SanDisk Solid State Drives (SSDs) for Big Data Analytics Using Hadoop and Hive

Madhura Limaye (madhura.limaye@sandisk.com)
October 2014
Table of Contents

Executive Summary .......................................................................................... 3
Key Findings of Flash-Enabled Big Data Analytics Testing. ................................ 3
Apache Hadoop ................................................................................................... 3
Data Analytics and Business Intelligence on Hadoop. ....................................... 3
Apache Hive and SanDisk SSDs ........................................................................ 4
AMPLab Benchmark for Hive ............................................................................ 4
Dataset ............................................................................................................. 5
AMPLab Data Analytics Queries ....................................................................... 5
Scan query set ................................................................................................. 5
Aggregate query set .......................................................................................... 6
Join query set ..................................................................................................... 6
Test Design ......................................................................................................... 6
Test Environment ............................................................................................... 7
Technical Component Specifications ................................................................. 7
Compute infrastructure ....................................................................................... 8
Network infrastructure ....................................................................................... 8
Storage infrastructure ......................................................................................... 8
Cloudera Hadoop configuration .......................................................................... 9
Operating System configuration ...................................................................... 10
Test Validation ................................................................................................... 10
Test Methodology .............................................................................................. 10
Results Summary .............................................................................................. 12
Results Analysis ................................................................................................. 14
Conclusions ....................................................................................................... 15
References ......................................................................................................... 15
Executive Summary

The IT industry is increasingly adopting Hadoop in big-data environments—an adoption that has increased dramatically in recent years. Many in the IT industry are now transitioning to Hadoop infrastructures for their business intelligence (BI) and data analytics requirements. We at SanDisk wanted to explore how our flash products (enterprise-grade solid state drives) can contribute towards achieving more performance for big-data business analytics. Towards that end, we conducted a number of tests on a Hadoop cluster with data analytics capabilities and SanDisk SSDs. This document describes our key findings as they relate to using SanDisk SSDs within a big-data analytics environment.

Key Findings of Flash-Enabled Big Data Analytics Testing

SanDisk tested a Hadoop cluster using the Cloudera® Distribution of Hadoop (CDH). This cluster consisted of one NameNode and six DataNodes. The cluster was set up for the purpose of determining the benefits of using solid-state disks (SSDs) within a Big Data analytics environment based on Apache Hadoop and Hive™. The Apache Hive data warehouse software is designed to query and manage large datasets residing in distributed storage.

SanDisk’s CloudSpeed Ascend™ SATA SSDs were used in the Hadoop cluster. CloudSpeed Ascend drives are specifically designed to address the growing need for SSDs that are optimized for mixed-workload applications in enterprise server and cloud computing environments.

SanDisk conducted the AMPLab Big Data benchmark with Hive on different Hadoop cluster configurations. The tests revealed that SanDisk SSDs can be deployed strategically in Big Data analytics environments based on Hadoop to provide significant performance improvement—with a 9-27% reduction in query response times—of data analytics queries. This reduces the time-to-results for these data analytics queries, allowing enterprises to make important business decisions more quickly.

The results of the AMPLab big-data benchmark testing are summarized and analyzed in the Results Summary and Results Analysis sections of this document.

Apache Hadoop

Apache Hadoop is a software framework that allows for the distributed processing of large data sets across clusters of computers, using simple programming models. It is designed to scale out to several thousands of servers in a single cluster, with each server node offering local computation and storage. Rather than relying on hardware 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 that is 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 perform actual file I/O directly with DataNodes.

Hadoop MapReduce is a software framework for processing vast amounts of data (multi-terabyte data-sets) in-parallel on Hadoop clusters (thousands of nodes) based on commodity hardware—and doing so in a reliable, fault-tolerant manner. A MapReduce job splits the input data-set into independent chunks that are processed by the map tasks in a completely parallel manner. Typically, both the input and output of a MapReduce job are stored on HDFS.

Data Analytics and Business Intelligence on Hadoop

Business intelligence and data analytics involve a set of algorithms and processes that are used to inspect, filter, analyze and transform raw data into meaningful and relevant information. This data is used for business decision
making in various departments within the enterprise, such as marketing, sales, product development and customer relationship management (CRM). Business intelligence systems typically parse through large volumes of data, which are available in a data warehouse system.

A data warehouse integrates data from a variety of sources within the business organization and creates a central repository of raw data, which can be analyzed and summarized into reports and used for various business activities. Data warehouses are also called Online Analytical Processing (OLAP) systems. Unlike Online Transaction Processing (OLTP) systems, these systems are comprised of fewer but more complex queries that are operating on very large volumes of data. Both traditional OLAP and OLTP systems use the Structured Query Language (SQL) to query for information.

In the Hadoop ecosystem, data analytics and business intelligence can be provided by a variety of products, including (but not limited to):

- **Apache Hive**: Hive is a data warehouse based on Hadoop/HDFS that understands SQL-like queries. It is important to note that Hive uses a SQL-like language called HiveQL. Hive converts the HiveQL queries into corresponding map-reduce tasks to perform the necessary analytics on the data warehouse stored in HDFS. Hive does not provide OLTP (online transaction processing) support; Rather, it functions exclusively as a data warehouse for batch-oriented data analytics workloads.

  There are also several open-source initiatives to adapt Apache Hive to alternative processing engines like Apache Spark and Apache Tez, instead of using the traditional Hadoop MapReduce paradigm. These initiatives are called Shark and Stinger, respectively.

- **Apache HBase**: HBase is an open-source, distributed, versioned, NoSQL column-based (columnar) database. It stores data as a collection of key-value pairs, with each key appearing at most once within the collection. The data is stored on HDFS.

- **Cassandra and MongoDB** are other NoSQL databases that are frequently used in the Hadoop environment to provide data analytics and business intelligence support for large volumes of data.

### Apache Hive and SanDisk SSDs

Typically, Hadoop environments use commodity servers, which are small servers with SATA HDDs used as local storage on the cluster’s server nodes. However, customers can gain significant performance benefits and faster results by using SSDs strategically within a Hadoop environment.

Why is this so? Hadoop data analytics workloads have a lot of variation in terms of their storage access profiles. Some of the data analytics workloads are I/O-intensive, and will likely benefit from the higher performance and low-latency features of SanDisk enterprise grade SSDs. Workloads having significant random access to the data will also see performance improvements using SanDisk SSDs.

This technical paper investigates the benefits of introducing SanDisk SSDs in a data analytics environment based on the Hive data warehouse and Hadoop, by using the AMPLab Big Data benchmark, which focuses on testing data analytics frameworks.

### AMPLab Benchmark for Hive

AMPLab Big Data benchmark is a benchmark tool that analyzes a variety of data analytics frameworks, including data warehouse/analytics systems based on Hadoop HDFS and MapReduce/Spark/Tez (example: Hive). The benchmark measures response time for a set of relational queries, including scans, aggregations and joins.
Dataset

The AMPLab Big Data benchmark provides three different datasets of varying sizes. These datasets are used as the input to the benchmark. The datasets are hosted on an Amazon S3 storage cloud.

The datasets are a set of unstructured HTML documents (‘Documents’), and two SQL tables that contain summary information about the documents (‘Rankings’ and ‘Uservisits’). The datasets are designed by using a scale factor, which is in terms of ‘nodes’. Each ‘node’ corresponds to ~25GB of the ‘Uservisits’ table, ~1GB of the ‘Rankings’ table and ~30GB of ‘Documents’, uncompressed. So, a ‘5nodes’ dataset will correspond to (5*(25+1+30)) GB, or 280GB of data.

The datasets must be copied over to HDFS on the Hadoop cluster before the benchmark queries can be attempted.

For the purpose of testing the benchmark at SanDisk, the ‘5nodes’ dataset hosted on the Amazon S3 storage cloud was downloaded and utilized on the on-premises big-data cluster:

<table>
<thead>
<tr>
<th>S3 Suffix</th>
<th>Scale Factor</th>
<th>Rankings (bytes)</th>
<th>Uservisits (bytes)</th>
<th>Documents (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/5nodes/</td>
<td>5</td>
<td>6.38GB</td>
<td>126.8GB</td>
<td>136.9GB</td>
</tr>
</tbody>
</table>

Table 1: AMPLab ‘5nodes’ dataset specifications

AMPLab Data Analytics Queries

The AMPLab Big Data benchmark provides scripts to run various data analytics queries on the chosen dataset. These scripts have been designed to run on an Amazon Web Services (AWS)-based cluster. The relevant queries were extracted from these scripts to adapt to the SanDisk on-premises cluster for testing purposes.

The queries executed for this technical paper included the ‘scan’, ‘aggregate’ and ‘join’ query sets. Some of these basic queries belonging to the benchmark were modified slightly with respect to where the output of the queries was written. Instead of relaying the output of the queries to ‘stdout’, a temporary cached table was created, and was later deleted for each query—and the output of the query was relayed to this cached table. This was done to shorten the run time for the benchmark.

As an example, for the aggregate query set discussed below, the total response time across all the queries in the set was close to 15,000 seconds when using ‘stdout’, and it was measured at 1,143 seconds when using the cached table, for the same cluster configuration.

Note that the cached table was used for the benchmark runs across all cluster configurations, thus making the results comparable.

Including the change for using the cached table, the specific queries used were as follows:

Scan query set

```sql
create table rankings_cached as SELECT pageURL, pageRank FROM rankings WHERE pageRank > 1000;
DROP TABLE IF EXISTS rankings_cached;
create table rankings_cached as SELECT pageURL, pageRank FROM rankings WHERE pageRank > 100;
DROP TABLE IF EXISTS rankings_cached;
create table rankings_cached as SELECT pageURL, pageRank FROM rankings WHERE pageRank > 10;
DROP TABLE IF EXISTS rankings_cached;
```
Aggregate query set

Aggregation functions (GROUP BY) are commonly used in database systems to form a set/group of rows based on certain criteria to give a more significant meaning or value to the data. This query set includes aggregate functions to create a group/set of rows based on the source IP address from the ‘Uservisits’ table.

```sql
create table uservisits_cached as SELECT SUBSTR(sourceIP, 1, 8), SUM(adRevenue) FROM uservisits GROUP BY SUBSTR(sourceIP, 1, 8);
DROP TABLE IF EXISTS uservisits_cached;
create table uservisits_cached as SELECT SUBSTR(sourceIP, 1, 10), SUM(adRevenue) FROM uservisits GROUP BY SUBSTR(sourceIP, 1, 10);
DROP TABLE IF EXISTS uservisits_cached;
create table uservisits_cached as SELECT SUBSTR(sourceIP, 1, 12), SUM(adRevenue) FROM uservisits GROUP BY SUBSTR(sourceIP, 1, 12);
DROP TABLE IF EXISTS uservisits_cached;
```

Join query set

A join query is another operation in database management systems, where records from two or more tables or databases are combined to create a set that can be saved as its own table. This table can then be used for further analysis. The following join queries are using the source IP address, the advertising revenue from these source IP addresses (or user visits) from the ‘Uservisits’ table and the pagerank information from the ‘Rankings’ table.

```sql
SELECT sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank FROM rankings R JOIN (SELECT sourceIP, destURL, adRevenue FROM uservisits UV WHERE UV.visitDate > "1980-01-01" AND UV.visitDate < "1980-04-01") NUV ON (R.pageURL = NUV.destURL) GROUP BY sourceIP ORDER BY totalRevenue DESC LIMIT 1;
SELECT sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank FROM rankings R JOIN (SELECT sourceIP, destURL, adRevenue FROM uservisits UV WHERE UV.visitDate > "1980-01-01" AND UV.visitDate < "1983-01-01") NUV ON (R.pageURL = NUV.destURL) GROUP BY sourceIP ORDER BY totalRevenue DESC LIMIT 1;
SELECT sourceIP, sum(adRevenue) as totalRevenue, avg(pageRank) as pageRank FROM rankings R JOIN (SELECT sourceIP, destURL, adRevenue FROM uservisits UV WHERE UV.visitDate > "1980-01-01" AND UV.visitDate < "2010-01-01") NUV ON (R.pageURL = NUV.destURL) GROUP BY sourceIP ORDER BY totalRevenue DESC LIMIT 1;
```

All of the above queries were executed sequentially against a Hive data warehouse that was hosted on the on-premises Hadoop cluster, and the response times were collected for these queries for different storage configurations of the Hadoop cluster. Multiple runs for the benchmark queries were conducted on each configuration, and the average response times for the queries across the multiple runs were determined.

**Test Design**

A Hadoop cluster using the Cloudera Distribution of Hadoop (CDH), consisting of 1 NameNode and 6 DataNodes, was set up for the purpose of determining the benefits of using solid-state disks (SSDs) in a data-analytics environment using Hive and Hadoop. The testing consisted of using the AMPLab benchmark queries on a ‘5nodes’ dataset hosted on different cluster configurations (as described in the Test Methodology section). The response time for the queries was recorded for the different configurations. The results of these tests are summarized and analyzed in the Results Analysis section of this paper.
Test Environment

The test environment consisted of one Dell PowerEdge™ R720 server being used as a NameNode and 6 Dell PowerEdge R320 servers being used as DataNodes in a Hadoop cluster. Network configuration consisted of a 10GbE private network interconnect that connected all the servers for internode Hadoop communication and a 1GbE network interconnect for the management network. The Hadoop cluster storage was switched between HDDs and SSDs, based on the test configurations. Figure 1 below shows a pictorial view of the environment, which is followed by the hardware and software components that were used within the test environment.

![Hadoop Cluster Topology](image)

Figure 1: Cloudera Hadoop Cluster Environment

Technical Component Specifications

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software if applicable</th>
<th>Purpose</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell PowerEdge R320</td>
<td>• RHEL 6.4</td>
<td>DataNodes</td>
<td>6</td>
</tr>
<tr>
<td>• 1 x Intel Xeon E5-2430 2.2 GHz, 6 core CPU, with hyperthread feature “ON”</td>
<td>• Cloudera Distribution of Hadoop 4.6.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 16GB memory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• HDD Boot drive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dell PowerEdge R720</td>
<td>• RHEL 6.4</td>
<td>NameNode, Secondary NameNode</td>
<td>1</td>
</tr>
<tr>
<td>• 2x Intel Xeon E5-2620 2Ghz 6 core CPUs, with hyperthread feature “ON”</td>
<td>• Cloudera Distribution of Hadoop 4.6.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cloudera Manager 4.8.1</td>
<td></td>
</tr>
<tr>
<td>Dell PowerConnect 2824 24-port switch</td>
<td>1 GbE network switch</td>
<td>Management network</td>
<td>1</td>
</tr>
<tr>
<td>Dell PowerConnect 8132F 24-port switch</td>
<td>10 GbE network switch</td>
<td>Hadoop data network</td>
<td>1</td>
</tr>
<tr>
<td>500GB 7.2K RPM Dell SATA HDDs</td>
<td>Used as Just a Bunch of Disks (JBODs)</td>
<td>DataNode drives</td>
<td>12</td>
</tr>
<tr>
<td>480GB CloudSpeed Ascend SATA SSDs</td>
<td>Used as Just a Bunch of Disks (JBODs)</td>
<td>DataNode drives</td>
<td>12</td>
</tr>
<tr>
<td>Dell 300GB 15K RPM SAS HDDs</td>
<td>Used as a single RAID 5 (5+) group</td>
<td>NameNode drives</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 2: Hardware components
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) | 4.6.0 | Cloudera’s Hadoop distribution

Table 3: Software components

**Compute infrastructure**

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

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

**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 NIC. The 1GbE management network was linked via a Dell PowerConnect 2824 24-port 1GbE switch. The 10GbE cluster network was linked via a Dell PowerConnect 8132F 10GbE switch.

**Storage infrastructure**

The NameNode used six 300GB 15K RPM SAS HDDs in a RAID5 configuration for the Hadoop file system. RAID5 was used on the NameNode to protect the HDFS metadata stored on the NameNode in case of disk failure or corruption. This NameNode setup was used across all of 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 depending on the configuration being tested. The specific configurations are discussed in detail in the Test Methodology section:

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 (SATA) SSD

In each of the above environments, the disks were used in a JBOD configuration. HDFS maintains its own replication and striping for the HDFS file data. That is why the JBOD configuration is recommended for the data nodes, resulting in better performance and better failure recovery.
Cloudera Hadoop configuration

The Cloudera Hadoop configuration consisted of the HDFS (Hadoop Distributed file system) configuration and the MapReduce configuration. The specific configuration parameters that were changed from their defaults are listed in Table 4. All remaining parameters were left at the default values, including the replication factor for HDFS (3).

<table>
<thead>
<tr>
<th>Configuration parameter</th>
<th>Value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfs.namenode.name.dir (NameNode)</td>
<td>/data1/dfs/nn, /data1 is mounted on the RAID5 logical volume on the NameNode</td>
<td>Determines where on the local file system the NameNode should store the name table (fsimage).</td>
</tr>
<tr>
<td>dfs.datanode.data.dir (DataNodes)</td>
<td>/data1/dfs/dn, /data2/dfs/dn. Note: /data1 and /data2 are mounted on HDDs or SSDs depending on which storage configuration is used</td>
<td>Comma-delimited list of directories on the local file system where the DataNode stores HDFS block data.</td>
</tr>
<tr>
<td>mapred.job.reuse.jvm.num.tasks</td>
<td>-1</td>
<td>Number of tasks to run per JVM. If set to -1, there is no limit.</td>
</tr>
<tr>
<td>mapred.output.compress</td>
<td>Disabled</td>
<td>Compress the output of MapReduce jobs.</td>
</tr>
<tr>
<td>MapReduce Child Java Maximum Heap Size</td>
<td>512 MB</td>
<td>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.</td>
</tr>
<tr>
<td>mapred.tasktracker.map.tasks.maximum</td>
<td>12</td>
<td>The maximum number of map tasks that a TaskTracker can run simultaneously.</td>
</tr>
<tr>
<td>mapred.tasktracker.reduce.tasks.maximum</td>
<td>6</td>
<td>The maximum number of reduce tasks that a TaskTracker can run simultaneously.</td>
</tr>
<tr>
<td>mapred.local.dir (job tracker)</td>
<td>/data1/mapred/jt, /data1 is mounted on the RAID5 logical volume on the NameNode.</td>
<td>Directory on the local file system where the JobTracker stores job configuration data.</td>
</tr>
<tr>
<td>mapred.local.dir (task tracker)</td>
<td>/data1/mapred/local OR /ssd1/mapred/local, depending on which storage configuration is used in the cluster.</td>
<td>List of directories on the local file system where a TaskTracker stores intermediate data files.</td>
</tr>
</tbody>
</table>

Table 4: Cloudera Hadoop configuration parameters
Operating System configuration

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

1. As per Cloudera recommendations for best practices, the swapping factor on the operating system was changed to 20 from the default of 60 to avoid unnecessary swapping on the Hadoop DataNodes. The parameter /etc/sysctl.conf was also updated with this value.

   # 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. With the ‘noatime’ option, the file access times are not written back, thus improving performance. 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

3. The open files limit was changed from 1024 to 16384. 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

Test Validation

Test Methodology

The purpose 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 AMPLab Big Data benchmark. The three configurations are described in detail in this section of the paper. Please note that there is no change to the NameNode configuration, and that it remains the same across all configurations.

1. All HDD configuration

   The Hadoop DataNodes use HDDs for the Hadoop distributed file system as well as Hadoop MapReduce.

   a. Each DataNode has two HDDs set up as JBODs. The devices are partitioned and formatted as ext4 file systems. These are 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 directories (mapred.local.dir).

2. HDD with SSD for intermediate data

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

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

   b. In addition to the HDDs being used on the DataNodes, there is also a single SSD on each DataNode, which is partitioned with a 4K-aligned boundary via fdisk (shown below) and then formatted to have ext4 file system.
Accelerate Oracle Backup Using SanDisk Solid State Drives (SSDs)

[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.

The SSD is then mounted via /etc/fstab to /ssd1 with the noatime option and is 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 accesses.

SSDs show their maximum benefit in Hadoop configurations with this kind of random I/O-intensive workload, and therefore this configuration is relevant to this testing. In contrast, HDDs show slower performance for random-access I/Os than for sequential I/Os, given the time it takes their mechanical drives to access each data point.

3. All SSD configuration

In this configuration, the HDDs of the first configuration are replaced with SSDs.

a. Each DataNode has two SSDs set up as JBODs. The devices are partitioned with a single partition with a 4K divisible boundary, as described in the text section (above) for the previous configuration.
b. These disks are then formatted as ext4 file systems using the 'make filesystem command, 'mkfs'.
c. They are then mounted via /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, the AMPLab big-data benchmark ‘scan’, ‘aggregate’ and ‘join’ Hive query sets are executed, and the response time for the queries is recorded for multiple runs. The average response time for each query in the three query sets is calculated across the multiple runs, for all the cluster configurations. The average response time results are shown in the next section.
Results Summary

AMPLab big-data benchmark runs were conducted on the three (3) configurations described in the Test Methodology section. The response time for completing each of the queries in the ‘scan’, ‘aggregate’ and ‘join’ Hive query sets on a ‘5nodes’ dataset was collected, for multiple runs. The average query response times across the multiple runs for the query sets are summarized in three figures: Figure 2, Figure 3 and Figure 4.

The X-axis on the graph shows specific queries and the Y-axis shows the response time in seconds. The runtimes are shown for the all-HDD configuration via the blue columns, the runtimes for HDD-with-SSD for intermediate data are shown via the red columns, and the runtimes for the all-SSD configurations are shown via the green columns.

All the graphs show the response times in seconds and lower values for the response time show better performance.

Figure 2 shows the average response time for the three queries in the ‘scan’ query set.

![Figure 2: Scan queries: average response times](image)

Figure 3 shows the average response time for the queries in the ‘aggregate’ query set across the three (3) different cluster configurations.

![Figure 3: Aggregate queries: average response time](image)
Figure 4 shows the average response times for the queries that made up the ‘join’ query set across the different cluster configurations listed in the Test Methodology section.

The results shown above in graphical format are also shown in tabular format in Table 5:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>‘Scan’ Hive Query Average Response Time (Seconds)</th>
<th>‘Aggregate’ Hive Query Average Response Time (Seconds)</th>
<th>‘Join’ Hive Query Average Response Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-HDD configuration</td>
<td>Q1: 43.11 Q2: 38.98 Q3: 29.72</td>
<td>Q1: 360.16 Q2: 378.86 Q3: 404.78</td>
<td>Q1: 412.53 Q2: 539.52 Q3: 1533.59</td>
</tr>
<tr>
<td>SSD for intermediate data</td>
<td>Q1: 44.02 Q2: 39.66 Q3: 27.60</td>
<td>Q1: 330.11 Q2: 339.58 Q3: 367.56</td>
<td>Q1: 376.72 Q2: 489.08 Q3: 1174.52</td>
</tr>
</tbody>
</table>

Table 5: Results Summary

To summarize, Table 6 and the graph in Figure 5 below show the total response time for all the queries within the ‘scan’, ‘aggregate’, and ‘join’ sets across the three different cluster configurations.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Total Response Time For ‘Scan’ Hive Query Set (Seconds)</th>
<th>Total Response Time For ‘Aggregate’ Hive Query Set (Seconds)</th>
<th>Total Response Time For ‘Join’ Hive Query Set (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All HDD configuration</td>
<td>111.826</td>
<td>1143.807</td>
<td>2485.64</td>
</tr>
<tr>
<td>SSD for intermediate data</td>
<td>111.2967</td>
<td>1037.263</td>
<td>2040.337</td>
</tr>
<tr>
<td>All SSD configuration</td>
<td>1071.867</td>
<td>943.05</td>
<td>1792.573</td>
</tr>
</tbody>
</table>

Table 6: Total response times for query sets
Results Analysis

Observations from the response time results summarized in the Results Summary section are as follows:

1. The hybrid HDD-SSD configuration, where a single SSD per DataNode is used for the MapReduce intermediate data, reduces the total 'aggregate' Hive queries response time by 9% and the total 'join' Hive query response time by 17%, compared to an all-HDD configuration.

2. Replacing all the HDDs on the DataNodes with SSDs can reduce the total 'aggregate' Hive query response time by 17% and the total 'join' Hive query response time by 27%, compared to an all-HDD configuration.

3. SSDs with the Hadoop data analytics infrastructure directly translate into faster completion of Hive queries on the Hadoop cluster.

4. Faster query processing within the data analytics and business intelligence framework of a SSD-enabled Hadoop cluster can help businesses make important business decisions faster using the faster time-to-results for the Hive queries. This helps businesses to improve their processes and make business decision-making much more efficient.

5. There was no significant difference in the Scan query response times between the HDD and the SSD configurations because the I/O utilization during these queries is low for the majority of the run, barring a few of intermittent peaks. As a result, the 'scan' query set is not storage-intensive, and therefore does not benefit from the use of SSDs, unlike the 'aggregate' and 'join' query sets.
Conclusions

The benefits that SanDisk SSDs can provide are summarized as follows:

SanDisk SSDs can be deployed strategically in big-data analytics environments based on Hadoop to provide faster query response times (9-27% faster) for specific Hive data analytics queries. Faster results from data analytics queries directly translate into faster business decisions, thus benefiting organizations in today’s fast-paced business environment. This technical paper discussed a deployment of CloudSpeed SATA SSDs with Hadoop and Hive. This deployment provided a proof-point for the performance benefits of SSDs within a Hadoop-based data analytics/business intelligence computing environment.

It is important to understand that Hadoop applications and workloads vary significantly in terms of their I/O access patterns, and therefore that all workloads may not benefit from the use of SSDs. 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.

References

SanDisk Corporate website: www.sandisk.com

Data analytics and business intelligence:

Data warehouse: http://en.wikipedia.org/wiki/Data_warehouse

Apache Hadoop: http://hadoop.apache.org/

Apache Hive: http://hive.apache.org/

Apache HBase: http://hbase.apache.org/

Cloudera Distribution of Hadoop: www.cloudera.com/content/cloudera/en/home.html

AMPLAB benchmark: https://amplab.cs.berkeley.edu/benchmark/
Products, samples and prototypes are subject to update and change for technological and manufacturing purposes.

SanDisk Corporation general policy does not recommend the use of its products in life support applications wherein a failure or malfunction of the product may directly threaten life or injury. Without limitation to the foregoing, SanDisk shall not be liable for any loss, injury or damage caused by use of its products in any of the following applications:

- Special applications such as military related equipment, nuclear reactor control, and aerospace
- Control devices for automotive vehicles, train, ship and traffic equipment
- Safety system for disaster prevention and crime prevention
- Medical-related equipment including medical measurement device

Accordingly, in any use of SanDisk products in life support systems or other applications where failure could cause damage, injury or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support or other such applications assumes all risk of such use and agrees to indemnify, defend and hold harmless SanDisk Corporation and its affiliates against all damages.

Security safeguards, by their nature, are capable of circumvention. SanDisk cannot, and does not, guarantee that data will not be accessed by unauthorized persons, and SanDisk disclaims any warranties to that effect to the fullest extent permitted by law.

This document and related material is for information use only and is subject to change without prior notice. SanDisk Corporation assumes no responsibility for any errors that may appear in this document or related material, nor for any damages or claims resulting from the furnishing, performance or use of this document or related material. SanDisk Corporation explicitly disclaims any express and implied warranties and indemnities of any kind that may or could be associated with this document and related material, and any user of this document or related material agrees to such disclaimer as a precondition to receipt and usage hereof. EACH USER OF THIS DOCUMENT EXPRESSLY WAIVES ALL GUARANTIES AND WARRANTIES OF ANY KIND ASSOCIATED WITH THIS DOCUMENT AND/OR RELATED MATERIALS, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT, TOGETHER WITH ANY LIABILITY OF SANDISK CORPORATION AND ITS AFFILIATES UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER LEGAL OR EQUITABLE THEORY FOR LOSS OF USE, REVENUE, OR PROFIT OR OTHER INCIDENTAL, PUNITIVE, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION PHYSICAL INJURY OR DEATH, PROPERTY DAMAGE, LOST DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrievable manner or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written consent of an officer of SanDisk Corporation.

All parts of the SanDisk documentation are protected by copyright law and all rights are reserved.

For more information, please visit: www.sandisk.com/enterprise

At SanDisk, we’re expanding the possibilities of data storage. For more than 25 years, SanDisk’s ideas have helped transform the industry, delivering next generation storage solutions for consumers and businesses around the globe.

951 SanDisk Drive | Milpitas | CA 95035 | USA

© 2014 SanDisk Corporation. All rights reserved. SanDisk is a trademark of SanDisk Corporation, registered in the United States and other countries. CloudSpeed and CloudSpeed Ascendar are trademarks of SanDisk Enterprise IP LLC. Other brand names mentioned herein are for identification purposes only and may be the trademarks of their holder(s). 10/15/14