Hadoop Distributions: Evaluating Cloudera, Hortonworks, and MapR in Micro-benchmarks and Real-world Applications

Vladimir Starostenkov, Senior R&D Developer,
Kirill Grigorchuk, Head of R&D Department

Table of Contents

1. Introduction 4

2. Tools, Libraries, and Methods 5

2.1 Micro benchmarks 6

2.1.1 WordCount 6

2.1.2 Sort 7

2.1.3 TeraSort 7

2.1.4 Distributed File System I/O 7

2.2 Real-world applications 7

2.2.1 PageRank 8

2.2.2 Bayes 8

3. What Makes This Research Unique? 9

3.1 Testing environment 9

4. Results 11

4.1 Overall cluster performance 11

4.2 Hortonworks Data Platform (HDP) 12

4.3 Cloudera's Distribution Including Apache Hadoop (CDH) 14

4.4 MapR 15

5. Conclusion 18

Appendix A: Main Features and Their Comparison Across Distributions 19

Appendix B: Overview of the Distributions 21

1. MapR 21

2. Cloudera 22

3. Hortonworks 23

Appendix C: Performance Results for Each Benchmarking Test 24

1. Real-world applications 24

1.1 Bayes 24

1.2 PageRank 25

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

2. Micro benchmarks 26

2.1 Distributed File System I/O (DFSIO) 26

2.2 Hive aggregation 27

2.3 Sort 28

2.4 TeraSort 29

2.5 WordCount 30

Appendix D: Performance Results for Each Test Sectioned by Distribution 32

1. MapR 32

2. Hortonworks 42

3. Cloudera 52

Appendix E: Disk Benchmarking 62

1. DFSIO (read) benchmark 62

2. DFSIO (write) benchmark 63

Appendix F: Parameters used to optimize Hadoop Jobs 64

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

1. Introduction

Perhaps, there is hardly any expert in the big data field who has heard nothing about Hadoop. Furthermore, very often Hadoop is used as a synonym to the term big data. It is most likely, that the wide usage and popularity of this framework may have given a jump-start to development of various distributions that derived from the initial open-source edition.

The MapReduce paradigm was firstly introduced by Google and Yahoo continued with development of Hadoop, which is based on this data processing method. From that moment on, Hadoop has grown into several major distributions and dozens of sub-projects used by thousands of companies. However, the rapid development of the Hadoop ecosystem and the extension of its application area, lead to a misconception that Hadoop can be used to solve any high-load computing task easily, which is not exactly true.

Actually, when a company is considering Hadoop to address its needs, it has to answer two questions:

- Is Hadoop the right tool for me?
- Which distribution can be more suitable for my tasks?

To collect information on these two points, companies spend an enormous amount of time researching into distributed computing paradigms and projects, data formats and their optimization methods, etc. This benchmark demonstrates performance results of the most popular open-source Hadoop distributions, such as Cloudera, MapR, and Hortonworks. It also provides all the information you may need to evaluate these options.

In this research, such solutions as Amazon Elastic MapReduce (Amazon EMR), Windows Azure HDInsight, etc., are not analyzed, since they require uploading business data to public clouds. This benchmark evaluates only stand-alone distributions that can be installed in private data centers.

2. Tools, Libraries, and Methods

Every Hadoop distribution selected for this research can be tried as a demo on a virtual machine. This can help you to learn specific features of each solution and test how it works.

It can be quite a good idea for a proof of concept stage to take a set of real data and run Hadoop on several virtual machines in a cloud. Although, it will not help you to choose a configuration for your bare metal cluster *(for details, see "Hadoop Operations" by Eric Sammer, a chapter "Planning a Hadoop Cluster")*, you will be able to evaluate whether Hadoop is a good tool for your system. The paper "Nobody ever got fired for using Hadoop on a cluster" by Microsoft Research provides instructions for choosing an optimal hardware configuration.

On the whole, evaluating performance of a Hadoop cluster is challenging, since the results will vary depending on the cluster size and configuration. There are two main methods of Hadoop benchmarking: micro-benchmarks and emulated loads. Micro-benchmarks are shipped with most distributions and allow for testing particular parts of the infrastructure, for instance TestDFSIO analyzes the disk system, Sort evaluates MapReduce tasks, WordCount measures cluster performance, etc. The second approach provides for testing the system under workloads similar to those in real-life use cases. SWIM and GridMix3 consist of workloads that have been emulated based on historical data collected from a real cluster in operation. Executing the workloads via replaying the synthesized traces may help to evaluate the side effects of concurrent job execution in Hadoop.

HiBench, a Hadoop Benchmark Suite by Intel, consists of several Hadoop workloads, including both synthetic micro-benchmarks and real-world applications. All workloads included into this suite are grouped into four categories: micro-benchmarks, Web search, machine learning, and analytical queries. To find out about the workload used in this benchmark, read the paper "The HiBench Benchmark Suite: Characterization of the MapReduce-Based Data Analysis."

We monitored CPU, disk, RAM, network, and JVM parameters with Ganglia Monitoring System and tuned the parameters of each job (see Appendix A) to achieve maximum utilization of all resources.

Figure 1 demonstrates how data is processed inside a Hadoop cluster. Combiner is an optional component that reduces the size of data output by a node that executes a Map task.

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

To find out more about MapReduce internals, read *"Hadoop: The Definitive Guide"* by Tom White.

To show how the amount of data changes on each stage of a MapReduce job, the whole amount of input data was taken as 1.00. All the other indicators were calculated as ratios to the input amount. For instance, the input data set was 100 GB (1.00) in size. After a Map task had been completed, it increased to 142 GB (1.42), see Table 1. Using ratios instead of the real data amounts allows for analyzing trends. In addition, these results can help to predict the behavior of a cluster that deals with input data of a different size.

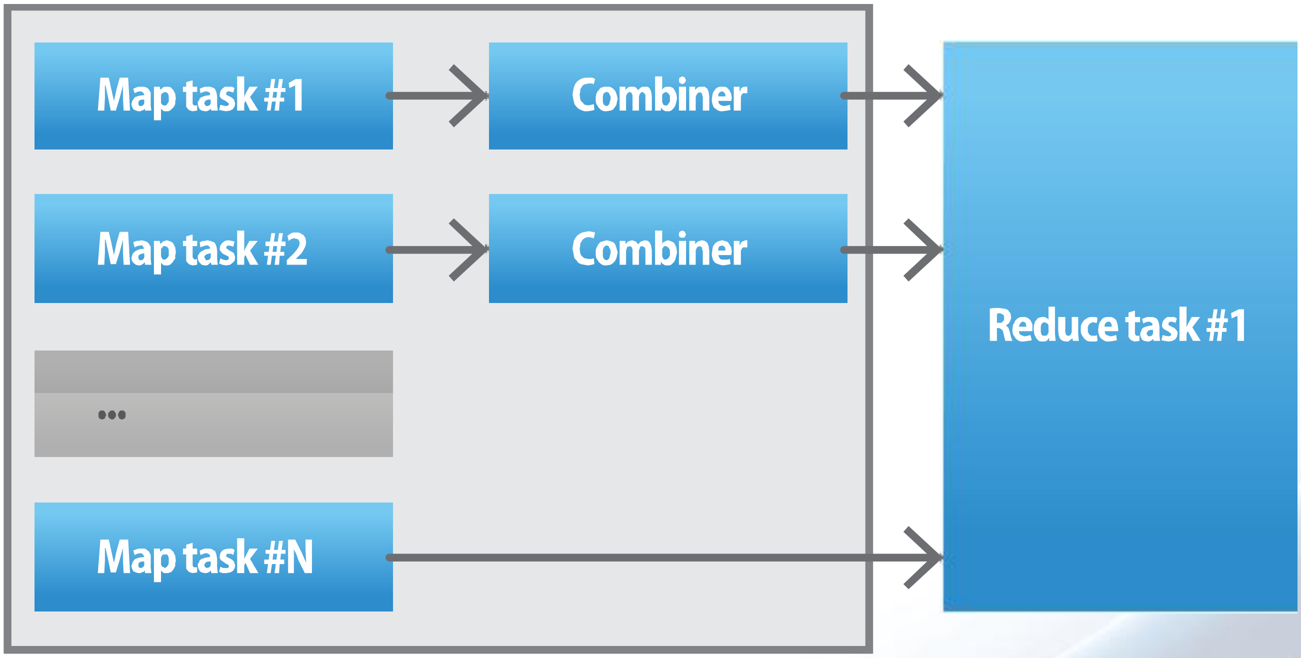


Figure 1. A simplified scheme of a MapReduce paradigm

2.1 Micro benchmarks

2.1.1 WordCount

WordCount can be used to evaluate CPU scalability of a cluster. On the Map stage, this workload extracts small amounts of data from a large data set and this process utilizes the total CPU capacity. Due to the very low load on disk load and network, under this kind of workload, a cluster of any size is expected to scale linearly.

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

Map input	Combiner input	Combiner output	Reduce output
1.00	1.42	**0.07**	**0.03**

Table 1. Changes in the amount of data at each of the MapReduce stages

2.1.2 Sort

This workload sorts out unsorted text data; the amount of data on all the stages of the Hadoop MapReduce process is the same. Being mostly I/O-bound, this workload has moderate CPU utilization, as well as heavy disk and network I/O utilization (during the shuffle stage). RandomTextWriter generates the input data.

Map input	Map output	Reduce output
1.0	1.0 (uncompressed)	1.0

Table 2. Changes in the amount of data at each of the MapReduce stages

2.1.3 TeraSort

TeraSort input data consists of 100-byte rows generated by the TeraGen application. Even though this workload has high/moderate CPU utilization during Map/Reduce stages respectively, it is mostly an I/O-bound workload.

Map input	Map output	Reduce output
1.0	0.2 (compressed)	1.0

Table 3. Changes in the amount of data at each of the MapReduce stages

2.1.4 Distributed File System I/O

The DFSIO test used in this benchmark is an enhanced version of TestDFSIO, which is an integral part of a standard Apache Hadoop distribution. This benchmark measures HDFS throughput.

2.2 Real-world applications

2.2.1 PageRank

PageRank is a widely-known algorithm that evaluates and ranks Web sites in search results. To calculate a rating, a PageRank job is repeated several times, which is an iterative CPU-bound workload. The benchmark consists of several chained Hadoop jobs, each represented by a separate row in the table below. In this benchmark, PageRank had two HDFS blocks per CPU core, which is the smallest input per node in this test.

Map input	Combiner input	Combiner output	Reduce output
1.0	1.0E-005	1.0E-007	1.0E-008
1.0	5.0		1.0
1.0	0.1		0.1

Table 4. Changes in the amount of data at each of the MapReduce stages

2.2.2 Bayes

The next application is a part of the Apache Mahout project. The Bayes Classification workload has rather complex patterns of accessing CPU, memory, disk, and network. This test creates a heavy load on a CPU when completing Map tasks. However in this case, this workload hit an I/O bottleneck.

Bayes

Map input	Combiner input	Combiner output	Reduce output
1.0	28.9	22.1	19.4
19.4	14.4	12.7	7.4
7.4	9.3	4.8	4.6
7.4	3.1	1.0E-004	1.0E-005

Table 5. Changes in the amount of data at each of the MapReduce stages

3. What Makes This Research Unique?

Although there are a variety of Hadoop distributions, tests evaluating their functional and performance characteristics are rare and provide rather general information, lacking deep analytical investigation of the subject. The wide majority of distributions are direct branches of the initial Apache Hadoop project and do not provide any additional features. Therefore, they were disregarded and are not covered in this research.

This research aims to provide a deep insight into Hadoop distributions and evaluate their features to support you in selecting a tool that fits your project best. For this benchmark, our R&D team selected three commonly used open-source Hadoop distributions:

- Cloudera's Distribution Including Apache Hadoop (CDH)v 4.3
- Hortonworks Data Platform (HDP) v1.3
- MapR M3 v3.0

In general, virtualized cloud environments provide flexibility in tuning, which was required to carry out tests on clusters of a different size. In addition, cloud infrastructure allows for obtaining more unbiased results, since all tests can be easily repeated to verify their results. For this benchmark, all tests were run on the ProfitBricks virtualized infrastructure. The deployment settings selected for each distribution provided similar test conditions, as much as it was possible.

In this research, we tested distributions that include updates, patches, and additional features that ensure stability of the framework. Hortonworks and Cloudera are active contributors to Apache Hadoop and they provide fast bug fixing for their solutions. Therefore, their distributions are considered more stable and up-to-date.

3.1 Testing environment

ProfitBricks was our partner that provided computing capacities to carry out the research. This company is a leading IaaS provider and it provides great flexibility in choosing node configuration. The engineers did not have to address the Support Department to change some crucial parameters of the node disk configuration, therefore they were able to try different options and find the optimal settings for the benchmarking environment.

The service provides a wide range of configuration parameters for each node, for instance, the CPU capacities of a node can vary from 1 to 48 cores, and RAM can be from 1 to 196 GB per node. This variety of options allows for achieving the optimal CPU/RAM/network/storage balance for each Hadoop task.

Unlike Amazon that offers preconfigured nodes, ProfitBricks allows for manual tuning of each node based on your previous experience and performance needs. InfiniBand is a modern technology used by ProfitBricks. It allowed for achieving the maximum inter-node communication performance inside a data center.

Cluster configuration:

Each node had four CPU cores, 16 GB of RAM, and 100 GB of virtualized disk space. Cluster size ranged from 4 to 16 nodes. Nodes required for running Ganglia and cluster management were not included into this configuration. The top cluster configuration featured 64 computing cores and 256 GB of RAM for processing 1.6 TB of test data.

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

4. Results

Each Cloudera and Hortonworks DataNode contained one disk. MapR distribution was evaluated in a slightly different way. Three data disks were attached to each DataNode following MapR recommendations. Taking this into account, MapR was expected to perform I/O sensitive tasks three times faster. However, the actual results were affected by some peculiarities of virtualization (see Figure 11).

The comparison of Cloudera and Hortonworks features showed that these two distributions are very similar (see Appendix A). It was also proved by the results of the tests (see Appendix B). The overall performance of Hortonworks and Cloudera clusters is demonstrated by Figure 4 and 6 respectively.

4.1 Overall cluster performance

Throughput in bytes per second was measured for a cluster that consisted of 4, 8, 12, and 16 DataNodes (Figures 2-7). The throughput of 8-, 12-, and 16-node clusters was compared against the throughput of a four-node cluster in each benchmark test. The speed of data processing of 8-, 12-, and 16-node clusters was divided by the throughput of a 4-node cluster. These values demonstrate cluster scalability in each of the tests. The higher the value is, the better.

Although data consistency may be guaranteed by a hosting/cloud provider, to employ the advantages of data locality, Hadoop requires using its internal data replication.

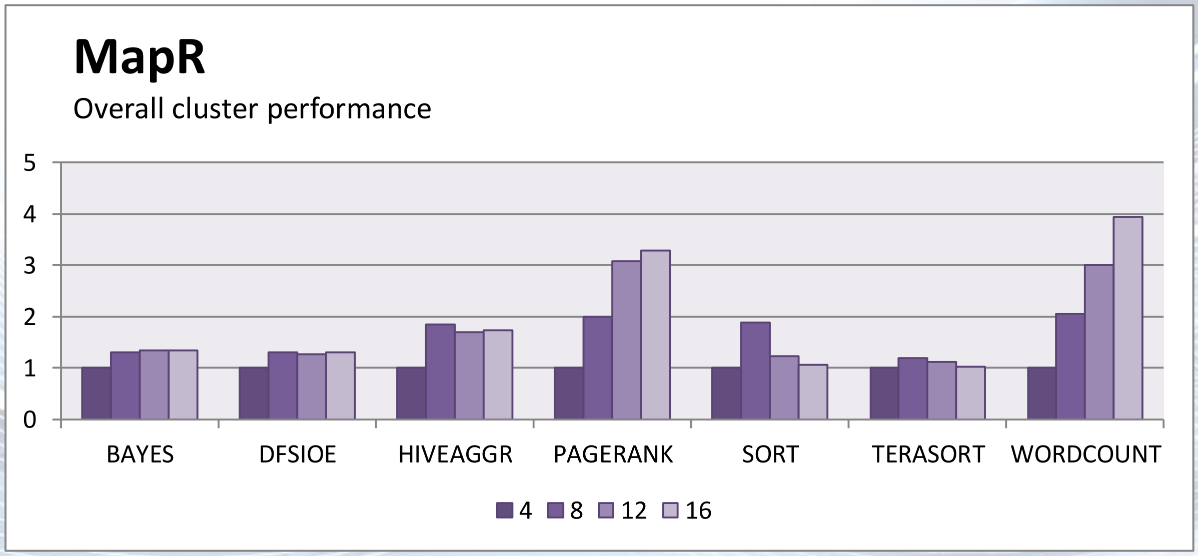


Figure 2. The overall performance results of the MapR distribution in all benchmark tests

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

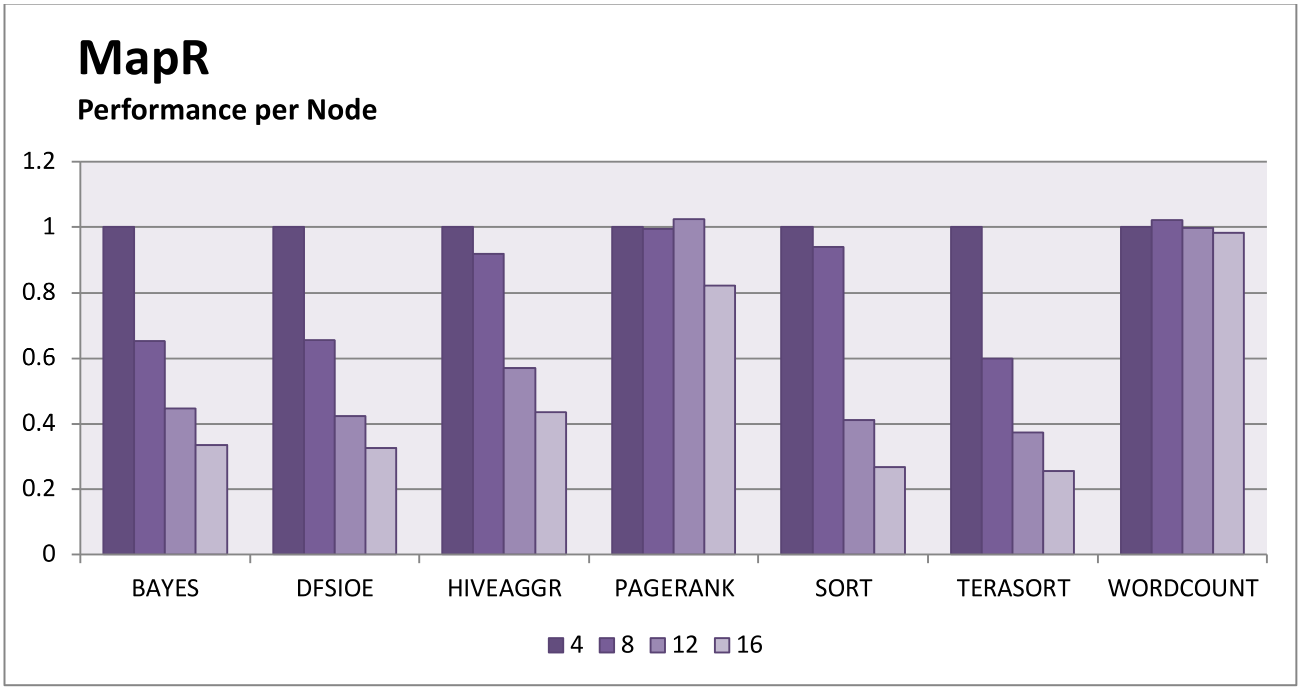


Figure 3. The average performance of a single node of the MapR cluster in all benchmark tests

Cluster performance scales linearly under the WordCount workload. It behaves the same in running PageRank until the cluster reaches an I/O bottleneck. The results of other benchmarks strongly correlated with DFSIO. Disk I/O throughput did not scale in this test environment, however, analyzing the reasons for that was not the focus of this research. To learn more about the drawbacks of Hadoop virtualization, read "Hadoop Operations" by Eric Sammer, Chapter "Planning a Hadoop Cluster".

As it was mentioned before, MapR had three disks per node. In case all nodes are hosted on the same server, the virtual cluster utilizes disk bandwidth—which is obviously limited—much faster. ProfitBricks allows for hosting up to 48-62 cores on the same server, which is equivalent to a cluster that consists of 12–15 nodes with the configuration described in this benchmark (four cores per node).

4.2 Hortonworks Data Platform (HDP)

In most cases, the performance of a cluster based on the Hortonworks distribution scaled more linearly. However, its starting performance value in disk-bound tasks was lower than that of MapR. The maximum scalability was reached in WordCount and PageRank tasks. Unfortunately, when the cluster grew to eight nodes, its performance stopped to increase due to I/O limitations.

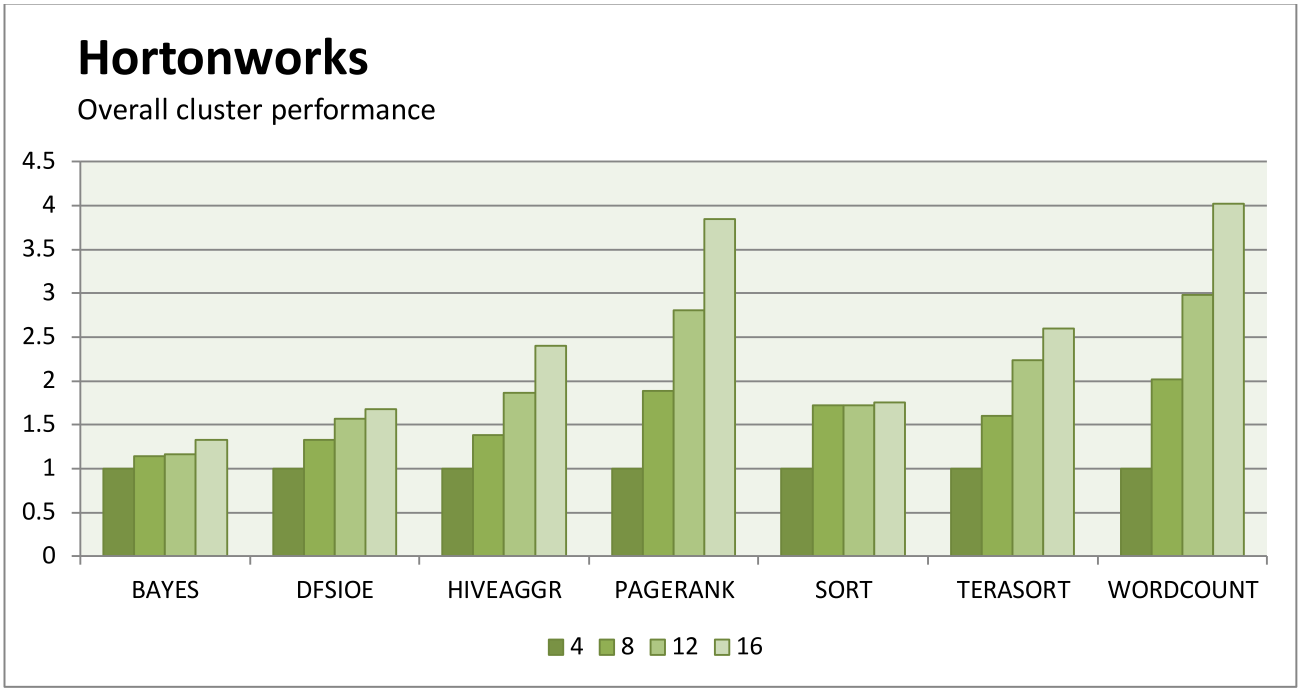


Figure 4. The overall performance results of the Hortonworks cluster in all benchmark tests

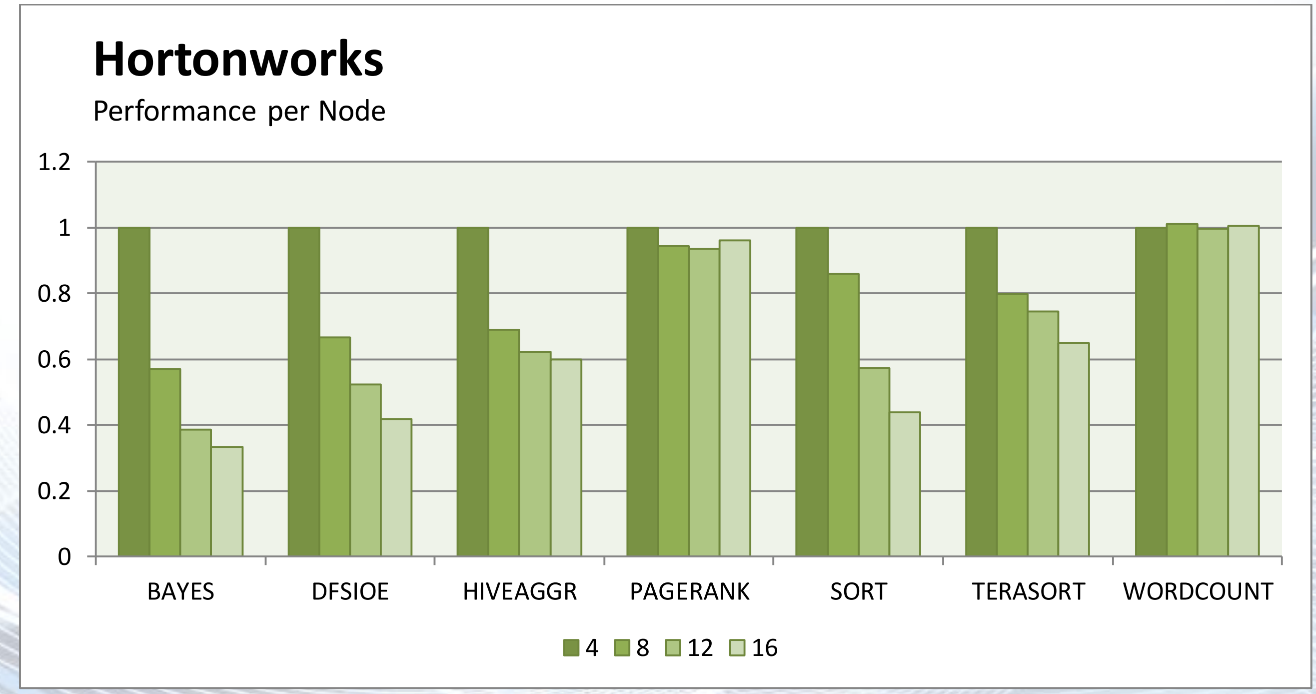


Figure 5. The average performance of a single node of the Hortonworks cluster in all benchmark tests

4.3 Cloudera's Distribution Including Apache Hadoop (CDH)

The Cloudera Hadoop distribution showed almost the same performance as Hortonworks, except for Hive queries, where it was slower.

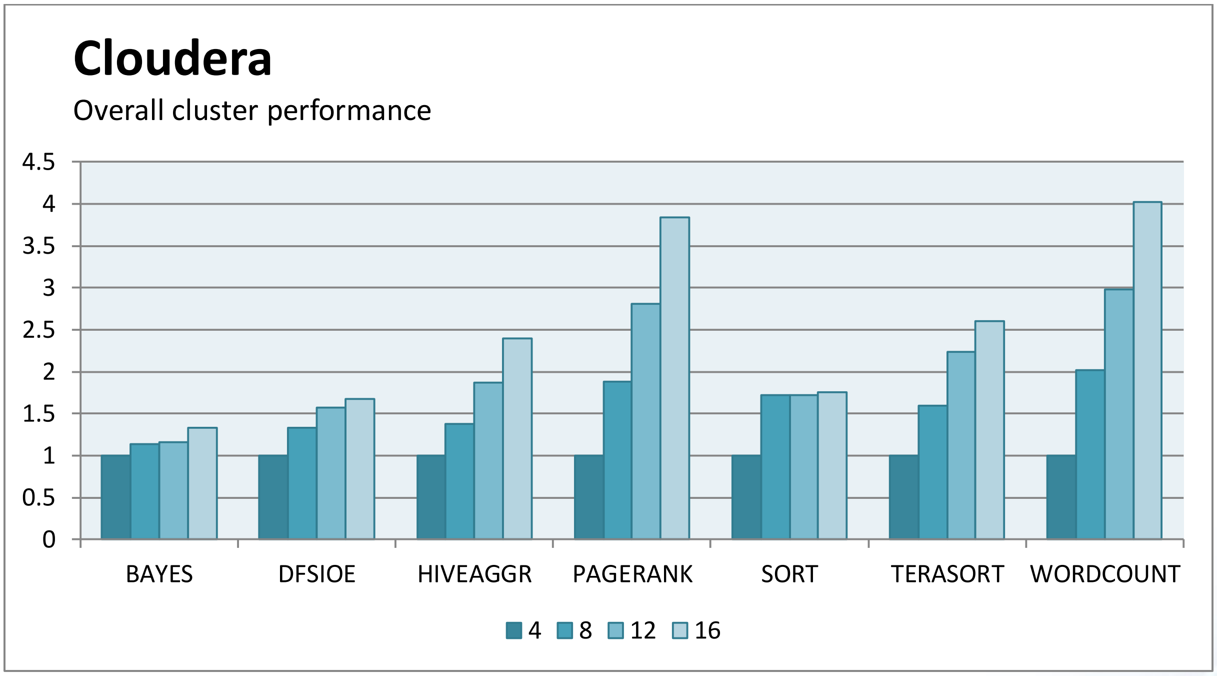


Figure 6. The overall performance results of the Cloudera cluster in all benchmark tests

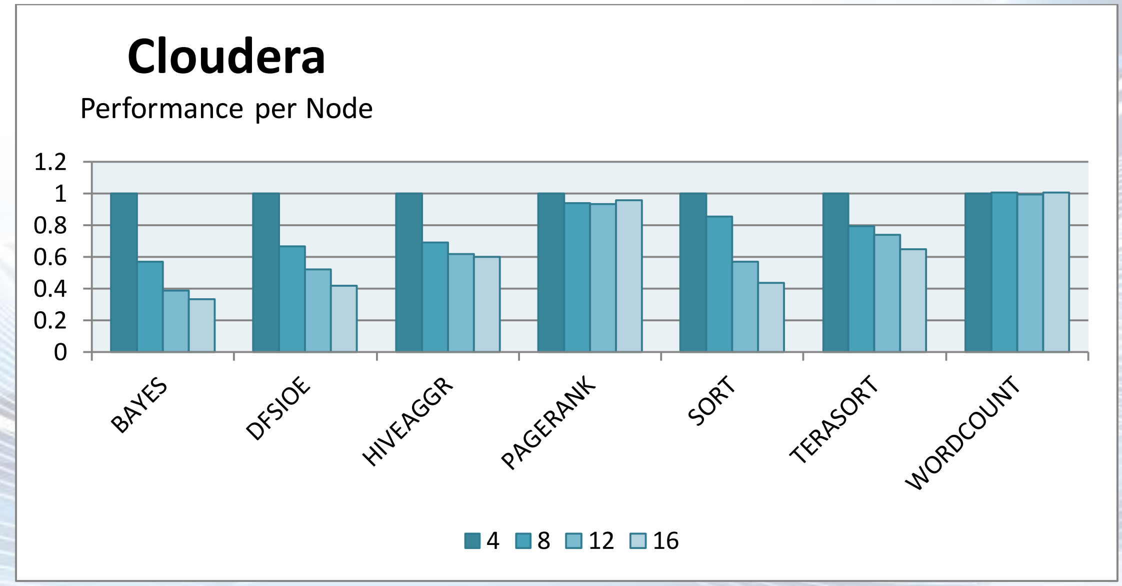


Figure 7. The average performance of a single node of the Cloudera cluster in all benchmark tests

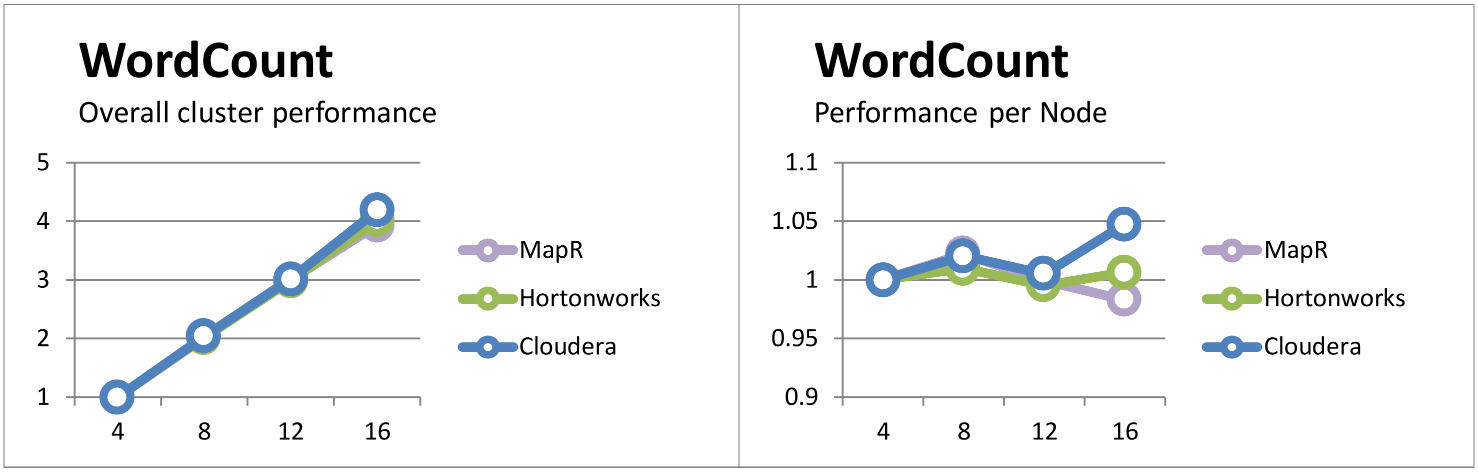


Figure 8. Throughput scalability measured in the CPU-bound benchmark

Cluster performance under CPU-bound workloads increased as expected: a 16-node cluster was four times as fast as a 4-node one. However, the difference in performance between the distributions was within the limit of an experimental error.

4.4 MapR

The performance results of the MapR cluster under the Sort load were quite unexpected.

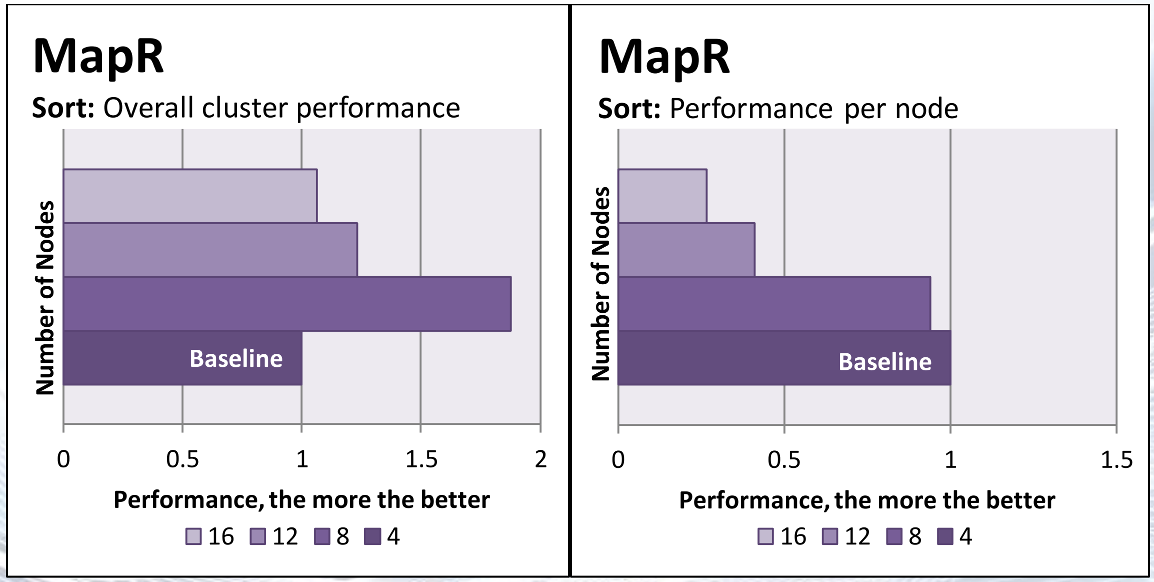
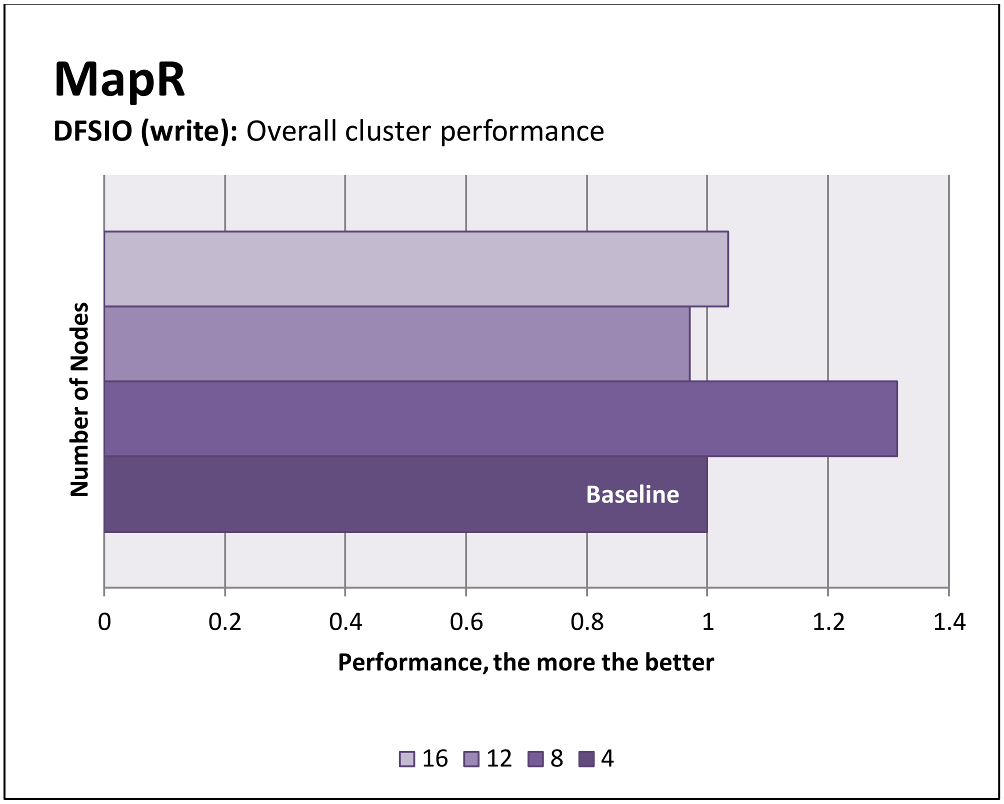


Figure 9. Performance results for MapR in the Sort benchmark

In the Sort task, the cluster scaled linearly from four to eight nodes. After that, the performance of each particular node started to degrade sharply. The same situation was observed in the DFSIO write test.



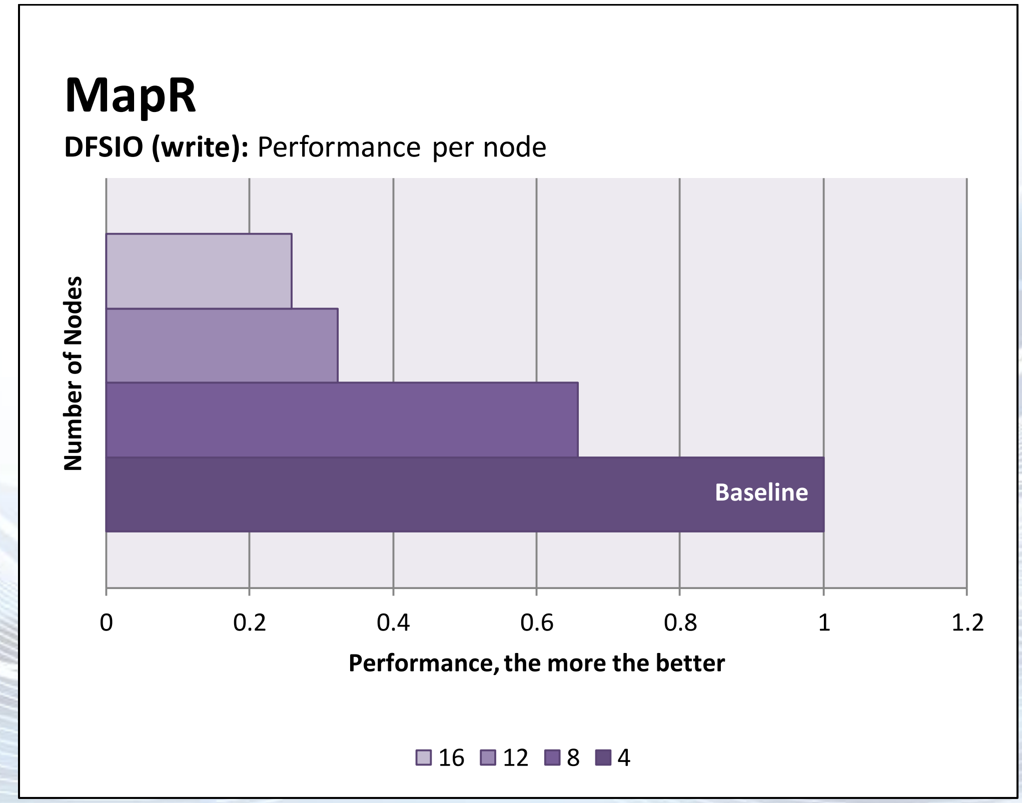
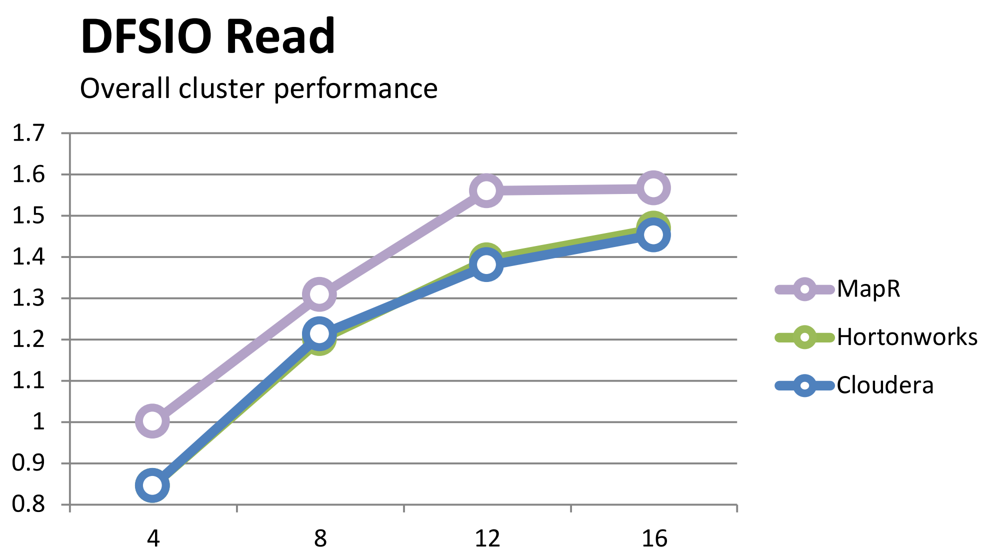


Figure 10. The MapR performance results in the DFSIO (write) benchmark

The virtualized disks of a 4-node cluster showed the reading/writing speed of 250/700 MB/sec. The overall cluster performance grew not in a linear way (see Figure 11), meaning that the total speed of data processing can be improved by the optimal combination of CPU, RAM, and disk space parameters.



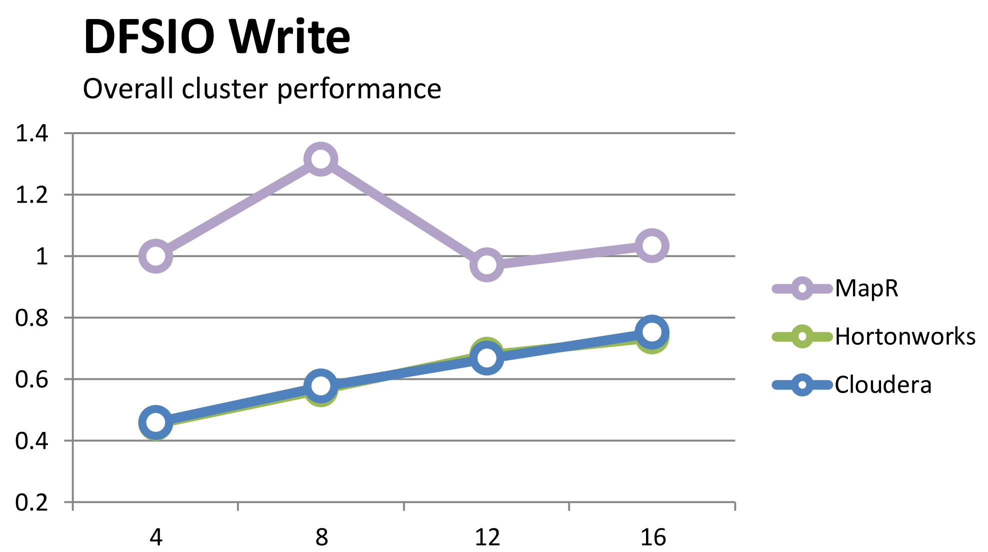


Figure 11. Performance results for the MapR, Hortonworks, and Cloudera distributions in the DFSIO (read/write) benchmark

5. Conclusion

Despite of the fact that the configuration of a cluster deployed in the cloud was similar to that of the one deployed on bare metal, the performance and scalability of the virtualized solution were different. In general, Hadoop deployed on bare metal is expected to scale linearly, until inter-node communication will start to slow it down or it reaches the limits of the HDFS, which is around several thousand of nodes.

The actual measurements showed that even though the overall performance was very high, it was affected by the limited total disk throughput. Therefore, the disk I/O became a serious bottleneck whereas the computing capacities of the cluster were not fully utilized. Apache Spark, which was announced by Cloudera when this research was conducted, or GridGain's In-Memory Accelerator for Hadoop can be suggested for using in the ProfitBricks environment.

It can be assumed that the type of Hadoop distribution has a much less considerable impact on the overall system throughput than the configuration of the MapReduce task parameters. For instance, the TeraSort workload was processed 2–3 times faster when the parameters described in Appendix E were tuned specifically for this load. By configuring these settings, you can achieve 100% utilization of your CPU, RAM, disk, and network. So, the performance of each distribution can be greatly improved by selecting proper parameters for each specific load.

Running Hadoop in clouds allows for fast horizontal and vertical scaling, however, there are fewer possibilities for tuning each part of the infrastructure. In case you opt for a virtualized deployment, you should select a hosting/IaaS provider that gives freedom in configuring your infrastructure. To achieve optimal utilization of resources, you will need information on the parameters set for network and disk storage and have a possibility to change them.

Liked this white paper?
Share it on the Web!

Appendix A: Main Features and Their Comparison Across Distributions

Component type	Implementation	Hortonworks HDP -1.3	Cloudera CDH 4.3	MapR M3 v3.0
File system		HDFS 1.2.0	HDFS 2.0.0	MapR-FS
- non-Hadoop access		NFSv3	Fuse-DFS v2.0.0	Direct Access NFS
- Web access	REST HTTP API	WebHDFS	HttpFS	*
MapReduce		1.2.0	0.20.2	**
- software abstraction layer	Cascading	x	x	2.1
Non-relational database	Apache HBase	0.94.6.1	0.94.6	0.92.2
Metadata services	Apache HCatalog	***Hive	0.5.0	0.4.0
Scripting platform	Apache Pig	0.11	0.11.0	0.10.0
- data analysis framework	DataFu	x	0.0.4	x
Data access and querying	Apache Hive	0.11.0	0.10.0	0.9.0
Workflow scheduler	Apache Oozie	3.3.2	3.3.2	3.2.0
Cluster coordination	Apache Zookeeper	3.4.5	3.4.5	3.4(?)
Bulk data transfer between relational databases and Hadoop	Apache Sqoop	1.4.3	1.4.3	1.4.2
Distributed log management services	Apache Flume	1.3.1	1.3.0	1.2.0
Machine learning and data analysis	Mahout	0.7.0	0.7	0.7

Hadoop UI	Hue	2.2.0	2.3.0	-
- data integration service	Talend Open Studio for Big Data	5.3	x	x
Cloud services	Whirr	x	0.8.2	0.7.0
Parallel query execution engine		*Tez (Stinger)*	Impala	****
Full-text search	Search		0.1.5	
Administration		Apache Ambari	Cloudera Manager	MapR Control System
- installation		Apache Ambari	Cloudera Manager	-
- monitoring		Ganglia	x	x
		Nagios	x	x
- fine-grained authorization	Sentry		1.1	
Splitting resource management and scheduling	*YARN*	*2.0.4*	*2.0.0*	-

Table 6. The comparison of functionality in different Hadoop distributions

* - via NFS
** - MapR has a custom Hadoop-compatible MapReduce implementation
*** - HCatalog has been merged with Hive. The latest stand-alone release was v0.5.0

**** - Apache Drill is at an early development stage
x - available, but not mentioned in the distribution documentation / requires manual installation or additional configuration

Appendix B: Overview of the Distributions

1. MapR

Summary

MapR has the MapRFS feature which is a substitute for the standard HDFS. Unlike HDFS, this system aims to sustain deployments that consist of up to 10,000 of nodes with no single point of failure, which is guaranteed by the distributed NameNode. MapR allows for storing 1–10 Exabytes of data and provides support for NFS and random read-write semantics. It is stated by the MapR developers that elimination of the Hadoop abstraction layers can help to increase performance 2x.

There are three editions of the MapR distribution: M3, which is completely free, M5 and M7, the latter two are paid enterprise versions. Although M3 provides unlimited scalability and NFS, it does not ensure high availability and snapshots that are available in M5 or instant recovery of M7. M7 is an enterprise-level platform for NoSQL and Hadoop deployments. MapR distribution is available as a part of Amazon Elastic MapReduce and Google Cloud Platform.

Notable customers and partners

- MapR M3 and M5 editions are available as premium options for Amazon Elastic MapReduce;
- Google partnered with MapR in launching Compute Engine;
- Cisco Systems announced support for MapR software on the UCS platform;
- comScore

Support and documentation

- Support contact details
- Documentation

The company

Based in San Jose, California, MapR focuses on development of Hadoop-based projects. The company contributes to such projects as HBase, Pig, Apache Hive, and Apache ZooKeeper. After signing an agreement with EMC in 2011, the company supplies a specific Hadoop distribution tuned for EMC hardware. MapR also partners with Google and Amazon in improving their Elastic Map Reduce (EMR) service.

2. Cloudera

Summary

Of all the distributions analyzed in this research, Cloudera's solution has the most powerful Hadoop deployment and administration tools designed for managing a cluster of an unlimited size. It is also open-source and the company is an active contributor to Apache Hadoop. Cloudera is a major Apache Hadoop contributor. In addition, the Cloudera distribution has its own native components, such as Impala, a query engine for massive parallel processing and Cloudera Search powered by Apache Solr.

Notable customers and partners

- eBay
- CBS Interactive
- Qualcomm
- Expedia

Support and documentation

- Support contact details
- Documentation

The company

Based in Palo Alto, Cloudera is one of the leading companies that provides Hadoop-related services and trainings for the staff. According to the company's statistics, more than 50% of its efforts are dedicated to improving such open-source projects as Apache Hive, Apache Avro, Apache HBase, etc. that are a part of a large Hadoop ecosystem. In addition, Cloudera invests into Apache Software Foundation, a community of developers who contribute to the family of Apache software projects.

3. Hortonworks

Summary

Being 100% open-source, Hortonworks is strongly committed to Apache Hadoop and it is one of the main contributors to the solution. Stinger is one of the company's initiatives that brings high performance, scalability, and SQL compliance to Hadoop deployments. YARN, the Hadoop OS, and Apache Tez, a framework for near real-time big data processing, help Stringer to speed up Hive and Pig by up to 100x.

As a result of Hortonworks's partnership with Microsoft, HDP is the only Hadoop distribution available as a native component of Windows Server. A Windows-based Hadoop cluster can be easily deployed on Windows Azure through HDInsight Service.

Notable customers and partners

- Western Digital
- eBay
- Samsung Electronics

Support and documentation

- Support contact details
- Documentation

The company

Hortonworks is a company headquartered in Palo Alto, California. Being a sponsor of the Apache Software Foundation and one of the main contributors to Apache Hadoop, the company specializes in providing support for Apache Hadoop. The Hortonworks distribution includes such components as HDFS, MapReduce, Pig, Hive, HBase, and Zookeeper. Together with Yahoo!, Hortonworks hosts the annual Hadoop Summit event, the leading conference for the Apache Hadoop community.

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

Appendix C: Performance Results for Each Benchmarking Test

1. Real-world applications

1.1 Bayes

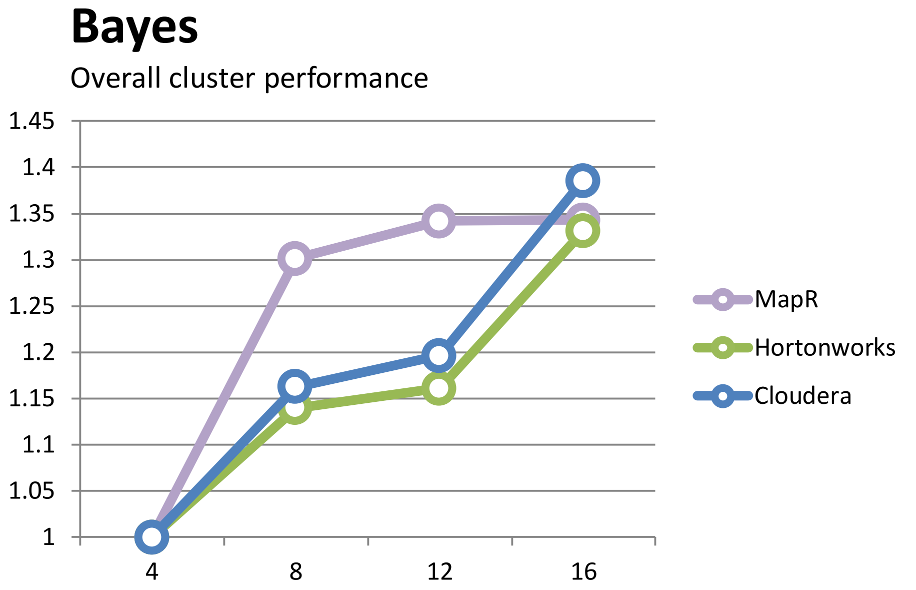


Figure 12. Bayes: the overall cluster performance

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

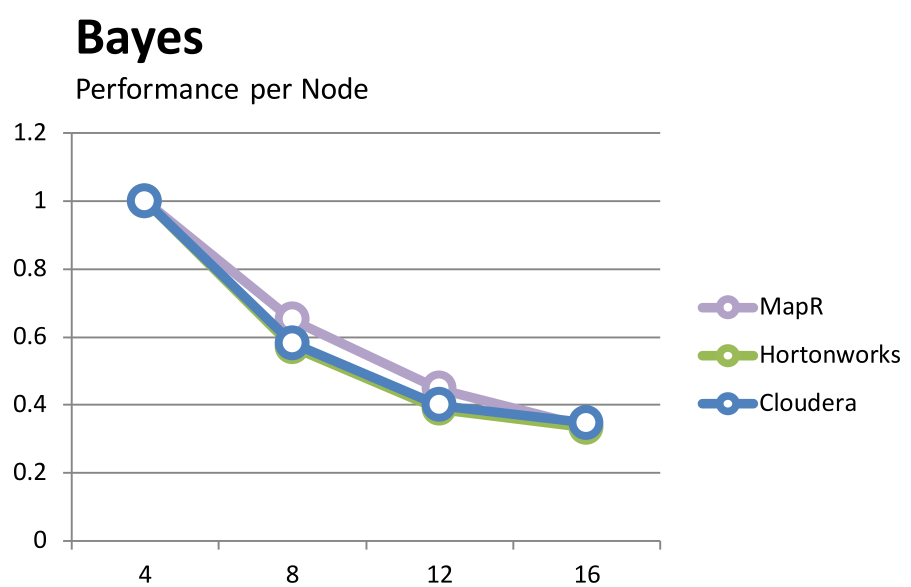


Figure 13: Bayes: the performance of a single node for each cluster size

1.2 PageRank

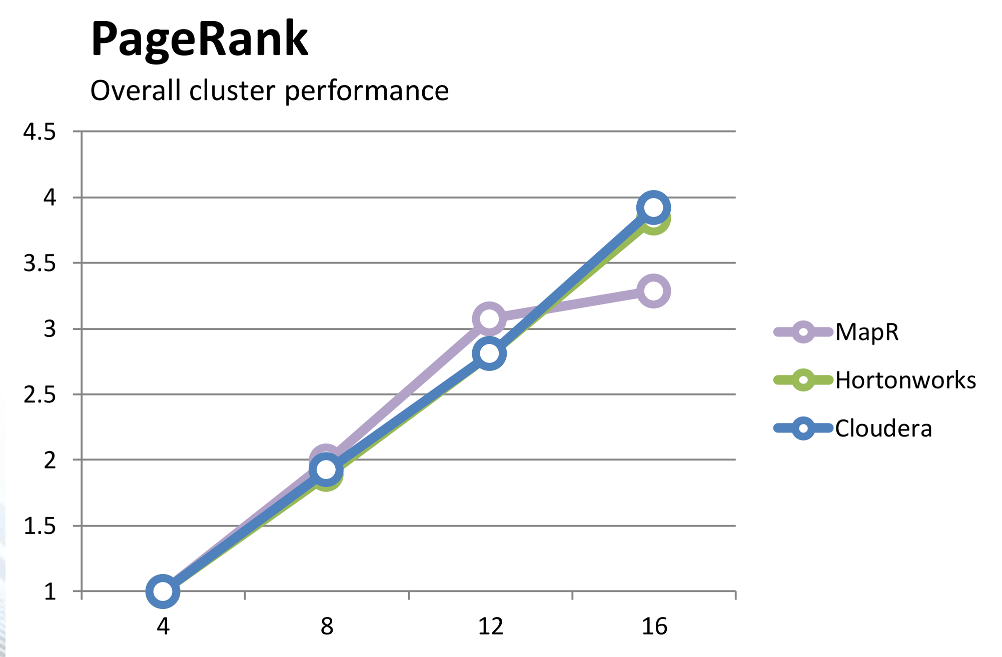


Figure 14. PageRank: the overall cluster performance

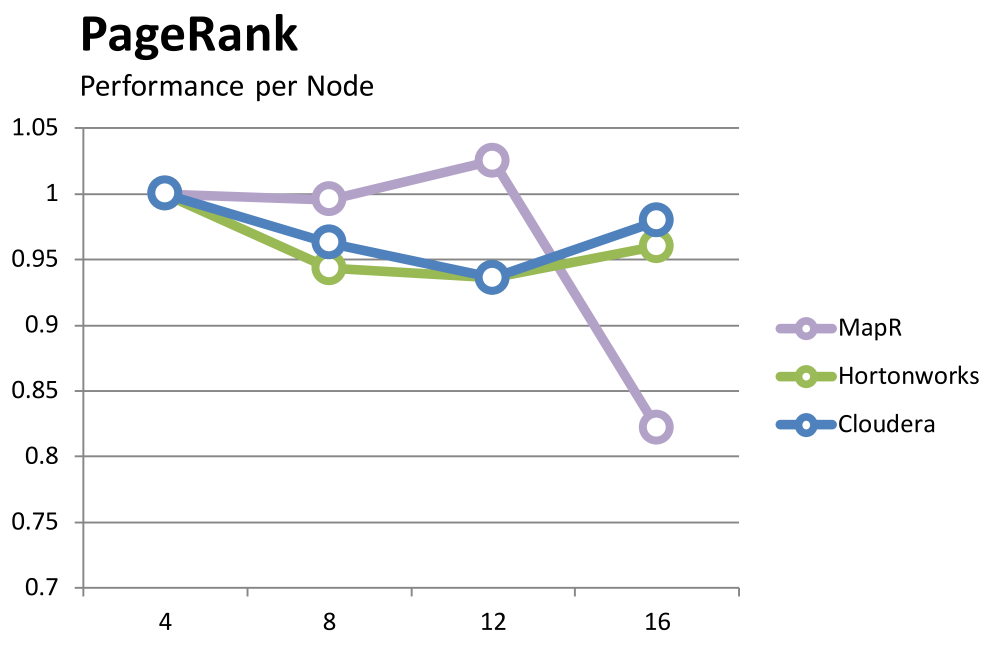


Figure 15. PageRank: the performance of a single node for each cluster size

2. Micro benchmarks

2.1 Distributed File System I/O (DFSIO)

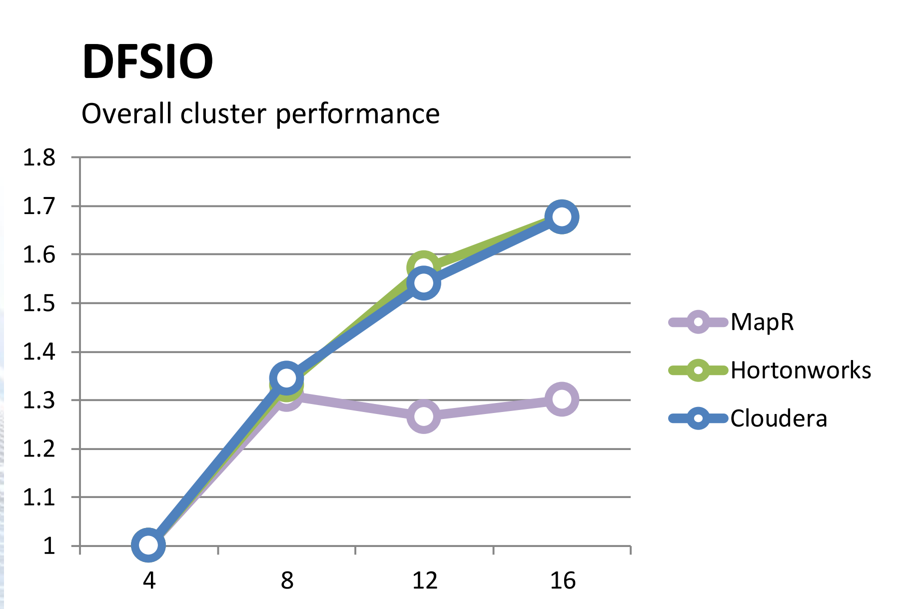


Figure 16. DFSIO: the overall cluster performance

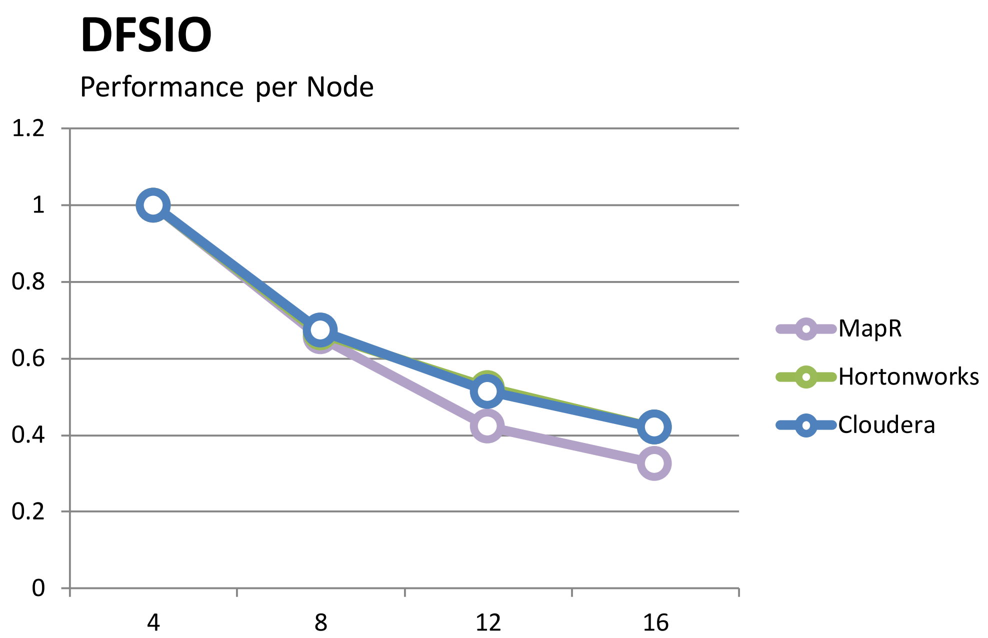


Figure 17. DFSIO: the performance of a single node for each cluster size

2.2 Hive aggregation

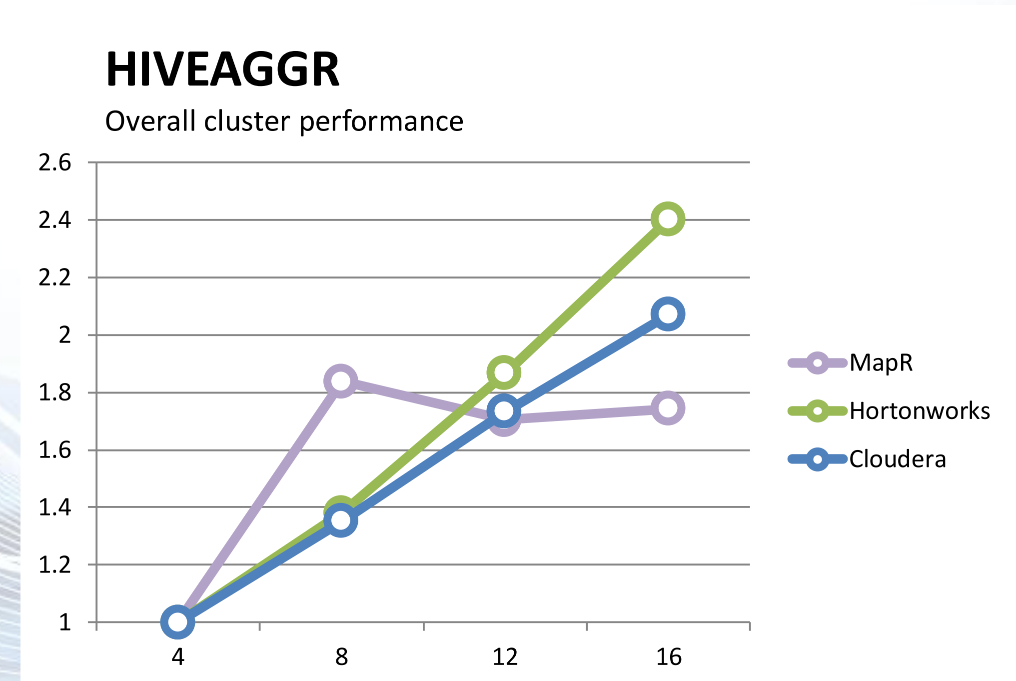


Figure 18. Hive aggregation: the overall cluster performance

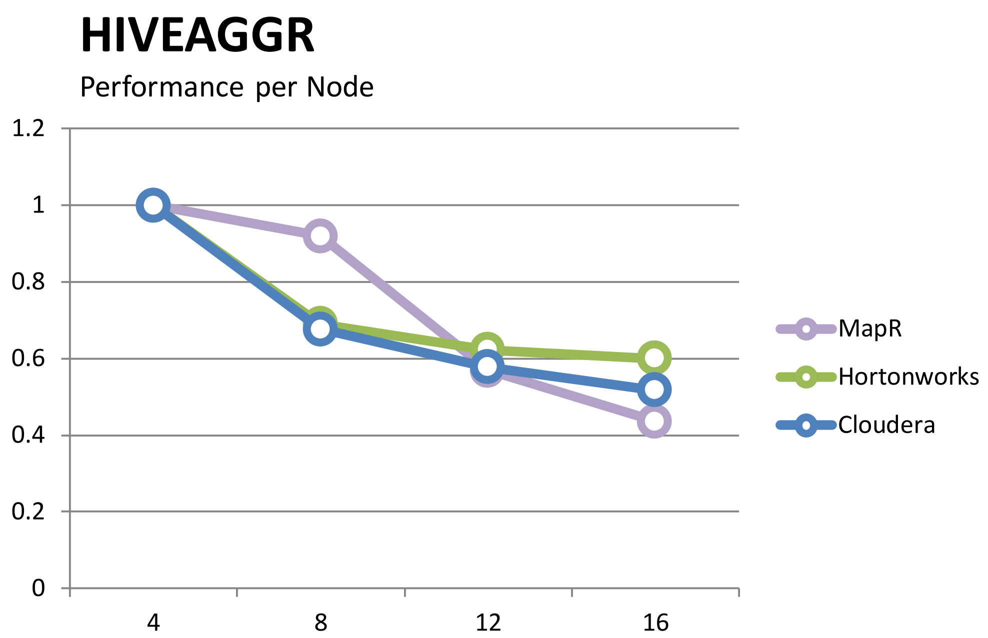


Figure 19. Hive aggregation: the performance of a single node for each cluster size

2.3 Sort

Figure 20. Sort: the overall cluster performance

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

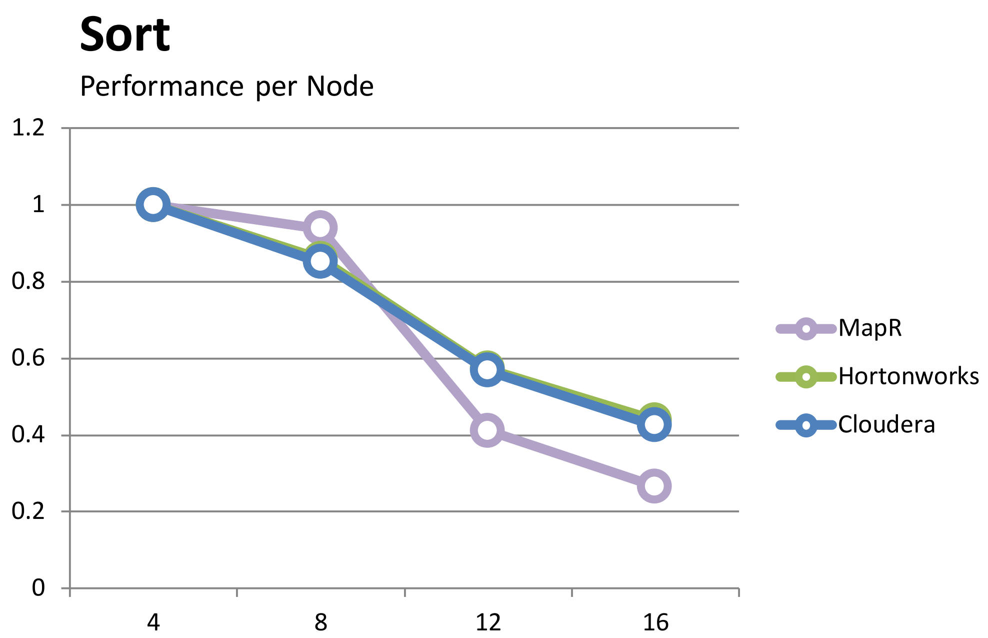


Figure 21. Sort: the performance of a single node for each cluster size

2.4 TeraSort

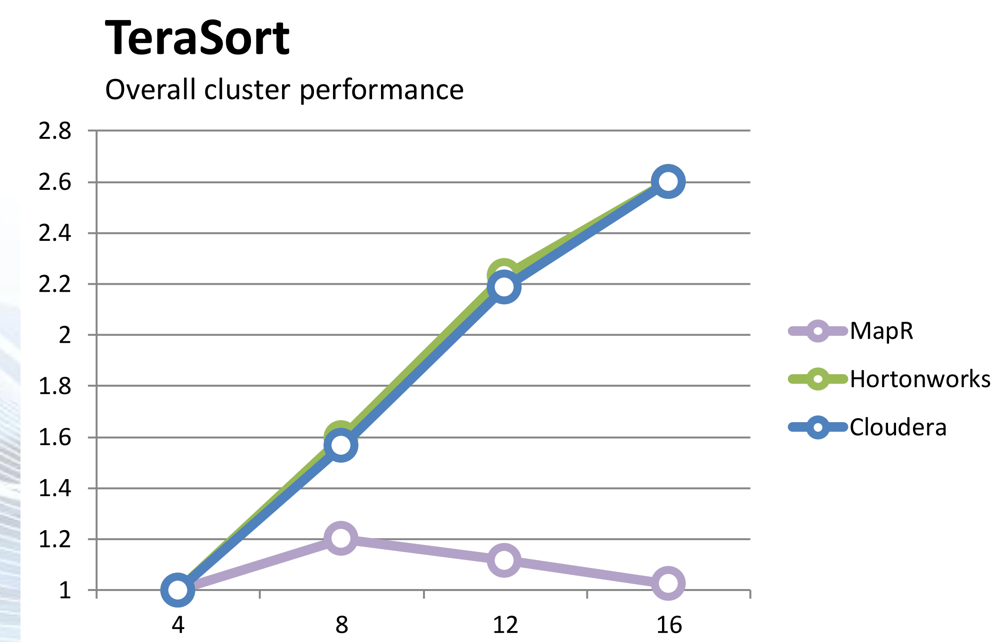


Figure 22. TeraSort: the overall cluster performance

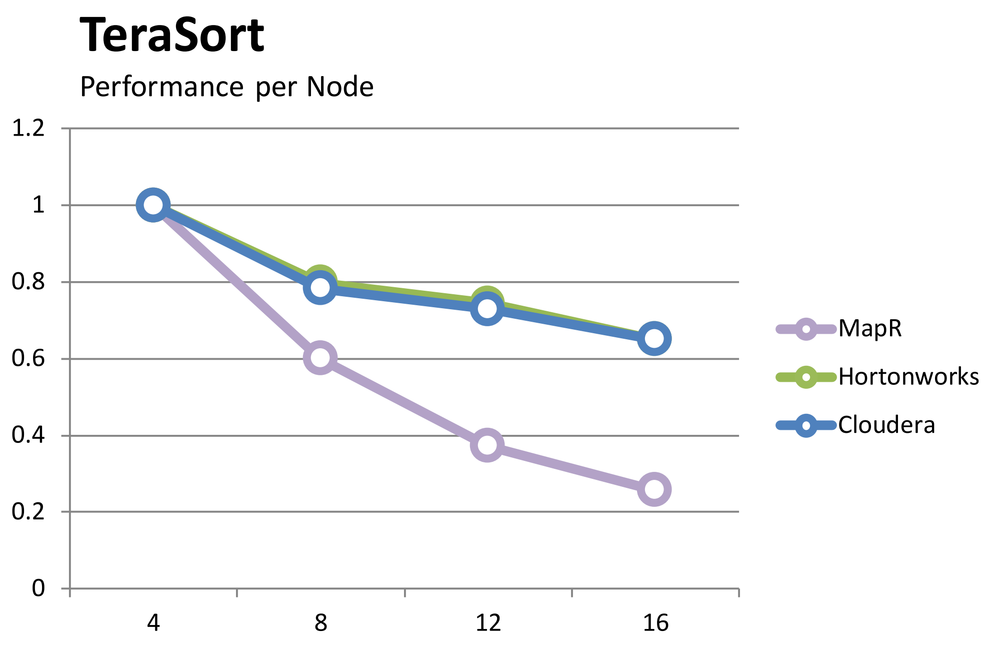


Figure 23. TeraSort: the performance of a single node for each cluster size

2.5 WordCount

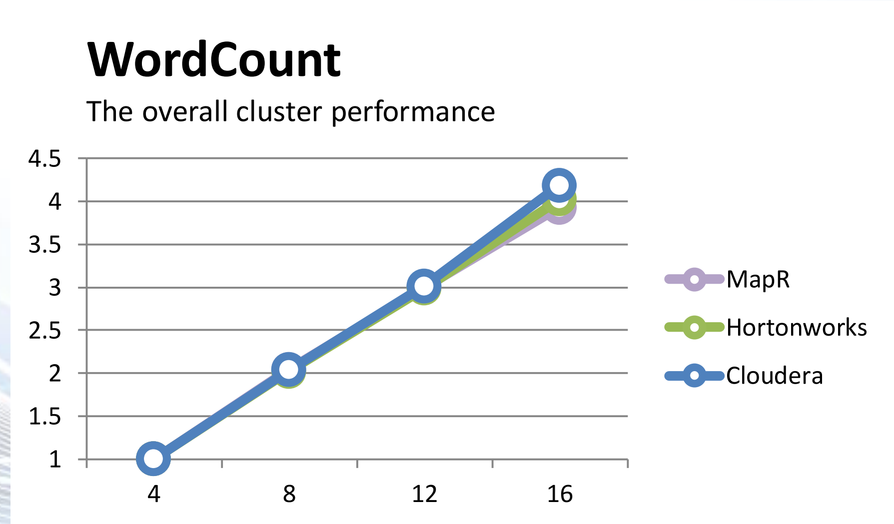


Figure 24. WordCount: the overall cluster performance

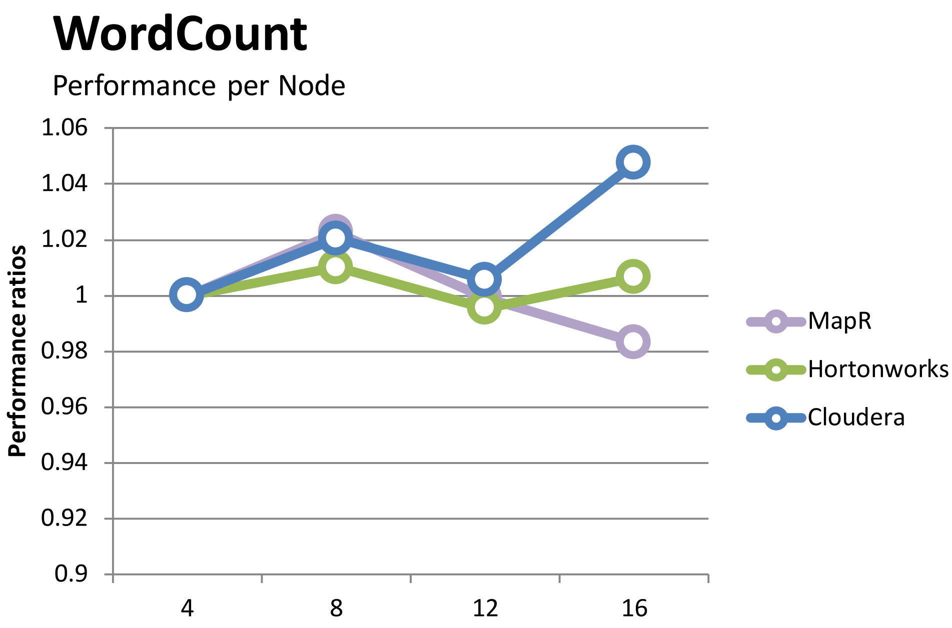


Figure 25. WordCount: the performance of a single node for each cluster size

Appendix D: Performance Results for Each Test Sectioned by Distribution

1. MapR

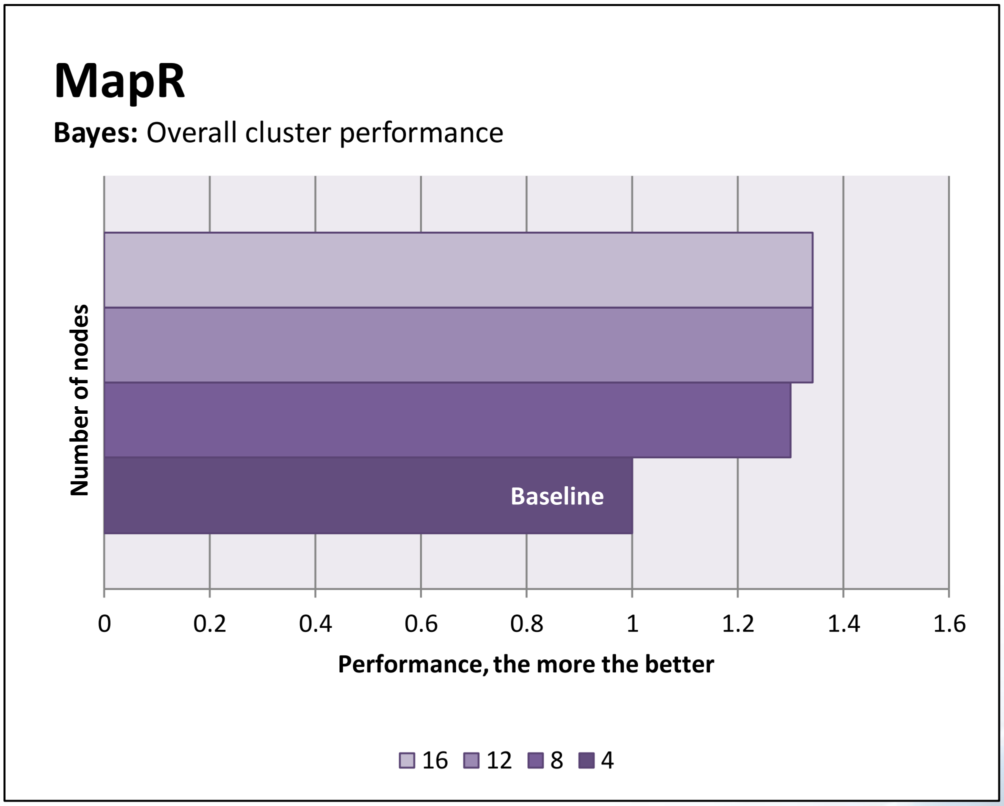


Figure 26. The overall performance of the MapR cluster in the Bayes benchmark

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

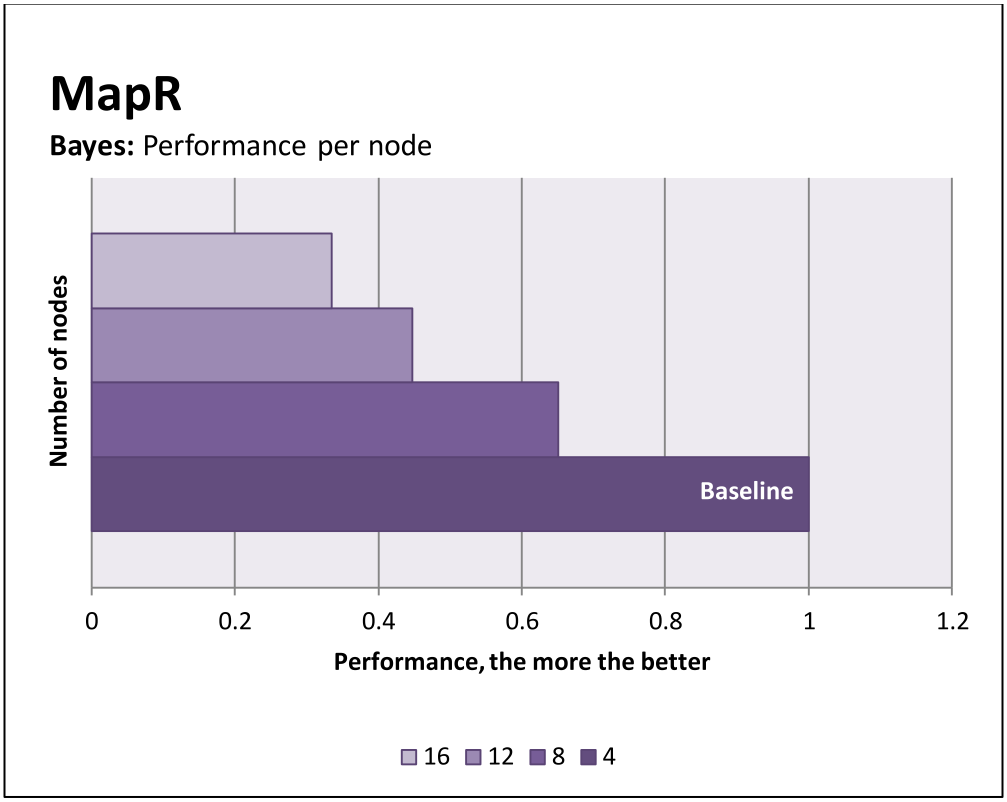


Figure 27. The performance of a single node of the MapR cluster in the Bayes benchmark

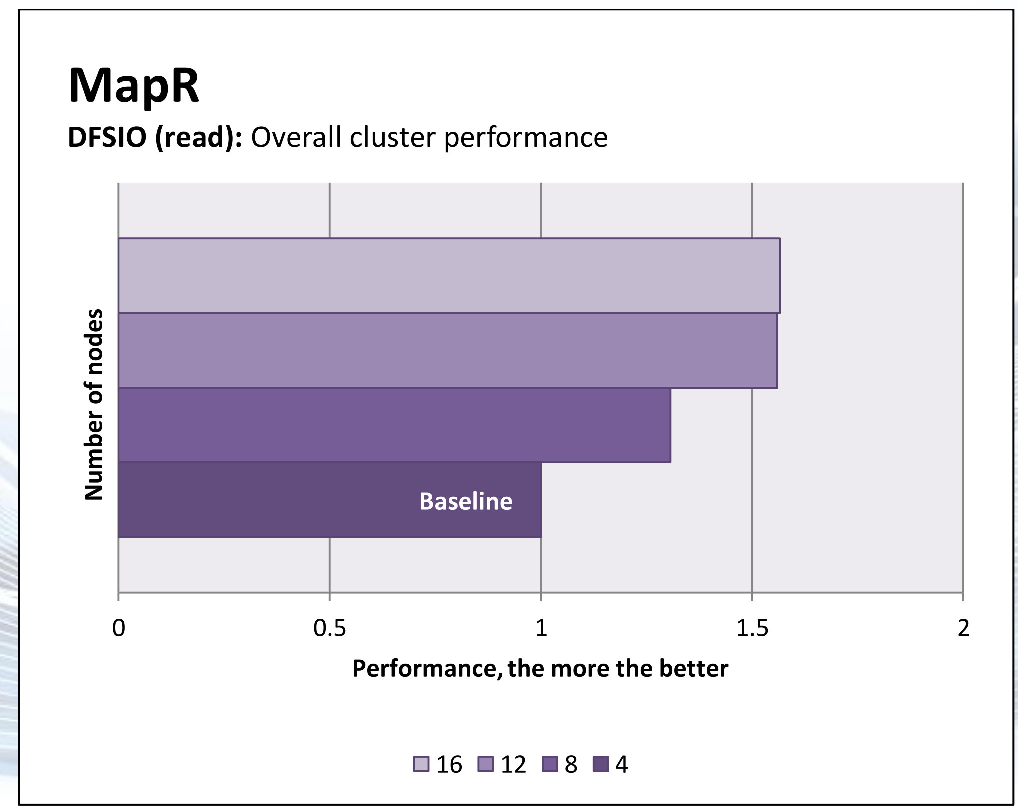


Figure 28. The overall performance of the MapR cluster in the DFSIO (read) benchmark

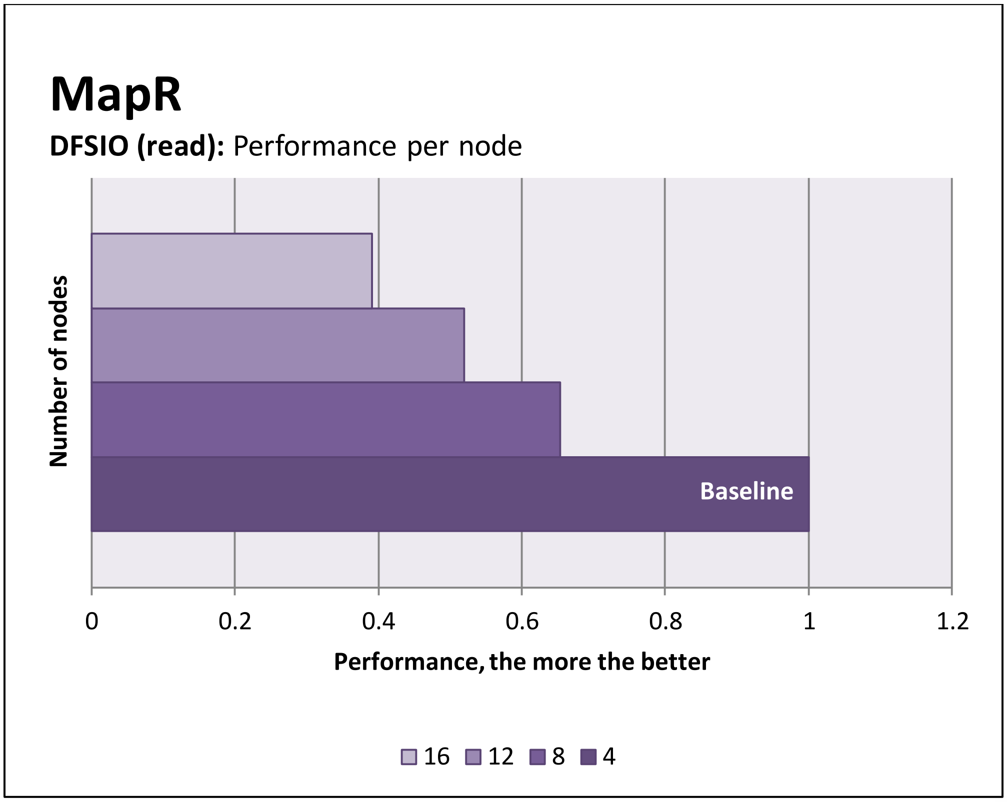


Figure 29. The performance of a single node of the MapR cluster in the DFSIO (read) benchmark

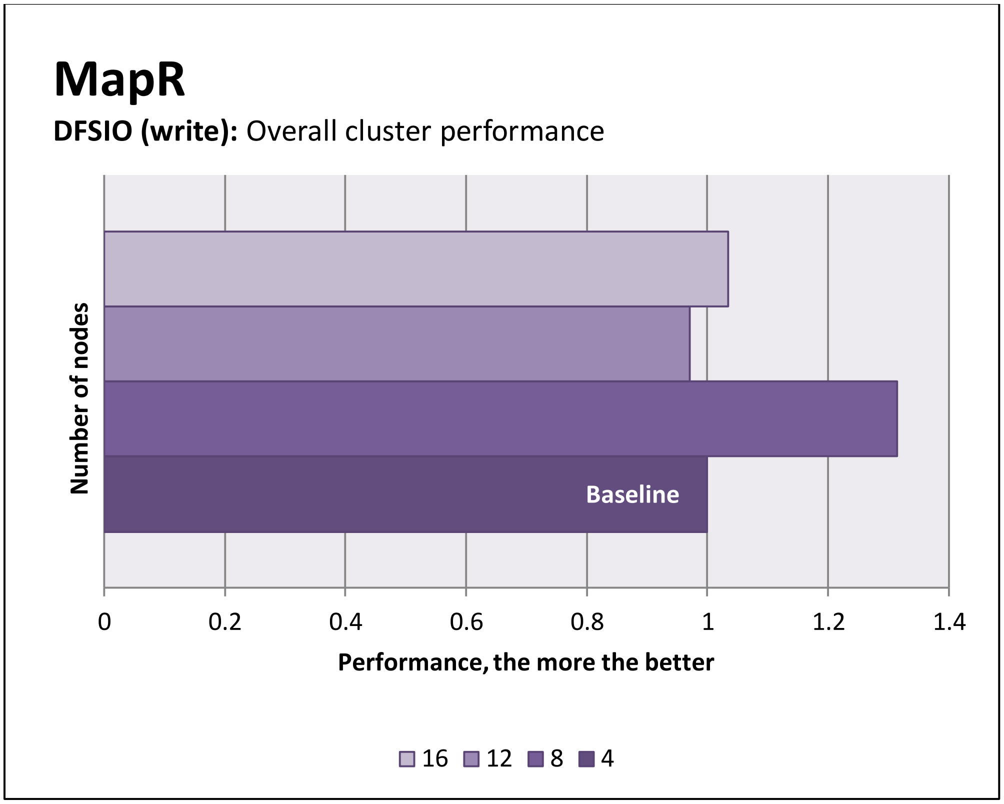


Figure 30. The overall performance of the MapR cluster in the DFSIO (write) benchmark

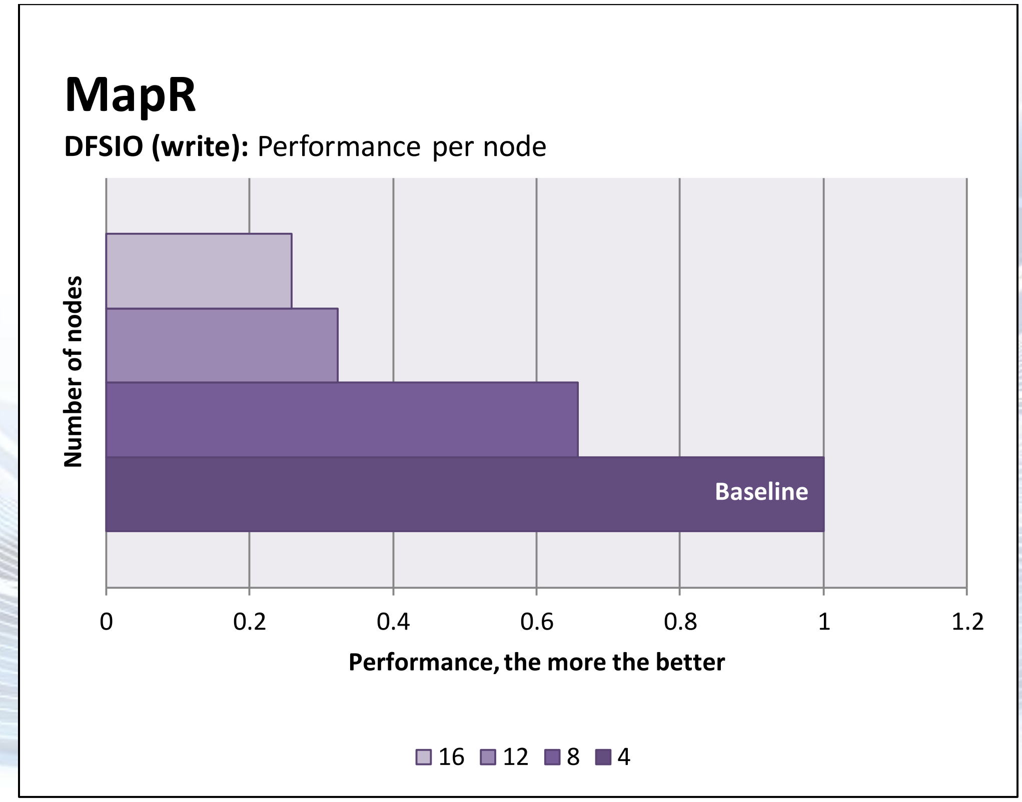


Figure 31. The performance of a single node of the MapR cluster in the DFSIO (write) benchmark

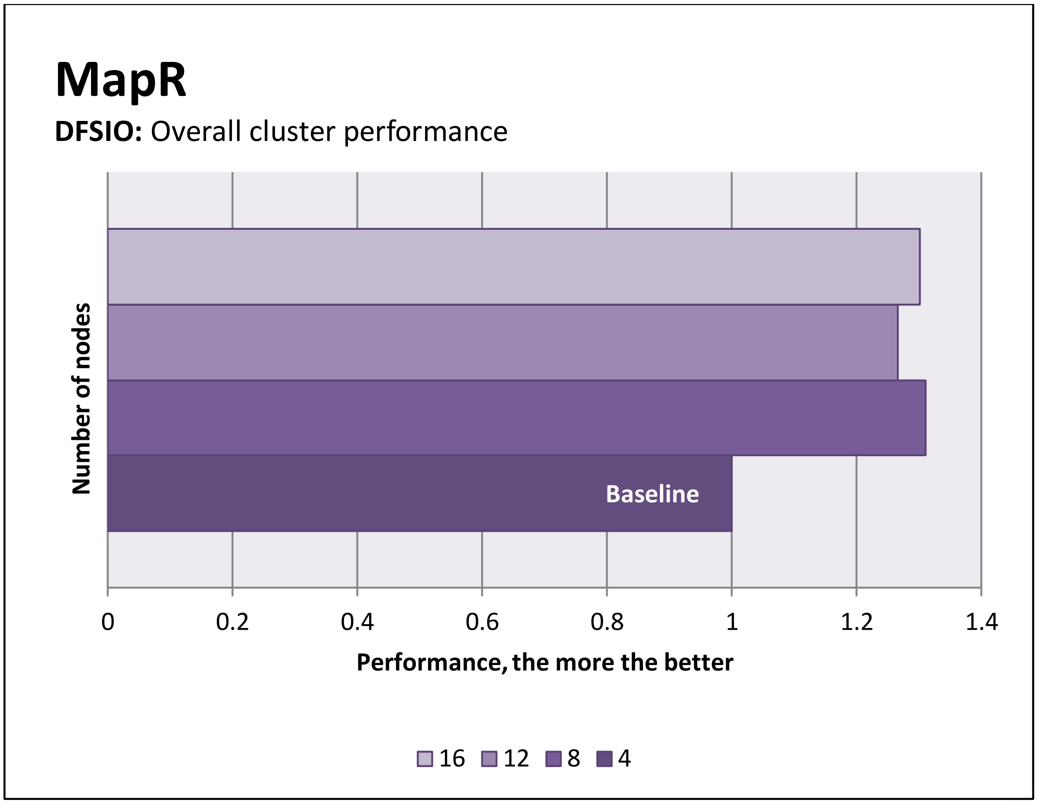


Figure 32. The overall performance of the MapR cluster in the DFSIO benchmark

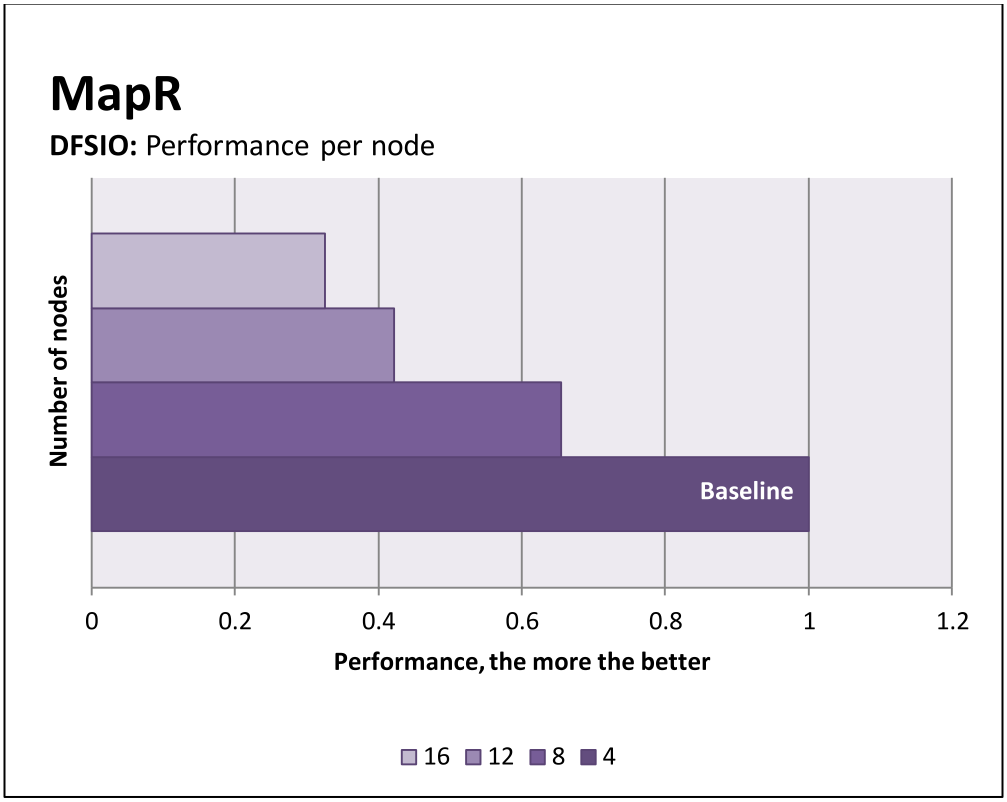


Figure 33. The performance of a single node of the MapR cluster in the DFSIO benchmark

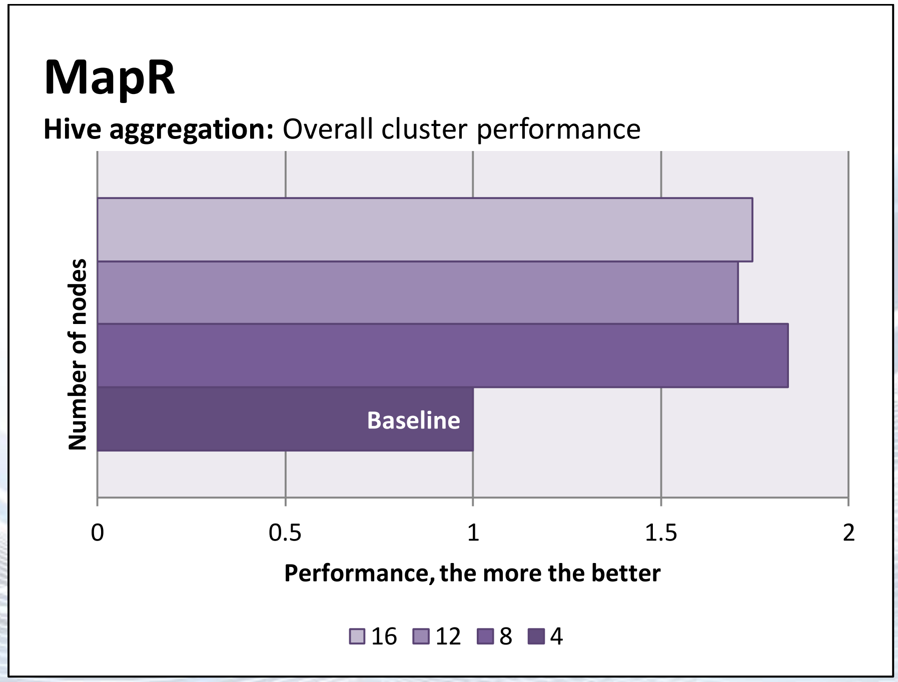


Figure 34. The overall performance of the MapR cluster in the Hive aggregation benchmark

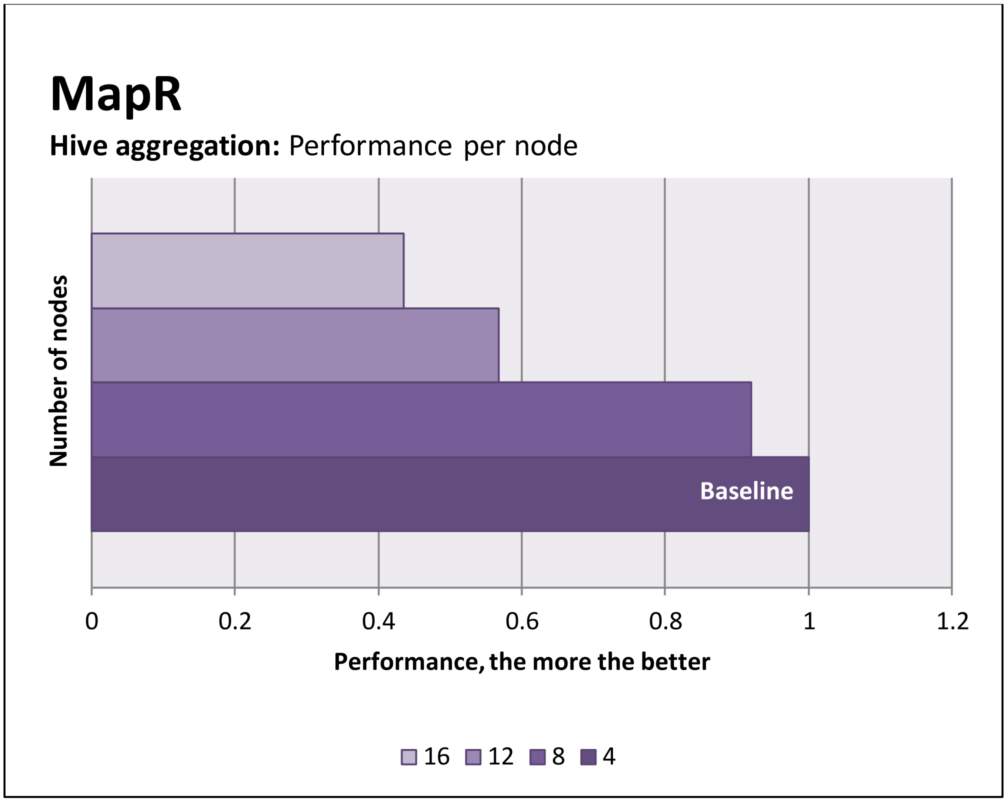


Figure 35. The performance of a single node of the MapR cluster in the Hive aggregation benchmark

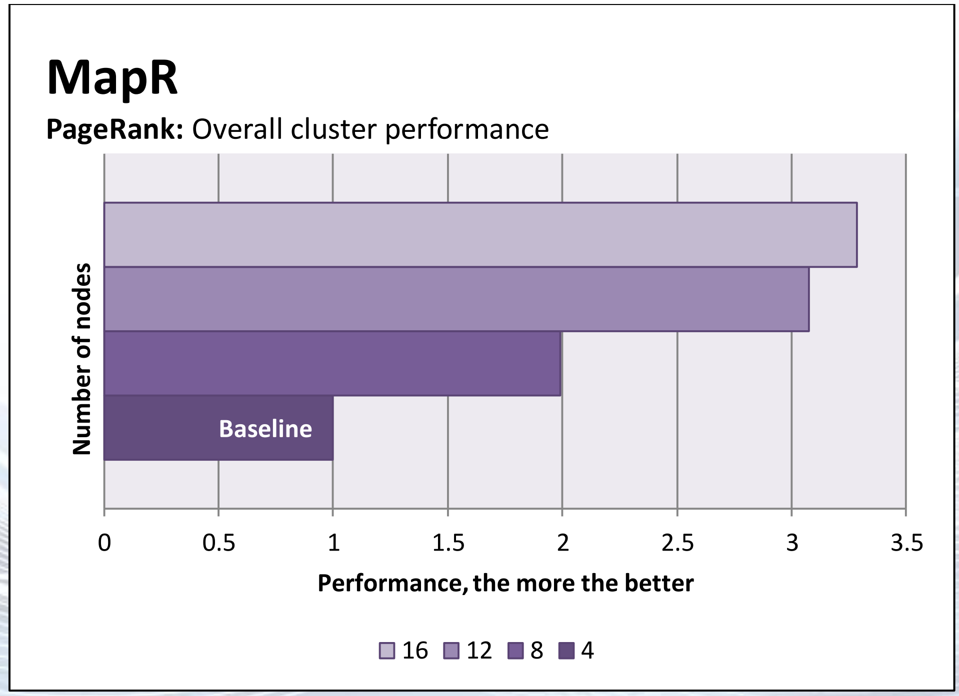


Figure 36. The overall performance of the MapR cluster in the PageRank benchmark

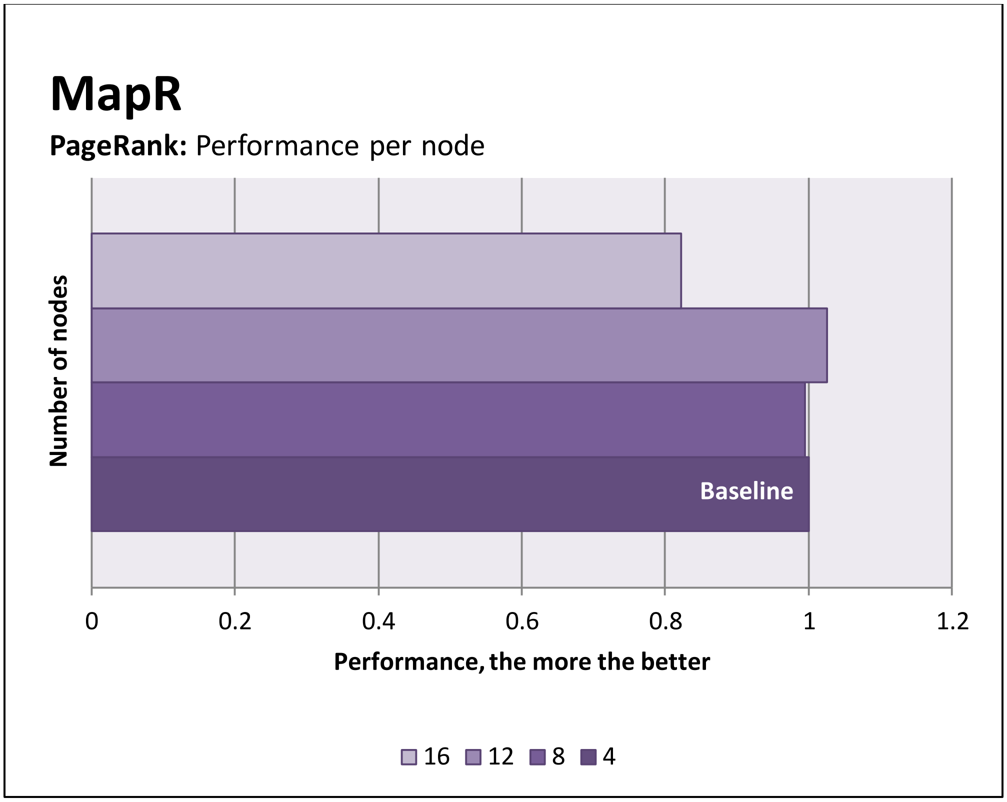


Figure 37. The performance of a single node of the MapR cluster in the PageRank benchmark

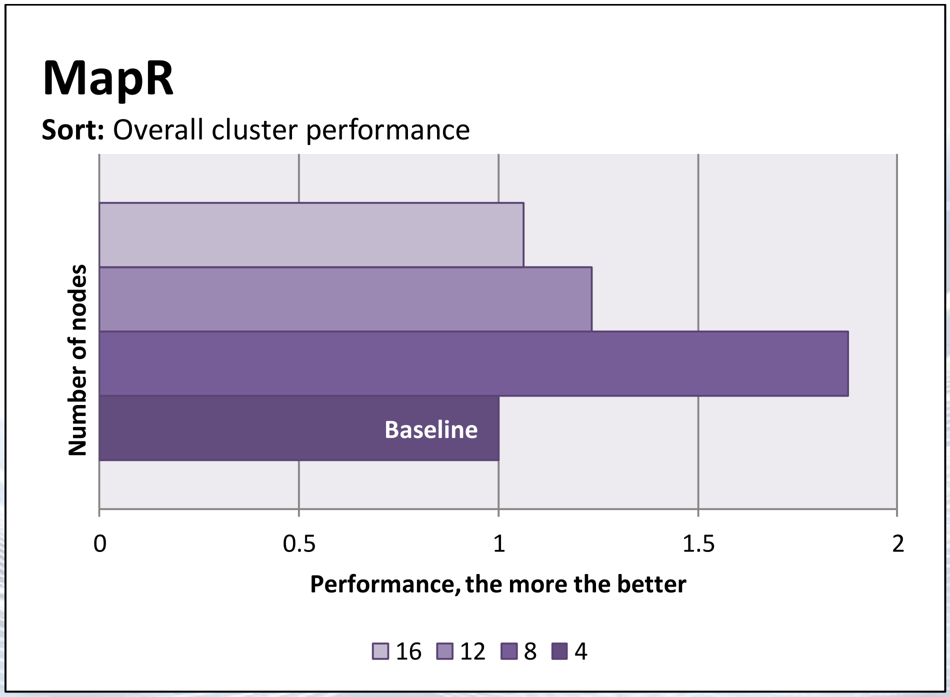


Figure 38. The overall performance of the MapR cluster in the Sort benchmark

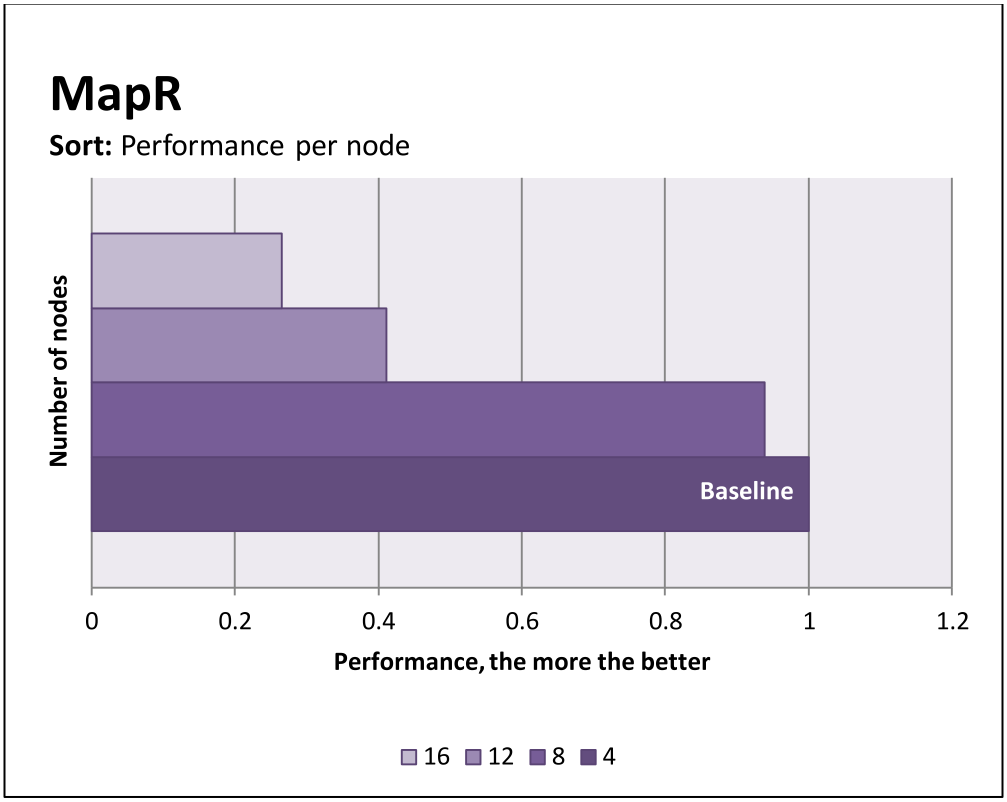


Figure 39. The performance of a single node of the MapR cluster in the Sort benchmark

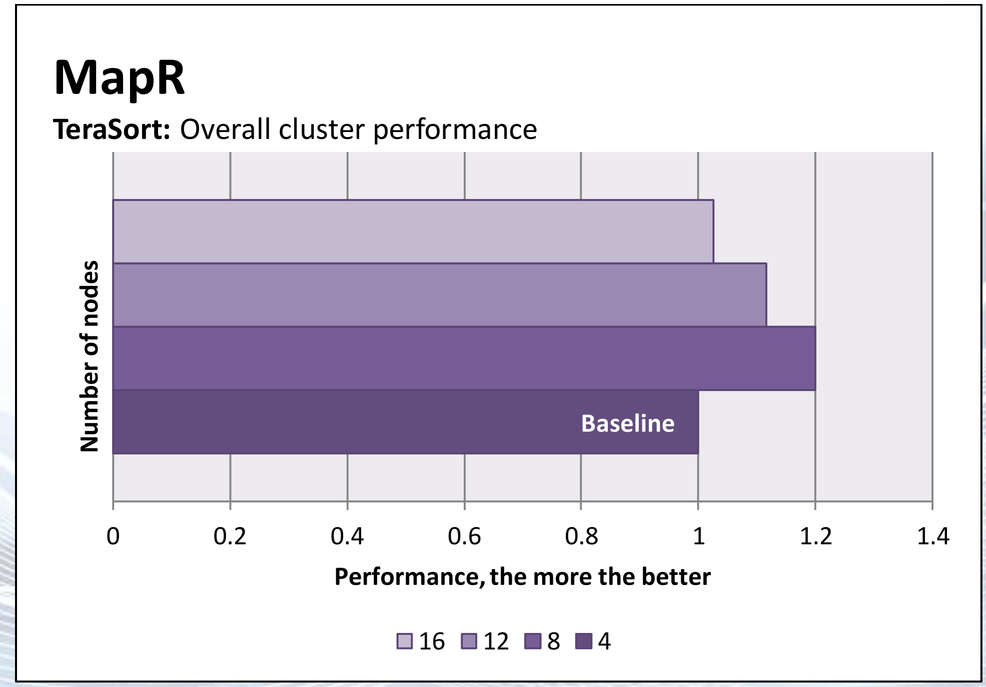


Figure 40. The overall performance of the MapR cluster in the TeraSort benchmark

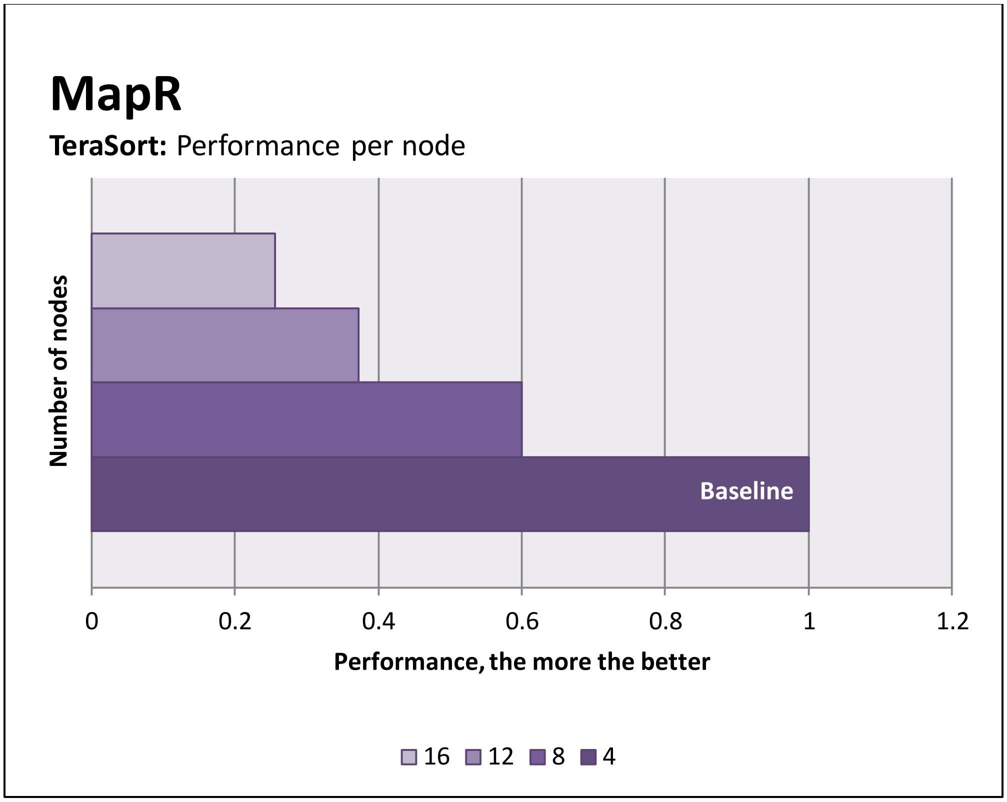


Figure 41. The performance of a single node of the MapR cluster in the TeraSort benchmark

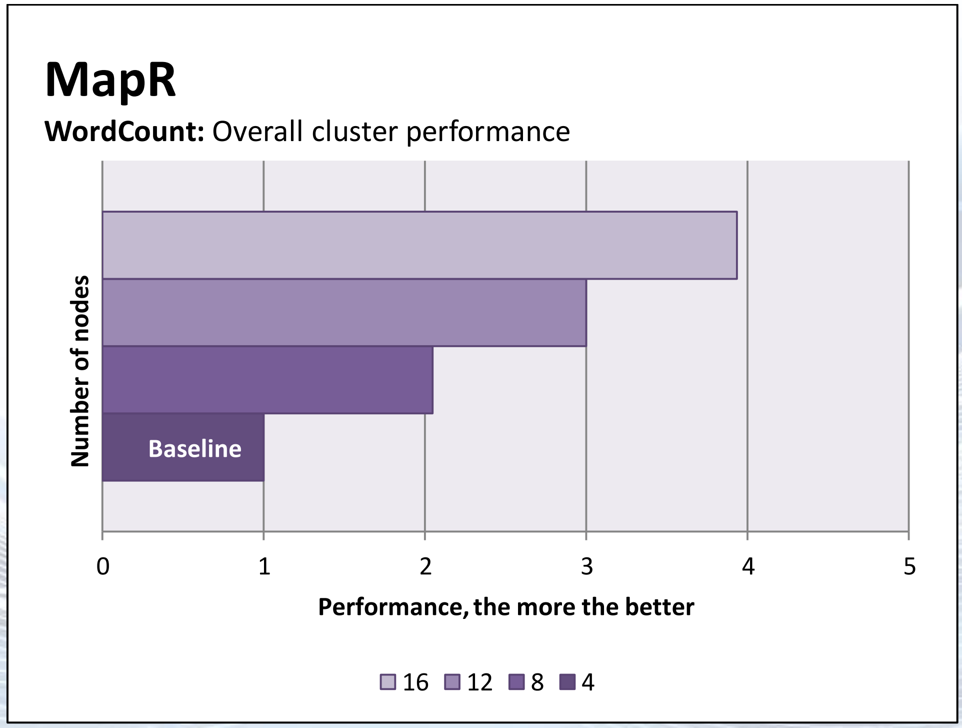


Figure 42. The overall performance of the MapR cluster in the WordCount benchmark

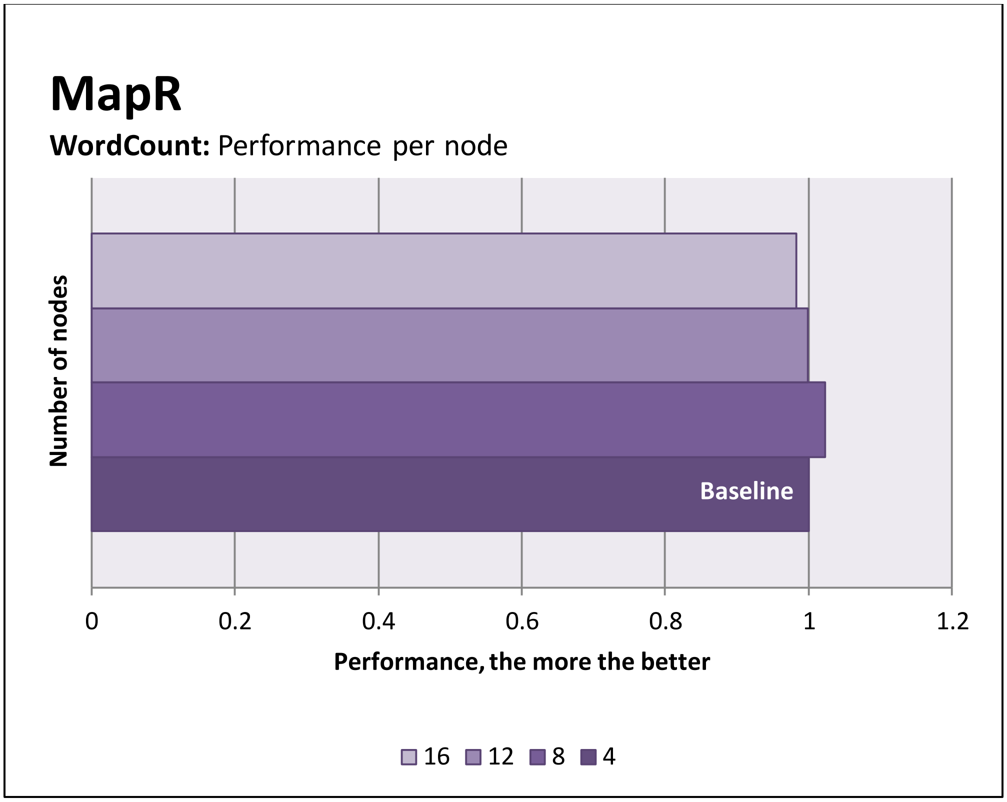


Figure 43. The performance of a single node of the MapR cluster in the WordCount benchmark

2. Hortonworks

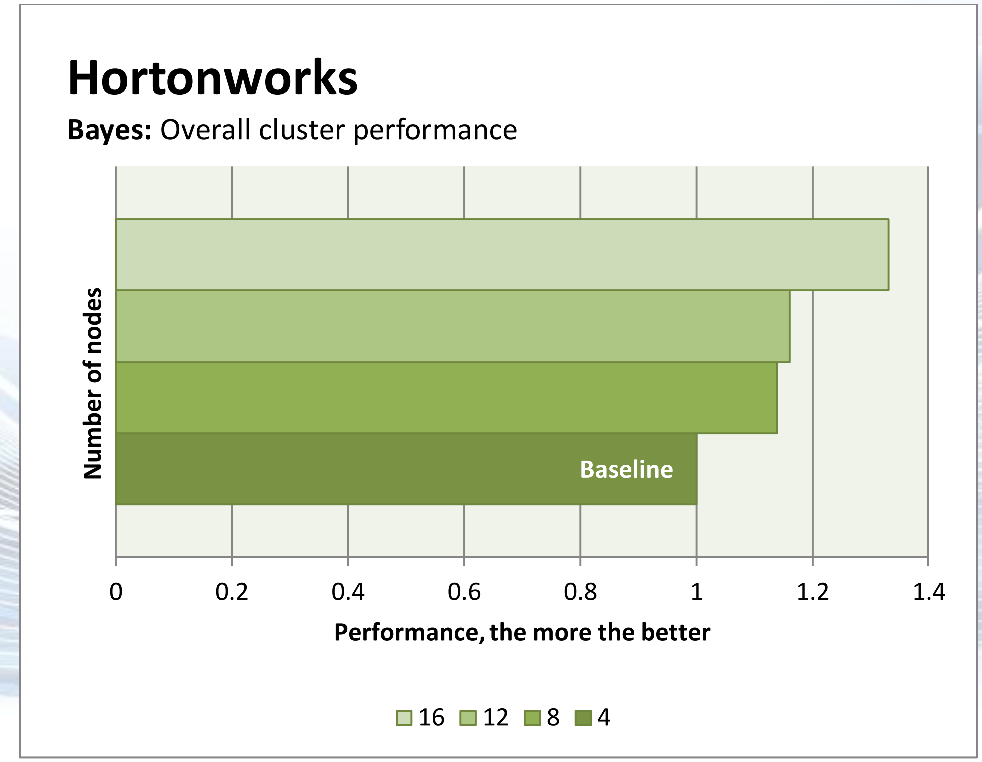


Figure 44. The overall performance of the Hortonworks cluster in the Bayes benchmark

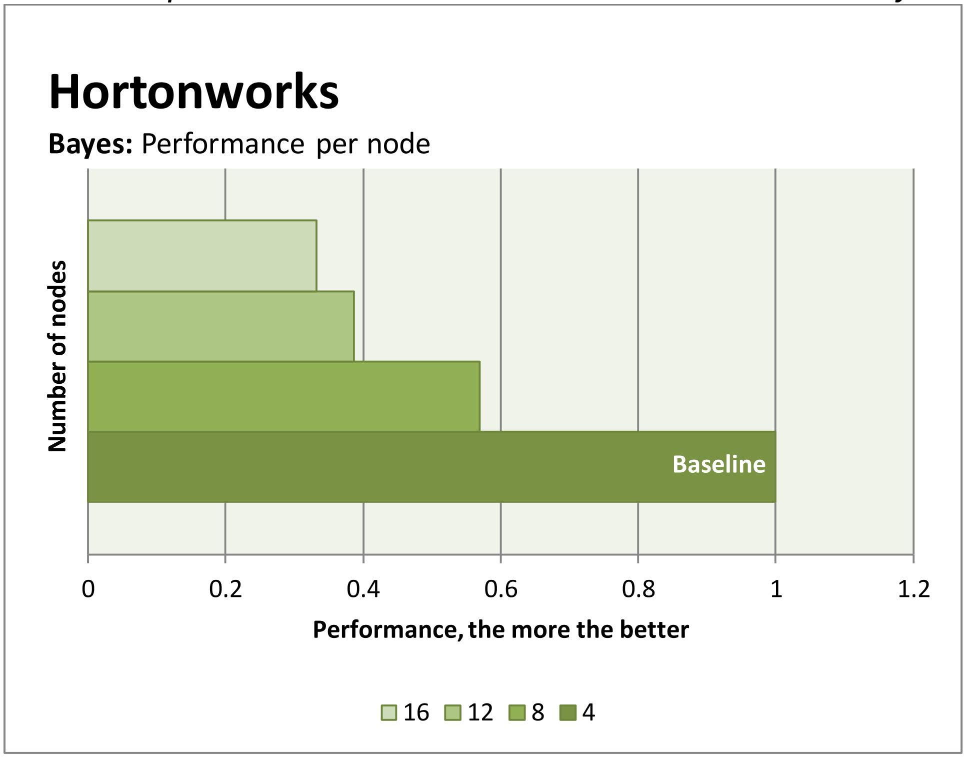


Figure 45. The performance of a single node of the Hortonworks cluster in the Bayes benchmark

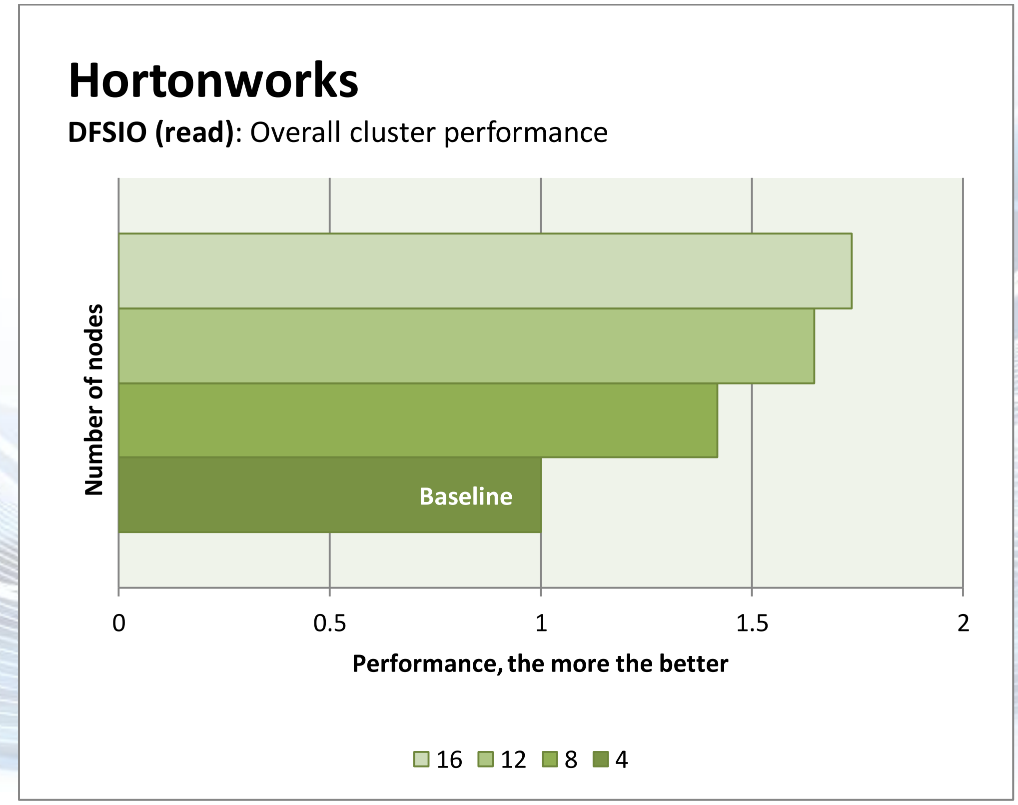


Figure 46. The overall performance of the Hortonworks cluster in the DFSIO (read) benchmark

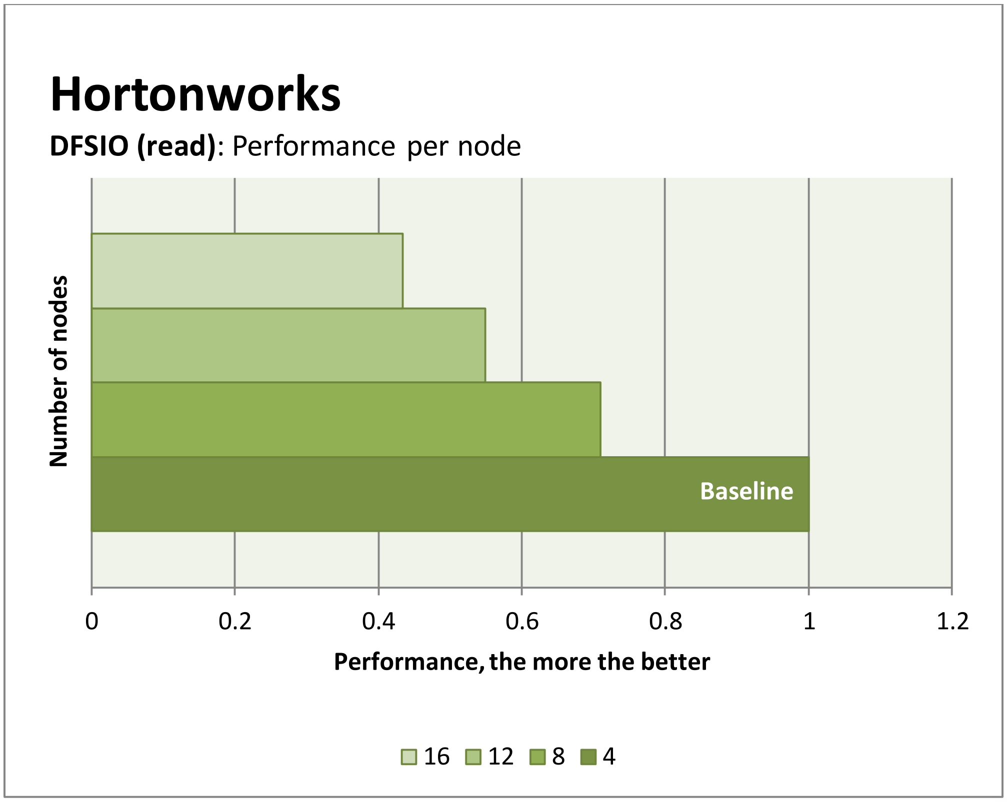


Figure 47. The performance of a single node of the Hortonworks cluster in the DFSIO (read) benchmark

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

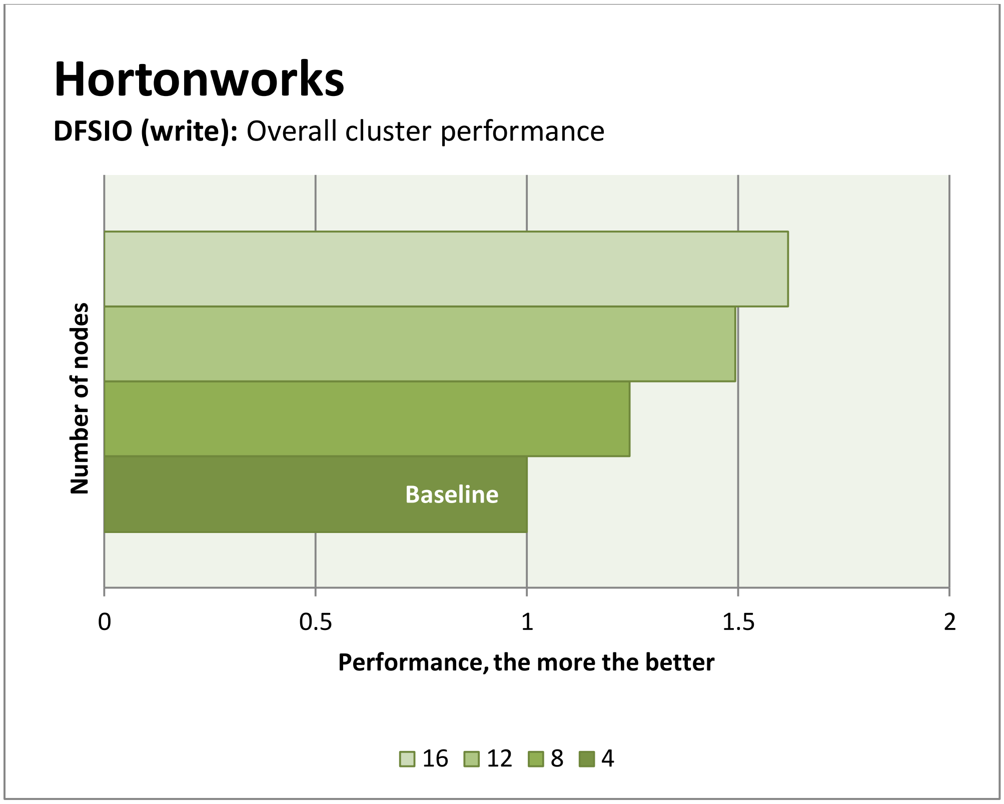


Figure 48. The overall performance of the Hortonworks cluster in the DFSIO (write) benchmark

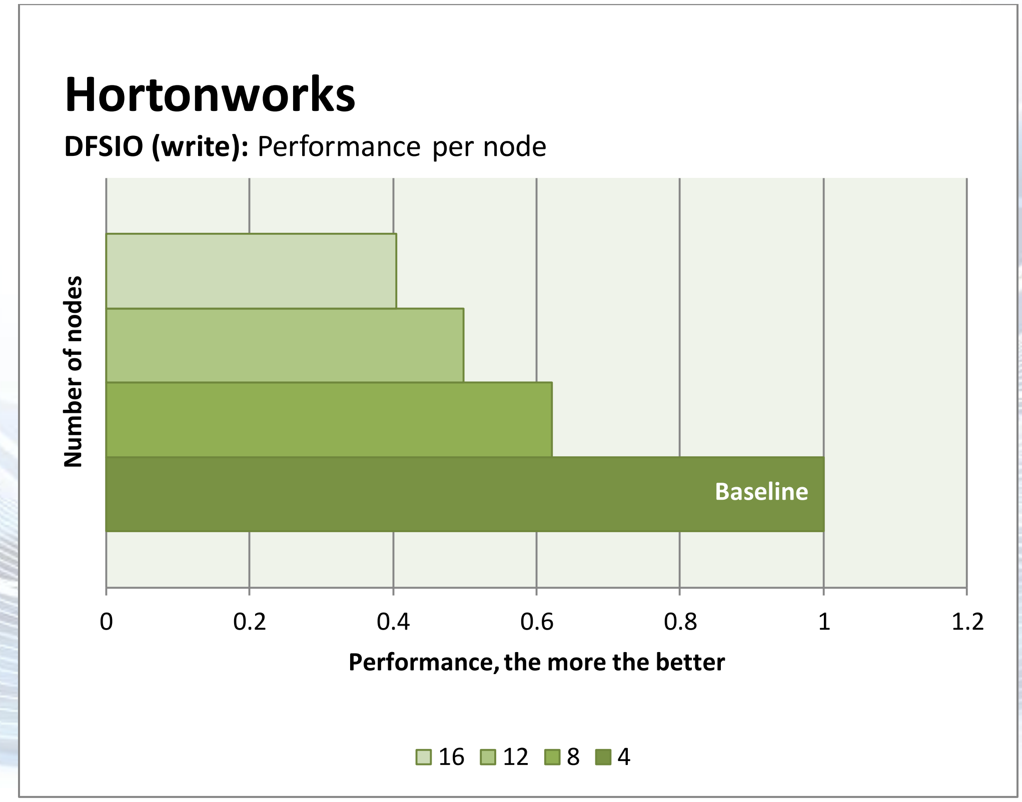


Figure 49. The performance of a single node of the Hortonworks cluster in the DFSIO (write) benchmark

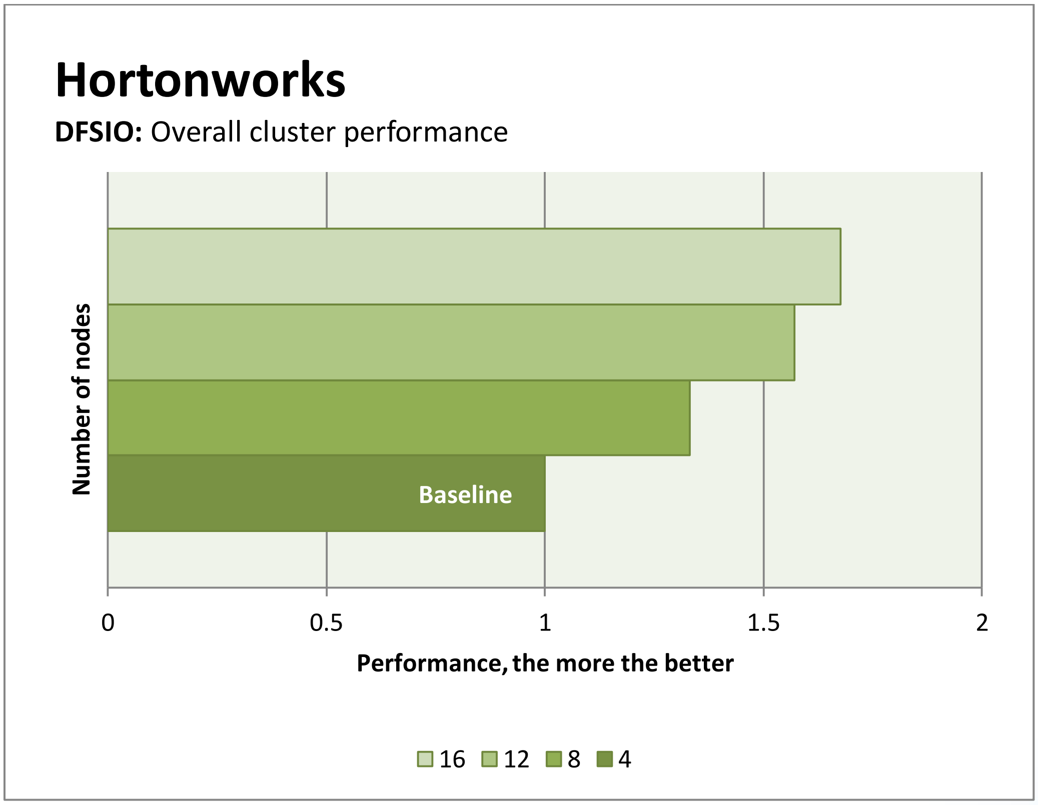


Figure 50. The overall performance of the Hortonworks cluster in the DFSIO benchmark

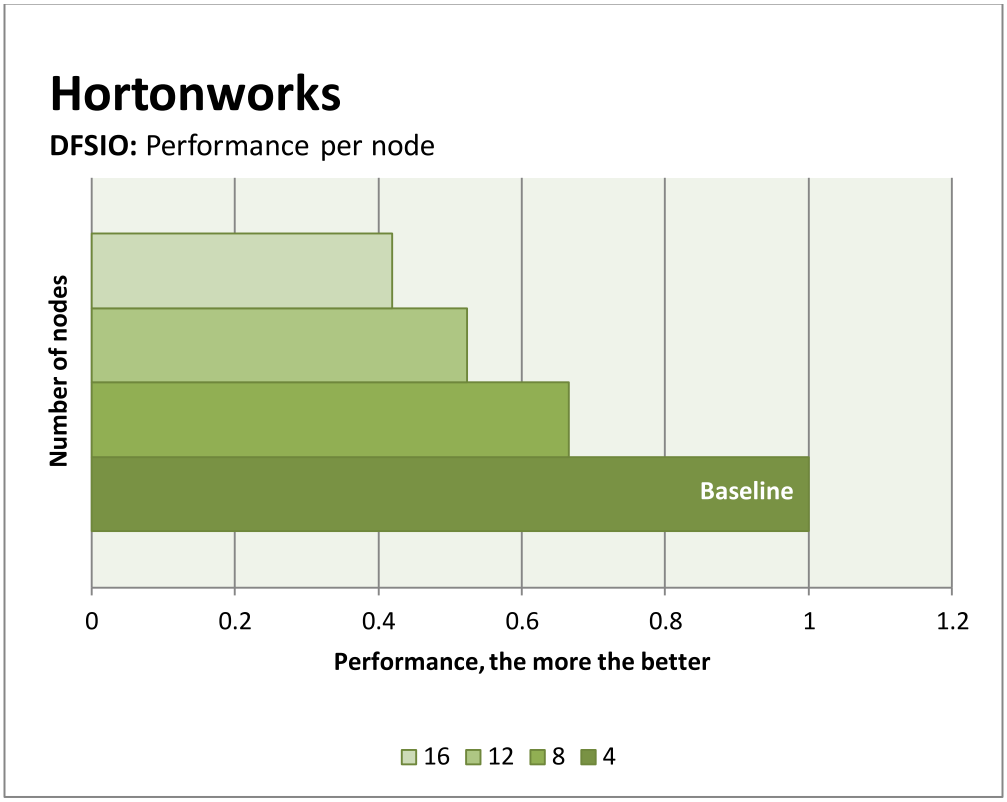


Figure 51. The performance of a single node of the Hortonworks cluster in the DFSIO benchmark

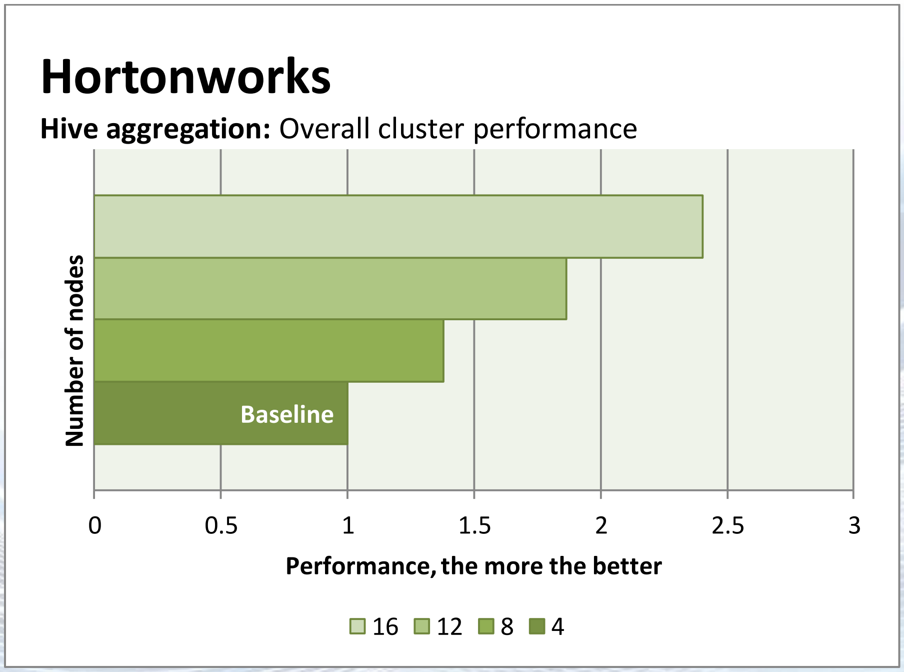


Figure 52. The overall performance of the Hortonworks cluster in the Hive aggregation benchmark

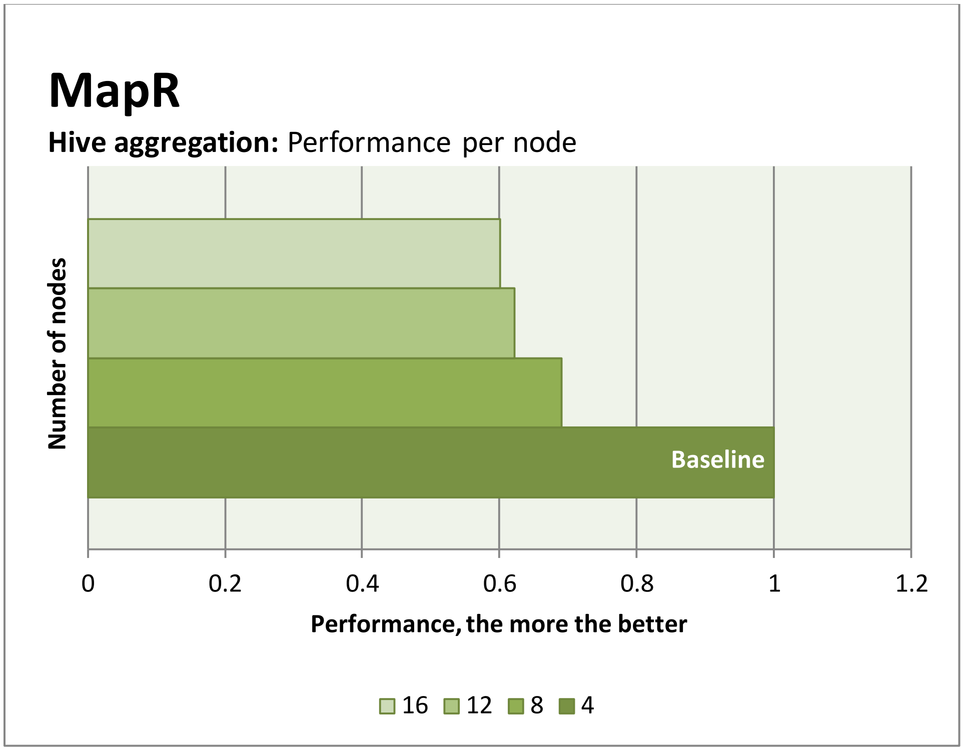


Figure 53. The performance of a single node of the Hortonworks cluster in the Hive aggregation benchmark

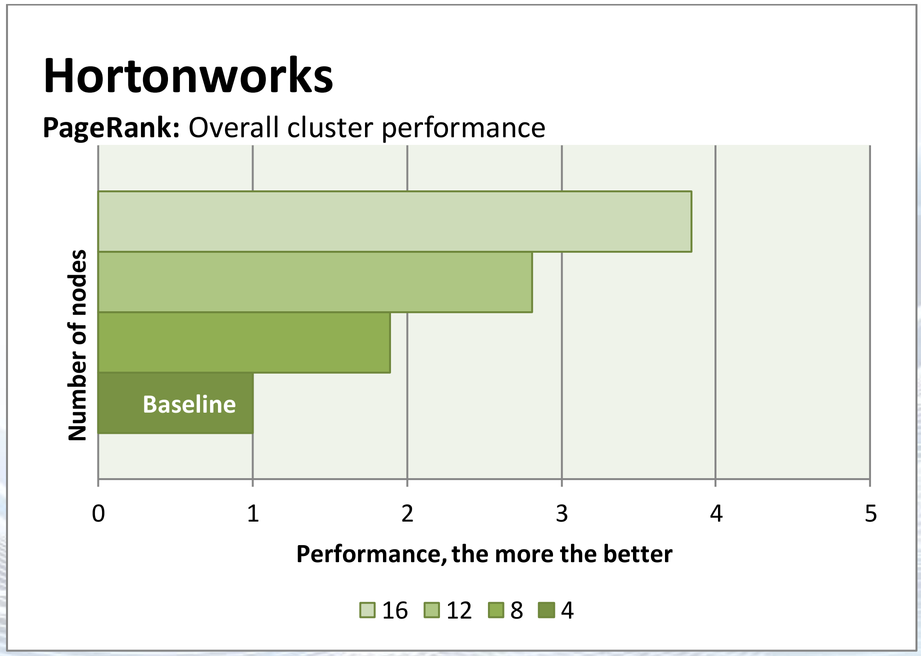


Figure 54. The overall performance of the Hortonworks cluster in the PageRank benchmark

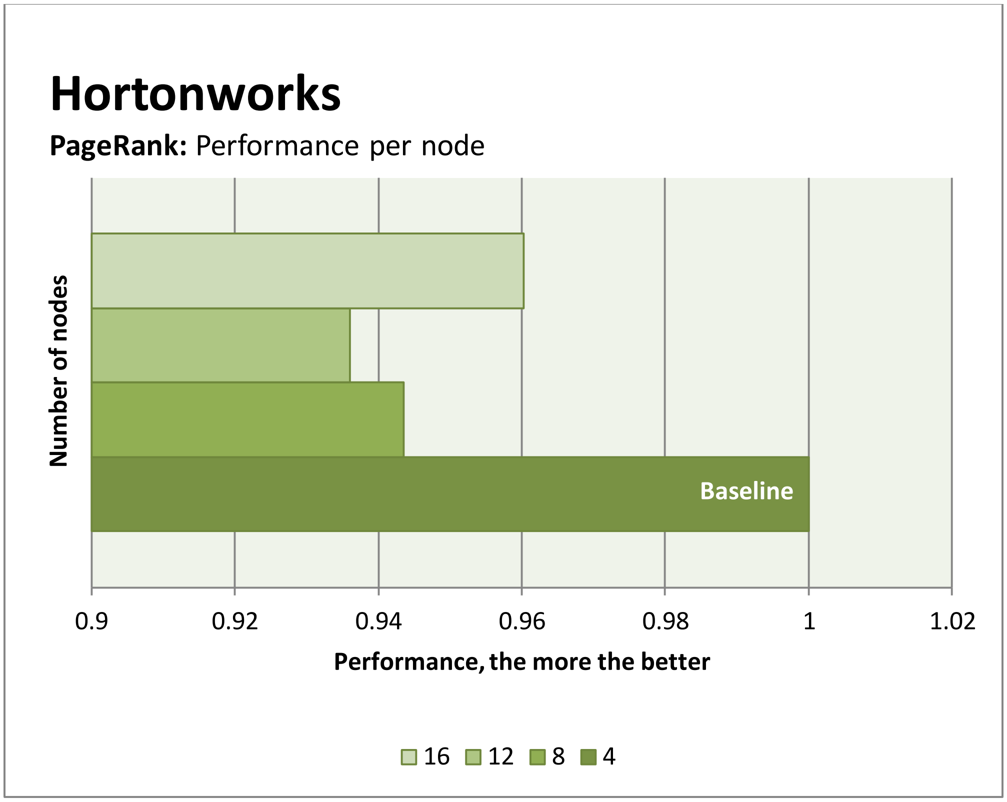


Figure 55. The performance of a single node of the Hortonworks cluster in the PageRank benchmark

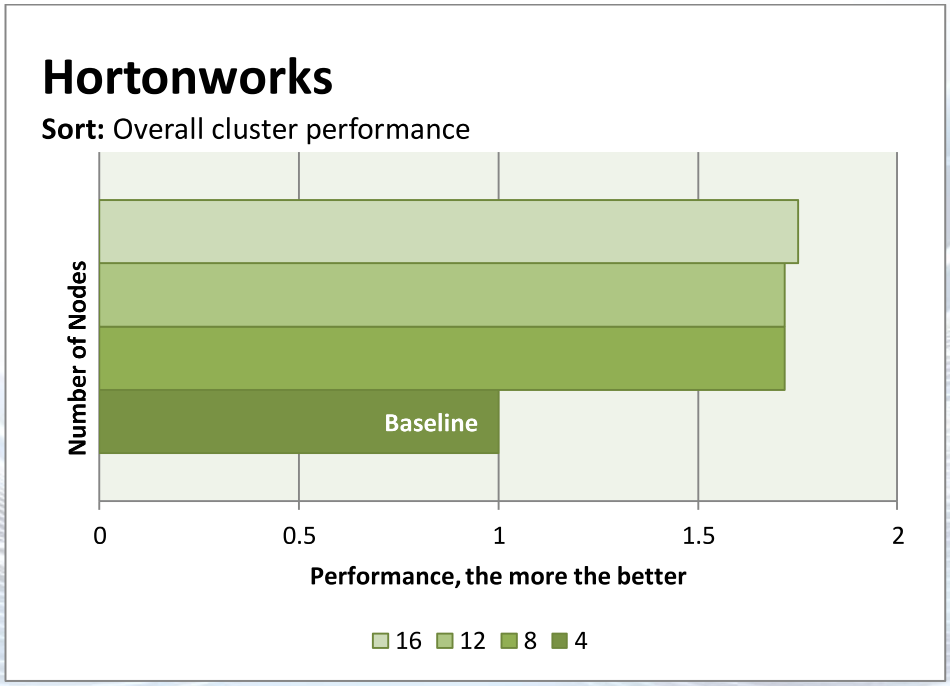


Figure 56. The overall performance of the Hortonworks cluster in the Sort benchmark

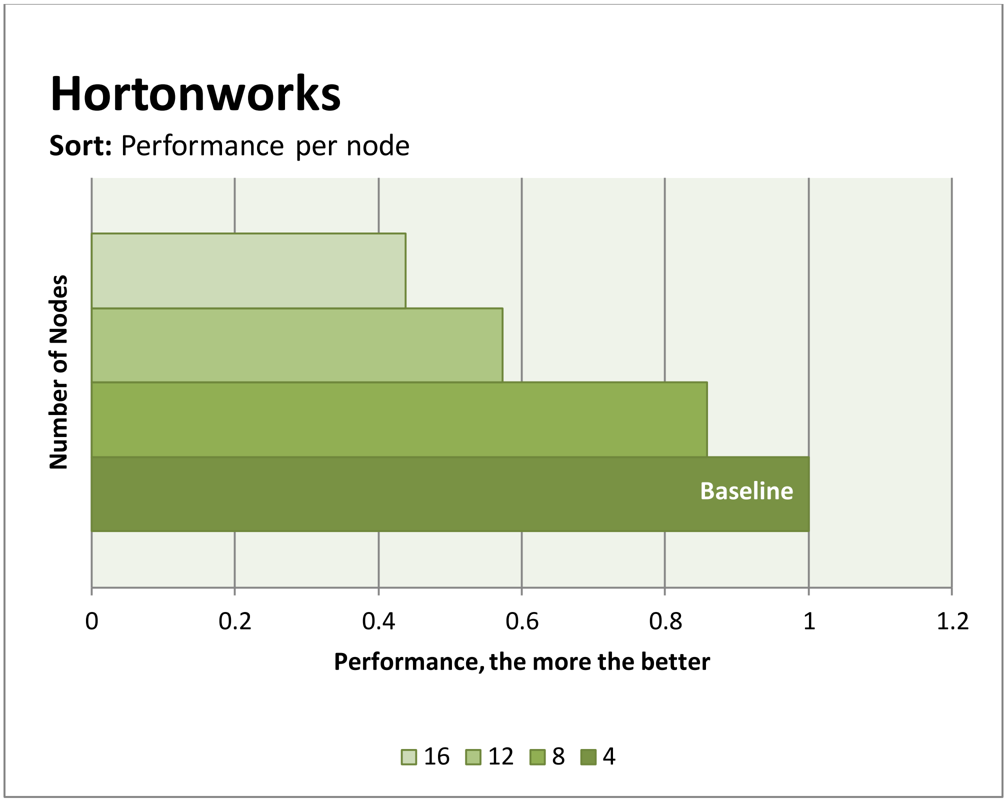


Figure 57. The performance of a single node of the Hortonworks cluster in the Sort benchmark

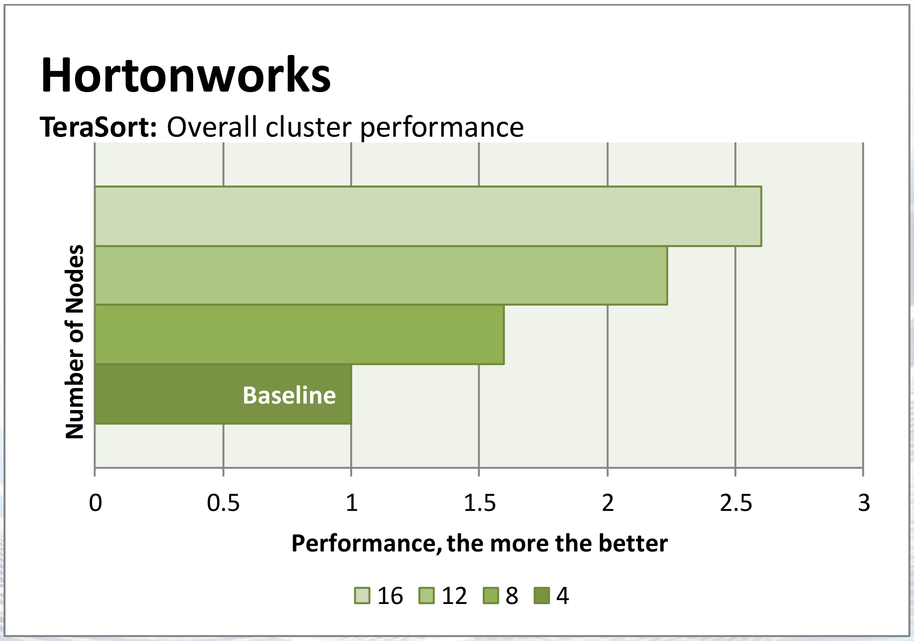


Figure 58. The overall performance of the Hortonworks cluster in the TeraSort benchmark

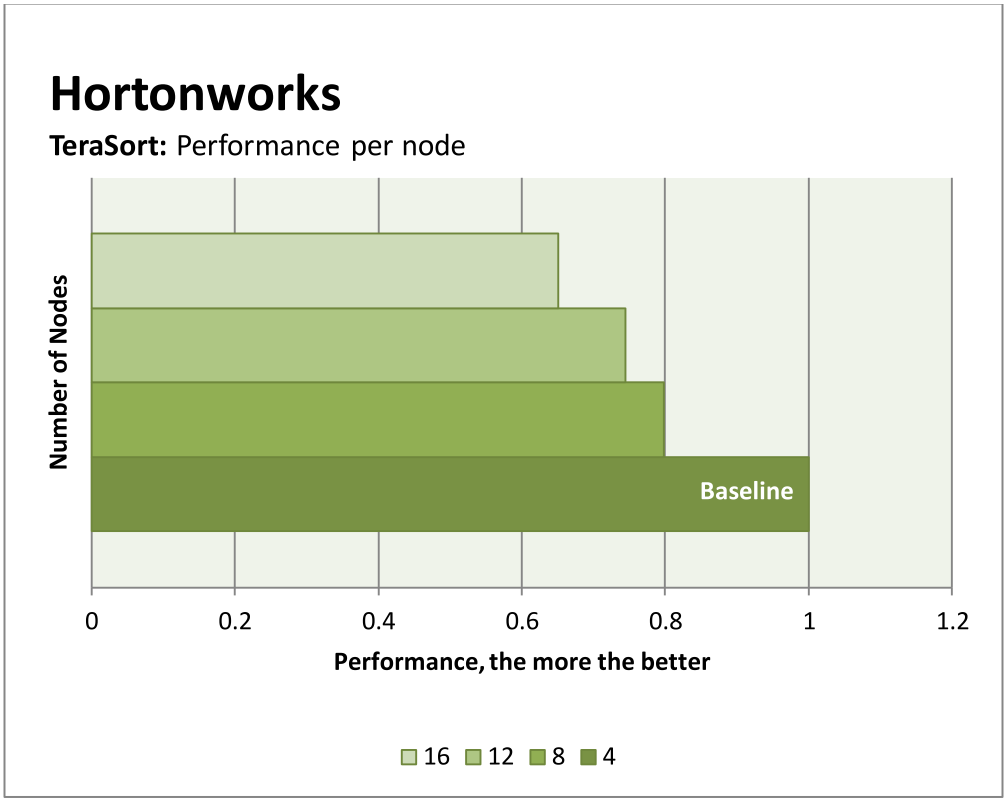


Figure 59. The performance of a single node of the Hortonworks cluster in the TeraSort benchmark

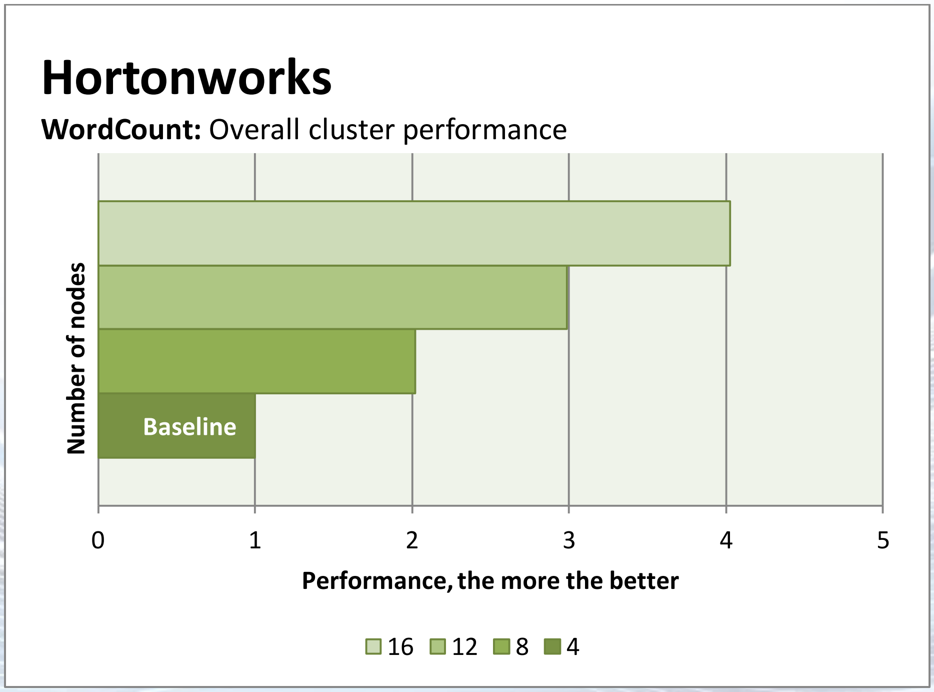


Figure 60. The overall performance of the Hortonworks cluster in the WordCount benchmark

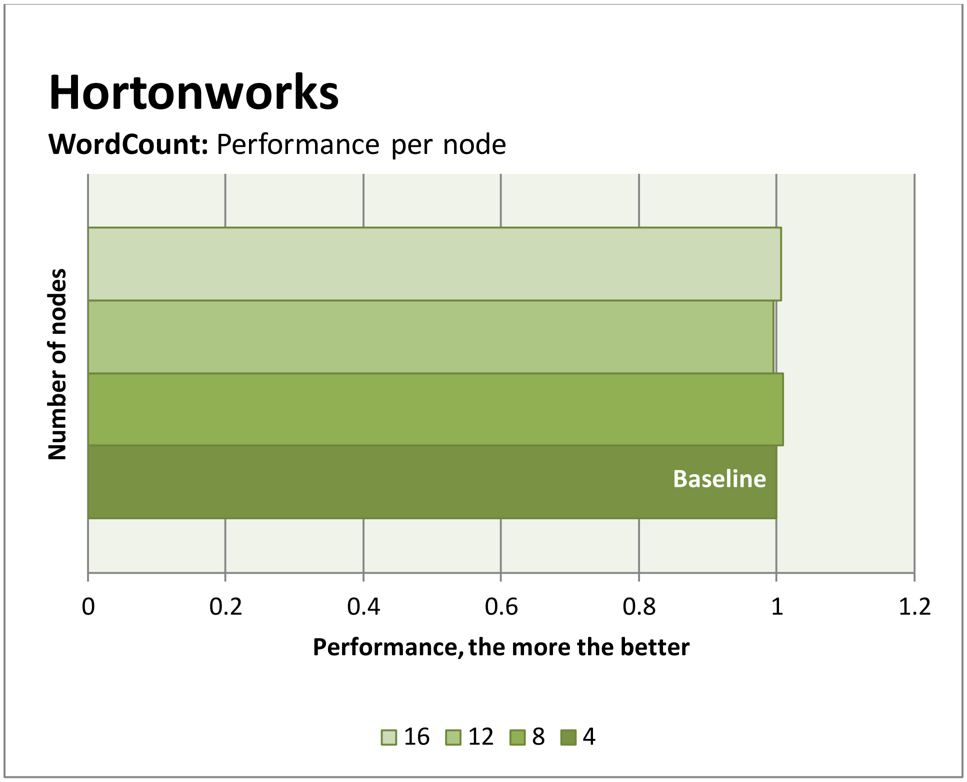


Figure 61. The performance of a single node of the Hortonworks cluster in the WordCount benchmark

3. Cloudera

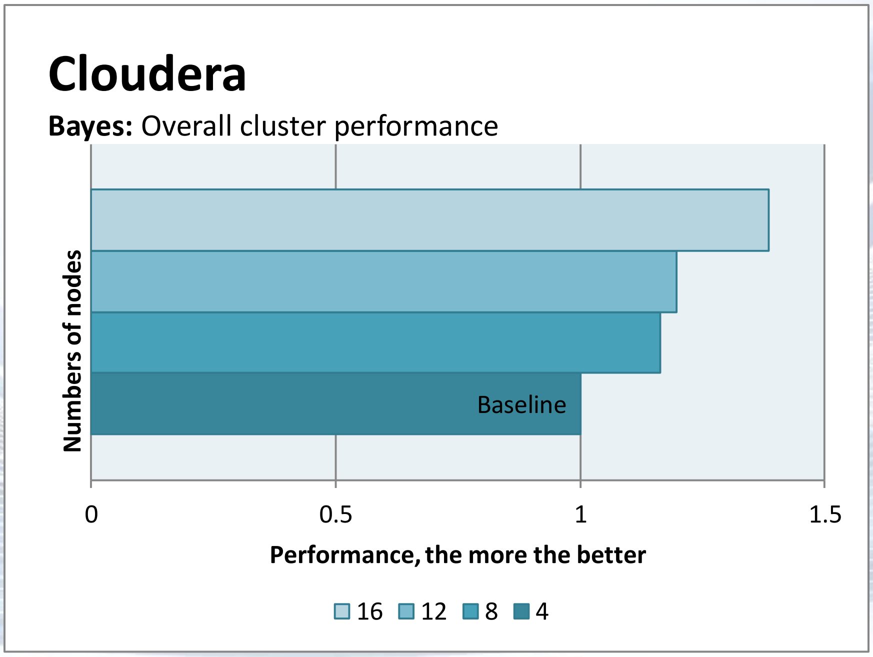


Figure 62. The overall performance of the Cloudera cluster in the Bayes benchmark

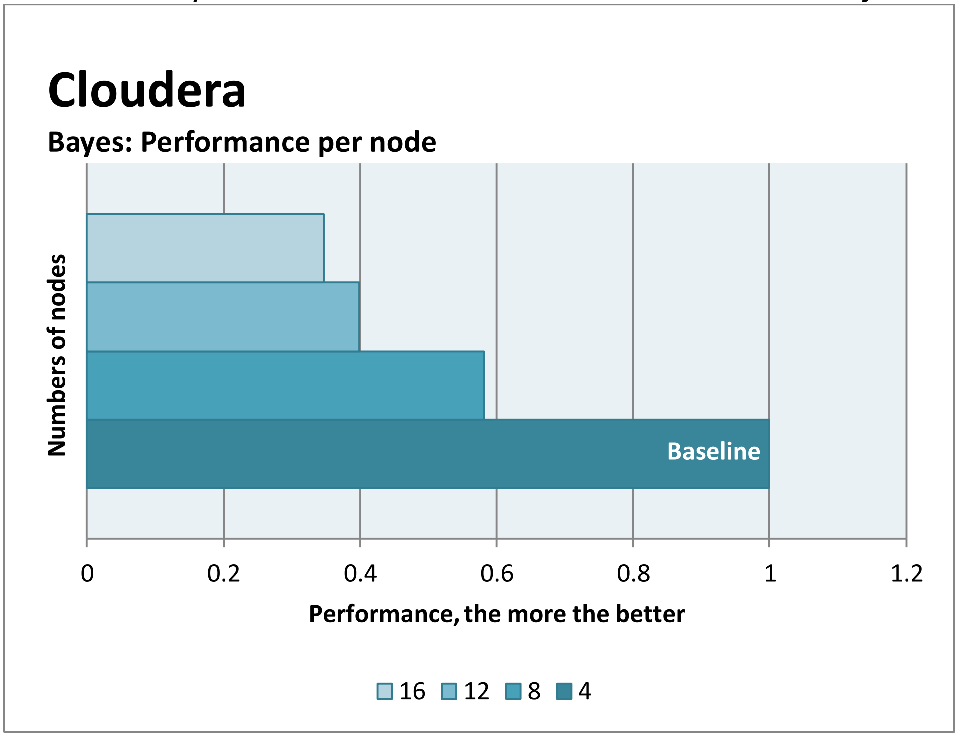


Figure 63. The performance of a single node of the Cloudera cluster in the Bayes benchmark

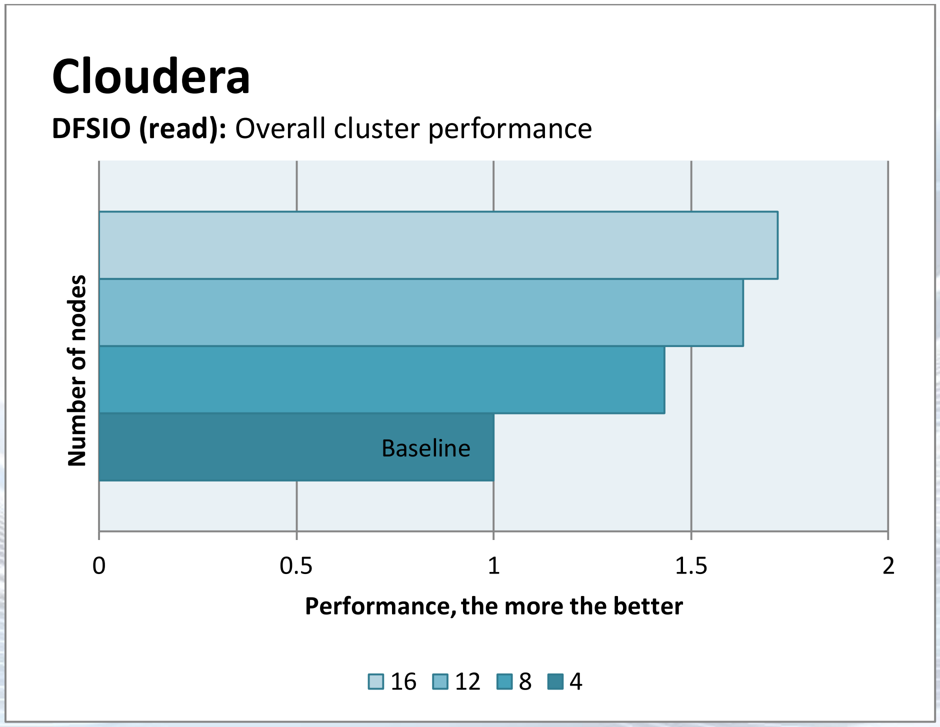


Figure 64. The overall performance of the Cloudera cluster in the DFSIO (read) benchmark

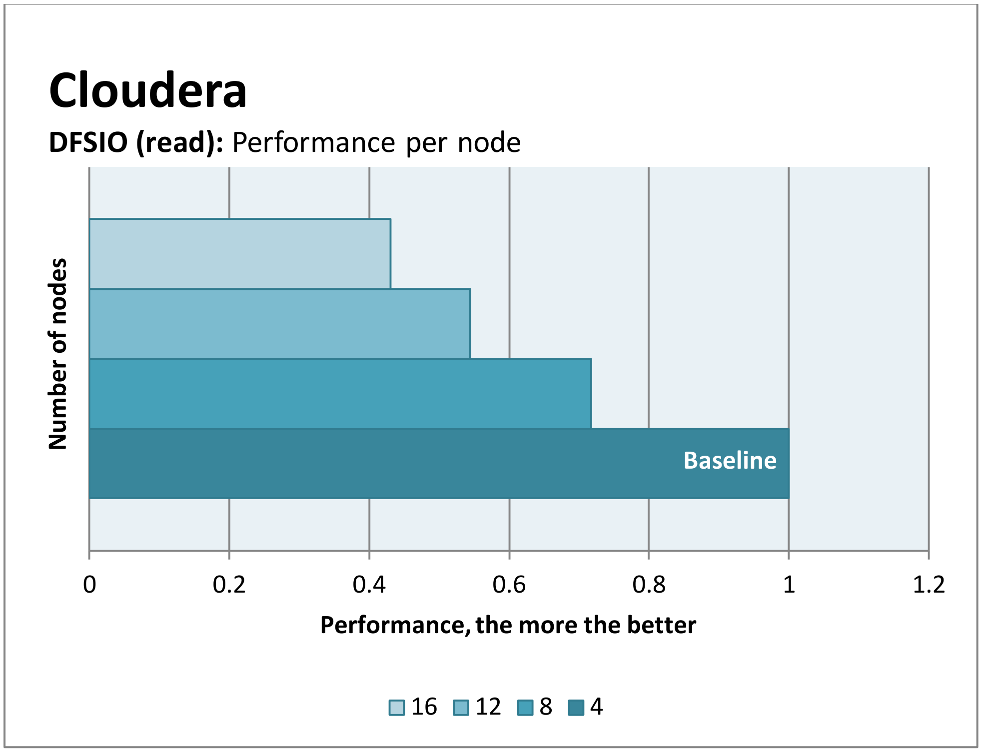


Figure 65. The performance of a single node of the Cloudera cluster in the DFSIO (read) benchmark

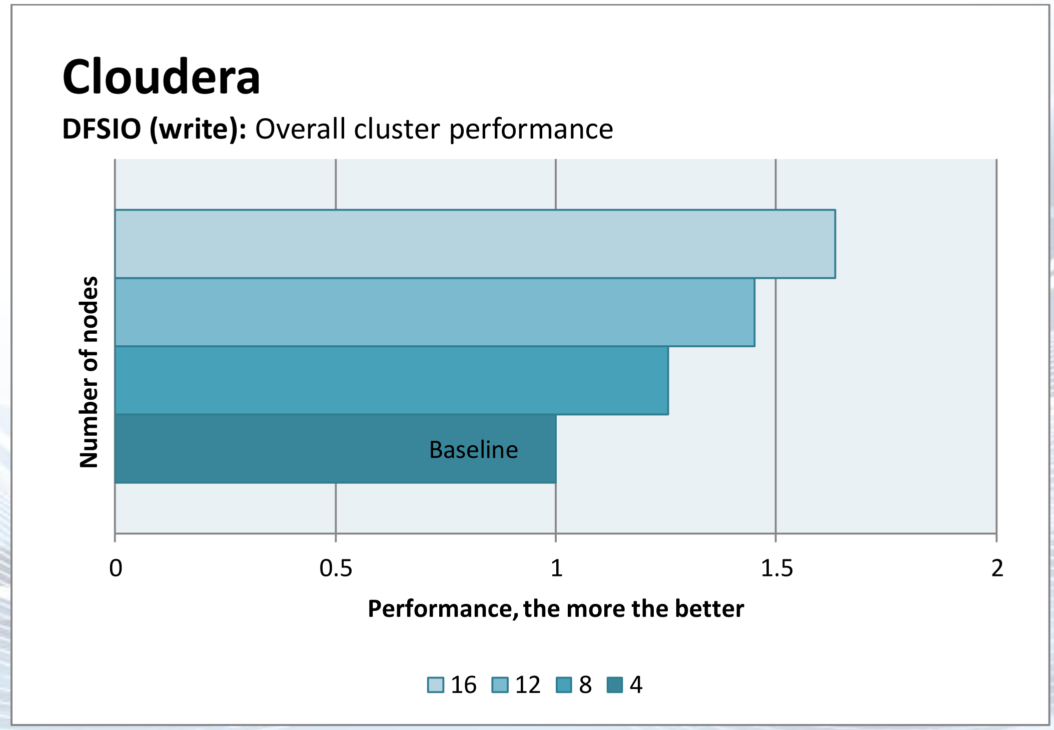


Figure 66. The overall performance of the Cloudera cluster in the DFSIO (write) benchmark

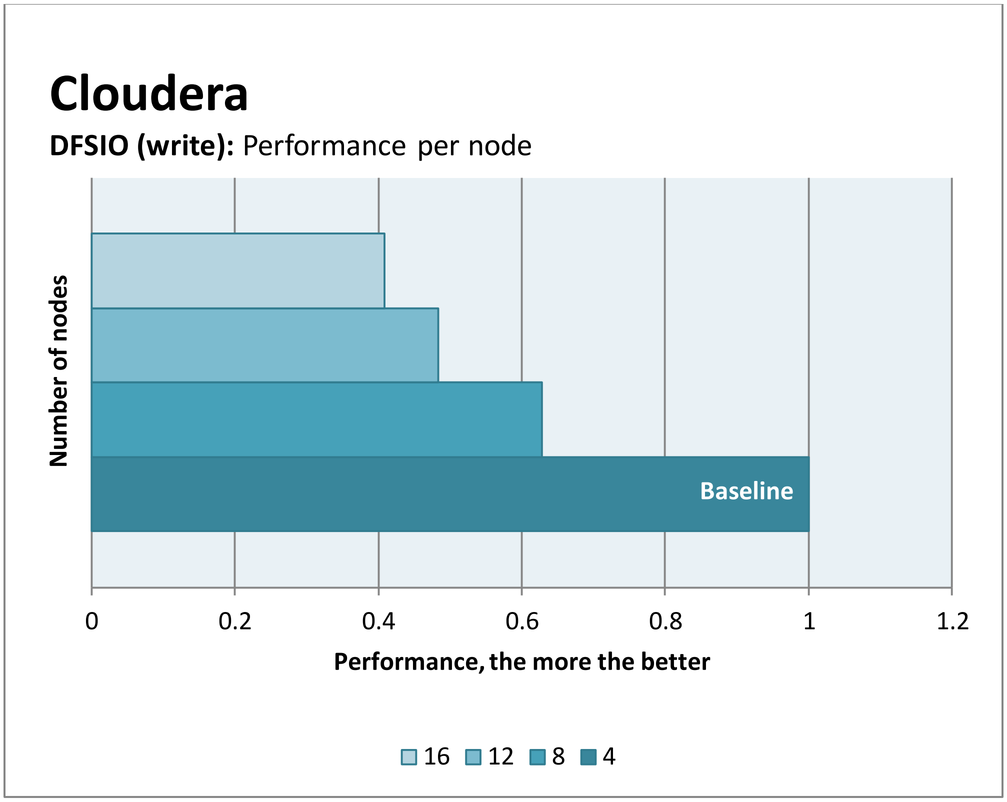


Figure 67. The performance of a single node of the Cloudera cluster in the DFSIO (write) benchmark

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros

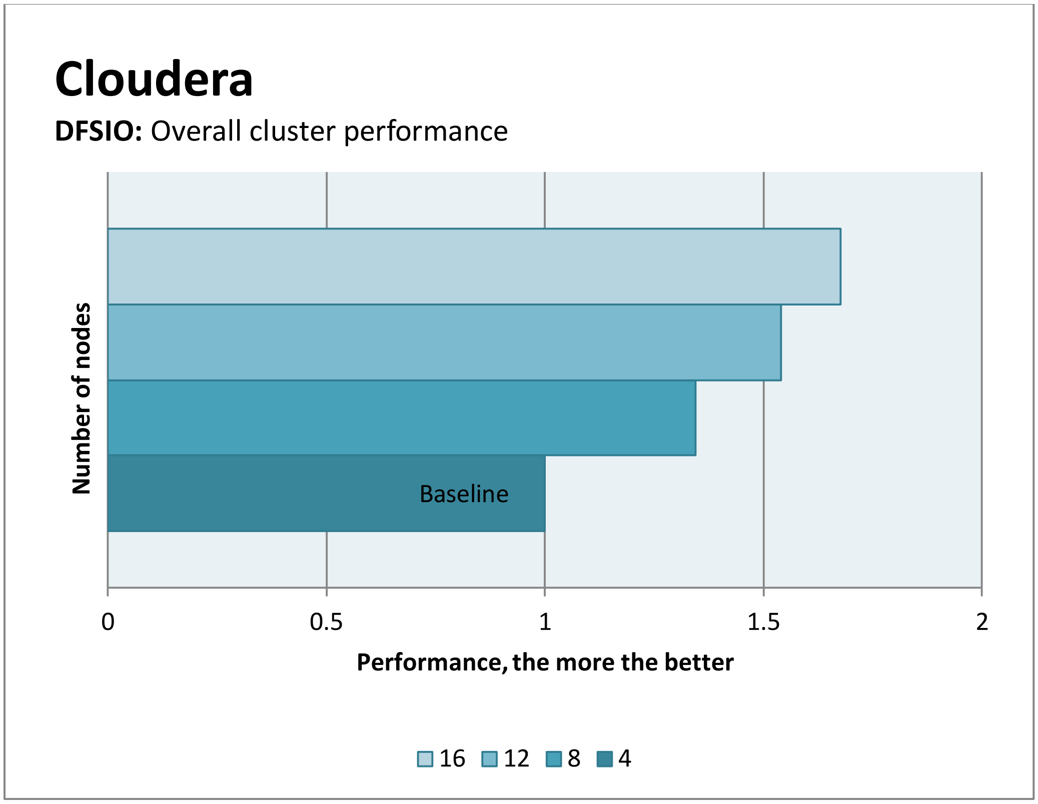


Figure 68. The overall performance of the Cloudera cluster in the DFSIO benchmark

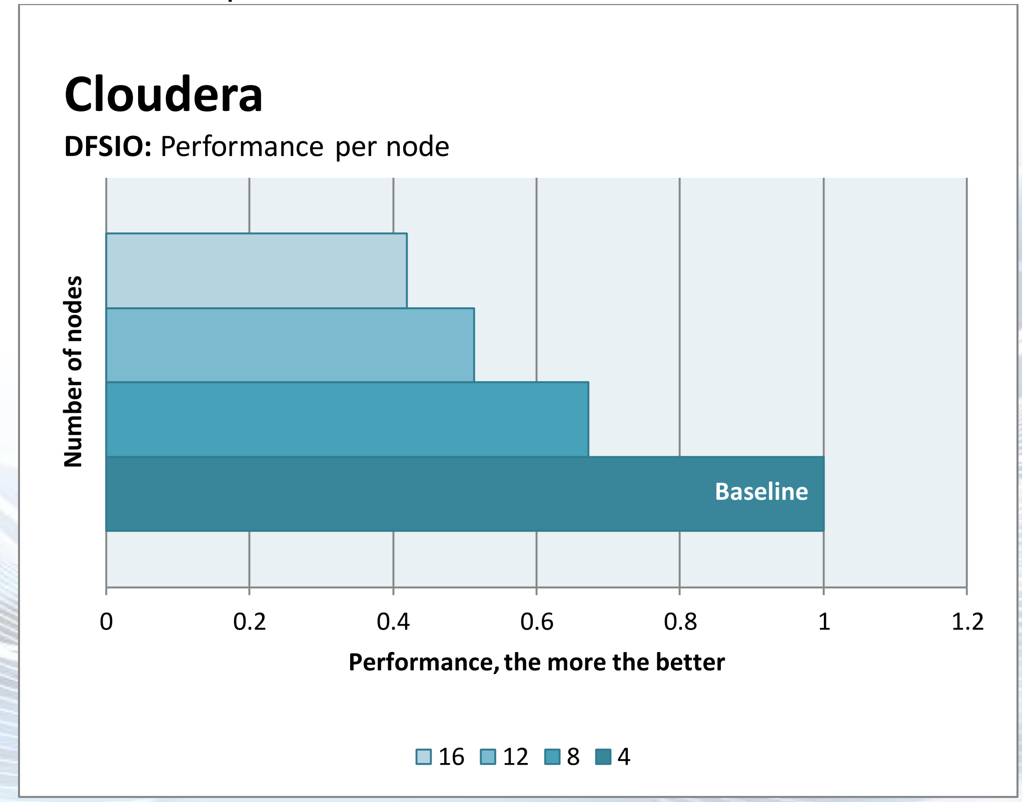


Figure 69. The performance of a single node of the Cloudera cluster in the DFSIO benchmark

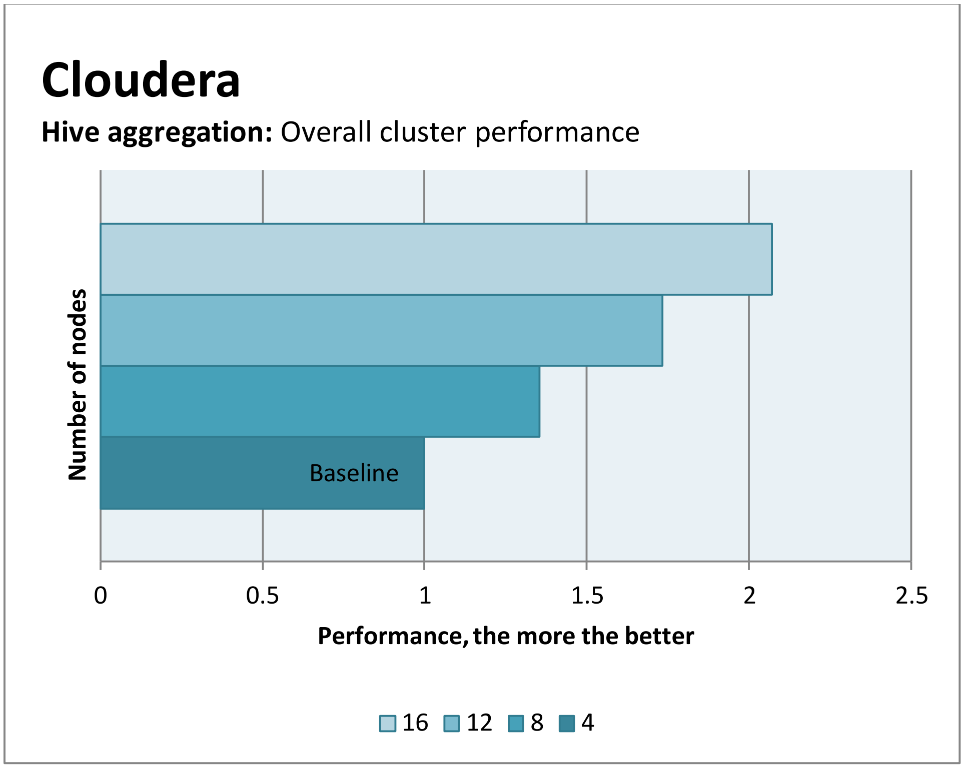


Figure 70. The overall performance of the Cloudera cluster in the Hive aggregation benchmark

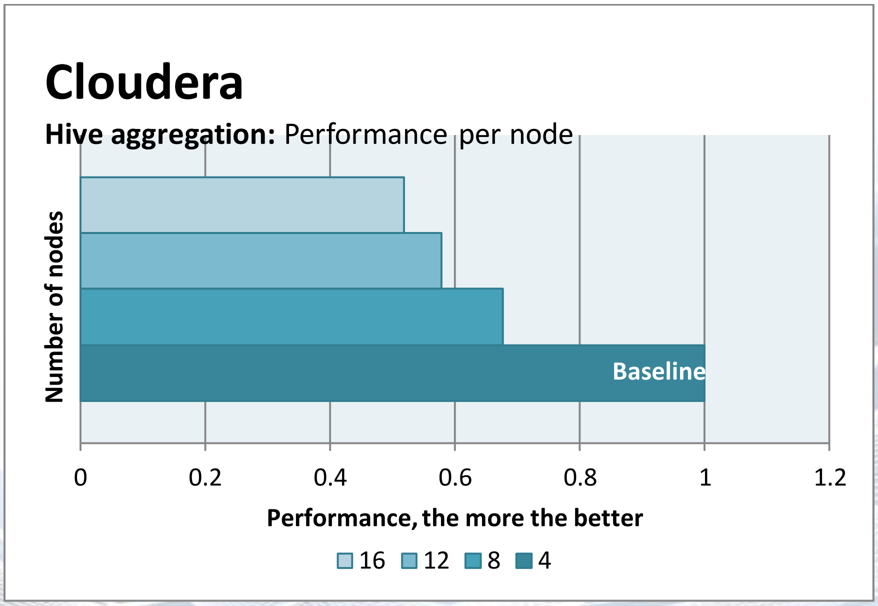


Figure 71. The performance of a single node of the Cloudera cluster in the Hive aggregation benchmark

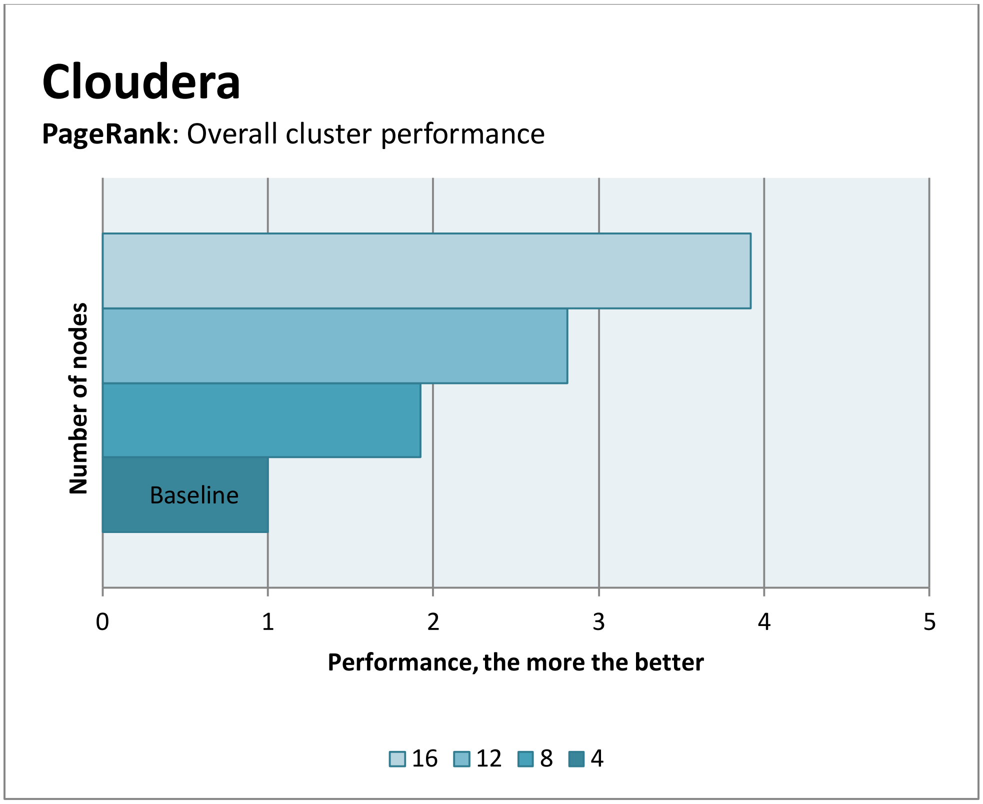


Figure 72. The overall performance of the Cloudera cluster in the PageRank benchmark

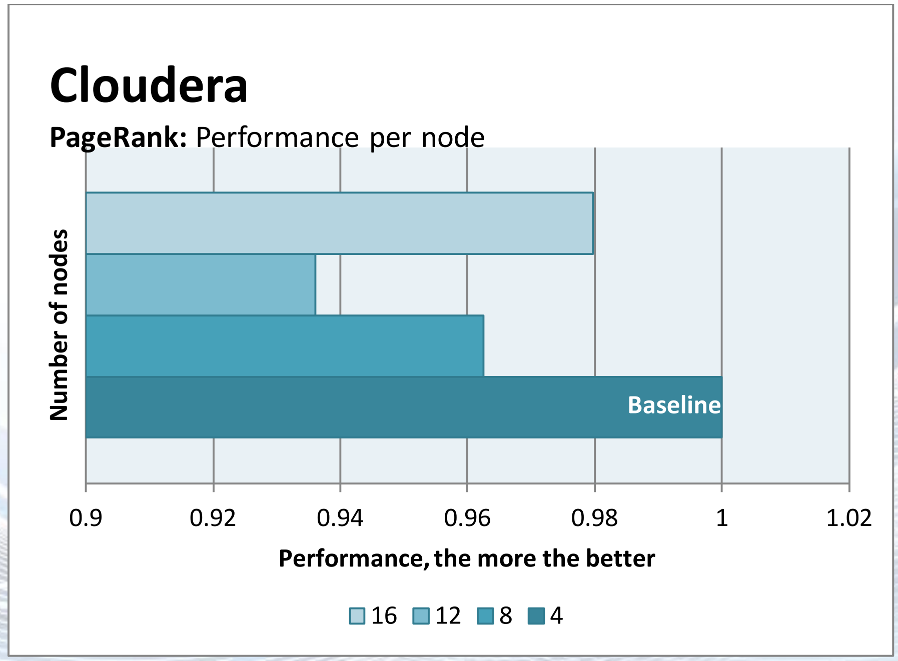


Figure 73. The performance of a single node of the Cloudera cluster in the PageRank benchmark

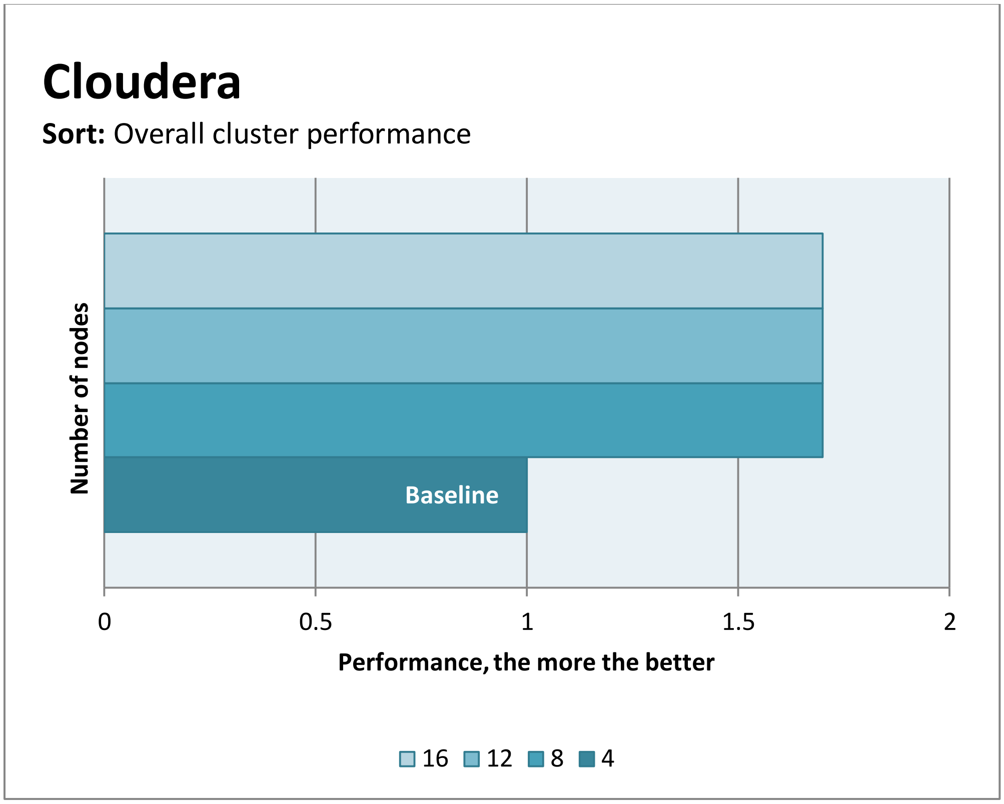


Figure 74. The overall performance of the Cloudera cluster in the Sort benchmark

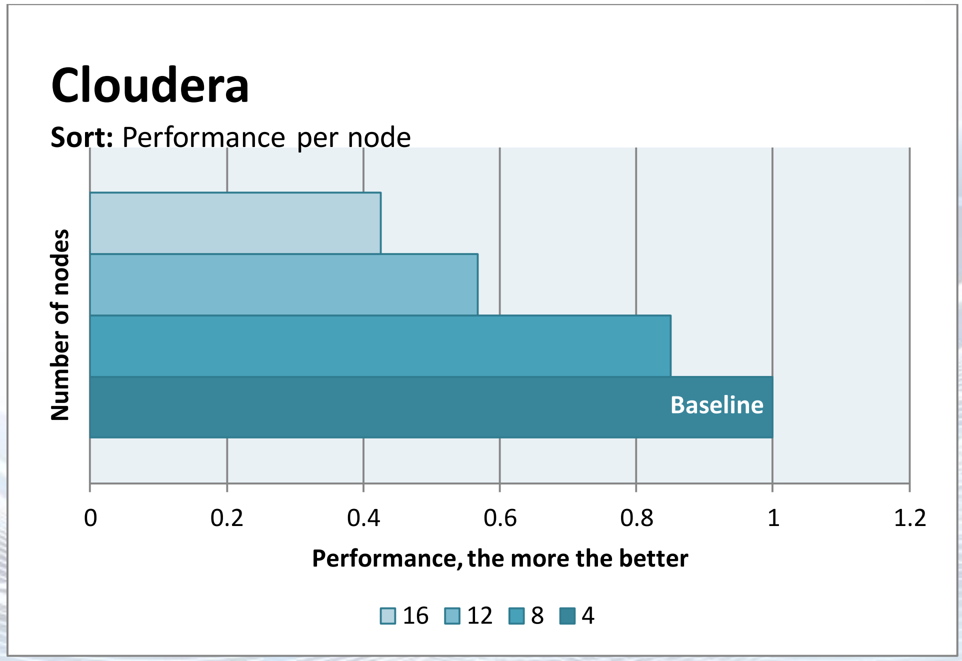


Figure 75. The performance of a single node of the Cloudera cluster in the Sort benchmark

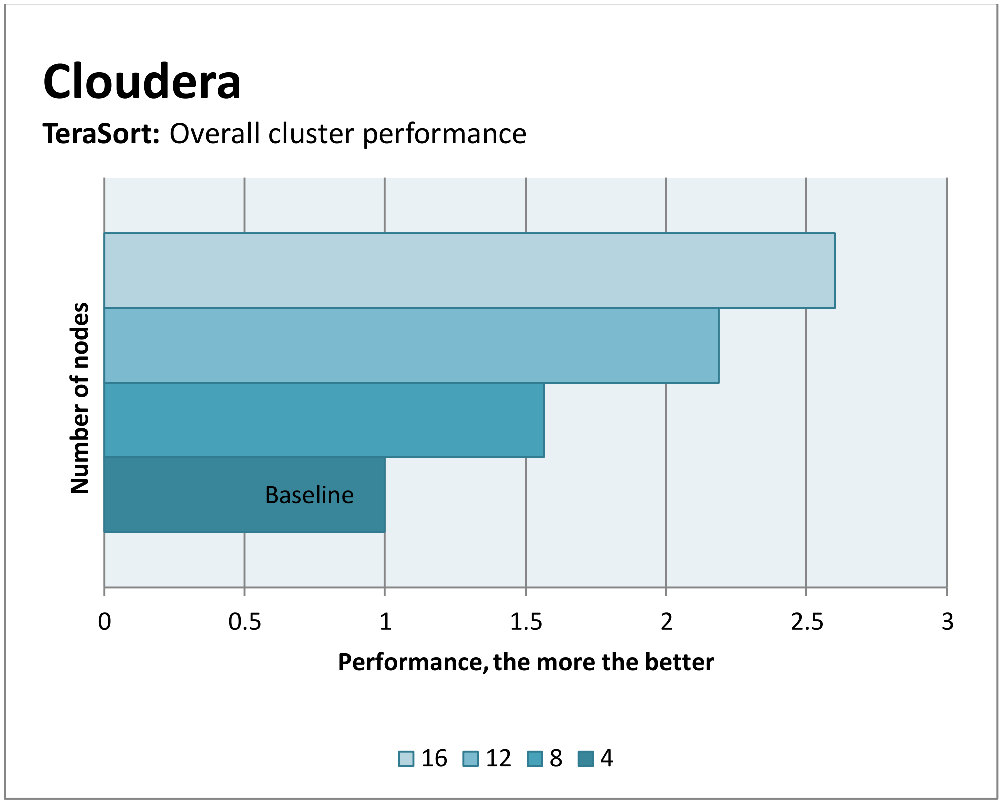


Figure 76. The overall performance of the Cloudera cluster in the TeraSort benchmark

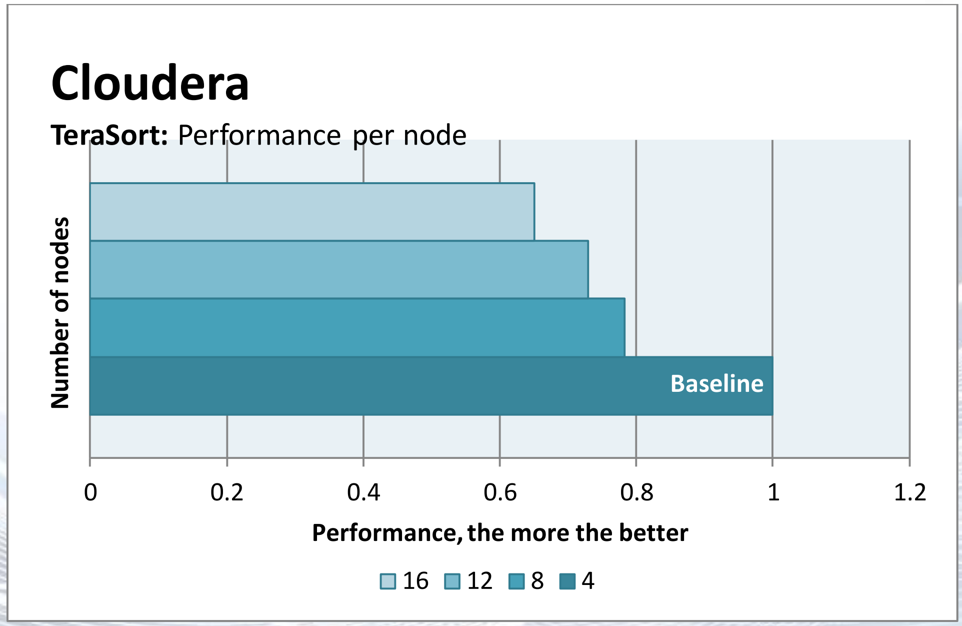


Figure 77. The performance of a single node of the Cloudera cluster in the TeraSort benchmark

+1 650 395-7002

engineering@altoros.com
www.altoros.com | twitter.com/altoros

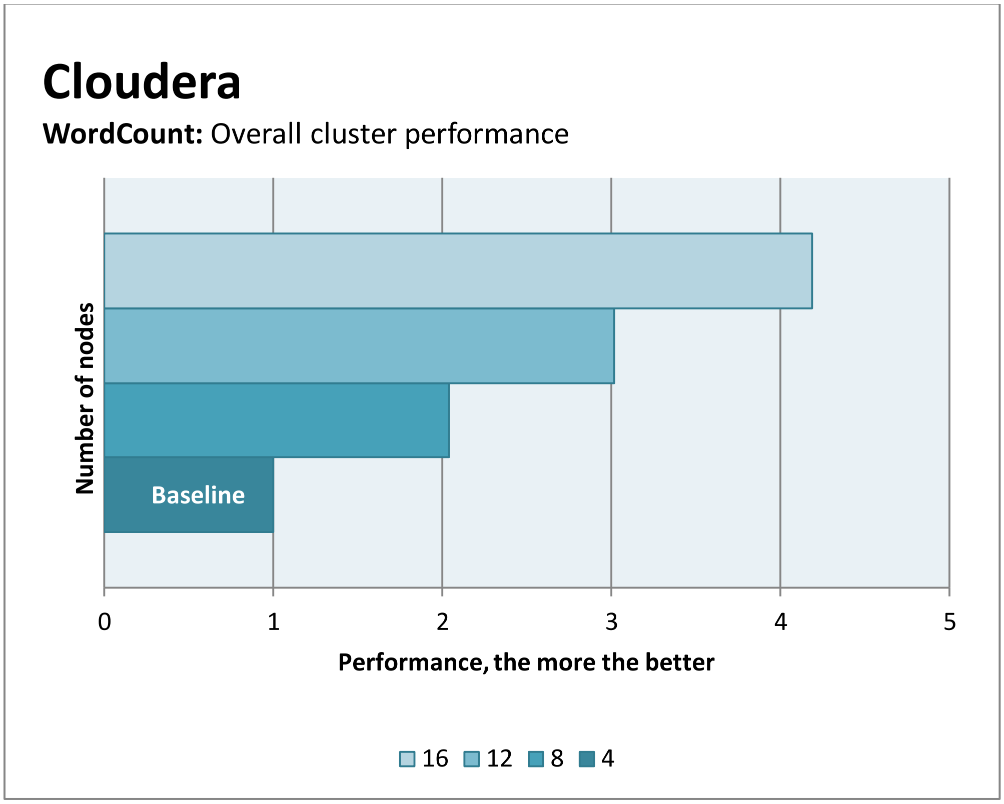


Figure 78. The overall performance of the Cloudera cluster in the WordCount benchmark

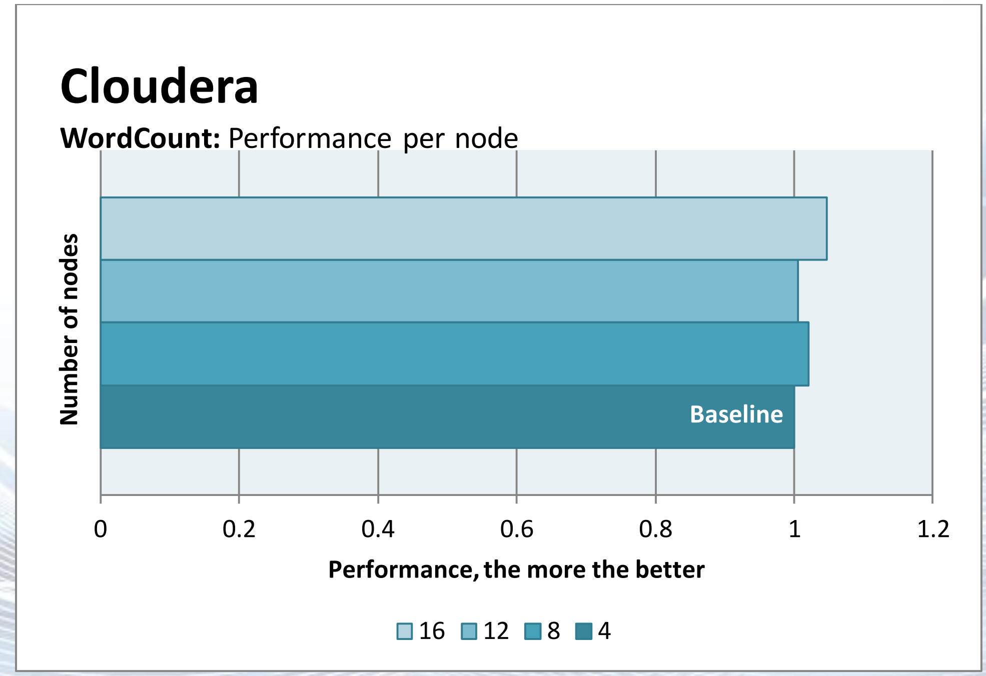


Figure 79. The performance of a single node of the Cloudera cluster in the WordCount benchmark

Appendix E: Disk Benchmarking

The line charts below demonstrate performance of the disks.

1. DFSIO (read) benchmark

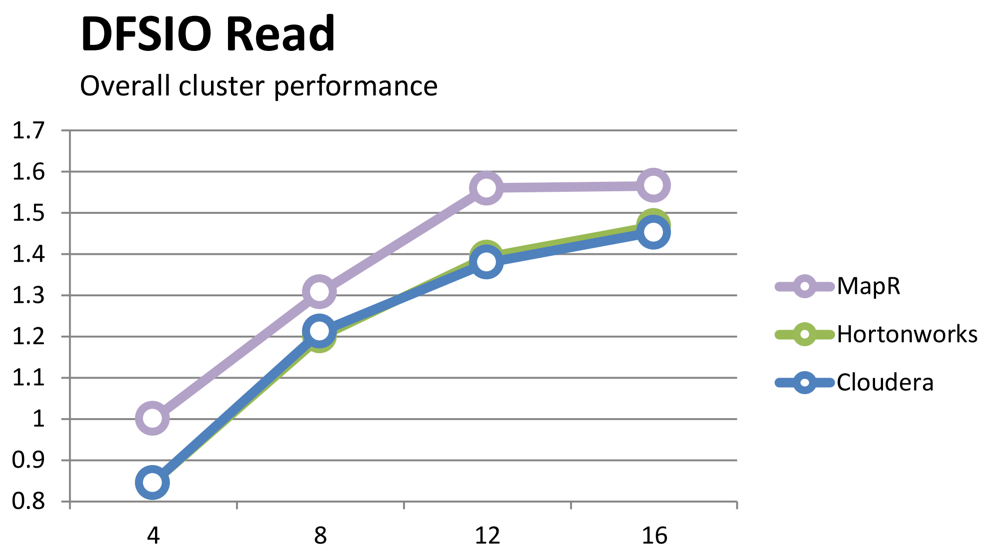


Figure 80. The overall performance of each distribution in the DFSIO-read benchmark, sectioned by the cluster size

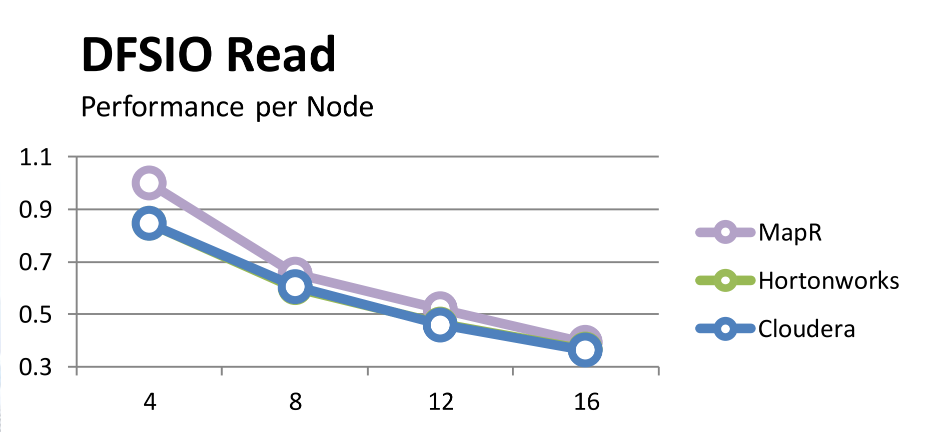


Figure 81. The performance of a single node of each distribution in the DFSIO-read benchmark, sectioned by the cluster size

2. DFSIO (write) benchmark

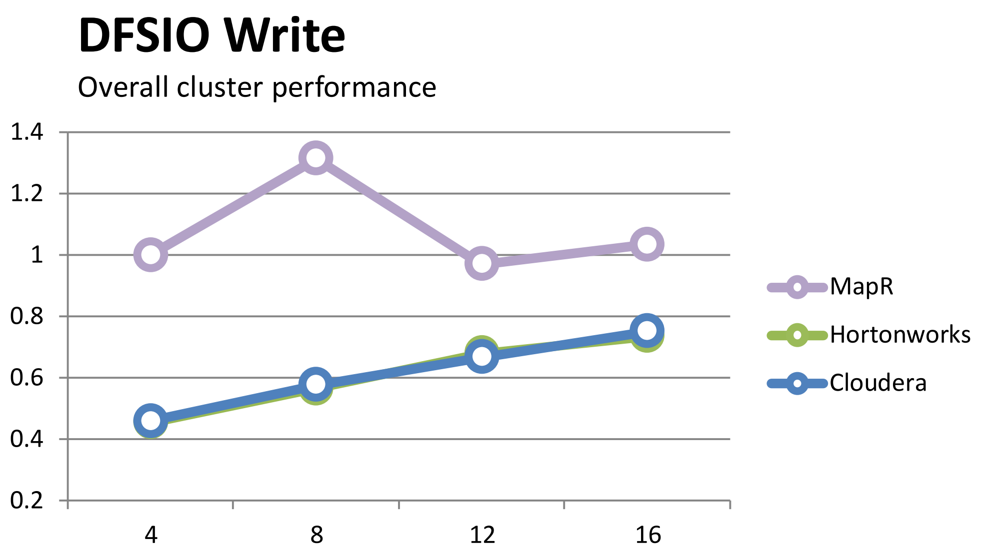


Figure 82. The overall performance of each distribution in the DFSIO-write benchmark, sectioned by the cluster size

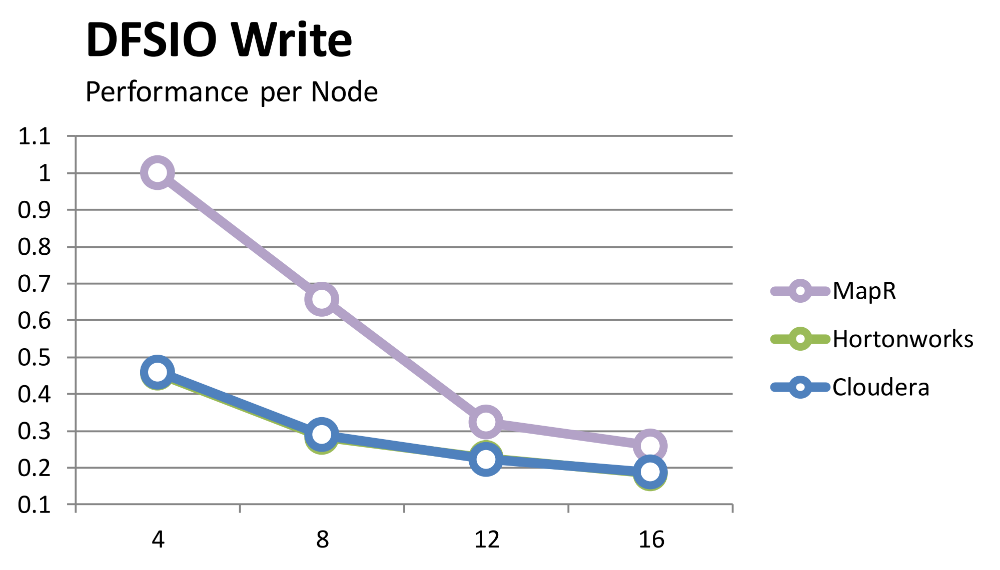


Figure 83. The performance of a single node of each distribution in the DFSIO-write benchmark, sectioned by the cluster size

Appendix F: Parameters used to optimize Hadoop Jobs

Parameter	Description
mapred.map.tasks	The total number of Map tasks for the job to run
mapred.reduce.tasks	The total number of Reduce tasks for the job to run
mapred.output.compress	Set true in order to compress the output of the MapReduce job, yse *mapred.output.compression.codec* to specify the compression codec.
mapred.map.child.java.opts	The Java options for JVM running a Map task. Use "*-Xmx*" to set max memory size. Use *mapred.reduce.child.java.opts* for Reduce tasks.
io.sort.mb	A Map task output buffer size. Use this value to control the spill process. When buffer is filled up to a *io.sort.spill.percent* a background spill thread is started.
mapred.job.reduce.input.buffer.percent	The percentage of memory relative to the maximum heap size to retain Map outputs during the Reduce process
mapred.inmem.merge.threshold	The threshold number of Map outputs for starting the process of merging the outputs and spilling to disk. 0 means there is no threshold, and the spill behavior is controlled by *mapred.job.shuffle.merge.percent*.
mapred.job.shuffle.merge.percent	The threshold usage proportion for the Map outputs buffer for starting the process of merging the outputs and spilling to disk
mapred.reduce.slowstart.completed.maps	The time to start Reducers in percentage to complete Map tasks
dfs.replication	HDFS replication factor
dfs.block.size	HDFS block size
mapred.task.timeout	The timeout (in milliseconds) after which a task is considered as failed. See *mapreduce.reduce.shuffle.connect.timeout*, *mapreduce.reduce.shuffle.read.timeout* and *mapred.healthChecker.script.timeout* in order to adjust timeouts
mapred.map.tasks.speculative.execution	Speculative execution of Map tasks. See also *mapred.reduce.tasks.speculative.execution.*
mapred.job.reuse.jvm.num.tasks	The maximum number of tasks to run for a given job for each JVM
io.sort.record.percent	The proportion of *io.sort.mb* reserved for storing record boundaries

	of the Map outputs. The remaining space is used for the Map output records themselves.
Example	$HADOOP_EXECUTABLE jar $HADOOP_EXAMPLES_JAR terasort \ -Dmapred.map.tasks=60 \ -Dmapred.reduce.tasks=30 \ -Dmapred.output.compress=true \ -Dmapred.map.child.java.opts="-Xmx3500m" \ -Dmapred.reduce.child.java.opts="-Xmx7000m" \ -Dio.sort.mb=2047 \ -Dmapred.job.reduce.input.buffer.percent=0.9 \ -Dmapred.inmem.merge.threshold=0 \ -Dmapred.job.shuffle.merge.percent=1 \ -Dmapred.reduce.slowstart.completed.maps=0.8 \ -Ddfs.replication=1 \ -Ddfs.block.size=536870912 \ -Dmapred.task.timeout=120000 \ -Dmapreduce.reduce.shuffle.connect.timeout=60000 \ -Dmapreduce.reduce.shuffle.read.timeout=30000 \ -Dmapred.healthChecker.script.timeout=60000 \ -Dmapred.map.tasks.speculative.execution=false \ -Dmapred.reduce.tasks.speculative.execution=false \ -Dmapred.job.reuse.jvm.num.tasks=-1 \ -Dio.sort.record.percent=0.138 \ -Dio.sort.spill.percent=1.0 \ $INPUT_HDFS $OUTPUT_HDFS

Table 7. Parameters that were tuned to achieve optimal performance of Hadoop jobs

About the author:

Vladimir Starostenkov is a Senior R&D Engineer at Altoros, a company that focuses on accelerating big data projects and platform-as-a-service enablement. He has more than five years of experience in implementing complex software architectures, including data-intensive systems and Hadoop-driven applications. Having strong background in physics and computer science, Vladimir is interested in artificial intelligence and machine learning algorithms.

About Altoros:

Altoros is a big data and Platform-as-a-Service specialist that provides system integration for IaaS/cloud providers, software companies, and information-driven enterprises. The company builds solutions on the intersection of Hadoop, NoSQL, Cloud Foundry PaaS, and multi-cloud deployment automation. For more, please visit www.altoros.com or follow @altoros.

Liked this white paper?
Share it on the Web!

+1 650 395-7002
engineering@altoros.com
www.altoros.com | twitter.com/altoros