Five Essential Strategies for Successful HPC Clusters

by Douglas Eadline, April 2014
Executive Summary

A successful HPC cluster is a powerful asset for an organization. At the same time, these powerful racks present a multifaceted resource to manage. If not properly managed, software complexity, cluster growth, scalability, and system heterogeneity can introduce project delays and reduce the overall productivity of an organization. At the same time, cloud computing models as well as the processing of Hadoop workloads are emerging challenges that can stifle business agility if not properly implemented.

The following essential strategies are guidelines for the effective operation of an HPC cluster resource:

1. **Plan To Manage the Cost of Software Complexity**
   Avoid creating your own cluster from scratch. An expensive expert administrative skill set is required for this approach. Consider a unified and fully integrated solution that does not require advanced administrator skills or large amounts of time to establish a production-ready HPC cluster.

2. **Plan for Scalable Growth**
   Assume the cluster will grow in the future and make sure you can accommodate growth. Many homegrown tools do not scale, or are complicated to use as clusters grow and change. The best approach is to use a tested and unified technology that offers a low-overhead and scalable approach for managing clusters. Growth should not impact your ability to administer the cluster as change occurs.

3. **Plan to Manage Heterogeneous Hardware/Software Solutions**
   Heterogeneous hardware is now present in virtually all clusters. Make sure you can monitor all hardware on all installed clusters in a consistent fashion. With extra work and expertise, some open source tools can be customized for this task. There are few versatile and robust tools with a single comprehensive GUI or CLI interface that can consistently manage all popular HPC hardware and software. Any monitoring solution should not interfere with HPC workloads.

Contents

Executive Summary ............. 2
Plan To Manage the Cost of Software Complexity .............................................. 2
Plan for Scalable Growth ........................................... 2
Plan to Manage Heterogeneous Hardware/Software Solutions .................... 2
Be Ready for the Cloud................................. 3
Have an answer for the Hadoop Question ........................................... 3

Introduction and Background . . 3
Defining Cluster Success Factors ....... 3
HPC Cluster Building Blocks ............. 4
Deployment Methodologies .................. 5
Essential HPC Cluster Strategies ........... 5
Strategy 1: Plan To Manage the Impact of Software Complexity ............... 5
Strategy 2: Plan for Scalable Growth ........................................... 7
Strategy 3: Plan to Manage Heterogeneous Hardware/Software Solutions ............. 9
Strategy 4: Be Ready for the Cloud .... 11
Strategy 5: Have an answer for the Hadoop Question............................ 12

Conclusion .................. 14
4. Be Ready for the Cloud
Make sure you use Cloud services that are designed for HPC applications including high-bandwidth, low-latency networking, exclusive node use, and high performance compute/storage capabilities for your application set. Develop a very flexible and quick Cloud provisioning scheme that mirrors your local systems as much as possible, and is integrated with the existing workload manager. Monitoring is important for all Cloud usage. An ideal solution is where your existing cluster can be seamlessly extended into the Cloud and managed/monitored in the same way as local clusters. That way, users can expect immediate access to Cloud resources.

5. Have an answer for the Hadoop Question
Plan for user requests for Hadoop or HBase capabilities in the near future. Hadoop configuration and management is very different than that of HPC clusters. Develop a method to easily deploy, start, stop, and manage a Hadoop cluster to avoid costly delays and configuration headaches. Hadoop clusters have more “moving software parts” than HPC clusters; any Hadoop installation should fit into an existing cluster provisioning and monitoring environment and not require administrators to build Hadoop systems from scratch.

Bright Cluster Manager addresses the above strategies remarkably well and allows HPC and Hadoop clusters to be easily created, monitored, and maintained using a single comprehensive user interface. Administrators can focus on more sophisticated, value-adding tasks rather than developing homegrown solutions that may cause problems as clusters grow and change. The end result is an efficient and successful HPC cluster that maximizes user productivity.

Introduction and Background
Extracting useful work from raw hardware can be challenging. On the surface, racking and stacking hardware seems easy enough. Connecting a handful of machines with Ethernet and InfiniBand, and installing the Linux OS and HPC software may present some challenges, but with a little effort it is not too difficult to get a system up and running. In the early days of Linux HPC clustering, such an approach served as an inexpensive way to bring HPC capabilities to an organization. However, many of those early HPC cluster administrators failed to realize the actual burden these “home brew” approaches placed on their organizations and budgets. Today, cluster-based HPC has evolved into a mature market segment with many more hardware and software choices than in the past. The expanded decision space puts further pressure on organizations to “get it right” when staging any type of HPC resource. In order to help managers, administrators, and users deploy and operate successful HPC clusters, this guide presents five essential strategies to help navigate today’s challenging HPC environment. In order to understand the motivations behind these strategies, the factors that define a successful cluster will be discussed below.

Defining Cluster Success Factors
The difference between a collection of raw hardware and a successful HPC cluster can be quite striking. When an HPC resource operates in a production environment, there are several factors that determine success. The first, and perhaps most important, is utilization rate (i.e. how much productive work is done on the cluster). If software or hardware issues cause downtime or reduced capacity, users often experience delays in research or engineering deadlines. By design, clusters should tolerate some hardware loss without affecting users.

The ability to easily change, update, or add to the cluster contributes to the overall utilization rate. This requirement is where home-brew systems can fail. Very often, the highly customized nature of home-brew systems does not tolerate change, and can cause significant downtime while updates are made by hand. A successful cluster must be able to tolerate change.
When the cluster is running, it is important to be able to monitor and maintain the system. Since clusters are built from disparate components, the management interface must handle multiple technologies from multiple vendors. Oftentimes this responsibility falls on the system administrators who must create custom (and sometimes complicated) scripts that glue together information streams coming from various points in the cluster. A successful cluster should provide tools that simplify the administrator’s workload, rather than make it more complex.

The true cost of operating a successful HPC cluster extends beyond the initial hardware purchase or power budget. A truly well run and efficient cluster also minimizes the amount of time, resources, and level of expertise administrators need to detect and mitigate issues within the cluster.

Users will request new software tools or applications. These often have library dependency chains. New compute and storage hardware will also be added over time. Administrative practices that can facilitate change without huge disruptions are essential. Home brew systems often operate on a “critical path” of software packages where changes often cause issues across a spectrum of applications. A successful cluster should accommodate user’s needs without undue downtime.

Finally, a successful cluster also minimizes the administrative costs required to deliver these success factors. The true cost of operating a successful HPC cluster extends beyond the initial hardware purchase or power budget. A truly well run and efficient cluster also minimizes the amount of time, resources, and level of expertise administrators need to detect and mitigate issues within the cluster.

HPC Cluster Building Blocks

The basic HPC cluster consists of at least one management/login node connected to a network of many worker nodes. Depending on the size of the cluster, there may be multiple management nodes used to run cluster-wide services, such as monitoring, workflow, and storage services. The login nodes are used to accommodate users. User jobs are submitted from the login nodes to the worker nodes via a workload scheduler.

Cluster building blocks have changed in recent years. The hardware options now include multicore Intel x86-architecture worker nodes with varying amounts of cores and memory. Some, or all, of the nodes may have accelerators in the form of NVIDIA GPUs or Intel Xeon Phi coprocessors. At a minimum, nodes are connected with Gigabit Ethernet (GbE), often supplemented by InfiniBand (IB). In addition, modern server nodes offer a form of Intelligent Platform Management Interface (IPMI)—an out-of-band network that can be used for rudimentary monitoring and control of compute node hardware status. Storage subsystems providing high-speed parallel access to data are also a part of many modern HPC clusters. These subsystems use the GbE or IB fabrics to provide compute nodes access to large amounts of storage.

On the software side, much of the cluster infrastructure is based on open-source software. In almost all HPC clusters, each worker node runs a separate copy of the Linux OS that provides services to the applications on the node. User applications employ message passing libraries (e.g., the Message Passing Interface, MPI) to collectively harness large numbers of x86 compute cores across many server nodes. Nodes that include coprocessors or accelerators often require user applications to use specialized software or programming methods to achieve high performance. An essential part of the software infrastructure is the workload scheduler (such as Slurm, Moab, Univa Grid Engine, Altair PBS Professional) that allows multiple users to share cluster resources according to scheduling policies that reflect the objectives of the business.
Deployment Methodologies

As clusters grew from tens to thousands of worker nodes, methods for efficiently deploying software emerged. Clearly, installing software by hand on even a small number of nodes is a tedious, error-prone, and time-consuming task. Methods using network and Pre-Execution Environment (PXE) tools were developed to provision worker node disk drives with the proper OS and software. Methods to send a software image to compute nodes over the network exist for both diskfull (resident OS disk on each node) and diskless servers. On diskfull hosts, the hard drive is provisioned with a prepackaged Linux kernel, utilities, and application libraries. On diskless nodes, which may still have disks for local data, the OS image is placed in memory (RAM-disk) so as to afford faster startup and reduce or eliminate the need for hard disks on the nodes. In either case, node images can be centrally managed and dispersed to the worker nodes. This eliminates the tendency for nodes to develop “personalities” due to administrative or user actions that alter nodes from their initial configuration.

Essential HPC Cluster Strategies

The following strategies provide background and recommendations for many of the current trends in HPC. There are many issues involved in successful HPC clustering and the following represent some of the important trends in the market.

Strategy 1: Plan To Manage the Impact of Software Complexity

HPC systems rely on large amounts of complex software, much of which is freely available. There is an assumption that because the software is “freely available,” there are no associated costs. This is a dangerous assumption. There are real configuration, administration, and maintenance costs associated with any type of software (open or closed). Freely available software does not eliminate these costs. Indeed, it may even increase them by requiring administrators to learn and master new packages and skills. Such a strategy assumes the administrator has picked the best package with an active development community behind it.

As previously mentioned, quick custom deployments may demonstrate an initial victory, but they are vulnerable to “breakage” when change is needed. They tend to be very “guru” dependent, requiring that the person who set-up the system maintain it. Replacing gurus can be very expensive and introduce long downtimes while the entire system is reverse engineered and re-architected by yet another guru.

Another cause of complexity is the lack of integration between software tools and utilities. There are many freely available tools for managing certain aspects of HPC systems, and these include packages like Ganglia (monitoring), Nagios (alerts), and IMPItool (out-of-band management). Each of these requires separate management access through a different interface (GUI or command line). Network switches are also managed through their own administrative interface.

A key ingredient to standing up your own cluster is an expert administrator. These administrators often have a specialized skill set that includes shell scripting, networking, package management and building, user management, node provisioning, testing, monitoring, network booting, kernel-module management, etc.

There are some freely available tools for managing cluster software complexity. These include projects such as Rocks, Waresulf, and oneSIS. While these help administrators manage cluster software deployment, they do not address tool integration. That is, they help manage the tools mentioned above, but do nothing to provide the administrator with a unified view of the cluster. If there are changes beyond the standard recipes offered by these packages, skilled administrators are often needed to fine-tune the cluster. This can involve scripts that make use of syntax and semantics specific to the management software itself. An example is configuration and operation of workload schedulers.

One of the best packages for managing software complexity is Bright Cluster Manager. This professional package allows complete control of the cluster software environment from a single point-and-click interface that does not require advanced systems-administration skills. Some of the provisioning features include the ability to install individual nodes or complete clusters from bare metal
within minutes. Administrators can create and manage multiple (different) node images that can be assigned to specific nodes or groups of nodes. Once the node images are in place, they can be changed or updated directly from the head node without the need to login/reboot the nodes. Packages on node images can be added or removed using standard RPM tools or YUM. Changes can be easily tracked and old node images restored to any node. In addition, node images can be configured as either diskless or diskfull with the option to configure RAID or Logical Volume Management (LVM). Bright Cluster Manager is an all-encompassing solution that also integrates monitoring and management of the cluster — including the selection and full configuration of available workload schedulers. As shown in Figure 1, complex image management scenarios can be addressed through use of the Bright GUI; a similar capability is available from Bright’s command line interface (not shown).

Recommendations for Managing Software Complexity

- Unless your goal is to learn about HPC cluster design, avoid creating your own cluster from scratch. If you choose this route, be sure your administrators have the required skill set. Also, document all changes and configuration steps for every package so other administrators can work with what you have created. This method is the most time consuming path to creating a cluster, and least likely to deliver against the success factors identified previously.
- Consider using a cluster management toolkit such as Rocks, Warewulf, or oneSIS. Keep in mind these toolkits help manage the standard packages mentioned above, but do not provide integration. The skill set for using these systems, particularly if you want to make configuration changes, is similar to creating a cluster from scratch. These tools help reduce the work needed for cluster set-up, but still require time to “get it right.”
- A fully integrated system like Bright Cluster Manager provides a solution that does not require advanced administrator skills or large amounts of time to set up a cluster. It helps eliminate the extra management costs associated with freely available software and virtually eliminates the need for expensive administrators or cluster gurus. This is the fastest way to stand up an HPC cluster and start doing production work.

![Figure 1: Bright Cluster Manager manages multiple software images simultaneously. In this screenshot, modifications to the default-image are about to be created and then registered for the purpose of revision control.](image-url)
Strategy 2: Plan for Scalable Growth

Almost all HPC clusters grow in size and capability after installation. Indeed, most clusters see several stages of growth beyond their original design. Expanding a cluster almost always means new or next-generation processors and/or coprocessors plus accelerators. The original cluster configuration often will not work with new hardware due to the need for updated drivers and OS kernels. Thus, there needs to be some way to manage growth.

In addition to capability, the management scale of clusters often changes when they expand. Many times, home-brew clusters can be victims of their own success. After setting up a successful cluster, users often want more node capacity. The assumption that “more is better” makes sense from an HPC perspective, however, most standard Linux tools are not designed to scale their operation over large numbers of systems. In addition, many custom scripts that previously worked on a handful of nodes may break when the cluster grows. Unfortunately, this lesson is often learned after the new hardware is installed and placed in service. Essentially, the system becomes unmanageable at scale — and may require complete redesign of custom administration tools.

Smaller clusters often overload a single server with multiple services such as file, resource scheduling, plus monitoring/management. While this approach may work for systems with fewer than 100 nodes, these services can overload the cluster network or the single server as the cluster grows.

A fully integrated system like Bright Cluster Manager offers a tested and scalable approach for any size cluster. Designed to scale to thousands of nodes, Bright Cluster Manager is not dependent on third-party (open source) software that may not support this level of scalability.

Smaller clusters often overload a single server with multiple services such as file, resource scheduling, plus monitoring/management. While this approach may work for systems with fewer than 100 nodes, these services can overload the cluster network or the single server as the cluster grows.
Recommendations for Managing Cluster Growth

- Assume the cluster will grow in the future and make sure you can accommodate both scalable and non-homogeneous growth. [i.e. expansion with different hardware than the original cluster (see below)].
- Though difficult in practice, attempt to test any homegrown tools and scripts for scalability. Many Linux tools were not designed to scale to hundreds of nodes. Expanding your cluster may push these tools past their usable limits.
- In addition, test all open source tools prior to deployment. Remember proper configuration may affect scalability as the cluster grows.
- Make sure you have a scalable way to push package and OS upgrades to nodes without having to rely on system-wide reboots. Make sure the “push” method is based on a scalable tool.
- A proven and unified approach, using a solution such as Bright Cluster Manager, offers a scalable, low overhead method for managing clusters. Changes in the type or scale of hardware will not impact your ability to administer the cluster. Removing the dependence on homegrown tools or hand configured open source tools allows system administrators to focus on higher-level tasks.

Figure 2: Bright Cluster Manager manages heterogeneous-architecture systems and clusters through a single GUI. In this screenshot, a Cisco UCS rack server provides physical resources for a Hadoop HDFS instance as well as for HPC.
Strategy 3: Plan to Manage Heterogeneous Hardware/Software Solutions

As mentioned, almost all clusters expand after their initial installation. One of the reasons for this longevity of the original hardware is the fact that processor clock speeds are not increasing as quickly as they had in the past. Instead of faster clocks, processors now have multiple cores. Thus, many older nodes still deliver acceptable performance (from a CPU clock perspective) and can provide useful work. Newer nodes often have more cores per processor and more memory. In addition, many clusters now employ accelerators from NVIDIA or AMD or coprocessors from Intel on some or all nodes. There may also be different networks connecting different types of nodes (either 1/10 GbE or IB).

Another cluster trend is the use of memory aggregation tools such as ScaleMP. These tools allow multiple cluster nodes to be combined into a large virtual Symmetric Multiprocessing (SMP) node. To the end user, the combined nodes look like a large SMP node. Oftentimes, these aggregated systems are allocated on traditional clusters for a period of time and released when the user has completed their work, returning the nodes to the general node pool of the workload scheduler.

The combined spectrum of heterogeneous hardware and software often requires considerable expertise for successful operation. Many of the standard open source tools do not make provisions for managing or monitoring these heterogeneous systems. Vendors often provide libraries and monitoring tools that work with their specific accelerator platform, but make no effort to integrate these tools into any of the open source tools. To monitor these new components, the administrator must “glue” the vendor tools into whatever monitoring system they employ for the cluster, typically this means Ganglia and Nagios. Although these tools provide hooks for custom tools, it is still incumbent upon the administrator to create, test and maintain these tools for their environment.

Managing virtualized environments, like ScaleMP, can be similarly cumbersome to administrators. Nodes will need the correct drivers and configuration before users can take advantage of the large virtual SMP capabilities.

Heterogeneity may extend beyond a single cluster because many organizations have multiple clusters, each with their own management environment. There may be some overlap of tools, however older systems may be running older software and tools for compatibility reasons. This situation often creates “custom administrators” that may have a steep learning curve when trying to assist or take over a new cluster. Each cluster may have its own management style that further taxes administrators.

Ideally, cluster administrators would like the ability to have a “plug-and-play” environment for new cluster hardware and software eliminating the need to weave new hardware into an existing monitoring system or automate node aggregation tools with scripts. Bright Cluster Manager is the only solution that offers this capability for clusters. All common cluster hardware is supported by a single, highly efficient monitoring daemon. Data collected by the node daemons are captured in a single database, and summarized according to the site’s preferences. There is no need to customize an existing monitoring framework. New hardware and even new virtual ScaleMP nodes automatically show up in the familiar management interface.

As shown in Figure 3, Bright Cluster Manager fully integrates NVIDIA GPUs and provides alerts and actions for metrics such as GPU temperatures, GPU exclusivity modes, GPU fan speeds, system fan speeds, PSU voltages and currents, system LED states, and GPU ECC statistics (Fermi GPUs only). Similarly, Bright Cluster Manager includes everything needed to enable Intel Xeon Phi coprocessors in a cluster using easy-to-install packages that provide the necessary drivers, SDK, flash utilities, and runtime environment. Essential metrics are also provided as part of the monitoring interface.

Multiple heterogeneous clusters are also supported as part of Bright Cluster Manager. There is no need to learn (or develop) a separate set of tools for each cluster in your organization.

Bright Computing’s integration with ScaleMP’s vSMP Foundation means that creating and dismantling a virtual SMP node can be achieved with just a few clicks in the cluster management GUI. Virtual SMP nodes can even be built and launched automatically on-the-fly using the scripting capabilities of Bright Cluster Manager.
Five Essential Strategies for Successful HPC Clusters

of the cluster management shell. In a Bright cluster, virtual SMP nodes can be provisioned, monitored, used in the workload management system, and have health checks running on them — just like any other node in the cluster.

Recommendations for Managing Heterogeneity

- Make sure you can monitor all hardware in the cluster. Without the ability to monitor accelerators or coprocessors, an unstable environment can be created for users that is difficult for administrators to manage.

- Heterogeneous hardware is now present in all clusters. Flexible software provisioning and monitoring capabilities of all components are essential. Try to avoid custom scripts that may only work for specific versions of hardware. Many open source tools can be customized for specific hardware scenarios thus minimizing the amount of custom work.

- Make sure your management environment is the same across all clusters in your organization. A heterogeneous management environment creates a dependency on specialized administrators and reduces the ability to efficiently manage all clusters in the same way. Any management solution should not interfere with HPC workloads.

- Consider a versatile and robust tool like Bright Cluster Manager with its single comprehensive GUI and CLI interface. All popular HPC cluster hardware and software technologies can be managed with the same administration skills. The capability to smoothly manage heterogeneous hardware and software is not available in any other management tool.

Figure 3: Bright Cluster Manager manages accelerators and coprocessors used in hybrid-architecture systems for HPC. In this screenshot, Bright allows for direct access to the NVIDIA GPU Boost technology by setting GPU clock speeds.
Strategy 4: Be Ready for the Cloud

Cloud computing offers certain flexibility not normally found in fixed-size, on-premise HPC clusters. Users with existing clusters can elastically expand (then contract) their capacity without large capital costs or set-up time. Those without clusters now have the capability to quickly spin-up HPC resources in the Cloud. While there are many Cloud providers, Amazon Web Services (AWS) and some others offer true HPC clouds that can deliver the type of computing environment that high performance applications require.

Integrating a Cloud solution into an in-house cluster can be difficult because the Cloud only provides the raw machines to the end users. Similar to an on-site cluster, the Cloud cluster needs to be configured and provisioned before use. All of the issues mentioned previously, from software complexity to system growth and scalability, as well as heterogeneous environments, need to be addressed before any meaningful HPC Cloud work can begin. Indeed, flexibility and ease of management is now even more important in the Cloud due to the ephemeral nature of Cloud clusters, (i.e., administrators may need to repeatedly set up and tear down Cloud instances over time). Long lead times for custom cluster configuration defeats the advantages of an on-demand Cloud environment.

In addition, because Cloud use is metered, monitoring cloud activity is vitally important. Unmonitored Cloud use can result in Cloud budget overruns and added expense.

Figure 4: Bright Cluster Manager manages physical nodes on the ground and in the cloud. In this screenshot, managed HPC workloads are running on the ground and in the cloud via Bright and the PBS Professional workload manager.
From a user’s perspective, the Cloud cluster needs to look as much as possible, like the existing cluster so they are not burdened with learning new system rules. In a sense, the end user should not be able to tell the difference (other than perhaps data transfers) between Cloud and cluster nodes.

An integrated solution like Bright Cluster Manager, shown in Figure 4, can provide a managed and transparent Cloud-based HPC cluster. The same powerful cluster provisioning, monitoring, scheduling and management capabilities that Bright Cluster Manager provides to onsite clusters extend into the cloud, ensuring effective and efficient use of the virtual cloud resources. Using Bright Cluster Manager, you can extend into public clouds, such as AWS, with only a few mouse clicks. There’s no need for expert knowledge of Linux or Cloud computing. The exact same administration interface is used for both local and cluster nodes. Installation/initialization, provisioning, monitoring, scheduling and management are identical. In this scenario, there is no learning curve required for Cloud HPC use.

**Recommendations for Cloud HPC**

- Cloud HPC offers an elastic and flexible method to expand your HPC capabilities. Make sure you use Cloud services that are designed for HPC applications including high-bandwidth, low-latency networking, exclusive node use, and very high compute and storage capacities.
- Develop a very flexible and rapid provisioning scheme so that you can spin-up clusters in the Cloud quickly. This environment should mirror your local systems as much as possible and be tested before you allow users to make use of the Cloud. Improperly configured Cloud instances may cause user jobs to fail silently while the “Cloud meter” keeps running.
- Monitoring is essential. Keep a close eye on Cloud usage to avoid unexpected costs.
- Integration with existing workload schedulers is essential so that users are not burdened with new procedures or rules for using Cloud resources.
- Consider the capabilities of Bright Cluster Manager where your existing cluster can be seamlessly expanded into the Cloud. The new capacity is managed in the same way as your existing cluster and users can start using these new resources immediately with minimal set-up.

### Strategy 5: Have an answer for the Hadoop Question

While Hadoop is not strictly an HPC application, its use in HPC is increasingly forcing systems administrators to answer the question: “Can I run Hadoop or HBase on this cluster?” (HBase is the Hadoop version of Google’s BigTable database.)

Instead of configuring and managing a sub-cluster or specialized Hadoop system, an automated solution such as Bright Cluster Manager can effortlessly bring Hadoop capability to a cluster.

Managing a Hadoop cluster is different than managing an HPC cluster. It requires mastering some new concepts, but from a management perspective, the hardware is basically the same. In one sense, a Hadoop cluster is actually simpler than most HPC configurations. Hadoop clusters use Ethernet and generally just CPUs. Hadoop is an open-source project and is often configured “by hand” using various XML files. Beyond the core components, Hadoop offers a vast array of additional components that also need management and configuration. Oftentimes Hadoop is configured as a “sub-cluster within a cluster” where a collection of nodes is configured to run the various Hadoop services. Hadoop processing is different than the typical HPC cluster as it has its own scheduler (YARN) and file system (HDFS). Integration with an existing cluster can be somewhat tedious, as a completely different set of daemons must be started on Hadoop nodes.

Instead of configuring and managing a sub-cluster or specialized Hadoop system, an automated solution such as Bright Cluster Manager can effortlessly bring Hadoop capability to a cluster. The latest version of Bright Cluster Manager (version 7) supports the leading distributions of Apache Hadoop (e.g., from the Apache Foundation, Cloudera, Hortonworks), enabling Bright’s customers to choose the one that best fits their needs while taking advantage of Bright Cluster Manager’s advanced capabilities as illustrated in Figure 5(a).
Beyond installation and provisioning, Bright Cluster Manager provides comprehensive monitoring and management of the Hadoop cluster through the same graphical user interface used for HPC clusters. As shown in Figure 5(b), Systems administrators can manage a native Hadoop cluster or sub-cluster with the same efficiency as an HPC cluster.

**Recommendations for Hadoop Strategy**

- Plan on users requesting Hadoop or HBase capabilities in the near future. Hadoop continues to mature, and is now in use complementing HPC and other types of clusters.
- Hadoop Clusters are configured differently than HPC clusters. Consider creating Hadoop sub-clusters in larger HPC clusters, or a separate stand-alone Hadoop cluster. Keep in mind the Hadoop sub-cluster is restricted to doing only Hadoop processing using its own workload scheduler.
- Hadoop management is very different than HPC cluster management. There is a large number of “moving parts” (services) in a Hadoop cluster. Develop a method to easily deploy, start, stop, and manage a Hadoop cluster.
- To avoid costly delays and configuration headaches, consider the Hadoop management capabilities in Bright Cluster Manager. Its consistent and flexible provisioning and monitoring environment will benefit administrators when they introduce Hadoop capability to their users.
Conclusion

A successful HPC cluster requires administrators to provision, manage, and monitor an array of hardware and software components. Currently, there are many trends in HPC clustering that include software complexity, cluster growth and scalability, system heterogeneity, Cloud computing, as well as the introduction of Hadoop services. Without a cogent strategy to address these issues, system managers and administrators can expect less-than-ideal performance and utilization. There are many component tools and best practices to be found throughout the industry. Bright Cluster Manager is the only comprehensive solution that provides a clear management path to developing a successful HPC strategy.

To find out more about Bright Cluster Manager please visit http://www.brightcomputing.com.