TPCx-HS on the Cloud!

Running Hadoop on OpenStack Cloud and

Tuning to Get Bare-Metal Performance

Michael Woodside, Sr. Engineering Manager

Big Data / Analytics, Open Source Solutions, Dell

C_Michael_Woodside@Dell.com

Outline

- Motivation for Work
- Methodology and Experiment Design
- Tests and Results
- Conclusion

Motivation

Motivation

- Dell builds solutions optimized for both Big Data/Analytics and for OpenStack Cloud
- The primary motivation was to understand and quantify Hadoop performance on the OpenStack cloud implementation for purposes of developing configurations optimized for Hadoop workloads and guiding, advising and providing necessary data-based information for achieving near bare-metal performance
- Document the cloud configurations and optimizations, and the hardware options that were used to achieve this performance
- Use a standards-based tool for evaluating the Hadoop performance on the OpenStack cloud - TPCx-HS

Methodology

Methodology

- Use bare-metal tests as our baseline for normalization
- Use TPCx-HS benchmark and tools as our workload and basis for comparison
- Define the configuration parameters to evaluate and adjust
  1. I – worker Instance configuration (number of instances, vCPU, RAM, storage per node)
  2. C – ratio of CPU over-subscription
  3. M – ratio of virtual to physical Memory over-subscription
  4. Ceph vs Local storage
  5. CPU Pinning/NUMA
  6. Disk Pinning
- Define initial instance configuration of 1 virtual instance per physical node and allocate all vCPU and memory (custom data point as initial configuration)
- Interate through the set of values of each parameter and test
- Use the value of each parameter that showed the best performance for each subsequent test
- Conclude with the best tested value for every parameter

Tests and Results

Instance Configuration Test

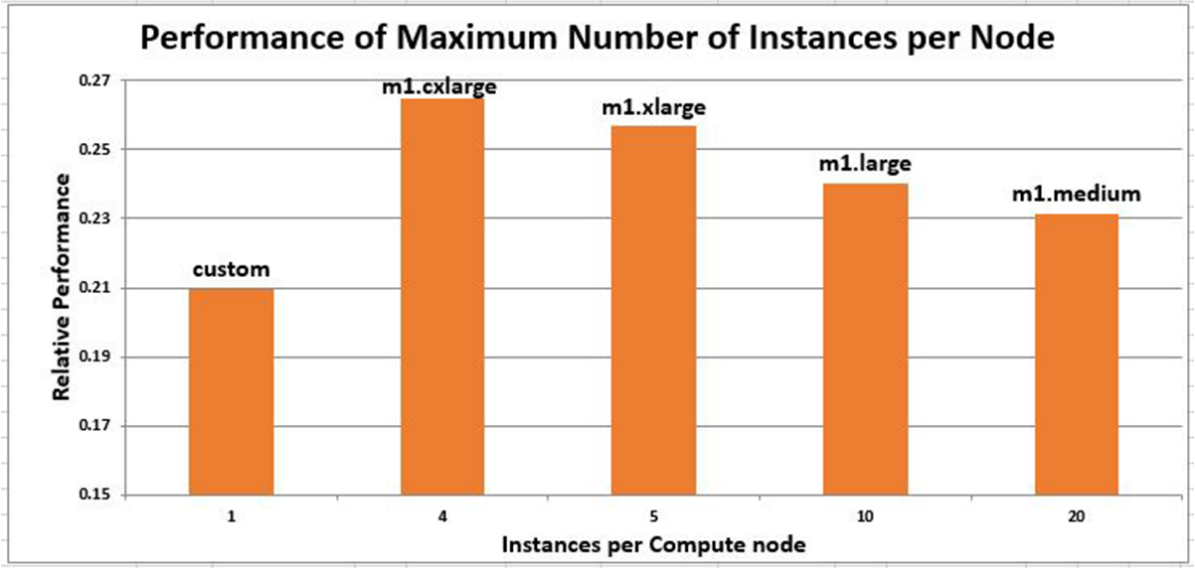
- Total vCPU was Fixed and evenly distributed between all worker instances
- Total Memory was Fixed and evenly distributed between all worker instances
- Number of Worker Instances per physical node was varied

Table 8: Instance Iteration 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

Instance Configuration - Results

- The range of results between the min. and max. workers per node was only ~5%
- The best worker instance configuration is (I5) 4 worker instances per physical node, with a relative performance increase of 5%



CPU Over-Subscription Test - Configuration

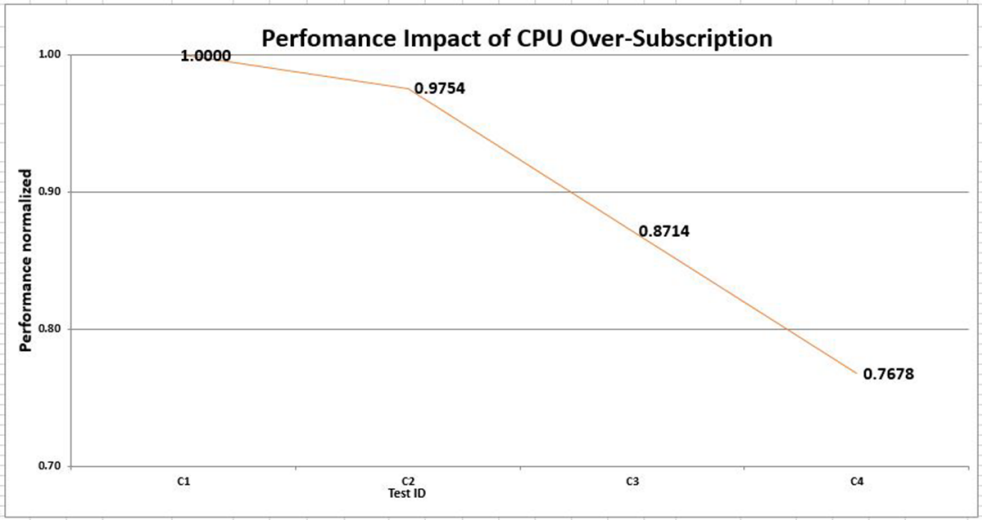
- *Worker Instance configuration was Fixed by I5 (number of instances, vCPU, RAM, storage)*
- CPU over-subscription per Worker Instance was varied (1x up to 4x)

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

CPU Over-Subscription Test - Results

- The best CPU over-subscription is (C1) none – No CPU Over-Subscription
- A 2x CPU over-subscription caused a performance degradation of ~2%
- A 3x CPU over-subscription caused a performance degradation of ~13%



Memory Over-Subscription Test - Configuration

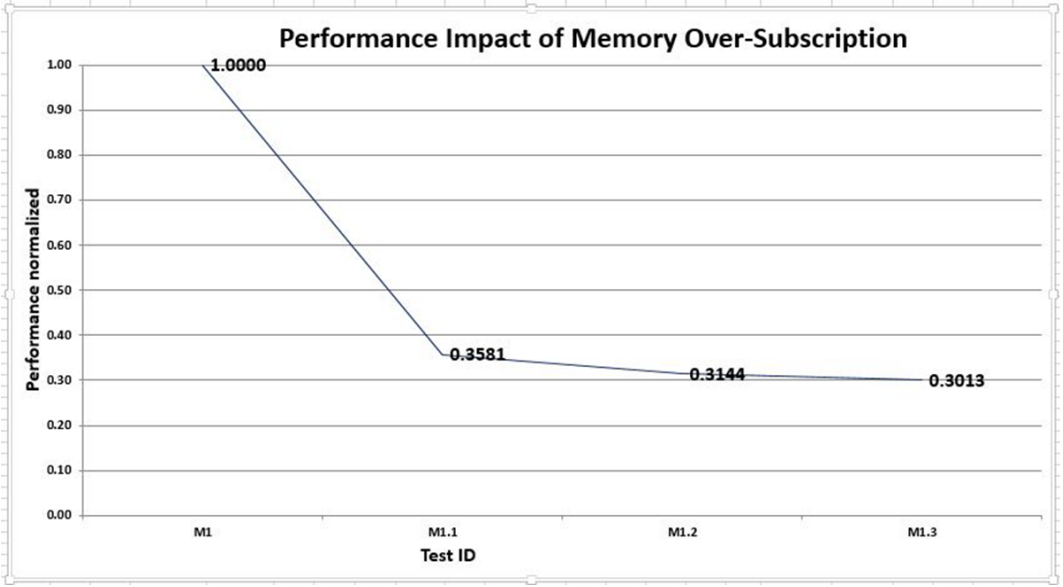
- *Worker Instance configuration was Fixed by I5 (number of instances, RAM, vCPU, storage)*
- *CPU over-subscription per Worker Instance was Fixed at 1:1 (C1)*
- Memory over-subscription per Worker Instance was varied (0% – 30%)

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

Memory Over-Subscription Test - Results

- The best Memory over-subscription is (M1) none – No Memory Over-Subscription
- A 10% Memory over-subscription caused a performance degradation of ~64%



HDFS on Ceph vs. Local Storage Test - Configuration

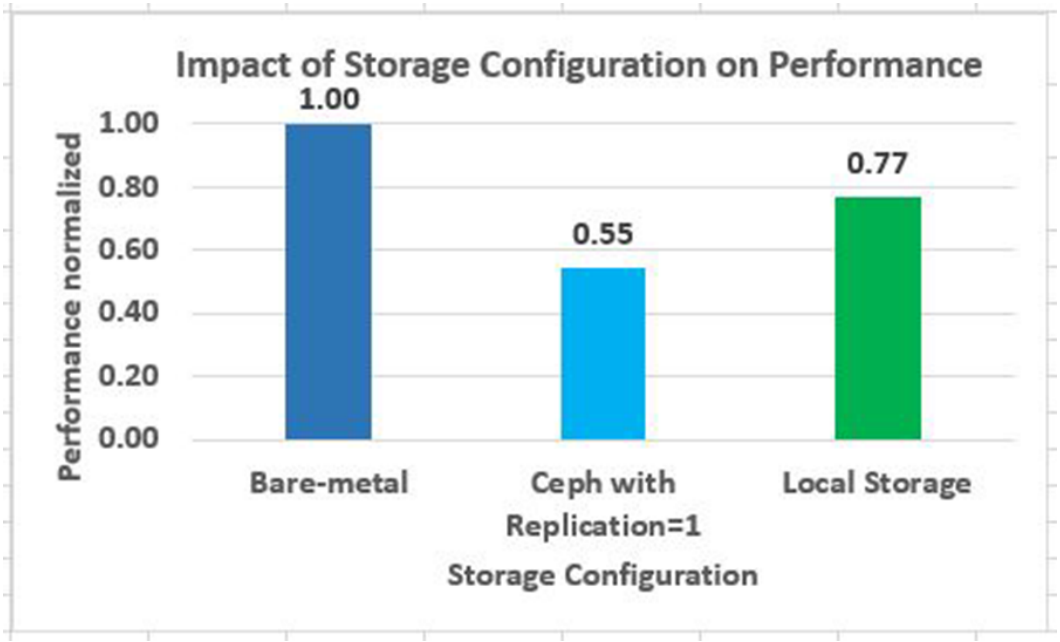
- The remaining tests used the same hardware that was used for Bare-metal runs, thus it is a true comparison between Hadoop on Bare-metal and on OpenStack cloud performance
- *Worker Instance configuration was Fixed by I5 (number of instances, RAM, vCPU, storage)*
- *CPU over-subscription per Worker Instance was Fixed at 1:1 (C1)*
- *Memory over-subscription per Worker Instance was Fixed at 1:1 (M1)*
- Ceph storage with replication=1, HDFS replication=3, switched to Local storage with HDFS replication=3

Table 11: HDFS on Local Storage Tests

Test ID	Ceph Replicas	HDFS Replication	Nova Flavor	Worker per Node/ Total	vCPU per Worker	Memory per Worker	Storage per Worker
Ceph	1	3	custom	4 / 12	10	20 GB	4 TB
Local	-	3	custom	4 / 12	10	20 GB	4 TB

HDFS on Ceph vs. Local Storage Test - Results

- HDFS on Local Storage improved performance by 22%



CPU Pinning/NUMA Test - Configuration

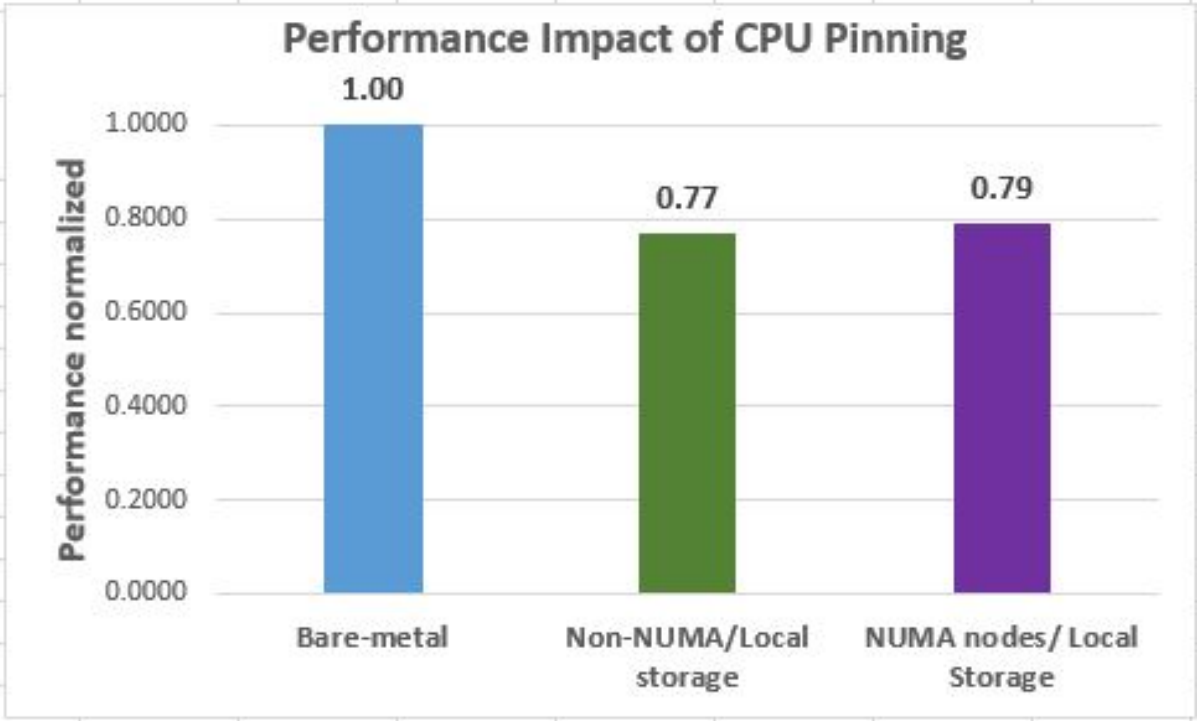
- *Worker Instance configuration was Fixed by I5 (number of instances, RAM, vCPU, storage)*
- *CPU over-subscription per Worker Instance was Fixed at 1:1 (C1)*
- *Memory over-subscription per Worker Instance was Fixed at 1:1 (M1)*
- *Local storage with HDFS replication=3*
- CPU pinning was applied

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	Storge Type
Non-NUMA	custom	4 / 12	10	20 GB	4 TB	Local
NUMA	custom	4 / 12	10	20 GB	4 TB	Local

CPU Pinning/NUMA Test - Results

- CPU Pinning improved performance by 2%



Disk Pinning Test - Configuration

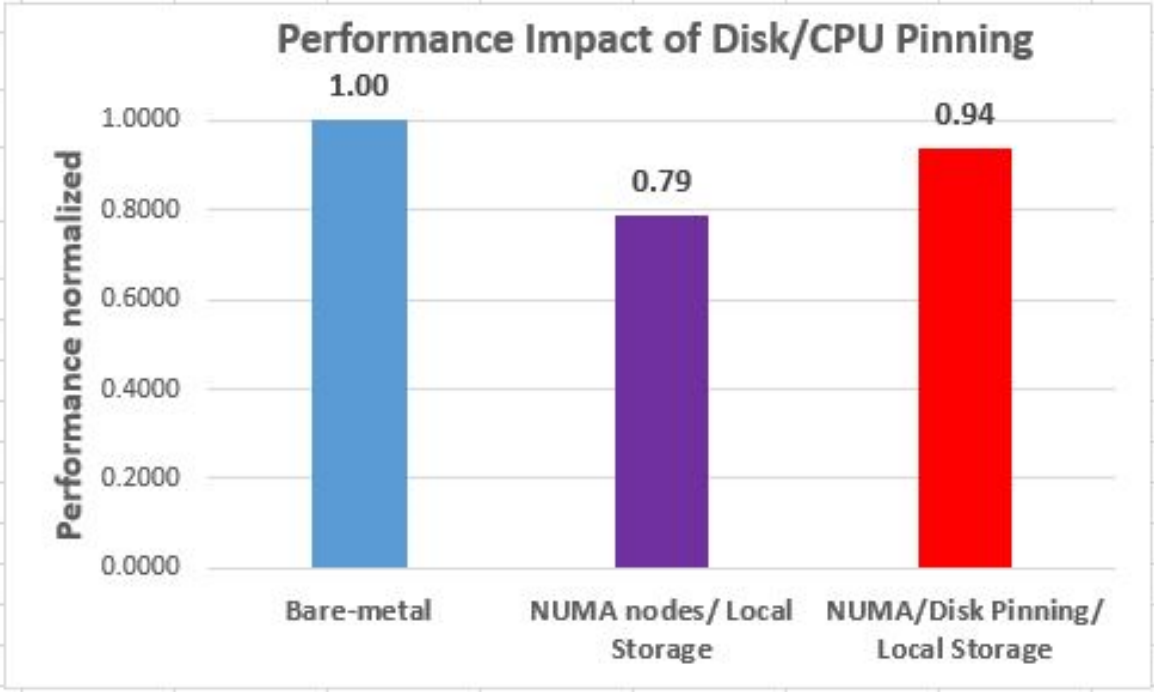
- *Worker Instance configuration was Fixed by I5 (number of instances, RAM, vCPU, storage)*
- *CPU over-subscription per Worker Instance was Fixed at 1:1 (C1)*
- *Memory over-subscription per Worker Instance was Fixed at 1:1 (M1)*
- *Local storage with HDFS replication=3*
- *CPU pinning*
- Disk pinning was applied

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

Disk Pinning Test - Results

- Disk Pinning improved performance by 15%

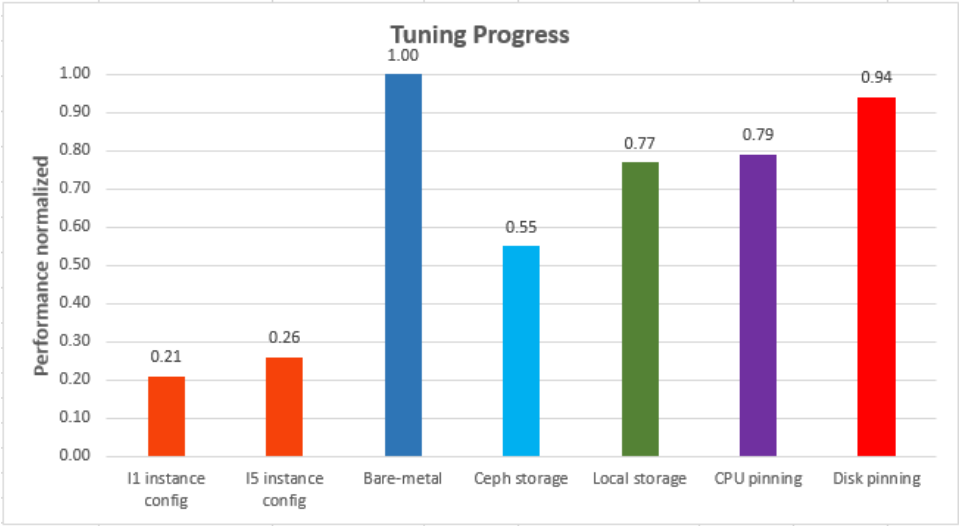


Conclusion

Conclusion

- Worker Instance configuration was Fixed by I5 (number of instances, RAM, vCPU, storage)
- CPU over-subscription per Worker Instance was Fixed at 1:1 (C1)
- Memory over-subscription per Worker Instance was Fixed at 1:1 (M1)
- Local storage with HDFS replication=3
- CPU pinning/NUMA was applied
- Disk pinning was applied

This configuration gives 94% performance of Bare-metal using TPCx-HS as the workload!



The power to do more