



WHITE PAPER

Increasing Hadoop Performance with SanDisk® Solid State Drives (SSDs)

July 2014

SanDisk®
a Western Digital brand

Western Digital Technologies, Inc.
951 SanDisk Drive, Milpitas, CA 95035

www.SanDisk.com

Table of Contents

1. Executive Summary	3
2. Apache Hadoop	4
3. Hadoop and SSDs	4
4. Terasort Benchmark	5
5. Test Design	5
6. Test Environment	6
7. Technical Component Specifications	7
Hardware	7
Software	7
8. Compute Infrastructure	7
9. Network Infrastructure	8
10. Storage Infrastructure	8
11. Cloudera Hadoop Configuration	9
12. Operating System Configuration	9
13. Test Validation	10
Test Methodology	10
14. Results Summary	12
15. Results Analysis	13
Performance Analysis	13
16. TCO Analysis (Cost/Job Analysis)	14
17. Conclusions	16
18. References	17

1. Executive Summary

This technical paper describes 1TB Terasort testing conducted on a Big Data/Analytics Hadoop cluster using SanDisk solid-state drives (SSDs). The goal of this paper is to show the benefits of using SanDisk SSDs within a scale-out Hadoop cluster environment.

SanDisk's CloudSpeed Ascend™ SATA SSDs product family was designed specifically to address the growing need for SSDs that are optimized for mixed-use application workloads in enterprise server and cloud computing environments. CloudSpeed Ascend SATA SSDs offer all the features expected from an enterprise drive, and support accelerated performance at a good value.

CloudSpeed Ascend SATA SSDs provide a significant performance improvement when running I/O-intensive workloads, especially with a mixed-use application that has random read and random write data access, compared with what is traditional sequential read and write data access with spinning HDDs. On a Hadoop cluster, this performance improvement directly translates into faster job completion, therefore better utilization of the Hadoop cluster. This performance improvement helps to reduce the cluster footprint by using fewer cluster nodes, therefore reducing the total cost of ownership (TCO).

For over 25 years, SanDisk has been transforming digital storage with breakthrough products and ideas that push the boundaries of what's possible. SanDisk flash memory technologies are used by many of the world's largest data centers today. For more information visit: www.sandisk.com.

Summary of Flash-enabled Hadoop Cluster Testing: Key Findings

SanDisk tested a Hadoop cluster using the Cloudera® Distribution of Hadoop (CDH). This cluster consisted of one (1) NameNode and six (6) DataNodes, which was set up for the purpose of determining the benefits of using SSDs within a Hadoop environment, focusing on the Terasort benchmark.

SanDisk ran the standard Hadoop Terasort benchmark on different cluster configurations. The tests revealed that SSDs can be deployed strategically in Hadoop environments to provide significant performance improvement (1.4x-2.6x for the Terasort benchmark)¹ and TCO benefits (22%-53% reduction in cost/job for the Terasort benchmark)² to organizations that deploy them. In summary, these tests showed that the SSDs are beneficial for Hadoop environments that are storage-intensive and that see a very high proportion of random access to the storage. This technical paper discusses the Terasort tests on a flash-enabled cluster, and provides a proof-point for SSD performance and SSD TCO benefits within the Hadoop ecosystem.

We should note that the runtime for the Terasort benchmark for a 1TB dataset was recorded for the different configurations. The results of these runtime tests are summarized and analyzed in the Results Summary and Results Analysis sections. Based on these tests, this paper also includes recommendations for using SSDs within a Hadoop configuration.

¹Please refer to results shown in Figure 3.

²Please refer to results shown in Figure 4.

2. Apache Hadoop

Apache Hadoop is a framework that allows for the distributed processing of large datasets across clusters of computers using simple programming models. It is designed to scale up from a single server to several thousands of servers, with each server offering local computation and storage. Rather than relying on the uptime for any single hardware device to deliver high availability, Hadoop is designed to detect and handle failures at the application layer, thus delivering a highly available service on top of a cluster of computers, each of which may be prone to failures.

Hadoop Distributed File System (HDFS) is the distributed storage used by Hadoop applications. A HDFS cluster consists of a NameNode that manages the file system metadata, and DataNodes that store the actual data. Clients contact the NameNode for file metadata or file modifications and they perform actual file I/O operations directly with DataNodes.

Hadoop MapReduce is a software framework for easily writing applications that process vast amounts of data (multi-terabyte data-sets) in parallel on large Hadoop clusters (with thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. A MapReduce job usually splits the input data-set into independent chunks that are processed by the Map tasks in a completely parallel manner. The framework sorts the outputs of the Maps, which are then input to the Reduce tasks. Typically, both the input and output of a MapReduce job are stored on the HDFS. The framework takes care of scheduling tasks, monitoring them and re-executing the failed tasks. The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves then execute tasks as directed by the master.

3. Hadoop and SSDs

Typically, Hadoop environments use commodity servers, with SATA HDDs as local storage on cluster nodes. However, SSDs, when used strategically within a Hadoop environment, can likely provide significant performance benefits.

Hadoop workloads have a lot of variation in terms of their storage access profiles. Some Hadoop workloads are compute-intensive, some are storage-intensive and some are in between. Many Hadoop workloads use custom datasets and customized MapReduce algorithms to execute very specific analysis tasks on the datasets.

SSDs in Hadoop environments will benefit storage-intensive datasets and workloads, especially those with a very high proportion of random I/O access. Additionally, Hadoop MapReduce workloads showing significant random storage accesses during the intermediate phases (shuffle/sort phases between the Map and Reduce phases) will also see performance improvements when using SSDs for the intermediate data accesses.

This technical paper discusses one such workload benchmark called Terasort.

4. Terasort Benchmark

Terasort is a standard Hadoop benchmark. It is an I/O-intensive workload that sorts a very large amount of data using the MapReduce paradigm. The input and output data are stored on HDFS. Terasort benchmark consists of the following components:

1. **Teragen:** This component generates the dataset for the benchmark. The dataset is in the form of key-value pairs generated randomly.
2. **Terasort:** This component performs the sorting based on the keys in the dataset.
3. **Teravalidate:** This component is used to validate the result of Terasort.

The testing discussed in this technical paper primarily focuses on Teragen and Terasort.

In order to start Teragen on a 1TB dataset, one should execute the following command as user 'hdfs' with the size of the dataset and the input directory as parameters. Note that the size of the dataset is provided as a parameter and is counted in multiples of 100 bytes:

```
# hadoop jar /opt/cloudera/parcels/CDH-4.4.0-1.cdh4.4.0.p0.39/lib/hadoop-mapreduce/  
hadoop-mapreduce-examples.jar teragen 10000000000 /user/sort-in
```

To start Terasort, use the following command, giving as input the input directory used by Teragen and the output directory to write the sorted data:

```
# hadoop jar /opt/cloudera/parcels/CDH-4.4.0-1.cdh4.4.0.p0.39/lib/hadoop-mapreduce/  
hadoop-mapreduce-examples.jar terasort /user/sort-in /user/sort-out.
```

5. Test Design

A Hadoop cluster using the Cloudera Distribution of Hadoop (CDH) consisting of one (1) NameNode and six (6) DataNodes associated with the HDFS file system was set up for the purpose of determining the benefits of using SSDs within a Hadoop environment, focusing on the Terasort benchmark. The testing consisted of using the standard Hadoop Terasort benchmark on different cluster configurations (described in the Test Methodology section). The runtime for the Terasort benchmark doing the sort on a 1TB dataset was recorded for the different configurations. The results of these runtime tests are summarized and analyzed in the Results Summary and Results Analysis sections. And finally, the paper ends with recommendations for using SSDs within a Hadoop configuration.

6. Test Environment

The test environment consisted of one (1) Dell PowerEdge R720 server being used as a NameNode for a Hadoop cluster with six (6) Dell PowerEdge R320 servers being used as DataNodes within this cluster. A 10GbE private network interconnect was used on all of the servers within the Hadoop cluster for Hadoop communication. Each of the nodes also used a 1 GbE management network. The local storage was varied, depending on the type of test configuration (HDDs or SSDs).

Figure 1. shows a pictorial view of the test environment, which is followed by a chart describing the hardware and software components that were used within the test environment.

Hadoop Cluster Topology

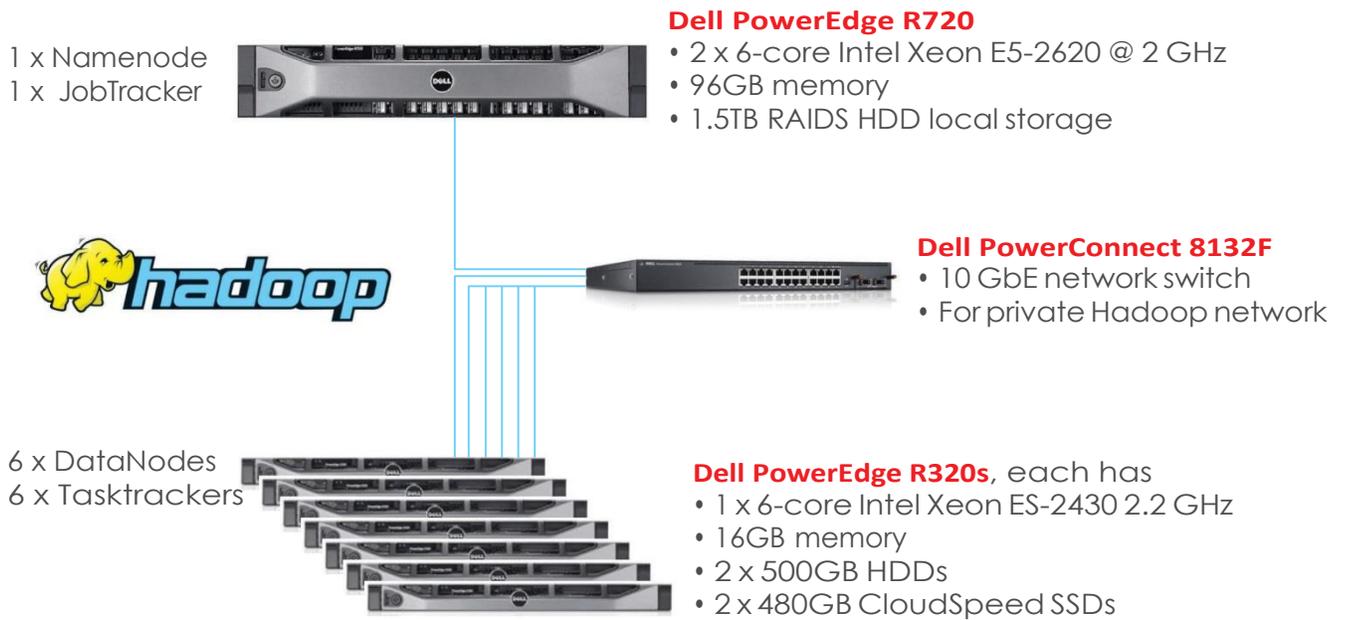


Figure 1.: Cloudera Hadoop Cluster Environment

7. Technical Component Specifications

Hardware

Hardware	Software if applicable	Purpose	Quantity
Dell PowerEdge R320 <ul style="list-style-type: none"> 1 x Intel Xeon E5-2430 2.2 GHz, 6 core CPU, hyper-threaded 16GB memory HDD Boot drive 	<ul style="list-style-type: none"> RHEL 6.4 Cloudera Distribution of Hadoop 4.4.0 	DataNodes	6
Dell PowerEdge R720 <ul style="list-style-type: none"> 2x Intel Xeon E5-2620 2 Ghz 6 core CPUs 96GB memory SSD bootdrive 	<ul style="list-style-type: none"> RHEL 6.4 Cloudera Distribution of Hadoop 4.4.0 Cloudera Manager 4.8.1 	NameNode, Secondary NameNode	1
Dell PowerConnect 2824 24-port switch	1 GbE network switch	Management network	1
Dell PowerConnect 8132F 24-port switch	10 GbE network switch	Hadoop data network	1
500GB 7.2K RPM Dell SATA HDDs	Used as Just a bunch of disks (JBODs)	DataNode drives	12
480GB CloudSpeed Ascend™ SATA SSDs	Used as Just a bunch of disks (JBODs)	DataNode drives	12
Dell 300GB 15K RPM SAS HDDs	Used as a single RAID 5 (5+1) group	NameNode drives	6

Table 1.: Hardware components

Software

Software	Version	Purpose
Red Hat Enterprise Linux	6.4	Operating system for DataNodes and NameNode
Cloudera Manager	4.8.1	Cloudera Hadoop cluster administration
Cloudera Distribution of Hadoop (CDH)	4.4.0	Cloudera's Hadoop distribution

Table 2.: Software components

8. Compute Infrastructure

The Hadoop cluster NameNode is a Dell PowerEdge R720 server with two hex-core Intel Xeon E5-2620 2 GHz CPU (hyper-threaded) and 96GB of memory. This server has a single 300GB SSD that was used as a boot drive. The server had dual power-supplies for redundancy and high-availability purposes.

The Hadoop cluster DataNodes included six (6) Dell PowerEdge R320 servers, each with one (1) hex-core Intel Xeon E5-2430 2.2 GHz CPUs (hyper-threaded) and 16GB of memory. Each of these servers had a single 500GB 7.2K RPM SATA HDD that was used as a boot drive. The servers have dual power supplies for redundancy and high-availability purposes.

9. Network Infrastructure

All cluster nodes (NameNode and all DataNodes) were connected to a 1GbE management network via the onboard 1GbE NIC. All cluster nodes were also connected to a 10GbE Hadoop cluster network with an add-on 10GbE network interface card (NIC). The 1GbE management network was connected to a Dell PowerConnect 2824 24-port 1GbE switch. The 10GbE cluster network was connected to a Dell PowerConnect 8132F 10GbE switch.

10. Storage Infrastructure

The NameNode used six (6) 300GB 15K RPM SAS HDDs in RAID5 configuration for the HDFS. This NameNode setup was used across all the different testing configurations. The RAID5 logical volume was formatted as an ext4 file system and was mounted for use by the Hadoop NameNode.

Each DataNode had one of the following storage environments, as listed below, depending on the configuration being tested. The specific configurations are discussed in detail in the Test Methodology section of this paper:

1. 2 x 500GB 7.2K RPM Dell SATA HDDs, OR
2. 2 x 480GB CloudSpeed Ascend SATA SSDs, OR
3. 2 x 500GB 7.2K RPM Dell SATA HDDs and 1 x 480GB CloudSpeed Ascend SATA SSD

In each of the above environments, the disks were used in a JBOD configuration.

11. Cloudera Hadoop Configuration

Configuration parameter	Value	Purpose
dfs.namenode.name.dir (NameNode)	/data1/dfs/nn /data1 is mounted on the RAID5 logical volume on the NameNode.	Determines where on the local file system the NameNode should store the name table (fsimage).
dfs.datanode.data.dir (DataNode)	/data1/dfs/dn, /data2/dfs/dn Note: /data1 and /data2 are mounted on HDDs or SSDs, depending on which storage configuration is used.	Comma-delimited list of directories on the local file system where the DataNode stores HDFS block data.
mapred.job.reuse.jvm.num.tasks	-1	Number of tasks to run per Java Virtual Machine (JVM). If set to -1, there is no limit.
mapred.output.compress	Enabled	Compress the output of MapReduce jobs.
MapReduce Child Java Maximum Heap Size	512MB	The maximum heap size, in bytes, of the Java child process. This number will be formatted and concatenated with the 'base' setting for 'mapred.child.java.opts' to pass to Hadoop.
mapred.tasktracker.map.tasks.maximum	12	The maximum number of map tasks that a TaskTracker can run simultaneously.
mapred.tasktracker.reduce.tasks.maximum	6	The maximum number of reduce tasks that a TaskTracker can run simultaneously.
mapred.local.dir (job tracker)	/data1/mapred/jt, /data1 is mounted on the RAID5 logical volume on the NameNode.	Directory on the local file system where the JobTracker stores job configuration data.
mapred.local.dir (task tracker)	/data1/mapred/local OR /ssd1/mapred/local, depending on which storage configuration is used in the cluster.	List of directories on the local file system where a TaskTracker stores intermediate data files.

Table 3.: Cloudera Hadoop configuration parameters

12. OperatingSystem Configuration

The following configuration changes were made to the Red Hat Enterprise Linux (RHEL) 6.4 operating system parameters.

- As per Cloudera recommendations, swapping factor on the operating system was changed to 20 from the default of 60 to avoid unnecessary swapping on the Hadoop DataNodes. This can also be changed to 0 to completely switch off swapping (in this mode, swapping happens only if absolutely necessary for the OS operation).

```
# sysctl -w vm.swappiness=20
```

2. All file systems related to the Hadoop configuration were mounted via `/etc/fstab` with the 'noatime' option as per Cloudera recommendations. For example, for one of the configurations, `/etc/fstab` had the following entries.

```
/dev/sdb1 /data1 ext4 noatime 0 0
/dev/sdc1 /data2 ext4 noatime 0 0
/dev/sdd1 /ssd1 ext4 noatime 0 0
```

3. The open files limit was changed from 1024 to 16384, per Cloudera recommendations. This required updating `/etc/security/limits.conf` as below,

```
* Soft nofile 16384
* Hard nofile 16384
```

And `/etc/pam.d/system-auth`, `/etc/pam.d/sshd`, `/etc/pam.d/su`, `/etc/pam.d/login` were updated to include:

```
session include system-auth
```

13. Test Validation

Test Methodology

The goal of this technical paper is to showcase the benefits of using SSDs within a Hadoop environment. To achieve this goal, SanDisk tested three separate configurations of the Hadoop cluster with the standard Terasort Hadoop benchmark on a 1TB dataset. The three configurations are described in detail as follows. Note that there is no change to the NameNode configuration and it remains the same across all configurations.

1. All-HDD configuration

The Hadoop DataNodes use HDDs for the HDFS, as well as Hadoop MapReduce.

- a. Each DataNode has two HDDs set up as JBODs. The devices were partitioned with a single partition and were formatted as ext4 file systems. These were then mounted in `/etc/fstab` to `/data1` and `/data2` with the `noatime` option. `/data1` and `/data2` were then used within the Hadoop configuration for DataNodes (`dfs.datanode.data.dir`) and `/data1` was used for task trackers directories (`mapred.local.dir`).

2. HDD with SSD for intermediate data

In this configuration, Hadoop DataNodes used HDDs as in the first configuration, along with a single SSD which was used in the MapReduce configuration, as explained below.

- a. Each DataNode has two HDDs setup as JBODs. The devices were partitioned with a single partition, and then were formatted as ext4 file systems. These were then mounted in `/etc/fstab` to `/data1` and `/data2` with the `noatime` option. `/data1` and `/data2` were then used within the Hadoop configuration for the nodes directories (`dfs.datanode.data.dir`).
- b. In addition to the HDDs being used on the DataNodes, there was also a single SSD on each DataNode, which was partitioned with a 4K-divisible boundary via `fdisk` (shown below) and then formatted to have ext4 file system.

```
[root@hadoop2 ~]# fdisk -S 32 -H 64 /dev/sdd
```

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to switch off the mode (command `'c'`) and change display units to sectors (command `'u'`).

```
Command (m for help): c
```

```
DOS Compatibility flag is not set
```

```
Command (m for help): u
```

```
Changing display/entry units to sectors
```

```
Command (m for help): n
```

```
Command action
```

```
  e extended
```

```
  p primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First sector (2048-937703087, default 2048):
```

```
Using default value 2048
```

```
Last sector, +sectors or +size{K,M,G} (2048-937703087, default 937703087):
```

```
Using default value 937703087
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

This SSD was then mounted in `/etc/fstab` as `/ssd1` with the `noatime` option. This SSD was used in the shuffle/sort phase of MapReduce by updating the `mapred.local.dir` configuration parameter within the Hadoop configuration. The shuffle/sort phase is the intermediate phase between the Map phase and the Reduce phase and it is typically I/O-intensive, with a significant proportion of random access. SSDs show their maximum benefit in Hadoop configurations with this kind of random I/O-intensive workloads, and therefore this configuration was relevant to this testing.

3. All-SSD configuration

In this configuration, the HDDs of the first configuration were swapped out to all SSDs.

- a. Each DataNode had two (2) SSDs set up as JBODs. The devices were partitioned with a single partition with a 4K divisible boundary (as shown in configuration 2 details above) and then were formatted as ext4 file systems. These were then mounted in /etc/fstab to /data1 and /data2 with the noatime option. /data1 and /data2 are then used within the Hadoop configuration for DataNodes (dfs.datanode.data.dir) and /data1 is used for task trackers (mapred.local.dir).

For each of the above configurations, Teragen and Terasort were executed for a 1TB dataset. The time taken to run Teragen and Terasort was recorded for analysis.

14. Results Summary

Terasort benchmark runs were conducted on the three configurations described in the Test Methodology section. The runtime for completing Teragen and Terasort on a 1TB dataset was collected. The runtime results from these tests are summarized in Figure 2 below. The X-axis on the graph shows the different configurations and the Y-axis shows the runtime in seconds. The runtimes are shown for Teragen (blue columns), Terasort (red columns) — and for the entire run, which includes both the Teragen and Terasort results (green columns).

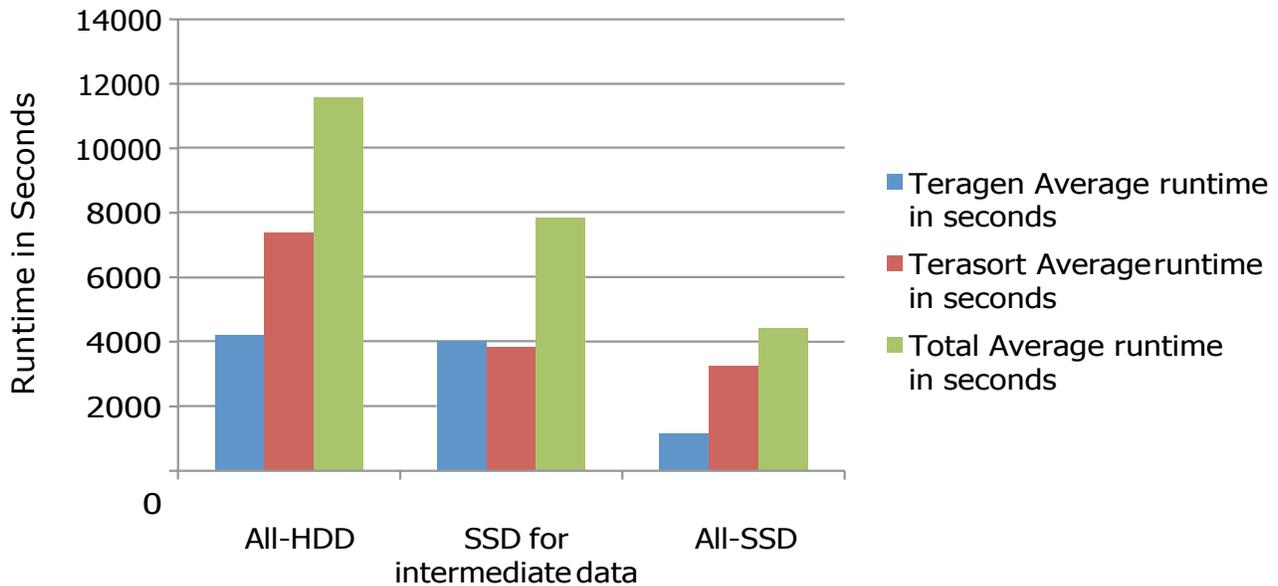


Figure 2.: Runtime comparisons

The results shown above in graphical format are also shown in tabular format in Table 4.:

Configuration	Teragen runtime (secs)	Terasort runtime (secs)	Total runtime (secs)
All-HDD configuration	4202.5	7386.5	11589
HDD with SSD for intermediate data	4014	3836	7850
All-SSD configuration	1148	3262.5	4410.5

Table 4.: Results summary

15. Results Analysis

Performance Analysis

The total runtime results (with the Teragen and Terasort runtimes taken together), as shown in Table 4. are, once again, shown in graphical format in Figure 3. below, with emphasis on the runtime improvements that was seen with the SSD configurations.

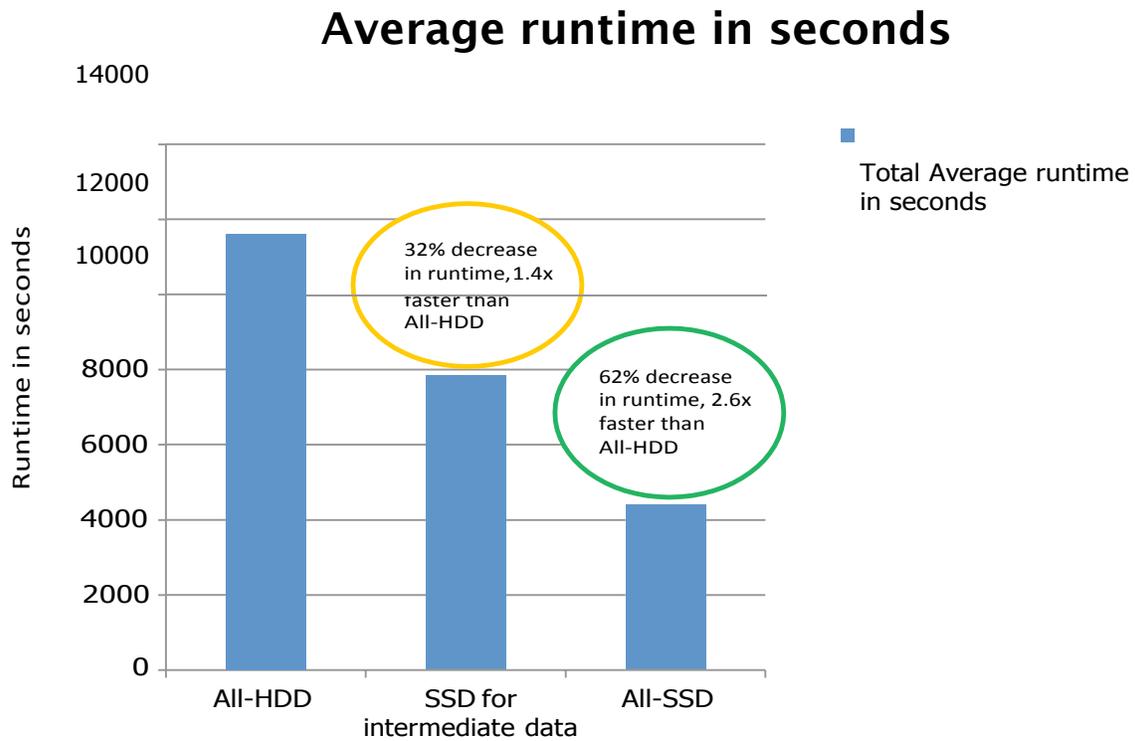


Figure 3.: Runtime comparisons – SSD vs. HDD

Observations from these results:

1. Introducing SSDs for the intermediate shuffle/sort phase within MapReduce can help reduce the total runtime of a 1TB Terasort benchmark run by ~32%, therefore completing the job 1.4x faster³ than it would have been on an all-HDD configuration.
 - a. For the Terasort benchmark, the total capacity required for the MapReduce intermediate shuffle/sort phase is around 60% of the total size of the dataset (so, for example, a 1TB dataset requires a total of 576 GB space for the intermediate phase).
 - b. The total capacity requirement of the shuffle/sort phase is divided up equally amongst all the available DataNodes of the Hadoop cluster. So, for example, in the test environment, for the 1TB dataset, around 96GB of capacity is required per DataNode for the MapReduce shuffle/sort phase.
 - c. Although the tests have used a 480GB CloudSpeed Ascend SATA SSD for the MapReduce shuffle/sort phase per DataNode, the same results could be achieved with a lower capacity SSD (with a capacity of 100GB or 200GB), which would have made the configuration more price-competitive than the 480GB SSD.
 - d. Note that the Teragen benchmark does not have a shuffle/sort phase, and therefore there is no significant change in the runtime from the all-HDD configuration.
2. Replacing all the HDDs on the DataNodes with SSDs can reduce the 1TB Terasort benchmark runtime by ~62%, therefore completing the job 2.6x faster⁴ than on an all-HDD configuration.
 - a. There are significant performance benefits when replacing the all-HDD configuration with an all-SSD configuration. Having faster job completion effectively translates to a much more efficient use of the Hadoop cluster by running more number of jobs within the same period of time.
 - b. More jobs on the Hadoop cluster will translate to savings in the total cost of ownership (TCO) of the Hadoop cluster in the long run (for example, over a period of 3-5 years), even if the initial investment may be higher due to the cost of the SSDs. This is discussed further in the next section.

16. TCO Analysis (Cost/Job Analysis)

Hadoop has been architected for deployment with commodity servers and hardware. So, typically, Hadoop clusters use cheap SATA HDDs for local storage on cluster nodes. However, it is important to consider SSDs when planning out a Hadoop environment. SSDs can provide compelling performance benefits, likely at a lower TCO over the lifetime of the infrastructure.

Consider the Terasort performance for the three configurations tested, in terms of the total number of 1TB Terasort jobs that can be completed over three years. This is determined by using the total runtime of one job to determine the number of jobs per day ($24 \times 60 \times 60 / \text{runtime_in_seconds}$), and then over three years (number of jobs per day * 3 * 365). The total number of jobs for the three configurations over three years is shown in Table 5.

³Please refer to results shown in Figure 3.

⁴Please refer to results shown in Figure 3.

Configuration	Runtime for a single 1TB Teragen/Terasort job in seconds	# Jobs in 1 day	# Jobs in 3 years
All-HDD configuration	11589	7.46	8163.60
HDD with SSD for intermediate data	7850	11.01	12051.97
All-SSD configuration	4410.5	19.59	21450.63

Table 5.: Number of jobs over 3 years

Also consider the total price for the Test Environment, as was described earlier in this paper. The pricing includes the cost of the one (1) NameNode, six (6) DataNodes, local storage disks on the cluster nodes, and networking equipment (Ethernet switches and the 10GbE NICs on the cluster nodes). Pricing for this analysis was determined via Dell’s Online Configuration pricing tool for rack servers and Dell’s pricing for accessories, and that pricing, as of May, 2014, is shown in Table 6. below.

Configuration	Discounted Pricing from http://www.dell.com
All-HDD configuration	\$37,561
HDD with SSD for intermediate data	\$43,081
All-SSD configuration	\$46,567

Table 6.: Pricing the configurations

Now consider the cost of the environment per job when the environment is used over three years (cost of configuration / number of jobs in three years).

Configuration	Discounted Pricing from http://www.dell.com	#jobs in 3 years	\$ / job
All-HDD configuration	\$37,561	8163.60	\$4.6
HDD with SSD for intermediate data	\$43,081	12051.97	\$3.57
All-SSD configuration	\$46,567	21453.06	\$2.17

Table 7.: \$ Cost/Job

Table 7. shows the cost per job (\$/job) across the three Hadoop configurations and it shows how the SSD configurations compare with the all-HDD configuration. These results are graphically summarized in Figure 4. below.

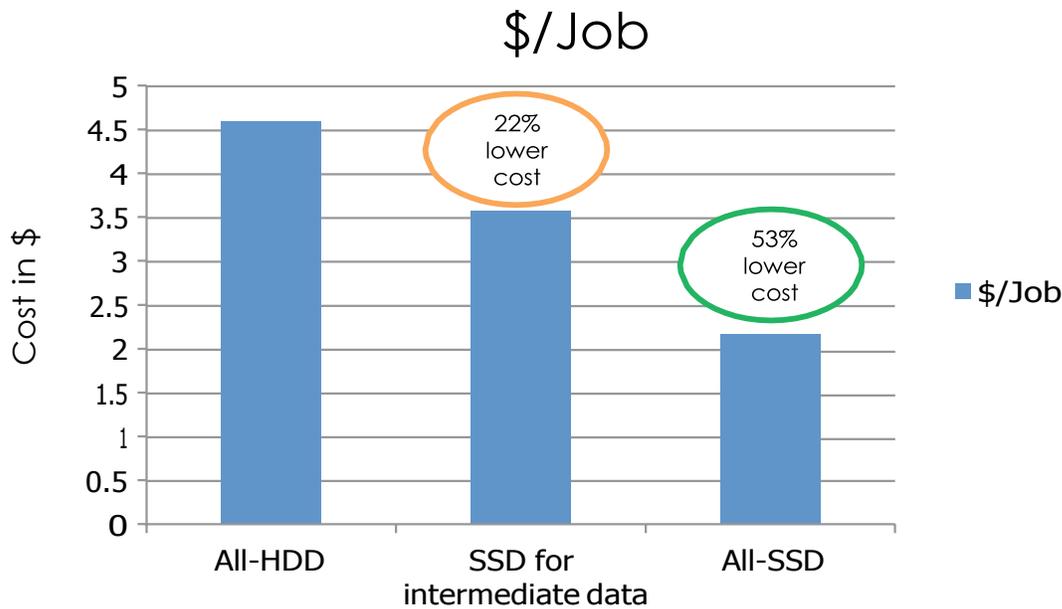


Figure 4.: Cost/job comparisons SSD vs. HDD

Observations from these analysis results:

1. Adding a single SSD to an HDD configuration and using it for intermediate data reduces the cost/job by 22%, compared to the all-HDD configuration.
2. The all-SSD configuration reduces the cost/job by 53% when compared to the all-HDD configuration.
3. Both the preceding observations indicate that, over the lifetime of the infrastructure, the TCO is significantly lower for the SSD Hadoop configurations.
4. SSDs also have a much lower power consumption profile than HDDs. As a result, the all-SSD configuration will likely see the TCO reduced even further.

17. Conclusions

SSDs can be deployed strategically in Hadoop environments to provide significant performance improvement (1.4x-2.6x for Terasort benchmark)⁵ and TCO benefits (22%-53% reduction in cost/job for Terasort benchmark)⁶ to organizations, based on SanDisk tests for Hadoop clusters supporting random access I/Os to storage devices.

These results show significant savings when flash storage is used in workloads, such as Terasort, that benefit from accelerating the performance of random I/O operations (IOPS) in association with access to storage devices. These performance gains will speed up the time-to-results for specific workloads that benefit from the use of SSDs in place of HDDs.

⁵Please refer to results shown in Figure 3.

⁶Please refer to results shown in Figure 4.

It is important to understand that Hadoop applications and workloads vary significantly in terms of their I/O access patterns, and therefore all workloads may not benefit from the use of SSDs. Some benefit more than others, depending on their pattern of access to the storage. It is therefore necessary to develop a proof of concept for custom Hadoop applications and workloads to evaluate the benefits that SSDs can provide to those applications and workloads.

18. References

1. SanDisk Corporate website: www.sandisk.com
2. Apache Hadoop: www.hadoop.apache.org
3. Cloudera Distribution of Hadoop: www.cloudera.com/content/cloudera/en/home.html
4. Dell Configuration and pricing tools
 - a. www.dell.com/us/business/p/poweredge-r320/pd
 - b. www.dell.com/us/business/p/poweredge-r720/pd

Specifications are subject to change. ©2014 - 2016 Western Digital Corporation or its affiliates. All rights reserved. SanDisk and the SanDisk logo are trademarks of Western Digital Corporation or its affiliates, registered in the U.S. and other countries. CloudSpeed Ascend is a trademark of Western Digital Corporation or its affiliates. Other brand names mentioned herein are for identification purposes only and may be the trademarks of their respective holder(s).

20160623

Western Digital Technologies, Inc. is the seller of record and licensee in the Americas of SanDisk® products.