International Journal of Engineering and Techniques - Volume 3 Issue 5, Sep - Oct 2017

RESEARCH ARTICLE OPEN ACCESS

MAPREDUCE WORKLOAD FOR DYNAMIC JOB ORDERING AND SLOT CONFIGURATION

D.sowmya[1], Dr.S.Madhavi[2] ,K.Vijaya kumar[3]

1(C.S.E, PVP SIDDHARATHA INSTITUTE OF TECHNOLOGY, and VIJAYAWADA)

2 (C.S.E,professor PVP SIDDHARATHA INSTITUTE OF TECHNOLOGY, and VIJAYAWADA)

3(C.S.E, asstPVP SIDDHARATHA INSTITUTE OF TECHNOLOGY, and VIJAYAWADA)

Abstract

Todays world the amount of data being generated is growing exponentially and use of internet is also increasing it leads to handle lots of data by internet service providers. MapReduce is one of the good solutions for implementing large scale distributed data application[21]. A MapReduce workload generally contains a set of jobs, each of job consists of multiple map and reduce tasks[21]. Map task executed before reduce task and map tasks can only run in map slot and reduce tasks can only run in reduce slot[21]. Due to that different job executions orders and map/reduce slot configurations for a MapReduce workload have different performance metrics and different system utilization[21]. Make span and total completion time are two key performance metrics. In [21] they proposes two algorithm for these two key metrics, The first class of algorithms mainly focuses on the job ordering optimization for a MapReduce workload under given slot configuration and the second class of algorithms perform optimizationfor slot configuration for a MapReduce workload.

Keywords — **MapReduce, Hadoop, Flow-shops, Scheduling algorithm, Job ordering.**

I. INTRODUCTION

MAPREDUCE is a widely used computing model for large scale data processing in cloud computing. A MapReduce job consists of a set of map and reduces tasks, where reduce tasks are performed after the[29] map tasks. Hadoop [21], an open source implementation of MapReduce, has been deployed in large clusters containing thousands of machines by companies such as Amazon and Facebook. In th at clusterand data center environments, MapReduce and Hadoop are used to support batch processing for jobs submitted from multiple users (i.e., MapReduce workloads). Despite many research efforts devoted to improvingthe performance of a single MapReduce job (e.g., [3], [21],[11]), there is relatively little [29]attention paid to the system performance of MapReduce workloads. Therefore, in [21] they tries to improve [29][30]the performance of MapReduce workloads.Makespan and total completion time (TCT) are two key performance metrics. Generally, makespan is defined as the time period since the start of the first job until the completion of the last job for a set of jobs[21]. It considers[31] the computation time of jobs and is often used to measure the performance and utilization efficiency of a system[22]. In contrast, total completion time is referred to as the sum of completed time periods for all jobs since the start of the first job[21]. It is a generalized makespan with queuing time (i.e., waiting time) included. We can use it to measure the satisfaction of the system from a single job's perspective through dividing[21] In [21]they target at one subset of production MapReduce workloads that consist of aset of independent jobs (e.g., each of jobs processes distinct data sets with no dependency between each other) with different approaches. For dependent jobs(i.e.MapReduce workflow), one MapReduce can only start only when its previous dependent jobs finish the computation subject to the input-output data dependency[21]. In [21]contrast, for independent jobs, there is an overlap computationbetween two jobs, i.e., when the current job completes its map-phase computation and starts its reduce-phase computation, the next job can begin to perform its map-phase computation

II. LITERATURE REVIEW

Wolf et al. [2] implemented flexible scheduling all location scheme with Hadoop fair scheduler. A primary concern is to optimize scheduling theory metrics, response time, makespan, stretch, and Service Level Agreement[21]. They proposed penalty function for measurement of job [21]completion time, epoch scheduling for partitioning time, moldable scheduling for job parallelization, and malleable scheduling for different interval parallelization. Dean et al. 2008 [1] have discussed MapReduceprogramming model. The [21]MapReduce model performs operations using the map and reduces functions. Map function gets input from user documents. It generates intermediate key/value for reducing function. It further processes intermediate key/value pairs and provide

output key/value pairs. At an entry level, [23] MapReduce programming model provided the best data processing results. Currently, it needs to process the large volume of data. So it provides some consequences whileprocessing and generating data sets[21]. It takes much execution time for task initialization, taskCoordination[21] , and task scheduling. Parallel data processing may lead to inefficient task execution and low resource utilization[21]. Verma et al. [3] proposed two algorithms for makespan optimization. First is a greedy algorithm job ordering method based on Johnson's Rule. Another is a heuristic algorithm called Balanced Pool. They have introduced a simple abstraction where each M apReduce job is representedas a pair[32] of map and reduce stag e duration. The Johnson algorithm was designed for building an optimal job schedule. This[21] framework evaluates the perfo mance benefits of the constructed schedule through an ex tensive set of simulations over a variety of realistic workloa ds[32]. It measures how many numbers of slots required for scheduling the slots dynamically with aparticular job deadline[21]. Tang et al. [4] have proposed three techniques to improve MapReduce performance. First technique is Dynamic Hadoop Slot Allocation. They categorized[21] utilized slot into the busy slot and idle slot respectively. The primary[21] concern is to increase the number of the busy slots and decrease number of idle slots. DHSA observes idle map and reduce slots[21]. Dynamic Hadoop Slot Allocation allocate the task only to the u nallocated map slots and due to Speculative Execution[21] Performance Balancing provides performance upgrade for a batch of jobs[21].

It gives the highest priority to failed tasks and next level priority to pending tasks. Due to slot prescheduling it improves the performance of slot utilization[21][23]. Tang, Lee and He[5] have proposed A Dynamic Slot Allocation Optimization Framework for improving the performance for a single job but at the expense of the cluster efficiency[21]. They proposed Hazardous Execution Performance Balancing technique for balancing the performance tradeoff between a single job and a batch of jobs[21] . Slot Pre Scheduling is the new tech nique and that can improve the data locality but with no impact on fairness[21]. Finally, integrating these two techniques, new technique is implemented called DynamicMR that can improve the performance of MapReduce w orkloads. Tang, Lee and He[6] have proposed MROrder : Flexible Job Ordering technique whichoptimizes the job order for online MapReduce workloads[21]. MROrder is designed to be flexible for different optimization metrics, e.g., makespan and total completion time. Kyparisis and Koulamas [7][24] considered a scheduling problem in two(stage hybrid flow shop, where the first stage consists of two machines formed an open shop and the other stage has only one machine. The main objective is to minimize the makespan, i.e., the maximum complet ion time of all jobs. They first show the problem is NP(hardin the strong sense, then we present two heuristics to sol ve the problem. Computational experiments show that the combined algorithm of the two heuristics performs well on randomly generated problem instances[26]. Agrawal et al. [8] haveproposed a method called Scheduling s hared scans of large data files and it is used to maximize scan sharing by grouping MapReduce[21] jobs into batches so that sequential scans of large files are shared among man y simultaneousjobs where it is possible. MRShare [9] is a sharing framework and it gives three possible work(sharing opportunities, they are scan sh aring, mapped outputs sharing, and Map function sh aring across multiple MapReduce jobs. Due to sharing it avoides[21][23] the redundant work and saves the processing tim e. Herodotou et al. [10] provideHadoop configuration opti mization policy. Starfish is a self(tuning framework and it canadjust the Hadoop's configuration automaticallyfor a MapReduce job[21]. Based on the cost(based model and sampling technique the utilization of Hadoop cluster can b e maximized and it also proposes a [23]system named El astisizerforcluster(sizing optimization and MapReduce job(level parameter configurations optimization, on the cloud platform, to meet desired require ments on execution time and cost for a given workload, based on a careful mix of job[21] profiling,estimation using black(box and whitebox models and simulation).

III. SYSTEM ARCHITECTURE

3.1 Slot allocation and Slot pre-scheduling process

In this module,we are going to perform two processes. Slot allocation Slot pre-scheduling process[24]. In this slot allocation process[21],we are going allocate the slot based on dynamic Hadoop slot allocation optimization mechanism.In the slot pre-scheduling process we are going to improve the data locality[24]. Slot Pre-Scheduling[24] technique that can improve the data locality while having no negative impact on the fairness of Map-Reduce jobs. Some idle slots which cannot be allocated due to the load balancing constraint during runtime, we can pre-allocate those slots of thenode to jobs to maximize the data locality[27]

3.2 Speculative Execution Performance Balancing

When a node has an idle map slot, we should choose pending map tasks firstbefore looking for speculative map tasks for a batch of jobs[28] .Hadoop Slot is executed for determining the path for performing the MapReduce job. After this,the Speculative based process starts to execute the determined optimized Multi-execution path[21]. Executing individual MapReduce jobs in each datacenter on corresponding inputs and thenaggregating results is defined asaMULTI execution path. This path used to execute the jobs effectively.

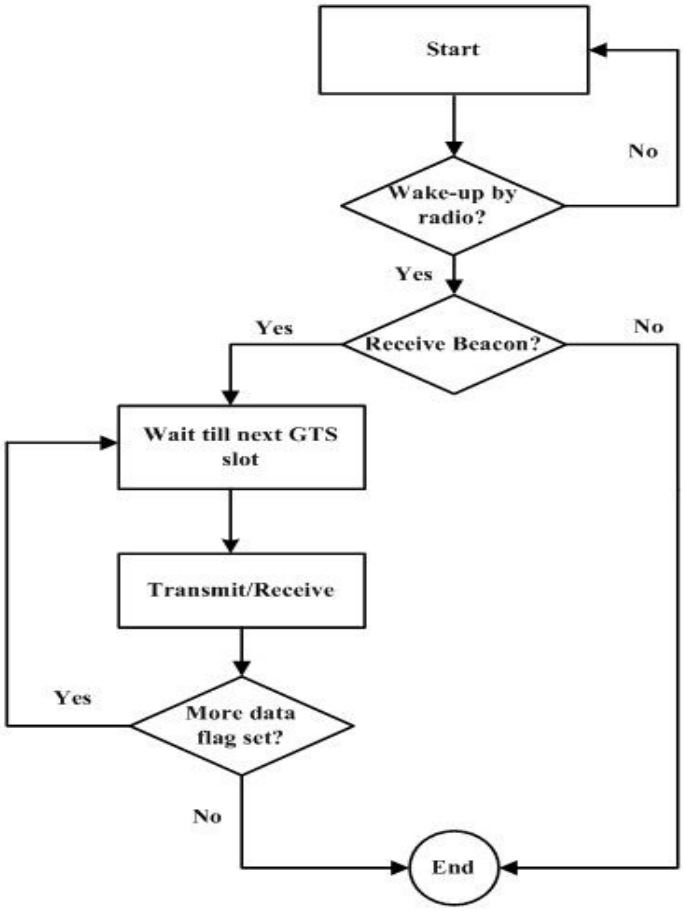

fig:system architure

IV. APPLICATIONS

1. Social Media: The Large data is generated from the social media platforms such as YouTube, Facebook, Twitter, LinkedIn, and Flickr. The amount of DATA being uploaded to the internet is rapidly increasing, with Facebook users uploading over 2.5 billion new Data every month. It can be used to improve applications performance by greatly reducing the file size and network bandwidth required to display your application.
2. Business Applications: online shopping application where the every item has data is shown. Company's data and scan copies of various documents.
3. Satellite images: This includes weather data or the data that the government captures in its satellite surveillance imagery.
4. Photographs and video: This includes security, surveillance, and traffic video.

V. CONCLUSION

The focuses on the job ordering and map/reduce slot configuration issues for MapReduce production workloads that run periodically in a data warehouse[21], [31][23] where the average execution time of map/reduce tasks for a MapReduce job can be profiled from the history run, under the FIFO scheduling in a Hadoop cluster.

Two performance metrics are considered, i.e., makespan and total completion time.[25] We first focus on the makespan[21].

We do a job ordering optimization algorithm and map/reduce slot configuration optimization algorithm. The total completion time can be poor subject to getting the optimal makespan, therefore,they proposed a newjob ordering algorithm and a map/reduce slot configuration algorithm to minimize the make span and [21] total completion time together. The theoretical analysis is also g greedy iven for our proposed heuristic algorithms, icluding approximation ratio, upper and lower bounds on makespan. Finally, we validate the effectiveness of algorithms total completion time together. The theoretical analysis is also g greedy iven for our proposed heuristic algorithms, including approximation ratio, upper and lower bounds on makespan. Finally, we validate the effectiveness of algorithms

VI. REFERENCES

[1] Amazon ec2 [Online]. Available: http://aws.amazon.com/ec2,2015.
[2] Apache hadoop [Online]. Available: http://hadoop.apache.org,2015.
[3] Howmanymapsandreduces [Online]. Available:http://wiki.apache.org/hadoop/HowMany MapsAndReduces, 2014.
[4] Lognormal distribution [Online]. Available:http://en.wikipedia.org/wiki/Log-normal_distribution, 2015.
[5] The scheduling problem [Online]. Available:http://riot.ieor.berkeley.edu/Applications/Scheduling/algorithms.html,1999.
[6] S. R. Hejazi and S. Saghafian, "Flowshop-scheduling problems with makespan criterion: A review,"Int. J. Production Res., vol. 43,no. 14, pp. 2895–2929, 2005.
[7] S. Agarwal, S. Kandula, N. Bruno, M. C. Wu, I. Stoica, and J.Zhou, "Reoptimizing data-parallel computing," inProc. 9th USE-NIX Conf. Netw. Syst. Design Implementation, 2012, p. 21.
[8] P. Agrawal, D. Kifer, and C. Olston, "Scheduling shared scans oflarge datafiles,"Proc. VLDB Endow., vol. 1, no. 1, pp. 958–969,Aug. 2008.
[9] W. Cirne and F. Berman, "When the herd is smart: Aggregate behavior in the selection of job request,"IEEE Trans. Parallel Dis-trib. Syst., vol. 14, no. 2, pp. 181–192, Feb. 2003.
[10] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy,and R. Sears, "Mapreduce online," in Proc.

7thUSENIX Conf.Netw. Syst. Design Implementation, 2010, p. 21.
[11] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Oper. Syst. Design Implementation
, 2004, vol. 6, p. 10.
[12] J. Dittrich, J.-A.-Quian e Ruiz, A. Jindal, Y. Kargin, V. Setty, and J.Schad, "adoop++: Making a yellow elephant run like a cheetah(without it even noticing)," Proc. VLDB Endowment, vol. 3,nos. 1–2, pp. 515–529, Sep. 2010.
[13] P.-F. Dutot, L. Eyraud, G. Mouni
e, and D. Trystram, "Bi-criteria algorithm for scheduling jobs on cluster platforms," inProc. 16th Annu. ACM Symp.Parallelism Algorithms Archit., 2004, pp. 125–132.
[14] P.-F. Dutot, G.Mounie, and D. n
Trystram,"Scheduling paralleltasks: Approximation algorithms," inHandbo ok of Scheduling:Algorithms, Models, and Performance Analysis, J. T. Leung, Ed. BocaRaton, FL, USA: CRC Press, ch. 26, pp. 26-1–26-24.
[15] A. Floratou, J. M. Patel, E. J. Shekita, and S. Tata, "Columnoriented storageechniques for mapreduce,"Proc. VLDB Endowment, vol. 4, no. 7, pp. 419–429, Apr. 2011.
[16] J. Gupta, A. Hariri, and C. Potts, "Scheduling a two-stage hybrid flow shop with parallel machines at the first stage,"
Ann. Oper.Res., vol. 69, pp. 171–191,
1997.
[17] J. N. D. Gupta, "Two-stage, hybrid flowshop scheduling problem,"J. Oper. Res. Soc., vol. 39, no. 4, pp. 359–364, 1988.
[18] H. Herodotou and S. Babu,"Profiling, what if analysis, and costbased optimization of mapreduce programs,"Proc. VLDB Endowment, vol. 4, no. 11, pp. 1111–1122,
2011.
[19] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin,and S. Babu, "Starfish: A self-tuning system for big data analycs," inProc. 5th Conf. Innovative Data Syst. Res., 2011, pp. 261272.
[20] S. Ibrahim, H. Jin, L. Lu, B. He, and S.Wu, "Adaptive disk I/O
scheduling for mapreduce in virtualized environment," intiProc.Int. Conf. Parallel Process., Sep. 2011, pp. 335–34
[21]. Sonali S. Birajadar, B. M. Patil, V. M. Chandode " Dynamic Job Ordering and Slot Configuration for MapReduce Workloads ",International Journal of Computer Applications (097 5 – 8887) Volume 173 – No.7, September 2017
[22] Vinayak Kadam,Rutuja Aughad,Priyanka ,aikwad ,Jidnyasa khanore,Pragati Naykodi" Optimization of Slot and Map Reduce Workload"IJARCCEVol. 5, Issue 12 Dec 2016
".
[23]. Sameer M. Khupse , Himanshu U Joshi
," Survey on Hadoop Performance Using Dynamic Job Ordering and Slot Configuration Approach Using Mapreduce "V ol. 4, Issue 12, December 2016 IJIRCCE
[24] MsUshapriyaM.,2 Ms.Abarna N
,"DISTRIBUTED FEATURE SELECTION FOR EFFICIENT ECONOMIC BIG DATA ANALYSIS
", IRJET, Volume: 04 Issue: 08 | Aug 2017

[25]. Edukoju Hareesh, N. Naveen Kumar,"Minimizing The Makes Pan Of Multiple MaprEduce Jobs Through Job Ordering Techniq eOlumev Iii/Issue5/Jun2017, International Journal Of Professional Engineering Studies
[26] Jian-Ming DongJue-Liang Huemail Author
Yong Chen Minimizing makespan in a two-stage hybrid flow shop scheduling problem with open shop in one stage Applied Mathematics-A Journal of Chinese UniversitiesSeptember 2013, Volume 28, Issue 3, pp 358–368
[27] B.PARAMESWARI, V.GOMATHIDynamic Hadoop Slot Allocation for MapReduceClusters in CloudNTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGYVOLUME 6 ISSUE 1 JUNE 2015 ISSN: 2349 9303

[28] Shanjiang Tang, Bu-Sung Lee, Bingsheng He," DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters "IEEE Transactions on Cloud Computing2014 vol. 2Issue No. 03 – July-Sept.
[29].Prathamesh Chaudhari,Gaurav S. Salve,Nilesh Ghadge
"Dynamic job ordering and slot configurations For MapReduce workloads",ISSN: 2454-132X Impact factor: 4.295
(Volume3, Issue2)ijariit
[30]"Optimization of Slot andMap Reduce Workload"Vinayak Kadam, Rutuja Aughad, Priyanka Gaikwa, Jidnyasa khanor, Pragati Naykod.ijjarcce
[31]"Enrichment of time efficiency for MapReduce Worloads through Job ordering "
Tanmayi Nagale & Rahul Kapse .Vol-3, Issue-5, 2017 ,ijirISSN: 2454-1362

[32]"Two Sides of a Coin: Optimizing the Schedule of MapReduce Jobs to Minimize Their Makespan and Improve Cluster Performance"Abhishek Verma,Ludmila Cherkasova,Roy H. Campbell