# Overview of the KAUST's Cray X40 System – Shaheen II

Bilel Hadri, Samuel Kortas, Saber Feki, Rooh Khurram, Greg Newby*
KAUST Supercomputing Laboratory (KSL)
King Abdullah University of Science and Technology (KAUST)
Thuwal, Saudi Arabia
Email:{bilel.hadri;samuel.kortas,saber.feki;rooh.khurram;greg.newby}@kaust.edu.sa
*Compute Canada, Toronto, Ontario Canada

***Abstract*— In November 2014, King Abdullah University of Science and Technology (KAUST) acquired a Cray XC40 supercomputer along with DataWarp technology, a Cray Sonexion 2000 storage system, a Cray Tiered Adaptive Storage (TAS) system and a Cray Urika-GD graph analytics appliance. This new Cray XC40 system installed in March 2015, named Shaheen II, will deliver 25 times the sustained computing capability of KAUST's current system. Shaheen II is composed of 6174 nodes representing a total of 197,568 processors cores tightly integrated with a richly layered memory hierarchy and dragonfly interconnection network. Total storage space is of 17 PB with additional 1.5 PB dedicated to burst-buffer. An overview of the systems specifications, the challenges raised in term of power capping, and the software ecosystem monitoring the usage will be presented and discussed.**

***Keywords-ShaheenII; KAUST, Cray XC40, Sonexion, TAS, DataWarp, monitoring***

## I. INTRODUCTION

Located on the shores of the Red Sea, King Abdullah University of Science and Technology (KAUST)[1], is an international, graduate research university in Saudi Arabia, dedicated to advancing science and technology through interdisciplinary research, education and innovation. Students, faculty, scientist and engineers conduct fundamental and goal-oriented research to address the world's pressing scientific and technological challenges. KAUST has established Core Labs and Major Facilities in order to support the research of faculty, scientists and graduate students on campus as well as regional and international academic collaborators around the world. One of the Core Labs responsible for providing HPC solutions at KAUST, is the KAUST Supercomputing Laboratory (KSL). Recognizing HPC as a key enabler of discovery across all fields of science, KAUST procured, a 16-rack IBM Blue Gene/P system in 2009, named Shaheen I, which was listed in the Top 500 [2] in June 2009 as the 14$^{th}$ fastest system in the world with an achieved performance of 190.9 Tflop/s.

In November 2014, the acquisition of a large Cray XC40 system was announced. The new XC40 system, called Shaheen II, expected to be available for general usage by June 2015, will increase the compute power at KAUST by a twenty-five fold. Along with the Cray XC40 compute system, KAUST will deploy a Cray Sonexion 2000 storage system, a Cray Tiered Adaptive Storage (TAS) system and a Cray Urika-GD graph analytics appliance. With this investment, KAUST will significantly augment its world-class academic and research facilities and capabilities to advance scientific discoveries.

## II. SYSTEM OVERVIEW

In this section, the Shaheen II HPC solution will be described from three different angles and complementary uses of the system, namely computing, storage and analytics.

### *A. Computing*

| | | |
|---|---|---|
| Main Components | 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 |
| Power | Up to 2.8MW | Water Cooled |
| Weight/Size | More than 100 metrics tons | 36 Compute cabinets, plus disk, blowers, management , etc.. |
| Speed | 7.2 PFLOS/S peak theoretical performance | Over 5 PFLOPS/S sustained LINPACK |
| Network | Cray Aries interconnect with Dragonfly topology | 57% of the maximum global bandwidth between the 18 groups of two cabinets |
| Storage | Sonexion Lustre appliance | 17.6 petabytes of usable storage. Over 500 GB/s bandwidth |
| | DataWarp Burst Buffer | Solid Sate Devices (SDD) fast data cache. Over 1 TB/s bandwidth, ( delivery September 2015) |
| | Tiered Adaptive Storage (TAS) | Hierarchical storage with 200 TB disk cache and 20 PB of tape storage, using a spectra logic tape library |

Figure 1. Specification of Cray XC40 Shaheen-II.

Fig. 1 summarizes the specifications of Shaheen II system. The system has 6,174 dual sockets compute nodes based on 16 cores Intel Haswell processors running at 2.3GHz. Each node has 128GB of DDR4 memory running at 2300MHz. Overall the system has a total of 197,568 processor cores and 790TB of aggregate memory.

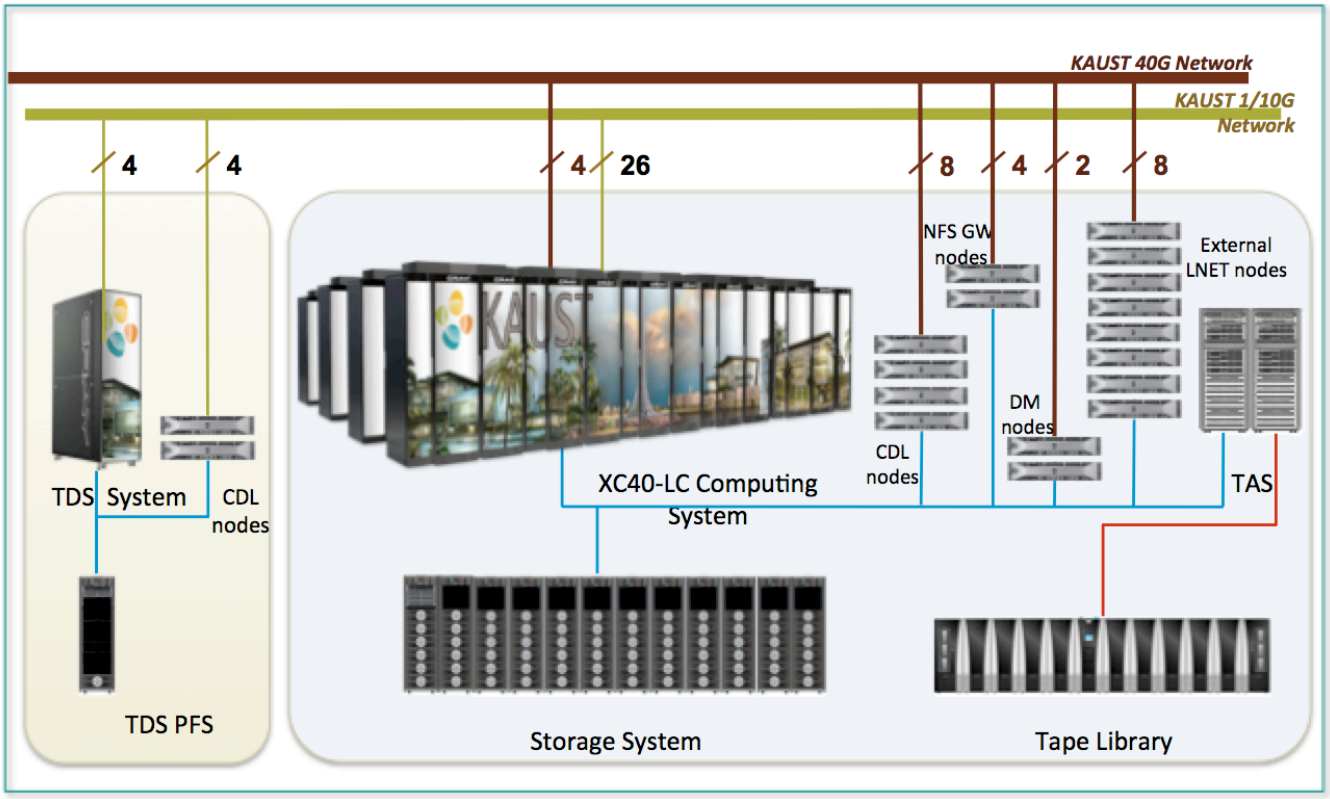


Figure 2. Overview on the KAUST Shaheen II systems networks.

The compute nodes are housed in 36 water-cooled XC40 cabinets, and connected via the Aries High Speed Network (HSN). The HSN is configured with 8 optical network connections between every pair of cabinets achieving therefore 57% of the maximum global bandwidth between the 18 groups of two cabinets. This will allow the design of the future upgrade with additional cabinets to accommodate more optical links between all cabinets with the same level of connectivity, i.e. 8 optical network connections between every pair of cabinets.

Shaheen II is delivering over 7.2 Pflop/s of theoretical peak performance and is capable of achieving over 5 Pflop/s of sustained LINPACK performance.

The design of Shaheen II was constrained by the availability of power - 2.3 MW, in the data center. During the acceptance test, 2.8 MW will be temporarily available for the LINPACK test. For normal production only 2.3 MW is available, until decommissioning Blue Gene/P system by the end of 2015.

In addition to the HSN in the XC40, Fig. 2 shows the network connectivity between the Shaheen II system and the network infrastructure at KAUST. The 40 GigE connections to other systems across the campus will be useful for mounting the parallel filesystem as NFS mount for example. The GPFS filesystem of Shaheen I will also be mounted on the XC40 via data mover servers to facilitate scientific data migration between the two systems.

### *B. Storage*

KAUST's system includes richly layered data storage architecture. The main data storage solution is a Lustre Parallel file system based on Cray Sonexion 2000 with a usable storage capacity of 17.2 PB delivering around 500 GB/s of I/O throughput. The Cray Sonexion 2000 installation is configured using 72 high performance Scalable Storage Units (SSU) and 144 Object Storage Services (OSS) with 4TB drives connected to the XC40 via 72 LNET router service nodes evenly distributed across the 36 cabinets.

The backup and archiving will be enabled with a Cray Tiered Adaptive Storage (TAS) system, which consists of a tape library with a total capacity of 20 PB upgradable to 100 PB. The TAS solution is paired with the TAS Connector for Lustre to provide a unique and tightly integrated solution for tiered data management directly from the proposed Lustre file system. This will provide higher I/O throughput to the tape library, with a 200 TB disk cache included as an automatic buffer between the Lustre filesystem and the TAS solution.

A Solid Sate Devices (SSD) based burst buffer solution using Cray DataWarp technology will be added in fall 2015 to provide I/O intensive applications with an additional storage layer providing over 1.2 TB/s of I/O bandwidth with close to 1.5 PB of capacity.

### C. *Analytics*

With this compute capacity and capability, nicely layered data storage infrastructure and cross-campus network connectivity, it is expected that KAUST scientists will be able to generate large amount of scientific data. Data Analytics using state of the art solution will become a pressing need for the user to analyze all scientific output of simulations processed on the Shaheen II system. A Cray Urika-GD data analytics appliance is installed at KAUST for the challenges of the discovery process, transforming these massive amounts of seemingly unrelated data into relevant insights and new scientific breakthroughs. The Urika-GD appliance consists of:

- Graph analytics platform, providing graph-optimized hardware with 2TB of global shared-memory, 64 Threadstorm4 processors with 128 hardware threads per processor and more than 75 TB of Lustre parallel filesystem for scalable I/O,
- Graph analytics database, providing an RDF triplestore and SPARQL query engine, and
- Graph analytics application services, providing management, security and data pipeline functions.

## III. KAUST SCIENTIFIC APPLICATIONS OVERVIEW

KAUST Supercomputing Laboratory's (KSL) computational workload is diverse in nature similar to most HPC centers. There are few users exploring unprecedented spatial and temporal resolutions, thus stretching the supercomputer to its limit, and there are traditional capacity users demanding compute cycles. Typical KAUST applications are shown in Table I.

TABLE I. TYPICAL KAUST APPLICATIONS

| Science Area | Codes |
|---|---|
| Atmospheric Modeling | WRF,WRF-Chem,HIRAM |
| Ocean Modeling | WRF, MITgcm |
| Combustion | NGA, S3D |
| CFD/Plasma | Plasmoid – in house code |
| Biology | In-house genomic motif identification code |
| Earthquake Seismology | SORD , SeisSol, SPECFEM_3D_GLOBE |
| Electromagnetism | In house explicit code |
| Big Data/ | Mizan - in house code (Analysis of Large Graphs) |
| Chemistry | VASP, LAMMPS, Gaussian, WEIN2k,Quantum Espresso |
| Seismic imaging/Oil & gas | In house 3D reverse time migration code |

Shaheen I configuration is favorable for applications that can scale to large core counts but not suitable for compute bound capacity applications. Limited memory of 4GB/node is also a limiting factor for reverse time migration, gene sequencing, and graph analytics codes. Moreover, most of the pre-built commercial codes would not run on BGP architecture. Shaheen I certainly had been extremely useful for certain class of algorithms. It has even served traditional capacity users like computational chemists, as evident from Fig 3, which shows the breakdown of Shaheen I utilization.

Machine has been heavily used by environment, combustion/CFD, and chemistry users. Earth scientists and biologists did not use Shaheen I heavily, mainly because of the abovementioned reasons.

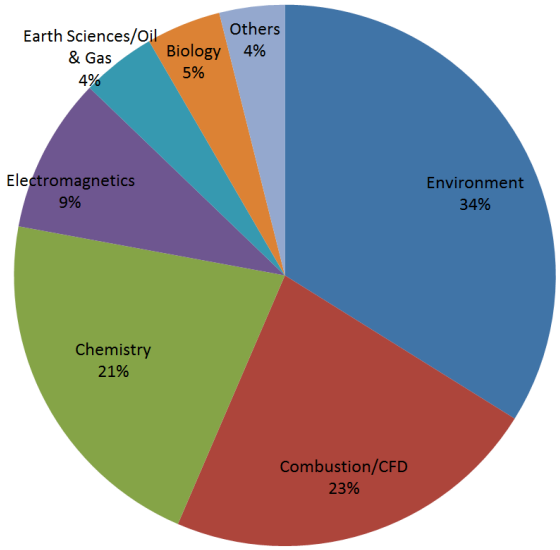


Figure 3. Shaheen I Utilization by Science Area (2009-2015).

Climate research group at KAUST [3] uses WRF to investigate dust storms, life cycle of dust particles in the boundary layer, entrainment of dust from the surface, development of elevated dust layers, and long-range dust transport. Typical mesh resolution is 510 by 435 by 40, time step is 60 seconds, and the total simulation time is in the range of 1 to 3 months.

An in-house 3D magneto-hydrodynamics (MHD) code developed at KAUST is used to study Plasmoid dynamics in magnetic reconnection [4]. "First of its kind" simulations performed on Shaheen I helped explain the discrepancy in the reconnection rate between theory and observations in nature.

NGA[5] is used to study turbulent mixing in strong shear flows, turbulent aerosol with Lagrangian particles, and turbulent soot flame with complex chemistry [6].

Computational Electromagnetism (CEM) research group at KAUST [7] develops novel frequency and time domain integral equation solvers for characterizing electromagnetic wave interactions on electrically large and multi-scale structures and applies them in problems of electromagnetics and photonics.

The infocloud group [8] at KAUST stores, queries and extracts knowledge from large graphs and sequences. Some of the application areas include: time series stock market data, web log analysis, bioinformatics – DNA, and text mining. The group developed Mizan [9], which is the fastest Pregel[10] implementation code, a graph analysis infrastructure introduced by Google.

Ocean modelling work [11] at KAUST is selected to lead the task of Red Sea modeling and forecasting in the

ARAMCO-KAUST Red Sea project. The model is based on the MIT general circulation ocean model (MITgcm)[12] which is widely used by the ocean modelling community. The MITgcm, a finite-difference ocean model, employs non-hydrostatic formulation enabling it to simulate fluid phenomena over a wide range of scales.

Computational earthquake seismology group [13] at KAUST uses codes like SORD, SeisSol [14], FD-KBO, and SPECFEM_3D_GLOBE [15]. SORD is a generalized 3D-finite-difference support-operator code for rupture dynamics. SeisSol, is an arbitrary high-order discontinuous Galerkin code for seismic wave propagation and rupture dynamics. FD-KBO is a highly optimized 3D-FD code for seismic wave propagation, and SPECFEM_3D_GLOBE is a 3D spectral-element code for global seismic wave propagation. Their work is related to seismic hazard and seismic risk mitigation, and as such falls in the category of socio-economically highly relevant research. This research will drive innovative simulation-based seismic hazard assessment.

Some of other codes used at KAUST are: S3D which is a 3D structured grid based explicit flow and species solver [16], which scales up to entire machine [17,18,]; in-house forward wave inversion (FWI) and reverse time migration (RTM) [19,20]; in-house ab initio genomic motif identification code [21]; NANOCPP [22], an in house FDTD simulator to observe phenomena like Anderson-localization and light condensation; and computational chemistry codes like VASP, LAMMPS, Gaussian, WEIN2k, Quantum Espresso [23,24,25,26]

The homogenous design of Shaheen II Cray XC40 with only Haswell makes it suitable for most, if not all, of KAUST applications. Aries network with adaptive routing provides low latency and high bandwidth solution for network bound applications. CPU/x86 based design provides an easy-to-use scalable solution for developers, production users, and Saudi industrial users who rely on commercial codes. Burst buffer, planned to be installed in September 2015, provides a faster access to disk – an attractive feature for data analytics and visualization purposes. 128 GB per node provides enough memory for reverse time migration codes and pre/post processing of simulation data. With a collaborative efforts of KSL scientists, Cray center of excellence at KAUST, and KAUST users, bulk of the applications workload will be adopted for accelerator based Shaheen II upgrade that is planned in 2017.

## IV. SHAHEEN II : SOFTWARE ECOSYSTEM

As described in Section I, Shaheen II system is tightly integrated with compute, storage and data analytics solutions, however it increases the difficulty to manage efficiently the hardware and software. Like any HPC center managing large systems, besides all the tools provided by the Cray programming environment, KSL supports hundreds of third party packages. The programming environment and the software installed by the staff are updated regularly and it is necessary to keep the installations consistent, up-to-date and providing reproducible performance and correctness of the results. This challenge is addressed in the section by describing the Shaheen II software ecosystem, that includes monitoring, software management, regression tools along with SLURM, the scheduling manager chosen by KAUST. These tools will help to ensure not only a better understanding of the usage of the system, but also a better coordination between system administration and science applications groups in order to detect earlier issues that will affect users and resolve them.

### *A. Monitoring tools*

Monitoring the usage of the system is a paramount factor to maintain and support efficiently large HPC systems. KSL has made the choice to port lightweight and tools that have been developed and maintained across several leadership-class supercomputing centers.

#### *a) Monitor software usage ALTD/XALT:*

What are the most linked, compiled and executed applications on the HPC system? How many users are using a given applications? How do we find who is using deprecated software or versions with bugs?

To address the above-mentioned enquiries, a prototype named the Automatic Library Tracking Database (ALTD) [27,28] developed and put into production on Cray XT systems in 2010 at NICS and the Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory (ORNL) is chosen for monitoring software usage. The ALTD infrastructure prototype automatically and transparently stores information about libraries linked into an application at compiling time and about every single run of this particular executable occurring on the machine here after. ALTD is invoked with the help of a wrapper of the command ld called at the linking stage of a compilation and another wrapper of launching command aprun. In these wrappers, before calling the initial command, information regarding the environment is taken from the executable. The information is saved in a database, that could be used for the subsequent posteriori statistical survey of the exact use of any application or library.

KSL has ported ALTD tool for the first time on Shaheen I. The analysis of the information collected after 6 months about our applications running on the system assisted in the design of benchmarks for Shaheen II acquisition by selecting the most used libraries and applications.

XALT [29,30] is a second generation of monitoring tool that is developed by combining the effort of ALTD and Lariat [31]. Developed at Texas Advanced Computing Center (TACC), Lariat collects data usage during different stages (submission, job launch, job ends and termination of the session). XALT improves the functionalities of ALTD and Lariat, by tracking users, codes and environments. It collects job-level and link-time level data and subsequent analytics. The data collected produce valuable information for reproducibility by providing the complete environment for successful compilation and execution for any given application. The installation of XALT on Cray XC40 is very valuable since more metrics are being recorded such as the number of threads and processes used.

*b) Monitor I/O operations: Darshan*

To analyze and understand the usage of I/O operations on leadership-class supercomputers, Darshan [32] was developed at Argonne National Laboratory as a lightweight and scalable I/O characterization tool. The tool transparently captures and records I/O access pattern information, timing and performance of the applications. Portable across several HPC platforms, Darshan is a set of libraries instrumenting MPI applications capturing both MPI-IO and POSIX file access, with limited information about HDF5 and PnetCDF access. Thanks to the data collected, one can not only monitor the system usage but also detect applications that are not using efficiently the parallel file system. Providing insight on application behavior, Darshan will help the support staff to provide a better service and help the users to achieve better scaling on the I/O system.

## B. SWTOOLS as an installation manager

To maintain infrastructure for software management of the third-party installation, SWTools [33] has been put in place. This tool developed at OLCF is designed to keep the installations consistent and up-to-date while trying to avoid problems encountered with previous software repositories. It relies on a structured hierarchy of directories where with naming, installations and documentations procedures are automatically kept compliant to a given standard. Consequently, this tool helps to ensure consistency of installations across packages and makes managing third-party software more efficient. In addition, SWTools reduces the difficulty and the apprehension for upgrading after applications, compiler or operating system upgrades.

SWTools contains several python scripts to build, link, and test any installed application. All these files are carefully versioned in a Git repository to ensure the preservations and traceability of the installation of any third party software installed by the staff.

For each application a brief description is required by SWTools. As soon as accepted by the tool, it is immediately translated in HTML format and made available to the user through the KSL website. That way, our users have automatically access to the most updated documentation rigorously inline with what is actually installed.

For each application or library installed via SWTools, a basic test example is required to validate the installation. It will be published on KSL website along with the description and will provide an excellent basis as elementary tests used to track an eventual regression analysis in the future.

## C. Jenkins as a performance non regression tracker

Along with the software management, KSL has developed an early version of an automatic testing tool for the non-regression testing of hardware components and software updates. This is needed to detect errors and/or performance degradation, allowing the center to monitor issues such as reproducibility. It is a python based tool, which compiles, launches, and checks the outputs of parallel jobs. The tool will be managed through a continuous integration of Jenkins server [34].

*a) motivation:*

On a big system like Shaheen II, so many software components are installed that failure/degradation of one of them may occur and have impact on user application performance or results. Still, to stay at the cutting edge of the research, our users often request the installation of the most updated version of an application and its dependencies. This request may occur way before any of these upgrade may have been carefully tested by CRAY or SUSE.

A regular upgrade of these components may eventually lead to some regressions than can impact the performance of these applications or of the OS itself.

We need to identify these regressions as soon as possible and to determine clearly the exact upgrade that caused it.

*b) Strategy to capture an eventual regression:*

Following actions have been put in place:

- on the system side (managed by CRAY and KAUST system administration team), every upgrade or modification is carefully traced through Bright Cluster Manager with a possibility of rollback. It concerns modifications occurring in the OS itself (that may impact filesystem performance for instance) or in the "Unix" widely used tools ( eg: sed, awk, grep, ...) that can be used in pre or post processing phases
- on the application side the KAUST Computational Scientist Team is responsible for installing and maintaining an applicative stack as independent as possible from the system low level stack.

This goal is achieved thanks to SWTools, forcing us to track precisely any installation and by the choice of rebuilding any prerequisite instead of relying on the one present in the OS distribution. We only made exception for the "Unix widely used tools" mentioned above (these have remained stable for the last 20 years or so):

- putting in place an easy mean to estimate the performance and result accuracy of each application installed on the system. For every application, the idea is to add systematically a comprehensive example with an automatized way to check its answers. At the installation of the software we carefully archive the obtained results and build a script allowing a comparison of any results compared to this reference.
- regularly testing the performance and results of every installed application, or specific benchmark demonstrating some specific feature of the system (eg filesystem bandwidth, node LINPACK performance...)

*c) Implementation:*

The automated test execution Jenkins application has been installed on a remote workstation and serves as non regression testing server as shown in Fig 4.

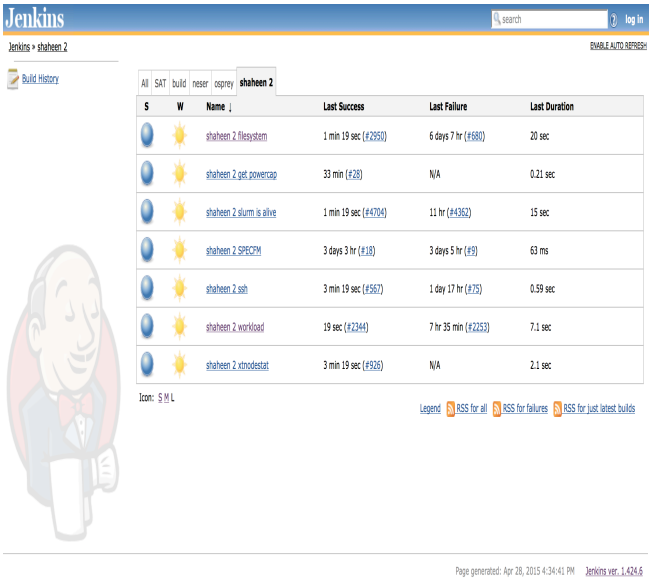


Figure 4. Jenkins interface summary.

Every application testing script or benchmark suite is made available as an elementary test unit. Regularly, through well-defined campaigns or random pick-up, a set of these tests will be initiated and run on Shaheen II via Jenkins. The results (performance or accuracy) will be carefully archived by Jenkins along with the detailed system state on the service, login or computer nodes (corresponding to a bright computing image tag) at that time.

In case of regression, the tests will be marked as failed, and Jenkins will automatically notify the Computational Scientist responsible for the application so that an eventual corrective action could be put in place, and eventually validated by the end user.

In the best interest of our users, any regression must be detected as soon as possible for:

- identifying more easily the cause of the regression (the sooner we detect it, the smaller change the system would have had)
- allowing an eventual rollback of the upgrade that could have had positive impact on other application at the same time.

The adopted strategy will be :

1. to correlate the frequency of tested application with their effective use by the users. This information is directly available from ALTD/XALT database where every execution on the system of an application in traced as well as the exact version of libraries it is built on. Browsing the database, in conjunction with the actual number of core hours read from the SLURM accounting database, one can easily gather which are the most used application and libraries and test them in priority.
2. for a given upgrade touching specific OS drivers, libraries or applications, to trigger the tests directly depending on these components. (available as well from ALTD thanks to the storing of the linking command).

## D. *SLURM as a power keeper*

Although the acceptance phase will benefit from a temporarily increasing power envelop of 2.8 MW in order to achieve expected LINPACK performance, one major constraint of the data center lies in the available power limits during normal operations. Indeed, before Shaheen I BG/P is decommissioned, a power envelop of 2.3 MW is available in production mode including computing resource, memory, network and storage. To exploit the machine to its full potential from the first day of production, efficient hardware and software power capping techniques are put in place. On a hardware side, a set of power breakers protect the whole system as well as each cabinet in case of overload of short circuit.

Additionally, the power consumption per rack and the whole power envelop are constantly monitored with a trigger sending alerts, if needed. On the software front, thanks the feature recently added to SLURM [35] by SchedMD, a global cap of the power can be maintained by conducting on-the-fly dynamic steering of the frequencies of every running jobs. The global power consumption is read after 30 seconds,. If the power consumption is above a certain threshold, the scheduler can decrease the power cap of any running job based on a policy defined at the launch of the SLURM controller. Naturally, in the case of an MPI application, the modification of power cap will be uniformly applied to each node of the job to preserve the parallel performance.

KAUST is currently testing the power capping feature which is only available with SLURM running in native mode. Native SLURM does not rely on aprun mechanism and manage its own allocation made though srun. Monitoring capabilities like RUR and optimal placement capabilities of the threads on the core still need to be ported from aprun to srun. SchedMD is currently working on this issue and KAUST will carefully validate it on the system. Some of the Shaheen II users may also be used to aprun command. As we do not want them to waste their time in finding the equivalent parameters needed by srun, an eventuality would be to code a parsing wrapper from aprun to srun. The principle is to translate every option available for aprun in the syntax of srun. By September 2014, Burst-buffer reservation and scheduling should also be added to SLURM in order for KAUST to be able to manage among users this increased I/O capabilities. At that time, a typical job will then request a set of nodes, an eventual access to a given capacity of the burst buffer during a certain amount of time. The job will then be scheduled depending on the simultaneous availability of all these resource.

## V. CONCLUSIONS

With the acquisition of the new Cray XC40, Shaheen II, KAUST is once again the owner of a world-class supercomputer. With the strategic investment in supercomputing, KAUST will significantly augment its world-class academic, research facilities and capabilities to advance scientific discoveries. Shaheen II will enable and grow collaboration with several in-Kingdom universities, industrial partners and other international leadership class supercomputers centers.

## REFERENCES


[1] King Abdullah University of Science and Technology KAUST: http://kaust.edu.sa

[2] TOP500 Supercomputer Site http://www.top500.org

[3] http://www.kaust.edu.sa/faculty/stenchikov.html

[4] http://www.kaust.edu.sa/faculty/samtaney.html

[5] http://ctflab.mae.cornell.edu/nga.html

[6] http://www.kaust.edu.sa/faculty/bisetti.html

[7] http://web.kaust.edu.sa/faculty/hakanbagci/people.html

[8] http://cloud.kaust.edu.sa/

[9] Z. Khayyat, K. Awara, A. Alonazi, H. Jamjoom, D. Williams, and Pa. Kalnis. Mizan: a system for dynamic load balancing in large-scale graph processing. In Proceedings of the 8th ACM European Conference on Computer Systems (EuroSys '13). ACM, New York, NY, USA, 169-182

[10] G. Malewicz , M. H. Austern , A. J.C Bik , J. C. Dehnert , I. Horn , N. Leiser , G. Czajkowski, Pregel: a system for large-scale graph processing, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, June 06-10, 2010, Indianapolis, Indiana, USA

[11] http://www.kaust.edu.sa/faculty/hoteit.html

[12] J. Marotzke, R. Giering, K. Q. Zhang, D. Stammer, C. Hill, and T. Lee (1999) Construction of the adjoint MIT ocean general circulation model and application to Atlantic heat transport variability. J. Geophysical Res., 104(C12), pp 29,529-29,547

[13] http://ces.kaust.edu.sa/

[14] J. De la Puente, M. Käser, and J. M. Cela, SeisSol Optimization, Scaling and Synchronization for Local Time Stepping, in Science and Supercomputing in Europe, pp. 300-3022, CINECA, Italy

[15] J. Tromp, D. Komatitsch, and Q. Liu. Spectral-element and adjoint methods in seismology. Communications in Computational Physics , 3(1):1–32, 2008

[16] J.H. Chen et. al., Terascale direct numerical simulations of turbulent combustion using S3D Comput. Sci. Disc. , 2009

[17] http://ccrc.kaust.edu.sa/Pages/Hong-Im.aspx

[18] http://www.kaust.edu.sa/faculty/chung.html

[19] http://www.kaust.edu.sa/faculty/schuster.html

[20] http://www.kaust.edu.sa/faculty/alkhalifa.html

[21] http://www.kaust.edu.sa/faculty/bajic.html

[22] www.primalight.org

[23] http://www.kaust.edu.sa/faculty/bredas.html

[24] http://www.kaust.edu.sa/faculty/schwingenschlogl.html

[25] http://www.kaust.edu.sa/faculty/cavallo.html

[26] http://www.kaust.edu.sa/faculty/manchon.html

[27] M. Fahey, N. Jones, B. Hadri, The Automatic Library Tracking Database, Conference: Cray User Group 2010, Edinburgh, UK.

[28] B. Hadri and M. Fahey, Mining Software Usage with the Automatic Library Tracking Database (ALTD), Procedia Computer Science, Volume 18, 2013, Pages 1834-1843, ISSN 1877-0509

[29] http://sourceforge.net/projects/xalt/

[30] K. Agrawal, M. R. Fahey, R. McLay, and D. James. 2014. User environment tracking and problem detection with XALT. In *Proceedings of the First International Workshop on HPC User Support Tools* (HUST '14). IEEE Press, Piscataway, NJ, USA, 32-40.

[31] https://github.com/TACC/lariat

[32] Darshan: http://www.mcs.anl.gov/research/projects/darshan/

[33] N. Jones and M. R. Fahey, "Design, Implementation, and Experiences of Third-Party Software Administration at the ORNL NCCS," Proceedings of the 50th Cray User Group (CUG08), Helsinki, Finland, May 2008.

[34] http://jenkins-ci.org/

[35] http://slurm.schedmd.com/power_mgmt.html