

# TPCx-HS on the Cloud!

Nicholas Wakou<sup>1</sup>(✉), Michael Woodside<sup>1</sup>, Arkady Kanevsky<sup>1</sup>,  
Fazal E Rehman Khan<sup>2</sup>, and Mofassir ul Islam Arif<sup>2</sup>

<sup>1</sup> Dell Inc., Round Rock, TX 78682, USA  
{Nicholas.Wakou, C.Michael.Woodside,  
Arkady.Kanevsky}@dell.com

<sup>2</sup> xFlow Research Inc., Software Technology Park, Islamabad, Pakistan  
{Fazal.Rehman, Mofassir.Arif}@xFlowresearch.com  
<http://www.Dell.com>  
<http://www.xFlowResearch.com>

**Abstract.** The introduction of web scale operations needed for social media coupled with ease of access to the internet by mobile devices has exponentially increased the amount of data being generated every day. By conservative estimates the world generates close to 50,000 GB of data every second, 90% of which is unstructured, and this growth is accelerating. From its origins as a web log processing system at Yahoo, the open source nature and efficient processing of Apache Hadoop has made it the industry standard for Big Data processing.

TPCx-HS was the first benchmark standard by a major Industry-Standard performance consortium for the Big Data space. TPCx-HS is a derivative of Apache Hadoop Workloads; Teragen, Terasort and Teravalidate. Ever since its release by the TPC in August 2014, all the 18 results published (as of August 2016) have been based on on-premise, Bare-metal hardware configurations.

This paper will show how Hadoop can be deployed on an OpenStack cloud using the OpenStack Sahara project and how TPCx-HS can be used to measure and evaluate the performance of the Cloud under Test (CuT). It will also show how an OpenStack cloud can be optimized to get the performance of TPCx-HS on the Cloud to match as closely as possible that on a Bare-metal configuration. Lastly, it will share results and experiences based on a Hadoop on Cloud Proof-of-Concept (POC), a study that was undertaken by the Dell Open Source Solutions team.

**Keywords:** Apache Hadoop · OpenStack · Big data · Cloud · TPCx-HS · Benchmark

## 1 Introduction

The complex nature of big data is primarily due to the unstructured nature of much of the data that is generated by modern technologies such as that from web logs, RFID, sensors, and smart phones [1]. This coupled with web scale operations of companies like Google, Yahoo and Facebook exponentially increased the amount of data being generated. This was the turning point in the big data life cycle which demanded the need for a system to efficiently manage and process these large amounts of data. Hadoop emerged from the efforts of these data giants and due to the open source nature

of several of its key components, quickly became the standard for managing large volumes of unstructured data. The two main components of Apache Hadoop platform are Hadoop Distributed File System (HDFS) and MapReduce. A number of major Computer companies now offer Hadoop-based solutions for analyzing big data use-cases. An industry-standard benchmark was therefore required to compare and differentiate these offerings. TPC Express Benchmark<sup>TM</sup>HS (TPCx-HS) was developed to provide an objective measure of hardware, operating system and commercial Apache Hadoop File System API compatible software distributions, and to provide the industry with verifiable performance, price-performance and availability metrics [2]. Traditionally Hadoop is deployed on customer premise and on physical servers. Due to its inherently efficient nature in terms of resource allocation and utilization, there appears little need for a cloud Hadoop deployment. This trend is changing with the advancements of cloud technology. A cloud deployment offers the ability to conveniently scale the cluster as needed. Multi-tenancy is also a big advantage when it comes to facilitating multiple users on the same physical hardware. Furthermore, a cloud deployment, coupled with multi-tenancy, is greatly complimented by the resources and security segregation. Each tenant has full control over their resources without incurring any risk to the resources managed by the other tenants [3]. Hadoop can now be run in a cloud in a way that is efficient and performance can be made comparable with physical hardware after a few configurations and tweaks. This paper provides recommendations for cloud configuration and Hardware options for running Hadoop workloads in the cloud.

## 2 Related Work

Google searches on Hadoop on Cloud show that some work has been done on running Hadoop on a public cloud [4] and on private clouds [5]. For the most part, moving from the traditional Bare-metal deployment of Hadoop to the Cloud is still seen as a challenge by Enterprises mainly due to performance concerns in the Cloud. One of the most related studies to this paper was work conducted by Accenture on a price-performance comparison of a Bare-metal Hadoop cluster and cloud-based Hadoop clusters. Accenture used their own TCO model and the Accenture Data Platform Benchmark which provided three real-world Hadoop applications to compare the execution-time performance of the clusters [6]. The above mentioned references make the case for running Hadoop on a cloud; the advantages and challenges. They show that despite performance challenges, it still makes sense to run Hadoop on the Cloud. This paper shows that with appropriate cloud configurations and settings, Hadoop performance on the cloud can match that on Bare-metal. This study deployed TPCx-HS Big Data workloads on an OpenStack Cloud.

## 3 System Under Test

There were 2 main Systems under Test; Cloud and Bare-metal. The Bare-metal system consisted of 4× Dell R730xd servers, the details can be found in Table 1.

**Table 1.** Bare-metal System under Test

Role	Model	Qty.	CPUs	Memory	Storage	Network adaptor
Name Node	Dell R730xd	1	2× Intel 12-Core E5-2690 v3	128 GB, 8 × 16 GB DIMMS, 2133 MT/s	16 × 1 TB (2.5", 7.2 K, HDD, SAS, JBOD), 2 × 300 GB (2.5", HDD, SAS, RAID 1)	Intel 2P 10G X520, Intel 2P 10G X520 + 2P 1G I350 rNDS
Data Node	Dell R730xd	3	2× Intel 12-Core E5-2690 v3	128 GB, 8 × 16 GB DIMMS, 2133 MT/s	16 × 1 TB (2.5", 7.2 K, HDD, SAS, JBOD), 2 × 300 GB (2.5", HDD, SAS, RAID 1)	Intel 2P 10G X520, Intel 2P 10G X520 + 2P 1G I350 rNDS

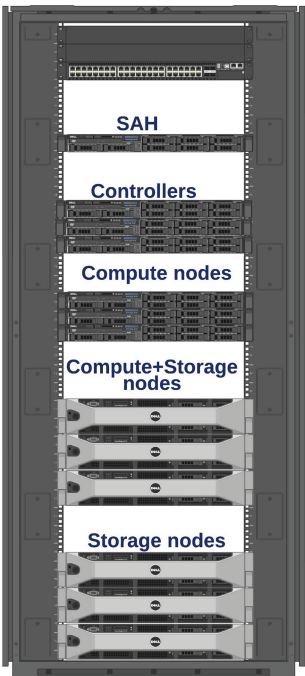
The Cloud system was based on the Dell Red Hat OpenStack Cloud Solution Reference Architecture (RA) version 4.0.1 based on Red Hat OpenStack Cloud Platform 7 (OSP7). Details are provided in Fig. 1. The Cloud setup used 2 types of storage configurations both shown in Fig. 1.

1. Ceph Storage for all Compute nodes; for both block and Nova ephemeral storage.
2. Local Storage on each Compute node; for both block and Nova ephemeral storage.

Sahara is an OpenStack project for deployment and management of Hadoop clusters. Sahara was used to deploy Cloudera CDH 5.3 that in turn used OpenStack APIs to manage Instances to run Hadoop. The Hadoop cluster consisted of Cloudera Manager Instance (EdgeNode), Namenode Instance (Master) and Datanode Instances (Workers). The physical resources allocated to these instances are listed in Table 2. Each Compute node was configured as a separate OpenStack availability zone in order to keep the Cloudera Manager and Namenode Instances on one Compute node and then evenly distribute the Worker Instance(s) across all Compute nodes. Unless otherwise stated, Datanode instances will be referred to as Workers.

**Table 2.** Resource allocation

Role	Number of instances	vCPU	Memory	Compute node
Cloudera manager instance	1	2	8 GB	Compute node 1
Namenode instance	1	6	16 GB	Compute node 1
Worker instances	3–60	40	80 GB	Compute nodes 1–3



### Networking

2xDell S4048 10 GbE  
1xDell S3048 1GbE

### SAH

1xR630

### Processors

2xIntel 12-Core E5-2650v4

### Memory

128GB,  
16GBx8 DDR-4, 2400MT/s

### Disks

4x372GB,2.5",  
SAS,SSD,  
RAID 10 400GB  
+ RAID10 344GB, 12GBps

### Network

Intel 2P 10G X520  
Intel 2P 10G X520  
+2P 1G I350 rNDS

### Ceph

### Storage nodes

3xR730xd

Each node:

### Processors

2xIntel 12-Core E5-2650v4

### Memory

128 GB, 8x 16GB, DDR-4, 2400 MT/s

### Disks

13x2TB, 3.5", 7.2K, HDD,  
SAS, JBOD  
2x300GB, 2.5", 15K, HDD, SAS,  
RAID1 278GB  
3x200GB, 2.5" SSD

### Network

Intel 2P 10G X520  
Intel 2P 10G X520  
+ 2P 1G I350 rNDS

### Controllers

3xR630

### Processors

2xIntel 12-Core E5-2650v4

### Memory

128GB, 16GBx8 DDR-4, 2400MT/s

### Disks

4x600GB, 2.5", SAS, HDD, 10K,  
RAID 10 1.2TB

### Network

Intel 2P 10G X520  
Intel 2P 10G X520  
+ 2P 1G I350 rNDS

### Compute nodes

3xR630

Each node:

### Processors

2xIntel 12-Core E5-2650v4

### Memory

128 GB, 8x 16GB, DDR-4, 2400 MT/s

### Disks

8x600 GB, 2.5", 10K, HDD SAS  
RAID10 2.2TB

### Network

Intel 2P 10G X520  
Intel 2P 10G X520  
+ 2P 1G I350 rNDS

### Compute + Local Storage nodes

3xR730xd

Each node:

### Processors

2x Intel 12-Core E5-2690 v3

### Memory

128 GB, 8x 16GB, DDR-4, 2133 MT/s

### Disks

16x1TB, 2.5", 7.2K, HDD,  
SAS JBOD  
2x300 GB, 2.5", HDD, SAS  
RAID 1

### Network

Intel 2P 10G X520  
Intel 2P 10G X520  
+ 2P 1G I350 rNDS

Fig. 1. Dell Red Hat OpenStack Platform

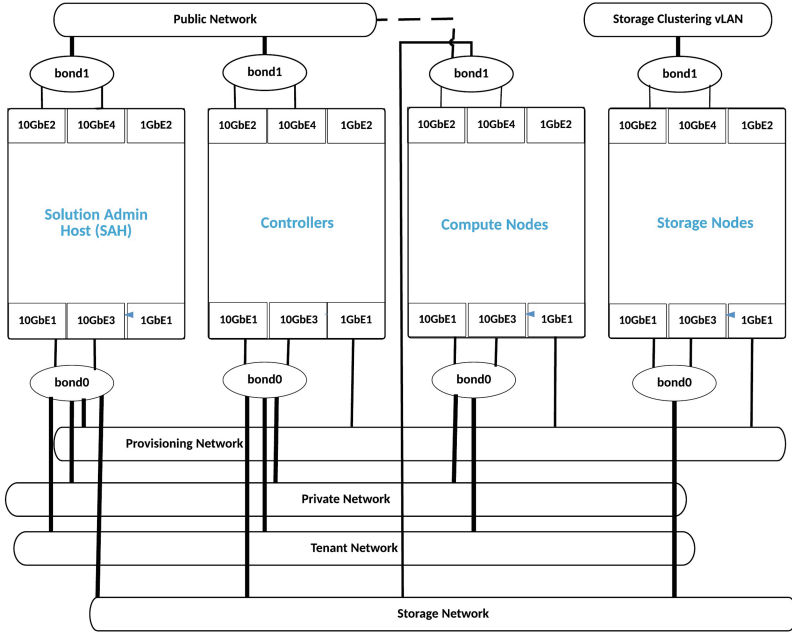


Fig. 2. Cloud architecture

### 3.1 OpenStack Sahara

OpenStack Sahara provides a robust interface to easily provision and scale Hadoop clusters. As an OpenStack component, OpenStack Sahara is fully integrated into the OpenStack ecosystem; for example, users can administer the entire Hadoop data processing workflow through the OpenStack dashboard (Horizon) – from configuring clusters, all the way to launching and running jobs on them [7].

A cluster deployed by Sahara consists of node groups. Node groups vary by their role, parameters and number of machines. In order to simplify cluster provisioning, Sahara makes use of two kinds of templates: node group templates and cluster templates. A cluster template is made up of multiple node group templates, while a node group template specifies the role and services that are needed in that group. The use of these templates reduces the cluster deployment and configuration time.

Sahara uses different plugins to provision a specific data processing framework or Hadoop distribution. There are several supported plugins for the OpenStack Kilo release including Vanilla, Cloudera, Ambari and Spark. We used the Cloudera (CDH) 5.3.0 plugin which allows the deployment and operation of a cluster with Cloudera Manager.

OpenStack deploys instances of the cluster based on a pre-built image with an installed OS. The image requirements for Sahara depend on the plugin and data processing framework version. With the Cloudera 5.3.0 plugin, we used the Centos 6.6 based image with preinstalled packages of CDH 5.3.0 [8].

### 3.2 Cloudera Manager 5.3

Cloudera Manager makes it easy to manage Hadoop deployments of any scale in production. Cloudera Manager 5.3 was used to configure and monitor the Hadoop clusters through its intuitive UI [9]. This functionality enabled us to easily configure the Hadoop clusters for testing different scenarios as discussed in Sect. 5: Performance Testing. Cloudera Manager UI helps with clusterwide monitoring of all hosts and services, cluster usage and health, and also with troubleshooting problems. The version summary of Hadoop components deployed is listed in Table 3.

**Table 3.** Cloudera Hadoop version summary

Component	Version	Release	CDH version
YARN	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
HDFS	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
hue-common	3.7.0+cdh5.3.0+134	1.cdh5.3.0.p0.24	CDH 5
Keytrustee Keyprovider	5.5.0+cdh5.5.0+0	1.cdh5.5.0.p0.1	Not applicable
hadoop-kms	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
HBase	0.98.6+cdh5.3.0+73	1.cdh5.3.0.p0.25	CDH 5
Hue	3.7.0+cdh5.3.0+134	1.cdh5.3.0.p0.24	CDH 5
Crunch (CDH 5 only)	0.11.0+cdh5.3.0+16	1.cdh5.3.0.p0.24	CDH 5
Llama (CDH 5 only)	1.0.0+cdh5.3.0+0	1.cdh5.3.0.p0.26	CDH 5
HttpFS	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
Hadoop	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
sentry	1.4.0+cdh5.3.0+126	1.cdh5.3.0.p0.26	CDH 5
MapReduce 2	2.5.0+cdh5.3.0+781	1.cdh5.3.0.p0.54	CDH 5
Lily HBase Indexer	1.5+cdh5.3.0+23	1.cdh5.3.0.p0.18	CDH 5
Flume NG	1.5.0+cdh5.3.0+79	1.cdh5.3.0.p0.18	CDH 5
Cloudera Manager Management Daemons	5.3.0	1.cm530.p0.166	Not applicable
Supervisord	3.0-cm5.3.0	Unavailable	Not applicable
Java 7	jdk1.7.0_67-cloudera	Unavailable	Not applicable
Cloudera Manager agent	5.3.0	1.cm530.p0.166	Not applicable

### 3.3 TPCx-HS

The results of this POC were derived from the TPCx-HS benchmark and as such are not comparable to published TPCx-HS results. TPCx-HS proved to be a viable option because of its focus on big data. TPCx-HS was developed to provide an objective measure of hardware, operating system and commercial Apache Hadoop File System API compatible software distributions, and to provide the industry with verifiable performance, price-performance and availability metrics [2]. Each run of the benchmark consisted of 2 iterations of HSGen, HSDataCheck, HSSort, HSValidate and provided us with the job run time.

## 4 Configurations

### 4.1 Hardware Configurations

Below are the hardware configurations that were used for different test scenarios discussed in Sect. 5. Table 4 shows the hardware configurations for Instance and Over-Subscription tests while Table 5 shows the hardware configurations for HDFS on local Storage, CPU Pinning/NUMA with HDFS on Local Storage and Disk and CPU Pinning/NUMA with HDFS on Local Storage tests. It should be noted that the hardware configurations used for this study are under the control of the Cloud and its administrator and are not visible to the user running the tests described in Sect. 5.

**Table 4.** Hardware configuration for instance and over-subscription tests

	Controller nodes	Compute nodes	Ceph storage nodes	
Server model	Dell R630	Dell R630	Dell R730xd	
CPU	Intel E5-2650 v4	Intel E5-2650 v4	Intel E5-2650 v4	
Memory	128 GB, 2400 MT/s	128 GB, 2400 MT/s	128 GB, 2400 MT/s	
BIOS version	2.0.1	2.0.1	2.0.1	
Firmware version	2.30.30.30	2.30.30.30	2.30.30.30	
HDD	4 × 600 GB, 2.5", SAS, HDD	8 × 600 GB, 2.5", SAS, HDD	2 × 300 GB, 2.5", SAS, HDD	3 × 200 GB, 2.5", SAS, SSD +13 × 2 TB, 3.5", SAS, HDD
HDD configuration	H730 RAID 10 with drives 0, 1, 2, 3;	H730 RAID 10 with drives 0, 1, 2, 3, 4, 5, 6, 7;	H730 RAID 1 with flex bay drives 12, 13;	JBOD drives 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17
Read policy	Adaptive read ahead	Adaptive read ahead	Adaptive read ahead	—
Write policy	Write back	Write back	Write back	—
Disk cache policy	Default	Default	Default	—

**Table 5.** H/W configuration for HDFS on local storage, CPU & Disk pinning tests

	Controller nodes	Compute with local storage nodes	
Server model	Dell R630	Dell R730xd	
CPU	Intel E5-2650 v4	Intel E5-2690 v3	
Memory	128 GB, 2400 MT/s	128 GB, 2133 MT/s	
BIOS version	2.0.1	2.0.1	
Firmware version	2.30.30.30	2.30.30.30	
HDD	4 × 600 GB, 2.5", SAS, HDD	2 × 300 GB, 2.5", SAS, HDD	16 × 1 TB, 2.5"n, SAS, HDD
HDD configuration	H730 RAID 10 with drives 0, 1, 2, 3;	H730 RAID 1 with flex bay drives 24, 25;	JBOD drives 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Read policy	Adaptive read ahead	Adaptive read ahead	–
Write policy	Write back	Write back	–
Disk cache policy	Default	Default	–

## 4.2 Hadoop Configurations

In order to implement different test scenarios, several of the Hadoop configurations were varied. Some of the configurations that remained common for all the tests are listed in Table 6 while the configurations that varied from test to test to find the optimum performance are listed in Table 7.

**Table 6.** Common Hadoop configurations

S. No.	Configuration name	Value
1	dfs.replication	3
2	dfs.blocksize	512 MB
3	mapreduce.map.cpu.vcores	1
4	mapreduce.map.memory.mb	1024 MB
5	mapreduce.reduce.cpu.vcores	1
6	mapreduce.reduce.memory.mb	2048 MB
7	yarn.scheduler.minimum-allocation-vcores	1
8	yarn.scheduler.minimum-allocation-mb	1024 MB
9	yarn.scheduler.increment-allocation-vcores	1
10	yarn.scheduler.increment-allocation-mb	512 MB
11	yarn.app.mapreduce.am.resource.mb	2048 MB
12	mapreduce.map.sort.spill.percent	0.8
13	mapreduce.task.io.sort.mb	256 MB
14	mapreduce.job.reduce.slowstart.completedmaps	0.8

**Table 7.** Variable Hadoop configurations

S. No.	Configuration name	Range
1	yarn.nodemanager.resource.memory-mb	4 GB–96 GB
2	yarn.nodemanager.resource.cpu-vcores	2–40
3	yarn.scheduler.maximum-allocation-mb	4 GB–96 GB
4	yarn.scheduler.maximum-allocation-vcores	2–40

## 5 Performance Testing

Based on prior research and virtualization experience [10], a few OpenStack configurations that were known to have a high impact on performance were selected for performance testing. These included over-subscription, use of local storage on Compute nodes, NUMA nodes and Disk Pinning. Test cases were developed to run and measure the performance of TPCx-HS workloads on these configurations.

### 5.1 Instance Configuration Tests

The goal of the instance tests was to understand the impact of Instance (VM) configurations on TPCx-HS performance.

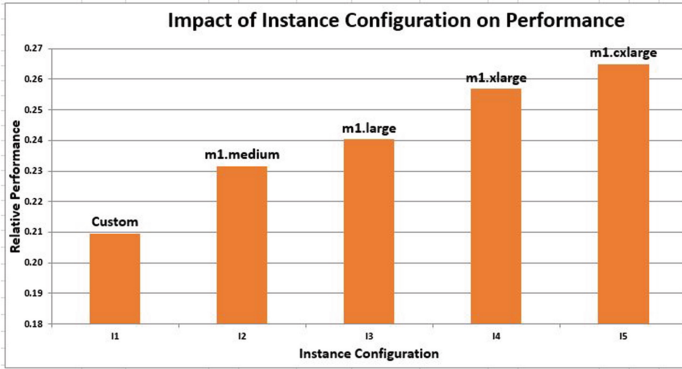
First, the number of workers versus the resources allocated to them was tested. In essence, it was necessary to determine whether a small number of instances, each with a higher resource allocation, would work better than a greater number of worker instances with relatively fewer resources allocated to each of them. In order to test the above scenarios, an arrangement (I1...I5) of vCPU, memory and number of instances was specified as shown in Table 8. TPCx-HS tests were executed on Hadoop clusters based on each of the specified instance configurations.

**Table 8.** Instance configuration tests

Test ID	Nova flavor	Worker per Node/Total	vCPU per Worker	Memory per Worker	Storage per Worker	Storage type
I1	custom	1/3	40	96 GB	16 TB	Ceph
I2	m1. medium	20/60	2	4 GB	1 TB	Ceph
I3	m1. large	10/30	4	8 GB	2 TB	Ceph
I4	m1. xlarge	5/15	8	16 GB	4 TB	Ceph
I5	m1. cxlarge	4/12	10	20 GB	4 TB	Ceph

These clusters were deployed using OpenStack Sahara. Note that m1.cxlarge was a custom flavor. These tests were run without resource over-subscription. In all tables, unless otherwise stated, Node refers to Compute Node.

While the Bare-metal hardware configuration shown in Table 1 was different from the cloud virtual machine configuration, its performance served as a datum point for comparison of results. The relative performance on the y-axis of Figs. 3 and 4 is performance compared to Bare-metal Hadoop performance datum. For the Instance configuration iterative test it was found that instance configuration I5, as shown in Table 8, provided best performance. Figure 3 shows that large-sized flavor configurations perform better. The only exception is the single-instance custom flavor which was configured with the largest size but performed poorest.



**Fig. 3.** Performance per instance configuration

Secondly the number of Instances per node were varied from 1 to 20 while ensuring that the number of vCPUs and YARN containers created remained constant. It was observed that performance peaks at 4 instances per node and then it tapers downwards. The result for this can be seen in Fig. 4 which shows performance with maximum number of instances for each flavor type.

The results of the instance configuration tests show that provisioning 4 m1.cxlarge configuration instances and allocating them the maximum amount of memory and vCPU resources provides the best performance. From these results, the TPCx-HS performance gain due to an optimal instance configuration can go up to 5%. The optimal instance configuration (I5) determined in this test is used in subsequent tests.

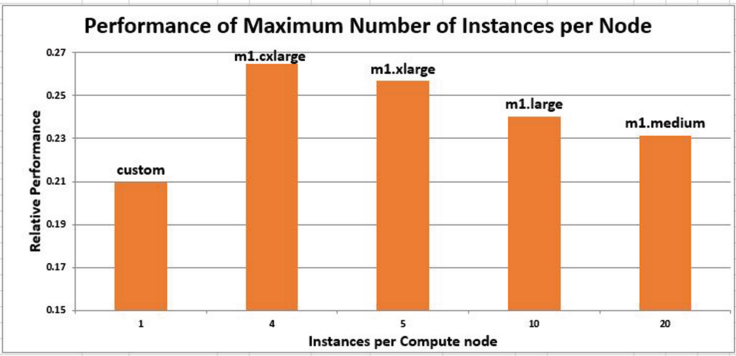


Fig. 4. Performance per number of instances

5.2 Over-Subscription Tests

In this section, the impact of resource over-subscription on aggregate performance was determined. Over-subscription was controlled by a Cloud administrator through the choice of Nova flavor.

**CPU Over-Subscription Tests.** In these tests, the number of instances and memory per instance was fixed, while the vCPUs assigned to each instance were increased according to the ratio shown in Table 9. For the test ID C1, the optimal configuration obtained in Sect. 5.1, I5, was used. The iterative tests for the vCPU over-subscription can be found below in Table 9.

Table 9. CPU over-subscription tests

Test ID	Ratio	Nova flavor	Worker per Node/Total	vCPU per Worker	Memory per Worker	Storage per Worker	Storage type
C1	1:1	custom	4/12	10	20 GB	4 TB	Ceph
C2	1:2	custom	4/12	20	20 GB	4 TB	Ceph
C3	1:3	custom	4/12	30	20 GB	4 TB	Ceph
C4	1:4	custom	4/12	40	20 GB	4 TB	Ceph

The number of worker instances and memory per instance values were fixed while the vCPUs per instance were increased to find optimal performance as shown in Table 9. The performance of each test was normalized by the performance of Test ID “C1”. 1:1 CPU subscription yielded the best performance and Fig. 5 shows that while there is a performance cost with over-subscription, it is not so drastic. An over-subscription ratio of 1:2 results in a 2.5% drop in performance and a 1:4 results in a 23% drop.

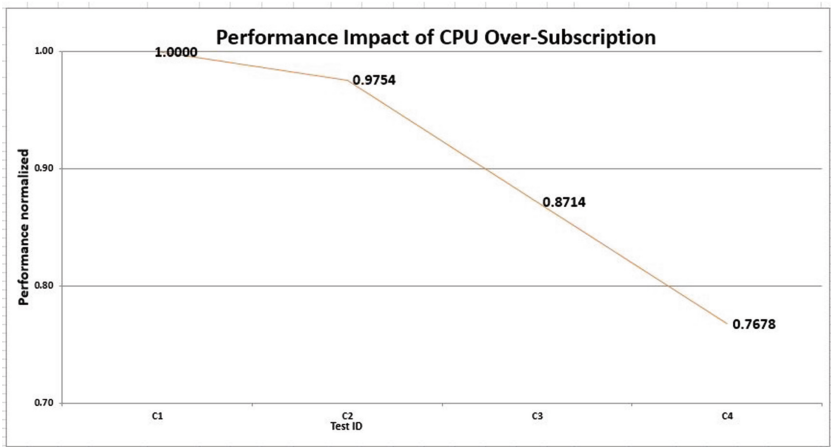


Fig. 5. CPU over-subscription

**Memory Over-Subscription Tests.** In these tests, the optimal arrangement of instances and vCPUs from the CPU over-subscription tests was used, while the memory per worker was increased according to the ratio shown in Table 10.

Table 10. Memory over-subscription tests

Test ID	Ratio	Nova flavor	Worker per Node/Total	vCPU per Worker	Memory per Worker	Storage per Worker	Storage type
M1	1:1	custom	4/12	10	20 GB	4 TB	Ceph
M1.1	1:1.1	custom	4/12	10	22 GB	4 TB	Ceph
M1.2	1:1.2	custom	4/12	10	24 GB	4 TB	Ceph
M1.3	1:1.3	custom	4/12	10	26 GB	4 TB	Ceph

The performance of each test was normalized by the performance of Test ID “M1” that has no over-subscription. As the ratio of vMem (virtual) to pMem (physical) was raised by 10% through 30%, Fig. 6 shows that there was a drastic drop in performance with memory over-subscription. A 10% memory over-subscription results in a 65% drop in performance while a 30% over-subscription results in a 70% drop. This test demonstrates that memory over-subscription should be avoided if performance is a consideration. Based on over-subscription tests, it was determined that memory over-subscription has a bigger impact on performance than CPU over-subscription.

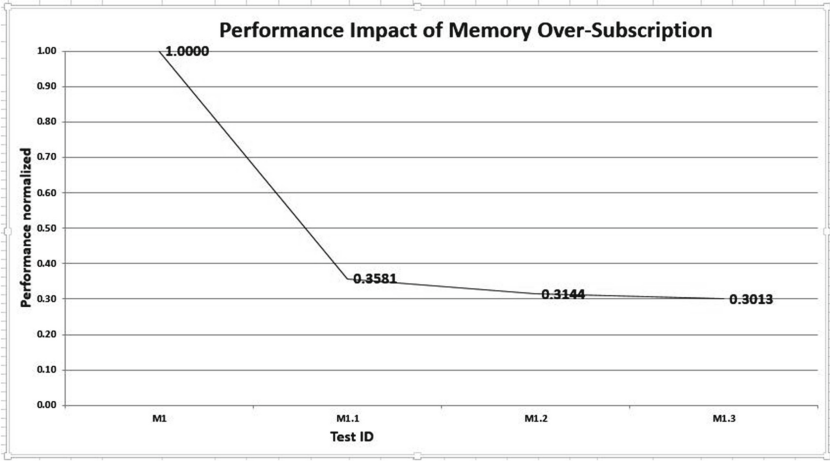


Fig. 6. Memory over-subscription

### 5.3 HDFS on Local Storage Tests

Some phases of the TPCx-HS workloads particularly HSGen (Teragen) are IOintensive. It was therefore important to use a storage configuration that would provide better performance. In that respect, tests were undertaken to compare HDFS performance on local storage to Ceph shared backend.

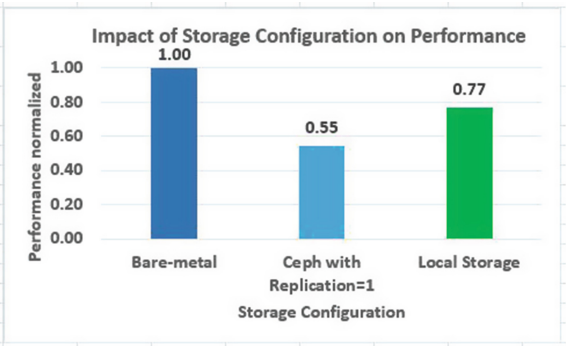
For HDFS on local storage tests, 3 Dell R730xd servers each with  $16 \times 1$  TB SAS data drives were added to the OpenStack Cloud. See Fig. 1 and Table 5 for the detailed hardware configuration. Notice that for these Local Storage tests the same hardware was used for Bare-metal and Cloud runs, thus it is a true comparison between Hadoop on Bare-metal and Hadoop on OpenStack cloud performance. Each physical node was configured as both OpenStack Compute and Cinder Volume node. On each server, all 16 SAS drives were added to a single volume group and then OpenStack Cinder was configured to use that volume group with LVM iSCSI Volume Driver [11]. Each node was configured as a separate availability zone to ensure that volume attachment comes from the same zone and the cinder volumes are local to a Compute node.

A cluster of 12 worker instances (4 on each R730xd) was deployed and each instance was attached to  $4 \times 1$  TB Cinder volumes for HDFS provided by the local volume server. Similarly, for Ceph-based configuration the same number (4) and size (1 TB) of volumes were attached to each Instance. HDFS replication was set at 3 for all the tests while Ceph replication was set at 1 to maintain the total replication factor of 3. See Table 11 below for test configurations.

**Table 11.** HDFS on local storage tests

Test ID	Ceph replicas	HDFS replication	Nova flavor	Nova flavor	Nova flavor	Nova flavor	Nova flavor
Ceph	1	3	custom	4/12	10	20 GB	4 TB
Local	–	3	custom	4/12	10	20 GB	4 TB

The local storage arrangement resulted in better performance than Ceph shared storage backend. Note that as shown in Fig. 1, the Compute hardware configuration used for Ceph storage tests was of a newer generation. Figure 7 below shows a significant performance improvement of 22% over Ceph shared storage with replication = 1 in spite of the newer Compute hardware used for Ceph Storage. The use of local storage minimizes network traffic and improves performance in an IO-bound environment. It should be noted that Ceph Storage has a lot of other advantages like resiliency that might out-weigh performance considerations.



**Fig. 7.** HDFS on local storage

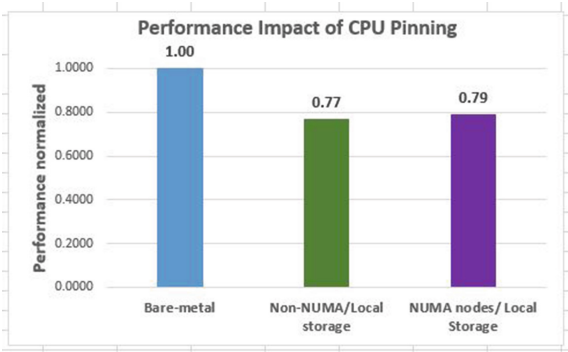
**5.4 CPU Pinning/NUMA with HDFS on Local Storage Tests**

The goal of this test is to understand the impact of NUMA awareness of the OpenStack scheduler on performance. To test the performance of using dedicated vCPUs with NUMA awareness, the 3 Compute Nodes (R730xds) in Sect. 5.3 above were configured to support the pinning of virtual machine instances to dedicated physical cores [12]. A cluster of 12 worker instances (4 on each R730xd) was then deployed where the vCPUs of each instance pins and exclusively use the physical CPUs. Each instance was attached to 4 × 1 TB Cinder volumes provided by the local volume server. See Table 12 below for the test configurations. Test ID “Non-NUMA” in Table 12 used the same configuration as HDFS on Local Storage described in Sect. 5.3.

**Table 12.** CPU pinning/NUMA with HDFS on local storage tests

Test ID	Nova Flavor	Worker per Node/Total	vCPU per Worker	Memory per Worker	Storage per Worker	Storage type
Non-NUMA	custom	4/12	10	20 GB	4 TB	Local
NUMA	custom	4/12	10	20 GB	4 TB	Local

An additional 2% performance improvement was observed by implementing CPU pinning in a configuration with HDFS on Local storage as shown in Fig. 8. The ability for the OpenStack scheduler to be aware of the underlying NUMA architecture typically optimizes the performance of individual Instances. In this test, processor affinity (CPU pinning) did not have a significant impact on performance. This could be attributed to the effects of the KVM hypervisor and should be a subject of further investigation.



**Fig. 8.** CPU pinning/NUMA with HDFS on local storage

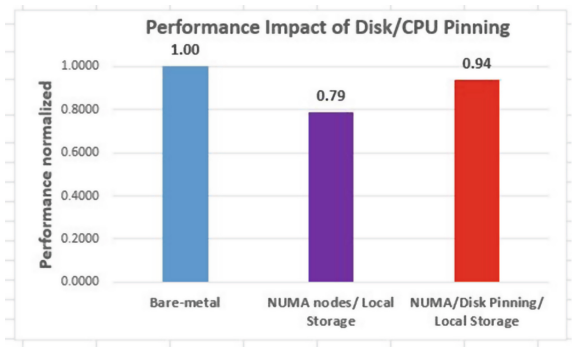
### 5.5 Disk and CPU Pinning/NUMA with HDFS on Local Storage Tests

The goal of this test is to understand the impact of disk pinning on performance. To test the performance of using dedicated disks as compared to shared local disks, 3 Dell R730xdS were configured to implement the pinning of physical disks to the virtual machine instances. Each server had 16 × 1 TB SAS data drives. 16 separate volume groups were created and one physical disk was assigned to each volume group. OpenStack Cinder was configured to use all of those volume groups, each as an individual storage backend with LVM iSCSI Volume Driver. A cluster of 12 worker instances (4 on each R730xd) were deployed and each instance was attached to 4 × 1 TB Cinder volumes provided by one of the volume groups from the local volume server [13]. Additionally the vCPUs of each instance were pinned to the physical CPUs. See Table 13 below for the tests configurations. Test ID “NUMA” used the same configuration of “NUMA” as described in Sect. 5.4.

**Table 13.** Disk and CPU pinning/NUMA with HDFS on local storage tests

Test ID	Nova flavor	Worker per Node/Total	vCPU per Worker	Memory per Worker	Storage per Worker	Storage type
NUMA	custom	4/12	10	20 GB	4 TB	Local
NUMA & Disk	custom	4/12	10	20 GB	4 TB	Local

Figure 9 below shows that a performance improvement of 15% is attributed to disk pinning. TPCx-HS workloads are IO-bound during HSGen (Teragen) and shuffle phase of HSSort (Terasort). In an IO-intensive environment, the ability by instances to access and pin physical disks directly has a significant performance impact as this test has shown. Further performance gains can be achieved by use of a raw device driver instead of LVM. This test also shows that by implementing disk pinning in a configuration that uses NUMA nodes with HDFS on local storage, TPCx-HS performance on the OpenStack cloud almost matches that on bare metal.



**Fig. 9.** Disk and CPU pinning/NUMA with HDFS on local storage

## 6 Conclusions

Hadoop-on-Cloud POC has shown that it is possible for the performance of Big Data workloads (like TPCx-HS on OpenStack) on the Cloud to match that on Bare-Metal. This improvement was achieved by using an optimal Instance configuration that was deployed on local storage and with the implementation of CPU and disk pinning. More performance gains can be realized by implementing the use of a raw device driver by Cinder instead of LVM used in this study. The net effect of the aggregation of optimizations shown in this paper and those that have been recommended should lead to better TPCx-HS performance on the Cloud than on Bare-metal. That has been shown to be possible in virtualized environments [10] and from the results of this POC it should be possible on the Cloud. Follow-up tests to this POC will strive to identify even more

optimizations. It is worth noting that in this paper, our recommendations for OpenStack configuration and hardware choices were considered from a performance perspective. In a production environment, Openstack and Hadoop data protection best-practices should be considered. This includes use of persistent storage and raising the HDFS replication factor to greater than the default (>3).

**Acknowledgments.** The authors would like to thank John Terpstra, Michael Pittaro, Randy Perryman, Michael Tondee and David Grimes for participating in the technical review meetings of the POC. Their input, feedback and guidance helped shape this investigation. Mr. Ashok Malani is recognized for his technical leadership of the xFlow Research team that did such a tremendous job performing the tests and drafting this paper.

## References

1. Navint: Why is big data important? (2012). [www.navint.com/images/Navint.BigData.FINAL.pdf](http://www.navint.com/images/Navint.BigData.FINAL.pdf)
2. TPC: Tpcx-hs (2016). <http://www.tpc.org/tpcx-hs/>
3. VMware: Virtualized hadoop performance with vmware vsphere 6 on highperformance servers (2015). <http://www.vmware.com/files/pdf/techpaper/Virtualized-Hadoop-Performance-with-VMware-vSphere6.pdf>
4. Stata, R.: Understanding hadoop-as-a-service offerings (2014). <http://www.datacenterknowledge.com/archives/2014/05/14/understanding-hadoop-service-offerings/>
5. Hurtgen, A.: Using apache hadoop on rackspace private cloud (2013). <https://support.rackspace.com/how-to/apache-hadoop-on-rackspace-private-cloud/>
6. Wendt, M.E.: Cloud-based hadoop deployments: benefits and considerations (2014). <https://goo.gl/re0Ov5>
7. OpenStack: Openstack sahara user documentation (2016). <http://docs.openstack.org/developer/sahara/userdoc/overview.html>
8. Mirantis: Openstack sahara kilo images (2016). <http://sahara-files.mirantis.com/images/upstream/kilo/>
9. Cloudera, I.: Cloudera manager free edition user guide (2012)
10. TPC: Dell poweredge r720xd with vmware vsphere 6.0 (2015). <http://www.tpc.org/5504>
11. OpenStack: Install and configure a storage node - openstack kilo (2015). <http://docs.openstack.org/kilo/install-guide/install/yum/content/cinder-install-storage-node.html>
12. RedHat: Cpu pinning and numa topology awareness in openstack compute (2015). <http://redhatstackblog.redhat.com/2015/05/05/cpu-pinning-and-numa-topology-awareness-in-openstack-compute/>
13. OpenStack: Openstack cinder multi-backend (2015). <https://wiki.openstack.org/wiki/Cinder-multi-backend>

Performance Evaluation and Benchmarking. Traditional  
- Big Data - Internet of Things  
8th TPC Technology Conference, TPCTC 2016, New  
Delhi, India, September 5-9, 2016, Revised Selected  
Papers

Nambiar, R.; Poess, M. (Eds.)

2017, XIII, 161 p. 55 illus., Softcover

ISBN: 978-3-319-54333-8