WHITE PAPER
Cloud Computing

Optimizing Hadoop* Deployments

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

Tuning Hadoop* clusters is vital to improve cluster performance, optimize resource utilization, and minimize operating costs. Tests conducted in Intel labs have established a number of best practices to help meet those goals.

EXECUTIVE SUMMARY

This paper provides guidance, based on extensive lab testing conducted with Hadoop* at Intel, to organizations as they make key choices in the planning stages of Hadoop deployments. It begins with best practices for establishing server hardware specifications, helping architects choose optimal combinations of components. Next, it discusses the server software environment, including choosing the OS and version of Hadoop. Finally, it introduces some configuration and tuning advice that can help improve results in Hadoop environments.

1 Overview

Having moved beyond its origins in search and Web indexing, Hadoop is becoming increasingly attractive as a framework for large-scale, data-intensive applications. Because Hadoop deployments can have very large infrastructure requirements, hardware and software choices made at design time can have a significant impact on performance and TCO.

Intel is a major contributor to open source initiatives, such as Linux*, Apache*, and Xen*, and has also devoted resources to Hadoop analysis, testing, and performance characterizations, both internally and with fellow travelers such as HP and Cloudera. Through these technical efforts, Intel has observed many practical trade-offs in hardware, software, and system settings that have real-world impacts.

This paper discusses some of those optimizations, which fall into three general categories:

- **Server hardware.** This set of recommendations focuses on choosing the appropriate hardware components for an optimal balance between performance and both initial and recurring costs.
- **System software.** In addition to the choice of OS and Java* Virtual Machine (JVM), the specific version of Hadoop and other software components have implications for performance, stability, and other factors.
- **Configuration and tuning.** The settings made to the Hadoop environment itself are an important factor in getting the full benefit of the rest of the hardware and software solution stacks.

It is important to note that Hadoop deployments will vary considerably from customer to customer and from project to project. The suggestions for optimization in this paper are meant to be widely relevant to Hadoop, but results may be quite different depending on actual workloads.

October 2010

Version 2.0

Table of Contents

2 General Hadoop Cluster Topology

A typical Hadoop cluster consists of a two- or three-level architecture made up of rack-mounted servers. Each rack of servers is interconnected using a 1 Gigabit Ethernet (GbE) switch. Each rack-level switch is connected to a cluster-level switch, which is typically a larger port-density 10GbE switch that may span hundreds or thousands of servers. Those cluster-level switches may also interconnect with other cluster-level switches or even uplink to another level of switching infrastructure.

Servers in a Hadoop cluster can be categorized in the following special-purpose capacities:

- **JobTracker.** Performs task assignment.
- **NameNode.** Maintains all file system metadata if the Hadoop Distributed File System (HDFS) is used; preferably (but not required to be) a separate physical server from JobTracker.
- **Secondary NameNode.** Periodically check-points the file system metadata on the NameNode.
- **TaskTracker.** Performs map/reduce tasks.
- **DataNode.** Stores HDFS files and handles HDFS read/write requests; preferably co-located with TaskTracker for optimal data locality.

Most Hadoop servers are configured in the TaskTracker and DataNode capacities, and these are considered "slave nodes."

For JobTracker and NameNode, it is important to consider deploying additional RAM and secondary power supplies to ensure the highest performance and reliability of these critical servers in the cluster. Given Hadoop's data distribution model, however, it typically does not make sense to deploy power redundancy on the slave nodes.

3 Server Hardware Configurations

One of the most important decisions in planning a Hadoop infrastructure deployment is the number, type, and configuration of the servers to be installed in the cluster. While the hardware considerations in this section are generally applicable to all servers in the Hadoop cluster, the focus here is on the slave nodes, which represent the majority of the infrastructure.

As with other workloads, depending on the specific Hadoop application, computation may be bound by I/O, memory, or processor resources. System-level hardware must be adjusted on a case-by-case basis, but the general guidelines suggested in this section provide a point of departure for that fine-tuning.

3.1 Choosing a Server Platform

Typically, dual-socket servers are optimal for Hadoop deployments. Servers of this type are generally more efficient, from a per-node, cost-benefit perspective, than large-scale multi-processor platforms for massively distributed implementations such as Hadoop. Similarly, dual-socket servers more than offset the added per-node hardware cost relative to entry-level servers through superior efficiencies in terms of load-balancing and parallelization overheads. Choosing hardware based on the most current platform technologies available helps to ensure the optimal intra-server throughput and energy efficiency.

To maximize the energy efficiency and performance of a Hadoop cluster, it is important to consider that Hadoop deployments do not require many of the features typically found in an enterprise data center server. The systems selected should use high-efficiency voltage regulators and be optimized for airflow; many vendors have designed systems based on Intel® Xeon® processors with those characteristics, largely for use by

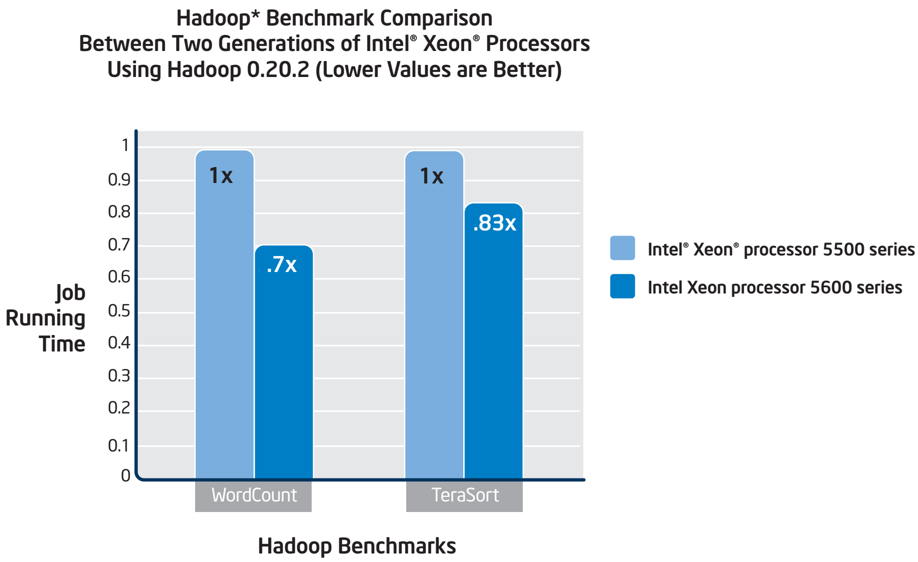

Figure 1. Successive generations of the Intel® Xeon® processor improve performance across a range of Hadoop* benchmarks.[2]

cloud computing or Internet data center providers. An example of such systems is the Intel® Xeon® processor 5600 series-based HP ProLiant* SL6000 family (used for much of the testing reported in this paper), which has been optimized for the cost, density, weight, and power consumption characteristics required in high-density computing environments.

3.2 Selecting and Configuring the Hard Drives

A relatively large number of hard drives per server (typically four to six) is recommended for general purpose Hadoop applications. For highly I/O intensive use cases, it is worth testing even higher ratios of disks to cores (for example, 12 drives per server). At the time of publication, the sweet spot for capacity is between 1 terabyte (TB) to 2 TB per drive. While it is possible to use RAID 0 to concatenate smaller drives into a larger pool, using RAID on Hadoop servers is generally not recommended because Hadoop itself orchestrates data provisioning and redundancy across individual nodes. This approach provides good results across a wide spectrum of workloads because of the way that Hadoop interacts with storage.

The optimal balance between cost and performance is generally achieved with 7,200 RPM SATA drives. This is likely to evolve quickly with the evolution of drive technologies, but it is a useful rule of thumb at the time of this writing. Hard drives should run in the Advanced Host Controller Interface (AHCI) mode with Native Command Queuing (NCQ) enabled, to improve performance when multiple simultaneous read/write requests are outstanding.

3.3 Memory Sizing

Sufficient memory capacity is critical for efficient operation of servers in a Hadoop cluster, supporting high throughput by allowing large numbers of map/reduce tasks to be carried out simultaneously. Typical Hadoop applications require approximately 12 gigabyte (GB) to 24 GB of RAM for servers based on the Intel Xeon processor 5600 series. It is recommended that dual in-line memory modules (DIMMs) be populated in multiples of six to balance across available DDR3 memory channels (that is, system configurations of 12 GB, 24 GB, and so on). As a final consideration, error-correcting code (ECC) memory is highly recommended for Hadoop, as a way to detect and correct memory errors introduced during storage and transmission of data.

3.4 Choosing Processors

The processor plays an important role in determining the speed, throughput, and efficiency of Hadoop clusters. The Intel Xeon processor 5600 series provides excellent performance for highly distributed workloads such as those associated with Hadoop applications.[1]

Lab testing was performed to establish the performance benefits of the Intel Xeon processor 5600 series relative to previous-generation Intel processors (see Figure 1). The results show the Intel Xeon processor 5600 series can provide as much as a 41 percent improvement in performance compared to the Intel® Xeon® processor 5500 series introduced in 2009.[2]

The latest Intel Xeon processor is not only faster at Hadoop tasks; it can also handle more throughput, defined here as the number of tasks completed per minute when the Hadoop cluster is at 100 percent utilization processing multiple Hadoop jobs. Intel lab tests have demonstrated that the Intel Xeon processor 5600 series provides substantially more throughput than previous processors, as shown in Figure 2. This characteristic allows Hadoop clusters to handle far larger data sets and more operations in a given amount of time.[3]

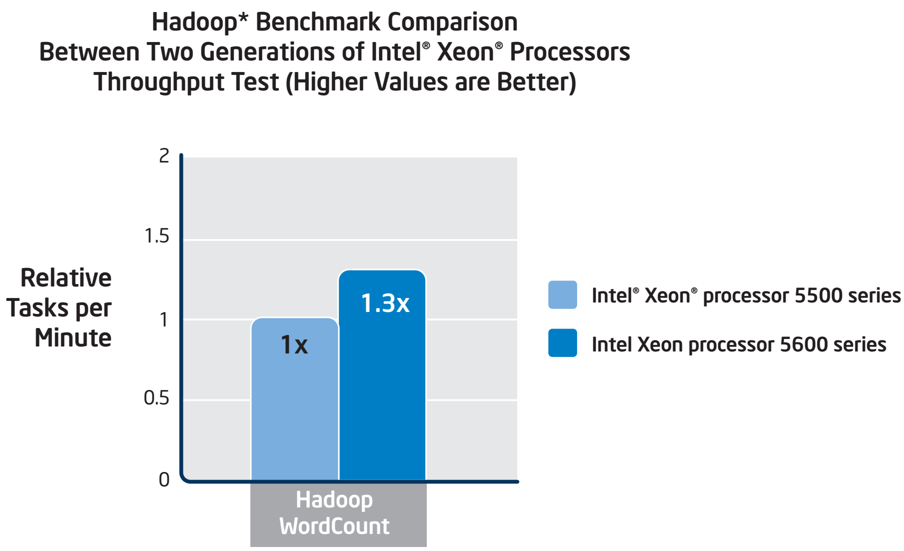

Figure 2. The Intel® Xeon® processor 5600 series provides higher throughput than its predecessors on selected Hadoop* benchmarks.[3]

In these internal Intel tests, Intel used HP ProLiant SL170z G6 servers that were configured identically with respect to networking, disk, and memory. To identify as much as possible the performance impact of two different processor generations, the tests were structured so that the Intel Xeon processors in the two systems were the only differences in hardware.

For Hadoop applications that are naturally compute intensive, such as WordCount and TeraSort, the benefit is clear. However, Intel testing also showed that for more I/O intensive applications, there was only a slight benefit to upgrading processors with no change in memory or disk. In such cases, the processor is under-utilized, and it is recommended that Hadoop architects increase the disk and memory in each node to fully take advantage of processing resources. This testing helps support Intel's recommendation of the Intel Xeon processor 5600 series as the server engine of choice for Hadoop clusters.

It is important to note that each Intel Xeon processor 5600 series typically has six cores, each of which can handle two threads when Intel® Hyper-Threading Technology (Intel® HT Technology) is enabled. Some Hadoop workloads, such as WordCount, can improve in performance by as much as 28 percent when running with Intel HT Technology enabled in the processor, compared to having the capability turned off.[4] It is recommended that customers running Hadoop clusters turn on Intel HT Technology in the BIOS. Consult your server OEM documentation for details.

3.5 Networking

Hadoop cluster performance and efficiency is highly dependent on network architecture and technology choices. A network strategy is beyond the scope of this paper, so the recommendations here are focused primarily on the server network interface. While Gigabit Ethernet is the most commonly deployed networking fabric for Hadoop today, it provides less than ideal bandwidth for many workloads. Hence, Intel testing has shown the benefit of channel bonding two GbE ports together for better I/O throughput.

In addition, networking hardware should support at least eight queues per port to ensure proper balancing of interrupt handler resources among the multiple processor cores in the server. This practice helps to avoid over-loading any single core with network interrupts. The Intel® 82576 Gigabit Ethernet Controller, which can be found on select server boards or on the Intel® Gigabit ET Dual Port server adaptor, is particularly recommended.

4 System Software Selection and Configuration

4.1 Selecting the Operating System and JVM

Using a Linux* distribution based on kernel version 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.[5] Such power inefficiency, multiplied over a large Hadoop cluster, could amount to significant additional energy costs.

For better performance, the local file systems (for example, ext3 or xfs) are usually mounted with the noatime attribute. In addition, Sun Java* 6 is required to run Hadoop, and the latest version (Java 6u14 or later) is recommended to take advantage of optimizations such as compressed ordinary object pointers.

The default Linux open file descriptor limit 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/sysctl.conf** file or alternate means.

4.2 Choosing Hadoop Versions and Distributions

When selecting a version of Hadoop for the implementation, organizations must seek a balance between the enhancements available from the most recent available release and the stability available from more mature versions. The lab testing reported in this paper was conducted with Cloudera's Hadoop distribution, hadoop-0.20.2-CDH3 beta 2 (hadoop patch level 320). Beginning with Hadoop version 0.20.0, important enhancements became available, including pluggable scheduling API, capacity scheduler, fair scheduler, and multiple task assignment.

One other potential advantage of using Hadoop 0.20.0 is in the area of performance. Intel's lab testing shows that some workloads within Hadoop can benefit from the multi-task assignment features in 0.20.0. Although the Map stage in v0.20.0 is slower and uses more memory than v0.19.1, the overall job runs at about the same speed or up to 8 percent faster in v0.20.0 in the case of Hadoop Sort.[6]

For companies planning Hadoop installations, it may be worthwhile to evaluate the Cloudera distribution, which includes RPM and Debian* packaging and tools for configuration. Intel has used Cloudera's distribution on some of its lab systems for performance testing. The primary source for securing the latest distribution is the Cloudera Web site (www.cloudera.com).

5 Hadoop Configurations and Tuning

To achieve maximum 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 optimization areas depends heavily on the unique characteristics of the individual application, so users are encouraged to experiment with their own systems and environment to achieve the best results.

5.1 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.

5.2 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.)

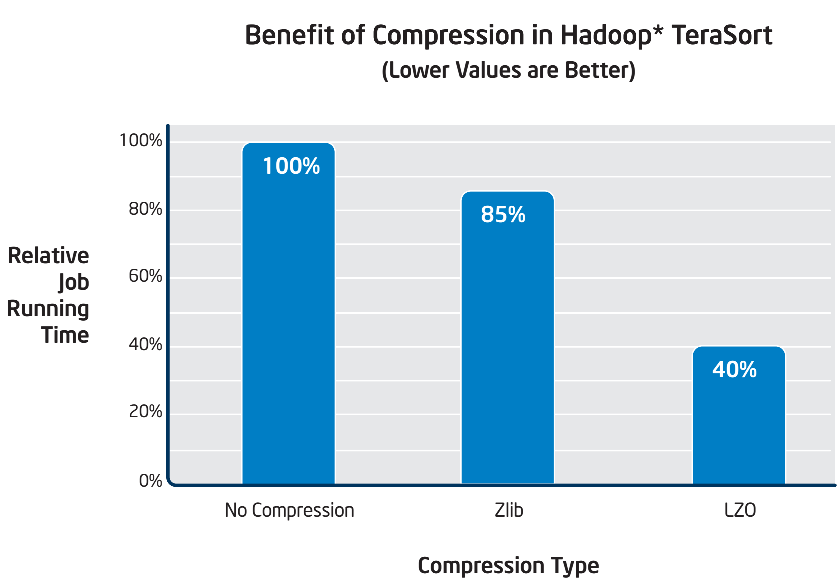

Figure 3. Using the LZO compression algorithm significantly improves Terasort speed.[7]

- **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.

5.3 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 in HDFS, and it is usually desirable to use a larger block size (such as 128 MB or even 256 M) for large file systems.

5.4 Map/Reduce-Specific Configurations

- **The maximum number of map/reduce 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 (files) to be merged at once in the map/reduce 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.preferIPv4Stack* set to true, to avoid timeouts in cases where the OS/JVM picks up an IPv6 address and must resolve the hostname.

5.5 Map Task-Specific Configurations

- **The total size of result and metadata buffers associated with a map task**, as specified by *io.sort.mb*, defaults to 100 MB and can be set to a higher level, such as 200 MB.
- **The percentage of total buffer size that is dedicated to the metadata buffer**, as specified by *io.sort.record.percent*, which defaults to 0.05, should be adjusted according to the key-value pair size of the particular Hadoop job.
- **Compression of intermediate results and final output**, as specified by *mapred.compress.map.output* and *mapred.output.compress*, should be enabled (especially for large clusters and large jobs). In addition, it is recommended to evaluate LZO as the compression codec (as specified by *mapred.map.output.compression.codec* and *mapred.output.compression.codec*). Intel's internal testing has demonstrated that using LZO compression with TeraSort reduces the job running time by 60 percent compared to no compression and is significantly faster than Zlib (see Figure 3).[7]

5.6 Reduce Task-Specific Configurations

- **The number of parallel copier threads during reduce shuffle phase**, as specified by *mapred.reduce.parallel.copier*, defaults to 5 and should be set to a larger number in the range of 16 to 25 for large clusters. Higher values may create I/O contention, so it is important to test for an optimal balance for the given application.

6 Conclusion

Achieving optimal results from a Hadoop implementation begins with choosing the correct hardware and software stacks. Fine-tuning the environment calls for a fairly 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 can help benefit organizations in the planning stages:

RECOMMENDED SYSTEM CONFIGURATION	
Server Processor	Intel® Xeon® processor 5600 series
Hard Disks	Four to six 1 terabyte (TB) or 2 TB 7200 RPM SATA drives
Memory	12-24 GB DDR3 R-ECC RAM
Network Interface Card (NIC)	1–2 1 gigabit NIC(s)
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	0.20.x (Cloudera's Distribution for Hadoop)

Once the preliminary system configurations are complete, the tuning advice given in this paper will help enable implementing organizations to improve their Hadoop environments further.

Join the conversation in The Server Room:
communities.intel.com/community/server

Benchmarking Detail and Disclaimers

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel® products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations (www.intel.com/performance/resources/benchmark_limitations.htm).

1 For more information on the Intel® Xeon® processor 5600 series, see http://www.intel.com/Assets/en_US/PDF/prodbrief/323501.pdf.

2 Source: Intel internal measurement as of August 8, 2010 running Hadoop* WordCount and TeraSort. Results: WordCount single job running time was 407 seconds on the Xeon® 5500® processor series and 289 seconds on the Intel® Xeon® 5600 processor series. TeraSort single job running time was 2,541 seconds on the Xeon processor 5500 series and 2,182 seconds on the Intel Xeon processor 5600 series. Hardware, cluster configuration, and settings were as follows:

(1 Namenode/JobTracker + 5 DataNode/TaskTracker, each has two port 1 GbE connectivity to a single GbE switch with channel bonding enabled.)

Intel Xeon processor 5600 series servers: HP ProLiant* z6000 G6 Server with 2x Intel® Xeon® processor X5670 2.93 GHz (12 cores per node), 24 GB DDR3 RAM, 6 SATA disks per node (All six for HDFS and intermediate results, sharing one for system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel® Hyper-Threading Technology enabled.

Intel Xeon processor 5500 series servers: HP ProLiant z6000 G6 Server with 2x Intel® Xeon® processor X5570 2.93 GHz (8 cores per node), 24 GB DDR3 RAM, 6 SATA disks per node (All six for HDFS and intermediate results, sharing one for system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel Hyper-Threading Technology enabled.

Software: Red Hat Enterprise Linux* 5.4 (with kernel 2.6.30x86_64). Ext4 file system, mounted with "noatime,nodiratime" options.). Sun JVM 1.6 (Java* version 1.6.0_14 Java SE Runtime Environment Java HotSpot* 64-bit server virtual machine). Cloudera distribution of Hadoop [hadoop-0.20.2-CDH3 beta 2 (hadoop patch level 320)].

3 Source: Intel internal measurement as of August 8, 2010 running Hadoop* WordCount and TeraSort. Results: Total completed tasks per minute of WordCount over Intel® Xeon® processor 5500 series was approximately 71.58, and over Intel® Xeon® process 5600 series was approximately 93.22. Hardware, cluster configuration, and settings were as follows:

(1 Namenode/JobTracker + 5 DataNode/TaskTracker, each has two port 1 GbE connectivity to a single GbE switch with channel bonding enabled.)

Intel Xeon processor 5600 series servers: HP ProLiant* z6000 G6 Server with 2x Intel Xeon processor X5670 2.93 GHz (12 cores per node), 24 GB DDR3 RAM, 6 SATA disks per node (All six for HDFS and intermediate results, sharing one for system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel® Hyper-Threading Technology enabled.

Intel Xeon processor 5500 series servers: HP ProLiant z6000 G6 Server with 2x Intel Xeon processor X5570 2.93 GHz (8 cores per node), 24 GB DDR3 RAM, 6 SATA disks per node (All six for HDFS and intermediate results, sharing one for system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel Hyper-Threading Technology enabled.

Software: Red Hat Enterprise Linux* 5.4 (with kernel 2.6.30x86_64). Ext4 file system, mounted with "noatime,nodiratime" options.). Sun JVM 1.6 (Java* version 1.6.0_14 Java SE Runtime Environment Java HotSpot* 64-bit server virtual machine). Cloudera distribution of Hadoop [hadoop-0.20.2-CDH3 beta 2 (hadoop patch level 320)].

4 Source: Intel internal measurement as of August 8, 2010 based on the following cluster and server configuration: 6 nodes (1 NameNode/JobTracker, 5 DataNode/TaskTracker) in each, configured with 2GbE connectivity to each server. Intel® Xeon® processor 5600 series servers: HP ProLiant* z6000 G6 Server 2 x Intel® Xeon® processor X5670 2.93 GHz (12 cores per node), 24 GB DDR3 RAM, 6 SATA disks per node (All six for HDFS and intermediate results, sharing one for file system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel® Hyper-Threading Technology (Intel® HT Technology) requires a computer system with an Intel® processor supporting Intel HT Technology and an Intel HT Technology-enabled chipset, BIOS, and operating system. Performance will vary depending on the specific hardware and software you use.

See www.intel.com/products/ht/hyperthreading_more.htm for more information including details on which processors support Intel HT Technology.

5 Source: Intel internal measurement as of September 14, 2009 based on running the same server with two different Linux* distributions: CentOS* 5.2 and Fedora* 11. Power (W) at idle was measured at 110W when running CentOS 5.2 and 68W when running Fedora 11. Server configuration was an Intel white box server based on the Intel® Server Board SB5000WB, 2 x Intel® Xeon® processor L5520, 16 GB RAM, 1 HDD.

6 Sources: Intel internal measurement on September 14, 2009 and September 1, 2009 running Hadoop* Sort benchmark. Raw score of cluster based on Intel® Xeon® processor 55xx series. Cluster architecture includes 5 nodes of Intel Xeon processor 55xx series. Each has 1 NameNode/JobTracker,

4 DataNode/TaskTracker. 1 x GbE connectivity to a single GbE switch. Intel Xeon processor 55xx series servers: 2 x Intel Xeon processor X5570 2.93 GHz (8 cores per node), 16 GB DDR3 RAM, 5 SATA disks per node (1 for system and log files, and the other 4 for HDFS and intermediate results). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. SMT (Simultaneous Multi-Threading) enabled.

Software: Red Hat Enterprise Linux* 5.2 (kernel 2.6.30 x86_64 x86_64). Ext3 file system (mounted with noatime enabled). Sun JVM 1.6 (Java* version 1.6.0_02 Java SE Runtime Environment Java HotSpot* 32-bit server virtual machine). Hadoop version 0.19.1 with patch JIRA Hadoop-5191vs. Hadoop 0.20.0.

7 Source: Intel internal measurement as of August 8,2010 running Hadoop* TeraSort. Results: TeraSort single job running time was 1477 seconds without compression, 1256 seconds with default(zlib) compression, and 586 seconds with LZO compression. Hardware, cluster configuration, and settings were as follows:

(1 NameNode/JobTracker + 32 DataNode/TaskTracker; each has 1 port 1 GbE connectivity to a single GbE switch)

Intel Xeon processor 5500 series servers: 2x Xeon processor X5570 2.93 GHz (8 cores per node), 24 GB DDR3 RAM, 4 SATA disks per node (All 4 for HDFS and intermediate results, sharing 1 for system and log files with isolated partition). Both EIST (Enhanced Intel® SpeedStep Technology) and Turbo mode disabled. Both hardware prefetcher and adjacent cache-line prefetch enabled. Intel Hyper-Threading Technology enabled.

Software: Red Hat Enterprise Linux* 5.4 (with kernel 2.6.30.10 x86_64). Ext3 filesystem, mounted with "noatime,nodiratime" options.). Sun JVM 1.6 (Java* version 1.6.0_14 Java SE Runtime Environment Java HotSpot* 64-bit server virtual machine). Hadoop 0.20.1 version.

Disclaimers & Legal Notices

1010/KH/MESH/PDF 322723-002US