International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

An Efficient Dynamic Job Ordering and Slot Configuration for Minimizing the MakespanOf MapReduce Jobs

Dr. Srikanth Thodeti
Professor , Dept: C.S.E.
Jayamukhi Institute of Technological Sciences, Narsampet, Warangal,Telangana, India
dr.srikanththodeti@gmail.com

Abstract:*MapReduce is a well-known parallel computing paradigm for massive data processing in clusters and data facilities.It is observed that different job execution orders and MapReduce slot configurations for a MapReduce workload have the enormously extraordinary efficiency related to the makespan, complete completion time, process utilization and otherperformance metrics. This paper proposes two classes of algorithms to diminish the lifespan and the total completion time for an offline MapReduce workload. Our first class of algorithms focuses on the job ordering optimization for a MapReduceworkload below a given map/reduce slot configuration. In distinction, our 2nd type of algorithms considers the scenariothat we will participate in optimization for map/reduce down slot configuration for a MapReduce workload.*

Keywords- MapReduce, Hadoop, Flow-shops, Scheduling algorithm, Job ordering

I. INTRODUCTION

A MapReduce job consists of a set of map and reducetasks, where reduce tasksare performed after the maptasks. Hadoop [2], an open source implementation ofMapReduce, has been deployed in largeclusters containingthousands of machines by companies such as Amazon andFacebook. Make span and total completion time are twokey performance metrics. Generally, make span is definedas the timeperiod since the start of the first job until thecompletion ofthe last job for a set of jobs. It considers thecomputationtime of jobs and is often used to measure theperformance andutilization efficiency of a system. Incontrast, total completiontime is referred to as the sum ofcompleted time periods for alljobs since the start of thefirst job. It is a generalized make span with queuing time(i.e., waiting time) included. We can use itto measure thesatisfaction to the system from a single job'sperspectivethrough dividing the total completion time by thenumberof jobs (i.e., average completion time).

Inthose cluster and data center environments, MapReduceand Hadoop are used to support batch processing for jobssubmitted from multiple users (i.e., MapReduce workloads). Despite many research efforts devoted to improvethe performance of a single MapReduce job. There are two key performance metricsi.e. Makespan and total completion time (TCT) and we aim tooptimize these matrics. Generally, make span is defined as thetimeperiod since the start of the first job until the completionof the last job for a set of jobs. It considers the computationtime of jobs and is often used to measure the performance andutilization efficiency of a system. In contrast, total completiontime is referred to as the sum of completed time periods for alljobs since the start of the first job. It is a generalizedmakespan with queuing time (i.e., waiting time) included. Wecan use it to measure the satisfaction to the system from asingle job's perspective through dividing the total completiontime by the number of jobs (i.e., average completion time).Therefore, in this paper, we aim to optimize these two metricsthe number of jobs (i.e., average completion time). Therefore,in this paper, we aim to optimize these two metrics.

Objectives:

• To improve the performance for MapReduce workloadswith job ordering and slot configuration optimizationapproaches.

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

• Propose slot configuration algorithms for make span andtotal completion time.

• Perform extensive experiments to validate theeffectiveness of proposed algorithms and theoreticalresults

II. RELATED WORK

The important problems are that we define and more issue in scheduling technique of MapReduce, the scheduling isone of the most critical criteria of MapReduce. There are many algorithm can address these problem with differenttechniques and methods. Some of them focus on dynamic slot allocation and straggler problem for speculative execution.Also many of them have been design deadline constrain to minimizing the total job completion time. In this section wedescribe some of these algorithm techniques.

Wolf et al. [2] implemented flexible scheduling allocationscheme with Hadoop fair scheduler. A primary concern is tooptimize scheduling theory metrics, response time, makespan,stretch, and Service Level Agreement. They proposed penaltyfunction for measurement of job completion time, epochscheduling for partitioning time, moldable scheduling for jobparallelization, and malleable scheduling for different intervalparallelization.

Dean et al. 2008 [1] have discussed MapReduce programmingmodel. The MapReduce model performs operations using themap and reduces functions. Map function gets input from userdocuments. It generates intermediate key/value for reducingfunction. It further processes intermediate key/value pairs andprovide output key/value pairs. At an entry level, MapReduceprogramming model provided the best data processing results.Currently, it needs to process the large volume of data. So itprovides some consequences while processing and generatingdata sets. It takes much execution time for task initialization,task coordination, and task scheduling. Parallel dataprocessing may lead to inefficient task execution and lowresource utilization.

Verma et al. [3] proposed two algorithms for makespanoptimization. First is a greedy algorithm job ordering methodbased on Johnson's Rule. Another is a heuristic algorithmcalled BalancedPool. They have introduced a simpleabstraction where each MapReduce job is represented as apair of map and reduce stage duration. The Johnson algorithmwas designed for building an optimal job schedule. Thisframework evaluates the performance benefits of theconstructed schedule through an extensive set of simulationsover a variety of realistic workloads. It measures how manynumbers of slots required for scheduling the slots dynamicallywith a particular job deadline.

Tang et al. [4] have proposed three techniques to improveMapReduce perormance. First technique is Dynamic HadoopSlot Allocation. They categorized utilized slot into the busyslot and idle slot respectively. The primary concern is toincrease the number of the busy slots and decrease number ofidle slots. DHSA observes idle map and reduce slots.Dynamic Hadoop Slot Allocation allocate the task only to theunallocated map slots and due to Speculative ExecutionPerformance Balancing provides performance upgrade for abatch of jobs. It gives the highest priority to failed tasks andnext level priority to pending tasks. Due to slot preschedulingit improves the performance of slot utilization.

Tang, Lee and He [5] have proposed DynamicMR: A Dynamic Slot Allocation Optimization Framework forimproving the performance for a single job but at the expenseof the cluster efficiency. They proposed Hazardous ExecutionPerformance Balancing technique for balancing theperformance tradeoff between a single job and a batch of jobs.Slot PreScheduling is the new technique and that can improvethe data locality but with no impact on fairness. Finally,integrating these two techniques, new technique isimplemented called DynamicMR that can improve theperformance of MapReduce workloads.Tang, Lee and He [6] have proposed MROrder: Flexible JobOrdering technique which optimizes the job order for onlineMapReduce workloads. MROrder is designed to be flexiblefor

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

different optimization metrics, e.g., makespan and totalcompletion time

The computing Resources are considering into map and reduce slots, which are primary computing units andstatically configure by administrator in advance. A MapReduce job execution has two main features: first, the slotallocation constrains that map slots allocate to map slots and reduce slots can be allocated to reduce task. There isimmense different performance and system utilization for a MapReduce over differ slot configurations. So if we use eachand every slot in both slot according to needs of node which affects the system utilization and performance. For that use aDynamic Hadoop Slot Allocation (DHSA) [7] technique to increase the slot utilization for MapReduce.Straggler Problem occur because of unavoidable run-time contention for processor, memory, network bandwidth andalso other resources causing great affect on delay of the whole job. For that use a Speculative Execution PerformanceBalancing (SEPB) [7] to balance the utilization for single job as well as multiple jobs.

Data Locality increase slot utilization efficiency and performance achieve great output for MapReduce workloads.After all, there is often struck between fairness and data optimization in a shared cluster among many users. For that usea Slot PreScheduling technique to achieve great significant data locality at the expense of the load balance nodes.User having specific job deadline so deadline are most important requirement which can improve the performance.MapReduce Task Scheduling algorithm for Deadline Constrains (MTSD) [8] has main goal on user's deadline constraintsproblem.

III. PROPOSED WORK

(Algorithm1) Slot allocation and Slot pre-scheduling process:

In this module, we are going to perform two processes. Slot allocation Slot pre- scheduling process.In this slot allocation process, 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. Slot Pre-Scheduling technique that can improve thedata 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.

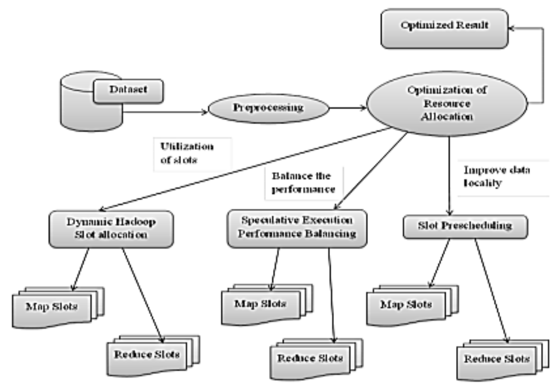

Fig.1 Overview of Dynamic Job Ordering

(Algorithm2) Speculative Execution Performance Balancing:

When a node has an idle map slot, we should choose pending map tasks first before looking for speculative map tasks for a batch ofjobs.Hadoop Slot is executed for determining the path for performing the MapReduce job. After this, the Speculative based process startsto execute the determined optimized Multi-execution path.Executing individual MapReduce jobs in each datacenter on corresponding inputs and then aggregating results is defined as aMULTI execution path. This path used to execute the jobs effectively.

IV. CONCLUSION

Dynamic slot configuration is one of the significantfactorswhile processing a large data set with MapReducemodel. It optimizes the enactment ofMapReduceframework. Each job can be scheduled usingany one of the scheduling policiesby the job tracker.Thetask managerswhich are presentin the task trackerallocateslots to jobs.As of the inspected paper,it isconcluded to prefer a dynamic slot allocation strategy thatincludes active jobs workload estimation, optimal slotassignment, and scheduling policy.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified dataprocessing on large clusters," in Proc. 6th Conf. Symp.Oper. Syst. Design Implementation, 2004

[2] J. Wolf, D. Rajan, K. Hildrum, R. Khandekar, V. Kumar,S. Parekh, K.-L. Wu, and A. balmin, "Flex: A slotallocation scheduling optimizer for mapreduceworkloads," in Proc. ACM/IFIP/USENIX 11th Int. Conf.Middleware, 2010

[3] A. Verma, L. Cherkasova, and R. H. Campbell, "Twosides of a coin: Optimizing the schedule of mapreducejobs to minimize their makespan and improve clusterperformance," in Proc. IEEE 20th Int. Symp. Model.,Anal. Simul.Comput.Telecommun. Syst.,2012.

[4] S. Tang, B.-S.Lee, and B. He, "Dynamic slot allocationtechnique for mapreduce clusters," in Proc. IEEE Int.Conf. Cluster Comput.,Sep. 2013, pp. 1–8.

[5] S. Tang, B.-S.Lee, and B. He, "Dynamicmr: A dynamicslot allocation optimization framework for mapreduceclusters," IEEE Trans.Cloud Comput., vol. 2, no. 3, pp.333–347, Jul. 2014.

[6] S. Tang, B.-S.Lee, and B. He,, "Mrorder: Flexible jobordering optimization for online mapreduce workloads,"in Proc. 19th Int. Conf. Parallel Process., 2013, pp. 291–304

[7] Shanjiang Tang, Bu-Sung Lee, Bingsheng He, "DynamicMR: A Dynamic Slot Allocation Optimization Frameworkfor MapReduce Clusters", IEEE Transaction on Cloud Computing, 2014

[8] Zhuo Tang · Junqing Zhou ·Kenli Li · Ruixuan Li, "A MapReduce task scheduling algorithm for deadlineconstraints" Springer Science + Business Media New York, 2012.

[9] W. Cirne and F. Berman, "When the herd is smart: Aggregatebehavior in the selection of job request," IEEE Trans. Parallel Distrib. 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. 7th USENIX 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. DesignImplementation, 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-criteriaalgorithm for scheduling jobs on cluster platforms," in Proc. 16thAnnu.ACM Symp. Parallelism Algorithms Archit., 2004, pp. 125–132.

[14] P.-F. Dutot, G.Mounie, and D. Trystram,"Scheduling paralleltasks: Approximation algorithms," in Handbo 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 storage techniques 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 hybridflow 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

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

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 analytics," in Proc. 5th Conf. Innovative Data Syst. Res., 2011, pp. 261–272.

[20] S. Ibrahim, H. Jin, L. Lu, B. He, and S. Wu, "Adaptive disk I/Oscheduling for mapreduce in virtualized environment," in Proc.Int. Conf. Parallel Process., Sep. 2011, pp. 335–344.

BIODATA

Author

Dr. SRIKANTH THODETI received his Master of Computer Applications (M.C.A.) from Kakatiya University Warangal, M.Tech.(C.S.E.) from J.N.T.U. Hyderabad and Ph.D. (Computer Science) from Dravidian University, A.P.. He Published 2 ISBN books & 16 papers in various reputed International Journals and National / International Conferences.