Available online at www.sciencedirect.com

ScienceDirect

Materials Today: Proceedings XX (2016) XXX–XXX

materialstoday: PROCEEDINGS

www.materialstoday.com/proceedings

An Enhanced and Efficient Approach for Improving the Performance of HPC Environment Using Map-Reduce With MARIANE ☆

Sathish Kumar.P.J [a], Dr.R.Jagadeesh Kannan [b]

[a]*Research Scholar, Bharath University, Chennai, Tamilnadu*
[b]*Pofessor, Dept of CSE, VIT University Chennai Campus, Chennai, Tamilnadu*

Abstract

Hadoop is the main open source tool which backs the acknowledgment of the Big data upheaval and depends on Googles MapReduce spearheading mission in the area of very much substantial measure of information stocking space and handling. As of late, functions and advancements tending to "Big Data" confrontations have turned out to be colossally critical in numerous ranges of business and logical registering. The Map-Reduce model has empowered an expansive arrangement of uses, has prodded the improvement of open-source programming advances such as Hadoop, and is making a whirlwind of enthusiasm for the scholarly in addition to researchers from industry around enhancements and optional ways to deal with taking care of the requests of breaking down extensive complex information sets. The HPC people group is constantly vigilant for approaches to influence more extensive product speculations to distinguish mass-market advances which could be viably connected to take care of issues particular to high-performance computing .For example there are a few cases where outing of the rush of mass-business sector innovations has fundamentally affected HPC. Utilizing bigger business sector advances and the related economy-of-scale is being a compelling method for lessening prices and drawing in a bigger group. HPC, much competent devices, and regularly expanding reenactment scales are creating information at a pace that surpasses our capacity to viably oversee, break down, and distribute it. Execution of MapReduce pattern gives a method for extensive information volumes that could be consistently handled through utilization of vast item PCs. Hadoop and MapReduce have generally been utilized for web information handling and just as of late been utilized for exploratory applications. Our recommended framework comprehends the effect of document framework, programming modes on execution and system. Acquirement of an application's execution is to a great extent work load subordinate. Configuration of each MapReduce frameworks needs to incorporate the Key components such as, precise output, Quick Response and High speed. Enhancement of the Fault resilience of Map Reduce Implementation Adapted for High Performance Computing Environments in Distributed and Parallel Computing is considered to be the primary objective of this work.

Selection and Peer-review under responsibility of International Conference on Processing of Materials, Minerals and Energy (July 29th – 30th) 2016, Ongole, Andhra Pradesh, India.

2214-7853
Selection and Peer-review under responsibility of International Conference on Processing of Materials, Minerals and Energy (July 29th – 30th) 2016, Ongole, Andhra Pradesh, India.

Keywords: MARIANE, MapReduce, Job Scheduling, Fault Tracker, Fault Tolerance, Fault Controller, Weight Based Approach

1. Introduction

MapReduce (MR) is a circulated processing model. The two critical elements of Hadoop are Map and Reduce. Map acknowledges set of info and changes it into middle key-esteem pair. Every mapper stacks the arrangement of records neighborhood to that system and procedures them. All the qualities connected with the same key would be joined by MapReduce library and exchanges these key-esteem sets to Reduce capacity. This can be considered as the main correspondence venture in MapReduce. Due to the littler arrangement of qualities the Reduce capacity will change over these qualities. Generally only one or zero yield quality is delivered per Reduce function [1]. The midway points are given to the client's reduce capacity through an iterator which permits us to operate arrangements of qualities which are much extensive to get in storage. Individual map assignments don't impart data to one another. They are oblivious about presence of different mappers. The limit of designing the jobs is vested with Hadoop, put forwarding them and managing their execution. Malfunctions are dealt by MapReduce. In the event that specific hub fizzles, then required information could be garnered from the nearby hub [2]. In Hadoop the task scheduling incorporates designating fitting assignments of the occupations to proper server. Hadoop Scheduling Model (HSM) comes under the category of The Slave cluster model / Master. All specialist hubs are being facilitated by the JobTracker. JobTracker is in charge of administration of all undertaking servers while TaskTracker performs assignments on the relating hubs. In JobTracker one can find the availability of the scheduler. The Scheduler is undertaken to ensure allotment of assets to TaskTrackers for performing the tasks [3]. MapReduce (MR) has every now and again been connected with large-scale texts and Hadoop. It is filled with the capability of organization and functions at Yahoo and Amazon in the cloud respectively! Also, Facebook for vast scale disseminated record indexing and database working, amongst different undertakings, have pushed MR to the bleeding edge of the information handling application area. The appropriateness of the model nevertheless reaches out a long ways past its utilization with information concentrated applications and diskbased frameworks, and can likewise be gotten to hold up under preparing little yet CPU intensive distributed applications [4]. There are different disseminated record frameworks such as Lustre, Parallel Virtual File System (PVFS), Hadoop Distributed File System (HDFS) and Google File System (GFS). The HDFS [6] and GFS [5] are the much widely recognized document framework sent in huge scale dispersed frameworks, for example, Google, Yahoo today and Face book. An open source RAID-0 style parallel record framework for clusters is PVFS [7]. It parcels a document into band of units and appropriates these bands to circles in a robin style. PVFS comprises of 1 metadata server and a few information servers. All information movement of document substance streams amongst customers and information server hubs in parallel without experiencing Metadata Server. The deadly burden of PVFS is that it doesn't give any Fault-Tolerance in its present. The disappointment of Single Server hub will influence the entire record framework.

A shared disk document framework is Lustre which is generally utilized for cluster of large scale. It is a structure with an open-standard with incredible modularity and attuned with interconnects, organizing segments and capacity equipment. For Linux this is accessible in the present state. To display the applications and calculations data-parallel MR [8], [9] a rising programming structure is being used. After its unique cluster-based usage, it has been actualized and assessed on different structures, which include the GPUs [11], Cell B.E. [10] and multi-center processors [12].

2. Literature Review

The term Big data refers to the monstrous information sets having substantial, more fluctuated as well as structure that is complex concerning challenges being keeping away, examination and figuring out of the further process and outcomes. The exploration procedure has a huge measurement to uncover the covered information, examples and mystery connections are called as Big Data. The Big Data development has increased various eyebrows to the extent the difficulties are alarmed. A few creators have found a plenty of difficulties which incorporate information stockpiling and protection. Sachchidanand Singh et al. clarified the idea, qualities and the use of Big Data and diverse offerings accessible in the business sector to investigate unstructured vast information as in [13]. Changqing ji et al. talked about the extent of Big data preparing in a Cloud Computing setting [14].

Design for overseeing and preparing Big Data utilizing framework innovations was proposed by Dan Garlasu et al. as in [15]. Xiaoxue Zhang et al portrayed the capacity difficulties of Big data and they broke down those utilizing Social Media Networks as illustrations as in [16]. They moreover characterized the associated exploration problems existent in accompanying arrangements: replica consistency, de-duplication, load balancing and small files problem. Similarly a few studies were done by Meiko Johnson on the security issues required with Big data.

3. Proposed Work

3.1 Proposed HPC

The community of HPC is ceaselessly vigilant for approaches to influence more extensive ware ventures to recognize mass-market advancements which could be adequately connected to take care of issues particular to HPC. There are a few illustrations where riding of the rush of mass-business sector advances has fundamentally affected HPC. Utilizing bigger business sector advances and the related economy-of-scale has been a viable method for diminishing expenses and drawing in a bigger group. The developing utilization of GPUs in HPC is a late case.

3.2 MapReduce

A structure used to compose applications that procedure a lot of information in Parallel on Clusters of product equipment assets in a solid, fault-tolerant way is called as Hadoop Map Reduce. At first MR Job partitions the information into individual pieces which are handled by Map occupations in parallel. Structure sorts out the maps' yields which are the contribution to the reduce undertakings.

3.3 Hadoop Distributed File System (HDFS)

HDFS is a record framework that traverses every one of the hubs in a Hadoop Cluster for information stockpiling. It connects together document frameworks on neighborhood hubs to convert it into an expansive record framework. HDFS enhances unwavering quality by reproducing information over various sources to overcome nodes hub.

3.4 Fault Tolerance

When the Hadoop utilizes task and information piece replication to adaptation to internal failure closes, we selected a hub particular fault-tolerance mechanism, as opposed to info of particular one. Considering the given methodology, hub disappointment does not affect information accessibility, and new hubs can be allotted fizzled work with little requirement for data relocation.

3.5 Task Tracker and Job Tracker: The MapReduce Engine

The Job Tracker will push out the work to TaskTracker hubs for accessing in cluster, endeavouring to maintain the task in nearest phase so information could be reasonably expected. With a track-aware document framework, the JobTracker deciphers which hub store information and any distinct are being differently adjacent. In this event, the work couldn't be facilitated over genuine hub where the information lives, requirement is provided to the hubs being in the similar track.

3.6 Worker Failure

The Server Pings each customers occasionally. In the event that no reaction is gotten from customers in a specific measure of time, the server denote the customers as fizzled. Any map assignments finished by the customers that are being reset back to particularly at their underlying unmoving state, and along these lines get to be qualified for planning on different customers.

3.7 Fault Tracker

The Fault-Tracker counsels the task culmination table given by Master Node (MN), and being Re-assigns fizzled assignments to finished hubs. Not at all like Hadoop, does the re-assignment exclude finding pertinent information pieces to every particular Task and replicating them by identifying the lump, if those lumps are not neighbourhood

to the saving hub.

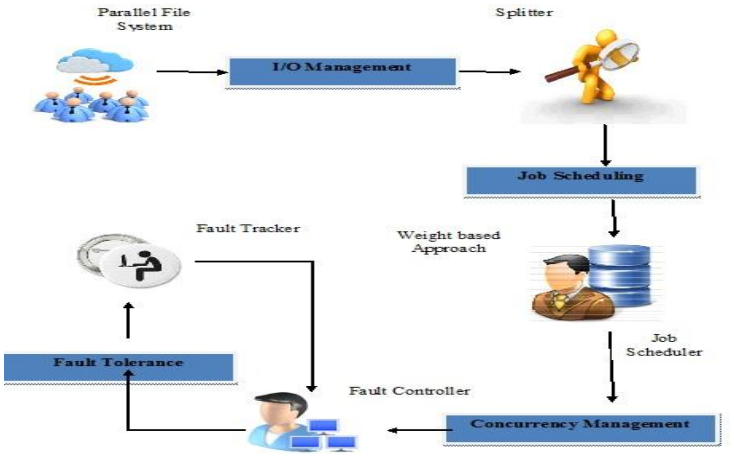

Figure 1: Improved HPC Environment Architecture

3.8 Algorithm

Job Scheduling: Weight based Approach

1. Source code and input data prepared by user
2. Forward the details to Job splitter
3. Divide the Job into number of tasks
4. Send divided task and input data to Job scheduler
5. Compute the each task weight using source code and input data

$$W(T) = \sum_{l=1}^{N} L + \sum_{i=1}^{M} I$$

where W (T) is the weight of the task, L is Source Code Line, I is the Input Data

6. Sort the weight of tasks in descending order
7. Job Scheduler request to get computation status from each node
8. Find the maximum weighted task from W (T)
9. Assign the task to corresponding node in the sorted order
10. Compute the task and completed

4. Result and Discussion

4.1 Experimental Setup

The proposed approach is implemented with following system configuration: OS-Windows 7 professional, Processer-Intel Pentium(R), speed-2.90 GHz, RAM-2GB.

4.2 Performance Accuracy

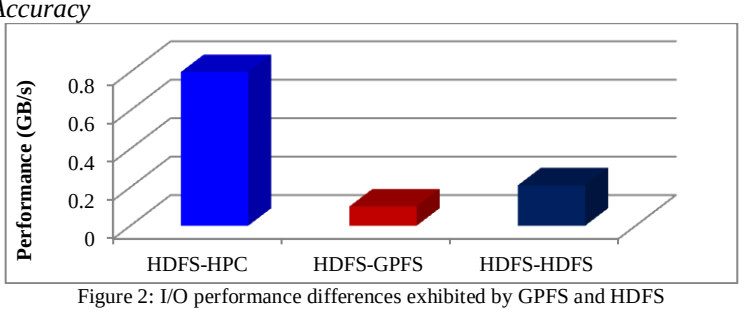

Figure 2: I/O performance differences exhibited by GPFS and HDFS

The framework of the MapReduce is processing very large or huge dataset, it ought to likewise be noticed that record framework read/write exchange rates without a doubt assume a given part in the execution appeared at this point. As Figure 2 describing, GPFS gives quicker document operations than its HDFS.

4.3 Time Consumption Accuracy

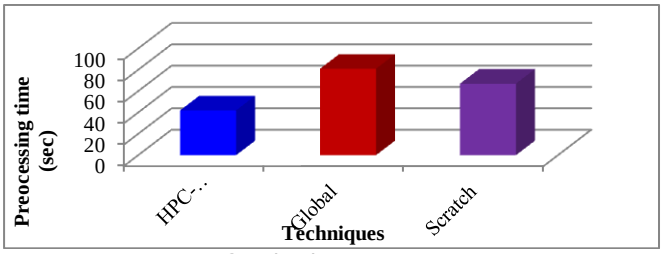

Figure 3: MARIANE Processing Time

4.4 Speed Accuracy

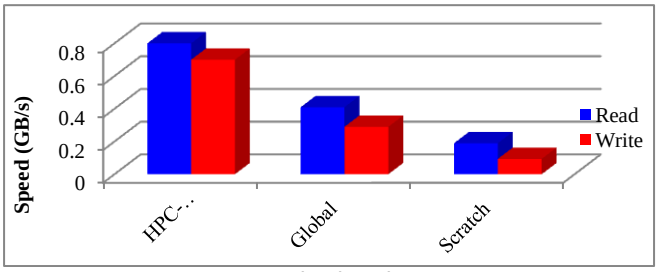

Figure 4: MARIANE Accuracy

MARIANE is tried with various pace appraised GPFS stores as Figure. 3 and 4 appears, the condition don't specifically map to significantly lesser handling times. Figure. 3 and 4 demonstrates the completion of MARIANE with the similar size dataset at the top, indicates 32-hub MARIANE clusters, upheld by three distinctive rate evaluated forms of GPFS, every positioning more than 1.7 billion given urls on the NERSC Cluster.

4.5 HADOOP and MARIANE Processing Accuracy

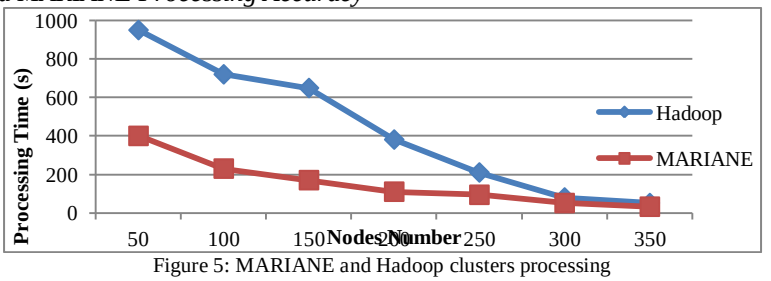

Figure 5: MARIANE and Hadoop clusters processing

The above mentioned figure 5 is describing about the HADOOP and MARIANE clusters preparation through every url that is about 1.7 billion in size within several number of clusters that all extending to 256 hub of clusters from 8 clusters. This trial, similar to the past one, was kept running on the NERSC cluster. Hadoop is significantly slower for little clusters as the information administration operation requires more hubs to imitate and spread information, with an end goal for making the neighbourhood for every hub.

4.6 MapReduce Performance

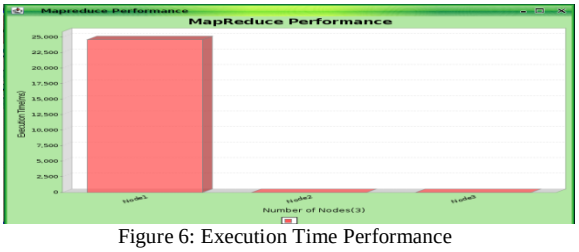

Figure 6: Execution Time Performance

The aforementioned figure 6 is depicting the time taken for execution and performance inside the few hubs of occupation. HPC-MARIANE is creating the time for execution of each employment.

4.7 MapReduce Job Executions

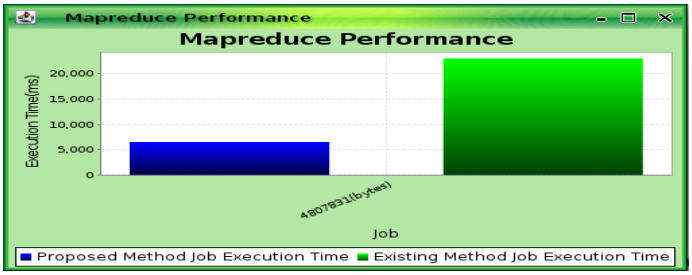

Figure 7 MapReduce Job Execution

The above mentioned figure 7 is discussing the comparison of existing and proposed system job done execution time. The proposed system job done execution time is better than existing system.

5. Conclusion

Our Suggested study is to deal with the Job Scheduling in MARIANE. Splitter deals with the specific occupations and it incorporate documents, for example, settings record, Host record and source code. It likewise isolates the employments into assignments. Weight Based Approach (WBA) was utilized to plan the employments. In weight based methodology, in view of hub status separate the hubs into three distinct classes like medium, quicker and slower. At that point part the occupation into various sizes such as 64 MB, 128 MB and 32 MB. Later, allocate Thirty Two MB task to slower hub, Sixty Four MB task to medium status hub, and One hundred and twenty eight MB task to speedier hub. In Concurrency Management, the Task Tracker screens undertaking process through Cluster Nodes (CN), and Broken Records channels and non-responsive hubs as fizzled.

5.1 Future Enhancement

Our next study will be on the basis of developing a yardstick, wide collection of shared disk file systems with MARIANE.

References

[1] Aysan Rasooli and Douglas G. Down, “A Hybrid Scheduling Approach for Scalable Heterogeneous Hadoop Systems”
[2] B.Thirumala Rao,Dr. L.S.S.Reddy, “Survey on Improved Scheduling in Hadoop:MapReduce in Cloud Environments”, International Journal of Computer Applications (0975 – 8887)Volume 34– No.9, November 2011
[3] Radheshyam Nanduri, Nitesh Maheshwari, Reddyraja. A and Vasudeva Varma, “Job Aware Scheduling Algorithm for MapReduce Framework”, 2011 Third IEEE International Conference on Coud Computing Technology and Science
[4] Zacharia Fadika, Madhusudhan Govindaraju, “LEMO-MR: Low overhead and Elastic MapReduce Implementation Optimized for Memory and CPU-Intensive Applications”.
[5] S. Ghemawat, H. Gobioff, S. Leung. “The Google file system,” In Proc. of ACM Symposium on Operating Systems Principles, Lake George, NY, Oct 2003, pp 29–43.
[6] Konstantin Shvachko, et al., “The Hadoop Distributed File System,” Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium on IEEE, 2010,
http://storageconference.org/2010/Papers/MSST/Shvachko.pdf.
[7] P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur. “PVFS: A parallel file system for Linux clusters,” in Proc. of 4th Annual Linux Showcase and Conference, 2000, pp. 317–327.
[8] Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51(1) (2008) 107–113
[9] L¨ammel, R.: Google’s MapReduce programming model — Revisited. Sci. Comput. Program. 68(3) (2007) 208–237
[10] de Kruijf, M., Sankaralingam, K.: MapReduce for the CELL B.E. Architecture. IBM Journal of Research and Development 52(4) (2007)
[11] He, B., Fang, W., Luo, Q., Govindaraju, N.K., Wang, T.: Mars: a MapReduce framework on graphics processors. In: PACT ’08: Proceedings of the 17th international conference on Parallel architectures and compilation techniques, New York, NY, USA, ACM (2008) 260–269
[12] Ranger, C., Raghuraman, R., Penmetsa, A., Bradski, G., Kozyrakis, C.: Evaluating MapReduce for Multi-core andMultiprocessor Systems. In: HPCA ’07: Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture, Washington, DC, USA, IEEE Computer Society (2007) 13–24
[13] Sachchidanand Singh, Nirmala Singh,‖ Big Data Analytics‖, International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India,2012.
[14] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, Keqiu Li,‖ Big Data Processing in Cloud Computing Environments‖, 12th International Symposium on Pervasive Systems, Algorithms and Networks (ISPAN), IEEE, pp. 17–23, 2012.
[15] Dan Garlasu, Virginia Sandulescu, Ionela Halcu,Giorgian Neculoiu, Oana Grigoriu, Mariana Marinescu and Viorel Marinescu, ―A big data implementation based on Grid computing‖, 11th RoEduNet international Conference, pp. 1-4, 2013
[16] Xiaoxue Zhang, Feng Xu, "Survey of Research on Big Data Storage",12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science (DCABES), IEEE, pp.76-80, SEPT.2013.

About Author

Mr.P.J.Sathish Kumar, Research Scholar at bharath University Chennai, also working as Assistant Professor in department of Information Technology at S.A.Engineering College. I am thankful to Bharath University and S.A.Engineering College and people who helped me in achieving this goal.