International Journal of Modern Communication Technologies & Research (IJMCTR)
ISSN: 2321-0850, Volume-2, Issue-5, May 2014

How Much Solid State Drive Can Improve the Performance of Hadoop Cluster? Performance evaluation of Hadoop on SSD and HDD

Piyush Saxena, Dr. Jerry Chou

***Abstract*—Hadoop and Map reduce today are facing huge amounts of data and are moving towards ubiquitous for big data storage and processing. This has made it an essential feature to evaluate and characterize the Hadoop file system and its deployment through extensive benchmarking.**

We have other benchmarking tools widely available with us today that are capable of analyzing the performance of the hadoop system but they are made to either run in a single node system or are created for assessing the storage device that is attached and its basic characteristics as top speed and other hardware related details or manufacturer's details.

For this, the tool used is HiBench that is an essential part of Hadoop and is comprehensive benchmark suit that consist of a complete set of Hadoop programs containing micro benchmarks and real world applications for the purpose of benchmarking the performance of Hadoop on the available type of storage device (i.e. HDD and SSD) and machine configuration. This is helpful to optimize the performance and improve the support towards the limitations of Hadoop system.

In this paper we will also present that external sorting algorithm in Hadoop (MapReduce) with SSD can outperform the algorithm run with hard disk. In addition, we also demonstrate that the power consumption can be drastically reduced when SSDs are used.

***Index Terms*— Hadoop, HDFS, SSD, HDD, HiBench, Benchmarking.**

I. Introduction to Hadoop and HDFS

In today's digital generation, a huge amount of data is been processed on the internet. Providing optimal data processing with good response time improvises the output to the requests by the client. There are many users that try to access the same data over the web and it is a challenging task for the server to deliver optimal result. The large amount of data the internet has to deal with every day has made traditional solutions extremely expensive. There are problems like processing large documents split into several independent sub-tasks, that are distributed with the available nodes, and processed in parallel. Due to this, MapReduce and Hadoop came into existence.

Hadoop is a free-of-cost, programming framework that is java-based and supports the processing of large amounts of data in a distributed environment.[1] It is developed by Apache. Hadoop has the capacity to run applications on systems that have thousands of nodes and involves multiple pentabytes of data. The Hadoop Distributed File System helps faster data transfer rates between the nodes and makes the cluster to continue performing operations uninterrupted in case of node failure. This system actually lowers the risk of complete system failure even when a significant no. of nodes are in-operative.[2]

Hadoop was motivated by MapReduce (Fig.1) that was introduced by Google, a software framework in which an application is broken down into numerous small parts. Any of these parts (also called fragments or blocks) can be run on any node in the cluster.[3]

MapReduce-based studies have been actively carried out for the efficient processing of big data on hadoop. Hadoop runs on clusters of computers that can handle large amounts of data and support distributed applications.[4] In the last few years, lots of research has been carried out to improve the performance of hadoop. One of the hindrances is the performance issues of the storage device used as it is connected to the system by a slower connecting interface like Bus. Even the difference in the Devices used for storage creates the hindrance.[5] The performance of the Hadoop system is also bound on the type of workload that we consider. This is why we consider HiBench as the standard model for testing Hadoop Distributed File System (HDFS). In this paper, we try to study and evaluate the performance of Hadoop Distributed File System on a Hadoop Cluster system that contains flash memory based SSD (Solid State Drive) and Hard Disk Drive by optimizing each parameter on HiBench.

II. Advantages of using Hadoop

Hadoop has got a huge force on the great Web 2.0 organizations like Google and Facebook that uses Hadoop to accumulate and supervise their enormous data sets. It has also established valuable for many other conventional enterprises. The five big advantages of Hadoop are [6]:

- Scalable

It can accumulate and allocate very big data sets transversely numerous inexpensive servers that work and compute in parallel. Dis-similar to the conventional relational data base systems (RDBMS) that can't scale to compute large amounts of data, Hadoop helps to run applications on hundreds of nodes involving penta bytes of data.

- Cost effective

Hadoop offers an economical storage for outsized amounts of data. The problem with conventional RDBMS is that it is

Manuscript received April 24, 2014

Piyush Saxena, Masters Of Technology (Computer Science and Engineering) Amity University Uttar Pradesh Noida, India

Dr. Jerry Chou, Computer Science and Engineering National Tsing Hua University Hsinchu, Taiwan

extremely costly to scale to a large degree in order to process massive volumes of data that is easily possible in the Hadoop System.

- Flexible

Hadoop provides easy access to new data sources and work with different types of data (both structured and unstructured) to create values from that data. Thus Hadoop can be use to derive valuable insights from data sources.

- Fast

Hadoop's exclusive storage technique is based on a distributed file system that essentially 'maps' data anywhere it is located on a cluster. The tools for data handing are on the same servers where the data is located, this results in much faster data processing.

- Resilient to failure

An advantage of using Hadoop is its fault tolerance. When information is shared with a single node, that info is also copied to other nodes in the cluster that means in case of failure there does another copy exist for use.

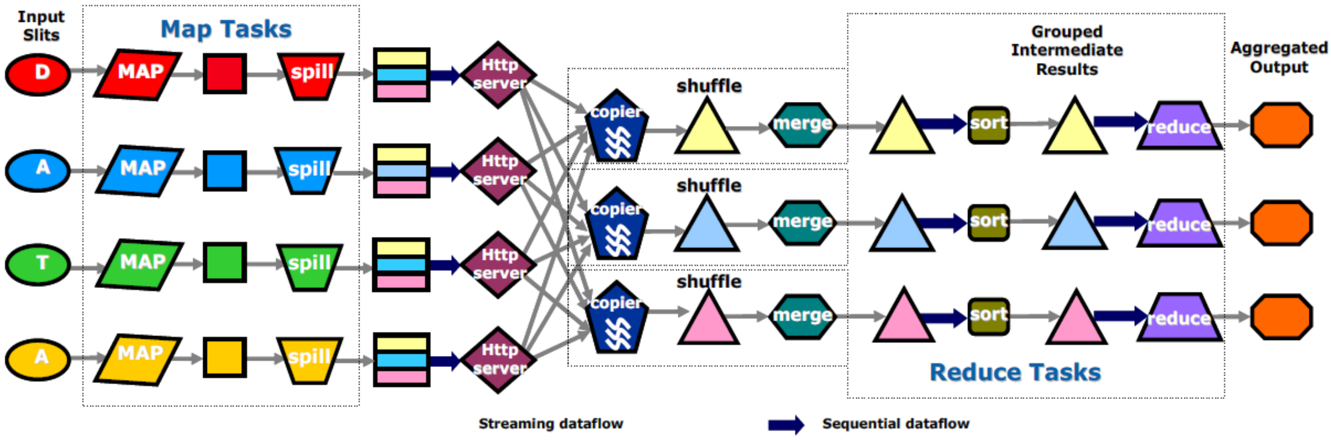


Fig 1.) Hadoop MapReduce Architecture

III. An Insight into SSD and HDD

A **solid-state drive** (**SSD**) (also known as a **solid-state disk** or **electronic disk**) (Fig.2) is a data storage device using integrated circuit assemblies as memory to store data indefatigably. SSD uses electronic components that are attuned with conventional block input/output (I/O) HDDs, thus allowing easier substitute in ordinary applications. SSDs use NAND-based flash storage memory, which has the capacity to retain data without power [7].

A **hard disk drive** (**HDD**) (Fig.3) is a storage device used for storing and retrieving digital data using rapidly rotating disks coated with magnetic material. HDD is non-volatile i.e. it retains its data even when power is switched off. Data stored is readable in a random-access manner, which means a single block of data can be stored or retrieved in any order. An HDD contains one or multiple, rigidly fixed, rotating disks with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces [7].

Solid state drives give large no. of benefits over conventional hard drives like:

1.) SSDs are More Durable: SSD is a non-mechanical design of NAND flash mounted on circuit boards, and are shock resistant. Hard Drives consist of a variety of moving parts making them vulnerable to shock and damage.

2.) SSDs are Faster: SSDs can have greater enhanced performance, immediate data access, faster boot ups, quicker file transfers, and in general superfast computing speeds than hard drives. HDDs can only access the data earlier the nearer it is from the R/W heads, whereas all sections of the SSD are accessible at the identical speeds.

3.) SSDs Consume less Power: SSDs use considerably a smaller amount of power at the highest point of load than hard drives. Their energy efficiency can make the systems cost effective and deliver longer battery life, less power strain on system, and a cooler computing environment. (Fig.4)

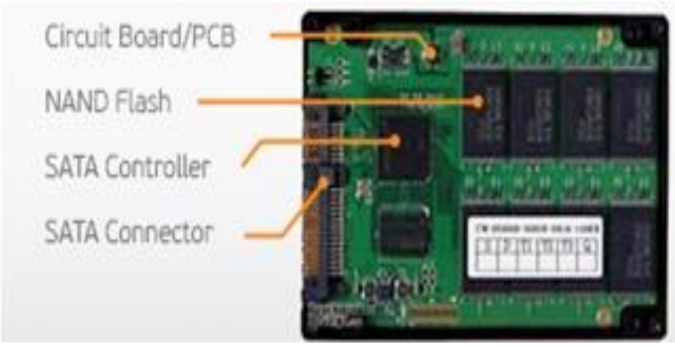


Fig 2.) NAND Flash based Solid State Drive

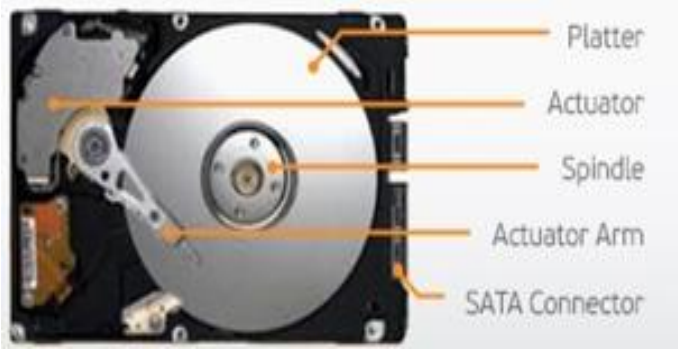


Fig 3.) Platter Based Hard Disk Drive

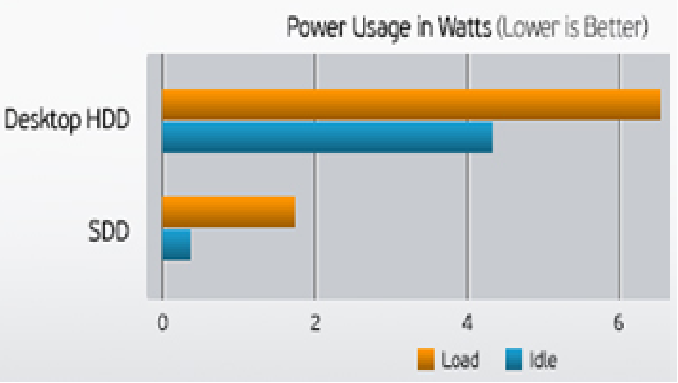


Fig 4.) Power Usage in Watts

4.) SSDs are Cooler: As an energy-efficient storage system, SSDs require very little power to operate that translates into significantly less heat output by your system.

5.) SSDs are Quieter: With no moving parts, SSDs run as silent operation and never disturb computing experiences, unlike loud, whirring hard disc drives.

IV. Alternative methods of Benchmarking Storage Devices

Disk Benchmarking is the process of running tests on a disk to determine its speed and latency. Disk benchmarking is the process of running software that accurately measures transfer speeds under various disk access scenarios (chronological, arbitrary 4K, deep line depth etc.). The endeavor is to create statistics in Mbps that recapitulate the speed individuality of a disk.

There are many alternative ways to do benchmarking of the Storage Devices and to test their latency, speed and other performance criteria.

1.) ATTO Disk Benchmarking: [8] The Atto Disk Benchmark is longer than any other disk benchmarking software. This utility was designed to measure regular disk drive performance but it is more than up to the task of measuring both USB flash drive and SSD speeds. The utility procedures disk performance rates for a range of sizes of file and displays the results in a bar chart showing read and write speeds at each file size. The results are in megabytes per second (Mbps). On comparing 100 GB of SSD (Fig.5.1) and HDD (Fig.5.2) performance, significant difference can be spotted. The Figure gives the performance graphs of the benchmarking.

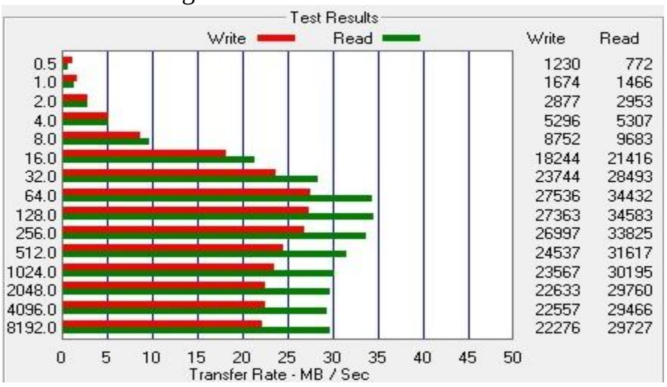


Fig 5.1.) ATTO Benchmarking of 100 Gb Solid State Drive

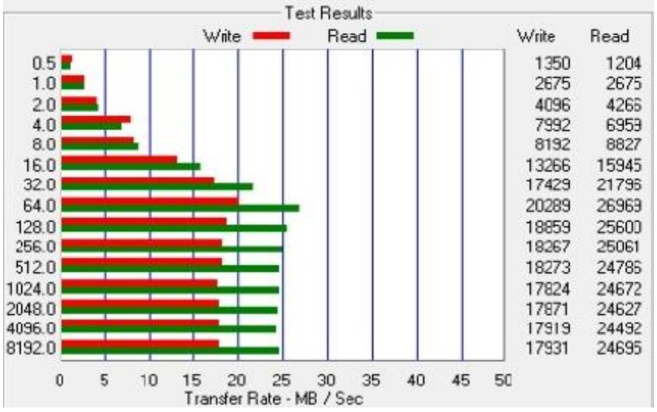


Fig 5.2.) ATTO Benchmarking of 100 Gb Hard Disk Drive

ATTO (Bench32.exe) options and features

a. Direct I/O
b. Force Write Access
c. I/O Comparison
d. Overlapped I/O
e. Queue Depth
f. Run Continuously
g. Test Pattern
h. Transfer Size

2.) HD Tune Pro: [9] HD Tune Pro is a hard disk / SSD utility with many options. It can be used to calculate the storage device's performance, examine for errors, check the health status (S.M.A.R.T.), securely erase all data, scans the surface for errors and Temperature display. (Fig.6)

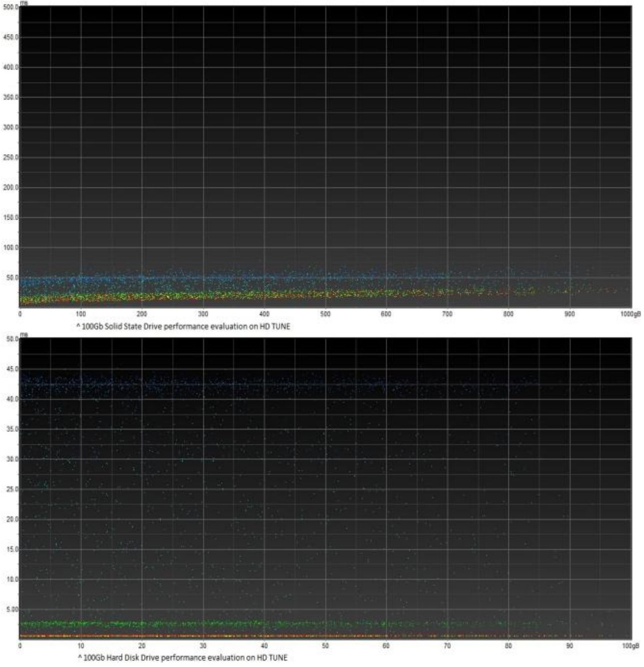
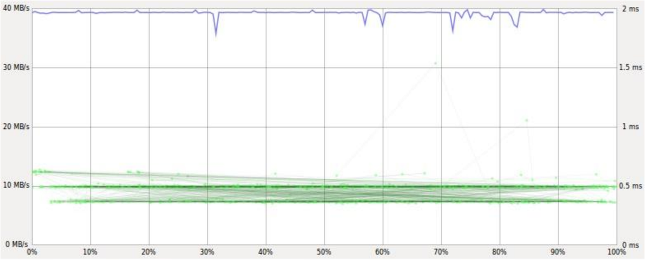


Fig 6.) HD TUNE Pro Benchmarking of SSD and HDD.

SMART is the Self-Monitoring, Analysis and Reporting Technology is a monitoring system for computer hard disk drives (HDDs) and solid-state drives (SSDs) to sense and account on various scales of trustworthiness, in the wish of anticipating failures. When a failure is spotted by

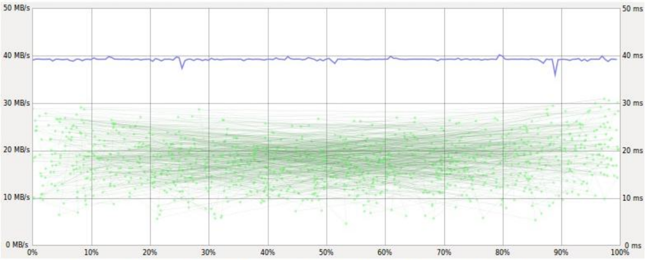
S.M.A.R.T., the user may decide to reinstate the drive to stay away from unanticipated outage and data failure. The producer may be able to use the S.M.A.R.T. data to find out where faults lie and avoid them from returning in potential drive designs.

3.) Linux Disk Utilities: [10] The disk utility is useful to find the model, sequential number, firmware, and the in general fitness evaluation of the hard disk, and to check if a SMART structure is enabled on the hard disk. It also lets user benchmark the performance of the storage device he is using SSD/HDD. (Fig.7)



^ 100Gb Solid State Drive performance evaluation on Linux Disk Utilities Benchmarking System

Fig 7.1.) Linux Disk Utilities Benchmarking of SSD



^ 100Gb Hard Disk Drive performance evaluation on Linux Disk Utilities Benchmarking System

Fig 7.2.) Linux Disk Utilities Benchmarking of HDD

V. HiBench (Hadoop Benchmarking Suit)

MapReduce and its popular open source application framework, Hadoop, are going toward ever-present for Big Data storage and computing, thus it is mandatory to quantitatively calculate and illustrate the Hadoop operations through extensive benchmarking. Two basic features of HiBench are:

1.) Categorization
  a. recognize the distinctive conduct of real-world apps.
  b. recognize the Hadoop structure and data flow model

2.) Assessment on diverse server platforms
  a. compute and contrast the performance of particular deployments
  b. trace the bottleneck of particular deployment
  c. trace the power effectiveness of particular deployment choices

3.) Assessment on different Hadoop versions
  a. examine the performance impact of new characteristics and optimizations in newer versions

In this paper, we present *HiBench*, a representative and whole benchmark suite for Hadoop, which contains a set of Hadoop applications including both synthetic micro-benchmarks and real-world applications. [11] The benchmark suite has ten workloads and are classified into 4 sub-divisions.

I. Classification Of Workloads [12]

1.) Micro benchmarks
2.) Web Search
3.) Machine Learning
4.) Analytical Query

1.) Micro Benchmarks [13]

Contains 4 Workloads:

1.) Sort: It is a representation of a large subset of real world MapReduce jobs that is transforming data from one representation to another. Sort requires an Input Output bound system resource utilization with the data access patterns as equal quantities of data access. The input data is generated using the *RandomTextWriter* program contained in the Hadoop distribution. Time taken by Reduce stage is twice the time taken by Map stage. (Fig.8.1)

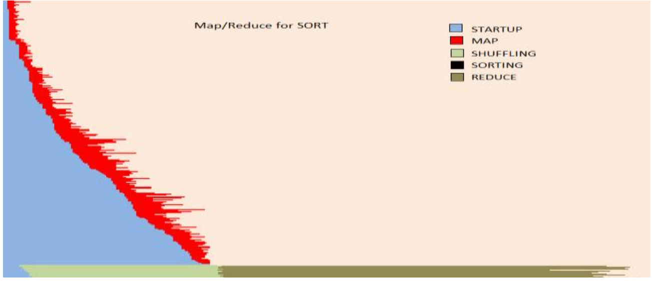


Fig 8.1) MapReduce for SORT workload

2.) Word Count: It is also a representation of a large subset of real world MapReduce jobs that is transforming data by extracting a small amount of interesting data from a large data set. Word Count requires a CPU bound system resource utilization with the data access patterns as reducing quantities of data access. The input data is generated using the *RandomTextWriter* program contained in the Hadoop distribution. Time taken by Reduce stage is nearly the same as the time taken by Map stage. (Fig.8.2)

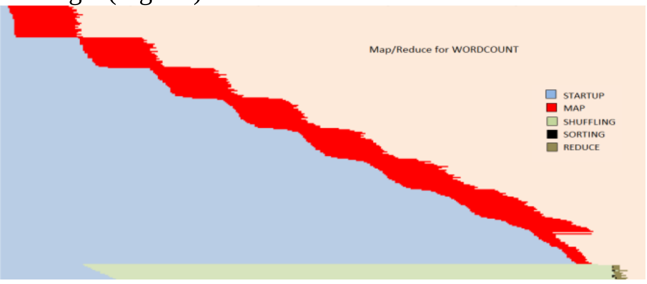


Fig 8.2) MapReduce for WORD COUNT workload

3.) TeraSort: It sorts 10 billion 100-byte records generated by the *TeraGen* program contained in the Hadoop distribution. TeraSort requires CPU bound system resource utilization during Map stage and Input Output bound system resource utilization during Reduce stage with the data access patterns as reducing and then growing quantities of data access. Time taken by Reduce stage is 1.5 times the time taken by Map stage. (Fig.8.3)

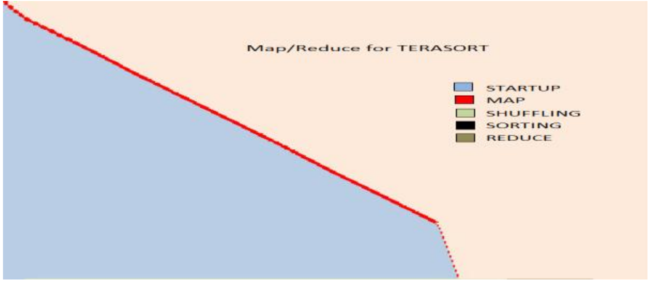


Fig 8.3) MapReduce for TERA SORT workload

4.) Enhanced DFSIO: Enhanced Distributed File System Input Output is used to evaluate the aggregated bandwidth delivered by HDFS. It computes the aggregated bandwidth by sampling the number of bytes read/written at fixed time intervals in each map task. During the reduce and post-dispensation stage, the sample of the map task are linear interposition and re-evaluation at a fixed plot samples, so as to compute the aggregated read/write throughput by all the map tasks. Enhanced DFSIO requires an Input Output bound system resource utilization with trivial data access patterns

2.) Web Searching [13]

Contains 2 Workloads:

1.) Nutch Indexing: It is the representation of one of the most important and significant use of MapReduce that is large scale search indexing systems. The Nutch Indexing workload is the indexing sub-system of Nutch, a popular open-source (Apache) search engine. Nutch requires Input Output bound system resource utilization, but shows high CPU Utilization in Map state with the compressed data that is accessed by Map state and decompressed into data and then even reduced to even fewer data. Time taken by Reduce stage is twice the time taken by Map stage.

2.) Page Ranking: It is an open source implementation of the page-rank algorithm in Mahout that is an open-source machine learning library. It is an open source implementation of the page-rank algorithm, a link analysis algorithm used in web search engines. Page Rank requires CPU bound system resource utilization with the data access patterns as reducing quantities of data access. Time taken by Reduce stage is 1.5 times the time taken by Map stage.

3.) Machine Learning [13]

Contains 2 Workloads:

1.) Bayesian Classification: It is the representation of one of the most important and significant use of MapReduce that is large scale machine learning. The workload implements the trainer part of Naive Bayesian (a popular classification algorithm for knowledge discovery and data mining). Bayesian Classification requires Input Output bound system resource utilization with high CPU utilization in map stage of the first job. The data access patterns shows growing and then reducing quantities of data access. Time taken by Reduce stage is 1.5 to 2 times the time taken by Map stage.

2.) K-means Clustering: It implements K-means that is a well-known clustering algorithm for knowledge discovery and data mining. Its input is a set of readings, and every reading is representing a numerical d-dimensional vector. K-means clustering requires CPU bound in iteration and Input Output Bound in clustering. The data access patterns shows reducing quantities of data access. Time taken by Reduce stage is nearly same as the time taken by Map stage.

4.) Analytical Query [13]

Contains 2 Workloads:

1.) Hive Join: It represents one of the most significant uses of MapReduce (i.e., OLAP-style analytical queries). They are intended to model complex analytic queries over structured (relational) tables - Hive Join computes the both the average and sum for each group by joining two different tables.

2.) Hive Aggregation: It represents one of the most significant uses of MapReduce (i.e., OLAP-style analytical queries). They are intended to model complex analytic queries over structured (relational) tables - Hive Aggregation computes the sum of each group over a single read-only table.

II. Performance evaluation of SSD and HDD

For the Performance evaluation and Analysis of the performance of SSD and HDD the considered work loads are Sort, Word Count and Tera Sort. The size of data taken for all the workloads is 6550021992 bytes that is 6.1001GB of data. [14][15][16]

Sort Work Load: Since Sort has an Input Output bound resource utilization it is easily observed that SSD (Fig.9.1) buffers the data much earlier and at a faster rate than HDD (Fig.9.2) that tends to buffer at a constant speed. Due to this reason the SSD had an earlier chance to start off with the Reduce phase as compared to the HDD. It can also be inferenced from the graduated behavior of the graph that HDD works in a much stabilized manner as compared to the SDD. Over all SSD finishes off its job with the processors 39seconds earlier than the HDD. This proves that the SSD works much faster than HDD in the scenario of Sort Workload.

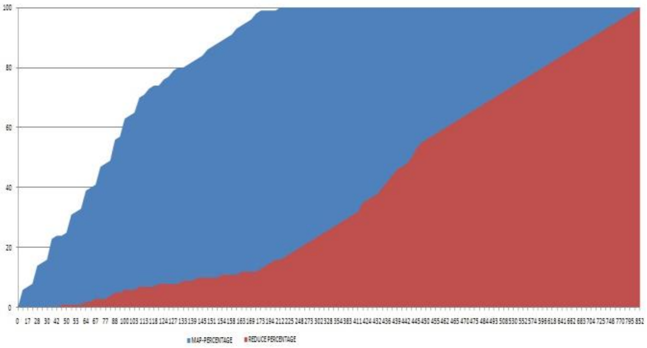


Fig 9.1.) SORT workload on Solid State Drive

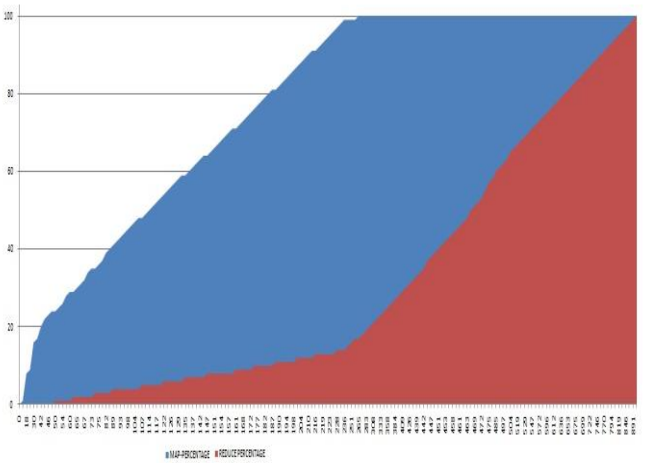


Fig 9.2) SORT workload on Hard Disk Drive

Word Count Work Load: Since Sort has a CPU bound resource utilization it is easily observed that SSD (Fig.10.1) and HDD (Fig.10.2) both buffers approximately at the same rate but with a little variation in the speed as SSD buffers about 3 seconds faster than HDD. Due to this reason the SSD had an earlier chance to start off with the Reduce phase at 47 seconds as compared to the HDD that starts at 49 seconds. It can also be inferred from the abrupt behavior of the graph that HDD takes a longer time in the reduce phase as compared to the SSd that takes less time. Over all SSD finishes off its job with the processors 6seconds earlier than the HDD that is not a very major time difference. But, still this proves that the SSD works faster than HDD in the scenario of WordCount Workload.

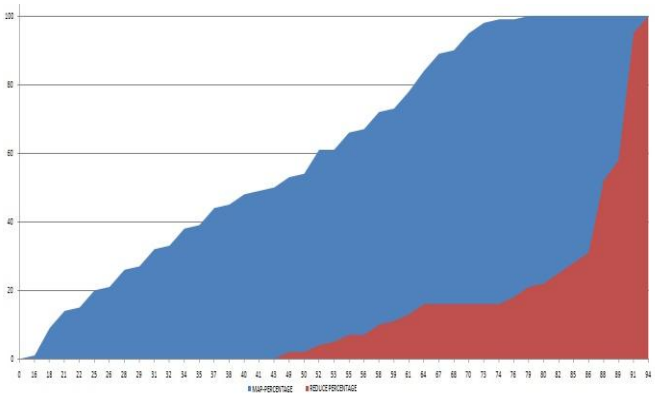


Fig 10.1) WORD COUNT workload on Solid State Device

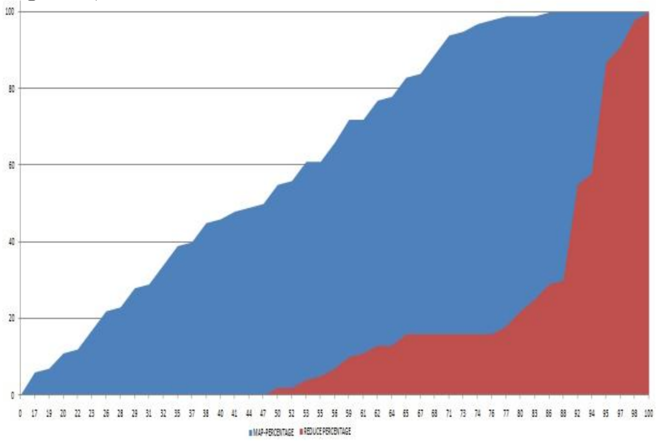
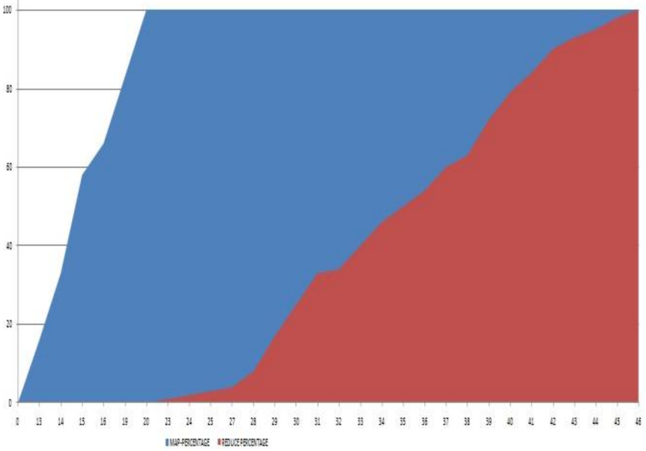
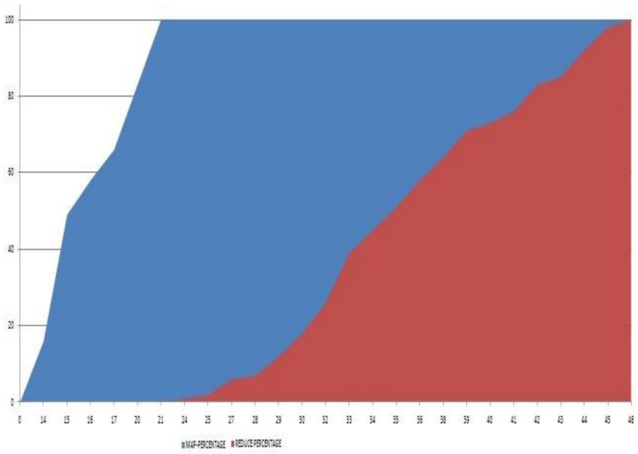


Fig 10.2) WORD COUNT workload on Hard Disk Drive

1.) Tera Sort Work Load: Since Sort has a CPU bound system resource utilization during Map stage and Input Output bound system resource utilization during Reduce stage it is easily observed that SSD (Fig.11.1) buffers the data much earlier (19.5 Sec) and at a faster rate than HDD (Fig.11.2) (21.5 Sec) that tends to buffer at an abrupt speed. Due to this reason the SSD had an earlier chance to start off with the Reduce phase at 23 second as compared to the HDD that starts at 24 second. It can also be observed that the reduce phase for SSD and HDD takes equal amount of time which means it has got no relation with the working of SSD or HDD and are totally dependent on processor. Over all SSD finishes off its job with the processors 1second earlier than the HDD that is not a negligible difference. But, still this proves that the SSD has lower latency than HDD in the scenario of TeraSort Workload



2.) Fig 11.1) TERASORT workload on Solid State Drive



3.) Fig 11.2) TERASORT workload on Hard Disk Drive.

III. CONCLUSION AND FUTURE SCOPE

From the above results and analysis the performance of SSD and HDD is nearly the same, but positive results can be seen for better performance of SSD than HDD. Also the difference in the performance is very small. So, an observation that can be monitored is that the Map phase in any of the workload is performing well until the random access memory is not consumed. Once the memory is falls short, it behaves the same as it needs to page the data in process to the nearest memory location as there is no excess shared memory available. This concludes that there is a need to involve a Distributed Shared Memory (DSM) to improve the performance of the SSD and HDD and get better significant results [17]. This can be easily implemented with the help of open source code ‘MemCache’ [18]. Also the performance of the SSD and HDD is also hindered due to the use of common Bus system to connect to the processing units. If another connection technique like InfiniBand is used to connect then better performance in terms of latency, speed of access and fault tolerance can be achieved [19].

In the future, a model to have DSM as a part should be used to be implemented with the use of InfiniBand that supports the use of Verbs and with the technology of Optical Fibers to achieve faster performances and Remote Dynamic Memory Access. [20][21][22]

IV. References

[1] Hadoop Home: http://hadoop.apache.org/

[2] Jacky Wu, "Hadoop HDFS & MapReduce", *Help Guidelines LSA Lab, NTHU, Taiwan* 2013.8.7.

[3] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", *SOSP’03*, October 19–22, 2003, *Bolton Landing, New York, USA. Copyright 2003 ACM.*

[4] Piyush Saxena, Satyajit Padhy, Praveen Kumar, “Optimizing Parallel Data Processing With Dynamic Resource Allocation”, *International Conference on Reliability, Infocom Technologies and Optimization*.,pp. 735-739, Jan. 29-31, 2013.

[5] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", *OSDI’04,* 2004, *Bolton Landing, New York, USA. Copyright 2004 ACM.*

[6] Piyush Saxena, Satyajit Padhy, Praveen Kumar, “Eliminating Homogeneous Cluster Setup for Efficient Parallel Data Processing”, *International Journal Of Computer Applications*., vol. 64, issue 17, February 2013.

[7] Seok-Hoon Kang, Dong-Hyun Koo, Woon-Hak Kang and Sang-Won Lee, “A Case for Flash Memory SSD in Hadoop Applications”, *International Journal of Control and Automation, Vol. 6, No. 1,* February, 2013.

[8] ATTO Benchmarking: http://www.attotech.com/disk-benchmark/

[9] HD Tune Benchmarking Home: http://www.hdtune.com/

[10] Linux Disk Utility Benchmarking: http://community.linuxmint.com/software/view/gnome-disk-utility

[11] HiBench Home: https://github.com/intel-hadoop/HiBench

[12] Shengsheng Huang, Jie Huang, Yan Liu, Lan Yi and Jinquan Dai, "HiBench: A Representative and Comprehensive Hadoop Benchmark Suite", *Reference Doccument, Intel Asia-Pacific Research and Development Ltd., Shanghai, P.R.China,* 2011.

[13] Shengsheng Huang, Jie Huang, Jinquan Dai, Tao Xie, and Bo Huang, "The HiBench Benchmark Suite: Characterization of the MapReduce-Based Data Analysis", *ICDE Workshops'10,* Oct. 2010, *2010 IEEE.*

[14] Lan Yi, “Experience with HiBench: From Micro-Benchmarks toward End-to-End Pipelines”, *WBDB 2013 Workshop Presentation, Intel China Software Center,* 2013.07.16.

[15] Dominique Heger, “Hadoop Performance Tuning - A Pragmatic & Iterative Approach”, *Research details by DHTechnologies - www.dhtusa.com,* 2013.

[16] Jason Dai, “Toward Efficient Provisioning and Performance Tuning for Hadoop”, *Apache Asia Roadshow 2010, Intel China Software Center,* June 2010.

[17] Remote Direct Memory Access : http://en.wikipedia.org/wiki/Remote_direct_memory_access

[18] Jithin Jose, D. K. Panda et-al “Memcached Design On High Performance RDMA Capable Interconnects”, *Lab Resources of Network---Based Computing Laboratory, Department of Computer Science and Engineering, The Ohio State University, USA.*

[19] InfiniBand Trade Association Home: http://www.infinibandta.org/

[20] Liang Ming , Dan Feng, Fang Wang, Qi Chen, Yang Li, Yong Wan, Jun Zhou, "A Performance Enhanced User-space Remote Procedure Call on InfiniBand*", *Photonics and Optolectronics Meetings (POEM).,* 2011.

[21] Fan Liang, Chen Feng, Xiaoyi Lu, Zhiwei Xu, "Performance Benefits of DataMPI:A Case Study with BigDataBench", *ACM SOFT BPOE ’14*, Mar 1, 2014, Salt Lake City, Utah, USA.

[22] Xiaoyi Lu, Nusrat S. Islam, Md. Wasi-ur-Rahman, Jithin Jose, Hari Subramoni, Hao Wang, and Dhabaleswar K. (DK) Panda, "High-Performance Design of Hadoop RPC with RDMA over InfiniBand", *National Science Foundation grants #OCI-0926691, #OCI-1148371 and #CCF-1213084,* 2013 IEEE.

Piyush Saxena Pursuing Master of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India, Area of Interest: Cloud Computing, Data Mining and Warehousing and Soft Computing.