An Analytics System on OpenStack™ for Manufacturing

Leveraging OpenStack and Object Storage to Reduce Costs and Increase Return on Investment

Author | Sanhita Sarkar, Western Digital Corporation
Executive Overview

The emergence of the software-defined economy is driving the architectural transformation of data centers. Organizations need a quick response to the demands of new services, with speed and agility. OpenStack™, a cloud-based computing platform and deployed as Infrastructure-as-a-Service (IaaS), provides organizations an increased level of flexibility to deploy storage, compute and network of resources within a data center, in a cost-effective manner. A few business benefits of OpenStack for Big Data workloads are the ability for rapid and dynamic provisioning of a cluster, ability to dynamically change the role of nodes providing elasticity, ability to scale resources as users grow, along with its open source benefits. Sahara in OpenStack has been developed to meet the need for agile access to Big Data and to provide unlimited scalability, elasticity and data availability.

This paper describes an analytics system using Sahara on a hosted OpenStack private cloud and how it can provide the flexibility of provisioning infrastructure resources to a real-world manufacturing workload with the growth of data and its users. The analytics system connects to an ActiveScale object storage system, an external object data store, and can dynamically spin up a virtual compute cluster for a set of jobs, without moving the data. This paper demonstrates performance of the above-mentioned analytics system within the cloud environment to be 80–99% of native with a reduced TCO and an improved ROI.
1. Introduction

When we think of computer resources in the cloud, we usually think of public clouds, with infrastructure or applications shared by millions of clients worldwide, through the internet. However, organizations with security concerns cannot move directly to public clouds, but must invest in private clouds, instead. A few good qualities of private clouds are offering resources (infrastructure and applications) as services; flexibility and scale to meet client demands; resource sharing among a large number of corporate users and others. Physical hardware running on premise takes time and cost for organizations to host it across multiple functions. Instead, an IaaS model in a private cloud allows for availability of hardware within hours instead of days, saving time to market for organizations. Additionally, OpenStack IaaS provides an increased level of flexibility, allowing for growing and shrinking resources on demand within a private cloud, as well within hybrid clouds, if needed. Out of various deployment models, hosted OpenStack private cloud is where a vendor hosts an OpenStack-based private cloud including underlying hardware and the OpenStack software. Figure 1 shows the trends of different cloud models over the next couple of years, with hosted private cloud being one of the growth areas. The paper describes a hosted OpenStack private cloud implementation within a corporate data center for executing manufacturing business processes along with supporting various corporate users who access the data for deriving insights into quality control and operational efficiency (Figure 2). Here, the data is centralized and lives within the remote private cloud, and accessed by corporate users.

On the OpenStack-managed infrastructure, Sahara is then used to rapidly create and manage Apache™ Hadoop® clusters and run workloads on them, without the need of any cluster management. Sahara can install and manage several clusters at the same time, providing tools to add nodes to the existing cluster, remove nodes for maintenance or to dynamically provision clusters of different sizes to meet the needs of varied workloads. It provides predefined configurable templates to install a Hadoop cluster in a few minutes just by specifying several parameters. Because Big Data workloads are heavy in terms of required CPU and RAM, it is highly recommended to enable hardware virtualization on the OpenStack compute nodes. In this case, Red Hat® Kernel-based Virtual Machine (KVM) have been used to build Big Data /Hadoop instances on the analytics system (Figure 4).

Manufacturing data comprising of complex binary files generated from the shop floors of multiple sites is continuously archived into an ActiveScale Object Storage System. As the data arrives, queries using Apache Hive™ user-defined functions (UDFs) which analyze device defect patterns are executed on the virtual machines provisioned by Sahara on the OpenStack cloud. Results are stored in a Hive table, and accessed by Impala query engine to display device defect quality metrics and sigma deviations from standard baseline, on a Tableau® visualizer. Predictive analytics is performed using K-means clustering machine learning algorithm enabled by an IBM® SPSS® engine to classify the defect patterns and detect any new patterns as a part of unsupervised learning (Figure 3).
cloud implementation using Sahara provisioning of Big Data clusters.

The rest of the paper will focus on the implementation and will compare results time to time with (5b) and (5a). Details of implementation and optimizations for (5b) are not addressed in this paper, but discussed in a separate paper.

2. Implementation Details

This section describes the implementation details of the manufacturing workflow on premise on a native system as well as on OpenStack Cloud.

2.1. Specifications for Native System

The specifications of the native implementation are as follows (Figure 5b):

- 1 ActiveScale Object Storage System
- Analytics System:
  - 16 Data/Worker nodes each with 24 cores /48 vcpus (Intel® Xeon® E5-2680 V3), 24 drives, 512 GB memory
  - 3 Master Nodes
  - 4 Edge Nodes
  - Dual 48-port 10GbE switches for data network
  - 48-port Management switch.
2.2.1. Resource Pool (Max Quota) Configurations on OpenStack

- 120 Instances
- 824 vcpus
- 8 TB memory
- 120 volumes
- 300 TB total disk volume
- Cloudera™ OpenStack Sahara Plugin: CDH-5.4.5-sahara-centos-S3A-v001 for Hadoop CDH plugin
- Guest OS: CentOS-6-x86_64-GenericCloud-1607 for Guest OS
- Three flavors are created for Hadoop VMs

2.2.2. Logical Hadoop Data Cluster Configuration

Configuration 1:
16 Node CDH cluster (784 vcpus, 4352 GB, 1.6TB disk space for HDFS)

- 16 VMs using flavor 2, as Hadoop Data Nodes
- 2 VMs using flavor 3, as Hadoop Master Nodes

Configuration 2:
32 Node CDH cluster (784 vcpus, 6400 GB, 3.2TB disk space for HDFS)

- 32 VMs using flavor 1, as Hadoop Data Nodes
- 2 VMs using flavor 3, as Hadoop Master Nodes

Figure 6 shows the software layout of an analytics system dynamically provisioned with OpenStack Sahara in the private cloud. The logical system within the cloud is connected to an externally located ActiveScale Object Storage System.
Results from OpenStack (16 VMs) configuration are compared to native non-cloud baseline configuration of 16 physical nodes. The following sections provide a summary of the performance results.

### 3. Performance Results

This section describes the performance characterization done for the manufacturing workload as follows:

- Uses **three types of queries** which are compute-intensive, I/O-intensive, and with a mixed compute and I/O operations;
- Uses **three scenarios** of query execution as follows:
  - **Scenario 1**: Hadoop HDFS for Input and Output files;
  - **Scenario 2**: ActiveScale Object Storage System used for Input files and Hadoop HDFS for Output files;
  - **Scenario 3**: ActiveScale Object Storage System used as a default Hadoop HDFS, for both Input and Output files.

- Compares performance results of the queries in the above three scenarios running on **native non-cloud system with those in OpenStack cloud**;
- Compares performance results of the queries in the above three scenarios within OpenStack cloud for **varying number of virtual machines**.

Two configurations of OpenStack have been used for performance characterization:

1. OpenStack (16 VMs) with 1 VM per node;
2. OpenStack (32 VMs) with 1 VM per NUMA node (socket). NUMA stands for Non-uniform Memory Access.

<table>
<thead>
<tr>
<th>Big Data Components</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturing process data persistence within ActiveScale Object Storage System</td>
<td>Custom code</td>
</tr>
<tr>
<td>Connectivity between ActiveScale Object Storage and analytics cluster</td>
<td>S3A Connector</td>
</tr>
<tr>
<td>OpenStack® Library release</td>
<td>12.0.4-0</td>
</tr>
<tr>
<td>Sahara</td>
<td>3.0.0 (comes with Liberty Release)</td>
</tr>
<tr>
<td>Operating System (Host)</td>
<td>Ubuntu® 14.04.5 LTS</td>
</tr>
<tr>
<td>Operating System (Guest)</td>
<td>RHEL 6.6 (CentOS 6.6)</td>
</tr>
<tr>
<td>Java® for Cloudera® functionality</td>
<td>Oracle® JDK 1.7</td>
</tr>
<tr>
<td>Cloudera® OpenStack Sahara plugin</td>
<td>CDH 5.4.5</td>
</tr>
<tr>
<td>Impala (comes with CDH 5.4.5)</td>
<td>Impala (comes with CDH 5.4.5)</td>
</tr>
<tr>
<td>Hive (comes with CDH UDFs)</td>
<td>Hive (comes with CDH UDFs)</td>
</tr>
<tr>
<td>Advanced Manufacturing Analytics and Machine Learning</td>
<td>Custom Code using Hive UDFs</td>
</tr>
<tr>
<td>Decoding</td>
<td>Hive SerDe</td>
</tr>
<tr>
<td>Statistical data discovery**</td>
<td>JMP from SAS</td>
</tr>
<tr>
<td>Predictive Analytics</td>
<td>SPSS from IBM®</td>
</tr>
<tr>
<td>Data Mining and Advanced Analytics**</td>
<td>SAS, R</td>
</tr>
<tr>
<td>Open Source User Interface for R**</td>
<td>RStudio®</td>
</tr>
<tr>
<td>Interactive Visualization</td>
<td>Tableau®</td>
</tr>
</tbody>
</table>

### 3.1. Performance of Compute-intensive Analytical Query

For **compute-intensive analytical query on OpenStack cloud** (Figure 7):

- The performance of scenario 2 is **similar** to scenario 1, when writing the output to HDFS.
- The performance of scenario 3 is a bit **lower** compared to scenarios 1 and 2, with an increased slowdown with larger input file sizes. The reason for this slowdown is as follows:

  Results from OpenStack (16 VMs) configuration are compared to native non-cloud baseline configuration of 16 physical nodes.

  The software specifications are shown in Table 1.
• Deletion and insertion of result files (while executing the ‘Insert overwrite’ SQL statement) takes longer time in the ActiveScale Object Storage System, compared to HDFS, thus increasing the overall query execution time.

• With larger input file sizes, the number of output files to be updated in the ActiveScale Object Storage System also increases, thus increasing the overall query execution time.

While comparing compute-intensive analytical query performance on OpenStack cloud with native (Figures 8a, 8b and 8c), we can summarize the results as follows:

• For all 3 scenarios, with smaller file sizes up to 100GB, compute-intensive analytical query performance on OpenStack (16 VMs) is 84%-94% of native (16 nodes). Beyond 100GB, OpenStack performs up to 10%-17% better than native. This is because of a denser distribution and load provisioning capability of OpenStack, compared to native. For smaller files with a sparse distribution, the overhead of OpenStack KVM hypervisor dominates over its benefits.

• In scenario 3, for larger file sizes, the benefit of OpenStack is lessened by the overhead of updating many large files, in contrast to other scenarios.

While comparing compute-intensive analytical query performance on OpenStack (32 VMs) with OpenStack (16 VMs) (Figures 8a, 8b and 8c), we can summarize the results as follows:

• In all 3 scenarios, the performance of OpenStack (32 VMs) is up to 18% higher than OpenStack (16 VMs), beyond 300GB. This is due to the process and memory residing on the same NUMA node with 32 VMs affinitized to 32 sockets in the cluster, compared to that with 16 VMs, where 1 VM shares two sockets on a node. So for larger file sizes, the overall OS overhead (context switching, scheduling) on 32 VMs is lowered, compared to that on 16 VMs.

3.1. Performance of I/O-intensive Analytical Query

For I/O-intensive analytical query on OpenStack cloud (Figure 9):

• The performance of scenario 2 is similar or
- better compared to HDFS (scenario 1), for all input file sizes ranging from 10GB to 300GB. For larger file sizes, 500GB and 1TB, lack of a bonded network played a role in slower performance with scenario 2, where the input files are being read from the ActiveScale Object Storage System.

- The performance of scenario 3 is lower than scenario 1 with the degradation being significantly higher for larger file sizes (~2x). The reasons for this slowdown with larger file sizes are two-fold: a) an increase in the number of output files to be updated in the ActiveScale Object Storage System and b) system limitation due to a lack of network bonding between the analytics system and ActiveScale Object Storage System.

While comparing I/O-intensive analytical query performance on OpenStack cloud with native (Figures 10a, 10b and 10c), we can summarize the results as follows:

- For scenario 1, with smaller file sizes up to 300GB, I/O-intensive analytical query performance on OpenStack (16 VMs) is 72%-99% of the native (16 nodes). Beyond 300GB, OpenStack performs 20% better than native. This is because of the benefits from a denser distribution and load provisioning capability of OpenStack and throughput of GFS, compared to the native ext3 filesystem for HDFS (Figure 10a).

- For scenarios 2 and 3, IO-intensive analytical query performance on OpenStack is 80%-99% of native (Figures 10b and 10c).

While comparing I/O-intensive analytical query performance on OpenStack (32 VMs) with OpenStack (16 VMs) (Figures 10a, 10b and 10c), we can summarize the results as follows:

- Using 1 LUN per VM, scenario 1 performs better on OpenStack (32 VMs) compared to OpenStack (16 VMs), because 32 LUNs provides better I/O throughput compared to 16 LUNS on the GFS (Figure 10a).

- For scenarios 2 and 3, the above benefit is not significant because most of the I/O is over the network, and moreover 32 VMs incur more network overhead on larger file sizes, compared to 16 VMs (Figures 10b and 10c).
3.3. Performance of Mixed Analytical Query

This section summarizes the performance results for an analytical query executing mixed compute and I/O operations (Figure 11) on OpenStack cloud, native and public cloud.

Considering the daily average binary data with an input file size of 50GB for 13 products from 4 manufacturing sites:

- Analytical query on OpenStack is ~14–15x faster when getting daily defect patterns for scenario 1, compared to the public cloud (projected from reported data). The public cloud projection is calculated for a best case scenario. The worst case scenario is reported to be ~24–48 hrs.
- Analytical query to get daily defect patterns for scenarios 2 and 3 on OpenStack and native is pretty much similar in performance, compared to scenario 1.
- Analytical query performance on OpenStack is 84–90% of native.
- The price/performance benefits of scenarios 2 and 3 over scenario 1 is a driving factor for customers while making purchase decisions because they don’t need to maintain a large static Hadoop cluster, but can leverage ActiveScale Object Storage System, instead.

* Out of 500GB-600GB data ingested per day to the Big Data platform on the public cloud, 50GB is the size of average daily binary data which is being used by the analytical query.

3.3. Summary of Performance Results

A prototype of an analytics system within a hosted OpenStack private cloud has been implemented for manufacturing.

It has Amazon™ Elastic MapReduce-like (EMR-like)-capabilities of dynamic cluster provisioning, with all data stored within an externally located ActiveScale Object Storage System. This yields efficient resource utilization for a desirable ROI. The solution provides a performance improvement of ~14–15x over a public cloud through process optimizations, for analyzing defect patterns of products from all MFG sites, at 1/4th of the TCO.

The query performance within OpenStack is 80–99% of native, depending on the scenario.

Table 2 summarizes a case study demonstrating the performance of compute-intensive and I/O-intensive workloads with varying input file sizes, within an OpenStack cloud.

The three scenarios used for the case study are as follows:

**Scenario 1**: HDFS for input and output files

**Scenario 2**: ActiveScale Object Storage System for Input files and HDFS for Output files

**Scenario 3**: ActiveScale Object Storage System is used as the default HDFS

Please refer to the Appendix for a list of parameters within an OpenStack cloud that has been tuned for the manufacturing workload to get the best possible results.

4. Total Cost of Ownership

This section compares the Total Cost of Ownership (TCO) for hosting a manufacturing production Big Data analytics platform on a public cloud with the TCO for the analytics system hosted on an OpenStack private cloud.

Calculations include the operational costs to host the ActiveScale analytics solution in a data center and the depreciation cost of the hardware over 5 years. The cost of the production environment in the public cloud is based on typical/reported pricing for a similar configuration described in Section 2, which has details about the OpenStack private cloud.

<table>
<thead>
<tr>
<th>Input File size (considering 3.5GB) as network bandwidth, 768 vcpus, 1 mapper/vcpu</th>
<th>Workload Type</th>
<th>OpenStack: Performance of ActiveScale Object Storage System as an external storage for input files (Scenario 2) vs. HDFS (Scenario 1)</th>
<th>Performance of ActiveScale Object Storage System as a default HDFS (Scenario 3) vs. HDFS (Scenario 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 300GB</td>
<td>Compute-intensive analytical workloads</td>
<td>Similar</td>
<td>Similar or 13% lower</td>
</tr>
<tr>
<td></td>
<td>I/O-intensive analytical workloads</td>
<td>Similar or lower (5%)</td>
<td>Similar for 10-20 output file updates</td>
</tr>
<tr>
<td>&gt;= 300GB</td>
<td>Compute-intensive analytical workloads</td>
<td>Similar or lower (5%)</td>
<td>Similar for 10-20 output file updates</td>
</tr>
<tr>
<td></td>
<td>I/O-intensive analytical workloads</td>
<td>Similar or lower (50%)</td>
<td>Lower (30%) for a larger number of output file updates</td>
</tr>
</tbody>
</table>

Table 2: Performance of compute-intensive and I/O-intensive workloads for varied input file sizes within OpenStack cloud; scenario 2 vs. scenario 1 and scenario 3 vs. scenario 1
OpenStack Private Cloud solution. The TCO for 5 years for both solutions is compared and shown in Figure 12.

The TCO for operating the analytics solution on a hosted OpenStack private cloud is 1/4th of that on the public cloud.

Dynamic provisioning capability of Big Data clusters using Sahara on OpenStack, allows for an efficient utilization of resources, thus yielding a better ROI.

Conclusions

The paper demonstrates how an analytics system hosted on an OpenStack private cloud can solve a real-world manufacturing problem and transform data to knowledge to wisdom, at a low Total Cost of Ownership (TCO) and a desired Return of Investment (ROI).

OpenStack provides an efficient solution as an IaaS with performance as good as native, along with the ability to dynamically provision Big Data clusters using Sahara. This allows for a better ROI due to an efficient utilization of available resources. Other solutions for provisioning will also be explored in near future.

On OpenStack cloud, the two scenarios with ActiveScale Object Storage System behave similarly to the native configuration.

With an ActiveScale Object Storage System as external storage, customers can achieve Amazon EMR-like capabilities within OpenStack cloud because they can:

- Scale the compute and storage independently
- Create, grow, shrink or destroy the compute clusters, on demand
- Avoid storing the data in the clusters for a lifetime
- Dynamically reuse the compute cluster for executing different applications, as the need arises

A combined solution of OpenStack IaaS with an ActiveScale object storage system can deliver a private cloud solution for analytics, similar to Amazon EMR but at a low cost.
Appendix

1. Tunables for ActiveScale Object Storage System
Load Balancer for OpenStack

1.1. HAProxy Load Balancer Tuning:
- Load balancer ‘haproxy’ is configured from
  single thread to multi-threaded, assigned
  threads to be equal to the number of vcpus.
- Dedicated server is assigned to the Load
  balancer.
- Each load balancer thread affinities to a
  physical core.
- Timeout for client and server is increased to
  3 minutes
- Max connections increased to 8000 for front end

1.2. Edited or Added the following parameters in
haproxy.cfg file (for a server with 24 vcpus)

```
maxconn 8000
nbproc 23
cpu-map 1 1
cpu-map 2 2
cpu-map 3 3
cpu-map 4 4
cpu-map 5 5
cpu-map 6 6
cpu-map 7 7
cpu-map 8 8
cpu-map 9 9
cpu-map 10 10
cpu-map 11 11
cpu-map 12 12
cpu-map 13 13
cpu-map 14 14
cpu-map 15 15
cpu-map 16 16
cpu-map 17 17
cpu-map 18 18
cpu-map 19 19
cpu-map 20 20
cpu-map 21 21
cpu-map 22 22
defaults
  timeout client 6m
  timeout server 3m
front-end
  maxconn 8000
  timeout client 3m
bind-process 1 2 3 4 5 6 7 8 9 10 11
```

2. S3A parameters in core-site.xml for OpenStack

2.1. The following parameters are added for S3A
connector in core-site.xml:

```
<property>
  <name>fs.s3a.impl</name>
  <value>org.apache.hadoop.fs.s3a.S3AFileSystem</value>
</property>
<property>
  <name>fs.s3a.proxy.host</name>
  <value>192.168.1.72</value>
</property>
<property>
  <name>fs.s3a.proxy.port</name>
  <value>9007</value>
</property>
<property>
  <name>fs.s3a.block.size</name>
  <value>67108864</value>
  <!-- 64MB -->
</property>
<property>
  <name>fs.s3a.attempts.maximum</name>
  <value>10</value>
</property>
<property>
  <name>fs.s3a.connection.establish.timeout</name>
  <value>500000</value>
</property>
<property>
  <name>fs.s3a.connection.timeout</name>
  <value>500000</value>
</property>
<property>
  <name>fs.s3a.paging.maximum</name>
  <value>100</value>
</property>
<property>
  <name>fs.s3a.connection.ssl.enabled</name>
  <value>false</value>
</property>
<property>
  <name>fs.s3a.endpoint</name>
  <value>s3.hgst.com</value>
</property>
<property>
  <name>fs.s3a.threads.max</name>
  <value>4</value>
</property>
```
3. Operating System Kernel Parameters for OpenStack

3.1. The following entries are added to /etc/sysctl.conf

```ini
net.core.rmem_max = 33554432
net.core.wmem_max = 33554432
net.ipv4.tcp_rmem = 4096 87380 33554432
net.ipv4.tcp_wmem = 4096 65536 33554432
net.ipv4.tcp_window_scaling=1
net.ipv4.tcp_timestamps=1
net.ipv4.tcp_sack=1
net.ipv4.tcp_tw_reuse=1
net.ipv4.tcp_keepalive_intvl=30
net.ipv4.tcp_fin_timeout=30
net.ipv4.tcp_keepalive_probes=5
net.ipv4.tcp_no_metrics_save=1
net.core.netdev_max_backlog=30000
net.ipv4.route.flush=1
fs.file-nr = 197600 0 362409
fs.file-max=10000000
net.ipv4.tcp_max_syn_backlog=16384
net.ipv4.tcp_synack_retries=1
net.ipv4.tcp_max_orphans=400000
```

3.2. The following entries are added to /etc/security/limits.conf file

```
* soft nofile 100000
* hard nofile 100000
```

© 2018 Western Digital, Inc. 5601 Great Oaks Pkwy, San Jose, CA 95119 USA. Produced in the United States 4/18. All rights reserved.

Western Digital logo is a registered trademark or trademarks of Western Digital Corporation or its affiliates in the US and/or other countries. Amazon, AWS, and Amazon S3 are trademarks of Amazon.com, Inc. or its affiliates. Apache®, Apache Hadoop, Apache Hive, and Hadoop® are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks. The OpenStack™ Word Mark is either a registered trademarks/service marks or trademarks/service mark of the OpenStack Foundation, in the United States and other countries and is used with the OpenStack Foundation’s permission. All other marks are the property of their respective owners.

This paper may contain forward-looking statements, including statements relating to expectations for cloud computing models, the market for our products, product development efforts, and the capacities, capabilities and applications of our products. These forward-looking statements are subject to risks and uncertainties that could cause actual results to differ materially from those expressed in the forward-looking statements, including development challenges or delays, supply chain and logistics issues, changes in markets, demand, global economic conditions and other risks and uncertainties listed in Western Digital Corporation’s most recent quarterly and annual reports filed with the Securities and Exchange Commission, to which your attention is directed. Readers are cautioned not to place undue reliance on these forward-looking statements and we undertake no obligation to update these forward-looking statements to reflect subsequent events or circumstances.