White Paper
Intel® Xeon® Processor-Based Servers
Big Data Analytics

Optimizing Hadoop* Deployments

Designing the solution stack to maximize productivity while limiting energy consumption and total cost of ownership

Executive Summary

This paper provides guidance, based on extensive lab testing conducted at Intel, to help IT organizations plan an optimized infrastructure for deploying Apache Hadoop*. It includes:

- **Best practices for establishing server hardware specifications**
- **High-level software guidance regarding the operating system (OS), Java Virtual Machine (JVM), and Hadoop version**
- **Configuration and tuning recommendations to provide optimized performance with reduced effort**

Overview

Apache Hadoop is an open source, distributed software platform for storing and processing data. Because of its affordability, massive scalability and ability to handle all data types, it has become the defacto standard for storing and analyzing the large volumes of unstructured and semi-structured data that businesses collect from web logs, sensors, e-mail, call centers, and many other sources. You can deploy Hadoop on a small cluster of industry-standard servers configured with direct-attached storage. As workloads and data volumes grow, you can scale performance and capacity by adding thousands of additional server nodes.

Because infrastructure requirements can be very large, hardware and software choices made at design time can have a significant impact on performance and total cost of ownership (TCO). Intel has devoted considerable resources to Hadoop analysis, testing, and performance characterizations, working independently, with the open source community, with leading vendors and with business customers to develop and implement optimized solutions.

Through these technical efforts, we have observed many practical trade-offs in hardware, software, and system settings that have real-world impacts. This white paper discusses those trade-offs and provides infrastructure recommendations that can be grouped into three categories.

- **Server hardware.** Choosing the right hardware components provides an optimal balance of performance, data capacity, initial costs, and recurring costs.
- **System software.** In addition to the choice of the OS and JVM, the specific version of Hadoop and other software components have implications for performance, stability, and functionality.
- **Configuration and tuning.** The settings made to the Hadoop environment itself are an important factor in extracting the best performance from the hardware and software solution stack.

Table of Contents

It is important to note that Hadoop deployments will vary considerably from customer to customer and from project to project, depending on actual workloads. Because of this, it is not always possible to provide detailed component recommendations. Where that is the case, we provide workload-focused guidelines to help architects make more informed decisions.

General Hadoop Cluster Technology

A typical Hadoop cluster consists of a two- or three-level architecture made up of rack-mounted servers interconnected in a tiered cluster topology. Every cluster includes one or more nodes for managing cluster-wide data placement and job allocation, and multiple slave nodes for storing the data and executing the analytic tasks.

Data placement and job allocation are performed by the following Java services:

- **JobTracker** breaks requests into multiple smaller tasks and assigns the tasks to the slave nodes that contain the required data. Tasks are executed as close to the data as possible to avoid the delays and overhead associated with moving data within the cluster.
- **NameNode** keeps track of data location and other metadata for the Hadoop Distributed File System (HDFS). Although NameNode can run on the same server as JobTracker, it is generally preferable to host this service on a separate physical system. NameNode can also be scaled horizontally on two or more physical servers using a relatively new feature, called HDFS Federation. The federated NameNodes and namespaces are independent of each other, which simplifies implementation and enables workload isolation for multiple user groups accessing the same cluster.
- **Secondary NameNode,** despite its name, is not a failover server for the NameNode. Instead, it periodically check-points the file system metadata that is generated in the NameNode. By default, Hadoop configures Secondary NameNode to run on the same server as NameNode, but this service can also be hosted on a separate physical system. (Using Secondary NameNode to recover a failed NameNode server requires restarting all the nodes in the cluster, so this strategy should only be used if the image and logs in the NameNode are inaccessible.)

Each slave node hosts the following three Java services:

- **DataNode** is a component of HDFS that stores and retrieves the data on the slave node at the request of NameNode.
- **TaskTracker** receives MapReduce tasks from JobTracker and runs them on the slave node.
- **MapReduce** is the analytic service that performs the assigned tasks.

Server Hardware Configurations

When planning a Hadoop cluster, the number, type, and configuration of the servers are among the most important decisions you will make. Typically, dual-socket servers based on the Intel® Xeon® processor E5 family are optimal for Hadoop deployments. These systems can handle a wide range of Hadoop workloads, and the performance they provide more than offsets the added per-node hardware cost relative to entry-level, single- socket systems.

In most scenarios, dual-socket servers will also be more efficient, from a cost-benefit perspective, than large-scale multi-processor platforms. In some cases, however, it may be advantageous to employ more scalable server nodes based on the Intel® Xeon® processor E7 family. These processors provide additional cores, larger cache, and greater system I/O bandwidth than the Intel Xeon processor E5 family. They also support larger memory configurations—up to 1.5 TB per processor. The higher memory capacity can be a significant performance advantage, since it helps to eliminate much of the latency and processing overhead associated with moving data back and forth between disk storage and server memory.

Intel internal tests have shown that replacing two-socket server nodes based on the Intel Xeon processor E5 family with 4-socket server nodes based on the Intel Xeon processor E7 v2 processor family can increase performance for I/O-intensive workloads by up to 2.7 times and for CPU-intensive workloads by up to 3.5 times.[1,5] Based on these results, comparable performance levels can be achieved using approximately one third as many servers. In addition, server nodes based on the Intel Xeon processor E7 v2 processor family include advanced on-chip reliability, serviceability, and availability (RAS) features for mission-critical applications.

The overall cost impact of choosing systems based on either Intel Xeon processor family will depend on your application virtualization strategy as well as many physical and operational factors within the data center, including the cost and availability of power, cooling, and rack space.

- For more information about deploying Hadoop on four-socket or larger servers, and for details about the performance tests, look for the Intel white paper: "Accelerating Big data Analysis with Intel® Technologies" on the Intel® Developer Zone.

Regardless of the socket-count, choose servers that use high-efficiency voltage regulators and are optimized for airflow to contain power and cooling costs as the cluster grows. Many Intel® Xeon® processor-based servers are designed specifically for cloud service providers and Internet data centers. These systems have been optimized for the cost, density, weight, and power consumption characteristics required in high-density computing environments, and tend to offer high value for Hadoop deployments.

There is no need to configure your slave nodes with redundant components for high availability. Hadoop automatically creates multiple replicas of all data (three copies by default) and stores the replicas on different physical servers. Any slave node in a Hadoop cluster can fail without data loss and without bringing down the cluster or interrupting service requests.

JobTracker and NameNode servers, on the other hand, can be centralized performance bottlenecks and single-points-of-failure and should be configured accordingly. There are several ways to provide higher scalability and availability for these critical server functions, including the following.

- **Option 1:** Configure JobTracker and NameNode servers with additional RAM and secondary power supplies.
- **Option 2:** Use two-socket or four-socket servers based on the Intel Xeon processor E7 v2 processor family. These systems are more scalable and provide exceptionally robust RAS features, such as advanced error correction, containment and recovery.
- **Option 3:** Use Federated HDFS to scale out NameNode functionality on multiple, independent servers.

Hadoop workloads vary widely, so it is important to configure the compute, storage, network and software resources in your Hadoop servers to match your specific requirements. There is a common misconception that all Hadoop workloads are I/O-bound. Intel has run extensive tests using a reference set of test workloads that, in aggregate, resemble real-world Hadoop applications. Based on in-depth instrumentation of the test clusters, we found that I/O and CPU utilization vary significantly, not only across workloads, but also across the different stages of processing in a typical MapReduce application. For more information about Hadoop workload characteristics, see the Intel white paper: Extract, Transform, and Load Big Data with Apache Hadoop*.

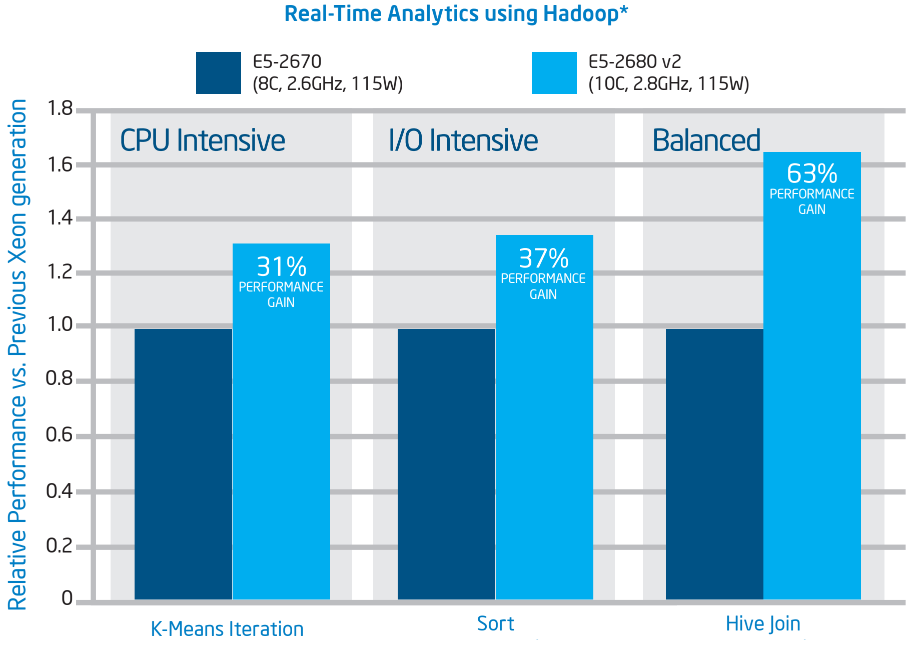


Figure 1. According to Intel internal testing, configuring Hadoop servers with the latest Intel® Xeon® processor E5 v2 product family rather than previous generation processors can provide significant performance gains for MapReduce applications that reflect a broad range of real-world scenarios, including CPU-intensive, I/O-intensive, and balanced workloads.

Choosing Processors

Server processors play an important role in determining the speed, throughput, and efficiency of a Hadoop cluster. As already mentioned, the Intel Xeon processor E5 product family provides excellent performance for highly distributed workloads, such as those associated with Hadoop applications. Unless you have strict budget limitations, it is generally preferable to use the most recent processor generation. Intel ran a series of tests to quantify the benefits of using the latest Intel Xeon processor E5 v2 family versus prior-generation processors. Performance was compared across the following three Hadoop benchmarks, all of which are representative of important, real-world workloads.

- **K-means Iteration** is a CPU-intensive workload that includes a well-known machine learning algorithm.
- **Sort** is an I/O-intensive workload that transforms data from one representation to another. It represents a large subset of MapReduce jobs and is used pervasively as a performance indicator.
- **Hive Join** is both CPU- and I/O-intensive and provides a good measure of performance for complex, OLAP-style analytics.

The results are shown in Figure 1. The performance gains were substantial: up to 31 percent for the CPU-intensive benchmark, up to 37 percent for the I/O-intensive benchmark, and up to 63 percent for the balanced (CPU-intensive and I/O-intensive) workload.[2,5]

Selecting and Configuring Storage Drives

Selecting storage drives for Hadoop servers requires balancing the demands of performance and capacity with the associated costs. In general, solid state drives (SSDs) are preferable if performance is a primary consideration, while traditional hard disk drives (HDDs) may be a better choice if cost-effective data capacity is more important in your implementation.

Intel® Solid-State Drives (Intel® SSDs) deliver roughly four times higher bandwidth than HDDs for both sequential read and sequential write operations. This extra bandwidth can eliminate or reduce potential storage bottlenecks, so applications can make better use of available compute resources when loading data into the cluster and when accessing data for queries and other processing tasks. Intel tests have shown Intel SSDs to deliver up to 2.7x faster job throughput[3,5] for I/O-intensive workloads, such as Sort and TeraSort (Figure 2).

Because they deliver better performance with fewer drives, SSDs, in some scenarios, will be more cost-effective than HDDs. You can reduce the number of drives per node by a factor of three or four without impairing performance, and you may be able to boost performance considerably higher with additional SSDs. On the other hand, if data capacity and budget are driving factors in your implementation, or if your target workloads are not I/O intensive, using HDDs is more cost-effective.

If your workloads and requirements are mixed, you can configure Hadoop servers with a combination of SSDs and HDDs. Consider using Intel® Cache Acceleration Software (Intel® CAS) to automatically move frequently used data onto the higher-performing SSDs. This approach delivers some of the performance advantages of SSDs, while still allowing you to scale data capacity per-node using more cost-effective HDDs. Intel CAS is transparent to applications, so no coding changes are required.

Memory Sizing

In general, efficient hardware utilization occurs when the number of simultaneous MapReduce tasks matches the number of hardware threads. For example, the Intel Xeon processor E5 v2 processor family provides up to 12 cores and 24 threads per processor, so a two-socket server would provide 48 hardware threads and would ideally support approximately 48 MapReduce tasks.

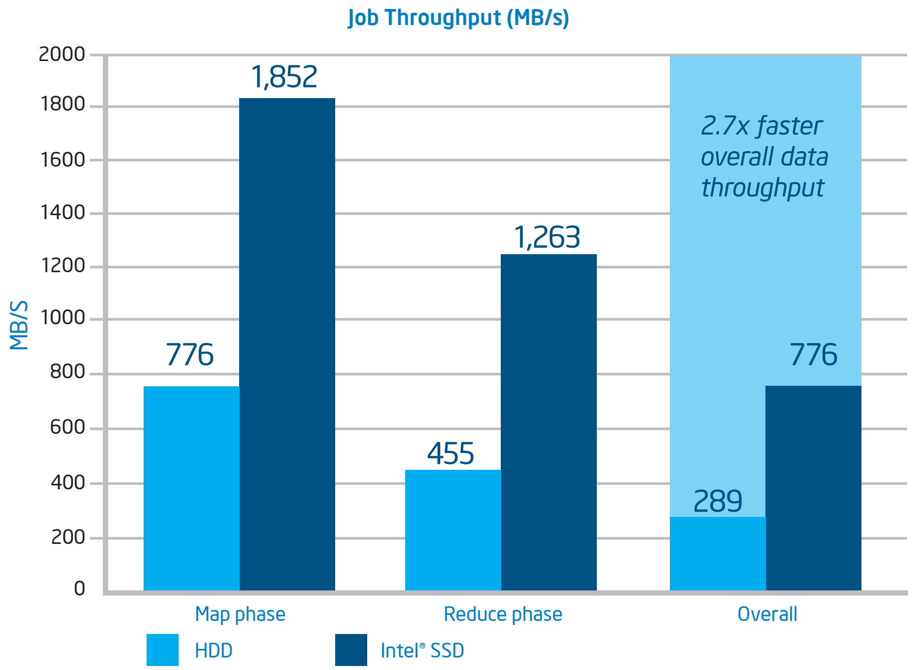


Figure 2. Using Intel® Solid State Drives (Intel® SSDs) in place of traditional hard disk drives (HDDs) can improve performance for I/O-intensive Hadoop workloads, such as Sort, by up to 2.7 times. If workloads are not I/O-intensive or if data capacity is more important than performance, HDDs may be a more cost-effective choice.

For high throughput, it is important that each MapReduce task have sufficient memory capacity. As a general rule, each task requires roughly one gigabyte (GB) of memory for good performance, including 512 MB for the Java Virtual Machine (JVM) and 512 MB for an estimated three reducers. If you have a server with 48 hardware threads, as described above, a good starting point is to configure the system with 48 GB of memory. When you tune your cluster using your own workloads, you can increase memory capacity if needed to avoid bottlenecks.

Networking

Hadoop performance and efficiency is highly dependent on network architecture and technology choices. A complete network strategy is beyond the scope of this paper, so the recommendations here are focused primarily on the server network interface.

While Gigabit Ethernet (GbE) has been the most commonly deployed networking fabric for Hadoop over the past few years, it provides less than ideal bandwidth for many workloads today. This is particularly true when the Hadoop cluster is based on newer generation servers configured with the latest Intel Xeon processors and Intel SSDs. As processing and storage performance increases, the throughput and latencies of a gigabit network can quickly become a constraint to overall performance.

Although it is possible to bond multiple Gigabit Ethernet server adapters to avoid bottlenecks, 10 Gigabit (10 GbE) Intel Ethernet Server Adapters generally provide a more cost-effective and higher performing solution. Total cost-per-terabyte is lower, setup is simpler, cabling requirements are reduced, and power consumption is lower. Perhaps most importantly for Hadoop environments, a 10 GbE network will make it easier to scale the cluster as workloads grow.

A 10 GbE network can eliminate bottlenecks and increase job throughput for many I/O-intensive Hadoop applications.

Time to Import Data into Hadoop* and Replicate it to Data Nodes for Processing

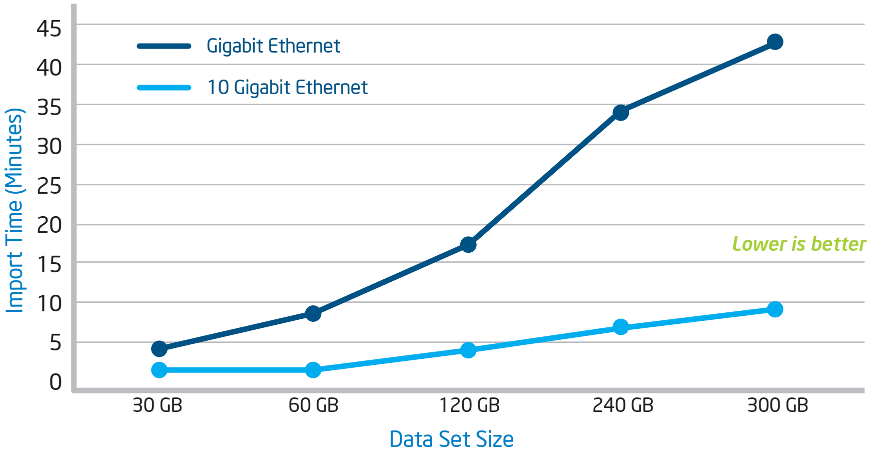


Figure 3. A cluster network based on 10 Gigabit Ethernet decreases data loading times by roughly 80 percent versus Gigabit Ethernet, and can help to eliminate potential performance bottlenecks across a wide range of I/O-intensive Hadoop workloads.

The most striking impact is typically seen when loading data into the cluster. During this process, data is loaded onto the NameNode, replicated three times for performance and high availability, and then distributed across the various slave nodes, a process known as a PUT operation.

Intel ran tests to compare the performance for PUT operations using GbE versus 10 GbE networking. Results showed that 10 GbE reduced PUT completion times by a factor of five,[4,5] (Figure 3). Since the time required to load data scales linearly with the size of the data set, the 80 percent performance benefit is increasingly important as data sets grow. For multi-terabyte data sets, using a 10 GbE network rather than a GbE network can potentially reduce wait times by several hours.

System Software Setup and Configuration

Selecting the Operating System and JVM

Using a Linux* distribution based on kernel versions 2.6.30 or later is recommended when deploying Hadoop on current generation servers because of the optimizations included for energy and threading efficiency. For example, Intel has observed that energy consumption can be up to 60 percent (42 watts) higher at idle for each server using older versions of Linux. Such power inefficiency, multiplied over a large Hadoop cluster, could amount to significant additional energy costs.

Linux has a number of tunable parameters that can impact Hadoop performance, and different kernel versions may have different default settings. For example, the default Linux open file descriptor limit in Linux kernel 2.6.30 is set to 1,024, which is usually too low for Hadoop daemons. This setting should be increased to approximately 64,000 using the /etc/security/limits.conf file or alternate means. If the Linux kernel 2.6.28 is used, the default open epoll file descriptor limit is 128, which is too low for Hadoop and should be increased to approximately 4,096 using the /etc/syscl.conf file or alternate means.

Sun Java* 6 is required to run Hadoop, and Oracle Java 6u14 or later is recommended to take advantage of optimizations such as compressed ordinary object pointers. Later versions may provide additional optimizations. Intel benchmarking teams used Oracle Java 7u13 with good results in recent performance benchmarks.

Choosing Hadoop Versions and Distributions

Hadoop is available as a free, open-source distribution from the Apache Software Foundation. If you decide to use an open-source version, you will most likely want to seek a balance between the enhancements available for the most recent releases and the stability available from more mature versions.

A number of commercial, vendor-supported Hadoop distributions are also available. The lab testing reported in this paper was conducted using Intel® Distribution for Apache Hadoop* Software (see the sidebar, Intel Distribution for Apache Hadoop Software).

Intel® Distribution for Apache Hadoop* Software

A number of vendors offer commercial versions of Hadoop software that include various enhancements and optimizations to simplify implementation, improve performance and reduce risk.

The Intel Distribution for Apache Hadoop Software (Intel Distribution) is one option. It includes the full open-source software stack, along with a combination of open-source and proprietary components designed to provide enterprise-class security and manageability.

The Intel Distribution is pre-integrated, extensively tested, and highly optimized for performance on Intel Architecture. Key features include advanced access controls, hardware-accelerated data encryption, automated performance tuning and a dashboard for monitoring and controlling the cluster environment.

For more information, visit the Intel Distribution web site.

Hadoop Configurations and Tuning

To achieve optimal results from Hadoop implementations, Intel lab testing has identified some key considerations for configuring the Hadoop environment itself. As with the other hardware and software recommendations discussed in this paper, the benefit of these optimizations depends heavily on the unique characteristics of the individual application, so users are encouraged to experiment with their own systems and enforcement to achieve the best results.

System Configurations

The file system's noatime and nodiratime attributes disable recording access information associated with the file system. Adding them into mount options can eliminate writes to the file system and result in measurable performance gains. Using these mount options for Hadoop intermediate storage and HDFS storage improves file system I/O performance, especially for I/O-bound workloads.

The file system read-ahead buffer size improves performance in sequential reads of large files, by prefetching additional blocks in anticipation. Generally, the default buffer size is 256 sectors, which should be increased to 1,024 or 2,048.

General Configurations

- **The numbers of NameNode and JobTracker server threads** that handle remote procedure calls (RPCs), specified by dfs.namenode.handler.count and mapred.job.tracker.handler.count, respectively, both default to 10 and should be set to a larger number (for example, 64) for large clusters.
- **The number of DataNode server threads** that handle RPCs, as specified by dfs.datanode.handler.count, defaults to three and should be set to a larger number (for example, eight) if there are a large number of HDFS clients. (Note: Every additional thread consumes more memory.)
- **The number of work threads on the HTTP server** that runs on each TaskTracker to handle the output of map tasks on that server, as specified by tasktracker.http.threads, should be set in the range of 40 to 50 for large clusters.

HDFS-Specific Configurations

- **The replication factor** for each block of an HDFS file, as specified by dfs.replication, is typically set to three for fault tolerance; setting it to a smaller value is not recommended.
- **The default HDFS block size,** as specified by dfs.block.size, is 64 MB. It is usually desirable to use a larger block size (such as 128 MB or even 256 MB) for large file systems.

MapReduce-Specific Configurations

- **The maximum number of MapReduce tasks** that run simultaneously on a TaskTracker, as specified by mapred.tasktracker.{map/reduce}.tasks.maximum, should usually be set in the range of (cores_per-node)/2 to 2x(cores_per_node), especially for large clusters.
- **The number of input streams threads** (files) to be merged at once in the MapReduce tasks, as specified by io.sort.factor, should be set to a sufficiently large value (for example, 100) to minimize disk accesses.
- **The JVM settings** should have the parameter java.net.prefer/Pv4Stack set to true, to avoid timeouts in cases where the OS/JVM picks up an IPv6 address and must resolve the hostname. The Java VM heap size should also be set to avoid fragmented heaps and frequent garbage collection. As discussed earlier in this paper, a setting of 512 MB per hardware thread represents a good starting point.

Map Task-Specific Configurations

- To optimize performance in a Hadoop cluster, it is critical to make efficient use of available memory. If the memory is insufficient, data must spill over to disk. A single spill is expected. If there is more than one spill per MapReduce task, all the associated data must be re-read and re-written, which increases I/O traffic by a factor of three. There are three relevant parameters: io.sort.mb is the total buffer space, io.sort.record.percent is the proportion between metadata buffers and key/value data, and io.sort.spill.percent is the buffer space threshold at which a spill is triggered. Optimal settings for these parameters depend on your data and on available memory and should be tuned for your specific workloads.
- **Compression of intermediate results and final output,** as specified by mapred.compress.map.output and mapred.output.compress, should be enabled for MapReduce tasks that shuffle lots of data between maps and reducers, such as sort and terasort workloads. (For more information on data compression, see the sidebar, Maximize Hadoop Performance through Optimized Data Compression.)

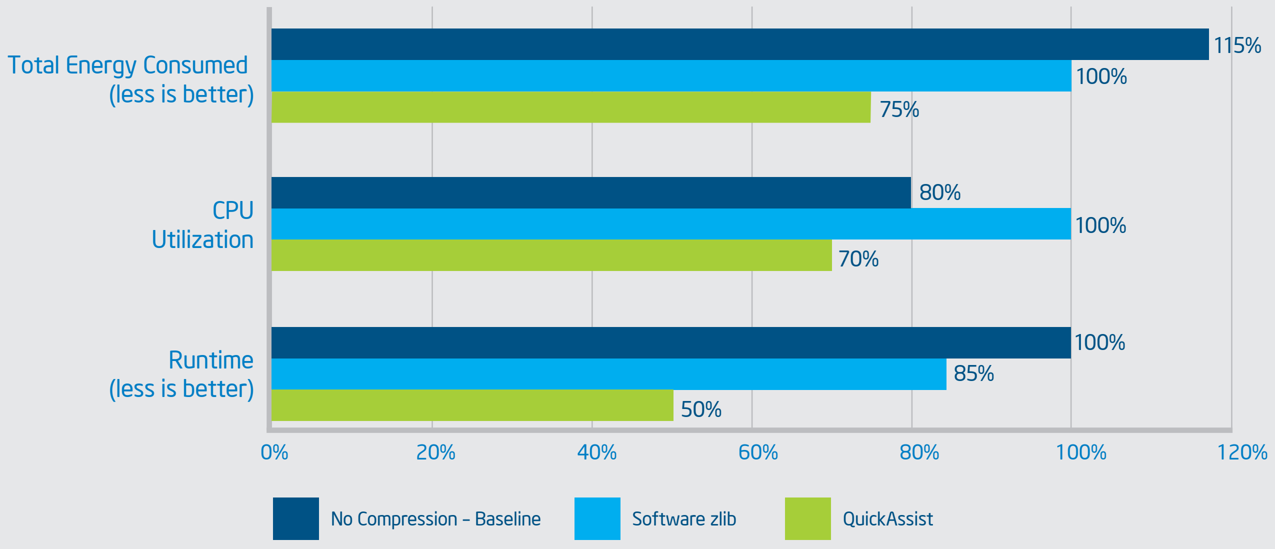
Maximize Hadoop* Performance Through Optimized Data Compression

Intel performance tests indicate that compressing data improves Hadoop performance by as much as 55 percent for I/O-intensive workloads. A number of different compression codecs can be used with Hadoop. Snappy is popular, because it delivers superior performance with reduced processing overhead compared with other codecs. However, Snappy is not compatible with zlib, which many IT organizations use for general-purpose data center file compression.

If zlib compatibility is important in your environment, consider using Intel® QuickAssist Technology to provide hardware assistance for zlib data compression. This strategy provides comparable performance to Snappy, while maintaining zlib compatibility.

For more information and test details, read the Intel solution brief, Accelerating Hadoop Applications Using Intel® QuickAssist Technology.

Hadoop Text Sort Example



Results based on Textsort benchmark running without data compression, with zlib data compression, and with zlib data compression plus Intel QuickAssist Technology. on two Data Node servers, each configured with 2 Intel® Xeon® processors E5-2680 v2, Intel® Communications Chipset 8950, 128 GB DRAM @ 1600 MHz, and an Intel® 82598EB 10 Gigabit Ethernet Controller card, with 1 x 32 GB Intel® X25-E Extreme SATA Solid State Drives for operating system storage, 8x300 GB Intel® Solid State Drives 710 Series, and a 6 GB/s link to an external storage array with 24x32 GB Intel X25-E Extreme SATA Solid State Drives. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Reduce Task-Specific Configurations

- **The number of parallel copier threads** during the reduce shuffle phase, as specified by mapred.reduce.parallel.copier, defaults to 5, but should be set to an optimal value based on performance testing with your specific workloads (typically between 5 and 25). This is an important setting, since performance losses as high as 5 to 30 percent are not uncommon if the value is either too high or too low.

Summary

Achieving optimal results from a Hadoop implementation begins with choosing appropriate hardware and software stacks. Fine-tuning the environment calls for in-depth analysis which can involve considerable time. However, the effort involved in the planning stages can pay off dramatically in terms of the performance and TCO associated with the environment. In addition to the testing at Intel labs described in this paper, the following composite system-stack recommendation offers a proven starting point for designing a best-fit Hadoop cluster.

Once your preliminary system configurations are complete, the tuning advice provided in this paper will help you make additional optimizations.

RECOMMENDED SYSTEM CONFIGURATION		
SERVER PLATFORM		
Two-socket for mainstream deployments	Four-socket for space- or power-constrained environments	
SERVER PROCESSOR		
Two-socket: Intel® Xeon® processor E5 v2 product family	Four-socket: Intel® Xeon® processor E7 product family	
STORAGE DRIVES		
For higher performance: Intel® SSD S3700	For higher capacity: 1 terabyte (TB) or 2 TB SATA drives	For mixed requirements: HDDS and Intel SSDs with Intel® Cache Acceleration Software (Intel® CAS) for transparent storage tiering
MEMORY		
1 GB per hardware thread (tune as needed based on workloads)		
NETWORK INTERFACE CARD (NIC)		
10 Gigabit Intel® Ethernet Server Adapters		
POWER SUPPLY		
80 PLUS* Gold Certified		
OPERATING SYSTEM		
Linux* based on kernel 2.6.30 or later		
JAVA* VIRTUAL MACHINE		
Sun Java 6u14 or later		
HADOOP VERSION		
Vendor-supported commercial distribution		

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

[1] Performance comparisons using a variety of standard Hadoop benchmarks. With the baseline results considered to be 100 percent, the following normalized results were measured for the platform under test: Sort (262%), terasort (292 percent), kmeans (350 percent), high-join (320 percent), pagerank (274%). Baseline configuration: 2S server configured with the Intel® Xeon® processor E5-2680, 8 x 16 GB DDR3 1600 MHz memory, 3 x Intel® Solid State Drive DC S3700 Series (400 GB), 3 x Intel Solid State Drive 710 Series (200 GB), 2 x Intel® Solid State Drive 710 Series (300 GB), 1 x Intel® X25-E Extreme (64GB) for the OS drive; CentOS release 6.3 (2.6.32-358.el6.x86_64, Apache Hadoop 1.0.4, Intel® 82599EB 10 Gigabit Ethernet Controller, LSI Logic SAS2308 PCI-Express Fusion-MPT SAS-2 Raid Controller. System under test configuration: 4S server configured with the Intel® Xeon® processor E7-4890 v2, 96 x 16 GB DDR3 1600 MHz memory, 12 x Intel® Solid State Drive DC S3700 Series (400 GB), 1 x Intel® X25-E Extreme (64GB) for the OS drive; CentOS release 6.4 (2.6.32-279.19.1.el6.6x86_64), Apache Hadoop 1.0.4, Intel® 82598EB 10 Gigabit Ethernet Controller, Intel Raid Controller RS2BL040. Software settings (both systems): BIOS: TurboBoost enabled, SpeedStep disabled, Hadoop Basic settings: DFS Replication set to 1, JVM Reuse enabled, OS Settings: 'deadline' disk scheduler, transparent large pages off, ipv6 disabled, iptables disabled, synchronize via ntp. For more information, look for the Intel white paper, "Accelerating Big data Analysis with Intel® Technologies" on the Intel® Developer Zone.

[2] Performance comparisons using a variety of standard Hadoop benchmarks. With the baseline results considered to be 100 percent, the following normalized results were measured for the platform under test: K-means Iteration (137% percent), Sort (137%), Hive Join (163 %). Baseline configuration: 2S server configured with the Intel® Xeon® processor E5-2680 (8-core, 2.70 GHz), 8 x 16 GB DDR3 1600 MHz memory, 3 x Intel® Solid State Drive DC S3700 Series (400 GB), 3 x Intel Solid State Drive 710 Series (200 GB), 2 x Intel® Solid State Drive 710 Series (300 GB), 1 x Intel® X25-E Extreme (64GB) for the OS drive; CentOS release 6.3 (2.6.32-358.el6.x86_64, Apache Hadoop 1.0.4, Intel® 82599EB 10 Gigabit Ethernet Controller, LSI Logic SAS2308 PCI-Express Fusion-MPT SAS-2 Raid Controller. System under test configuration: 2S server configured with the Intel® Xeon® processor E5-2680 v2 (10-core, 2.8 GHz), 10 x 16 GB DDR3 1600 MHz memory, 12 x Intel® Solid State Drive DC S3700 Series (400 GB), 1 x Intel® X25-E Extreme (64GB) for the OS drive; CentOS release 6.4 (2.6.32-279.19.1.el6.6x86_64), Apache Hadoop 1.0.4, Intel® 82598EB 10 Gigabit Ethernet Controller, Intel Raid Controller RS2BL040. Software settings (both systems): BIOS: TurboBoost enabled, SpeedStep disabled, Hadoop Basic settings: DFS Replication set to 1, JVM Reuse enabled, OS Settings: 'deadline' disk scheduler, transparent large pages off, ipv6 disabled, iptables disabled, synchronize via ntp.

[3] Performance comparison using Sort benchmark. Test configurations: Name Node running on a 2S server configured with 2 Intel® Xeon® processor X5570 platform with 4 cores and 8 threads per processor; Data Node running on two 2S servers, each configured with 2 Intel® Xeon® processor E5-2680 with 8 cores and 8 threads per processor (HT off) and 128 GB of DDR3 1600 MHz memory. The Data Node servers were connected together with an Intel® 82598EB 10-Gigabit Ethernet network and to the Name Node server with an Intel® 82574L Gigabit Network. Software included CentOS 6.2_64 (kernel 2.6.32-279.19.1.el6.x86_64), Java Hotspot* 1.7.0_13 64-bit Server VM (build 23.7-b01), and Apache Hadoop 1.0.4 (with several performance fixes). Baseline result of 289 MB/s using 8x500 GB HDDs with a sustained data rate of 115 MB/s. Test result of 776 MB/s using 8x400 GB Intel® SSD DC S3700.

[4] Performance comparison using geometric mean of SPECint*_rate_base2006, STREAM*_MP Triad, and Linpack* benchmark results. Baseline geometric means score of 166.75 on prior generation 2S Intel® Xeon® processor X5690 platform based on best published SPECrate* scores to www.spec.org and best Intel internal measurements on STREAM_MP Triad and Linpack as of 5 December 2011. New geometric mean score of 306.74 based on Intel internal measured estimates using an Intel® Rose City platform with two Intel® Xeon® processor E5-2690, Turbo and EIST Enabled, with Hyper-Threading, 128 GB RAM, Red Hat Enterprise Linux* Server 6.1 beta for x86_6, Intel® Compiler 12.1, THP disabled for SPECfp_rate_base2006 and enabled for SPECint_rate_base2006.

[5] Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

♻ Please Recycle 1213/DBF/HBD/PDF 330003-001US