launch it across all nodes and check both performance and accuracy.
  - Interconnect: Aries Network
    - Evaluating the bandwidth of all links in any allocation of nodes. A Cray-developed topology-aware MPI program is used along with environment variable settings that enforce minimal-path routing.
  - File systems Lustre Sonexion 2000 and Datawarp
    - Executing IOR to check the bandwidth of the parallel file systems to be above 500 GBs/ and 1.5TB/s for Lustre and DataWarp respectively

Compute node test

- Check the performance of LINPACK
  - Using Intel optimized LINPACK Benchmark for a matrix size N = 55,000,
    - This size tests most of the memory and consistently yields near-asymptotic performance on the Haswell nodes
  - Wrapping into an MPI program that runs separate, identical LINPACK benchmark on each node
    - The interconnect is used only to determine the node on which each rank is running and to collect results for analysis and outlier identification
  - Gathering and sorting the results depending on performance.
    - Parsing the node number, CPU frequency, GFLOP/s performance and residual (accuracy)
    - Identifying nodes that perform significantly worse
  - Test completes within 6 minutes

LINPACK output

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

```
Node nid00008
Intel(R) Optimized LINPACK Benchmark dataCurrent date/time: Wed Mar 22 14:10:11 2017

CPU frequency:    3.599 GHz
Number of CPUs: 2
Number of cores: 32
Number of threads: 32

Parameters are set to:

Number of tests: 1
Number of equations to solve (problem size) : 55000
Leading dimension of array                  : 55000
Number of trials to run                     : 1
Data alignment value (in Kbytes)            : 1

Maximum memory requested that can be used=24201101024, at the size=55000

========= Timing linear equation system solver ========

Size LDA Align. Time GFlops Residual Residual(norm) Check
55000 55000 1 113.094 980.796 1.7961e-09 2.11745e-02   pass

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
55000  55000  1       980.7967 980.7967

Residual checks PASSED
```

Critical issues detected

- **Performance issue:**
  - **After each maintenance, 2-3 nodes are generally detected with performance lower than 935 GFLOP/s.**
  - **Usually memory issue is related to either runs with a performance near or below 550 GFLOP/s or when the execution time exceeds the wall clock limit of 10 minutes.**
- **Power capping issue::**
  - **from July 2015 to Dec. 2016, Shaheen was running under power & cooling constraints, using initially two static queues and later adopting SLURM dynamic power capping**
  - **Detected nodes that were not correctly configured with a performance under 800 GFLOP/s**

Critical issues detected (2)

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

- CPU frequency issue:
  - Randomly some nodes were set to a lower frequency after the nodes rebooted.
    - updates on CAPMC and SLURM
  - SLURM 17.02, a critical issue has been detected, srun would inadvertently set the CPU frequency maximum to the minimum value supported on the node.
    - The result obtained by the node performance test showed only one node that was capped at 1.2GHz, while the rest of nodes reached expected performance. Nevertheless, when testing several nodes individually, all of them reported a low performance with a CPU frequency set at 1.2GHz.

CRAY

slurm
workload manager

Critical issues detected (3)

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

- Correctness:
  - Performance of a given node is in the acceptable range, however the residual is above the threshold, which means that the answer is incorrect.
  - This typically corresponds to a faulty socket with inaccurate results that will impact dramatically any scientific results.
  - Since production, 12 sockets have been detected as faulty and sent back to Intel for further analysis by Cray on-site engineers.

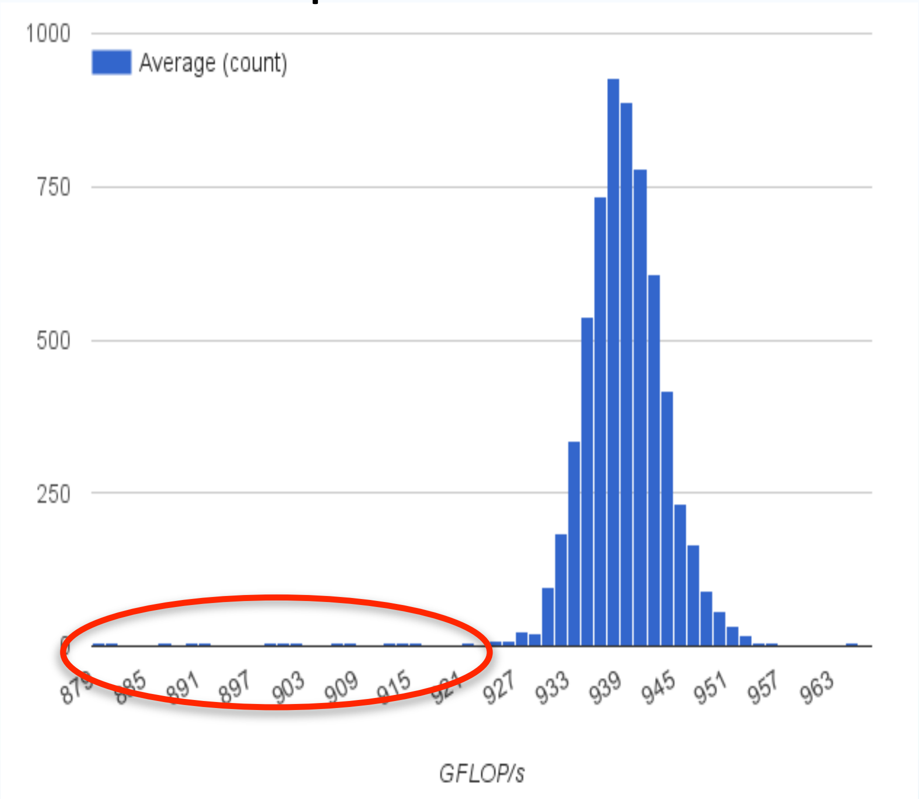
Critical issues detected (5)

- Thermal issue:
  - Examining the performance data stored so far, the overall performance of the nodes is quite stable and does not vary.
  - However, during the iteration of tests of the same node to validate the results, variation from 910 back to 990 GFLOP/s.
  - This issue is typically linked with a thermal issue .Cray on-site engineers to determine the faulty socket needing to be replaced.
  - These sockets impact application performance reproducibility.

- Performance decrease over time !

Performance Issue

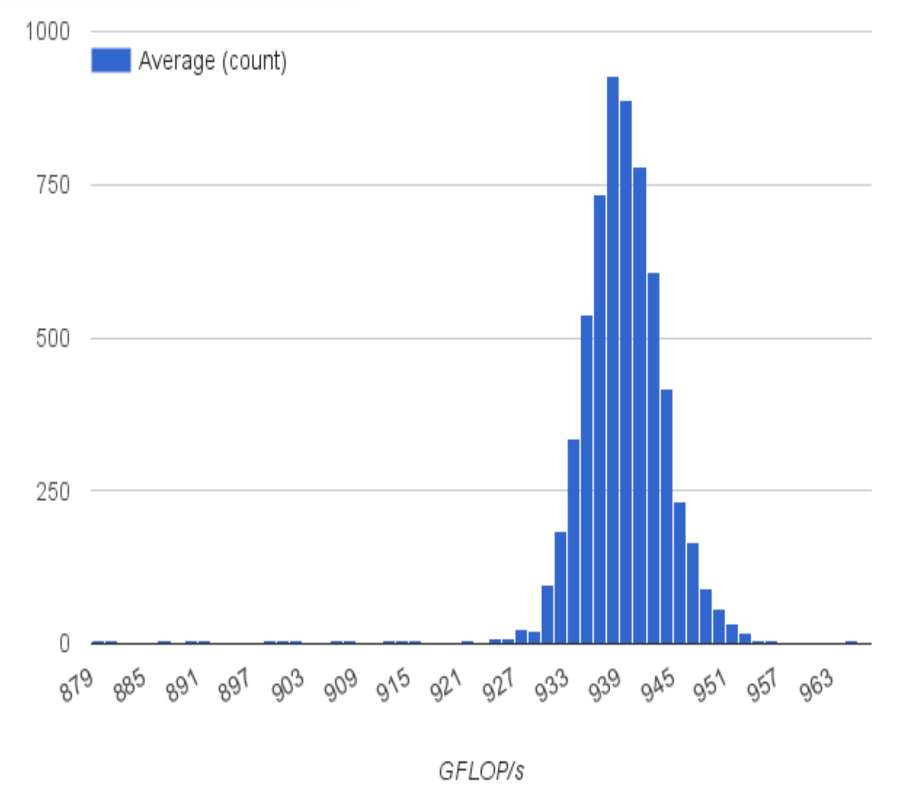
LINPACK performance March 2016

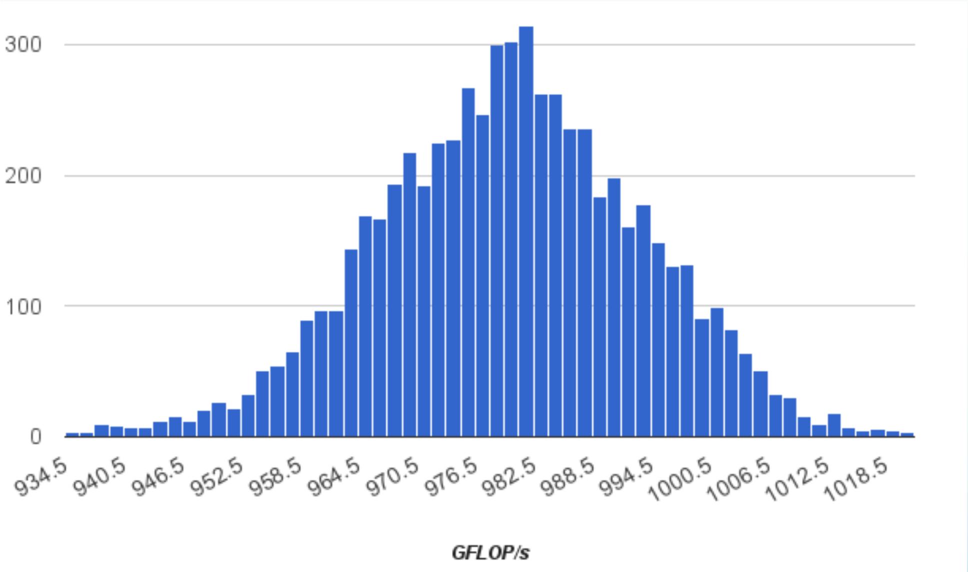


- **Over time, it has been noticed that more and more nodes are performing below this range, and some of the nodes reached a poor performance of 879 GFLOP/s (less than 75% of peak)**
- **Around 100 nodes with a performance lower than 930 GFLOP/s.**
- **This is far from the expected performance, and thus the scientists aiming at applications targeting performance would not be able to exploit fully the potential of the Cray XC40.**

Trimming Effect

King Abdullah University of Science and Technology



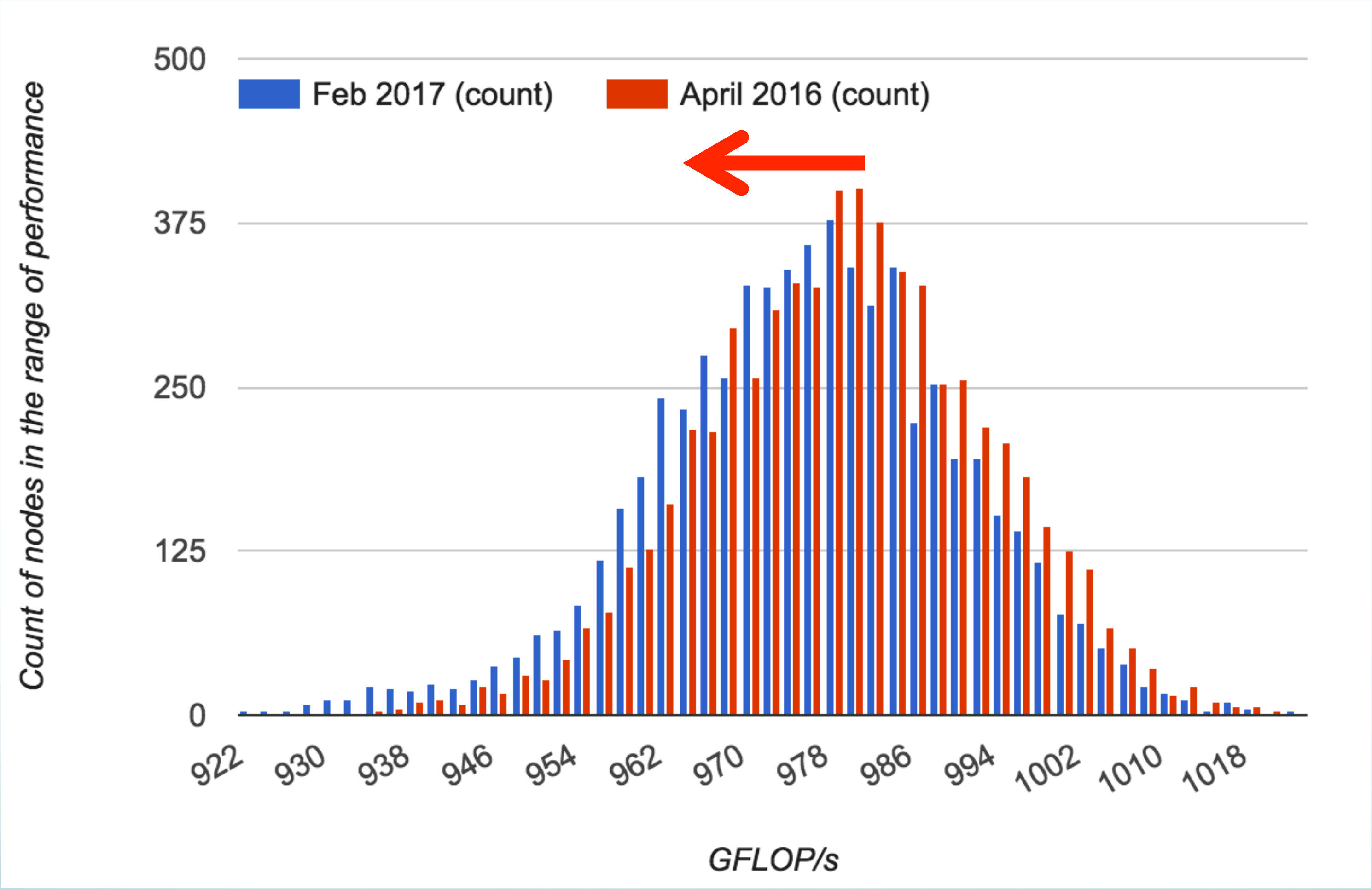


LINPACK performance March 2016

LINPACK performance April 2016

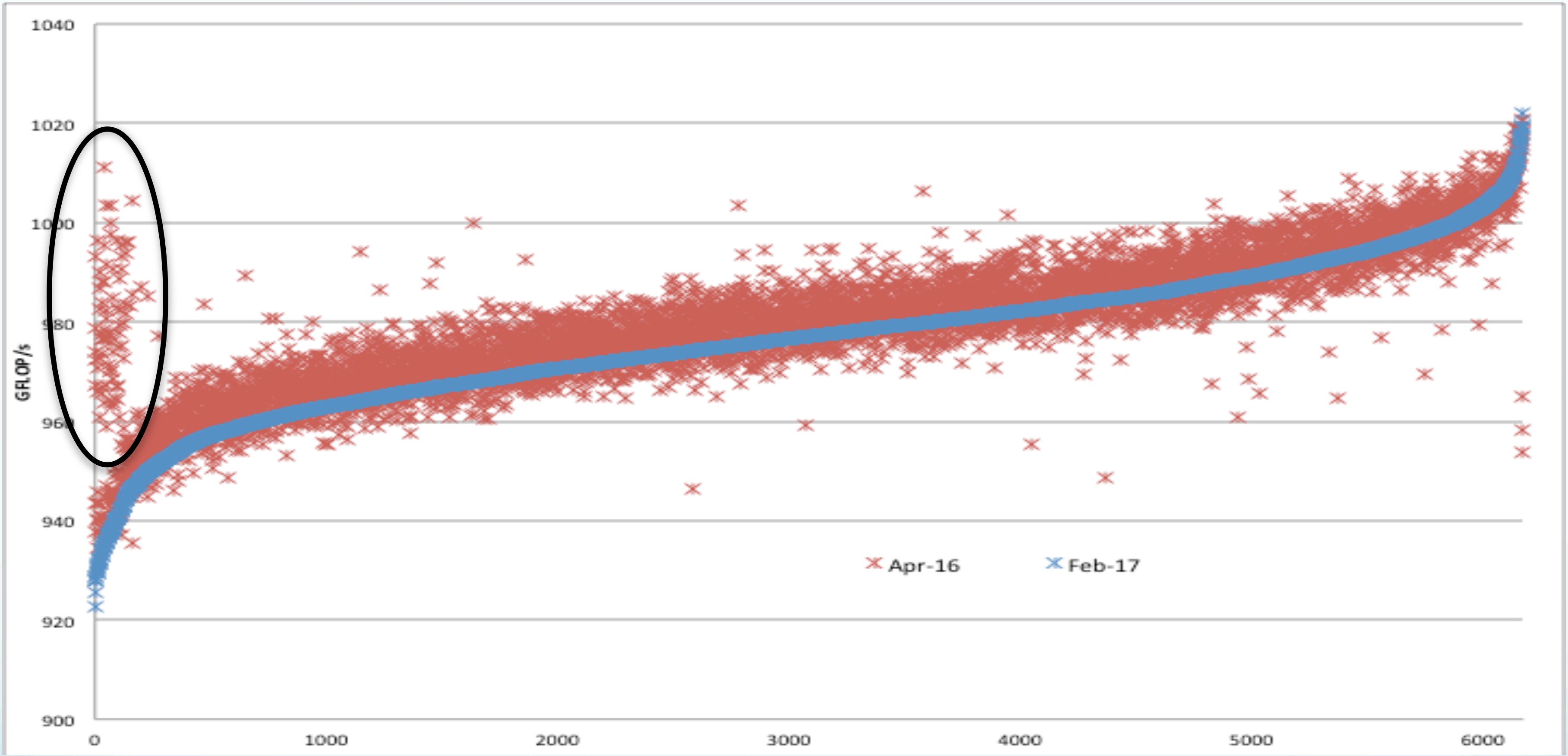
- **On April 2016, requested Cray to perform the trimming procedure on all Shaheen nodes**
- **Much improved performance was reached, with an enhancement up to 10%, with a range of performance distribution from 935 to 1025 GFLOP/s with an average of 980 GFLOP/s**

Node Performance Variability (1)



Even tough the average of all nodes is quite stable (only 0.4% of variation, from 980 to 976 GFLOP/s), we clearly observe a shift of the majority of nodes towards a lower value, toward the left side for the February 2017 runs.

Node Performance Variability (2)



Several hundred of nodes that lost close to 8% of their original performance from as high as 1015 GFLOP/s down to 940 GFLOP/s.

Test links

- **Test_links was originally developed for the Gemini network on Cray XE/XK systems for Blue Waters.**

- **Cray recently redesigned for Cray XC systems for Aries interconnect for KAUST needs to validate the HSN health.**

- **Test_links evaluates the bandwidth of all interconnect links in any allocation of nodes and identifies links with lower (by a specified amount) than the best or the average bandwidth for links of the same type.**

- **Bandwidths for each link are also recorded in tables for comparison between sets of results obtained at different times, so that one may determine whether and how individual link performance has changed over time.**

Test links (2)

- Four different types of links are evaluated, including:
  - the PCIe links from the four compute nodes on a blade to the single Aries router on that blade,
  - the copper links between blades in the same chassis,
  - the copper links between blades in different chassis of the same group,
  - the optical links between groups.
- Test completes in about 7min using 6174 nodes.

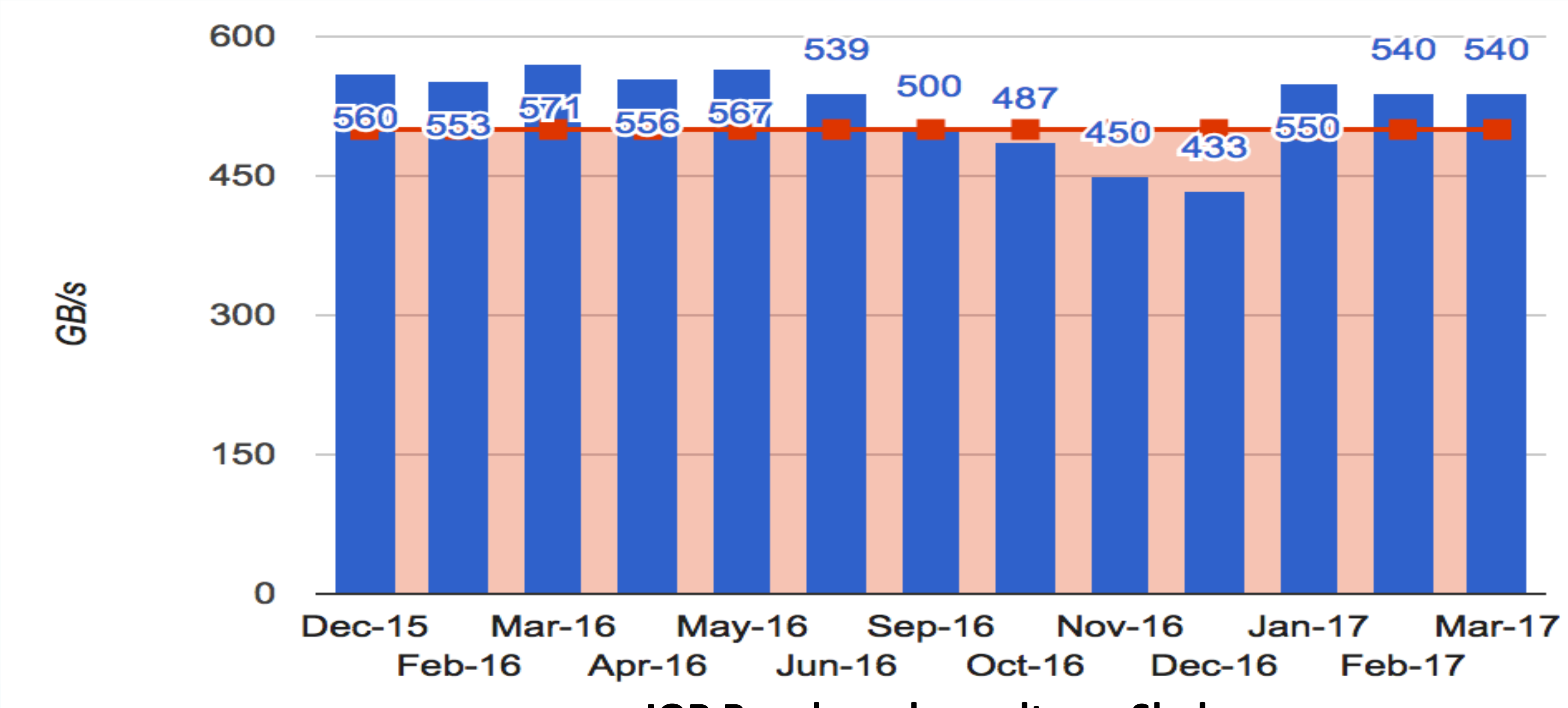
	Description	Expected Performance
Dimension 1	Optical links between groups	60 GB/s
Dimension 2	Copper links between different chassis	8.5 GB/s
Dimension 3	Backplane within a chassis	3.5 GB/s
Dimension 4	PCIe connections from nodes to aries router	5 GB/s

Test links (3)

```
ANALYSIS OF RESULTS FOR ARIES DIMENSION  2

 For aries with            3  available compute nodes:
 Highest  bandwidth for   3 nodes      8.19175
 Lowest   bandwidth for   3 nodes      8.10847
 Average  bandwidth for   3 nodes      8.15648
 Std. dev.     for   3 nodes     0.241659E-01

 For aries with            4  available compute nodes:
 Highest  bandwidth for   4 nodes      8.55541
 Lowest   bandwidth for   4 nodes      4.34258
 Average  bandwidth for   4 nodes      8.51178
 Std. dev.     for   4 nodes     0.772598E-01


OUTLIER(MORE THAN 5.0% BELOW AVERAGE)FOR ARIES DIMENSION 2
NID  tx ty tz  NID  tx ty tz BdwidthGB/s nodes % deviation
1644, 4,1,11,  1775,  4,3,11,   4.34258      4       48.98
1708, 4,2,11,  1775,  4,3,11,   6.59278      4       22.55
1775, 4,3,11,  1644,  4,1,11,   4.34258      4       48.98
1775, 4,3,11,  1708,  4,2,11    6.59278      4       22.55
```

→ **Need to disable blade containing nid01755 and repeat the test**

IOR tests on Sonexion 2000



IOR Benchmark results on Shaheen

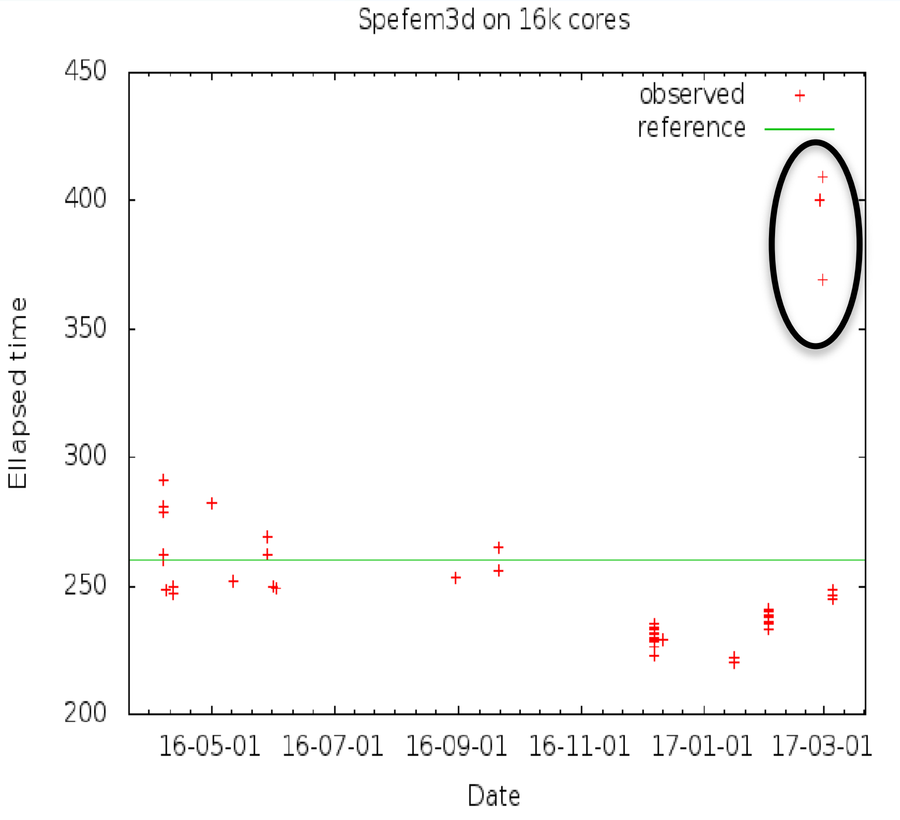
- **Performance decrease due to several issues:**
  - **Usage (up to 70% used), client update parameters, A variation on the load on individual OSTs of up to 20% was observed, and some of them were at close to 85 % of capacity**
- **January 2017, moved manually all large files directly to TAS**

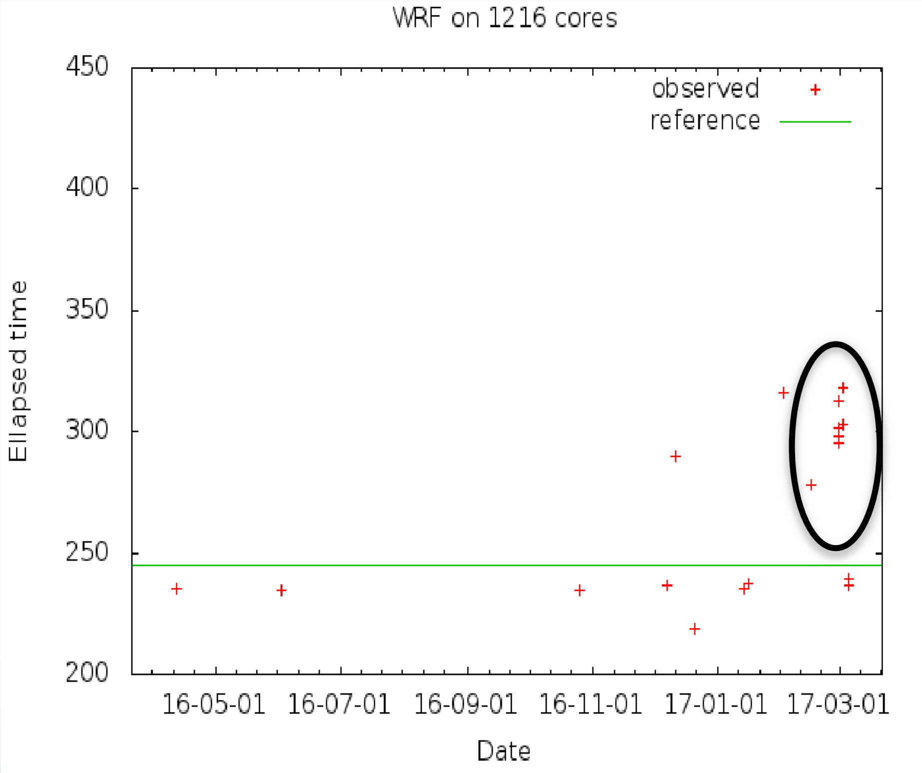
Performance Tests using Scientific Applications: Integration tests

- At the end of our testing protocol, typical use cases using actual applications on actual data sets representative of the Shaheen workload are executed.
- All tests have in common the process of compiling and running through the scheduler, but they stress diverse components of the environment (I/O, compute power in memory use, and network).
- Out of the ten application tests available, we usually pick 4 or 5 of them to confirm the overall stability and availability of the whole environment.

Applications Performance

King Abdullah University of Science and Technology

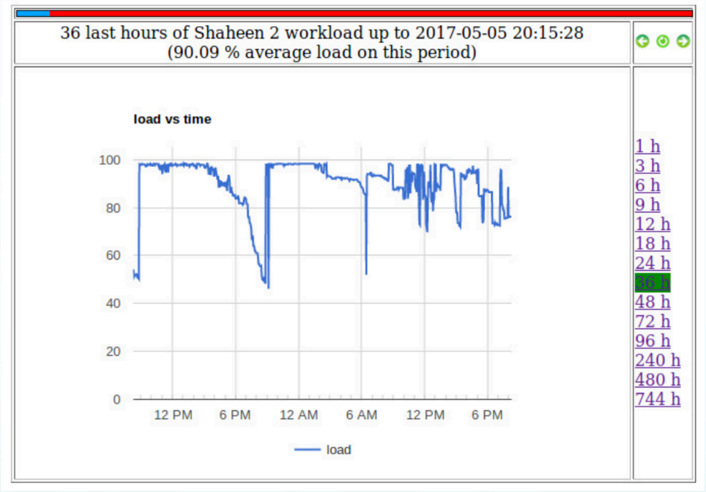




Notice a classical pattern of degraded times observed at the beginning of a regression step, helping us to confirm that a problem needs to be fixed, and the nominal result obtained again once the problem has been solved.

Automated Regression Framework

- A three-component monitoring environment
  - A Jenkins integration server for continuous monitoring of SLURM scheduling environment.
  - A Python extracting script that computes the current load on Shaheen 2 from Jenkins log files, and stores this information in a MYSQL table as tuples (timestamps, load, Jenkins_job).
  - A PHP/Jquery based website allowing easy browsing over time of the load history



- Self-described testing framework
  - To complement these monitoring tools, KTF, (KAUST Testing framework), written in Python allows one to describe, store, run, monitor and collect the results of a given test in a very straightforward way

Benefits

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

In the last 2 years, our use of this regression procedure has provided four essential benefits:

1. **A drastic decrease of user tickets received soon after a downtime**

   → Provide a better service to users.

2. **A significant gain in performance**

   → Observed up to a 10% performance improvement on a full scale code

3. **An improved reproducibility of user experiments run at large scale**

4. **More detailed history of observed hardware and software problems**

   → Allowing us to provide more accurate data to vendors about any performance degradation

Conclusions

- Successfully provided a clean environment with near-zero ticket issue received within 24 hours following a maintenance.
- This protocol takes on average 1 hour and 30 minutes
- Some tests have already been included in the Cray testing suite and adopted by Cray on-site engineers.
- An automated version of this process is currently under testing, enabling ‘on-the-fly’ performance evaluation and even earlier detection of potential issues.
- Test_links is available. Ask your Cray on-site staff.
- A first version of the components of this framework is planned to be released to the community soon.


THANKS
© Shadi Khoja Photography 2012