Deploying Hortonworks Data Platform (HDP) on VMware vSphere®

A TECHNICAL REFERENCE ARCHITECTURE

APRIL 2016

vmware®

Hortonworks

Table of Contents

Overview

Hadoop users are deploying the Hortonworks Data Platform (HDP) on VMware vSphere® today for mission-critical applications in the large scale. With customer deployments exceeding 200 physical servers and more than 400 virtual machines for mission critical applications, HDP on vSphere is a serious business application environment for enterprises today.

This paper describes the technical reference architecture that VMware and Hortonworks together propose for a larger deployment of HDP on vSphere. The design principles given here are derived from example deployments that have undergone several months of heavy use and that have been proven to be a reliable and scalable combination. The information in this reference architecture refers to these example deployments to give a sense of the scale that is achievable using a virtualized Hadoop architecture. The architecture here makes use of Direct Attached Storage (DAS) for all of its data. Other storage architectures such as a configuration leveraging Network Attached Storage (NAS) are also applicable, but those are out of scope for this discussion.

Why Virtualize Hadoop on vSphere?

Organizations choose to use vSphere as the platform for hosting Hadoop-based applications for several reasons:

- To provide Hadoop clusters as a service quickly to the end users and the development community, reducing time to business insight into the data
- To quickly deploy, scale up, and remove clusters through automation
- To capitalize on the most suitable hardware and storage to make use of the Hadoop design ethos, thereby lowering costs to the end user
- To provide a multi-tenant platform with support for heterogeneous Hadoop distributions

NOTE: vSphere offers support for multi-tenancy through the security and performance isolation inherent in sets of virtual machines that exist in separate resource pools. vSphere also offers resource usage isolation by providing controls to mitigate contention for hardware resources.

- To optimize the use of existing VMware investments by extending Hadoop services to the current computing platform

Technical Architecture

The Virtual Machine and Host Server Architecture

Small and medium-sized Hadoop clusters are those that fit entirely within a group of hosts on one datacenter rack. These Hadoop clusters are naturally easier to set up and simpler to manage, although they may not present the same level of complexity and tolerance to failure as larger architectures do. The reference architecture described here is one for a multi-rack deployment. This means that considerations such as the following need to be addressed:

- the number of host servers contained on each rack
- the number of virtual machines per host server
- the number of racks and servers that make up a Hadoop cluster
- the number of host servers per vSphere cluster
- the network communication paths between the host servers
- the hardware failure conditions and the mechanisms for recovery from them

Each of these subjects are discussed in this document.

Racks, vSphere Server Hosts, and Virtual Machines

In the reference architecture that we provide here as an example implementation, there are 8 racks and 160 host servers (20 host servers per rack) deployed. This hardware can be used for one or more Hadoop clusters, if needed. There are two virtual machines on each vSphere host server for the Worker nodes – the decision to do this is entirely enterprise-specific and is determined by the degree of consolidation required. Two virtual machines per host is one of the simpler examples. There could well be more than two virtual machines per host server, with the right sizing factors applied to them. Having two virtual machines per host server reduces the number of virtual machines affected, should any host server fail. The initial Hadoop cluster size is made up of 320 virtual machines. The cluster sizing can be expanded to larger numbers over time to suit the business need.

The virtual machines on each host server are sized to occupy just under half of the full physical memory of each host server. A 6% portion of the physical memory of the host server is set aside for the vSphere ESXi hypervisor's use. The remaining memory is split evenly between the two resident virtual machines on the host servers. This same principle of evenly splitting the available memory and CPUs between virtual machines would apply if there were four virtual machines or more on each host server. The main driving force behind this choice is the neat fit of the virtual machine's configured memory within that of a non-uniform memory access (NUMA) node (with one, two or more virtual machines fitting into each NUMA node) and the correct fit of virtual CPUs within a socket's core count. With Hadoop workloads, we avoid the overcommitment of physical memory so as to place no constraints on the Java processes that make up the system. The physical racks and host servers for one part of the architecture are shown in Figure 1. One design for the vCenter cluster layout is also given here.

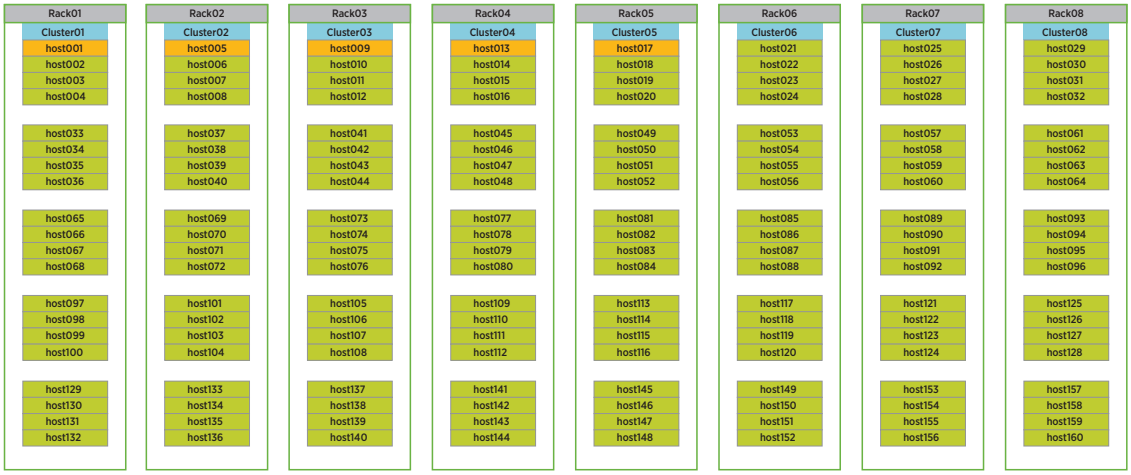

Figure 1: The rack and host server layout with single per-rack vSphere clusters and host servers containing master processes shown in orange color

In Figure 1, the eight initial racks are shown numbered across the top row and the host servers are labeled "host<nnn>". Each rack contains 20 host servers by way of illustration. Each vSphere cluster encompasses one rack's collection of host servers in this design. This is a straightforward approach to vSphere cluster design. Five of the host servers, shown in orange, are set aside for hosting the virtual machines that contain the Master services in Hadoop (such as the ResourceManager, NameNode, and other master processes). Those particular five host servers also host a virtual machine containing a number of Hadoop Client processes. The remaining set of host servers, shown in green, is allocated for hosting the Hadoop Worker processes (DataNode, NodeManager and the containers that execute parts of each compute job). Since the architecture uses DAS on each host server, there are Hadoop Distributed File System (HDFS) data blocks managed by the DataNode process on each host server that has a Worker configuration.

Figure 2 shows another variant of the architecture with the same number of servers and rack layout, but this time using vSphere clusters that cross the physical racks, as an alternative approach. This more advanced architecture is in use at certain enterprise sites today and is a standard approach at some, for fault tolerance reasons at the rack level.

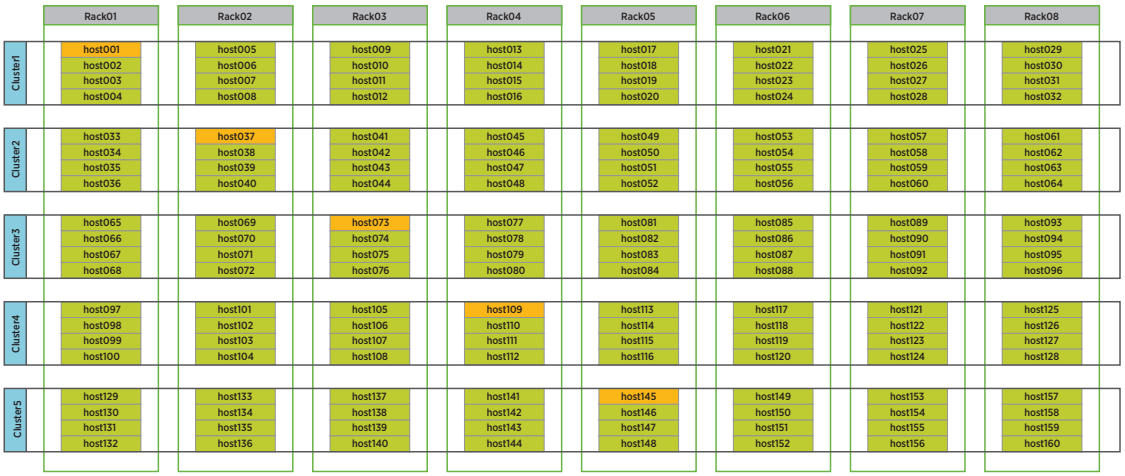

Figure 2: The rack and host server layout with cross-rack clusters and host servers containing master processes shown in orange color

The vSphere clusters are spread across all of the racks. There are 32 host servers in each vSphere cluster, as prescribed by vSphere itself. (This number is 64 host servers per cluster for vSphere 6 and later.) The spreading of vSphere clusters across the racks is done in order to ensure that any single rack failure cannot take out an entire vSphere cluster.

The Master virtual machines are shown in Figure 3, with one Master virtual machine residing on each of the five host servers mentioned above. The host servers here are the outer boxes labeled "host<nnn>" and the virtual machines are named using the convention "mst<nn>". Various Hadoop master roles are spread out across the five master virtual machines. The determination of master role placement onto the virtual machines is done by the Hadoop architect with advice from the Hadoop distro vendor. The placement of Hadoop roles may differ for your particular requirements. The layout given is shown in Figure 3 is one example deployment.

Note the the main **NameNode** and the *Standby NameNode process*, used for HDFS High Availability, are hosted on different servers and on different racks. High Availability for NameNode processes is done in Hadoop using a pair of processes that are in Active/Passive configuration, with a warm standby. All metadata changes that occur on the HDFS filesystem are captured in a separate edit log file that is written to by the Active NameNode and read by the Passive NameNode. The passive NameNode applies the same changes to its own copy of the metadata,

The same principle applies to other primary and secondary pairs of processes. If a full rack were to fail due to a power loss to that rack, then based on this design, the survivor process would live on to continue the work of the cluster. This is an important consideration for production Hadoop clusters.

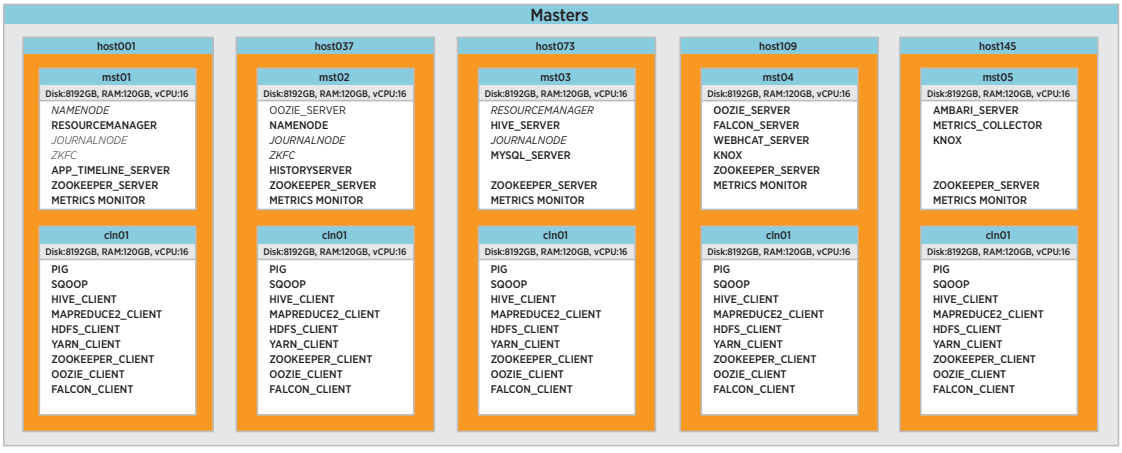

Figure 3: The five Master virtual machines (mst), mapped into five host servers (host<n>) with client (cln) virtual machines resident alongside them on the host servers

The Master node virtual machines are configured differently from the Worker nodes. This is based on the Hadoop master services that are used. Different combinations of the Hadoop master services can be seen in Figure 3 operating within the five Master virtual machine layouts. Since Master nodes don't do direct processing of the Hadoop jobs, the extra memory on the host servers can be used to add Edge/Client nodes in separate "Client" virtual machines. The Edge/Client nodes are the points where users have SSH access to the client libraries for common services like Pig, Hive and Sqoop. By co-locating the Edge/Client nodes with the Master Hadoop Nodes in separate virtual machines on the same host servers, the architecture allows for future growth of the master service processes. For instance, if HDFS were to grow to be large enough, it may become necessary to increase the heap size for the NameNode process.

In the reference architecture, the Client processes may be mixed in with the Master processes, as they are running on separate virtual machines on the same host servers. Virtual machine isolation ensures that the Client virtual machines do not impact the Master virtual machines' performance or data handling in any way. This is a key point to pay careful attention to in your own design. If the Client processes are not considered to incur heavy processing loads in your own workloads and would not interfere with the operation of the Master processes, then they may be considered for virtual machines that share the same physical hardware. Client virtual machines may also be separated out onto separate host servers from those that host the Master processes, if required. Additional edge nodes in new virtual machines can be added that do not share host servers with the master virtual machines.

The layout of the Worker virtual machines, with two such virtual machines shown per host server, is given in Figure 4. Each worker virtual machine is named in "wrk<nn>" format. The example virtual disk size, RAM size and number of virtual CPUs for each virtual machine is given in the title bar for each virtual machine. These are configurable for different architectures and hardware setups. The choices made here (8,192GB disk space, 120GB of RAM and 16 virtual CPUs per virtual machine) are designed to allocate the available physical disks, RAM and CPUs of the host servers evenly across the two resident Worker virtual machines. These sizings and numbers of Worker virtual machines will differ according to your particular circumstances – they serve here as an example and are not set in stone, by any means. You may choose to have four or more Worker virtual machines per host server, for example.

Figure 4: The Worker virtual machines (wrk01-02) and their resident Hadoop processes with two virtual machines per host server.

The host servers on which these virtual machines execute have 256GB of physical memory and two CPU sockets of 8 cores each. There are therefore 32 logical cores or hyperthreads on each host server. This allows for 16 virtual CPUs to be used in each virtual machine, thereby exactly committing the number of logical cores (hyperthreads) to virtual CPUs. The server hardware specifications may well differ in your case and the virtual machine layout will differ accordingly. The configurations above are given as an example. Refer to the best practices given in the technical papers on the VMware website at http://www.vmware.com/bde for more details on this subject.

Some organizations prefer to take a more conservative approach and make the total number of virtual CPUs in all virtual machines running on one host server equal to the total physical core count on the host server instead. This is a performance trade-off that is decided on the basis of the application requirements – the decision is taken between the application architect, the Hadoop implementation team and the virtualization administrator. Many different combinations are available – we have chosen one here from our experience to serve as an example.

Disk Allocation

The local DAS disks are arranged in the reference architecture so that each set of six physical disks is allocated to one of the virtual machines through the virtualization mechanisms. There could be more or fewer disks allocated per virtual machine, but a typical recommendation is a minimum of six. This arrangement is shown in Figure 5. Each physical disk spindle is setup in the vCenter tool to be a vSphere datastore of its own. A vSphere datastore provides an abstraction away from the physical hardware. One or more vSphere datastores can be associated together to become a datastore at a higher level. Those datastores are then used to hold the virtual disk or VMDK files that make up the virtual machine. One simple and recommended way to do this is to map one vSphere datastore to one disk spindle/device and then place one VMDK file in that datastore.

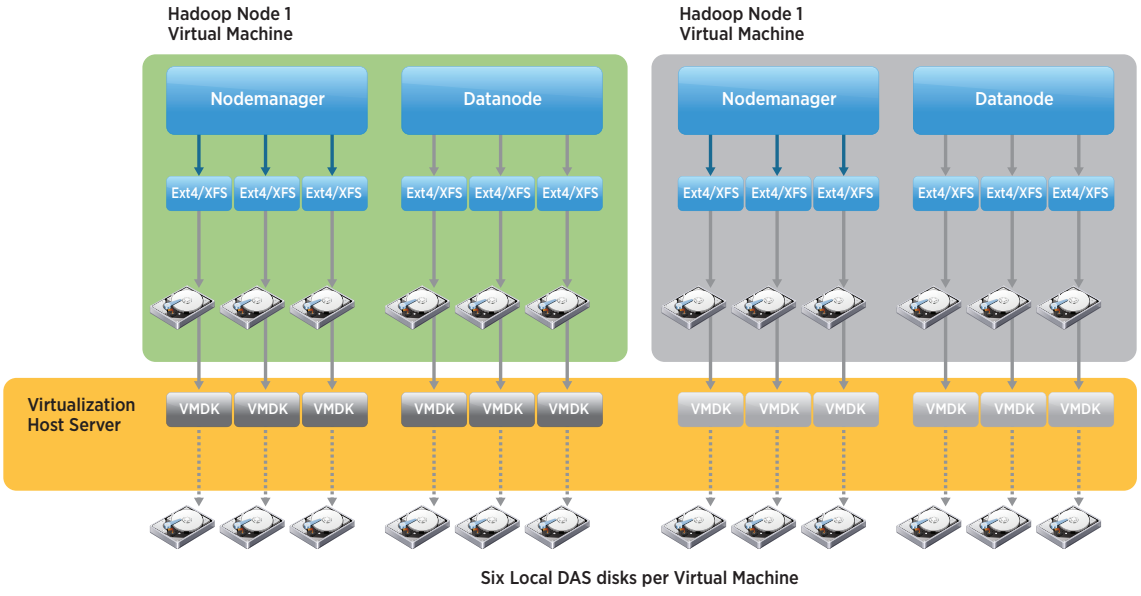

Figure 5: Virtual machines with Hadoop processes mapped to VMDKs and physical DAS disks

Each virtual machine has disk partitions in its own guest operating system. Those partitions map to guest OS directories as the applications see them. Those directories then represent the destinations for HDFS data reads/writes in the case of the DataNode process. Other directories are the destinations for reads/writes to the temporary (shuffle/spill) data in the case of the NodeManager along with its running containers. These data access types are isolated from each other as far as the different types of traffic (HDFS/temporary) are concerned. This is also true for each virtual machine's access to its own set of disks as shown in Figure 5. This isolation of I/O bandwidth consumption between the processes and virtual machines provides the minimal interference and thus the best opportunity for performance isolation.

In an example Hadoop deployment at a large site, there are 18 physical disks per host server. For those host servers that contain Worker virtual machines (i.e., those running the Hadoop DataNode and NodeManager processes) eight physical disks are allocated to each virtual machine. Each Worker virtual machine is one of a pair that is placed on each host server, so the pair of virtual machines take up 16 of the disks for Hadoop purposes. The two remaining disks are set aside for guest operating system data within the virtual machines.

At the VMware vCenter level, a vSphere datastore may be created for each individual disk spindle. Subsequently, these low-level datastores may be aggregated together or used individually to contain one or more Virtual Machine Disk (VMDK) files.

In the deployed Hadoop cluster, the DAS disks on the worker servers are a mixture of different capacity sizes from 1.2TB to 2TB.

Networking Architecture

Networking is an area on which a lot of attention is focused in deploying Hadoop both on native and on virtualized systems. The following is a real-world network architecture that is deployed in the reference architecture today. It is not the only one that could be applied, but we believe it is representative of many enterprise network architectures.

A leaf-and-spine layout for the networking is employed in the architecture, as shown in Figure 6. The leaf switches are capable of 10Gb transfer speeds whereas the spine switches are capable of supporting a 40Gb layer 3 mesh connection. Each host server contained in the racks has two 10GbE ports. Within a host server, the ports are connected to two separate leaf switches. Each rack contains 4 host servers that belong to one vSphere cluster (shown in Figure 1). The host servers within that one vSphere cluster are all connected to the same pair of leaf switches, so there is a single layer 2 subnet for all 32 hosts within one vSphere cluster. This means that the layer 2 networking traffic between servers is handled by the two leaf switches to which all hosts are connected in that one vSphere cluster. Layer 3 network traffic, going from one vSphere cluster to another, requires use of the spine switches and that traffic then traverses one of the pair of leaf switches that support the second vSphere cluster's host servers.

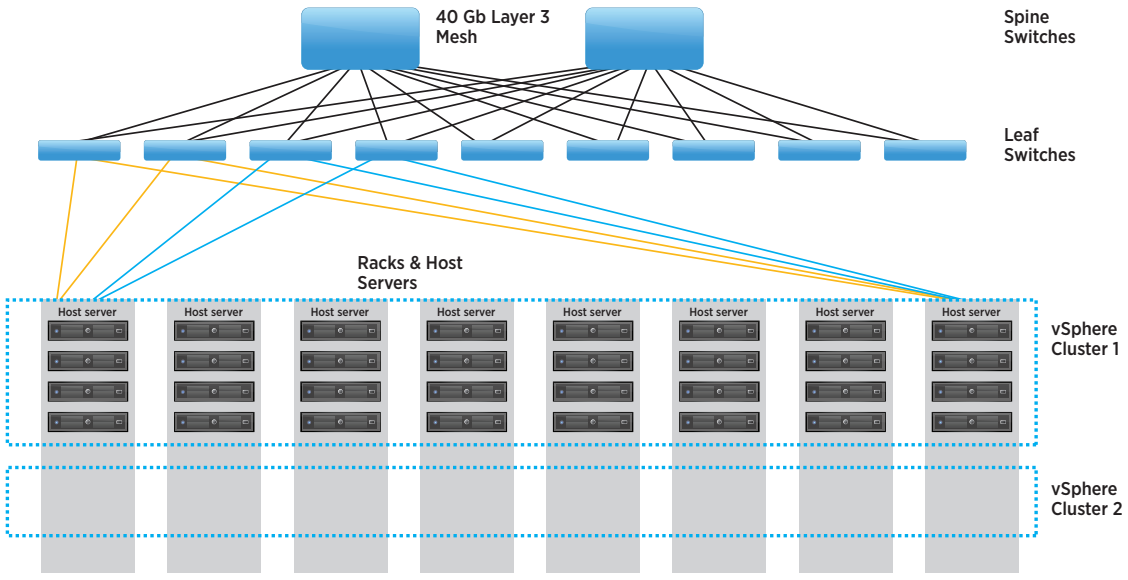

Figure 6: An Example Network Configuration

Designers of a Hadoop cluster need to design for networking many physical host servers and virtual machines together. They need to make a set of decisions about network addressing, such as choice of subnets and allocation of IP addresses.

Leaf and Spine networking is used in this reference architecture setup as it is commonly found at modern organizations that use Hadoop. At the time the initial reference implementation setup was done, it was created without using VMware's NSX™ for networking. We recommend that NSX with virtual networks be used to provide a flat address space and hide the complexities of the underlying physical networks. VMware NSX is in use at many sites and has been proven to be useful in providing, among other functionality, a flat network address space (essential for the large numbers of IP addresses that are required). There are some hurdles to overcome when NSX is not in place in the large reference architecture, due to separate VLANs and DHCP Servers being used for each vSphere cluster.

Using NSX for network virtualization, virtual networks are isolated from any other virtual network and from the underlying physical network by default, delivering the security principle of least privilege. Virtual networks are created in isolation and remain isolated unless specifically connected together. No physical subnets, no VLANs, no access control lists (ACLs), no firewall rules are required to enable this isolation. Any isolated virtual network can be made up of workloads distributed anywhere in the data center.

Subnets

If more than 255 IP addresses are required in one subnet (as is required in the reference architecture layout) then a standard Class C network with one subnet is not sufficient. Without NSX present in the architecture multiple subnets are required to be deployed with each vSphere cluster causing further complexity of systems administration.

IP Address Allocation

The networking team makes a decision between using static IP addresses for each virtual machine or using a DHCP method for assignment. In general, using static allocation of IP addresses to the individual nodes/virtual machines is preferred. However, DHCP allocation can also work, though it may involve more complexity of leasing IP addresses to machines that may not be active for long periods of time. Some sites choose to use ranges of static IP addresses for all of the virtual machines that are created. That is an acceptable approach as an alternative to the DHCP one.

If DHCP is employed, then the lease times on IP addresses require careful consideration. In the reference architecture, DHCP is configured so that the lease on an IP address is released after a virtual machine does not renew its lease within one week. This is useful for situations where virtual machines are shut down for over one week. The virtual machine will in that case no longer get the same IP address if it is brought back to life after that time. For the above reasons, the various complexities should be considered when choosing DHCP.

The infrastructure provides shared DNS and NTP services to support the deployment of multiple clusters. Forward and reverse lookup in DNS is particularly important for Hadoop clusters, so it is a good idea to ensure that functionality is available from the beginning.

Using vSphere Big Data Extensions

This section gives outline information on the use of the VMware vSphere Big Data Extensions (BDE) tool that is used to provide the base infrastructure on which the Hadoop software runs. BDE creates and configures virtual machines, configures their operating systems and then places the virtual machines onto the optimal vSphere host servers, according to a set of best practices. One example of these best practices is the tool ensures that physical memory is not overcommitted on the server host by the set of virtual machines it contains.

The BDE tool works with the vCenter server to clone a new set of virtual machines at different user-driven sizes. The system administrator, working with the big data application architect, can then provision the Hadoop cluster onto those virtual machines using a tool of their choice, such as Ambari.

vSphere BDE consists of a management server and a template virtual machine that are deployed as one unit, or virtual appliance, into the vCenter environment. After the vSphere Big Data Extensions VMware vSphere vApp™ is setup and configured with the available storage and network resources, it can then be used through a graphical user interface (GUI) as part of the vSphere Web Client, or through its own command-line interface (CLI). Figure 7 shows an outline of the vSphere Big Data Extensions architecture, with cloning of the pre-shipped template virtual machine, to form the basis for a new Hadoop cluster. The template virtual machine has the CentOS guest operating system as a default. This can be changed by the administrator, however.

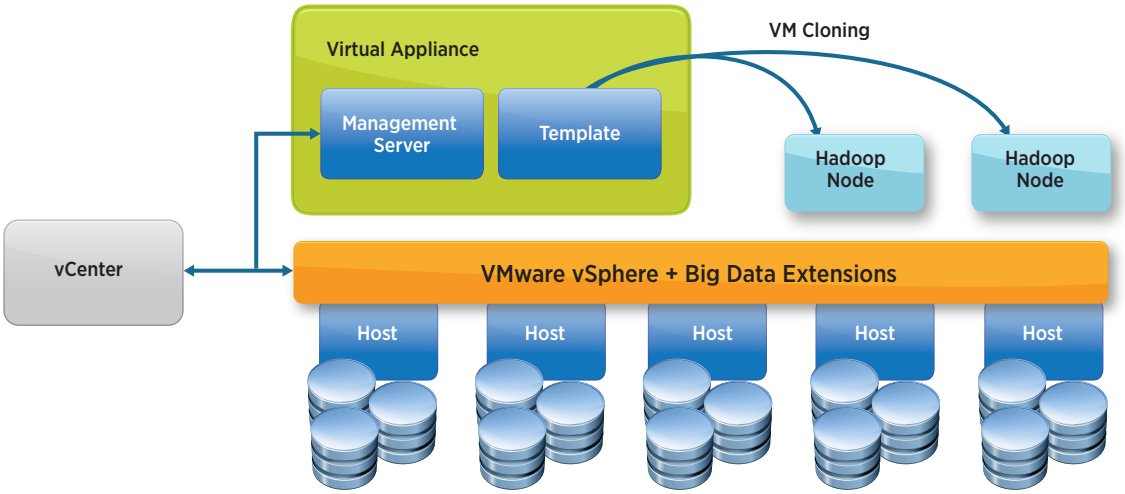

Figure 7: Architecture of vSphere Big Data Extensions

vSphere Big Data Extensions enables users or administrators to customize the various virtual machines that they create, either interactively through a web GUI or by means of using the Command-Line Interface (CLI) tool with a cluster configuration specification file. There are a number of samples of these JSON-based cluster specification files available in the "samples" directory on the vSphere Big Data Extensions management server.

Deployment Topology Types

The vSphere platform allows various topologies to be used for Hadoop cluster deployment. One such topology is called the Hadoop Virtual Extensions (HVE). A full description of all the features of HVE is given in [2]. Other topologies supported are named "host-as-rack" and "rack-as-rack". Here, we concentrate on the HVE topology as it is more suited to larger, DAS-based deployments.

VMware has contributed a set of technical features to the Apache community at the source code level that optimize Hadoop deployments on virtual platforms. Where the original Apache Hadoop implementation was aware of physical racks and host servers, HVE goes further and allows Hadoop distributions to be virtualization-aware. This enhances the support for failure and data locality in a virtual environment.

One such feature is concerned with HDFS data block replication, which is particularly applicable in a DAS environment. If there were three virtual machines, that replicate a certain HDFS data block, running on one host server and that host server were to fail, then that data block would be compromised. Through additions to the Hadoop topology at the source code level, HVE incorporates the concept of a Node Group to prevent such occurrences. Any virtual machines that belong to the same Node Group are said to be on the same physical host server. All the user needs to do to activate this is to specify the HVE type of topology at Hadoop cluster creation time, having uploaded a file containing the mapping of physical host servers to racks.

A description of the HVE functionality is given in JIRA HADOOP-8468. Hortonworks HDP distributions support HVE for this HDFS functionality today. The HVE data block handling is tested in a variety of real-world implementations of HDP on vSphere and has been proven to work very well.

Host Server Maintenance

Systems managers need to perform maintenance tasks on the infrastructure over time. One virtualized Hadoop implementation team solved this issue by writing code that orchestrates the rebooting of its servers to automate this process. This custom environment works by conducting the following operations:

1. Connects to the VMware vCenter Server™ and queries the ESXi server host that the administrator wants to perform maintenance on, to retrieve a list of all the current virtual machines on that host
2. Migrates all virtual machines that are not running the DataNode or ZooKeeper daemons to another host in the cluster. The master virtual machines's data in this case is held on Virtual SAN (VSAN) storage, allowing vMotion, while the worker virtual machines' data is on JBOD style storage.
3. Determines which customer clusters are affected if the DataNode and ZooKeeper virtual machines are shut down
4. Retrieves connection details for the Ambari managers for the affected clusters from the configuration management database (CMDB)
5. Connects to each application manager and instructs Ambari to put the virtual machine into Ambari's maintenance mode so the customer can see that work is being performed on their system

   Entering a host into Ambari maintenance mode achieves two things:

   a. Suppresses any alerts from the monitoring system (deployed in the provisioned virtual machines)

   b. Graphically represents the fact that an administrator is doing planned work on the server, so users do not need to worry
6. Powers off the virtual machine and enters the ESXi host into maintenance mode
7. After the maintenance work is completed, takes the ESXi host out of maintenance mode and powers on the virtual machines associated with the host
8. Reconnects to each Ambari server to start all the roles on those virtual machines
9. Finally, after the services have started, takes the Hadoop environment within the virtual machines out of maintenance mode in Ambari

Conclusion

Various enterprise deployments of HDP on vSphere have shown that the Hadoop ecosystem performs on the vSphere platform at a comparable level to native and has distinct added benefits to offer. Providing Hadoop cluster creation and management on VMware vSphere has greatly improved the way that big data applications are delivered on the Hadoop base. Developers and other staff can now obtain a Hadoop cluster much more rapidly for testing or development without concerning themselves with the underlying hardware. Following the cluster creation phase, they can then expand and contract their Hadoop clusters at a much more rapid pace than would have been available to them with a native implementation. Different teams can also share the same hardware, given available capacity, for different Hadoop cluster uses, with different versions available at one time.

Virtualizing Hadoop workloads, based on HDP and vSphere has become the standard practice for many enterprises' big data business.

References

1. VMware vSphere Big Data Site
   http://www.vmware.com/bde
2. The Hadoop Virtualization Extensions – White Paper
   http://www.vmware.com/files/pdf/Hadoop-Virtualization-Extensions-on-VMware-vSphere-5.pdf
3. Skyscape Cloud Services (UK Ltd.) Case Study White Paper
   http://www.vmware.com/files/pdf/products/vsphere/VMware-vSphere-Skyscape-Cloud-Services-Deploys-Hadoop-Cloud.pdf
4. "Virtualizing Hadoop", a book by Trujillo, G., et al., published in 2015 by VMware Press

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com