Incorporating a Test and Development System Within the Production System

Nicholas P. Cardo, Marco Induni
Swiss National Supercomputer Centre (CSCS)
HPC Operations
Lugano, Switzerland

Abstract—**Test and Development Systems (TDS) often get traded off for investments into more computational capability. However, the value a TDS can contribute to the overall success of a production resource is tremendous. The Swiss National Supercomputer Centre (CSCS) has developed a way to provide TDS capabilities on a Cray CS Storm System by utilizing the production hardware, with only a small investment. An understanding of the system architecture will be provided, leading up to the creation of a TDS on the production hardware, without removing the system from production operations.**

Keywords; TDS, HPC

I. Introduction

Test and Development Systems (TDS) provide a vital functionality allowing both testing and production operations to continue simultaneously without impacting each other. Often times, this capability is traded off due to the additional costs associated with the TDS hardware. But what if the TDS could be built from the production hardware? Gaining the ability to test and port prior to upgrading and entire system is both beneficial to all as well as reduces the overhead and time required to achieve the final upgrade production result.

II. System Description

CSCS manages a pair of Cray CS Storm systems for the Swiss Federal Office of Meteorology and Climatology (MeteoSwiss). The systems are identical to each other and are each self-contained within a 48U cabinet, providing additional failover capabilities at the system/cabinet level. Mechanisms and tools have been put in place to allow for easy movement of the production runs between the two systems. In the event of a problem, the workload is quickly and easily moved to the other system. While one system runs the production workload, the second system is used for research and development activities. In the event of a problem on the production system, these activities are suspended and the production runs switch to the second system.

The computational component of each system consists of 12 hybrid compute nodes. Each node is configured with:

- 8 x NVIDIA Tesla K80 GPUs
- 256 GB of memory
- 2 x Intel Xeon CPU E5-2690 v3

Also included in each system are 5 post-processing nodes without GPUs. Each node is configured with:

- 256 GB of memory
- 2 x Intel Xeon CPU E5-2690 v3

Interactive access is provided by 3 login nodes. Each node is configured with:

- 128 GB of memory
- 2 x Intel Xeon CPU E5-2690 v3

Each system is managed by a pair of management workstations configured in High Availability (H/A). Although the system was first installed with the Cray Advanced Cluster Engine (ACE) system management software, it is now utilizing the Bright Cluster Manager for overall management of the system.

Access to the computational and post-processing resources is managed with the Slurm workload manager. Partitions are defined such that the workload can target a particular type of node appropriate for the computational work to be performed.

Throughout the whole design of the systems, High Availability was a prominent feature. The capabilities to mitigate failures is necessary due to the criticality of the time sensitive end product. Internal to each system, the required number of nodes in each category is less than the number of physical nodes available allowing for the loss of individual nodes within a system without impacting the ability to complete the required workload. All the nodes are managed by a pair of management workstations configure for high availability allowing for the loss of a single management workstation without impacting the ability to complete the required workload. Stepping back to the cabinet level, each cabinet is completely independent of the other. Furthermore, each cabinet has power and cooling specific to each cabinet. Therefore, since the cabinets are independent down to the power and cooling, they too are configured for high availability. Each piece of the design was carefully specified in order to eliminate many single points of failure as possible.

An additional feature in the high availability of these systems is the ability sever any reliance on facility supplied functionality. Internally, this is referred to as emergency mode. There will always be times where external influences could affect certain functionality within the system. In emergency mode, these systems continue to operate independently of external factors.

III. Motivation

The principal end product of these systems in the production weather forecast for Switzerland. The criticality of the results is time sensitive and requires that the forecasting system always remain operational. The nature of the workload leaves only very brief periods where testing of changes may be possible, drastically increasing the time required to perform software validation when updating the system's software stack.

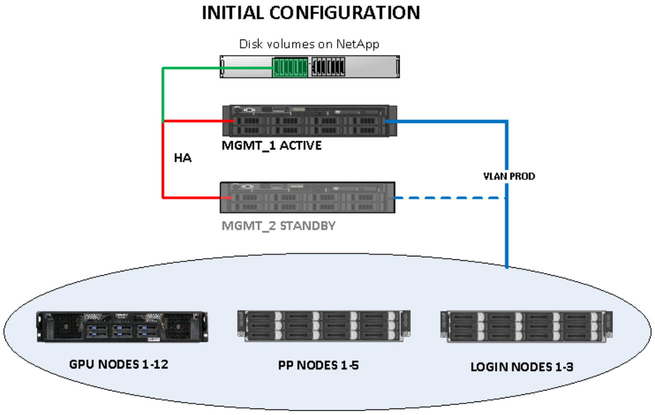
Without a TDS, production operations could be affected while performing testing. This is the classic justification showing the value of a TDS, but in many cases is passed over to acquire additional production computing resources. The need for the functionality, but the money to purchase an independent TDS system was not. The unique hardware configuration of the nodes made cost of purchasing additional nodes not an option.

The difficulties in testing a new version of an operating system under these restrictions becomes monumental. This problem reached a breaking point when it became necessary to upgrade from Red Hat 6 to Red Hat 7. With this upgrade, the entire software stack changed, requiring a complete revalidation of the production workload. This became further complicated by the inability for the ACE to successfully move to Red Hat 7. Without a TDS, this process was dragging on for over 8 months.

As the man-hours mounted and the pain threshold increased with no progress, it became clear that something drastic and innovative would be required in order to move the entire process forward. It is often said that to see the solution, one should take a big step backwards and look at the whole problem with fresh eyes. In this case, it resulted in a simple statement: create a TDS. The dynamics of the problem now became the process of how to create a TDS in order to allow for independent testing under a new operating system rather than how to perform the operating system upgrade. If a TDS were available, then the upgrade process would be simplified. Simple and easy…

IV. Implementation Details

An examination of the hardware within a single system, revealed that there was sufficient hardware within the production system to allow for enough nodes to be removed from production service and used for testing without impacting the ability to meet production needs.



The system includes two System Management Servers connected in High Availability (HA). By breaking the HA capability, two independent System Management Servers can be produced. This allows one to move forward with the upgrade while the other remains in full production. Additional disk drives were added to the server to permit the creation of separate volumes for each of the operating systems.

By adjusting VLAN network configurations, nodes could be managed by either the production Management Server or the TDS Management Server without the need to re-cable network connections. As long as any single node belonged to only one of the two systems, production or TDS, the desired functionality could be achieved. This is a rigid requirement that governs how the nodes are to be used and ensures stable production computing at all times.

With the configuration limitations now understood, the process of moving forward could now begin. The first step was to eliminate probably the largest impediment to the upgrade, ACE. Constantly fighting with the limitations of ACE for the move to Red Hat 7 only continued to work against a successful outcome. However, an important consideration needed to be taken into account, is support. The Cray CS Storm systems were under maintenance with Cray and it was important to retain support for the entire system with a single vendor. Therefore, Bright Cluster Manager (BCM) was selected to manage the system with the licenses obtained through Cray.

With no easy migration path from ACE to BCM, it was determined the best course of action was to perform a fresh install utilizing BCM. The H/A configuration of the system management workstations for one of the systems was broken to allow each to operate independently. This enabled the capability to perform a full upgrade of the operating system and installation of BCM independently on one of the management servers without any impact to production operations.

The next step was to build the base image for the nodes. There are three node types:

1. Login
2. Compute
3. Post-Processing

All three node types were determined have similar software and hardware requirements with the exception being the compute nodes have GPUs installed along with the necessary drivers. To simplify the installation, the decision was made to have one single base image with customizations overlaid on top. The more unique images there are to maintain, the more chances there are for them to become out of sync with each other. By using a single base image, this problem is eliminated and also provides the benefit of reducing the amount of work necessary.

The downside to a fresh install is that all additional software packages that were determined to be necessary over the previous life of the system had to be installed again. While time was gained by developing a single base image, the gains were quickly lost by the nature of determining dependencies on software packages as many were missed from the initial base install. However, as the process progressed, the base image became more and more in line with the needed user environment.

Each of the systems is configured with 1 additional node of each type that is not required in to meet the computational requirements of the production suite. This was a redundancy choice made when the system was originally configured in order to mitigate node failures. These nodes could be taken down on the production system and brought up on the TDS. This is possible as each of the two management workstations knows the configuration of the entire system. Therefore, any node, or quantity of nodes, could be booted into either the TDS or production system. This provided the means to obtain nodes for the development and testing of an image for production use. At this point, a safe method for performing the upgrade of operating system was gained without impacting production operations.

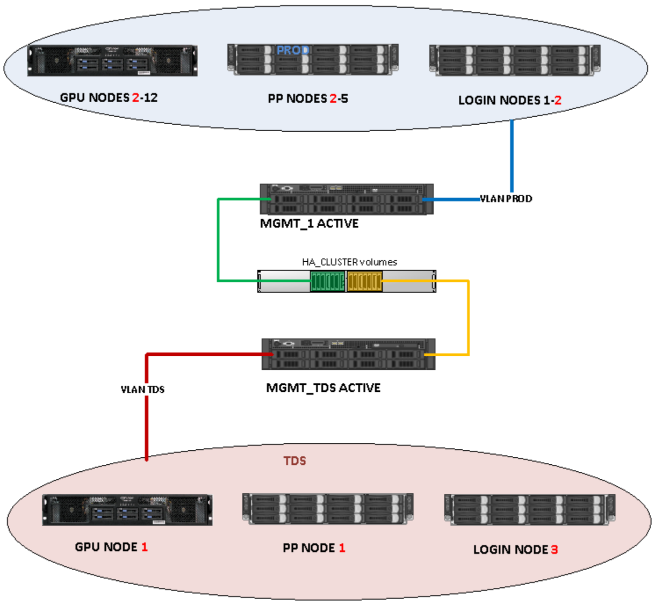
Once the operating system was complete, the next step focused on getting the applications rebuilt and tested. Initially, 1 node of each type was booted into with the upgraded operating system to provide a software development environment to build the applications. Limited testing of the applications was also possible provided the scale of the applications fit within a single node. Again, this activity was able to progress without impacting production operations.

After building and initial testing of the applications, it was time for scale up tests to full production status. As previously mentioned, one system is used for production operations while the second is used for research and development activities. Only in the event of a problem on the production system will the research and development system then become the production system. Therefore, all this work has been performed on the R&D system knowing that if necessary all nodes could be turned over for production use with a quick reboot of the nodes.

Eventually the time arrived to take more nodes for testing. The nodes available for production R&D use were decreased and the nodes switch over to the TDS for testing. The quantity of nodes was slowly increased until eventually the full the system was booted for testing under the TDS. Until the time came to fully boot all nodes under the TDS, production operations remained in service on the R&D system.

Because of the relatively quick reboot of the nodes, the system could be easily scheduled to allow for full-scale testing for a of a day and then returned back for full production R&D work. The final TDS enabled configuration provided the means to complete the necessary application rebuilds and testing without interrupting production operations.

FINAL CONFIGURATION WITH TDS



Once certified for production use, the time came for the big switch, keeping in mind there is a second system running. Through a careful choreography, R&D activities and the production workload swapped systems. This now put the production workload on Red Hat 7 while the R&D activities continued on Red Hat 6. Keep in mind that with a quick reboot and VLAN adjustment, the Red Hat 7 system could be returned back to Red Hat 6 in the event of problems. After a few days, the Red Hat 7 system was determined to be stable while running the full production workload, opening the door to upgrade the second system. The Red Hat 6 system was taken down and booted under Red Hat 7. Once again with careful choreography, the production and R&D activities swapped systems, restoring them back to the primary usage. For safety, one system was kept with the ability boot back into the Red Hat 6 environment. Once it was determined to no longer be necessary to return to Red Hat 6, the H/A configuration of the management server was restored.

V. SUMMARY

This implementation has been highly successful and provided a means to rebuild and test the production weather suite on new levels of the operating system while maintaining production and R&D operations. With everything in place, the entire process can be used over and over again, allowing for TDS activities to occur in parallel with production operations. This solution provides safety as with a simple node reboot, the previous software levels are restored.

The only requirement for this solution to work is to have a second management server available and the capability to use VLANs in the network configuration. In the case highlighted within this paper, the systems came preconfigured with two management workstations configured for high availability. The only money investment into this solution was for additional drives to create a second volume in order to keep the operating systems separated and provide the capability of switching between them.

Acknowledgment

This work was supported by the Swiss National Supercomputing Centre (CSCS) and the Swiss Federal Office of Meteorology and Climatology (MeteoSwiss).

A special thanks to Cray Inc. and Bright Computing for their assistance with the upgrade and development of procedures to make this possible.

References

1. NVIDIA Corporation, “NVIDIA,” [online].
Available: https://www.nvidia.com
2. Cray Inc., “Cray,”, [online].
Available: https://www.cray.com
3. Red Hat Inc., “Red Hat,” [online].
Available: https://www.redhat.com
4. Federal Office of Meteorology and Climatology, [online].
Available: http://www.meteoswiss.admin.ch
5. Bright Computing, “Bright Cluster Manager,” [online].
Available: http://www.brightcomputing.com/product-offerings/bright-cluster-manager-for-hpc