

# HPC Systems Engineering, Suffering, and Administration

## Abstract:

Systems chasing exascale often leave their administrators chasing yottascale problems. This BOF is a forum for the administrators, systems programmers, and support staff behind some of the largest machines in the world to share solutions and approaches to some of their most vexing issues and meet other members of the community. This year we are focusing on how to best share with others in the community; training users, new administrators, and peers; management of expectations; and discussing new tools, tricks, and troubles.

## Participants:

Matt Ezell, HPC Systems Administrator, Oak Ridge National Laboratory  
Adam Hough, Senior Computer Systems Engineer, Petroleum Geo-Services (PGS)  
Daniel Kamalic, Manager of Research Computing, Boston University College of Engineering  
Jason Kincl, HPC Systems Administrator, Oak Ridge National Laboratory  
William Scullin, Senior HPC Systems Administrator, Argonne National Laboratory  
Jenett Tillotson, Senior Cluster Systems Administrator, Indiana University  
Isaac Traxler, Manager of Storage and Infrastructure for HPC @ LSU, HPC @ LONI  
Adam Yates, Manager of Information Technology, CCT @ LSU

## Long Description:

Since SC08, this BOF has grown from a meeting focused on the IBM Blue Gene to a general discussion of issues at scale. Each year has been a learning experience with it starting off as pure discussion; moving to a series of presentations about Top 25 systems; changing to a panel of tools developers and senior administrators use with lots of discussion on the usage. The goal of connecting the system administrators and engineers who support HPC platforms for scientific computing and build a larger community has remained the same.

Attendance has varied from under twenty to over sixty. Recent years have presented a challenge due to funding issues generally restricting the availability of systems administrators and engineers to travel. A core group is committed to promoting, supporting the organization of this community, and helping develop the tangibles:

<participant info>

## Description of the session format:

This session will be a panel of experts with short presentations. We will start with slides from each presenter giving their background and the current machines they work with. There will be one slide that is a list of tools and resources which the presenters have collected and worked with from their varied experience. We expect this to

take thirty to forty minutes.

The last forty to sixty minutes will be a roundtable discussion of best practices, how to recruit and train young system administrators, and sharing problems that others might have already solved. This part of the session will continue the goal of fostering a community of experts that may call on each other for brainstorming and suggestions at need.

## Deliverables:

Our deliverables will be a report summarizing the conclusions drawn during the meeting, what the problems are, what is working well, what the solutions are and recommended next steps. Another deliverable will be a tools matrix based on tools each institution/sys-admin uses. We will send around sign in sheets with a checkbox if the attendee would like to become part of a community of system administrators which would be a goto for never experienced problems or problems with what seems to be no solution.

# SC13 BOF Table

Tools		ANL	BU	CCT	IU	LSU HPC	ORNL	PGS
<b>Monitoring</b>	Nagios	X		X	X		X	X
	Xymon					X		
	Zabbix				X			
	ZenOSS				X			
<b>Inventory Tracking</b>	HP Service Manager							X
	openDCIM						X	
	RackMonkey							
	Racktables			G		X		
<b>Syslog Analysis</b>	Log Analyzer					X		
	SEC				X	X	X	
	Splunk						X	
<b>Ticket System</b>	HP Services Manager							X
	OTRS					X		
	RT	X	X		X		X	
<b>Deployment</b>	Bright Cluster Manager						X	
	Cobbler		X	X		X	X	
	GeDI						X	
	RHN Sattelite			X	X	X		

	SystemImager		X					
	Xcat				X	X		X
	Warewulf			X				
<b>Config Management</b>	Bcfg2	X					X	
	Cfengine						X	X
	Chef							
	Puppet		X			L		
	Salt	G				G		
<b>Resource Management</b>	Cobalt	X						
	LoadLeveler				X	X		
	Moab			X	X	X	X	
	OpenStack		X	G				
	Torque			X	X	X	X	
	Univa Grid Engine		X					X
<b>File Systems</b>	GPFS	X	X		X	X		X
	Lustre							
	Panasas					X		X
	WAFL			X				
<b>Documentation</b>	DokuWiki					X	X	
	MediaWiki			X		X		
	Moin		X					
<b>Backup</b>	Dirvish					X		
	TSM			X		X		

Site Legend

Symbol Legend

<b>ANL</b>	Argonne National Laboratory
<b>BU</b>	Boston University College of Engineering
<b>CCT</b>	LSU Center for Computation & Technology
<b>IU</b>	Indiana University
<b>LSU HPC</b>	LSU High Performance Computing
<b>ORNL</b>	Oak Ridge National Laboratory
<b>PGS</b>	Petroleum Geo-Services

<b>X</b>	Use
<b>L</b>	Leaving
<b>G</b>	Going to

### Site Legend

- **ANL** - Argonne National Laboratory
- **BU** - Boston University College of Engineering
- **CCT** - LSU Center for Computation & Technology
- **IU** - Indiana University
- **LSU HPC** - LSU High Performance Computing
- **ORNL** - Oak Ridge National Laboratory
- **PGS** - Petroleum Geo-Services

### Symbol Legend

- **X** - in use
- **L** - leaving
- **G** - going to

## Tools

### Monitoring

Nagios (<http://www.nagios.org>) [ANL] [CCT] [IU] [ORNL] [PGS]  
 Xymon (<http://xymon.org>) [LSU HPC]  
 Zabbix ([https://www.zabbix.org/wiki/Main\\_Page](https://www.zabbix.org/wiki/Main_Page)) [IU]  
 ZenOSS (<http://www.zenoss.com/>) [IU]

### Inventory/Tracking

[PGS]  
 HP Service Manager (<http://h30499.www3.hp.com/t5/HP-Service-Manager/ct-p/sws-SMgr>)  
 openDCIM (<http://www.opendcim.org>) [ORNL]  
 RackMonkey (<https://flux.org.uk/projects/rackmonkey/>)  
 Racktables (<http://racktables.org>) [LSU HPC]

## Syslog Analysis

Log Analyzer (<http://logalyzer.adiscon.com>) [LSU HPC]  
SEC (<http://simple-evcorr.sourceforge.net>) [IU] [LSU HPC] [ORNL]  
Splunk (<http://www.splunk.com>) [ORNL]

## Trouble Tickets

HP Service Manager (<http://goo.gl/SGqBAw>) [PGS]  
OTRS (<http://www.otrs.com/>) [LSU HPC]  
RT (<http://bestpractical.com/rt/>) [ANL] [BU] [IU] [ORNL]

## Deployment

Bright Cluster Manager (<http://www.brightcomputing.com>) [ORNL]  
Cobbler (<https://fedorahosted.org/cobbler/>) [BU] [CCT] [LSU HPC] [ORNL]  
GeDI [ORNL]  
RHN Satellite (<https://access.redhat.com/home>) [CCT] [IU][LSU HPC]  
SystemImager (<http://www.systemimager.org>) [BU]  
Xcat (<http://sourceforge.net/projects/xcat/>) [IU] [LSU HPC] [PGS]  
Warewulf (<http://warewulf.lbl.gov/trac>) [CCT]

## Config Management

Bcfg2 (<http://trac.mcs.anl.gov/projects/bcfg2/>) [ANL] [ORNL]  
Cfengine (<http://cfengine.com/>) [ORNL] [PGS]  
Chef (<http://wiki.opscode.com/display/chef/Home>)  
Puppet (<https://puppetlabs.com>) [BU] [LSU HPC]  
Salt (<http://saltstack.com/community.html>) [LSU HPC]

## Resource Management

Cobalt (<http://trac.mcs.anl.gov/projects/cobalt>) [ANL]  
LoadLeveler (<http://www-03.ibm.com/systems/software/loadleveler/>) [IU] [LSU HPC]  
Moab (<http://www.adaptivecomputing.com/products/moab/>) [CCT] [IU] [LSU HPC] [ORNL]  
OpenStack (<http://www.openstack.org>) [BU]  
TORQUE (<http://www.adaptivecomputing.com/products/open-source/torque/>) [CCT] [IU] [LSU HPC] [ORNL]  
Univa Grid Engine (<http://www.univa.com/products/grid-engine.php>) [BU] [PGS]

## File Systems

GPFS (<http://www-03.ibm.com/systems/software/gpfs/>) [ANL] [BU] [IU] [LSU HPC] [PGS]  
Lustre (<http://www.opensfs.org/lustre/>) [IU] [LSU HPC] [ORNL]  
Panasas (<http://panasas.com>) [LSU HPC] [PGS]

## Documentation

Dokuwiki (<https://www.dokuwiki.org/dokuwiki>) [LSU HPC] [ORNL]  
MediaWiki (<http://www.mediawiki.org/>) [CCT] [LSU HPC]  
Moin (<http://www.moin.org>) [BU]

## Backup

Dirvish (<http://www.dirvish.org>) [LSU HPC]

TSM (<http://www-03.ibm.com/software/products/en/tivostormana/>) [CCT] [LSU HPC] [PGS]

## Speakers & Topics

### Warewulf- Adam Yates

Warewulf is a system / cluster deployment toolkit made by the folks from Berkley.

It focuses on the concept of a centralized data store for all of your configuration data as well as system images. Your node filesystem is compressed separately from kernel bootstrap images. There are multiple modules that you can add for additional feature sets outside of the basic data store including provisioning, IPMI, monitoring, ICR (Intel cluster ready), node health checks integrated with your favorite job scheduler and a few others. It allows for both stateless and stateful provisioning of a system. It also provides some scripts for auto-generation of node filesystems based around RHEL and its flavors (CentOS, SL, etc.)

Jeff Layton with HPC Admin Magazine did a great series on building a cluster from bare metal to up and running science within an hour or two:

There are four parts in total; beginning with the first:

<http://www.admin-magazine.com/HPC/Articles/Warewulf-Cluster-Manager-Master-and-Compute-Nodes>

### OpenStack - Daniel Kamalic

OpenStack is a tool for building public/private Infrastructure as a Service (IaaS) clouds. The technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering users to provision resources through a web interface.

We will present a plan for a new nonprofit public cloud based on OpenStack -- the Massachusetts Open Cloud (MOC), to be deployed at MGHPCC, and soon to be renamed to reflect an initial dual-datacenter deployment together with UTK/ORNL. The current dominant single-provider public cloud is over five times the size of the

next fifteen largest clouds combined, and presents barriers to innovation as tenants are tied to the infrastructure deemed profitable by that cloud provider. Those infrastructures have not typically catered to HPC, leading many sysadmins to consider the cloud model unsuitable for HPC. The goal of the MOC is to create a non-profit public cloud marketplace where participants from research institutions, government, and private industry can participate in implementation and operations, can deploy the infrastructures they choose, and can each determine how to charge for their services.

OpenStack's architecture provides inherent security between untrusted users, tenants, and even administrators. When managing friendly user communities within a private network, HPC sysadmins may not need to worry much about security, but OpenStack's security model facilitates the sharing of resources between extremely large sets of user communities, including corporate partners. We can lead into a discussion of general security best practices.

<http://www.slideshare.net/kamesh001/massachus>

### **Blue Gene Control System - William Scullin**

The Blue Gene control system is a swiss army knife for Blue Gene administrators with a centralized database facility collecting user job information, hardware and software events, performance data, system inventory, and environmental statistics. While closely tied to the IBM Blue Gene platform, the control system is fully open source and other vendors or projects could abstract it for use with other platforms to provide functionality unavailable even in combination with several other open and closed source tools.

### **Monitoring - Isaac Traxler**

Monitoring is a critical part of system administration. A monitoring system should be easy to bring up, easy to maintain, low resource utilization, scalable, easy to extend, reliable and most importantly -- good at letting you know what needs attention. After trying many systems I have settled on Xymon. As an old Big Brother user, Xymon was an easy stretch. Strangely we were in a long Zabbix rollout that was looking like it might never end and we kept missing important events. A Xymon monitor was set up in a few days to keep us from missing important alerts while we worked on Zabbix. Eventually the irony occurred to us and we tabled Zabbix and focused on Xymon.

The best feature of Xymon is its matrix of things that need attention (imagine a matrix with machines for rows and problem categories for columns and colored icons to show which items were broken and how broken they are). This screen has its own dedicated monitor on my desktop.

The Xymon central piece is written as a collection of C programs that communicate via shared memory. Events flow through the shared memory like it was an event bus. Xymon uses very little cpu and memory (a dual Xeon with 2 GB of memory sits idle monitoring 100s of servers). Zabbix maxed out a 6 core, 12 GB machine at about 1/4 of the load our current Xymon server supports.

Xymon natively is a client/server model with a client running on each monitored host (no client necessary for things that can be tested externally: ping, ssh, http, smtp, ...). The base client handles cpu, memory, disk space, inodes, and a few other things. Xymon client is designed to embrace external tests. A test is essentially anything that can

run and conclude good or bad and then open a network connection back to the server and deliver a text result. Extensions can be written in shell, compiled languages, interpretive languages, practically anything. I use Simple Event Correlator to send Xymon events. Xymon has a large number of pre-written and tested external tests that can often just be downloaded and used (without change).

Xymon can react to events via its alerts system. The alert system is based on pattern matching with the ability to do things like send an e-mail, page, send SMS and/or run an arbitrary script (great for managing who gets what alarm when).

When an event occurs, along with basic status, the event can send additional info (disk event might send a full `df -h` along). Xymon keeps a history of every event change that you can then look back over. Patterns become obvious (every Sunday night this machine goes down). Many tests by default graph via MRTG/RRD with graphs for different time periods. Custom tests can feed data into RRD and produce graphs.

In summary, Xymon is lightweight, easy to deploy, easy to extend, easy to configure alerting and easy to use to discover the current state of things.

### **Backup: Dirvish - Isaac Traxler**

Backup is an important function that often receives a lot of attention. Restoration on the other hand is easily overlooked. I find that in the stressful times of needing to restore that trying to master obscure commands is a difficult thing. Wouldn't it be wonderful if backups could be scanned and restored using commands that we use every day?

Dirvish is built on top of rsync. It uses rsync to build an initial backup. It then takes advantage of the feature of rsync that allows a new sync tree can be made based on the existing tree (duplicates are hardlinked instead of rsynced). This is the way Time Machine (OS X) works. Dirvish is Time Machine without the gui.

Since Dirvish builds trees for backups (I use a bank per machine and a vault per filesystem). You can use normal `cd`, `find`, `locate` commands to discover what file(s) you want to restore. Restores can be done via `scp` or `rsync`. This means that in time of stress, a restore means that I do not have to deal with obscure commands.

Dirvish has very minimal hardware requirements. I use an old server that I added a Sata controller and 4 3TB drives to (drives are in a software raid 5 configuration). This is a very economical storage solution. Pricing combined with the ease of use make Dirvish very attractive. If you are interested in more info on Dirvish, please visit <http://isaac.lsu.edu/dirvish>.

### **The Importance of Configuration Management - Matt Ezell**

Configuration management is essential to building and maintaining consistent and reproducible systems.

Configuration management can help ease the following common tasks:

- Initial system deployment
- Configuration consistency
- System documentation
- High availability
- Security monitoring
- Security patching



- Capacity planning
- Disaster recovery

It is not required to configure a complicated configuration management package to get started; simple shell scripts to push out configuration files can be a good first step toward configuration management. As environments become more complex, the benefits of configuration management become more obvious.

Many open-source configuration management systems exist today, including cfengine, puppet, chef, bcfg2, salt, and ansible. Choosing the right solution for a given environment can be difficult, as each tool has its own strengths and weaknesses.