O'REILLY®

Hadoop Virtualization

Courtney Webster

VMware vSphere Big Data Extensions

Operational Simplicity with Performance

Optimal Resource Utilization

Robust Scalable Platform

www.vmware.com/bde

vmware

Hadoop Virtualization

Courtney Webster

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo O'REILLY®

Hadoop Virtualization

by Courtney Webster

Copyright © 2015 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editors: Julie Steele and Jenn Webb **Illustrator:** Rebecca Demarest

February 2015: First Edition

Revision History for the First Edition:

2015-01-26: First release

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Hadoop Virtualization*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights

ISBN: 978-1-491-90674-3

[LSI]

Table of Contents

The Benefits of Deploying Hadoop in a Private Cloud

Abstract

Hadoop is a popular framework used for nimble, cost-effective analysis of unstructured data. The global Hadoop market, valued at $1.5 billion in 2012, is estimated to reach $50 billion by 2020.[1] Companies can now choose to deploy a Hadoop cluster in a physical server environment, a private cloud environment, or in the public cloud. We have yet to see which deployment model will predominate during this growth period; however, the security and granular control offered by private clouds may lead this model to dominate for medium to large enterprises. When compared to other deployment models, a private cloud Hadoop cluster offers unique benefits:

- A cluster can be set up in minutes
- It can flexibly use a variety of hardware (DAS, SAN, NAS)
- It is cost effective (lower capital expenses than physical deployment and lower operating expenses than public cloud deployment)
- Streamlined management tools lower the complexity of initial configuration and maintenance
- High availability and fault tolerance increase uptime

This report reviews the benefits of running Hadoop on a virtualized or aggregated (container-based) private cloud and provides an overview of best practices to maximize performance.

Introduction

Today, we are capable of collecting more data (and various forms of data) than ever before.[2] It may be the most valuable intangible asset of our time. The sheer volume ("big data") and need for flexible, low-latency analysis can overwhelm traditional management systems like structured relational databases. As a result, new tools have emerged to store and mine large collections of unstructured data.

MapReduce

In 2004, Google engineers described a scalable programming model for processing large, distributed datasets.[3] This model, MapReduce, abstracts computation away from more complicated tasks like data distribution, failure handling, and parallelization. Developers specify a processing ("map") function that behaves as an independent, modular operation on blocks of local data. The resulting analyses can then be consolidated (or "reduced") to provide an aggregate result. This model of local computation is particularly useful for big data, where the transfer time required to move the data to a centralized computing module is limiting.

Hadoop

Doug Cutting and others at Yahoo! combined the computational power of MapReduce with a distributed filesystem prototyped by Google in 2003.[4] This evolved into Hadoop—an open source system made of MapReduce and the Hadoop Distributed File System (HDFS). HDFS makes several replica copies of the data blocks for resilience against server failure and is best used on high I/O bandwidth storage devices. In Hadoop 1.0, two master roles (the JobTracker and the Namenode) direct MapReduce and HDFS, respectively.

Hadoop was originally built to use local data storage on a dedicated group of commodity hardware. In a Hadoop cluster, each server is considered a node. A "master" node stores either the JobTracker of MapReduce or the Namenode of HDFS (although in a small cluster as shown in Figure 1, one master node could store both). The remaining servers ("worker" nodes) store blocks of data and run local computation on that data.

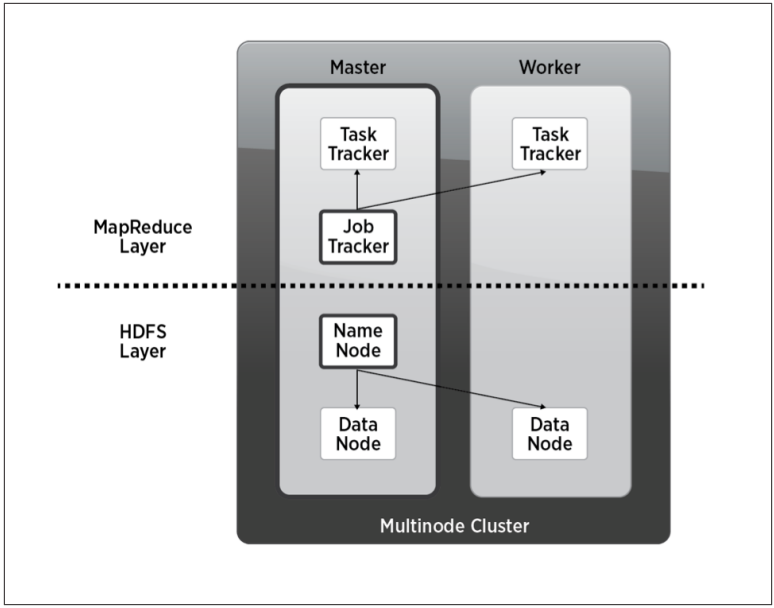

Figure 1. A simplified overview of Hadoop[5]

The JobTracker directs low-latency, high-bandwidth computational jobs (TaskTrackers) on local data. The Namenode, the lead storage directory of HDFS, provides rack awareness: the system's knowledge of where files (Data Nodes) are stored among the array of workers. It does this by mapping HDFS file names to their constituent data blocks, and then further maps those data blocks to Data Node processes. This knowledge is responsible for HDFS's reliability, as it ensures non-redundant locations of data replicates.

Hadoop 2.0

In the newest version of Hadoop, the JobTracker is no longer solely reponsible for managing the MapReduce programming framework. The JobTracker function is distributed, among others, to a new Hadoop component called the Application Master. In order to run tasks, ApplicationMasters request resources from a central scheduler called the ResourceManager. This architectural redesign improves scalability and efficiency, bypassing some of the limitations in Hadoop 1.0. A new central scheduler, the ResourceManager, acts as its key replacement. Developers can then construct ApplicationMasters to encapsulate any knowledge of the programming framework, such as MapReduce. In

order to run their tasks, ApplicationMasters request resources from the ResourceManager. This architectural redesign improves scalability and efficiency, bypassing some of the limitations in Hadoop 1.0.

Virtualizing Hadoop

As physically deployed Hadoop clusters grew in size, developers asked a familiar question: can we virtualize it?

Like other enterprise (and Java-based) applications, development efforts moved to virtualization as Hadoop matured. A virtualized private cloud uses a group of hardware on the same hypervisor (such as vSphere [by VMware], XenServer [by Citrix], KVM [by Red Hat], or Hyper-V [by Microsoft]). Instead of individual servers, nodes are virtual machines (VMs) designated with master or worker roles. Each VM is allocated specific computing and storage resources from the physical host, and as a result, one can consolidate their Hadoop cluster onto far fewer physical servers. There is an up-front cost for virtualization licenses and supported or enterprise-level software, but this can be offset with the cluster's decreased operating expenses over time.

Virtualizing Hadoop created the infrastructure required to run Hadoop in the cloud, leading major players to offer web-service Hadoop. The first, Amazon Web Services, began beta testing their Elastic MapReduce service as early as 2009. Though public cloud deployment is not the focus of this review, it's worth noting that it can be useful for ad hoc or batch processing, especially if your data is already stored in the cloud. For a stable, live cluster, a company might find that building its own private cloud is more cost effective. Additionally, regulated industries may prefer the security of a private hosting facility.

In 2012, VMware released Project Serengeti—an open source management and deployment platform on vSphere for private cloud environments. Soon thereafter, they released Big Data Extensions (BDE), the advanced commercial version of Project Serengeti (run on vSphere Enterprise Edition). Other offerings, like OpenStack's Project Sahara on KVM (formerly called Project Savanna), were also released in the past two years.

Though these programs run on vendor-specific virtualization platforms, they support most (if not all) Hadoop distributions (Apache Hadoop [1.x and 2.x] and commercial distributions like Cloudera, Hortonworks, MapR, and Pivotal). They can also manage coordinating applications (like Hive and Pig) that are typically built on top of a Hadoop cluster to satisfy analytical needs.

Case Study: Hadoop on a Public Versus Private Cloud

A company providing enterprise business solutions initially turned to the public cloud for its analytics applications. Ad hoc use of a Hadoop cluster of 200 VMs cost about $40k a month. When their developers needed consistent access to Hadoop, the bills would spike by an additional $20-40k. For $80k, they decided to build their own 225 TB, 30-node virtualized Hadoop cluster. Flash-based SAN and server-based flash cards were used to enhance performance for 2-3 TB of very active data. Using Project Serengeti, it took about 10 minutes to deploy their cluster.

Another Form of Virtualization: Aggregation

Cloud computing without virtualization

Thus far, virtualization refers to using a hypervisor and VMs to isolate and allocate resources in a private cloud environment. For clarity, "virtualization" will continue to be used in this context. But building a private cloud environment isn't limited to virtualization. Aggregation (as a complement to or on top of virtualization) became a useful alternative for cloud computing (see B in Figure 2), especially as applications like Hadoop grew in size.

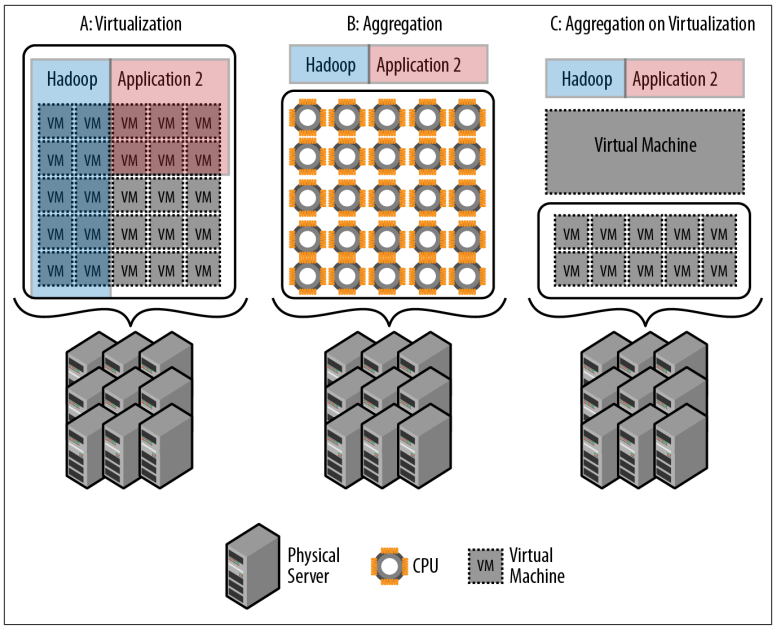

Figure 2. Strategies for cloud computing

Virtualization partitions servers into isolated virtual machines, while aggregation consolidates servers to create a common pool of resources (like CPU, RAM, and memory) that applications can share. System containers can run a full OS, like a VM, while others (application containers) contain a single process or application. This allows multiple applications to access the consolidated resources without interfering with each other. Resources can be dynamically allocated to different applications as their loads change.

In an initial study by IBM, Linux containers (LXC) and control groups (cgroups) allowed for isolation and resource control in an aggregated environment with less overhead than a KVM hypervisor.[6] The potential overhead advantages should be weighed against some limitations with LXC, such as the restriction to only run on Linux and that, currently, containers offer less performance isolation than VMs.

If an industry has already invested in virtualization licenses, aggregation can be used on a virtualized environment to provide one "super" VM (see C in Figure 2). Unless otherwise specified, however, the terms "aggregation" and "containers" here imply use on a bare metal (non-virtualized) environment.

Cluster managers

Cluster managers and management tools work on the application level to manage containers and schedule tasks in an aggregated environment. Many cluster managers, like Apache Mesos (backed by Mesosphere) and StackIQ, are designed to support analytics (like Hadoop) alongside other services.

Hadoop on Mesos

Apache Mesos provides a foundational layer for running a variety of distributed applications by pooling resources. Mesos allows a cluster to elastically provision resources to multiple applications (including more than one Hadoop cluster). Mesosphere aims to commercialize Mesos for Hadoop and other enterprise applications while building add-on frameworks (like distributed schedulers) along the way. In 2013, they released Elastic Mesos to easily provision a Mesos cluster on Amazon Web Services, allowing companies to run Hadoop 1.0 on Mesos in bare-metal, virtualized, and now public cloud environments.

Benefits of Hadoop in a Private Cloud

In addition to cost-effective setup and operation, private cloud deployment offers additive value by streamlining maintenance, increasing hardware utilization, and providing configurational flexibility to enhance the performance of a cluster.

Agility and Operational Simplicity with Competitive Performance

Deploy a Scalable, High-Performance Cluster with a Simplified Management Interface

- Benchmarking tools indicate that the performance of a virtual cluster is comparable to a physical cluster
- Built-in workflows lower initial configuration complexity and time to deployment
- Streamlined monitoring consoles provide quick performance read-outs and easy-to-use management tools
- Nodes can be easily added and removed for facile scaling

Competitive performance

Since a hypervisor demands some amount of computational resources, initial concerns about virtual Hadoop focused on performance. The virtualization layer requires some CPU, memory, and other resources in order to manage its hosted VMs,[7] though the impact is dependent on the characteristics of the hypervisor used. Over the past 5 to 10 years, however, the performance of VMs have significantly improved (especially for Java-based applications).

Many independent reports show that when using best practices, a virtual Hadoop cluster performs competitively to a physical system.[8,9] Increasing the number of VMs per host can even lead to enhanced performance (up to 13%).

Container-based clusters (like Linux VServer, OpenVZ, and LXC) can also provide near-native performance on Hadoop benchmarking tests like WordCount and TeraSuite.[10]

With such results, performance concerns are generally outweighed by the numerous other benefits provided by a private-cloud deployment.

Rapid deployment

To deploy a cluster, Hadoop administrators must navigate a complicated setup and configuration procedure. Clusters can be composed

of tens to hundreds of nodes—in a physical deployment, each node must be individually configured.

With a virtualized cluster, an administrator can speed up initial configuration by cloning worker VM nodes. VMs can be easily copied to expand the size of the cluster, and problematic nodes can be removed and then restored from backup images. Some virtualized Hadoop offerings, like BDE, can entirely automate installation and network configuration.

Using containers instead of VMs offers deployment advantages as well, as it takes hours to provision bare metal, minutes to provision VMs, but just seconds to provision containers. Like BDE, cluster managers can also automate installation and configuration (including networking software, OS software, and hardware parameters, among others).

Improved management and monitoring

A Hadoop cluster must be carefully monitored to meet the demands of 24/7 accessibility, and a variety of management tools exist to help watch the cluster. Some come with your Hadoop distribution (like Cloudera Manager and Pivotal's Command Center), while others are open source (like Apache Ambari) or commercial (like Zettaset Orchestrator). Virtualization-aware customers are already using hypervisor management interfaces (like vCenter or XenCenter) to simplify resource and lifecycle management, and a virtualized Hadoop cluster integrates as just another monitored workload.

These simplified provisioning and management tools enable Hadoop-as-a-service. Some platforms allow an administrator to hand off preconfigured templates, leaving users to customize the environment to suit their individual needs. More sophisticated cloud management tools automate the deployment and management of Hadoop, so companies can offer Hadoop clusters without users managing any configurational details.

Scalability

Modifying a physical cluster—removing or adding physical nodes—requires a reshuffling of the data within the entire system. Load balancing (ensuring that all worker nodes store approximately the same amount of data) is one of the most important tasks when scaling and maintaining a cluster. Some hypervisors, like vSphere Enterprise Ed-

ition, include distributed resource schedulers that can perform automatic load balancing.

To scale an aggregated system, cluster managers just need to be installed on new nodes. When the cluster scheduler is made aware of the new node, it will automatically absorb the offered resources and begin scheduling tasks on it.

Improved Efficiency

Create a Robust, High-Utilization Cluster

- Rather than monopolizing dedicated hardware, a private cloud cluster allows for mixed workflows for higher utilization.
- High availability and fault tolerance increase the uptime of a cluster during unanticipated outages and failures or routine maintenance.

Higher resource utilization

A physical deployment model monopolizes its dedicated hardware. Physical Hadoop clusters are often over-engineered—they are built to handle an estimated peak capacity, but left underutilized the rest of the time. Any complementary application (like a NoSQL or SQL database) requires its own dedicated hardware as well.

In a virtual deployment, resources like CPU and RAM are partitioned for the Hadoop cluster, freeing up resting resources for other tasks. Co-locating VMs running Hadoop roles (like MapReduce jobs) with VMs running other workloads (such as Hive queries on HBase) can balance the use of a system.[5] Multiple workloads can be run concurrently on the same hardware with a minimal effect on results (less than a 10% difference when compared to utilizing separate, independent workloads on a standalone cluster).[11]

An aggregated cloud also offers higher utilization. Though isolated from one another, all applications access the same pool of resources. The system can elastically scale resources for each application. Theoretically, a high-load application could use the entirety of aggregated resources (like CPU, RAM, and memory) until loads on other applications increase.

Minimizing downtime with high availability and fault tolerance

High availability (HA) protects a cluster during planned and unplanned downtime. Failovers can be deliberately triggered for maintenance or are automatically triggered in the event of failures or unresponsive service.

Virtualized HA solutions monitor hosts and VMs to detect hardware and guest operating system (OS) failures. If a server outage or failed network connection is detected, VMs from the failed host are restarted on new hosts without manual intervention (see Figure 3).[12] In the case of an OS failure, VMs are automatically restarted. In aggregated environments, failed workloads automatically failover to a new node with available resources.

HA in a Hadoop cluster can protect against the single-point failure of a master node (the Namenode or JobTracker). If desired, the entire cluster (master nodes and worker nodes) can be uniformly managed and configured for HA.[12]

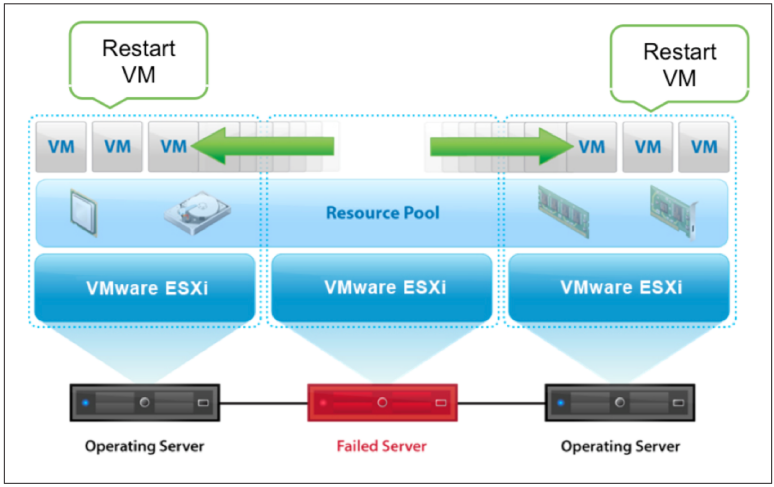

Figure 3. High availability monitoring[12]

In a virtualized environment, fault tolerance (FT) provides continuous availability by creating a live, up-to-date shadow (a secondary instance) of a VM. Though an FT system is not able to detect if the application fails or a guest OS hangs,[13] it triggers a failover procedure to the secondary VM if a VM stops due to a hardware outage or loss of network connectivity. This helps prevent data loss and decreases

downtime. Combining HA and FT can create maximum availability for a virtualized Hadoop cluster.

Flexibility

Utilize Options for Flexible Configuration

- A cluster can be built using DAS, SAN/NAS, or a hybrid combination of storage.
- Configurational options create elastic scalability to address fluctuating demands.

Hardware flexibility

By using commodity hardware and built-in failure protection, Hadoop was designed for flexibility. Virtualization takes this a step further by abstracting away from hardware completely. A private cloud can use direct attached storage (DAS), a storage attached network (SAN), or a network attached storage (NAS). SAN/NAS storage can be more costly but offers enhanced scalability, performance, and data isolation. If a company has already invested in non-local storage, their Hadoop cluster can strategically employ both direct- and network-attached devices. The storage or VMDK files for the Namenode and JobTracker can be placed on SAN for maximum reliability (as they are memory- but not storage-intensive) while worker nodes store their data on DAS.[5] The temporary data generated during MapReduce can be stored wherever I/O bandwidth is maximized.

Case Study: Hadoop on NAS

A major shipping company uses a virtualized Hadoop cluster to perform web log analysis (detecting mobile devices accessing the website), ZIP code analysis (which ZIP codes are the highest source or destination for shipments), and shipment analysis (to determine patterns that may delay a package). Their cluster is hosted on EMC Isilon NAS storage. From their perspective, Isilon helps drive down the total cost of ownership by eliminating the "triple replicate penalty" with data storage. Additionally, they've found that a fast enough network

can perform competitively to DAS (equalizing the playing field in terms of data locality).

Configurational flexibility

As previously described, organizing computation tasks to run on blocks of local data (data locality) is the key to Hadoop's performance. In a physical deployment, this necessitates that worker nodes host data and compute roles in a fixed 1:1 ratio (a "combined" model). For this model to be mimicked in a virtual cluster, each hypervisor server would host one or more VMs that contained data and compute processes (see A in Figure 4). These configurations are valid, but difficult to scale under practical circumstances. Since each VM stores data, the ease of adding or removing nodes (simply copying from a template or using live migrate capabilities) would be offset by the need to rebalance the cluster.

If instead, compute and data roles on the same hypervisor server were in separate VMs (see B in Figure 4), compute operations could be scaled according to demand without redistributing any data.[14] Likewise in an aggregated cloud, Apache Mesos only spins up TaskTracker nodes as a job runs. When the task is complete, the TaskTrackers are killed, and their capacity is placed back in the consolidated pool of resources.

This "separated" model is fairly intuitive (since the TaskTracker controls MapReduce and the Namenode controls the data storage) and the flexibility of virtualized or aggregated clusters makes it relatively simple to configure.

In addition to creating this elastic system (where compute processes can be easily cloned or launched to increase throughput), the separated model also allows you to build a multi-tenant system where multiple, isolated compute clusters can operate on the same data. The same cloud could host a production Hadoop cluster as well as a development and QA environment.

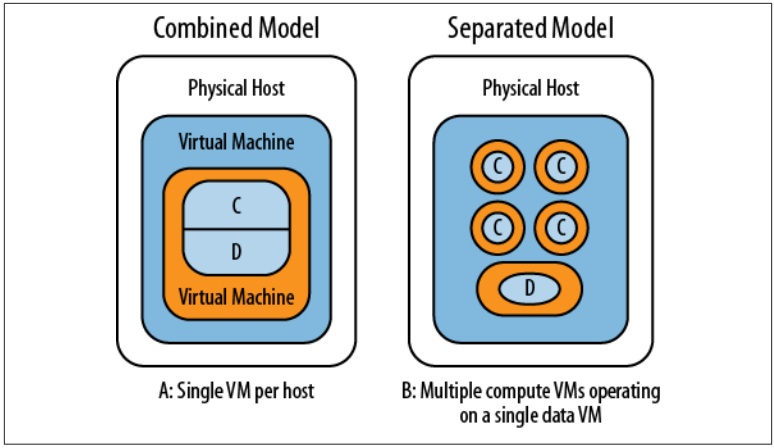

Figure 4. Configurational flexibility with compute and data processes (C: compute; D: data). Figure modified from VMware's "Deploying Virtualized Hadoop Systems with VMware vSphere Big Data Extensions (BDE)."[5]

Virtual rack awareness

Using the separated model does carry an added complication. If compute processes can scale on demand, the cluster's topology is dynamic (compared to the fixed structure of a physical deployment).

In a virtualized cluster, a compute and data node on the same hypervisor server communicate over a virtual network in memory (without suffering physical network latencies). It is important that the system maintain virtual "rack awareness" to preserve the performance advantage of data locality. To accommodate this need, VMware contributed a tool called Hadoop Virtualization Extensions (HVE) to Hadoop.[15] HVE accounts for the virtualization layer by grouping all VMs on the same host in a new domain. For best performance, the cluster can use this domain to direct computation to perform on the same hypervisor server hosting the data. It can also intelligently place data replicates on separate hosts to provide maximum protection in the event of a hardware failure.

Separately scaled data and compute processes presents a similar challenge in an aggregated environment. If an application can request tasks and be offered resources throughout the entire cloud, how does it know which nodes store the data required? Adding constraints allows an application to reject resource offers that don't meet its requirements.[16] A technique called "delay scheduling," in which an application can wait if it cannot launch a local task, can result in nearly optimal data locality.[17]

Conclusions

Apply the Resources and Best Practices You Already Know

If planning your first cluster or a deployment overhaul, it is important to consider the following:

- Current data storage
- Estimated data growth
- The amount of temporary data that will be stored during MapReduce processing
- Throughput and bandwidth needs
- Performance needs
- The resources (hardware and software) you have available to dedicate to the cluster
- The resources (hardware and software) you'd need to purchase to dedicate to the cluster

It would be difficult to specify an ideal architecture for every Hadoop cluster, as analytical demands and resource needs vary widely. But planning a private cloud cluster doesn't necessitate a large learning curve either.

Many companies can utilize resources (e.g., virtualization licenses, cluster managers, DAS, or SAN/NAS storage) they already have. Additionally, many of the best practices an IT department already puts in place (like avoiding VM contention and optimizing I/O bandwidth) translate well to configuring a high-performance Hadoop cluster.

Benefits of Virtualizing Hadoop

Whether Hadoop is deployed in a physical system or in a private or public cloud, the goals of a well-established infrastructure are the same. In any implementation, Hadoop should provide:

- Cost-effective and high-performance data analysis
- Failure-tolerant data storage
- Scalable capability for future growth
- Minimal downtime

For Hadoop administrators and users, a private cloud offers unique benefits with comparable (even improved) performance. Rapid deployment and built-in workflows ease initial configuration complexity, to the point where developers can use built-in tools to configure their own test environments without involving IT. Management tools make it easier to monitor and analyze performance, and features like high availability and fault tolerance decrease downtime.

The need for low-latency data management systems will grow in coming years, and enterprise applications continue to move from physical systems into cloud computing infrastructures. Using a private cloud can create a scalable, streamlined Hadoop cluster built to accommodate data science's evolving landscape.

Notes

1. "Global Hadoop Market (Hardware, Software, Services, HaaS, End User Application and Geography) - Industry Growth, Size, Share, Insights, Analysis, Research, Report, Opportunities, Trends and Forecasts Through 2020"

2. "Big Data: The Next Big Thing." Nasscom, New Delhi, 2012.

3. "MapReduce: Simplified Data Processing on Large Clusters"

4. Hadoop: The Definitive Guide

5. VMware, Inc. "Deploying Virtualized Hadoop Systems with VMware vSphere Big Data Extensions." 2014.

6. Felter, W., A. Ferreira, R. Rajamony, and J. Rubio. "An Updated Performance Comparison of Virtual Machines and Linux Containers." IBM Research Report, 2014.

7. Microsoft IT Big Data Program. "Performance of Hadoop in Hyper-V." MSIT SES Enterprise Data Architect Team, 2013.

8. Buell, J. "A Benchmarking Case Study of Virtualized Hadoop Performance on VMware vSphere 5."

9. Buell, J. "Virtualized Hadoop Performance with VMware vSphere 5.1." VMware, Inc., 2013.

10. Xavier, M.G., M.V. Neves, and A.F. De Rose. "Performance Comparison of Container-based Virtualization MapReduce Clusters." *IEEE*, 2014.

11. Intel, VMware, and Dell. "Scaling the Deployment of Multiple Hadoop Workloads on a Virtualized Infrastructure." 2013.

12. Hortonworks and VMware. "Apache Hadoop 1.0 High Availability Solution on VMware vSphere." 2011.

13. Buell, J. "Protecting Hadoop with VMware vSphere 5 Fault Tolerance." VMware, Inc., 2012.

14. Magdon-Ismail, T., et al. "Toward an Elastic Elephant – Enabling Hadoop for the Cloud." *VMware Technical Journal* 2, no. 2 (December 2013): 56-64.

15. VMware, Inc. "Hadoop Virtualization Extensions on VMware vSphere 5."

16. B. Hindman et. al. "Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center." UC Berkeley, 2010.

17. M. Zaharia et al. "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling." *Proceedings of the 5th European Conference on Computer Systems*, 2010.

About the Author

Courtney Webster is a freelance writer with professional experience in laboratory automation, automated data analysis, and the application of mobile technology to clinical research. You can follow her on Twitter *@automorphyc* and find her blog at *http://automorphyc.com*.