International Journal of Engineering &Technology, 7 (4) (2018) 6977-6980

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.26653
Research paper

Improving hadoop performance in heterogeneous big data environments by dynamic slot configurations in mapreduce hadoop programming model

Abolgasem M. Ali Enfais [1] *, Adam Amril Jaharadak [1], Amad Abdelkarim El Marghani [1]

[1] School of Graduate Studies, Management and Science University (MSU), Shah Alam, Selangor, Malaysia
**Corresponding author E-mail: gsooma77@gmail.com*

Abstract

Hadoop has been developed as a platform solution for processing a large scale of data in parallel for different applications in Cloud computing. A Hadoop system can be characterized based on three main factors: cluster, workload, and user. Each of these factors can be described in heterogeneous environment, which reflects the heterogeneity degree of the Hadoop system. This paper investigates the effect of heterogeneity in each of these factors on the performance of Hadoop for different schedulers. Three schedulers which consider different levels of Hadoop heterogeneity are used for the analysis: FIFO, Fair sharing, and COSHH (Classification and Optimization based Scheduler for Heterogeneous Hadoop). Performance issues are introduced for Hadoop schedulers and comparative performance analysis between different cases of jobs submission. These jobs are processed in heterogeneous data environments and, under fixed or reconfigurable slot between map and reduce tasks for Hadoop MapReduce java programming clustering model. The results showed that when assigning tunable knob between map and reduce tasks under certain scheduler like FIFO algorithm, the performance enhanced about 81.42% especially in cases of heterogeneity environment where the workload is decreased significantly and the utilization of computational resources in increased obviously.

Keywords: *Hadoop; MapReduce; Scheduling Algorithms; Workload; Heterogeneous Data.*

1. Introduction

We are residing within the era of large data. In these days a tremendous amount of knowledge is generating everywhere as a result of advances within the web and verbal exchange applied sciences and the pursuits of men and women using smartphones, social media, internet of things, sensor contraptions, online offerings and lots of more. In a similar way, in improvements in knowledge applications and broad distribution of application, a couple of government and commercial organizations such as monetary institutions, healthcare institution, schooling and research division, power sectors, retail sectors, lifestyles sciences and environmental departments are all producing enormous amount of information every day. For examples, the process of enhancing the contrast of computerized tomography (CT) images using a normalization technique that depends on the image size i.e. every increase in image size requires enhancing of processing capabilities per one image. The process creates large data which should be taken in mind when dealing with plenty of images from one hospital then in turn from many hospitals. The matter that enhance us to deal with a massive data [1]. International data enterprise (IDE) said that 2.8 ZB (zettabytes) knowledge of universe had been saved in the year of 2012 and this may reach up to forty ZB through 2020 [14]. In a similar fashion Facebook processes round 500 TB (terabytes) knowledge per day [2] and Twitter generates eight TB data daily [3]. Many developed countries including Malaysia started using E-commerce facilities, there is a necessity from involved country to adopt many measures to well build computer-mediated environment for the retail business to run smoothly. But for growing economies along with increased demand, it's essential to coup up for proper solutions, major part taken on responsibility of government support and minor part taken by private sector, the image which reflects importance to come with solutions that take large data in mind [15]. The huge datasets no longer handiest comprise structured form of knowledge but greater than seventy five percent of the dataset includes uncooked, semi-structured and unstructured type of data [4]. This large quantity of information with one of a kind codecs can be viewed as giant information.

The derivation of big knowledge is indistinct and there are a lot of definitions on huge data. For examples, Matt Aslett outlined massive knowledge as "tremendous data is now virtually universally understood to refer to the recognition of larger business intelligence through storing, processing, and examining data that was previously ignored because of problem of normal data management applied sciences" [5]. Recently, the term of giant data has got a brilliant momentum from governments, industry and research communities [6]. Significant information is outlined as a term that encompasses using tactics to capture, approach, analyze and visualize potentially significant datasets in a cheap timeframe now not obtainable to usual IT applied sciences [7].

Recently, big and parallel data processing has become the main concern for modern data analysis and knowledge extraction for feasible prediction and decision making in many fields of businesses. Data recently has become more complicated, generated in huge volumes and in many cases mixed between structured and unstructured data, so analyzing it exposes many challenges. Moreover, modern data are acquired from different servers' source and from different users in real-time which add more complexity to the

Copyright © 2018 Abolgasem M. Ali Enfais et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

analysis processes. Analysis of these big complicated and parallel data is always beneficial in term of mining and extraction useful information and for knowledge discovery purposes.

Hadoop recently is considered the most promising open source platform for the big and parallel data processing in real-time. Understanding the Hadoop architecture and how it works is a challenge by itself. The target of the utilization of using Hadoop is always to processes the big parallel data in reduced computational cost of the usage of the processors and memory resources as well as clustering efficiently the data in heterogeneity classifications. The accuracy of the data classification is aim also of researching this topic as well as reducing the latency time when answering the end-users enquiries for the purposes of data mining or certain knowledge discovery.

Hadoop MapReduce is a programming model and software framework that supports data-intensive distributed applications. MapReduce is a parallel programming technique derived from the functional programming concepts and is proposed by Google for large-scale data processing in a distributed computing environment. This Apache project is an open source framework for reliable, scalable, distributed computing and data storage. It can rapidly process vast amounts of data in parallel on large clusters of computer nodes. Hadoop MapReduce was inspired by Google's MapReduce [8] and Google File System (GFS) [9] papers.

It was observed that the proficiency in clustering data with big data in cloud was significantly enhanced with the suggested mechanism. Xiong, Luo & Dong, (2014) introduced an innovative Snake like Data Placement (SLDP) methodology for handling a large-scale Hadoop cluster that was organized in a heterogeneous way [10]. A Heterogeneity-Aware algorithm (Haag) was formulated in order to segregate the several nodes all through various Virtual Storage Tiers (VST). The clustered data was positioned in different fragments of nodes in accordance with the sensitivity of data concerned relying upon a Hotness Proportional (HP) replication mechanism. The disk space utilization was significantly alleviated by means of implementing a replication with HP attribute computed. Thus, the sensitivity computed was considered to articulate SLDP methodology that was constructed as an amalgamation of both HP and Haag. Apache Hadoop infrastructure was utilized to recognize the proficiency of the designed approach. It was evidently ascertained that the endorsed methodology was effectual in terms of disk utilization, energy conservation and a better performance realization in a Hadoop cluster framed in a heterogenic manner. Though the mechanism devised was more proficient in several ways, the time factor consumed for processing and clustering the data was not in a minimized way and hence, further optimization was required with this methodology.

Xiong, Luo & Dong, (2015) endorsed an inventive data placement strategy termed as Snakelike Data Placement (SLDP) for processing the clusters available with Hadoop structure [11]. The devised SLDP procedure was deployed in a linear strategic way given by following stages:

- Initially, the Data Nodes that were available with information was segregated into stipulated order of Virtual Storage Tiers (VST) by means of implementing Haag.
- The disk space utilization was assessed in prior through SLDP and the disk space was allocated on the whole at the initial stage itself.
- The Hotness Proportion (HP) was assessed for all data blocks and replication factor was computed in accordance with the HP concerned.
- HP attribute got fluctuated on the basis of the time realized and hence, the disk space was preserved in prior to processing.
- The data was disseminated into data blocks in a way that resembled a practice slow-swimming of a snake.
- The most sensitive block of data was positioned in an initial data node.

The devised SLDP methodology proficient in governing the energy consumption of data nodes and maintain it in a minimized way. The issue prevailing with this technique was the lack of a quick optimization methodology that was capable of working with hefty-sized data.

2. Methodology

Deng et al., (2016) recommended a parallel K-means clustering methodology for processing huge amount of data deployed by means of a Common Information Model (CIM) in order to perform a proficient interchange of either data or information within a network. The approach was completely utilized for enhancing the efficacy of sensitive data-oriented applications. The information gathered from the CIM was disseminated into several systems that dwell within a clustered ambience of MapReduce. The devised mechanism was effectual for assessing and validating data prevailing within CIM. The overall workflow of the proposed methodology was illustrated in Figure 1.

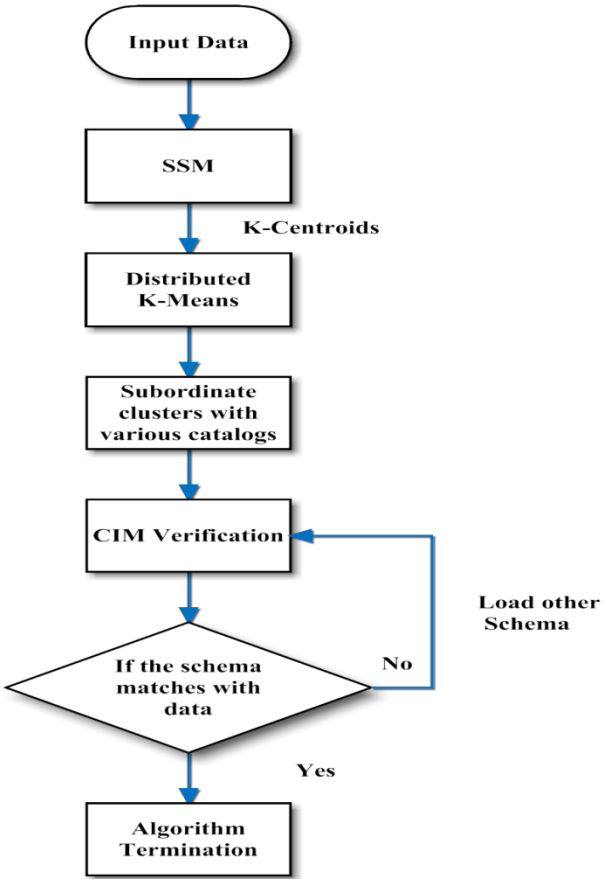


Fig. 1: Overall Workflows of the CIM Approach.

- The information or some other task fed into a clustered Hadoop framework was primarily fragmented into minor segments.
- Moreover, those fragmented segments were archived into HDFS for further processing.
- The jobs enrolled in a Datanode was trailed by means of an implemented Job Tracker within a network.
- The Map functionality was further utilized to create a corresponding match with appropriate Datanodes.
- The matching of tasks was performed by means of gathering the data from the indigenous mappers of the localized Datanodes.
- Ultimately, the final outcome was completely organized by means of the reducing agents.

The sub disseminated work load was frequently cross checked via the CIM points depicted. The task load realized within a working out nodes existing in MapReduce structure was managed by means of implementing a load balancing pattern that was built on the basis of a genetic algorithm. The mapper in the MapReduce

structure was responsible for mitigating the task load and conserve the data node from becoming overloaded. A typical mapper in a MapReduce procedure comprised of certain attributes given by,

- Overall time consumed for Copying data.
- The time span spent for processing a complete data segment.
- Span consumed for merging the input data with that of the appropriate outcome file in order to mitigate the reducing procedures in future.
- Time period gained for draining out the occupied buffer space.

The ascertained mechanism was efficient in managing the data clusters in MapReduce structure, also exhibited a robust load balancing criterion with Data-nodes. The drawback prevailing within the approach devised was the allocation and re-adjustment of Data-nodes with the cluster in a network.

Park, Kang, Hong & Cho, (2016) précised all sorts of the prevailing approaches that was capable of enhancing the indigenous nature of data within each and every node. Recommended a data replication mechanism for resolving the duplication and allotment issues. The problem of data duplication was alleviated by means of projecting an inventive methodology given by Least Recently Frequently Access (LRFA) replica eviction policy. Both the sort of data retrieved in a most recurrent and least recurrent way was assessed in HDFS [12].

- On realizing a comparatively minimized admitting incidence than that of the HDFS processing instance, then the data was mitigated by LRFS mechanism.
- The aforementioned task was accomplished in order to utilize the occupancy of disk space in an effectual manner.

The proposed LRFA methodology was proficient in improvising the indigenous tendency of data though, it was not as much effectual while realized on a real-time environment.

The given work uses K-Means clustering algorithm on a benchmark MRI dataset from OASIS database, in order to cluster the data based upon their visual similarity, using WEKA. Until a threshold size it worked out and after that compelled WEKA to prompt an emergency message "out of memory" on display. A Map/Reduce version of K-means is implemented on top of Hadoop using R, so as to cure this problem. The given algorithm is evaluated using Speedup, Scale up and Size up parameters and it neatly performed better as the size of the input data gets increased [13].

Thakare, et al., (2016) endorsed a survey on several clustering mechanisms and presented an effective processing methodology to cluster and process the big data. Initially, the dimensionality of the original data available on the whole was reduced by means of splitting into numerous sub-parts via implementation of the MapReduce structure. Complete data were fed into a master node and furthermore, the data concerned was processed by allocating into a several sub-nodes termed as Datanodes that were organized into a multi-level tree arrangement. Afterward, the fragmented data was correlated and processed as clusters. In order to cluster the big data, the feasibility of implementing numerous clustering algorithms like k-means algorithm and bisecting k-means algorithms were examined

- In k-means clustering methodology, the procedure was initiated with a supposition of keeping all data within a single cluster.
- Subsequently, after realizing the similarity in existing between the data the data were further augmented into several homogeneous clusters.
- In case of implementing bisecting k-means algorithm, it was the assimilation of both hierarchical k-means and k-means algorithm.
- The cluster was primarily opted for fragmenting with utilization of the k-means algorithm and was proceeded until necessitated number of clusters were attained.

These clustered structures were capable of realizing the efficacy in processing a huge amount data. Even the complicated associations were revealed by means of processing with the clustering mechanisms. Flowchart of heterogonous environment demonstrated in figure 2.

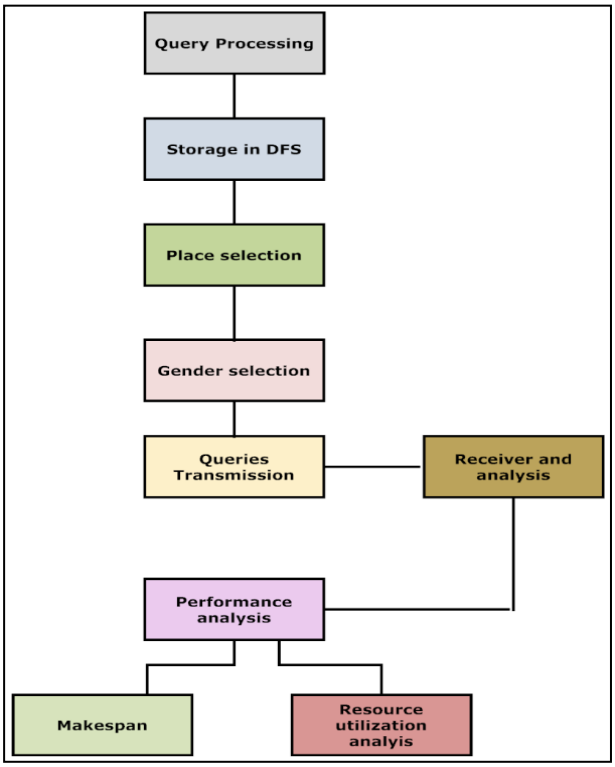


Fig. 2: Flowchart of Heterogonous Environment Implementation.

3. Applied methodology (simulation)

The development environment is accomplished in java coding with integrated interface management of the NetBeans, and the WAMP server is supporting all the required server services, therefore the implementation can be done properly step by step following these steps.

- Firstly, slot configurations for Heterogeneous Hadoop clusters. Then, Heterogeneous Hadoop cluster is chosen for the further step in implementation.
- Query Processing in Heterogeneous servers

The query processing in the respective heterogeneous servers consists of two different fields denoting the servers and their storage.

The modules which are used for client server communication process in Receive and analysis phase are,

- Receive
- Analysis
- Send.

Followed by the Receive and analysis, the respective fields displayed in receive and client process queries are Name, Place and Gender.

- Query Processing in Heterogeneous servers with modules

After selecting the heterogeneous cluster, and getting the complete usage about the modules, variables are stored in the individual DFS.

- Adding data in Query processing

Choosing the respective place, gender and entering the name from the query processing fields.

- Sending Queries

Once the data is loaded in the server and stored in DFS, the respective fields are selected and client query fields are send, received and corresponding data is loaded in the server.

- Receive and analysis

To match and check the values present in receive and analysis phase is same as the DFS server.

- Sending values to respective receiver (client)

Sending the values to the respective receiver client queries and it is again sent to the server.

- Performance analysis

The makespan and resource utilization are calculated and values are shown in the corresponding fields. The Query processing fields consists of patient details along with disease specification and makespan and resource utilization focuses on calculating the processing speed.

4. Results

Time taken for query processing in Heterogeneous server is 70,000 millisecond (ms) and existing system is 13,000 ms.
Comparison of Query Processing over existing system:

Percentage of Improvement =

$$\frac{\text{ET for existing} - \text{ET for QP}}{\text{ET for existing}} \times 100 \quad (1)$$

$$= \frac{70000-13000}{70000} * 100$$

$$= 81.42\%$$

Hence, the query processing system is 81.42% efficient than the existing system which shown in figure 3.

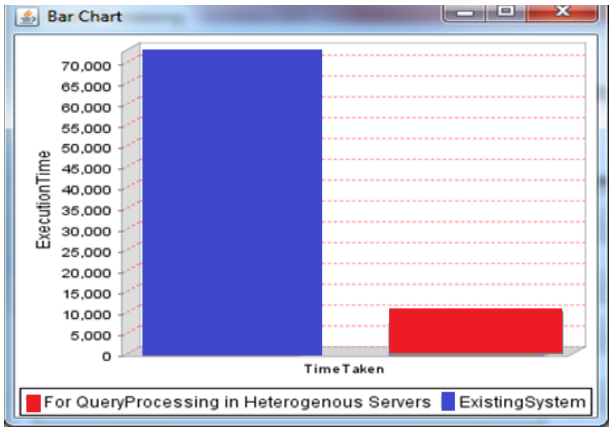


Fig. 3: Query Processing in Proposed System versus Existing System.

5. Discussion and conclusion

The Internet and the World Wide Web are initially designed to move data and information from one location to another in a reliable and most efficient manner. The idea of 'sharing' at the time was worded in the form of work and research documents essential to the few who made use of the technology. After almost 20 years since Dr. Barnes-Lee created the web, the idea of sharing has taken on a whole new dimension. The success and popularity of social networking sites show that the idea of online sharing has been successfully taken to the social and personal level. The sharing of information from the most important to the most mundane ones could now be done in any digital forms: texts, documents, photos, videos and live streaming. It is undeniable that social networking is hugely popular and is here to stay. The use of social networking by individuals, students, lectures, organization and governments opens up a great opportunity of beneficial rules but at the same time it increased work load due content of massive data in usage[16]. Such thing necessitates researchers in related field to work on systems that can utilize and process huge data in real-time factor. Therefore, this study emerged on developing solution to problem of wasted time in large data processing in heterogeneous environment and decreasing value of makespan in Hadoop.

The Query processing fields consists of patient details along with disease specification and makespan and resource utilization focuses on calculating the processing speed and RAM processing in that field. The utilization of resource and configuration mechanisms are used to decrease the makespan between map and reduce tasks as discussed. The experimental outcomes demonstrate the efficacy and robustness of schemes under both simple workloads and more complex mixed workloads. The effect of heterogeneity in each of these factors on the performance of Hadoop schedulers is clearly stated. Performance metrics and for Hadoop schedulers executions are used in evaluation. In this paper, the performance results of the existing and proposed techniques are analyzed and compared in terms of makespan, resource utilization, execution time. The proposed query processing shows 81.42% improvement in execution time than the existing system. From the results and discussion, it is concluded that the proposed technique provides better results when compared with the existing technique.

References

[1] Al-Ameen, Z., Sulong, G., & Johar, M. G. M. (2012). Enhancing the contrast of CT medical images by employing a novel image size dependent normalization technique. *International journal of Bioscience and bio-technology*, *4*(3), 63-68.

[2] Issa, J. A. (2015). Performance evaluation and estimation model using regression method for hadoop WordCount. IEEE Access, 3, 2784-2793 https://doi.org/10.1109/ACCESS.2015.2509598.

[3] Zhao, Y., Wu, J., & Liu, C. (2014). Dache: A data aware caching for big-data applications using the MapReduce framework. Tsinghua science and technology, 19(1), 39-50. https://doi.org/10.1109/TST.2014.6733207.

[4] Zhang, Z., Cherkasova, L., & Loo, B. T. (2013, April). Benchmarking approach for designing a mapreduce performance model. In Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (pp. 253-258). ACM. https://doi.org/10.1145/2479871.2479906.

[5] Rizvandi, N. B., Taheri, J., & Zomaya, A. Y. (2011). Some observations on optimal frequency selection in DVFS-based energy consumption minimization. Journal of Parallel and Distributed Computing, 71(8), 1154-1164 https://doi.org/10.1016/j.jpdc.2011.01.004.

[6] Rizvandi, N. B., Zomaya, A. Y., Boloori, A. J., & Taheri, J. (2012). On modeling dependency between mapreduce configuration parameters and total execution time. arXiv preprint arXiv:1203.0651.

[7] Jaiswal, N., & Bhatt, M. (2017). Big-data application using the mapreduce framework.

[8] Lämmel, R. (2008). Google's MapReduce programming model—revisited. *Science of computer programming*, *70*(1), 1-30. https://doi.org/10.1016/j.scico.2007.07.001.

[9] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). *The Google file system* (Vol. 37, No. 5, pp. 29-43). ACM. https://doi.org/10.1145/1165389.945450.

[10] Xiong, R., Luo, J., & Dong, F. (2014, November). Sldp: a novel data placement strategy for large-scale heterogeneous hadoop cluster. In advanced cloud and big data (CDB), 2014 second international conference on (pp. 9-17). IEEE. https://doi.org/10.1109/CBD.2014.57.

[11] Xiong, R., Luo, J., & Dong, F. (2015). Optimizing data placement in heterogeneous hadoop clusters. Cluster computing, 18(4), 1465-1480. https://doi.org/10.1007/s10586-015-0495-z.

[12] Park, D., Kang, K., Hong, J., & Cho, Y. (2016, April). An efficient hadoop data replication method design for heterogeneous clusters. In proceedings of the 31st annual acm symposium on applied computing (pp. 2182-2184). Acm. https://doi.org/10.1145/2851613.2851945.

[13] Shaikh, T. A., Shafeeque, U. B., & Ahamad, M. (2018). An Intelligent Distributed K-means Algorithm over Cloudera/Hadoop.

[14] Singh, S., & Liu, Y. (2016). A cloud service architecture for analyzing big monitoring data. Tsinghua Science and Technology, 21(1), 55-70. https://doi.org/10.1109/TST.2016.7399283.

[15] Johar, M. G. M., & Awalluddin, J. A. A. (2011). The role of technology acceptance model in explaining effect on e-commerce application system. *International Journal of Managing Information Technology*, *3*(3), 1-14. https://doi.org/10.5121/ijmit.2011.3301.

[16] Hamat, A., Embi, M. A., & Hassan, H. A. (2012). The use of social networking sites among Malaysian university students. *International Education Studies*, *5*(3), 56. https://doi.org/10.5539/ies.v5n3p56.