LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-702862

Further Automate Planned Cluster Maintenance to Minimize System Downtime during Maintenance Windows

R. Springmeyer

September 13, 2016

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Contents

Introduction

This report documents the integration and testing of the automated update process of compute clusters in LC to minimize impact to user productivity.

Description: A set of scripts will be written and deployed to further standardize cluster maintenance activities and minimize downtime during planned maintenance windows.

Completion Criteria: When the scripts have been deployed and used during planned maintenance windows and a timing comparison is completed between the existing process and the new more automated process, this milestone is complete.

This milestone was completed on Aug 23, 2016 on the new CTS1 cluster called Jade when a request to upgrade the version of TOSS 3 was initiated while SWL jobs and normal user jobs were running. Jobs that were running when the update to the system began continued to run to completion. New jobs on the cluster started on the new release of TOSS 3. No system administrator action was required. Current update procedures in TOSS 2 begin by killing all users jobs. Then all diskfull nodes are updated, which can take a few hours. Only after the updates are applied are all nodes are rebooted, and then finally put back into service. A system administrator is required for all steps. In terms of human time spent during a cluster OS update, the TOSS 3 automated procedure on Jade took 0 FTE hours. Doing the same update without the Toss Update Tool would have required 4 FTE hours.

Automated update procedure (Toss Update Tool)

The Toss Update Tool is an automated update procedure for LC's production clusters of 2SU or more. The automated update process allows for systems to be updated more often, and without admin intervention. Existing user jobs will be allowed to finish, users should not experience hangs, and new jobs will only be run on the new TOSS release scheduled for that day. This will be rolled out into production initially with CTS1, and TLCC2 systems as they upgrade to TOSS 3.0. System administrator involvement should be reduced to investigating nodes that fail to reboot properly, or fail their node diagnostics at boot time. The Toss update tool was created with input from the Information Management, Graphics, and Security group, System Administration group, and CEA.

Implementation of Toss Update Tool

Novel ways of using RedHat tools were required in order update a cluster while running jobs when every node except one is stateless. Using multipath and iscsi we were able to devise a way to continue running diskless compute nodes while updating the servers that provided their image. Using a squash file system on non-primary management nodes we reduced their time to update to nothing more than a reboot. These RPS nodes act like diskfull nodes that can operate standalone, but are actually stateless nodes. During the update of any node type, state is maintained in the Redis database. This allows nodes to know what they are doing, what they are supposed to do next and how many other nodes in the center are doing certain things. This is critical to ensure that a denial of service is not created on shared resources in a center.

Cluster node detail

Compute Clusters are made up of

- Primary management node - Manages the cluster
- RPS nodes - Provide iscsi targets for diskless nodes
- Login nodes - User interface to the compute cluster
- Gateway nodes - Provide access to public and nfs networks
- Lustre routers - Provide access to lustre file systems
- Compute nodes - Resource for users to allocate

Primary management

The primary management node is the only node in the cluster that will have state. It is responsible for booting the RPS nodes, and possibly other stateless nodes. All configuration management is contained on this node. A Redis server is run on this node to manage the update of the cluster. The use of Redis will allow us to manage multiple updates in a center at once as more clusters begin to use this tool. This node can be installed via kickstart from the "center" management node, or from DVD.

RPS nodes

The RPS nodes have similar if not the same hardware as the primary management node. These nodes have local disk to serve block devices to other stateless nodes. RPS nodes are not installed. They run in a squash file-system. This means they get an image from the

primary management node at boot time, copy it to memory, and run from memory. They have no requirements for the primary management node to be running after they boot. To pick up an update, they only need to reboot. They don't need to install updates, and then reboot. All RPS nodes use the same image to boot from.

Login nodes

These nodes boot using the same iscsi image that compute and other stateless nodes use to boot. They have a local disk to provide /tmp and swap space to the multiple users. To update, they only need to reboot.

Gateway, Lustre and Compute nodes

These node types boot off the same iscsi image. This image is served via an iscsi target and multipath'd to allow for RPS nodes to be rebooted without affecting the root file system. To update these nodes, they only need to be rebooted.

Update Process

The Primary management node will create images for the RPS, and iscsi nodes. It will then reboot, and push the images to all RPS nodes. As this is happening, no new jobs will start until all RPS nodes have updated, and the compute nodes have rebooted onto the new image.

RPS nodes wait for their image to become ready; when it does, they will begin rebooting to pick up the new image. At any time at least half of the RPS nodes will be up to continue providing iscsi service.

Login nodes will wait for their image to become ready, and all RPS nodes to update. They will then reboot with a 10 minute warning to users. After the reboot they will put themselves back into service.

Gateway nodes will wait for their image to become ready, and all RPS nodes to update. They will then begin rebooting to pick up the new image. At any time, at most only one quarter of these nodes will be down to allow IP routing to continue. A service called BIRD (a dynamic routing daemon) is run on our nodes to change default routes as gateway nodes reboot.

Lustre routers will wait for their image to become ready, and all RPS nodes to update. They will then begin rebooting to pick up the new image. At any time, at most only one quarter of these nodes will be down to allow Lustre routing to continue. Lustre has built in support for dealing with down routers.

Compute nodes will wait for their image to become ready, and all RPS nodes to update. If they are currently running a job they will finish the job and reboot to pick up the new image. If the node is idle, it will reboot. Drained nodes will be left alone. As the compute nodes reboot, they will put themselves back into service provided they pass diagnostics. Once back in service new jobs are allowed to begin running on the latest TOSS release.

Attachment 1: Milestone Definition Text

Milestone (ID#5591): Further Automate Planned Cluster Maintenance to Minimize System Downtime during Maintenance Windows		
Level: 2	**Fiscal Year**: FY16	**DOE Area/Campaign**: ASC
Completion Date: 8/23/16		
ASC nWBS Subprogram: CSSE		
Participating Sites: LLNL		
Participating Programs/Campaigns: ASC		
Description: A set of scripts will be written and deployed to further standardize cluster maintenance activities and minimize downtime during planned maintenance windows.		
Completion Criteria: When the scripts have been deployed and used during planned maintenance windows and a timing comparison is completed between the existing process and the new more automated process, this milestone is complete.		
Customer: ASC		

Milestone Certification Method: Professional documentation, such as a report or a set of viewgraphs with a written summary, is prepared as a record of milestone completion. The "handoff" of the developed capability (product) to a nuclear weapons stockpile customer is documented.
Supporting Resources: LLNL FOUS staff

Attachment 2: Handoff Letter