OAK RIDGE National Laboratory

CUG 2019 Birds of a Feather

Large-scale System Acceptance Testing: Procedures, Tests, and Automation

Chairs:
Verónica G. Vergara Larrea
Reuben D. Budiardja

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

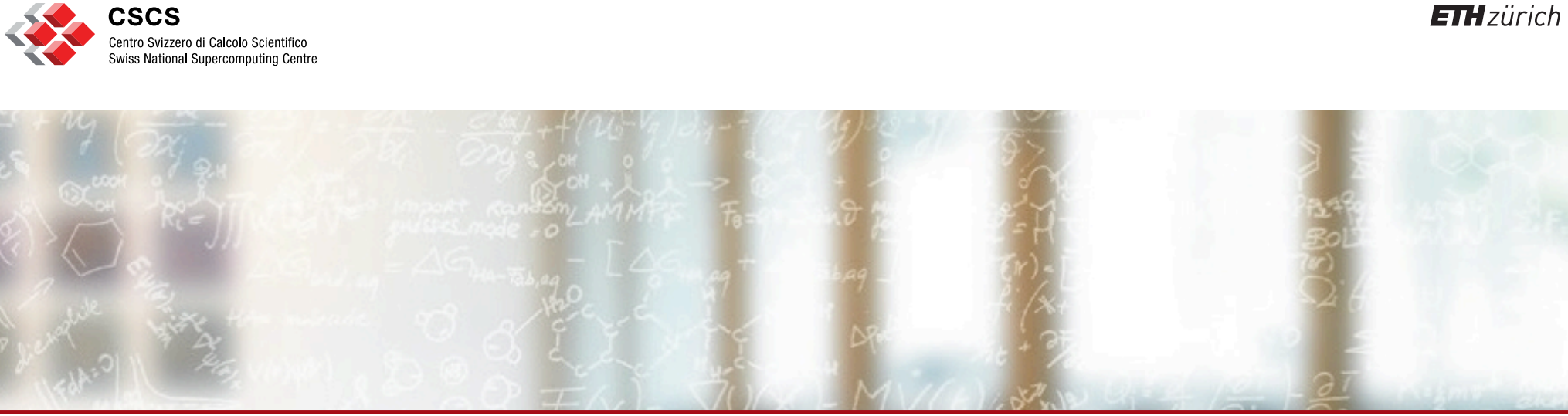
U.S. DEPARTMENT OF ENERGY

BOF Goals

- Share acceptance test procedures
- Learn about tools used successfully for testing
- Compile resources, tools, and possibly tests

Presenters

- Nick Cardo (CSCS)
- Tina DeClerk (NERSC)
- Bilel Hadri (KAUST)
- Celso Mendes (NCSA)
- Verónica G. Vergara Larrea / Reuben Budiardja (OLCF)



Large-Scale System Acceptance Testing: Procedures, Tests, and Automation

Nicholas P. Cardo, CSCS

May 7, 2019

Introducing Piz Daint

- 5704 XC50 Compute Nodes
  - Intel® Xeon® E5-2690 v3 @ 2.60GHz
    - 12 cores, 64GB RAM
  - NVIDIA® Tesla® P100 16GB
- 1813 XC40 Compute Nodes
  - Two Intel® Xeon® E5-2695 v4 @ 2.10GHz
    - 2 x 18 cores, 64/128 GB RAM
- 8.8 PB Sonexion 3000 Lustre Scratch
- 12x Optical Inter-group cables (AOC)

Servicing all scientific disciplines with a diverse workload…

Acceptance Testing

- Structured in Phases *(2 weeks more or less)*
  - Hardware Validation: *all hardware delivered and working properly*
  - Functional Testing: *system operates as a whole from system level to user level*
  - Performance Testing: *expected performance levels have been achieved (and variability)*
  - Availability Testing: *system demonstrates reliability and readiness for the users*
- Test Selection
  - Key scientific applications representative of select disciplines
    - Material Science, Chemistry, Physics, Climate, Biology, Economics, Machine Learning…
  - Key application kernels
    - N-body, dgemm, magma, stencils, MPI, OpenMP, Affinity, Spec Suite, HDF5, NetCDF, streams, …
  - Representative benchmarks
    - IOR: *HPC I/O Benchmark (https://github.com/hpc/ior)*
    - HPCG: *High Performance Conjugate Gradients (http://www.hpcg-benchmark.org)*
    - HPL: *High Performance Linpack (http://www.netlib.org/benchmark/hpl/)*
    - Network: *all-to-all, p2p, latency,…*
    - CUDA Memory: *latency, performance, variability)*
    - huge pages: *memory access latencies*
    - …
  - Others
    - Visualization *graphical capabilities of the GPU accelerators*
    - User Environment: *quality of service*
    - Connectivity: *access methods to the system/network connectivity performance*
    - Workload Management: *large volumes of job, flexible scheduling, reservations, specific nodes, GPU setup*

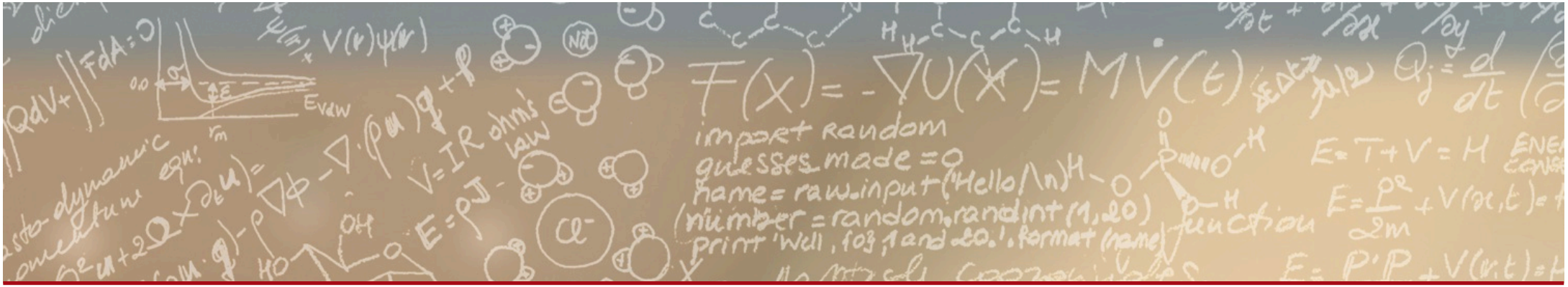
Sustained & High System Stress

Testing Execution and Tools

- Test Execution
  - Specific testing designated for specific days based on identified phases
  - Testing tracked and gated by a coordinator
  - Specific people charged with specific activities
  - Upon completion of each phase the results are reviewed and discussed
  - Upon completion of all phases the results are reviewed and discussed

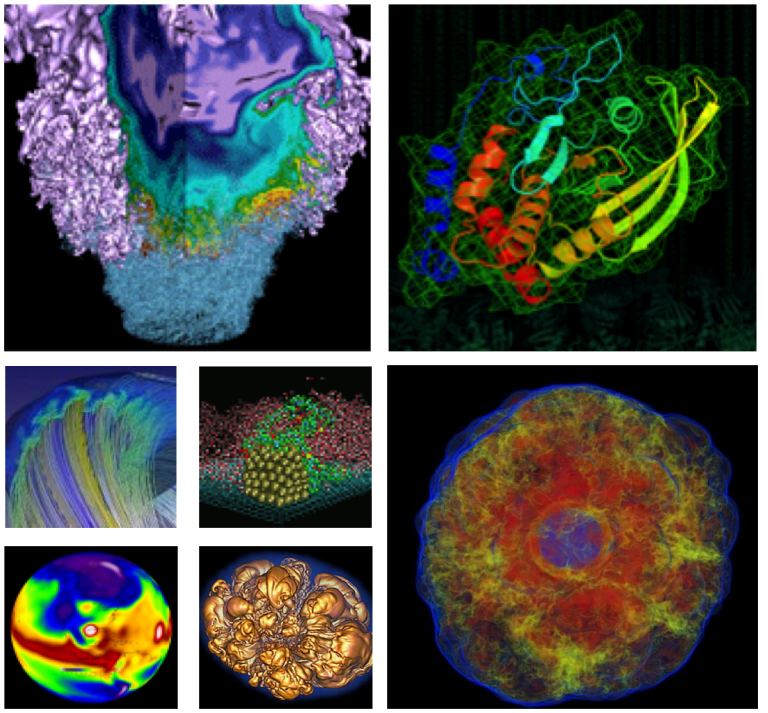
- Testing Tools

  - GitHub: *revision control non-sensitive repositories*
  - GitLab: *revision control internal repositories*
  - Jenkins: *continuous integration (https://jenkins.io)*
  - ReFrame: *regression test framework (https://reframe-hpc.readthedocs.io/en/latest/)*
  - Scripts: *custom written as needed*
  - cksys: *system level test harness*
    - *Cray supplied health checks are insufficient to check accelerator health*



Thank you for your attention.

Acceptance Testing



Tina Declerck

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Components of Acceptance Testing

- Statement of Work
- Acceptance Tests
  - Functionality Tests
  - Reliability Tests
  - System Tests
  - Performance Tests
  - Availability Tests
- Testing should be performed on the system when in a production configuration
- Tests are defined and agreed upon with the vendor in advance
- Vendor performs the tests and the site observes

Functionality, Reliability, and System Tests

- Ensure the system works as expected
- Functionality
  - Basic system works as expected and all components are available
    - User based - Compilers, modules, basic commands, file system, etc.
    - System based – configuration options, image creation, boot, shutdown, etc.
- Reliability
  - Ensure components with the ability to failover work – pull power on secondary server
  - Ensure components with the ability to operate in degraded mode work as expected- fans, power supplies, etc.
- System Tests
  - Warm and cold boot within a specified time limit

Performance Tests

- As defined in the SOW. Cover several aspects of system performance
  - Application Benchmarks
  - Data science benchmark
  - Memory bandwidth
  - MPI performance
  - Bi-section bandwidth
  - Network performance
  - File system
    - Bandwidth performance
    - Metadata performance

Availability

- System needs to run for 30 days in a 45 day window
  - A specified amount of time is allowed for maintenance
  - Users are on the system
  - Node or system failure rates must be within the limits specified by the SOW
  - The system must at some point run for 7 continuous days
  - Any failure to meet specification will stop the clock or require a re-start of the test

BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Blue Waters Acceptance Testing: Overview

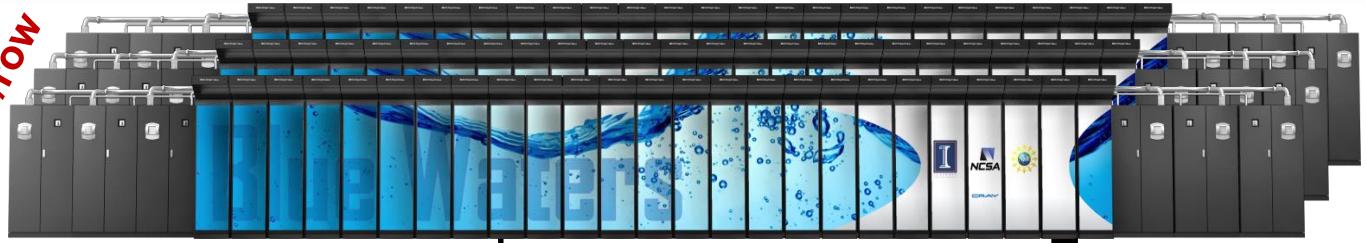
Celso Mendes, Brett Bode, William Kramer

NCSA

GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

1 – Introduction - Blue Waters: Sustained-Petascale System

288 cabinets:
12 rows x 24/row



Aggregate Memory – 1.6 PB

10/40/100 Gb Ethernet Switch

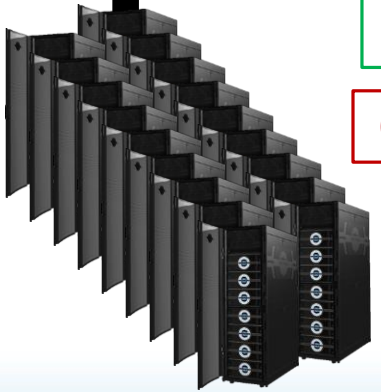
External Servers

IB Switch

>1 TB/sec

120+ Gb/sec

100 GB/sec



Spectra Logic: 300 usable PB

Sonexion: 26 usable PB

Peak: 13.3 PF
XE: 7.1 PF
XK: 6.2 PF

3D-Torus:
24 × 24 × 24

Deployment: 2012

Operation: 04/2013

27+ B. core-hrs

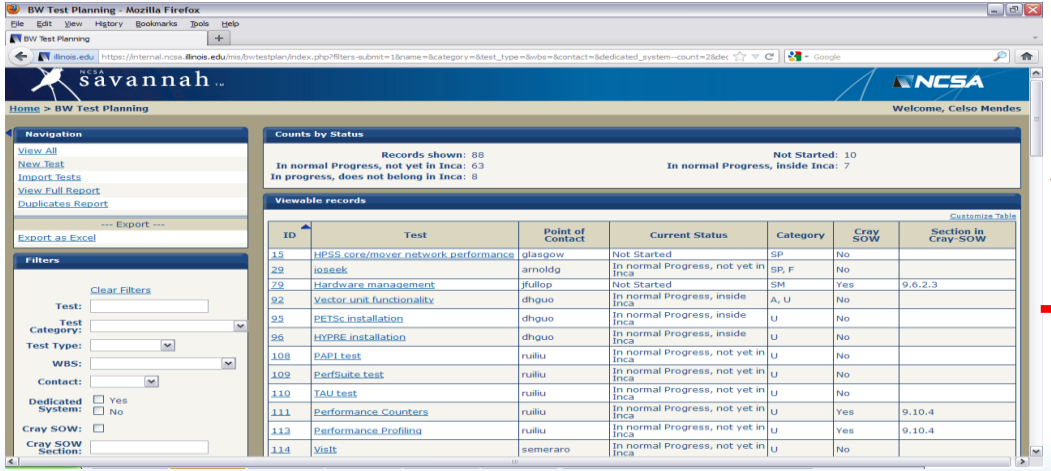
2 - Acceptance Testing

a) Structure of Test Plan: *what does acceptance look like at your center?*

- Acceptance Testing: Test Design + Execution + Verification
- Detailed test-design phase: 2011 – Test Matrix (ref: CUG-2012)

actions

filters



stats

TESTS (300+)

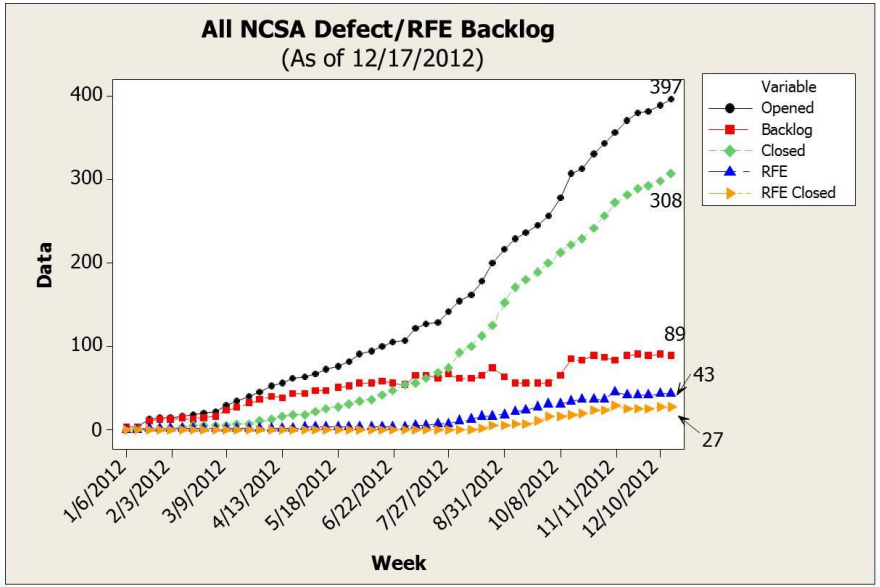
2 - Acceptance Testing

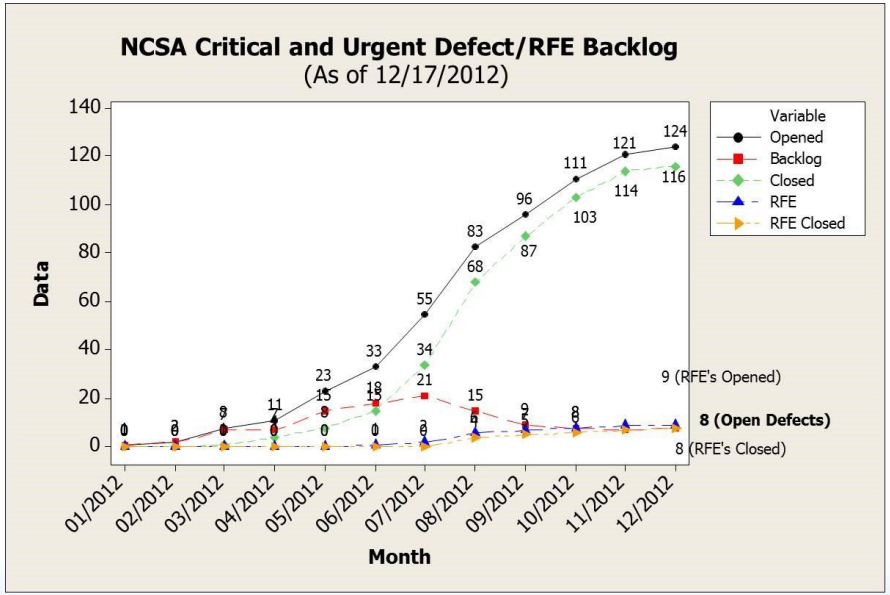
a) Structure of Test Plan: *How long does it take? How large is your team?*

- Timeframe of acceptance:
  - Jan-Jun/2012: Design and preparation of tests on TDS and Early-Sc.
  - Jul-Sep/2012: “Testing” of all tests – debugging, refining, etc
    - Many tests applied to Cray software on TDS
  - Oct-Nov/2012: Bulk of acceptance testing, availability evaluation
  - Dec/2012: Reporting, acceptance decisions
- Personnel involved in testing:
  - Entire Blue Waters team: ~40 people
  - Varied levels of participation and responsibility

2 - Acceptance Testing (cont.)

- Defects found during testing period, filed to CrayPort





2 - Acceptance Testing (cont.)

b) Test Selection: *How do you determine which tests to use? What about job sizes to use? Do you use benchmarks, full applications, or a mix?*

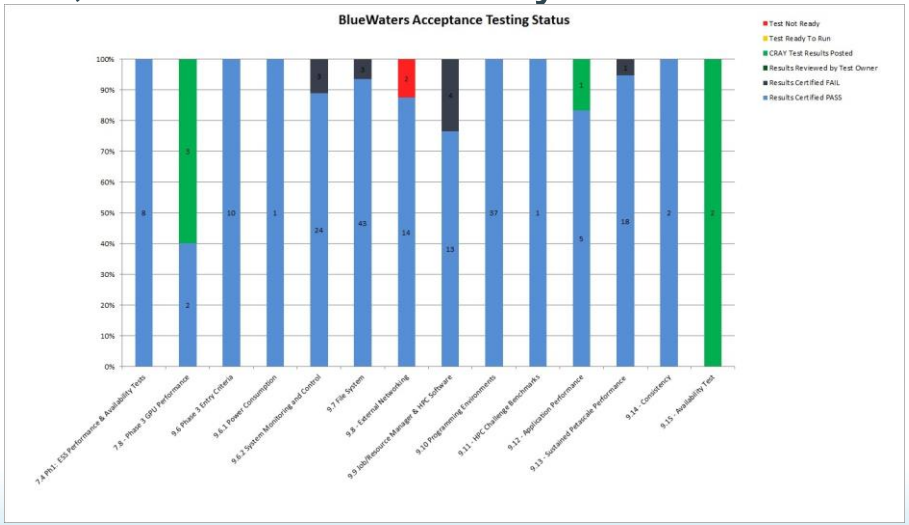
- Two classes of tests:
  - Tests directly derived from the NCSA/Cray contract (SOW): 219 tests
  - Tests specified by NCSA based on other system specs
- Job sizes:
  - Many full-system tests, to demonstrate sustained-petascale perform.
  - Some smaller tests to verify functionality
- Types of tests - Mix:
  - Full applications: Petascale apps, SPP apps – various areas of science
  - Benchmarks (HPCC, IOR, etc)

3 - Testing Tools

a) How do you execute your acceptance test? *By hand vs. home-grown tool vs. open source vs. comm*

- Special job queue created for tests, controlled manually 24/7
  - Test execution: *test owner*
  - Job scheduler: Joe Glenski!
- Tracking of progress: daily
  - Checked jointly by NCSA/Cray
  - All results stored at internal Wiki
    - Results classified into 5 levels

e.g. status on Oct.17, 2012:

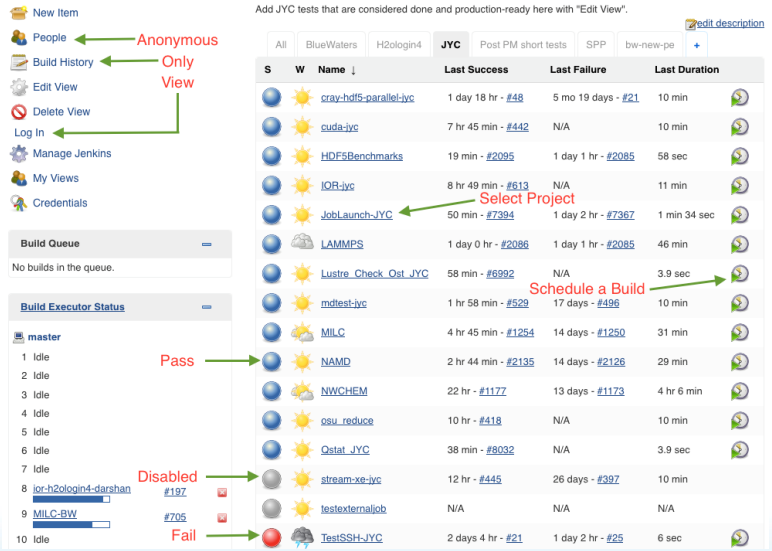


3 - Testing Tools (cont.)

b) Have you considered other tools? *e.g.: Gitlab CI, Jenkins, CTest, etc.*

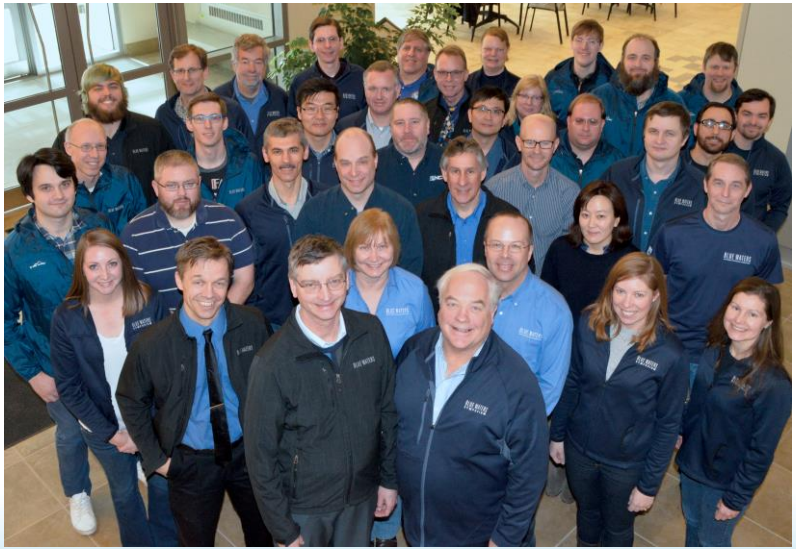
- NCSA is using Jenkins for regression testing on Blue Waters
- Tests run periodically or on demand
- Historical results remain available
- Help from R.Budiardja (ORNL)
  - Described in paper @ CUG'2017

GUI of BW-Jenkins:



Acknowledgments

- Funding: NSF OCI-0725070/ACI-1238993, State of Illinois
- Personnel: **NCSA** Blue Waters team, Cray site team

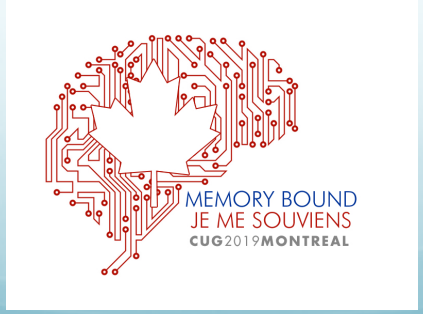


KAUST Supercomputing Lab: Acceptance and Regression

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

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

SHAHEEN
SUPERCOMPUTING LABORATORY



Shaheen 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
	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 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 of some system components were checked only before releasing the system back to the users as soon as possible.

- 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

- 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

Objective and Design

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

- **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, SRT last about one hour

Benefits

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

In the last 3 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**

OAK RIDGE National Laboratory

OLCF Acceptance Testing

Verónica G. Vergara Larrea

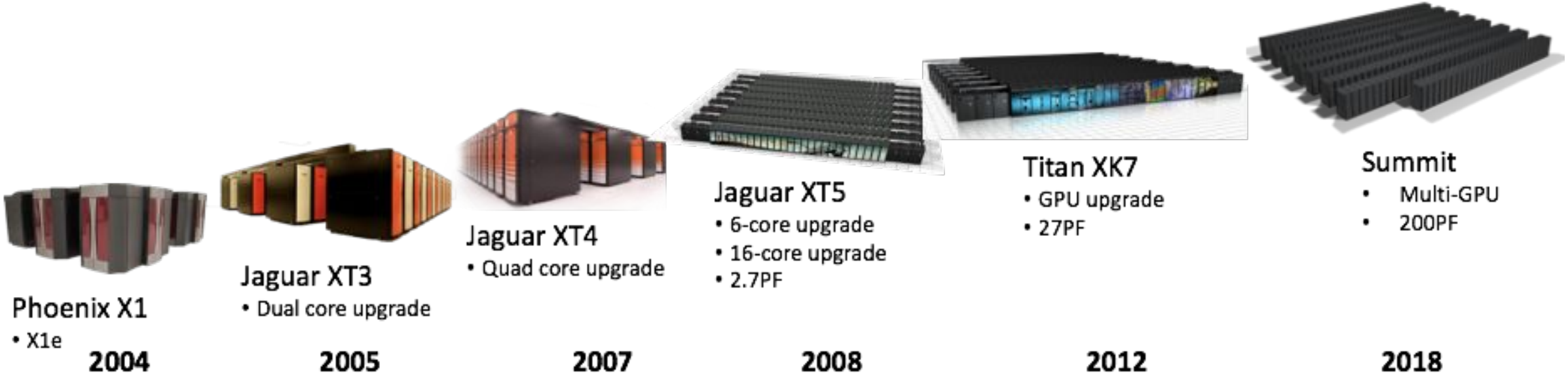
Reuben Budiardja

Outline

- What is the OLCF?
- Summit
- Acceptance Testing
- Post-acceptance Testing

What is the Oak Ridge Leadership Computing Facility?

- Deploy and operate computational and data resources required to tackle global challenges.
- Offer leadership-class computing resources to researchers who have many of the largest computing problems in science.
- Partnering has been essential to delivering science in a rapidly changing computational environment.



Summit

Compute System

- 256 compute racks
- 4,608 compute nodes
- Mellanox EDR IB fabric
- 200 PFLOPS – FP64
- ~11 MW, 70F cooling water
- 10.2 PB Total Memory

summit

Compute Rack

- 18 Compute Servers
- Warm water (70°F direct-cooled components)
- RDHX for air-cooled components

39.7 TB Memory/rack
55 KW max power/rack

Compute Node

- 2 x POWER9
- 6 x NVIDIA GV100
- NVMe-compatible PCIe 1.6 TB SSD

25 GB/s EDR IB- (2 ports)
512 GB DRAM- (DDR4)
96 GB HBM- (3D Stacked)
Coherent Shared Memory

IBM POWER9
- 22 Cores
- 4 Threads/core
- NVLink

NVIDIA GV100
- 7 TF
- 16 GB @ 0.9 TB/s
- NVLink

GPFS File System

250 PB storage
2.5 TB/s read, 2.5 TB/s write
(**2.5 TB/s sequential and 2.2 TB/s random I/O)

Acceptance Testing

HW
- Hardware Acceptance Test: Complete hardware diagnostics.

FT
- Functionality Test: Demonstrate that basic hardware and software functionality meet essential requirements.

PT
- Performance Test: Demonstrate that the system hardware and software meet performance and scalability requirements of the suite of applications defined in the Agreement.

ST
- Stability Test: Demonstrate stability across a mix of simulated code development activity and production simulations.

Acceptance Tests

	Languages				Compilers				Features / Prog. Models									Libraries												Motifs							Other					I/O				
Full Applications	FORTRAN	FORTRAN 2003	C	C++	XL	LLVM	GNU	PGI	GPU accelerated	Uses Multiple GPUs	Pthreads	C++11 threads	OpenMP 3.1	OpenMP 4	OpenACC	CUDA	Kokkos	MPI	BLAS	LAPACK	ESSL	Magma	cuBLAS	Scalapack	cuFFT	FFTW	PETSc	Trilinos	Hypre	AmgX	Structured grids	Unstructured grids	Dense linear algebra	Sparse linear algebra	Particles	Monte Carlo	FFT	Burst Buffer	SCR	GPUDirect	UVM	GlobalArrays	HDF5	HDF5 parallel	NetCDF	NetCDF parallel	ADIOS
CHROMA			X	X	X	X	X		X				X			X		X	X			X	X							X			X		X				X	X					
NAMD				X					X				X					X							X	X				X				X		X								
LAMMPS				X			X		X	X			X			X	X	X						X		X								X		X								
LSMS	X			X			X		X	X			X			X		X	X				X										X			X						X				
ACME	X	X			X		X	X	X				X		X			X												X	X								X				X	X	
GTC	X							X	X	X			X		X			X								X		X	X	X	X		X	X											X
NWCHEM	X		X	X	X		X	X	X		X		X			X		X	X	X	X		X	X								X										X			
Profugus				X		X	X		X							X											X								X										
Additional benchmarks, kernels, and mini-apps												X		X																							X	X				X				

Acceptance Tests (cont'd)

- Benchmarks
  - contractual, feature specific
- Real-world Applications
  - past workloads and expected workloads
- Workflows
- Tools!
  - Profilers (nvprof)
    - Test in single- and multi-host modes on applications
    - Test traces, profiles, analysis metrics for CUDA programs
  - Debuggers (ARM DDT)
    - Ensure it can run at-scale (20% full system) reliable and start within 5 minutes
    - Used in *offline* (non-interactive) mode
    - Breakpoints set on application, output captured and validated with script

OLCF Test Harness

Harness Input

```
test = app2 t1_n1_ppn4
```

Harness driver `runtests.py`

```
##########################################
   Batch ID     Build     Submit     Check
##########################################
   224932       0         0          0
   224961       0         0          5
   269003       0         0          -1
   285580       0         0          1
```

SVN or Git repository

app1

t1_n1_ppn2

app2

t1_n1_ppn4

...

Working directory

app2

Source/

`src/`
`build.sh`

t1_n1_ppn4/

`job_script.sh`
`check.sh`

Post-acceptance Testing

- A shortened version of the acceptance test is prepared to validate new versions of the HPC SW stack.
- Expand tests to include applications from the new allocation cycle.
- Multi-stage testing:
  - Starts at the smaller scale on the test & development system
  - Schedule a 8-12 hour testshot on Summit
- Delicate balance: downtime vs. risk of introducing issues into production
- Automate regression testing: