BLUE WATERS

SUSTAINED PETASCALE COMPUTING

May 11, 2017

Application-Level Regression Testing Framework using Jenkins

Timothy A. Bouvet, NCSA, University of Illinois
Reuben D. Budiardja, ORNL, Oak Ridge, Tennessee
Galen W. Arnold, NCSA, University of Illinois
Gregory H. Bauer, NCSA, University of Illinois

NCSA

GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

CRAY

Talk Outline

- Introduction and Motivation
- Implementation & Deployment
  - Basic Configuration
  - Connecting to multiple HPC systems
  - Security consideration, authentication, authorization
- Application Test Structure
- Actual Use Cases

Introduction & Motivation

- The complexities of monitoring an HPC system:
  - Layers and versions of software stack
  - System-level configurations
  - I/O components (memory, filesystem)
  - Diversity in user need and usage
  - Variable network traffic

Aggregate impact?

- Performance regression may occur over time
  - requires historical data to easily detect
- **Application-level regression testing**

Goals

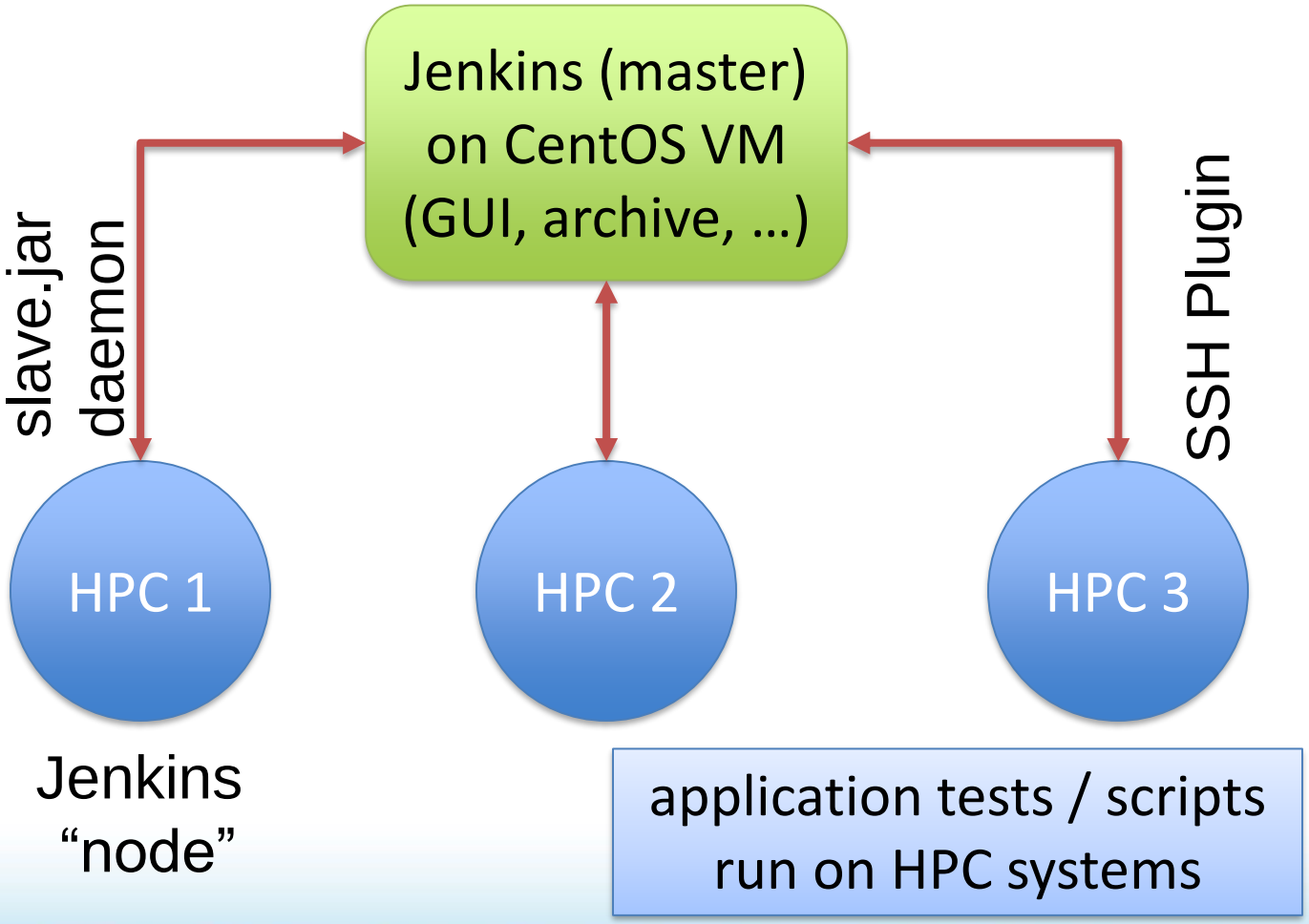
- Monitor usability and performance
- User-level experience
  - no privilege access, minimum ‘specialized’ tests
- Early detection of regressions
  - Tests are run during production period
- Automated and consistent test solution
- Store historical data
- **Solution: Jenkins Automation Server**

What is Jenkins?

- An “open source automation server”
- Automate user-level tasks
  - Execute arbitrary commands
  - Build software & submit jobs
  - Archive test results
  - Notify (email, text, …) on error
- “Never sleeps or takes vacation”
- Chosen for
  - versatility
  - multitudes of plugins (plot, ssh, …)
  - large community support

Implementation & Deployment

Basic Configuration



Accessing (Remote) HPC Test Systems

Jenkins "Node"

- Requires running "slave.jar" daemon
- Daemon connects back to master on specific port
- Remote execution appears "local"
  - no need to copy files / data
  - can access lustre, /home as if they are local resources

NICS

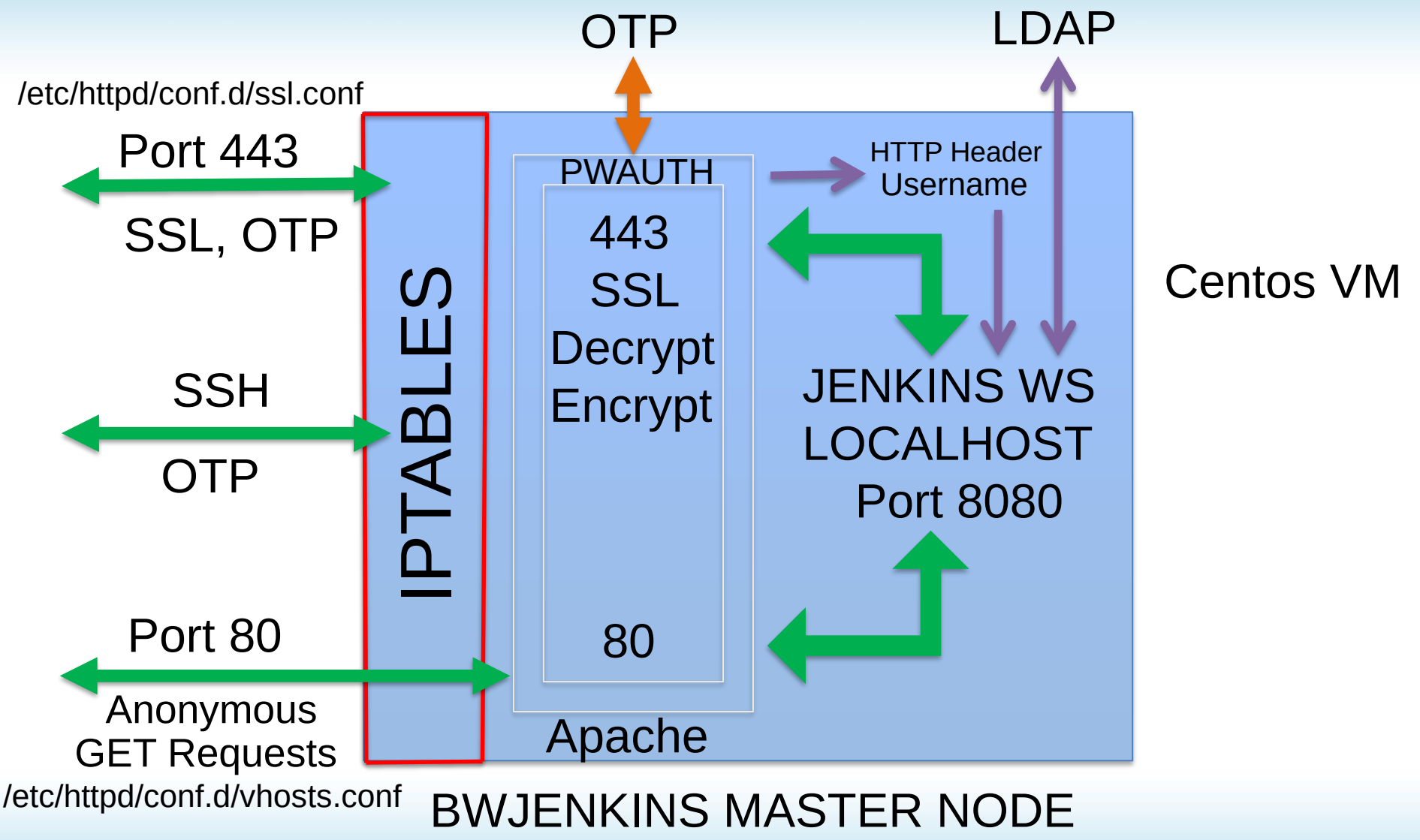
With SSH Plugin

- Need to set up passwordless login with SSH keys
- Closer resemblance to user experience
- Have to manually manage data movement (file copying, getting results back) to

NCSA

Security Considerations

- Jenkins allows execution of arbitrary command
  - use Apache HTTPD server in front of Jenkins', pass requests via mod_proxy
  - allow authenticated access only from institutional IPs via firewall rules
  - use RSA OTP for authentication via pwauth + custom session management (details in paper)
- Use LDAP for authorization with Jenkins "matrix-based" security


OTP
LDAP
/etc/httpd/conf.d/ssl.conf
Port 443
SSL, OTP
SSH
OTP
Port 80
Anonymous
GET Requests
/etc/httpd/conf.d/vhosts.conf
IPTABLES
PWAUTH
443
SSL
Decrypt
Encrypt
80
Apache
HTTP Header
Username
JENKINS WS
LOCALHOST
Port 8080
Centos VM
BWJENKINS MASTER NODE

Jenkins Global Security

Authorization

- ○ Anyone can do anything
- ○ Legacy mode
- ○ Logged-in users can do anything
- ◉ Matrix-based security

User/group	Overall					Credentials					Agent							Job								Run		View				SCM
	Administer	ConfigureUpdateCenter	Read	RunScripts	UploadPlugins	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Read	Workspace	Delete	Update	Configure	Create	Delete	Read	Tag
Sub_UTK	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
arnoldg	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑
bw_admin	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
bw_seas	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☐	☑	☑	☑	☐	☑	☑	☑	☐	☑	☑
rbudiard	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Anonymous	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☐	☐	☐	☑	☐

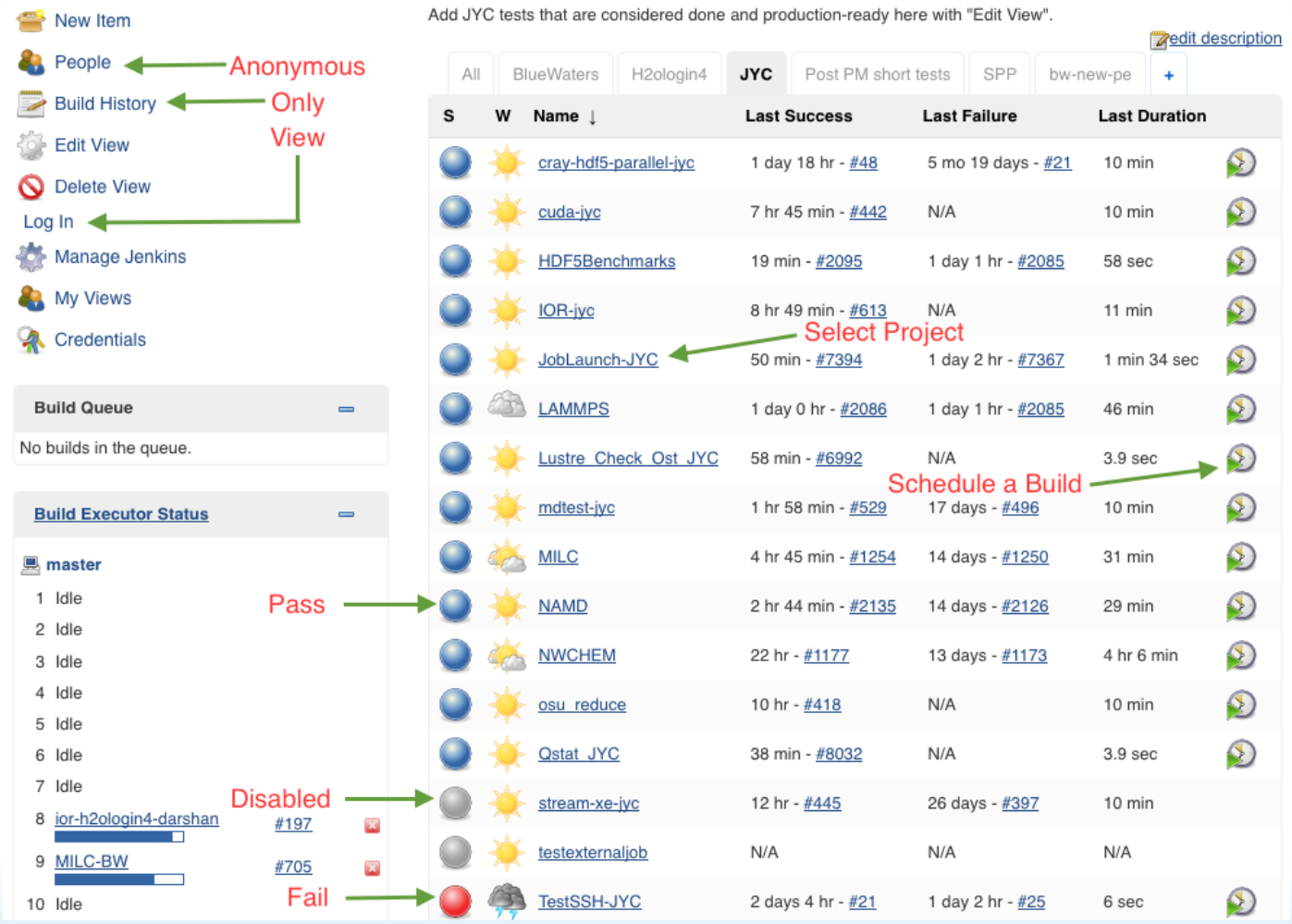
Application Test Structure

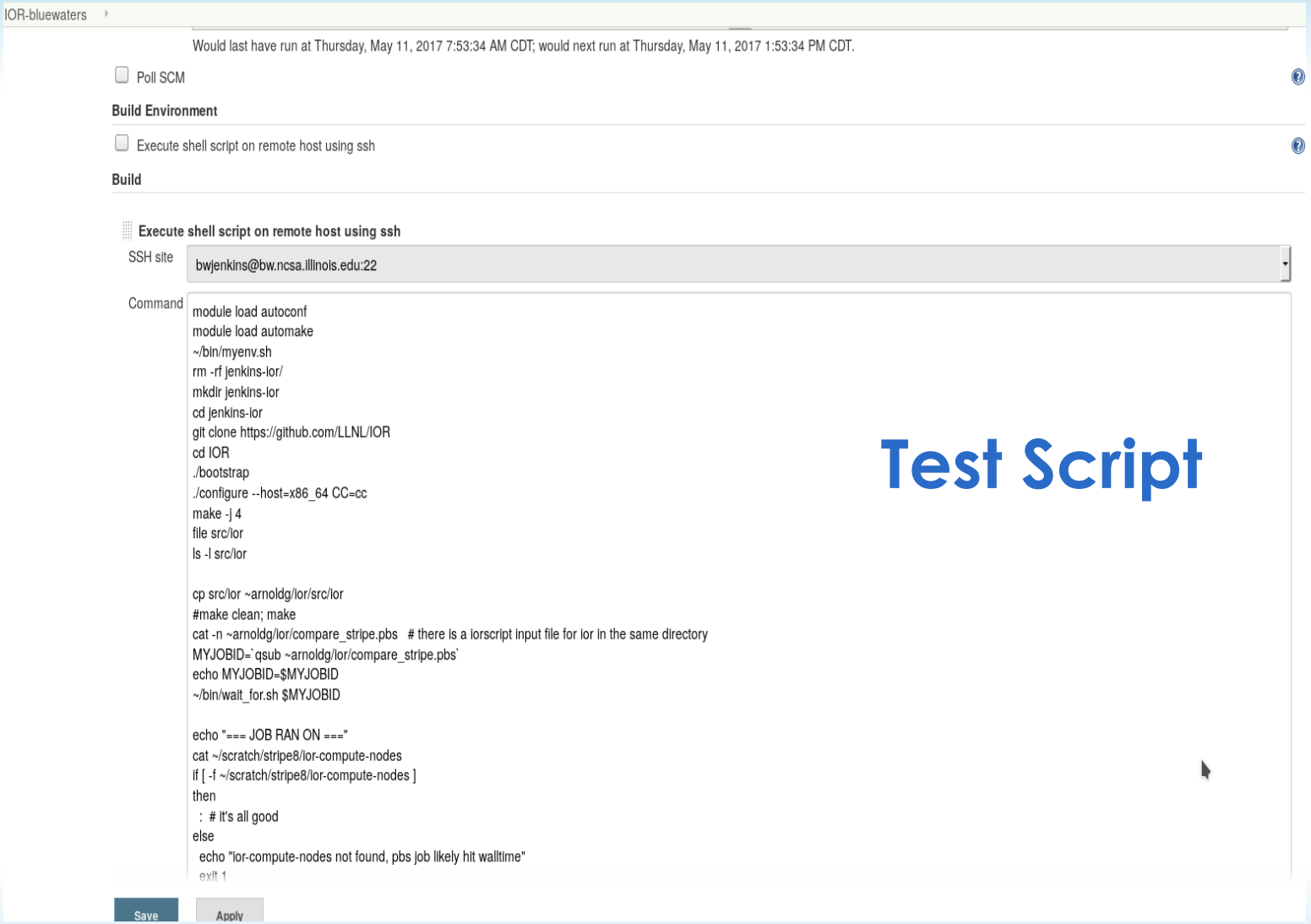
An Application Test: Jenkins Project

- Name and description - what is tested, frequency, resources
- Limit number of “builds” kept - conserve server storage
- Source code management - subversion or git
- Build trigger - manual or automated schedule
- Build commands – shell commands to execute test
- Post-build Actions – generate plots, notifications on failure

Project can be easily copied from an existing one

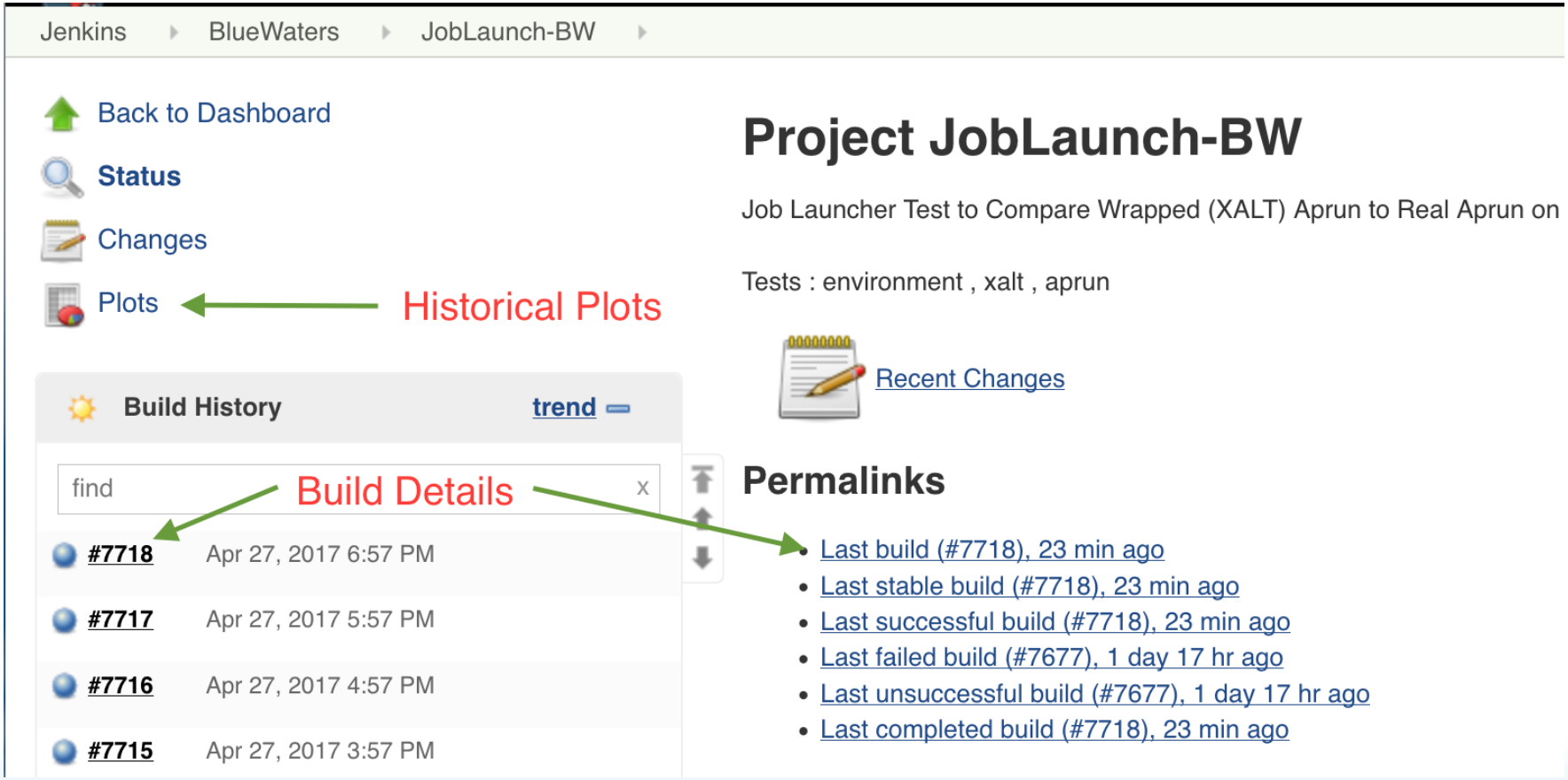
Admin Gui Home Page



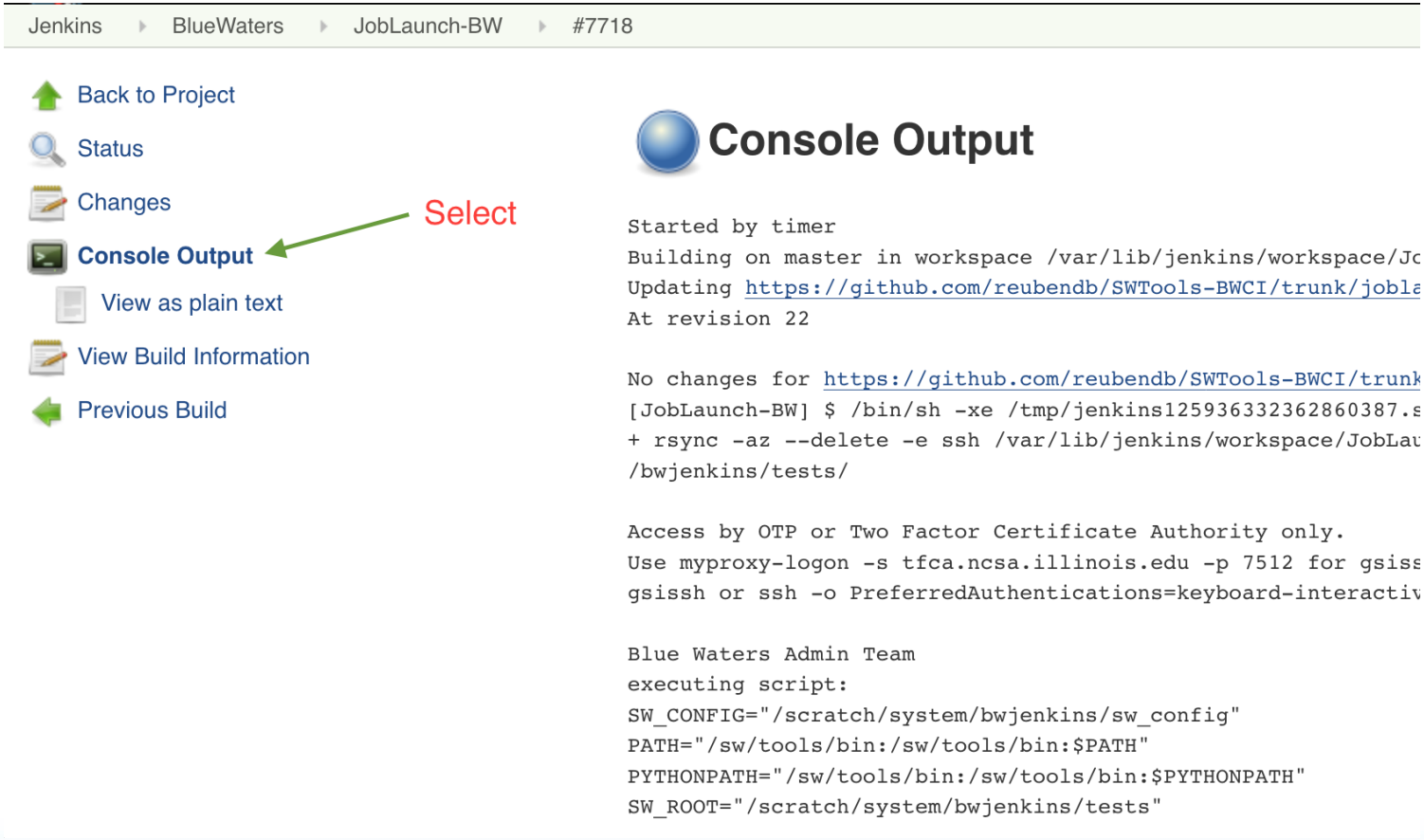

IOR-bluewaters
Would last have run at Thursday, May 11, 2017 7:53:34 AM CDT; would next run at Thursday, May 11, 2017 1:53:34 PM CDT.
Poll SCM
Build Environment
Execute shell script on remote host using ssh
Build
Execute shell script on remote host using ssh
SSH site
bwjenkins@bw.ncsa.illinois.edu:22
Command
module load autoconf
module load automake
~/bin/myenv.sh
rm -rf jenkins-ior/
mkdir jenkins-ior
cd jenkins-ior
git clone https://github.com/LLNL/IOR
cd IOR
./bootstrap
./configure --host=x86_64 CC=cc
make -j 4
file src/ior
ls -l src/ior
cp src/ior ~arnoldg/ior/src/ior
#make clean; make
cat -n ~arnoldg/ior/compare_stripe.pbs # there is a iorscript input file for ior in the same directory
MYJOBID=`qsub ~arnoldg/ior/compare_stripe.pbs`
echo MYJOBID=$MYJOBID
~/bin/wait_for.sh $MYJOBID
echo "=== JOB RAN ON ==="
cat ~/scratch/stripe8/ior-compute-nodes
if [-f ~/scratch/stripe8/ior-compute-nodes]
then
: # it's all good
else
echo "ior-compute-nodes not found, pbs job likely hit walltime"
exit 1
Save
Apply

Test Script

Project View



Example Console Log



Use Cases

Project Failure Email Notification

See <http://bwjenkins.ncsa.illinois.edu/job/Lustre_Check_OST_BW/7218/display/redirect>

Started by timer

Building on master in workspace <http://bwjenkins.ncsa.illinois.edu/job/Lustre_Check_OST_BW/ws/>

executing script:

SW_WORKDIR="/scratch/system/bwjenkins/sw_workdir"

JOB_NAME="Lustre_Check_OST_BW"

snx11003-OST010c-osc-ffff885fc0cd3c00: check error: Resource temporarily unavailable

[SSH] exit-status: 255

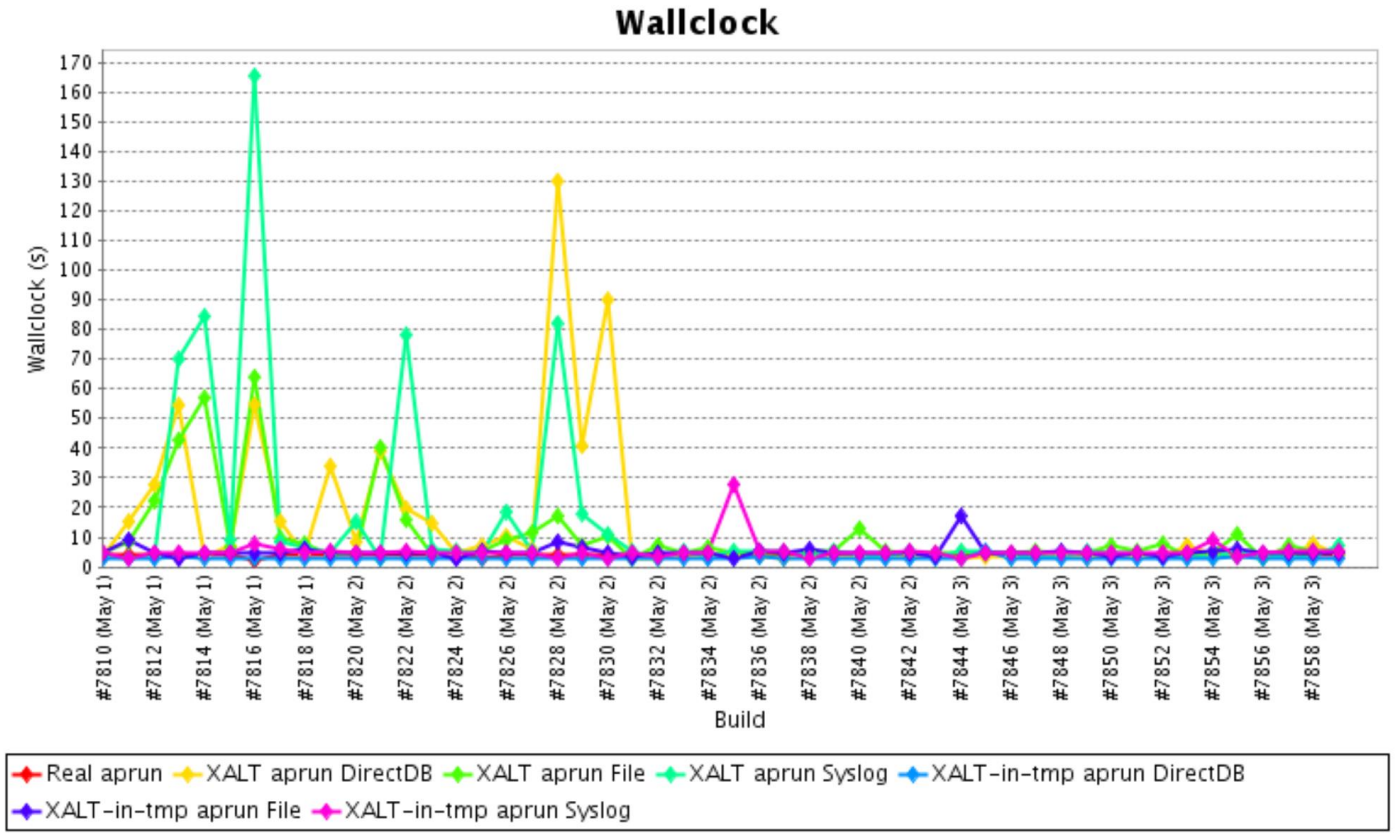
Build step 'Execute shell script on remote host using ssh' marked build as failure

Recording plot data

Saving plot series data from:
<http://bwjenkins.ncsa.illinois.edu/job/Lustre_Check_OST_BW/ws/bw_active>

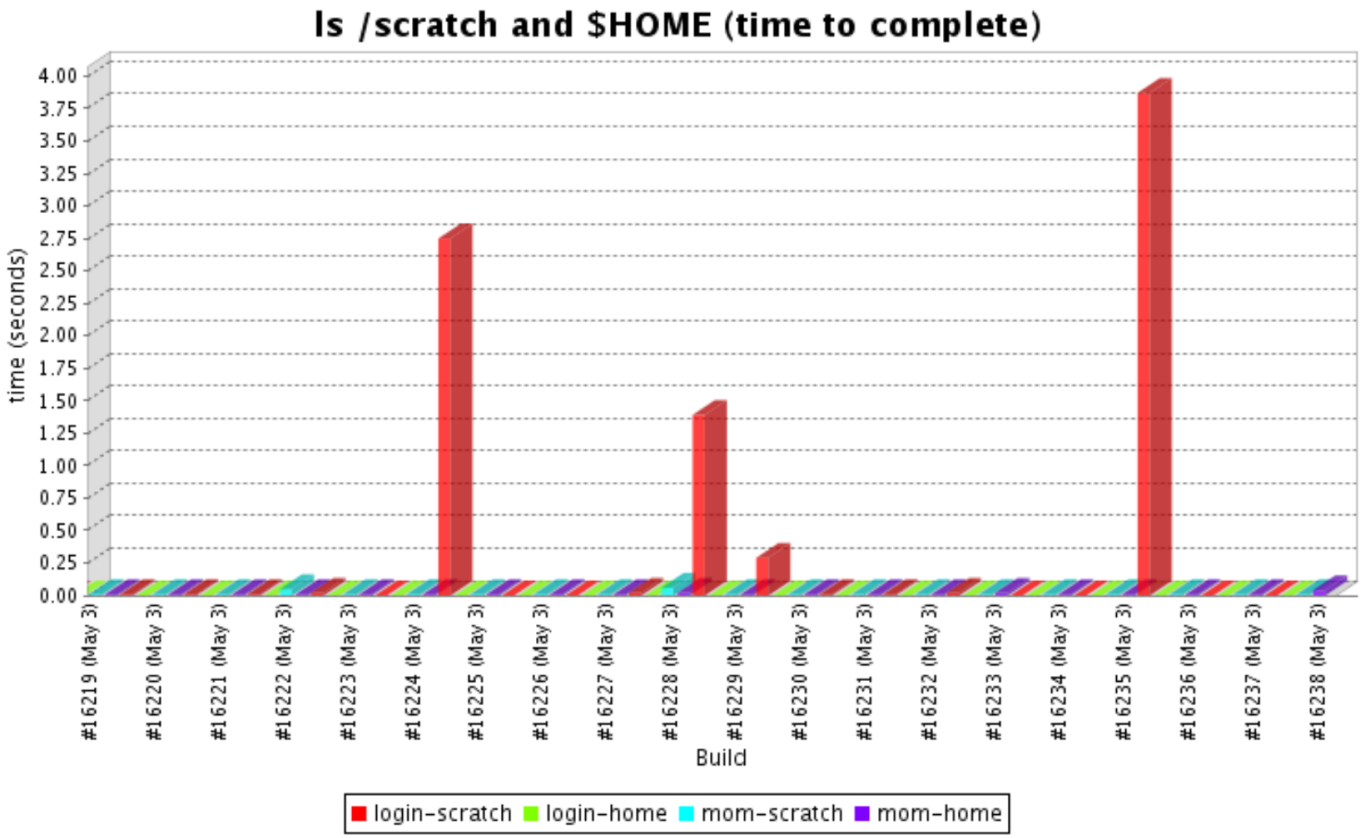
Slow JobLaunch-BW Plot

top

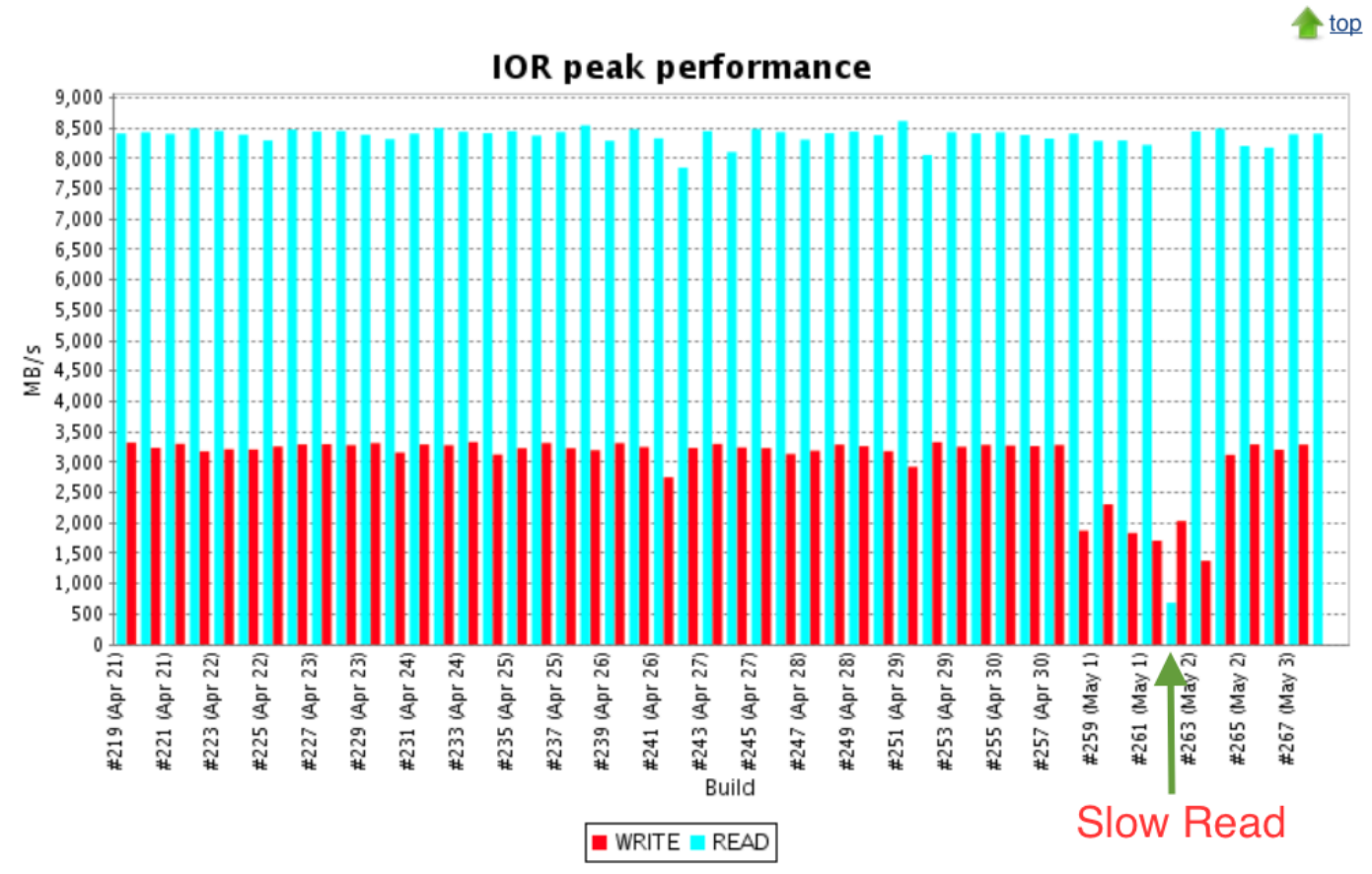


Slow Lustre Scratch from a login node

top



Slow IOR-login Test Example

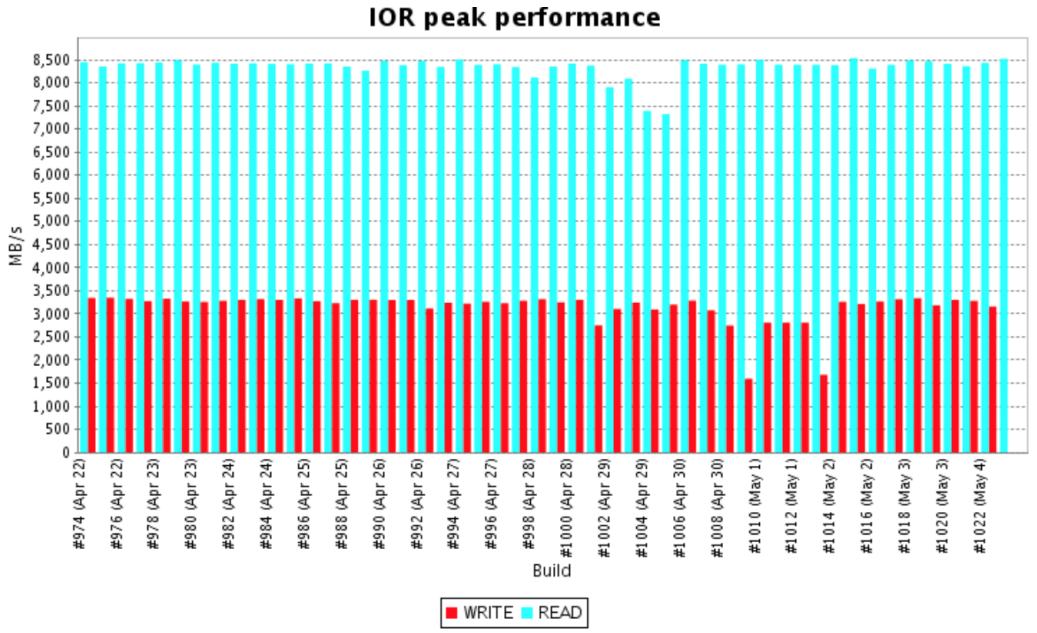


IOR-BlueWaters test failures

Build History trend

find

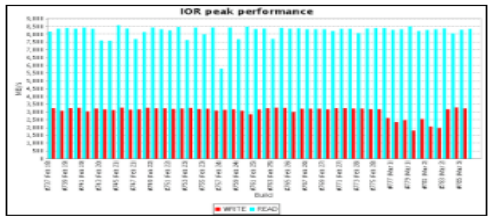
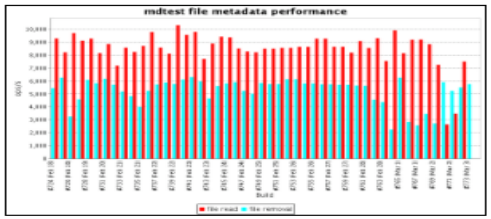
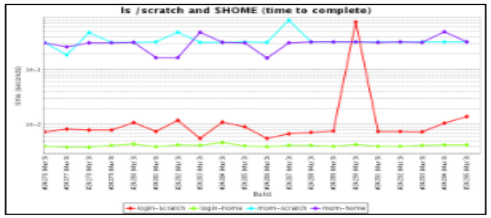
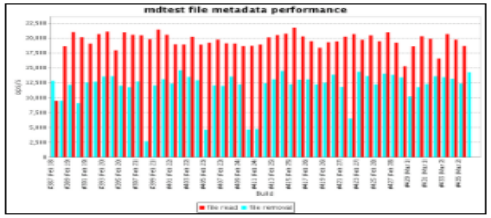
#1023
May 4, 2017 1:53 PM
#1022
May 4, 2017 7:53 AM
#1021
May 4, 2017 1:53 AM
#1020
May 3, 2017 7:53 PM
#1019
May 3, 2017 1:53 PM
#1018
May 3, 2017 7:53 AM
#1017
May 3, 2017 1:53 AM
#1016
May 2, 2017 7:53 PM
#1015
May 2, 2017 1:53 PM
#1014
May 2, 2017 7:53 AM
#1013
May 2, 2017 2:41 AM
#1012
May 1, 2017 7:53 PM



Two Failed Tests that went over 10 minute wallclock.
These failures indicate an issue with the filesystem.

Plot Usage : Wiki Filesystem dashboard

Jenkins Monitoring, see also : http://bwjenkins.ncsa.illinois.edu/view/BlueWaters/

scratch	 higher is better. If low only once, build might have used busy OSTs.	 higher is better. Trouble if mdtest and IOR rates are both low.
scratch and $HOME ls test		 Low is better. This is a good measure of how the filesystem "feels" to users. Times < 1.0s are generally good. These tests are on a 5 minute interval so this graph is approximately the last hour's performance.
home	No IOR testing: We don't do large block i/o on $HOME.	 higher is better.

Conclusion

- Regression testing framework, enabling:
  - Reproducibility of tests
  - Regression testing
  - Rapid reaction ← more tests are added as needed
- Has been deployed at
  - Blue Waters at NCSA
  - NICS
  - Compute and Data Environment for Science (CADES) at ORNL (in progress)
- Future work: integrate with other change managers