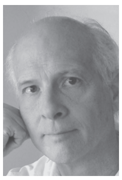
Cory Lueninghoener on Managing HPC Clusters

An Interview

RIK FARROW


Rik is the editor of *;login:*.
rik@usenix.org


Cory Lueninghoener is the lead of the HPC Architecture Team at Los Alamos National Laboratory. He has co-hosted the LISA Configuration Management Workshop for the past few years and is always looking for more efficient ways to manage acres of computers with only a handful of people.
cluening@gmail.com

I first met Cory during a LISA conference, when he volunteered to write an article about getting started with configuration management (CM) for ;login: [1]. I could tell he was passionate about CM from his article, and through his participation in the Configuration Management Workshop, and the LISA short topics book [2] that he had written with Narayan Desai.

I knew that Cory worked at Los Alamos National Labs on high performance computing (HPC) clusters, and that got me even more interested. I've learned a lot about modern clusters through working with people on Hadoop projects, but I thought for sure that HPC was very different from a Hadoop cluster. And it turned out that I was right.

Rik: I am guessing that you didn't start out by managing large numbers of HPC clusters. Tell us a little bit about how you wound up in this position?

Cory: You're right! I started out managing just my Linux desktop machine when I was in college, and I expanded that by a couple orders of magnitude as a summer student with Argonne National Laboratory's Mathematics and Computer Science Systems Group. I started managing HPC systems as a grad student, with a few little clusters. That number grew as I worked with Argonne's Leadership Computing Facility and Los Alamos National Laboratory's High Performance Computing Division. Now I'm on a team at Los Alamos that manages about 20 HPC resources ranging in size from around 20 nodes up to around 10,000 nodes.

Rik: You mention being a grad student, and I am guessing that you weren't in grad school to become a sysadmin. I'm curious about what you were studying before you entered the challenging realm of riding herd on multiple clusters.

Cory: My Master's degree is in computer science, focusing a lot on machine learning and artificial intelligence. But my advisor was a computational chemist, and that's where my interest in high performance computing started: he was just starting to move from large single-system image machines to clusters and had funding to buy some small clusters. I quickly found that I was more interested in helping manage the clusters and work with a wide variety of scientific users than find my own specific project to run on them, and that's what brought me to where I am today.

Rik: More and more people are managing clusters, used either as nodes for VMs, Hadoop (MapReduce), or grid computing. How is managing HPC clusters different?

Cory: Today's HPC clusters are generally tightly coupled systems that consist of a collection of compute nodes, a dedicated high-speed network, one or more global file systems, and a handful of support systems. A queueing system runs on top of the hardware that users submit jobs to. These jobs can range in size from a few tens of processors up to hundreds of thousands of processors, and jobs may run for only a few minutes or for a day or longer. Nodes running these jobs are generally dedicated to a single job at a time, giving each job its maximum resources.

Managing HPC systems can be tricky—once a job is running on a set of nodes, we want to do as little administration work on the nodes as possible. Tightly coupled HPC jobs usually cannot survive a single node reboot, so any needed reboots need to be scheduled while a node is completely idle. Changing software on a node running a job is also avoided, as an unexpectedly heterogeneous job environment can wreak havoc on a job. Even subtle changes can affect jobs: simply running a monitoring script too frequently and without coordination across the cluster can introduce enough operating system jitter to affect large running jobs.

Rik: What is the role of CM in managing HPC? Is it any different from what other people are using CM for, based on your experiences at the Configuration Management Workshop?

Cory: The role of configuration management on an HPC system is very similar to any other site. We use our configuration management tools to push out regular updates, install user-requested software, ensure compliance, and ease system management in general. Some of the details differ, however. For example, I mentioned earlier the importance of ensuring identical software environments on all of our compute nodes. Whereas an administrator of a network of desktop systems may expect his systems to be all unique, or an administrator of a VM hosting cluster may not care what kernel version is running on her systems, we require that all of our systems be as identical as possible. Here at the lab, we also use configuration management as a sort of self-documenting system.

We are able to hand new administrators a copy of our central configuration repository, consisting of all of our modified configuration files, local packages, and anything else that makes one of our systems unique. This quickly gives them a focused view of our configuration and how it differs from a default installation. We can do the same with auditors to demonstrate the level of control we have over our systems and our knowledge of what we are currently running.

Rik: Do your HPC systems run standard operating systems, which presumably makes them easier to manage?

Cory: In general, yes. Using standard Linux distributions makes things easier for a lot of people: users are familiar with developing code on standard distributions; vendors support compilers, debuggers, and other development tools on specific distributions; and administrators are much more familiar with standard distributions than company-proprietary operating systems. Cluster vendors will often customize the distributions, though, to support the cluster's high-speed interconnect, provide more compilers or other tools than are available by default, or improve the user experience in other ways.

Rik: You mentioned that HPC systems are broken into three roles: management, execution engines, and storage. Does storage get managed separately, or are all three categories handled as if they were a single system?

Cory: Yeah, that is a pretty standard way to split up a system. While management systems like login nodes, resource management nodes, and boot servers are almost always treated as part of the cluster, the storage can be very site-specific. On smaller clusters, a storage system consisting of a single node that serves NFS exports to the compute nodes is a common simple solution. In that case, treating the storage server as another management node is the easiest approach. On larger systems, having a dedicated storage cluster that serves up file systems like GPFS, Panasas, or Lustre is more common. These systems are generally treated as separate resources and managed as more independent entities, especially if they serve their file systems to multiple clusters.

Rik: In HPC, I've learned (from listening to talks at LISA) just how critical the network is in maintaining good performance. How do you monitor the network?

Cory: Really intense HPC jobs do a lot of communication between each of the nodes in the job, so a reliable high speed network is indeed critical. The amount of network monitoring that can be done on a cluster depends on what interconnect is used. Infiniband is a popular interconnect on commodity clusters, and OpenFabrics provides a number of tools to help monitor connectivity, speed, errors, and other aspects of an Infiniband network. We use these tools to monitor our networks and automatically disable nodes that are not performing optimally. Higher end systems, such as Cray's MPPs and IBM's Blue Gene systems, use custom high speed networks that are designed to require less monitoring and provide more information when a problem does occur. Either way, keeping the high speed network working is an important part of running an HPC system.

References

[1] Cory Lueninghoener, "Getting Started with Configuration Management," April 2011, vol. 36, no. 2: https://www.usenix.org/publications/login/april-2011-volume-36-number-2/getting-started-configuration-management.

[2] Narayan Desai and Cory Lueninghoener, *Configuration Management with Bcfg2*: https://www.usenix.org/lisa/books/configuration-management-bcfg2.