Journal of Computer Science Engineering and Software Testing
Volume 4 Issue 1

A Survey on Optimal Scheduler: Improving Efficiency in Parallel Execution Tasks in Hadoop

Aishwarya S[1], Kirthikka Devi D[1], Nandhini J[1], Renown Manjuna G[2]
[1]UG students, [2]Assistant Professor
Department of Computer Science and Engineering, Easwari Engineering College , Chennai-600089 , India
Email: *aishus7959@gmail.com ; kirthikka952@gmail.com; nandhinijeyachandran@gmail.com; renown144@gmail.com*

Abstract
Hadoop's implementation of the Map Reduce programming model pipelines the data processing and provides fault tolerance. Input data is partitioned and distributed as map tasks to individual cluster nodes for parallel execution. Map task splits the input data that is on the Hadoop Distributed File System and map function is applied to the input data. iShuffle finds the number of map output partitions and it places map output partition to nodes. Shufflers and the shuffle manager are the components used in iShuffle. The shuffler implements an operation which pushes the output data of mapping process to different nodes. Here, multiple servers are used to produce results in a short time. Data sets related to air pollution are collected. They are processed by the servers. This increases the efficiency and reduces the job completion time.

Keywords: *MapReduce, Hadoop, partitions, coupling, air pollution, datasets, servers, status.*

INTRODUCTION

Hadoop is an implementation of the MapReduce programming. It is used for processing huge amount of data in parallel. Job performed by Hadoop consists of two phases. Each phase contains multiple map tasks or reduce tasks. Map task converts a set of data into another set of data .Reduce task, which takes the output from a map task as an input and combines those data into a smaller set of data. Dealing with pollution data sets, air pollution is a major problem. The critical impact of air pollution on human health and environment lead the scientists to look for advance techniques for monitoring and predicting the urban air quality. Recent developments in data measurement techniques have led to collection of various types of data about air quality. Such data is extremely voluminous and must be processed at high velocity. In the proposed system, multiple servers are used to produce results. The results are found on the basis of execution time and efficiency. Data sets related to air pollution are collected considering vehicular emission and industrial pollutants. A certain processing time for this approach show that it is suitable to tackle large scale air pollution prediction problems.

REVIEW OF LITERATURE

In this study, a review on Big Data, air pollution datasets and MapReduce process is done.

In paper [1], as mentioned by YanfeiGuo ,Hadoop's implementation of the Map Reduce programming model pipelines the data processing and provides fault tolerance. Input data is partitioned and distributed as map tasks to individual cluster nodes for parallel execution. Map task splits the input data that is on the Hadoop Distributed File System and map function is applied to the input data. iShuffle finds the number of map output partitions and it places map output

Page 1-6 © MAT Journals 2018. All Rights Reserved

partition to nodes. Shufflers and the shuffle manager are the components used in iShuffle. The shuffler implements an operation which pushes the output data of mapping process to different nodes. The shuffler implements a shuffle-on-write operation that pushes the map output data to different nodes, every time such data is written to local disks. But there are some issues in this paper. Intensive communications between nodes can significantly delay job completion time. The coupling of shuffle and reduce phases in a reduce task presents challenges to attain high performance in multi-tenant environments.

In paper [2], as mentioned by Fan Zhang, Majd F. SakrMapReduce offers a distributed parallel processing framework. One can expect scalable performance in general MapReduce operations. There exist conditions by which the scalability will be limited. One of the key hurdles is the inability to remove unexpected or uninteresting elements from large data sets. Power law distribution implies that few nodes generate large amounts of data. In this article, they proposed a technique to take an arbitrary dataset and compute a power law distributed background model that bases its parameters on observed statistics. This model can be used to determine the suitability of using a power law or automatically identify high degree nodes for filtering. However, the issues are that, for Shuffle-intensive applications, whose execution time is dominated by the movement of the intermediate data, impacts the execution time more compared to the Map-intensive jobs. Another drawback observed is that the system performance decreases when the cluster size is increased.

In paper [3], as mentioned by HaripriyaAyyalasomayajula and Edgar Gabriel, MapReduce is a functional-programming paradigm in which the problem is decomposed into a set of map and reduce tasks .The user provides inputs as key/value pair to the map function that generates intermediate key/value pairs which are passed as inputs to the reduce function. The reduce function then combines all intermediate values associated with the same intermediate key.

Though successfully deployed in numerous industrial and academic settings, the MapReduce programming model has some major downsides. It does not map well to a number of algorithms and real-world problems and thus users often force-fit their algorithm to the MapReduce pattern and obtain poor performance. Hence the authors have used apache spark and compared the processed results of MapReduce and Spark models and have found that Spark performance is 20-25% better than MapReduce.

In paper [4], according to Lamari and Slaoui, the research is intensified in the direction of parallel clustering methods. Although there are many parallel programming models, the MapReduce paradigm is considered as the most prominent model. This paper shows a parallel design of a heuristic algorithm for clustering using MapReduce. In this heuristic algorithm, clustering is performed by partitioning categorical data according to a relational analysis procedure. The proposed design, called PMR-Transitive, is a single-scanned parameter-free heuristic which determines the number of clusters automatically. The experimental results on real-life and synthetic data sets demonstrate that PMR Transitive produces good quality results. In the near future, an extension to the PMR-Transitive heuristic algorithm is found to manage multiple data types and to keep its benefits. This method can be tested on a multi-nodes cluster environment to evaluate the performance.

Page 1-6 © MAT Journals 2018. All Rights Reserved

In paper [5], according to M.Kumar, G.Sikka, the use of MapReduce as a programming model has become pervasive for processing such wide range of Big Data Applications in cloud computing environment. Apache Hadoop is an implementation of MapReduce concept and it is used for processing, analyzing large scale data intensive applications in a scalable and fault tolerant manner. Several scheduling algorithms have been proposed for Hadoop considering various performance goals. In this work, a new scheme is introduced to aid the scheduler in identifying the nodes on which stragglers can be executed. From the performance analysis 27% improvement in terms of the overall execution time has been observed over Hadoop Fair Scheduler (HFS)

In paper [6], according to Zacheilas and Kalogeraki, the MapReducemodel has been utilized by major computing companies in order to perform large-scale data processing. However, the problem of efficiently scheduling MapReduce workloads in cluster environments can be challenging due to the observed trade-off between the performance and the corresponding cost. In this paper, they describe an approach for scheduling MapReduce workloads taking into consideration the performance/budget trade-off. Their approach has two contributions: (i) A novel scheduler for identifying near-optimal resource allocations and (ii) An automatic configuration of basic tasks parameters that allows us to further minimize the budget and execution time. This approach improves the performance of the workloads as much as 50%. Future work is to apply task adjustment should be always when the scheduler determines the per job slots allocations.

In paper [7], Alberto Fernández, Sara del Río have proposed “divide-and-conquer” a distributed procedure in a fault-tolerant way to adapt for commodity hardware. Researches have been conducted on imbalanced data classification, first to present the outcomes for imbalanced data classification and to analyze the behaviour of standard pre-processing techniques. This model has some issues in understanding the structure of these methods that are proposed to overcome the imbalanced data problem in Big Data and to address the challenges on imbalanced datasets

In paper [8], according to Xu-qing Chai, the resources of a Hadoop cluster is limited, the Hadoop cluster must select some specific tasks to allocate limited resources in order to get the maximal profit. In this paper, they gave maximal profit problem for a given candidate task set. A candidate set with a sequence is described and sequence-based scheduling strategy is proposed. In order to improve the efficiency of s sequence, some pruning strategies are applied. Finally, they proposed a timeout handling algorithm when some task ran in out of time. Experiments show that the total profit of the proposed algorithm is very close to the ideal maxima and isobviously bigger than related scheduling algorithms under different experimental settings.

In paper [9], according to Anandkrishna, Incremental processing is generally used where data is refreshed periodically to reflect small changes to the input dataset. To reduce the delay in computing the data again, methods which selectively compute the data is introduced. It incorporates the concept of Bloom Filter. Bloom filter is a space-efficient data structure, that can check if the data is modified or not. Traditional systems process whole data when a small percentage of data is changed. This consumes time and CPU clock cycles. To reduce the wastage of CPU clock cycles, a system is proposed.

Page 1-6 © MAT Journals 2018. All Rights Reserved

Using Bloom Filter helps to improve the performance of the system up to 17 % comparing with the existing system. Improvements in the fault tolerance is an area of development

In paper [10], according to Sreedhar, the clustering of datasets has become a challenging issue in the field of big data analytics. The K-means algorithm used for finding similarities between entities based on distance with small amount of data. This study deals with two approaches to cluster large datasets using MapReduce. The first approach, K-Means HadoopMapReduce which focuses on the MapReduce implementation of standard K-means .The second approach enhances the quality of clusters to produce clusters with maximum intra-cluster and minimum inter-cluster distances for large datasets. The results of the proposed approaches show significant improvements in the efficiency of clustering in terms of execution times. Experiments on K-means and proposed algorithms show that this approach is effective and efficient. Future work should improve the performance of MapReduce jobs for large datasets.

TITLE	AUTHOR	TECHNIQUE	ISSUE	RESULT
An insight into imbalanced Big Data classification: outcomes and challenges	Alberto Fernández, Sara del Río,Nitesh V. Chawla, Francisco Herrera	Divide-and-Conquer	In understanding the inner structure of these methodologies that have been proposed to overcome the imbalanced data problem in Big Data	Outcomes for imbalanced classification and analysed behaviour of standard pre-processing techniques
Improving Mapreduce for Incremental Processing Using Map Data Storage	AnandkrishnaR,Dhananjay Kumar	Bloom filter	Improvements in the fault tolerance by the distribution of the Bloom filter data is an area of development.	Improve the performance of the system up to 17 % when compared to existing system.
Tolhit – A Scheduling Algorithm for Hadoop Cluster	M. Brahmwar, M. Kumar and G. Sikka	Job scheduling in Hadoop cluster	The scheduler is also not efficient enough in identifying the slow tasks which protract the overall execution time.	27% improvement in terms of the overall execution time has been observed over Hadoop
Clustering large datasets using K-means modified inter and intra clustering (KM-I2C) in Hadoop	ChowdamSreedhar , NagulapallyKasiviswanath and PakantiChenna Reddy	K-means algorithm	Performance of map and reduce jobs to suit large datasets is low	36% improvement in the efficiency of clustering in terms of execution times
Empirical Discovery of Power-Law	Fan Zhang, Senior Member, IEEE, Majd F.	Power law distribution	Performance decreases when the cluster size is	High degree nodes for filtering are

Page 1-6 © MAT Journals 2018. All Rights Reserved

Distribution in MapReduce Scalability			increased.	identified and it is be scaled to work with big data.
Air Quality Simulations using Big Data Programming Models	HaripriyaAyyalasomayajula, Edgar Gabriel Peggy Lindner, Daniel Price	Apache Spark	It does not map well to a number of algorithms and real-world problems and thus users obtain poor performance	Spark performance is 20-25% better than MapReduce
A Pareto-based scheduler for exploring cost-performance trade-offs for MapReduce workloads	Nikos Zacheilas and VanaKalogeraki	Pareto-based scheduler	Task adjustment should be always when the scheduler determines the per job slots allocations	Workload performance is improved by 50%
Profit-oriented task scheduling algorithm in Hadoop cluster	Xu-qing Chai, Yong-liang Dong and Jun-fei Li	pruning strategy, timeout handling strategy	Designing pruning strategies for every data set is difficult	Total profit is very close to the ideal maxima (90%)
iShuffle: Improving Hadoop Performance with Shuffle-on-Write	YanfeiGuo, JiaRao, Dazhao Cheng, and Xiaobo Zhou,	iShuffle	Communication delay between nodes reduces the performance	iShuffle reduces job completion time by as much as 29.6% and 34% in single-user and multi-user clusters, respectively.
Clustering categorical data based on the relational analysis approach and MapReduce	YasmineLamariand Said ChahSlaoui	Relational analysis approach	To manage multiple data types and to treat outliers while keeping its characteristics and benefits	The experimental results on real-life and synthetic data sets demonstrate that PMR Transitive produces good quality results

CONCLUSION AND FUTURE WORK

From the review of various journals, it is concluded that, the efficiency and performance of the MapReduce concept is improved. But this improvement will not be sufficient when the data grows into a larger size. So, multiple servers are used for processing the data and for reducing the job completion time. In the near future, there is still more work to be done to optimise the MapReduce to support fast memory processing.

REFERENCES

1. YanfeiGuo, Member, JiaRao, Member, Dazhao Cheng, and Xiaobo Zhou "iShuffle: Improving Hadoop Performance with Shuffle-on-Write", *IEEE Transactions on Parallel and*

Page 1-6 © MAT Journals 2018. All Rights Reserved

Distributed Systems, VOL. 28, NO. 6, JUNE 2017
2. Fan Zhang, Majd F. Sakr Kai Hwang, and Samee U. Khan“ Empirical Discovery of Power-Law Distribution in MapReduce Scalability”, *2017 IEEE Transactions on Cloud Computing*
3. HaripriyaAyyalasomayajula, Edgar Gabriel" Air Quality Simulations using Big Data Programming Models", *2016 IEEE Second International Conference on Big Data Computing Service and Applications*
4. Lamari and Slaoui*J Big Data (2017)*" Clustering categorical data based on the relational analysis approach and MapReduce", *Journal of Big Data*
5. M. Brahmwar, M. Kumar and G. Sikka *(2016)*, "Tolhit – A Scheduling Algorithm for Hadoop Cluster ", Twelfth International Multi-Conference on Information Processing
6. Zacheilas and Kalogeraki*EURASIP Journal on Embedded Systems* (2017), " A Pareto-based scheduler for exploring cost-performance trade-offs for MapReduce workloads", *EURASIP Journal on Embedded Systems*
7. Alberto Fernández, Sara del Río ,Nitesh V. Chawla , Francisco Herrera, “An insight into imbalanced Big Data classification: outcomes and challenges”, *Complex Intell. Syst. (2017)*
8. Xu-qing Chai, Yong-liang Dong and Jun-fei Li *(2016)*,“ Profit-oriented task scheduling algorithm in Hadoop cluster”, Journal on Embedded Systems
9. Anandkrishna, R,Dhananjay Kumar (2016), “Improving Mapreduce for Incremental Processing Using Map Data Storage”, 4th International Conference on Recent Trends in Computer Science & Engineering
10. Sreedhar*et al. J Big Data (2017)*, "Clustering large datasets using K-means modified inter and intra clustering (KM-I2C) in Hadoop", *Journal of Big Data*

Page 1-6 © MAT Journals 2018. All Rights Reserved