

# Deploying an Apache Hadoop\* Cluster? Spend Your Time on BI, Not DIY

Intel® Xeon® processor family

Oracle Big Data Appliance\*

Achieve near-real-time results from your Apache Hadoop\* cluster, without days or weeks of specialized tuning



## Executive Summary

Do-it-yourself (DIY) Apache Hadoop\* clusters are appealing to many organizations because of the apparent cost savings from using commodity hardware and free software distributions. Despite these apparent savings, many organizations are opting to purchase pre-built clusters, such as the Oracle Big Data Appliance\*, because they know that a commodity cluster requires considerable time and specialized engineering skills to deploy, optimize, and tune for real-time data analysis.

Powered by fast, efficient Intel® Xeon® processors, the Oracle Big Data Appliance is a pre-built and optimized solution for Apache Hadoop big data analysis that can be deployed in minutes. It can then be tuned for near-real-time analysis in minutes or hours, instead of the days or weeks it would take to tune a DIY Apache Hadoop cluster. This paper describes the performance tuning techniques used in benchmark testing that resulted in an Oracle Big Data Appliance performing nearly two times faster than a comparable DIY cluster.

## Oracle Big Data Appliance\* includes:

- Oracle Sun x86\* servers powered by the Intel® Xeon® processor family
- InfiniBand\* and Ethernet connectivity
- Cloudera Enterprise, including Impala\*
- Oracle NoSQL Database\*
- Comprehensive security, including authentication, authorization, and auditing capabilities
- Oracle Linux\*, Oracle Java Hotspot VM\*, and the Oracle R Distribution\*

## The Do-It-Yourself Myth

Businesses rely on up-to-date analysis of customer habits, sales data, and other factors for improving products and services, correcting manufacturing problems, and fine-tuning operations to maximize productivity. Faced with massive—and growing—amounts of data to quickly analyze, many organizations are turning to fast-performing big-data solutions. In fact, according to an IDG Enterprise survey on big data, 80 percent of enterprises already have deployed or are planning to deploy big data projects in the next 12 months.<sup>1</sup>

For companies just getting started down the path of big data analytics, solutions based on Apache Hadoop are appealing

because Apache Hadoop can run on commodity infrastructure, it is based on open source technology, and the software is available for free. But setting up an Apache Hadoop cluster using commodity hardware can be much more challenging than many organizations expect. A DIY solution requires that you carefully research and purchase servers with processors that can support rapid analytics and reporting needs. You also need to consider scalability and compatibility with existing hardware and software. Your network components need to keep up with query demands or else they can drag down the entire system.

**Table of Contents**

- Executive Summary ..... 1
- The Do-It-Yourself Myth ..... 1
- Faster Time to Value with Oracle Big Data Appliance..... 2
- Benchmark Testing: Oracle Big Data Appliance Versus a DIY Apache Hadoop Cluster..... 2
  - Test Configuration..... 2
- Hardware and Software Tuning Nearly Doubles Performance..... 3
  - Encryption..... 3
  - File-System Configuration ..... 3
  - Data Compression..... 4
  - Java Virtual Machine ..... 4
    - Java Garbage Collection ..... 4
  - General Apache Hadoop Tuning 5
    - YARN Container Size..... 5
    - Parallel Execution..... 5
    - Vectorization..... 5
    - Uber Task..... 5
  - Tuning Large Pages in Linux\*... 5
  - General Tuning Results ..... 5
- Workload-Specific Tuning Takes Performance Even Higher ..... 5
  - Workload-Specific Tuning Offers Big Performance Gains..... 5
- Low Effort, Big Gains..... 5
- Intel and Oracle: Engineered Together..... 6

Even if you purchase and deploy the ideal hardware and software for your environment, it can take days to implement effective security controls and optimize the Apache Hadoop cluster—and that’s all before you run even your first query. But successfully running a query is a far cry from achieving real-time analytics. To fully realize the potential of Apache Hadoop could require the costly services of a specialized Apache Hadoop engineer working for weeks or months to fine-tune the hundreds of variables within the software.

**Faster Time to Value with Oracle Big Data Appliance**

The Oracle Big Data Appliance offers an attractive alternative to purchasing, deploying, configuring, and fine-tuning a DIY Apache Hadoop cluster. The pre-built appliance is an integrated, optimized solution powered by the Intel Xeon processor family. With the pre-configured Oracle Big Data Appliance, you can generate your first query in minutes. Moreover, after only a few hours of fine-tuning your environment, you can experience near-real-time results for large queries. Recent benchmark testing

Table 1. Configuration of the Oracle Big Data Appliance\* and the do-it-yourself hardware cluster

	ORACLE BIG DATA APPLIANCE*	DO-IT-YOURSELF CLUSTER
Number of nodes	6	6
<b>CONFIGURATION PER NODE</b>		
CPU	Intel® Xeon® processor E5-2699 v3 (2.30 GHz) with Intel® Hyper-Threading Technology enabled	Intel Xeon processor E5-2699 v3 (2.30 GHz) with Intel Hyper-Threading Technology enabled
Memory	DDR4 128 GB RAM	DDR4 128 GB RAM
Storage	Twelve 4 TB hard-disk drives (HDDs)	One 64-GB solid-state drive for the OS installation Twelve 4 TB HDDs
Network	InfiniBand* network (1 connection) Observed maximum throughput: 24 Gb/sec	10 Gb network (1 connection)
OS Version	Oracle Linux Enterprise 6*	CentOS 6.6*
Cloudera CDH* version	5.4.4 with modified configuration	5.3.3 with minimal changes (to better utilize the hardware resource)

by Intel engineers demonstrated that an Oracle Big Data Appliance solution with some basic tuning achieved nearly two times better performance than a comparable DIY cluster built on comparable hardware.

**Benchmark Testing: Oracle Big Data Appliance versus a DIY Apache Hadoop Cluster**

Intel engineers performed benchmark tests to determine how much faster an Oracle Big Data Appliance performs over a comparable DIY Apache Hadoop cluster. This paper describes the results of those tests and outlines the configuration and tuning areas that Intel engineers looked at to enhance security and to quickly achieve near-real-time performance. This paper is not intended to be a step-by-step tuning guide, but it demonstrates what is possible with even a minor investment in time and effort.

**Test Configuration**

Intel engineers tested two Apache Hadoop six-node clusters: one on the Oracle Big Data Appliance, the other on DIY hardware. Configuration details are

provided in Table 1. Tests were performed using the BigBench\* big data benchmarking suite with a 2 TB input dataset.

To compare the performance of the two clusters, Intel engineers used the following methodology:

1. Established BigBench baseline numbers for both clusters.
2. Applied Intel best-known methods (BKMs) for performance tuning to the Oracle Big Data Appliance cluster and measured BigBench performance gains. (This was done as an iterative process following each tuning adjustment.)
3. Explored additional tuning options and measured performance gains through targeted micro-benchmarks.



Out of the box, the Oracle solution performed more than 1.2 times faster than the DIY cluster. After some general Apache Hadoop performance tuning, the Oracle Big Data Appliance performed nearly 2 times faster than the DIY cluster. The following sections describe the tuning performed by Intel engineers to achieve these results.

**Hardware and Software Tuning Nearly Doubles Performance**

There are numerous ways to tune the Oracle Big Data Appliance for your needs and environment. The goal for Intel engineers in these test scenarios was to identify simple optimizations that could be made quickly, without specialized engineering expertise, to achieve near-real-time performance

from the Apache Hadoop cluster. Engineers focused their attention on seven areas:

- Encryption
- File system
- Data compression
- Java Virtual Machine\*
- Apache Hadoop
- Large pages
- Workload-specific tuning

Some of these functionalities, like encryption and compression, are already integrated into the Oracle appliance today. Many of the other configuration changes will be incorporated into a future software update.

“Oracle’s Big Data Appliance is a big win for customers. This easy-to-deploy system is built on powerful Intel® processors and bundles the full suite of Cloudera Enterprise software, ensuring easy, secure, and managed real-time analytics right out of the box.”

—Charles Zedlewski, VP Products, Cloudera

**Encryption**

Your big data workloads will include sensitive data about your customers and business. With major security breaches in the news almost every week, you cannot afford to leave data unprotected. Yet that is exactly what many businesses do when they avoid using encryption because of the performance penalty that it typically exacts.

Fortunately, you no longer need to choose between performance and data protection. Beginning with Intel Xeon processor E3 v3 family, Intel provides built-in Intel® Data Protection Technology with Advanced Encryption Standard New Instructions (AES-NI). This technology, based on the industry-standard AES encryption protocol, reduces performance latency for encryption and decryption operations. Thanks to the collaboration between Intel and Cloudera, you can enable Hadoop Distributed File System\* (HDFS\*) encryption for an entire Cloudera Enterprise cluster with no significant performance penalty.<sup>2</sup> Intel performance tests have shown that AES-NI can accelerate encryption performance in an Apache Hadoop cluster by up to 5.3 times and decryption performance by up to 19.8 times.<sup>3</sup>

To verify that performance does not need to sacrifice strong security, Intel engineers enabled full-disk AES-NI encryption for the drives and workloads used in the benchmark test scenarios.

**File System Configuration**

XFS\* is a popular file system for organizations working with very large data sets. That is because in addition to being adept at handling a wide range of workloads, recent improvements in journaling allow XFS to scale much more effectively than other systems, such as Ext4\*.

In order to measure the performance benefits from XFS, Intel engineers installed XFS to replace the Ext4 file system running under HDFS on the Oracle cluster. As seen in Figure 1, the Oracle solution demonstrated measureable performance gains for XFS over Ext4 as the read/write thread count increased.

## Performance gains for XFS over Ext4 file system



Figure 1. Benchmark test performance gains achieved after replacing the Ext4\* file system with the XFS\* file system

### Data Compression

Compression is useful because it reduces data storage needs, reduces network traffic, and can even accelerate performance by significantly reducing input/output (I/O) for reads/writes. Compression also allows more data to be held in main memory, where it can be accessed much faster than from disk storage.

For compression to boost performance, however, it must use efficient algorithms that ensure the costs of compressing data do not outweigh the benefits. The default compression algorithm for the Apache Hive\* table is ZLIB—an effective but slower algorithm compared to newer compression options. Changing compression to the faster-performing Snappy\* algorithm increased BigBench performance by approximately 5 percent.

### Java Virtual Machine

Intel and Oracle Java virtual machine (JVM\*) engineers are continuously working together to optimize JVM performance. These collaborative efforts have resulted in a 32-times performance improvement since 2007 for Java applications running on servers powered by Intel Xeon processors.<sup>5</sup>

The Oracle Big Data Appliance takes advantage of those Intel and Oracle Java optimizations to help increase overall performance for Apache Hadoop and contribute to the goal of realizing near-real-time performance in the test scenarios.

### Java Garbage Collection

Java offers several options for garbage collection, but the parallel collector is the default garbage collector for Java. It uses multiple threads to scan through and compact the heap. The parallel collector stops application threads when performing a minor or full collection, which makes it a good choice for applications that can tolerate application pauses. Unfortunately, those pauses can add latency that might impact the performance of your big data cluster.

The Garbage-First\* (G1) collector is a better choice for supporting heaps larger than 4 GB. The G1 collector divides the heap into regions and uses multiple background threads to scan those regions, giving preference to those containing the most garbage objects.

To improve performance in benchmark testing, Intel engineers:

- Upgraded the Java Development Kit\* (JDK\*) to version 1.8 update 60

- Moved long-lived services, such as DataNode and NameNode, to the G1 collector
- Tuned the parallel collector for mappers and reducers

Collectively, these changes helped reduce BigBench run time and contribute to overall performance gains for the Oracle Big Data Appliance.

### General Apache Hadoop Tuning

General tuning was performed in the following areas to enhance Apache Hadoop performance on the Oracle Big Data Appliance.

“Oracle’s collaboration with Intel has enabled us to provide a simple, highly-optimized solution for customers looking for faster time-to-value from their big data analytics solution.”

—Neil Mendelson, Vice President, Big Data and Advanced Analytics, Oracle

### YARN Container Size

With the default YARN container size set to 1 GB and both master and worker services running on the same server, engineers observed a larger number of concurrent mappers/reducers competing for resources, an extremely high load average, and occasional loss of data nodes.

Increasing the container size to 2 GB contributed to greater performance while also achieving a lower load average and a smaller number of concurrent mappers/reducers. Increasing the container size also eliminated the loss of data nodes.

### Parallel Execution

Parallel execution enables Apache Hadoop to execute MapReduce jobs concurrently.

Prior to enabling parallel execution, engineers observed that the MapReduce cluster exhibited heavy-tailed characteristics for job processing times with reduced cluster-resource utilization. By configuring the test Apache Hadoop cluster to execute MapReduce jobs in parallel, engineers observed the following benefits:

- An overall performance increase in BigBench
- Concurrent jobs (phases) filled the resource utilization gap created by tails of MapReduce
- Bigger queries showed even greater improvement

### Vectorization

Vectorization allows Apache Hadoop to process a batch of rows together instead of processing one row at a time. Each batch consists of a column vector. Batch processing the columnar data can improve the instruction pipelines and cache usage. Enabling vectorization also contributed to overall performance gains for the test Apache Hadoop cluster on the Oracle Big Data Appliance.

### Uber Task

MapReduce is ideal for large queries that take minutes or hours to complete, but it is not designed for smaller, interactive queries. A simple MapReduce job that processes a small amount of data might still take tens of seconds, or even minutes, to run.

Uber tasks can improve performance for smaller queries by using a separate JVM to run MapReduce tasks, so that the tasks are executed sequentially on one node.

Using Uber tasks improved overall performance for the Oracle test cluster by increasing the efficiency of smaller queries. Other benefits from using Uber tasks included a reduced JVM overhead and startup cost and conservation of resources for other jobs in a high-load environment.

### Tuning Large Pages in Linux\*

Smaller pages in Linux increase the size of page tables, which negatively impacts performance for Apache Hadoop queries. By increasing the number of large pages from 64 to 16,000, engineers were able to improve the BigBench performance for 25 out of 30 queries by an average of approximately 3 percent. The remaining five queries were negatively impacted by less than 1 percent. Overall, testing showed a 2 percent improvement after tuning the pages. For ideal performance on your system, fine-tune page sizes for each application.

 **Nearly 2x**  
faster performance  
after general tuning

### General Tuning Results

After implementing the general tuning changes described above, the Oracle Big Data Appliance performed nearly two times faster than the DIY cluster.

### Workload-Specific Tuning Takes Performance Even Higher

Engineers then performed workload-specific tuning in order to boost performance even further by:

- Fully utilizing the cluster’s resources, CPU cores, and memory
- Reducing JVM overhead
- Reducing idle resources

 **Nearly 3x**  
faster performance  
after workload-specific tuning

The approach for tuning these workload-specific areas was to focus on five representative queries and tune them in iterations. After each iteration, engineers fine-tuned the changes and reran the queries until peak performance was attained. Then, the optimized tuning was applied to all additional queries in the test scenario. Only Apache Hadoop parameters were adjusted, not the queries themselves.

### Workload-Specific Tuning Offers Big Performance Gains

After a few hours of iterative tuning and testing with BigBench, the Oracle Big Data Appliance was performing 2.86 times faster than the DIY cluster. Although every environment is different, these performance gains are representative of what you might achieve with some basic tuning to your workloads on an already highly optimized and efficient Oracle Big Data Appliance, powered by Intel Xeon processors.

### Low Effort, Big Gains

DIY Apache Hadoop solutions are initially attractive because of apparent savings in up-front hardware and software costs. Despite the initial savings in capital, DIY clusters are not always the best option for organizations looking to achieve faster time-to-value for their big data solutions. By definition, DIY solutions use components that are not integrated with each other. That means that they might require days or weeks of highly specialized engineering time to optimize the disparate hardware and software components.

In contrast, the Oracle Big Data Appliance is a turnkey solution that you can deploy and use in minutes. Powered by Intel Xeon processors, the Oracle appliance makes full use of integrated Intel technologies for fast, efficient processing, data compression, and data encryption.

As shown in Intel's benchmark testing, the Oracle Big Data Appliance demonstrated nearly three times faster performance over the DIY cluster after simple Apache Hadoop, system, and workload tuning that did not require advanced engineering expertise. These tests demonstrate that the Oracle Big Data Appliance is an attractive solution for organizations looking to achieve near-real-time analytics for business intelligence.



### Intel and Oracle: Engineered Together

Thanks to a long-standing and continuing partnership between Intel and Oracle, you can expect to see continued innovations and performance enhancements over time. Some of the tuning enhancements in this paper will be incorporated into future updates to Oracle Big Data Appliance software. Intel will share new technologies and techniques as they become available.

For more information about Oracle Big Data Appliance, see [oracle.com/engineered-systems/big-data-appliance/index.html](http://oracle.com/engineered-systems/big-data-appliance/index.html).

For more information on Intel Xeon processor family and big data, see [intel.com/content/www/us/en/big-data/big-data-analytics-turning-big-data-into-intelligence.html](http://intel.com/content/www/us/en/big-data/big-data-analytics-turning-big-data-into-intelligence.html).



<sup>1</sup> IDG Enterprise. "Big Data and Analytics: The Big Picture." March 2015. <http://www.idgenterprise.com/report/big-data-and-analytics-the-big-picture>.

<sup>2</sup> Intel white paper. "Intel® Xeon® Processor E5-2600 v3 Accelerates Hadoop HDFS Encryption." 2015. [http://www.intel.com/newsroom/kits/xeon/e7v3/pdfs/xeon\\_e7v3\\_cloudera-aes-ni.pdf](http://www.intel.com/newsroom/kits/xeon/e7v3/pdfs/xeon_e7v3_cloudera-aes-ni.pdf).

<sup>3</sup> Intel solution brief. "Fast, Low-Overhead Encryption for Apache Hadoop\*." February 2013. <http://www.intel.com/content/dam/www/public/us/en/documents/articles/intel-distribution-for-apache-hadoop-encryption-solution-brief.pdf>.

<sup>4</sup> Corbet, Jonathan. LWN.net. "XFS: The filesystem of the future?" January 2012. <https://lwn.net/Articles/476263/>.

<sup>5</sup> Intel white paper. "Accelerating Performance for Server-Side Java\* Applications." January 2014. <https://software.intel.com/sites/default/files/accelerating-performance-for-server-side-java-applications.pdf>.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase.

For more complete information about performance and benchmark results, visit <http://www.intel.com/performance>.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com](http://intel.com).

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

Copyright © 2015 Intel Corporation. All rights reserved.

\* Other names and brands may be claimed as the property of others.