



Efficient Hybrid Framework for Parallel Resource and Task Scheduling in the Map Reduce Programming

S. Valarmathi, S. Hemalatha

Department of Computer Science and Engineering
Sri Shakthi Institute of Engineering and Technology, Tamilnadu,
India

Abstract- *The applications like data intensive of large-scale is run by data centers based on Map Reduce implementation. MapReduce runs on large clusters needs vast amounts of energy which parallel increase cost of the data centers, Energy reduction has to be incorporated into the map reduce implementation to improve the efficiency of the data center. In order to achieve the good performance, model a framework based on Service Level Agreement (SLA) and its strategies. The parallel scheduler is modeled for resource and task using particle swarm optimization to manage the assignments of map and reduce task. The Resource management is carried to manage a resource slot which reduces the consumption of energy when running the application achieves optimal schedules. Performance evaluation of the frameworks is compared with state of approaches which concludes that the proposed framework outperforms in terms of efficiency and effectiveness.*

Keywords: *Resource scheduling, task scheduling, map reduce framework*

I. INTRODUCTION

Industries of today, reliant increasingly on large scale data analytics for their business decisions of day to day life. This reliant on large scale data decision motivated the development of MapReduce. MapReduce is a model of program for data-intensive computation which becomes popular in recent years. MapReduce is an implementation on cluster for processing and producing huge data sets with a parallel, distributed algorithm. Together MapReduce is a framework for problems to be parallelizable process across large datasets with the help of huge number of computers (nodes), collectively known as a cluster or a grid. In MapReduce the jobs is partitioned as small tasks like map and reduce tasks and execute parallel among large number of machines (nodes).

In today's companies, the MapReduce frameworks performance and efficiency have become critical to their success. The map and reduce tasks is collectively called as job which is scheduled concurrently on multiple machines which reduces the running time of a job. The job scheduler is the central component of MapReduce system. Job completion time is minimized by spanning the jobs of map and reduces tasks, which is done by job scheduler. The key-value block is give as input to map tasks where the key-value pair is stored in file system of distributed environment, the map tasks performs a user-specified map function and the output will be the intermediate key-value pairs. Subsequently, the reduce task collects and apply the reduce function of user-specified on a collected key-value pairs for final output. From task to task and from job to job the resource consumptions run time varies. Several recent studies have reported that production workloads often have diverse utilization profiles and performance requirements [2], [3]. Hadoop is the most famous frame for implementing the MapReduce. The Hadoop cluster consists of commodity machines of large number where one node acts as a master and other nodes as slaves. The resource manager is runs on a master node which is responsible for task scheduling on the slave nodes. The local node manager runs on slave node which is responsible for allocating and launching resources for each task. For this above process the Java Virtual Machine(JVM) is launched by the task tracker. The JVM executes the map or reduce function.

II. BACKGROUND

Map reduce programming model will be breaking down each job as small map tasks and reduce tasks and executing in parallel across a large number of machines, MapReduce can significantly reduce the running time of data-intensive jobs. However, despite recent efforts toward designing resource-efficient MapReduce schedulers to offer sub-optimal job performance in the task management due to nature that tasks can have highly varying resource requirements during their lifetime, which makes it difficult for task-level schedulers to effectively utilize available resources to reduce job execution time. PRISM, a fine-grained resource-aware MapReduce scheduler that divides tasks into phases, where each phase has a constant resource usage profile, and performs scheduling at the phase level which demonstrate the importance of phase-level scheduling by showing the resource usage variability within the lifetime of a task using a wide-range of MapReduce jobs. A phase-level scheduling algorithm will improve execution parallelism and resource utilization without introducing stragglers[1].

III. LITERATURE REVIEW

A. Two Level Scheduler

The Quality of Service's (QoS) efficiency improvement is considered, which is based on meta-scheduler and Backfill strategy based light weight Virtual Machine Scheduler[4] for jobs distribution, specifically conservative backfilling is used. The proposed two- level scheduler focuses on optimizing the overall resource utilization and guaranteeing increased performance of the application. The jobs at the higher level are executed by the selection of proper resources with the help of user centric meta-scheduler deals. For the increased resource utilization the scheduler used is system centric Virtual Machine (VM) which optimally distribute the jobs. For the ideal host VM development we proposed a scheduling heuristics.

B. High Volume Computing

The workloads with three types of throughput is discovered systematically from the data centers: services, data processing applications which produce, store and analyze massive data or also called as Big data application, and interactive real-time applications is a VoIP(Voice Of IP)application which assure the real time QOS by long duration maintenance of a user session, whose main goals are to maximize the size of throughput by requests which is been processed, or supported increased number of simultaneous subscribers, respectively, and a new term High Volume Computing (in short HVC) is mainly for throughput-oriented workloads. The main aim of a HVC workloads is to maximize the size of throughput by requests, or processed data, or the increased number of simultaneous subscribers. The HVC is compare with other computing paradigms, e.g., high throughput computing [9], warehouse-scale computing [10], and cloud computing. The main basement for developing the data center system for HVC workload is metrics and benchmarks which is an ongoing work for HVC systems.

C. Cost Model

The development of data center made the usage of services increases in cost of resources. These increase in cost made the development of cost model. Which must take into account the complexity in power delivery, cooling, and required levels of redundancies for a given SLA [6]. The main aim of this model is to reduce the resource cost by using equipments.

D. Cost –Aware Heterogeneous Workloads

Analysis on cost describes that the server cost is high in the total cost. The High cost of data centers gives a burden on providers of resource and server, which gives end user the services, and hence, how to reduce cost is an urgent issue. A new swap on classical provisioning of resource problem: heterogeneous workloads are the main life of large-scale data centers, and provisioning of resource solutions do not works on this heterogeneity. So propose a related provisioning of resource solutions to reducing the maximum resource utilization of workloads. Hence cost of server and infrastructure can be saved through reducing the maximum resources demands of workloads. The experiments developed describes that the result could save the cost of server aggressively with the help of non-cooperative solutions that are most spread which is used in state-of-the-practice organizing the data centers.

E. PrIter

In data analysis applications Iterative computations are increased using. These applications contain data set of very large size. The large data set size need to be assessed fast for these iterative computations applications. The prioritization provides the chance for accessing the iterative computations. The data points are so far accessed as a whole without any importance, so we prioritize the computations by its importance which helps convergence speed to be increased significantly on iterative process. A PrIter- Prioritized Iterative computation [8], which is developed and executed based on HadoopMapReduce. PrIter uses iMapReduce to supply the functionality of iterative computations. For the quick access to the value/state design at the reduce side the state table is maintained, a priority value is given to each node by the user and online top-k query is maintained. PrIter also helps in checking load balancing, and fault-tolerance. A file-based PrIter[8] is developed for providing the scalability.

Table I.Comparison Table

Paper Title	Description	Limitations
Design and Implementation of a Two Level Scheduler for HADOOP Data Grids[4]	Two- level scheduler focuses on optimizing the overall resource utilization and guaranteeing increased performance of the application	System is computationally inexpensive in the predicting and linking the recognized entities.
High Volume Computing: Identifying and Characterizing Throughput Oriented Workloads in Data Centers for the first time[5]	The main aim of a HVC workloads is to maximize the size of throughput by requests, or processed data, or the increased number of simultaneous subscribers. The HVC is compare with other computing paradigms.	Classification of the labeled data for sentimental analysis yields high precision and recall value.

Cost Model for Planning, Development and Operation of a Data Center[6]	The increase in cost made the development of cost model. The main aim of this model is to reduce the resource cost by using equipments.	This paper focuses on collecting the opinions of specific topic and it generates a summary automatically. It uses hashtags and human annotated semantic tags.
Cost –Aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers[7]	The experiments developed describes that the result could save the cost of server aggressively with the help of noncooperative solutions that are most spread which is used in state-of-the-practice organising the data centers.	User defined selection criteria is applied for filtering tweets on local context with high scalability.
A General Model for Virtual Machines Resources Allocation in Multi-Tier Distributed Systems[11]	Focused on resource allocations not about the task allocation.	Resource and task allocation both is going to be concentrate. The heterogeneous resource and task scheduling will also going to be performed.
PrIter: A Distributed Framework for Prioritizing Iterative Computations[8]	The data points are so far accessed as a whole without any importance, so we prioritize the computations by its importance which helps convergence speed to be increased significantly on iterative process. A PrIter- Prioritized Iterative computation [8], which is developed and executed based on HadoopMapReduce.	It uses word similarity and word context to detect ill-formed words which is easy for the users to understand

IV. PROPOSED SYSTEM

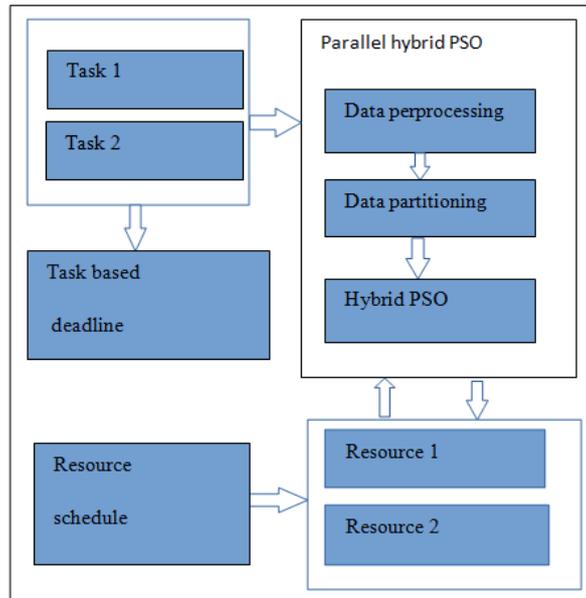


Fig 1: Flow of Framework

We model a framework to enhance the MapReduce performance in terms of increasing the efficiency and reducing the execution time. In our model we first perform Data Preprocessing in Hadoop distributed file system. In Data Preprocessing the data collection is pre-processed using parsing and random partitioning method. Once the data is stored in Hadoop Distributed File System(HDFS), the stored data need to be partitioned among the different servers in Hadoop cluster. The sample data is partitioned as training set, validation set and test set. The Genetic Algorithm is applied for this sample data. This algorithm is used to schedule the resource and task. Since resource states vary dynamically, the information monitored or predicted has to be timely updated through extraction functionality, to guarantee online estimation of resource conditions. In our system, resource sensors and prediction models are periodically has been placed and executed to generate up-to-date information for users. Practical Swarm Optimization (PSO) is selected as the auto optimization strategy for Hybrid resource scheduling algorithm (PH-PSO). This algorithm is used to perform parallel scheduling of resource and task and helps in improving the efficiency of MapReduce and reduces the execution time of the model. This flow is show in the fig 1.

V. METHODS TO BE USED

A. Data Preprocessing

The data preprocessing is the phase that need to be done before scheduling the resources and task. Hadoop framework is been established with complex distributed systems composed of resources and Tasks. Resource information extraction process has been build to monitor and predict Hadoop resource state information for our system architecture. Task users do not need traversal of all the nodes or hadoop expertise to get information. We design a uniform and friendly interface component for accessing the information. Computing system architecture maintains a service container for taking jobs; such container should be reused for seamless fusion between a environment and task or workload. Executing jobs is the fundamental function of a Hadoop system, we deploy resource sensors on computing nodes since it's inevitable, they also run and sleep dynamically to reduce overhead, while we deploy other components out of computing nodes to avoid extra overhead.

B. Genetic Algorithm

This algorithm is used to schedule the resource and task. The steps involved in this algorithm as follows,

- (a) Fix a prediction model of machine learning algorithm, and set its default hyper-parameters. Separate the sample set into three parts: training set, validation set and test set.
- (b) Feed the learning algorithm with a sample of training set, repeat it one by one until all samples are used. For some algorithms, the training procedure runs only once; for others, iterations are needed.
- (c) Feed the trained model with all samples of validation set, record the errors between true data and predicted ones.
- (d) Fix an optimization algorithm, which evolves the hyper-parameters of prediction model for better fitness (performance).
- (e) When a termination condition is met, optimized prediction model is achieved, and then tested using test set.

C. PSO and PH-PSO

PSO is an algorithm used for parallel schedule of resource and task. The random particles population is initialized with the PSO algorithm and by updating the particle generations searches a solution space of multidimensional for optima. The local best solution which is discovered by itself and the global best solution which is discovered by swarm are used to direct each particle movement. There is no need for traversal of all nodes expertise by the hadoop users to get the information. Preprocessing the sample set with corresponding features as well as candidate model with corresponding hyper parameters according to particle representation. The validation set is used for Fitness evaluation to be done in parallel which helps to evaluate candidate model and the fitness of the particle is calculated. The local best position and local best fitness is updated if particle fitness is better than local best fitness, similarly the global best fitness and global best particle is updated if a particle's fitness is better than the global best fitness. Once the condition is satisfied the velocity and position of each particle is updated. The output will be global best position; with the help of global best position's representation the sample set is prepared with selected features and with selected hyper parameter prediction model.

VI. CONCLUSION

In this analysis, Energy reduction has to be incorporated into the mapreduce implementation to improve the efficiency of the data center. The parallel scheduler for resource and task using particle swarm optimization is to manage the map and reduce tasks assignment. The Resource management is carried to manage a resource slots which reduces the consumption of energy when running the application achieves optimal schedules. Performance evaluation of the frameworks is compared with state of approaches which concludes that the framework outperforms in terms of efficiency and effectiveness.

REFERENCES

- [1] Qi Zhang, Mohamed Faten Zhani, Yuke Yang, Raouf Boutaba, Fellow, and Bernard Wong, "PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce", in IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, APRIL/JUNE 2015, pp. 182-194.
- [2] R. Boutaba, L. Cheng, and Q. Zhang, "On cloud computational models and the heterogeneity challenge," J. Internet Serv. Appl., vol. 3, no. 1, 2012, pp. 1-10.
- [3] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in Proc. Eur. Conf. Comput. Syst., 2010, pp. 265-278.
- [4] G.Sudha.Sadhasivam,N.NagaveniNagaveni, Vasanth Ram Rajarathinamproposed "Design and Implementation of a Two Level Scheduler for HADOOP Data Grids" in International conference on Advances in Recent Technologies in communication and computing,source: IEEE Xplore, November 2009,pp.884-886.
- [5] Jianfeng Zhan, Lixin Zhang, Ninghui Sun, Lei Wang, Zhen Jia, and Chunjie Luo,"High Volume Computing: Identifying and Characterizing Throughput Oriented Workloads in Data Centers",in arXiv: 1202.6134v2 [cs.DC] 14 jan 2103.
- [6] Chandrakant D. Patel, Amip J. Shah,"Cost Model for Planning, Development and Operation of a Data Center", in Internet Systems and Storage Laboratory HP Laboratories Palo Alto, HPL-2005-107(R.1) ,June 9, 2005.

- [7] Jianfeng Zhan, Member, IEEE, Lei Wang, Xiaona Li, Weisong Shi, Chuliang Weng, Wenyao Zhang, and Xiutao Zang, “*Cost-Aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers*”, IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 11, NOVEMBER 2013, pp. 2155-2168.
- [8] Yanfeng Zhang, Qixin Gao, Lixin Gao, Fellow, and Cuirong Wang, “*PrIter: A Distributed Framework for Prioritizing Iterative Computations*”, in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 9, SEPTEMBER 2013, pp. 1884-1893.
- [9] M. Livny *et al.* 1997. *Mechanisms for High Throughput Computing*, SPEEDUP Journal 1(1), 1997
- [10] R. E Bryant. 2008. *Data intensive scalable computing*. Retrieved January 5, 2010, from: <http://www.cs.cmu.edu/bryant/presentations/DISC-concept.ppt>.
- [11] Campegiani, P. ; Lo Presti, F.; “*A General Model for Virtual Machines Resources Allocation in Multi-tier Distributed Systems*” published in *Autonomic and Autonomous Systems*, 2009. ICAS '09. Fifth International Conference on 20-25 April 2009. PP 162-167.