*Adv Syst Sci Appl2020; **01**; 13-26*
*Published online at http://ijassa.ipu.ru/index.php/ijassa/article/view/761*

# Enhancing Hadoop Performance in Homogenous and Heterogeneous Big Data Environments by Dynamic Slot Configuration

Ekhlas K. Hamza

[1)] *Control and Systems Engineering Department, University of Technology, Iraq*
*E-mail: 100374@uotechnology.edu.iq, drekhlaskadhum@gmail.com,*

**Abstract:** Hadoop is one of the most famous platform solutions for processing large volume and scale of data in parallel processing 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 as either heterogeneous or homogenous, which reflects the heterogeneity degree of the Hadoop systemThe objective of this proposed research work is to investigate the degree of influence of heterogeneity for each of these factors on the performance of Hadoop based on different schedulers. Three schedulers are considered with different levels of Hadoop heterogeneity and are tested and analyzed: the first algorithm considered is the FIFO (First in First out), the second is the Fair sharing, and the final is the COSHH (Classification and Optimization based Scheduler for Heterogeneous Hadoop). Performance issues are related to Hadoop schedulers and comparative performance analysis between different cases of jobs submission. These jobs are processed in different homogenous or 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 schedulers like FIFO algorithm, the performance enhanced significantly especially in cases of heterogeneity environment where the workload decreased significantly and the utilization of computational resources increase was obvious.

***Keywords***: Hadoop, MapReduce, big data, slot configuration, scheduling algorithms, MakeSpan, resources utilization.

## 1. INTRODUCTION

Recently, big and parallel data processing has become a main concern for modern analysis and knowledge extraction for feasible prediction and decision making in many fields of businesses. Data has recently become more complicated when generated in huge volumes and in many cases mixed between structure and unstructured data. Thus, analyzing it exposes many challenges. Moreover, modern data is acquired from different server sources and from different users in real-time, which add more complexity to the analysis processes [1],[2]. Analyzing these big complicated and parallel data is always beneficial in terms of mining and extraction of useful information and for knowledge discovery purposes [3].
Recently, Hadoop has been considered as the most promising open source platform for the big and parallel data processing in real-time [4]. Understanding the Hadoop architecture and how it works is a challenge by itself. The target of utilizing Hadoop is always to process the big and parallel data thereby reducing the computational costs of the usage of the processors and memory resources [5] as well as clustering efficiently the data in homogeneity and heterogeneity classifications. The accuracy of the data classification is aimed at researching

this topic, as well as reducing the latency time when answering the end users' enquiries for the purposes of data mining or discovery of certain knowledge [6].

From the beginning, this research has been motivated from certain challenging factors, which are as follows: Firstly, simulating the real-time Hadoop homogenous and heterogeneous environments in one desktop. Secondly, to simulate different users' enquiries and even if they are heterogeneous users it may mean loading the data to Hadoop from different distributed file systems and servers. Besides that, the MakeSpan or alternatively the workload should be determined especially for job enquiries submitted to Hadoop from different distributed file systems and the data obtained from different heterogeneous clusters [7]. The focus here is to analyze the volume of workload assigned to each job, which is assigned to the Hadoop system by determining the consumption of the processors and memory sources as well as the time taken for completing the job. Most important of all is how to reduce the MakeSpan or the workload in Hadoop MapReduce architecture, especially for obtaining data from heterogeneous clusters loaded from different distributed file systems (servers).

Mainly fixing the number and the ratio between map and reduce processes by using traditional scheduling algorithms such as FIFO and Fair Sharing algorithms seems to have some difficulties as some jobs maybe stuck sometimes, especially in case of FIFO scheduling [8]. Further, the job convergence is not optimized especially in the case of dealing with intensive heterogeneous data environments. This will lead to increased workload (Make Span), which consumes more computational resources and resulting in more time processing. The proposed system implements a mechanism to dynamically allocate slots for map and reduce tasks or alternatively it means optimally adjusting the ratio between map and reduce processes by using more efficient scheduling algorithms for optimized classification in a heterogeneous data environment. This, as the results show, will decrease the Make Span and more efficiently utilize the processors and memory resources.

## 2. RESEARCH BACKGROUND AND LITERATURE REVIEW

In this section, the main concepts of this research field are defined and explained to understand the system architecture of the proposal and how it works in processing and analyzing big and medical data in real time. At this point it is important to understand from the beginning the characteristics and definition of the modern big data. Big data can be identified mainly in three parameters, which are complexity, volume and dynamicity. As regards the complexity parameter, this will make sense when it is understood that the generated data in recent modern scenarios are coming from different sources and formats. This can be text, audio, videos, symbols or numbers, which are different extensions and formats. It is clear that dealing with such unstructured data will impose enormous complexity for purpose of analysis and data mining. There should be efforts during pre-processing to make such data more uniformed in order to make pattern recognition feasible. With reference to the volume parameter, the volume and the size of the data are increased rapidly, dynamically and spontaneously in real-time. This can become clear if we imagine that in modern scenarios of medical data collection to support personal health records and monitoring health status remotely, the data are collected by different sensors for different health parameters such as heart rate, blood pressure, body temperature and glucose levels. All these data are collected simultaneously and in parallel in one database center for further analysis so as to send certain pattern recognition or trend analysis as results for end users to allow them to monitor their health conditions. In this case the data is growing exponentially in size, volume and dynamically in real-time with different formats and structures.

In the centralized management efforts of the big data analysis, another added complexity factor is represented as the data come from different distributed sources and in many cases from heterogeneous sources. Hadoop recently appeared as the most interesting open-source software platform utilized for distributed storage and parallel processing of big data by using the MapReduce programming model [9]. The main parts of the Hadoop architecture consist of the storage part, which is known as Hadoop Distributed File System (HDFS) and the processing part, which is the MapReduce programming model. Hadoop's main job is to split large blocks of data and distribute them across nodes in different clusters in parallel and simultaneously [10]. The simplest cluster contains one master node and multiple slave nodes. The master node keeps tracking the tasks acomplished in the slave nodes by the job tracker in each master node.Some researchers have discussed scheduling techniques in [ 12-15]. They have provided scheduling-aware data, prefetching and eviction mechanisms based on Spark, Alluxio and Hadoop.

The main advantage of the MapReduce programming model in Hadoop is scaling the data processing over multiple computing nodes. First, the Map process takes a set of data and converts it to another form by simply breaking the data into (Key/Value pairs) format. Thereafter, the reduce process takes the output from the map task as input and then shuffles the data combined, as shown in Figure 1. The jobs for assigned jobs of Hadoop MapReduce are scheduled from the beginning based on certain scheduling algorithms such as First Input First Output (FIFO), Fair Sharing and others.

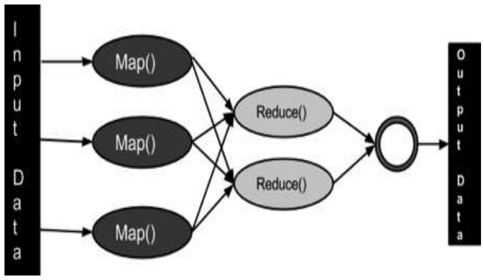



**Fig. 1.** Hadoop Map and Reduce Processes [10]

It is always desirable to reduce the number of maps and processes so as to decrease the workload in Hadoop processing and that depends on the power of the scheduling algorithms as well as the ratio between maps, which reduces implementation. This research work focuses in reducing the workload of the Hadoop processing jobs (referred to as same meaning as Make Span) by adjusting the ratio between maps and reducing the optimal minimum number to achieve the different tasks even in homogenous or heterogeneous data clustering to answer different end-users' enquiries.

The incoming jobs submitted to the Hadoop system can be considered heterogeneous with respect to certain factors such as the number of tasks, data volume, required computation, arrival rates and the execution times. It is worthy to notice that Hadoop resources are different in capabilities of data storage as well as the processing units. This leads to different priorities and minimum requirements for sharing between different users taking into consideration that the number and type of jobs assigned to each user can be different as well. The mean execution time for certain jobs is considered as indicator of the heterogeneity degree for both the workload and clusters in the Hadoop system.

Mainly fixing the number and the ratio between map and reducing processes by using the traditional scheduling algorithms such as FIFO and Fair Sharing algorithms, seem to have some difficulties as some jobs maybe stuck sometimes, especially in case of FIFO scheduling. Moreover, the job convergence is not optimized especially in case of dealing with intensive heterogeneous data environment. This will lead to increase in workload

(MakeSpan), which consumes more computational resources and resulting in more time processing. Table 1 shows the comparisons of various scheduling techniques.

**Table 1.** Show the comparative a of various scheduling techniques

| *Scheduler* | *Taxonomy* | *Working* | *Preemption* | *Idea of implementation* | *Environment heterogeneous or homogenous* | *Advantages* | *Disadvantages* |
|---|---|---|---|---|---|---|---|
| FIFO | Non- adaptive | Better work with small clusters | NO | schedule works based on their significances in first-come first-out. | homogenous | .1-rate of entire cluster arrangement method is less 2. simple to implement andcapable | 1-considered only for singlekind of work. 2. Low performance when run manykinds of works. 3. poor response periods for short worksrelated to large works. |
| Fair Scheduling | adaptive | Better work with small clusters | YES | do a equal spreadingof calculatepropertie s among the users/jobs in the system | homogenous | 1-less difficult. 2. mechanisms well when both lesser and large clusters. 3. it can offerfast responseperi ods for small jobs mixed with larger works. | .1-does not consider thework weight of each nodule. |
| COSHH | adaptive | Better work with large clusters | YES | planned to improve the mean completion time of jobs | Heterogeneous | .1improve the overall system per formance. 2. addresses the fairness and the minimum share requirements. | |

## 3. ARCHITECTURE OF THE PROPOSED SYSTEM

In both the homogeneous and heterogeneous Hadoop environments, the two modules are initially added to the job tracker, which is described below.

- Workload Monitoring (WM)
- Slot Assignment (SA)

The function of WM is to collect previous workload information asexecution times are needed for completed tasks. Based on this collected information, the workload of the current map and reduced tasks are estimated. Thereafter, the role of SA is to adjust the slot ratio between map and reduce tasks by using estimated information received from WM. This adjustment is performed on each slave node. Based on adjusting the slot ratio as well as the current slot status, the job tracker is used to assign tasks to slave nodes by both the workload monitor and slot assigner functions. In addition, the task tracker module along with the task manager function for checking the number of individual maps, reduce tasks running on the node. The performance of checking is done based on the new slot ratio received from the job tracker by running the task on each slave node.Fig. 2 shows the architecture of the proposed method

A batch of jobs is submitted from a user to the job tracker. The three modules involved in the job tracker are workload monitor, slot assigner and scheduler. The status of slot assignment and task assignment is communicated between job tracker and task tracker in order to perform the data communication with large-scale information. The task is split into several parts, which are functioned by different slots. Thus, various slot configurations are performed and each slot contains separate task tracker and task manager. The four functions involved in the proposed architecture are:

- The present workloads and their functions are estimated.
- The best slot assignment of each node is decided in order to communicate between the job tracker and task tracker.
- The task is assigned to each slave node.
- Constant monitoring of the task execution as well as the slot occupation situation.

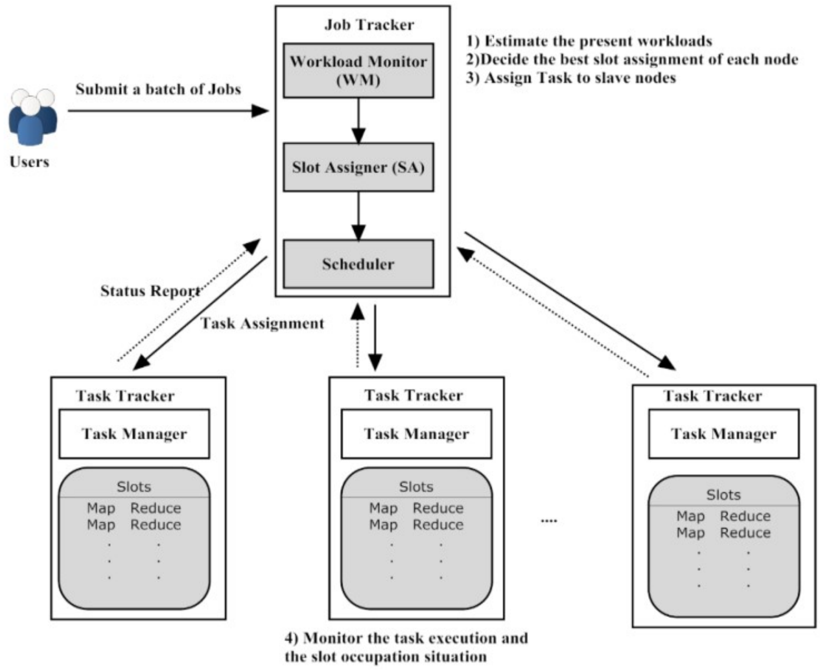



**Fig. 2.** Architecture of the proposed method [11]

The modules involved in the proposed method to minimize the MakeSpan are as follows:

- Homogenous environment
- Heterogeneous environment
- MakeSpan analysis
- Slot configuration prediction
- Evaluation

### *3.1. Homogenous environment*

The homogenous system is defined as the homogenous characteristics of both workload and cluster. Based on the job sizes,the homogenous system can be classified into either homogenous-small or homogenous-large. If all the jobs are small, then the system is defined as homogenous-small. If all the jobs are large, then the system is classified as homogenous-large. Mostly, the job size parameters affect the performance of the Hadoop Schedulers. The average completion time of all schedulers is almost equal. As both the cluster and workload are homogenous, the COSHH algorithm suggests all resources as the best choice for all job classes. The two conditions used to develop homogenous computing environment are as follows:

- The same storage representations along with the same results are considered as correct in hardware and software module of processor for operations on floating point numbers.

– The correct floatingpoint value is transmitted and guarantee is provided to the communication layer during the process of floating point number between processors.

The module involved in the homogenous phase is described as a flow diagram, as shown in Figure 3. The five representations of data-analyzing Hadoop benchmarks are described in the Heterogeneous phase as a module for MakeSpan and slot configuration prediction. The benchmarks are derived from MapReduce Benchmarks Suite, which are described below.

– Inverting the index task: Generate certain list of the words as document indexing when entering a text file as an input.
– Rating the frequency to generate the Histogram: as example, calculating the histogram from certain input of movie rating data.
– Counter for certain word tasks: this job can be described as counting the occurrence of each word when entering input text-file.
– Classifier: when inputting the movie rating data then, classify these movies into predefined clusters.
– Grepping task: to find certain patterns on the files from certain input text files.

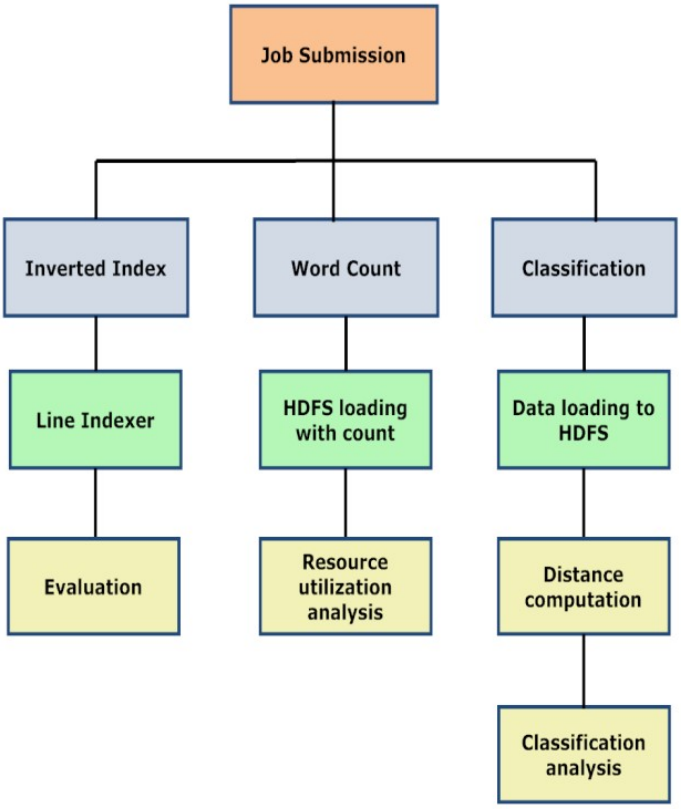



**Fig. 3.** Flow diagram of homogeneous environment

The implementation of the proposed system architecture is achieved by simulating the medical unstructured big data in cloud environment developed in a single desktop using Java programming in NetBeans IDE and Wamp Server services all working together, as depicted in Figure4.

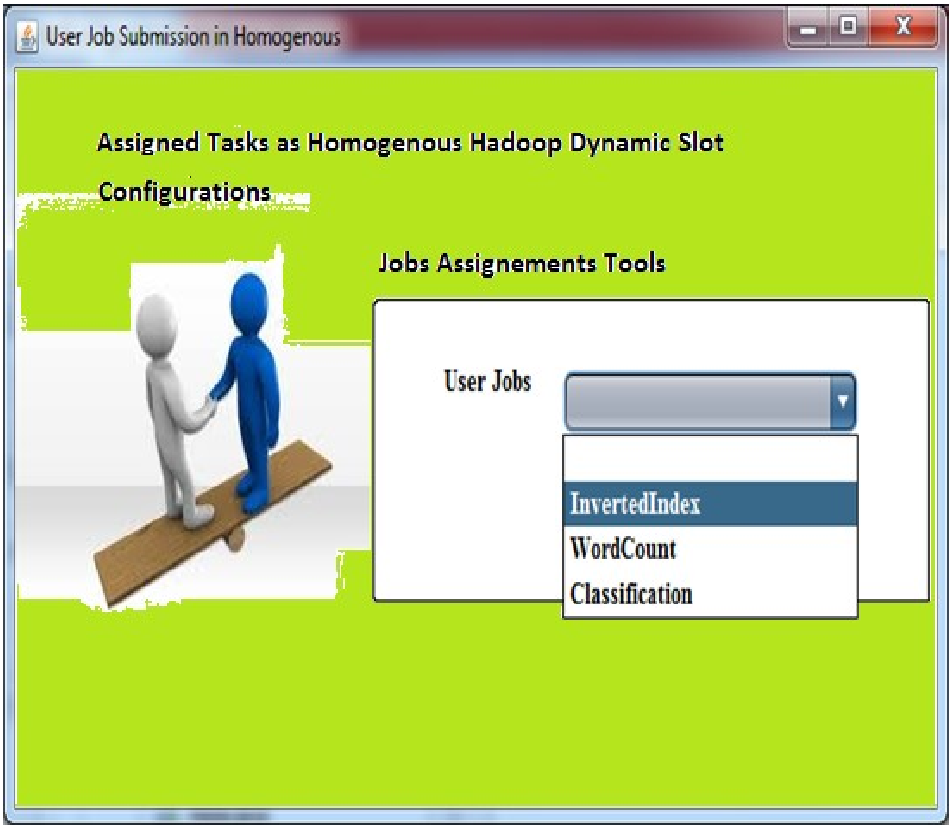



**Fig. 4.** Developed Cloud Computing Platform for Medical Big Data of Homogenous Tasks

### *3.2. Heterogeneous environment*

Based on the workload and cluster factors, the user is either classified as homogenous or heterogeneous. Here, these factors are considered as heterogeneous, thus, the user is also selected from a heterogeneous system.The challenging issue for the schedulers in this system is the arrival rate of different size jobs. Based on the challenges,the system is classified into three such as heterogeneous-small, heterogeneous-equal and heterogeneous-large. If the arrival rate is high for small jobs, then the system is classified as heterogeneous-small. If the arrival rate is equal for all job sizes, then the system is classified as heterogeneous-equal. If the arrival rate is high for large jobs, then the system is classified as heterogeneous-large. Due to the resolving resource and job mismatch problem schedulers algorithm, the average completion time is intended to be improved in all three heterogeneous systems using the proposed architecture. In the proposed method, the modules are divided into three sectionsas follows:

- Query Process
- Three Different Queries
- Sub-query

The processes involved in the proposed heterogeneous platform of medical big data is illustrated in the Figure below.

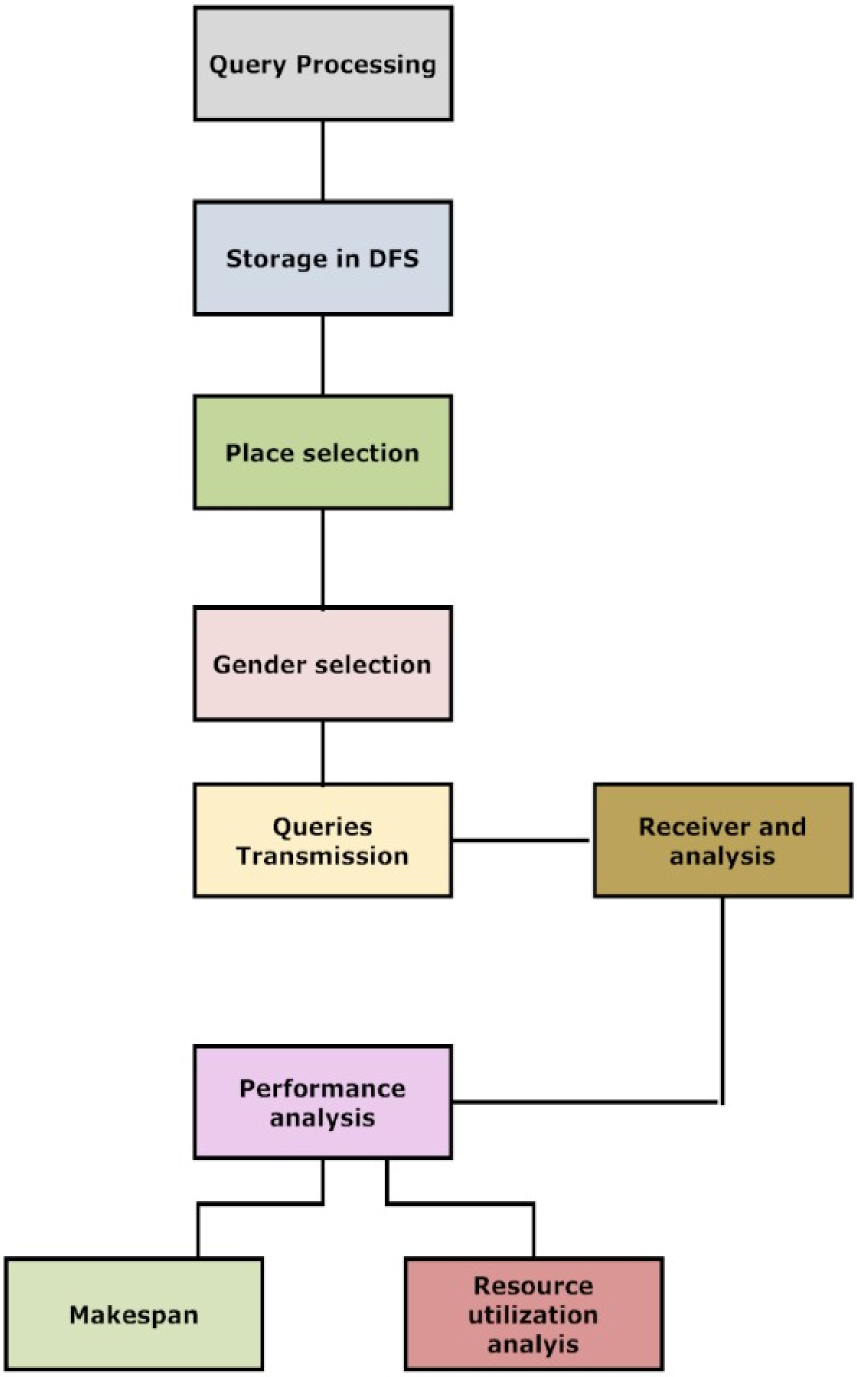



**Fig. 5.** Processes Flow Diagram of Heterogeneous Environment

The implementation of the heterogeneous part of the proposed architecture methodology is depicted from the clients' enquiry side as well as the distributed servers' side, as shown inFigures 6 and 7.

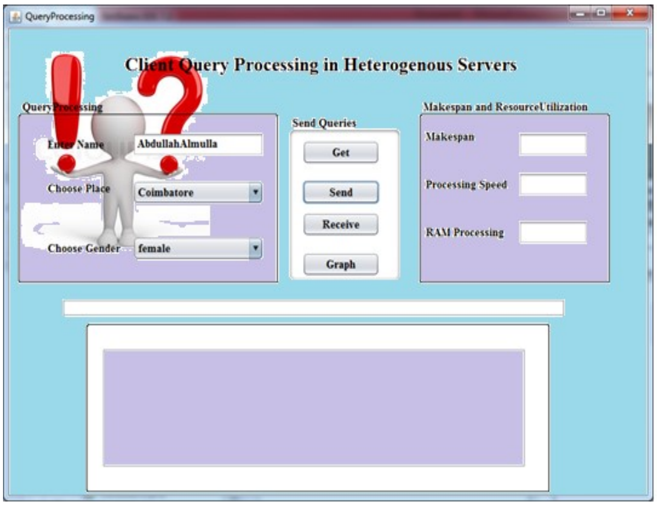



**Fig. 6.** Client Query from Heterogenous Distributed Server in Medical Big Cloud Environment


**Fig. 7.** Processed Data from Heterogenous Distributed Servers in Medical Big Cloud Environment

### *3.3. MakeSpan prediction*

The proposed algorithm Tunable Slot Configurations for Reduced MakeSpan (TuSCRD) is used to improve the performance of a batch of MapReduce jobs by adjusting a basic system parameter with the objective of the proposed system. The significant challenge available in the cloud is a prediction of MakeSpan. The three modules involved in the MakeSpan prediction are described below:

- Launch a Hadoop Cluster
- Tuning the slot assignment
- Tunable Slot Configurations for Reduced MakeSpan (TuSCRD) under FIFO

A classic Hadoop cluster consists of multiple slave nodes and a single master node. The master node executes the JobTracker section, which acts for the scheduling jobs along with coordination between each job's executions of tasks. Each slave node executes the TaskTracker part from the architecture for hosting the execution of MapReduce jobs.

The task is stored as input data in the data node, which is derived as local whereas the remote is denoted by storing all the tasks in another node. If the data nodes are unknown, then the reduce tasks are considered as remote on all nodes and map tasks are considered as scheduler on the input node. Every task is non-preemptable and shares the same deadline and release time as its job. Initially, the system model is considered without failure and with pipeline/speculative execution. Then, jobs are submitted continuously to the cluster and the master scheduler needs to schedule their tasks on the slaves without prior knowledge of future job arrivals.

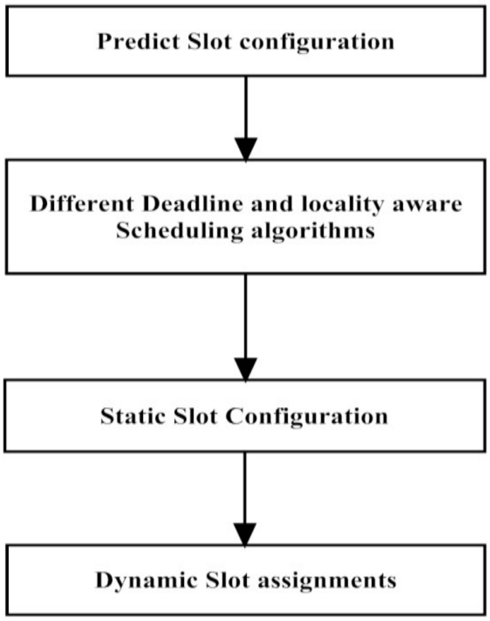



**Fig. 8.** Flow diagram of Slot Configurations

## 4. PERFORMANCE ANALYSIS OF HOMOGENOUS HADOOP SUBMITTED JOBS

MakeSpan and resource utilization are calculated once any homogeneity tasks such as Line Indexer, word counting and classification are clicked or the heterogeneity task of enquiring about medical data from different distributed servers. Figure 9 shows an example of the MakeSpan and CPU processing time as well as the memory space consumed during processing.

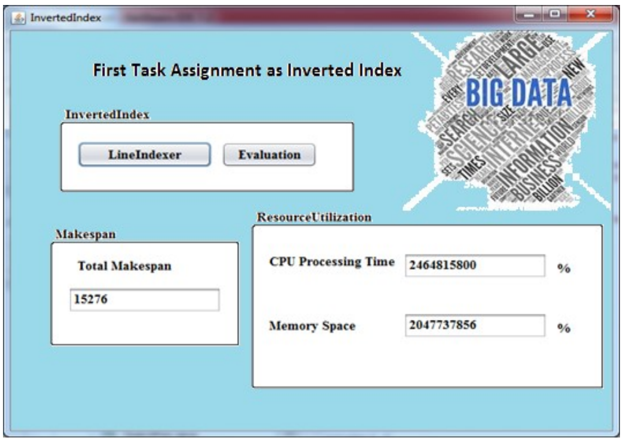
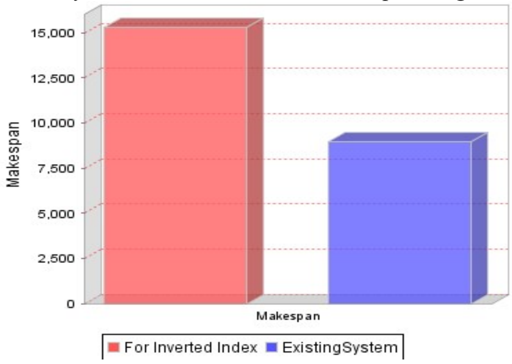



**Fig. 9.** Evaluation of the Inverted Index Homogenous Task in term of Makepsan and Processing time

Table1explains thecomparative values of MakeSpan between the inverted index under tunable slot configuration and the existing system. This shows that the inverted index is slightly higher compared tothe existing system as the medical big data are loaded from the same file.

***Table 1.*** *Performance of tunable slot Solution in Hadoop homogenous Inverted Task*




On the other hand,Tables 2 and 3 show significant enhancements of reducing the MakeSpan workload where the medical big data are loaded from different distributed files.

***Table 2.*** *Performance of tunable slot Solution in Hadoop Classification*

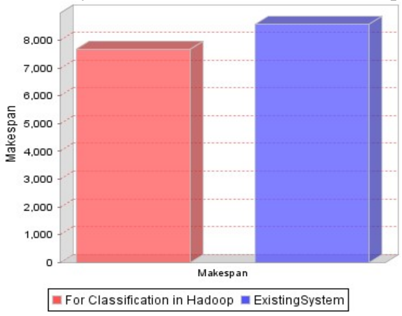
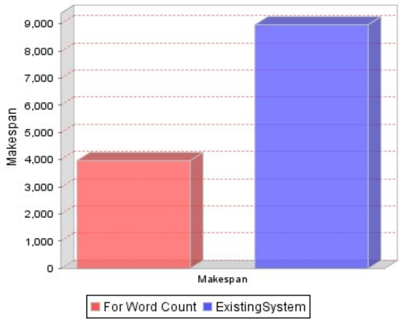



***Table 3.*** *Performance of tunable slot Solution for Word Count*




It is clear even in the homogenous Hadoop big data environment but when loading the data from different files the MakeSpan load is reduced significantly.

In the second part of the implementation of the tunable slot configuration of the MapReduce Hadoop, the environment considered is heterogeneous medical big data environment. In these cases, the end-clients can have different specific enquiries and the data are loaded and processed in different distributed servers.Thereafter, the MakeSpan workload and resources utilization in terms of CPU processing and memory space are determined, as shown in the following Figure.

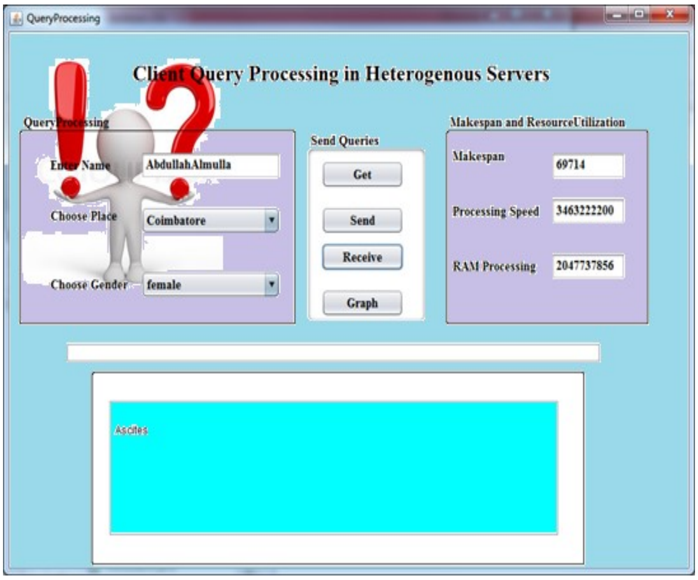
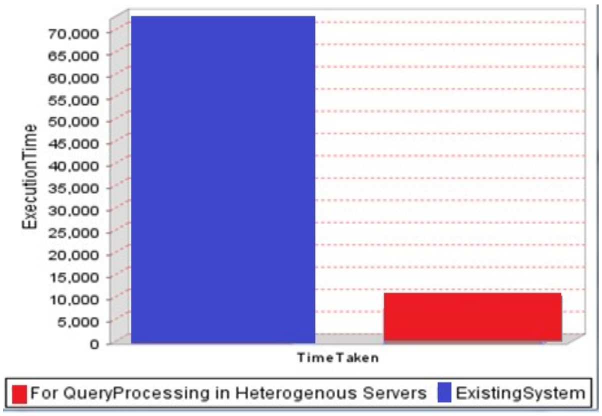



**Fig. 10.** Client Query from Heterogenous Distributed Server workload and resources utilization

Table4 below shows that the workload significantly reduces, in case of loading the data from heterogeneous distributed files from distributed servers with the proposed technique of tunable slot configuration for the Hadoop MapReduce platform of processing the big data.

***Table 4.*** *MakeSpan Work Load Analysis of answering end-client enquires from heterogeneous Hadoop big data environment.*




In brief from the presented data analysis, it clearly shows that allowing the slot between Map and Reduce Processes to be tunable under certain scheduling algorithm in Hadoop big data environment, will result in significant workload reduction and computational resources utilization, especially when the heterogeneity level increased when loading the data from different distributed files and servers.

Recent researchers [16-21] have been presented relating to the big data mobile network. They have discussed the ultimate research of wireless classifications for big data in the structure of a smart factory, which are a) mobile nodes and b) animatedly changing heavy traffic load. These can cause differences in traffic and energy consumptions of cluster head nodes in categorized networks. However, supporting the program big data transmission over 5G wireless systems forces many new and open challenges because the program big data services are both time-subtle and bandwidth-intensive over time-varying wireless channels with controlled wireless resources. The new generations of mobile devices have high processing power and storage but they lag behind in terms of software systems for big data storage and processing. Hadoop is a accessible platform that arrange for spread storage and computational capabilities on clusters of commodity hardware.

## 5. CONCLUSION AND FUTURE WORKS

The problem of high workload and therefore, increased computational resources in Hadoop MapReduce Platform for big data processing, is an attempt to mitigate the optimized Hadoop solution by allowing the slot configuration between maps and reduce the tunable according to certain job priority or homogeneity as well as the available resource spontaneously.

The simulation of the proposed system shows that a high percentage of reduced workload and increased utilization of the computational resources can simply be observed by the lesser number of map and reduce tasks needed to accomplish certain jobs. This is shown in the proposed solution for three jobs submitted in a homogenous Hadoop environment where the data in some cases are loaded from different distributed files in different servers as well as for different job queries in heterogeneous Hadoop clustering environment as the data is loaded as well from distributed files. It is recommended to test as well the other scheduling algorithms such as Fair Sharing and Classification and optimizationbased scheduler for heterogeneous Hadoop (COSHH). It is expected that the performance will be more optimized specially for the COSHH scheduling algorithm with dynamic slot configuration.

## REFERENCES


1. Bhosale, H. S., Gadekar, D. P. (2014). A Review Paper on Big Data and Hadoop. *International Journal of Scientific and Research Publications*, 4(10), 1.
2. Big Data. (2015). New Delhi: Springer, India, Private Ltd.
3. Hilbert, M. (2016). Big data for development: A review of promises and challenges. *Development Policy Review*, 34(1), 135-174.
4. Che, D., Safran, M., Peng, Z. (2013, April). From big data to big data mining: challenges, issues, and opportunities. *In International Conference on Database Systems for Advanced Applications* (pp. 1-15). Springer Berlin Heidelberg.
5. O'Driscoll, A., Daugelaite, J., Sleator, R. D. (2013). 'Big data', Hadoop and cloud computing in genomics. *Journal of biomedical informatics*, 46(5), 774-781.
6. Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, 41(4), 70-73.
7. Raju, R., Amudhavel, J., Pavithra, M., Anuja, S., Abinaya, B. (2014, March). A heuristic fault tolerant MapReduce framework for minimizing makespan in Hybrid Cloud Environment. *In Green Computing Communication and Electrical Engineering* (ICGCCEE), 2014 International Conference on (pp. 1-4). IEEE.
8. Rasooli, A., Down, D. G. (2014). COSHH: A classification and optimization based scheduler for heterogeneous Hadoop systems. *Future generation computer systems*, 36, 1-15.
9. Shvachko, K., Kuang, H., Radia, S., Chansler, R. (2010, May). The hadoop distributed file system. *In Mass storage systems and technologies* (MSST), 2010 IEEE 26th symposium on (pp. 1-10). IEEE.
10. McTaggart, C. (2014). Hadoop/MapReduce. Object-oriented framework presentation, CSCI5448.

11. Dwivedi, K., & Dubey, S. K. (2014, September). Analytical review on Hadoop Distributed file system. *In Confluence The Next Generation Information Technology Summit (Confluence)*, 2014 5th International Conference- (pp. 174-181). IEEE.
12. Frederic Nzanywayingoma, Yang Yang (2018) Task scheduling and virtual resource optimisingin Hadoop YARN-based cloud computing environment *Int. J. Cloud Computing*, Vol. 7, No. 2, 2018
13. Ibrahim Abaker, Targio Hashem, et al (2018) MapReduce scheduling algorithms: a review. *The Journal of Supercomputing* 10 December 2018
14. Sa'ed Abed, Duha S. Shubair (2018) Enhancement of Task Scheduling Technique of Big Data Cloud Computing *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems* (icABCD)
15. Xiaomin Li, Di Li, Song Li, Shiyong Wang, Chengliang Liu (2017) Exploiting Industrial Big Data Strategy for Load Balancing in Industrial Wireless Mobile Networks *IEEE Access* (Volume: 6) 29 December 2017
16. Xi Zhang, Qixuan Zhu (2019) Information-Centric Virtualization for Software-Defined Statistical QoS Provisioning Over 5G Multimedia Big Data Wireless Networks *IEEE Journal on Selected Areas in Communications* (Volume: 37, Issue: 8, Aug. 2019)
17. Johnu George, Chien-An Chen, Radu Stoleru, Geoffrey G. Xie (2019) Hadoop MapReduce for Mobile Clouds. *IEEE Transactions on Cloud Computing* (Volume: 7, Issue: 1, Jan.-March 1 2019)
18. Sudip Misra, Ayan Mondal, Swetha Khajjayam (2019) Dynamic Big-Data Broadcast in Fat-Tree Data Center Networks With Mobile IoT Devices IEEE Systems Journal ( Volume: 13, Issue: 3, Sept. 2019)
19. Muhammad Rehan Raza, Ahmad Rostami, Lena Wosinska, Paolo Monti (2019) A Slice Admission Policy Based on Big Data Analytics for Multi-Tenant 5G *Networks Journal of Lightwave Technology* (Volume: 37, Issue: 7, April1, 1 2019)
20. Cristo Suarez-Rodriguez, Ying He, Eryk Dutkiewicz (2019) Theoretical Analysis of REM-Based Handover Algorithm for Heterogeneous Networks *IEEE Access* (Volume 7) 17 July 2019