

# Virtually Allocated Resources in Phase-Level using MapReduce in Hadoop

<sup>1</sup>D. Arunmozhi

Department of Computer Science and Engineering  
Sathyabama University

<sup>2</sup>Dr. T. Prem Jacob

Faculty of Computing  
Sathyabama University

**Abstract**— MapReduce is a simple programming model for a large scale data processing. MapReduce is a grouping the models of map and reduce with a related implementation. It is used for processing and generating large number of data sets. The task can be reduced by using mapreduce concept it can highly varying resource requirements so it is difficult for task level schedulers to promote available resources can be reduce to job execution time. The existing systems that focus on task level scheduling, so it has some drawbacks in job performance. Phase level scheduling algorithm is aim of achieving high performance job and resource utilization. The proposed systems introduces the phase-level scheduling algorithm that increase to fulfill the execution and resource utilization without introducing laggards using fail over resilience algorithm. Phase level offers high resource application and provides one third times increase it improvements in job running time assessed to the current hadoop schedulers.

**Keywords**— Cloud computing, MapReduce, Hadoop, Resource allocation, Scheduling.

## I INTRODUCTION

In cloud environment users pay for resources users. Hence it provides the business model of the cloud is key to the uniformity between the price and the quality of service. Cloud computing provides a service to another company and accessed over the internet. Cloud services are available as private cloud, public cloud, community cloud and hybrid cloud. In public cloud offers over the internet and it is operated by cloud provider. In private cloud is completely operated by organization or a third party. Cloud computing can significantly reduce the cost and complexity of operating the computer.

The main challenge of resource management is hadoop clusters systems. It is beginning of the resource model adopted in MapReduce. MapReduce is a job of collection of Map and Reduce tasks that can be scheduled concurrently on multiple machines resulting in significant reduction in job running time. Many large companies such as Google, Facebook, and Yahoo routinely use MapReduce to process large volumes of data on a daily basis. Consequently, the performance and efficiency of MapReduce frameworks have become critical to the success of today's internet companies. A central component to a MapReduce system is its job scheduler. Its role is to create a schedule of Map and Reduce tasks, spanning one or more jobs that minimizes job completion time and maximizes resource utilization. A schedule with too many concurrently running tasks on a single machine will result in heavy resource contention and long job completion time. Conversely, a schedule with too

few concurrently running tasks on a single machine will cause the machine to have poor resource utilization [9]. Hadoop is an open sources framework. Hadoop allows distributed processing of large dataset across cluster of computers.

Hadoop application works in environment that provides computation across clusters of computer and distributed storage. Hadoop common, Hadoop Yarn, Hadoop distributed file system and Hadoop Mapreduce are the four modules of Hadoop framework. Hadoop mapReduce is a software framework. MapReduce has two different tasks they are map task and reduce task. Map task takes input data and converts into a collection of data. Reduce task takes the output from map task as input and merges these data tuples into smaller set of tuples. All the time reduce task is performed after the map task. The main advantages of hadoop system are hadoop framework allows the user to fastly write and test distributed systems. The data are automatically distributibuted in efficient manner. Hadoop is a java based and also it is compatible on all the platforms.

The job scheduling problem becomes significantly easier to solve if we can assume that all map tasks have homogenous resource requirements in terms of CPU, memory, disk and network bandwidth. Indeed, current MapReduce systems, such as Hadoop MapReduce Version 1.x, make this assumption to simplify the scheduling problem. These systems use a simple slot-based resource allocation scheme, where physical resources on each machine are captured by the number of identical slots that can be assigned to tasks. Unfortunately it practice in run-time resource consumption varies from task to task and from job to job. Several recent studies have reported that production workloads often have diverse utilization profiles and performance requirements. Failing to consider these job usage characteristics can potentially lead to inefficient job schedules with low resource utilization and long job execution time [9].

Phase level scheduling requires resources information for each job. PRISM is ideal for environment where jobs that are executed repeatedly with the same input size which is common in many production clusters. The accuracy of the job profiles can be improves over time. PRISM can fall back to use task level resources information specified for hadoop Yarn, while phase level resources is not in use.

## II REVIEW OF RELATED WORK

Qi Zhang et al [9] describes the phase level resource analyze the run time task resource requirement in each phase for various hadoop jobs. It has 16 node cluster, each node acting as a master node and the others are 15 slave nodes. The task run-time resource analysis describes the fixed sized container for each task can guide to inefficient scheduling decisions. Phase level scheduling scheme that allocates resources to the phase that each task is simultaneously executed. The characteristic of phase level resource is better bin-pack task on machine to achieve higher resources utilization compared to task-level schedulers. Phase level job scheduling algorithm improves the job execution without introducing stragglers.

J. Polo et al[8] uses the Resources-aware Adaptive Scheduler (RAS) is used to estimate the number of tasks to be run in parallel for each job in order to meet some performance objectives shows in RAS in the form of goal reached time and was highly evaluated and validated. They introduce the concept job slot, it is an execution slot from bound to a particular job and a particular map and reduce task type within that job. The number of slots in a Hadoop cluster is fixed throughout the proposed solution can be reduced to a different of task assignment or slot assignment problem. A resource model consisting of a resource container like our job slot it means replace across the job tasks.

M.Malekimajd et al[7] proposed that each locality of our economy are now managed by data driven decision making process. Big Data are assisting by the mapreduce programming model at framework layer. Cloud computing provides varying the cluster and efficient cost solutions for allocating on demand large cluster. New upper and lower bounds and linear programming model are overlapping each other. They validate job execution time bounds and evaluate the capacity allocation for optimal solution. The cloud based hadoop cluster depends on the execution costs of heterogeneous task to be reduced using optimization model.

M.Zaharia et al[14] to find the delay scheduling, when the job allocated to next schedule it takes more time to launches the task. Delay scheduling achieves optimal data locality in a variety workload can increase measure of time up to two times more than the preserving fairness. Delay scheduling is related to vast variety of scheduling policy before fair sharing. Data concentrate on cluster computing systems like mapreduce and Dryad grows in familiar. It is need to share clusters between users. Using delay scheduling, it is possible to achieve 100% locality by relaxing fairness.

H.Herodotou et al[5] introduces the starfish concept, it is a self tuning system for big data analysis. The workload of a hadoop deployment can be considered at different levels. Starfish architecture can be categorized into job level tuning, workflow level tuning and workload level tuning. Starfish causes a different void can be qualify the Hadoop user and applications to get good performance automatically throughout the data analysis. The approach enables Starfish to handle the important interactions arising among at different levels.

J. Dean and S. Ghemawat[3] implements the mapreduce runs on large clusters of commodity machines and

highly scalable. Everyday Programmers are executed the mapreduce job. Many systems have delivered restricted program model and it use the restrictions to parallelize the computation automatically. A key comparison between these systems and mapreduce is that exploit a restricted model to correlate the user program immediately and to present the transparent fault tolerance.

A.Rasmussen et al[10] implements the concept Themis MapReduce is read and write the data records to disk exactly twice. The minimum possible amount of data sets that cannot fit in memory. A wide variety of mapreduce jobs are clicking log analysis DNA read sequence alignment and page rank are performed by ThemisMR. Due to failure occurs ThemisMR to replay the computation necessary to recover that lost data.

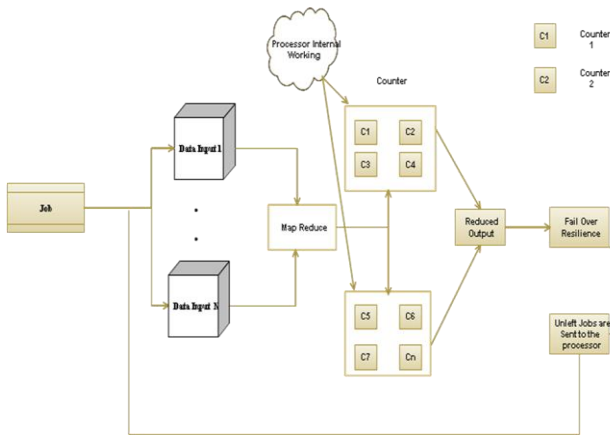
M.Zaharia et al[15] designs a Longest Aproximate Time to End (LATE) algorithm can improves hadoop response time. LATE treats the tasks to make sure that only the slowest get speculatively executed and also caps the amount of speculative tasks to limit disagrees for shared resources and avoid thrashing.

A.Verma et al[11] introduce a framework and technique to address the problem and to provide a new resource analyzing and provisioning service in mapreduce environment. MapReduce job needs to be finished within a certain time. A set of resource provisioning options is applied in the scaling rules. The forecast completion time of generated resource provisioning options are within 10% constant times in our 66-node hadoop cluster.

Y.Yu and Micheal Isaed[13] offers a new programming models are Dryad and DryadLINQ system. A DryadLINQ program is a forming program composed of LINQ expressions performing free operations on datasets. Dryad and DryadLINQ systems are discuss the tradeoff and connection to the parallel and distributed databases.

## III PROPOSED SYSTEM

The main contribution of the review is to demonstrate the importance of phase-level. In a phase-level, we perform a task or process with heterogeneous resource requirements. Phase-level scheduling which improves execution parallelism and performance of task. At the time of assigning the job, PRISM offers higher degree of parallelism than current hadoop cluster. It refers at the phase-level to improve resource utilization and performance. In the thread will assign jobs to the processor, but the thread do not know about the processor. After the jobs are assigned to the processor, information about the job size will go the MapReduce and then it know the processor capacity and it assign the work based on their Processor capacity. The processor will complete all works that are assigned. It does not leave any unfinished jobs as like the existing ones. After that the results of each processor go to the fail over resilience mechanism. In the mechanism it checks if all the jobs are finished by the processor it will go the success result or otherwise if any jobs are incomplete it go to the MapReduce and it find the processors and make that processor to complete the jobs.



IV CONCLUSION AND FUTURE WORK

MapReduce is a programming model for huge data sets. Hadoop is an open-source implementation of mapreduce. Applications are data mining, web indexing and scientific simulation. Hadoop performance is closely tied to task scheduler who totally assumes that the cluster nodes are parallel and tasks that make progress linearly and uses these acceptances to decide when the speculatively re-execute the tasks that appear to be laggard using fail over resilience algorithm.

Finally, there are many interesting avenues for future exploration. In particular, study the problem of meeting job deadlines under phase level scheduling is must. The assumption of all machines has identical hardware and resource capacity. It is interesting to study the profiling and scheduling problem for machines with heterogeneous performance characteristics. Finally, improving the scalability of PRISM using distributed schedulers is also an interesting direction for future research.

REFERENCES

[1] R. Boutaba et al (2012), "On cloud computational models and the heterogeneity challenge," J. Internet Serv. Appl., vol. 3, no. 1, pp. 1–10.

[2] T. Condie, N. Conway, P. Alvaro, J. Hellerstein, K. Elmeleegy, and R. Sears, (2010), "MapReduce online," in Proc. USENIX Symp. Netw. Syst.Des. Implementation, p. 21.

[3] J. Dean and S. Ghemawat, (2008) "Mapreduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113.

[4] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, (2011), "Dominant resource fairness: Fair allocation of multiple resource types," in Proc. USENIX Symp. Netw. Syst. Des. Implementation, pp. 323–336.

[5] H. Herodotou et al (2011), "Starfish: A self-tuning System for Big Data Analytics," in Proc. Conf. Innovative Data Syst. Res., pp. 261–272.

[6] M. Isard et al (2009), "Quincy: Fair scheduling for distributed computing clusters," in Proc. ACM SIGOPS Symp. Oper. Syst. Principles, pp. 261–276.

[7] M.Malekimajd, D.Ardagna, M.Ciavotta, A.M.Rizzi and M.Passacantando, "Optimal MapReduce Job Capacity Allocation in Cloud Systems".

[8] J. Polo, C. Castillo, D. Carrera, Y. Becerra, I. Whalley, M. Steinder, J. Torres, and E. Ayguad\_e, (2011), "Resource-aware adaptive scheduling for mapreduce clusters," in Proc. ACM/IFIP/USENIX Int. Conf. Middleware, pp. 187–207.

[9] Qi Zhang, M.Faten Zhani, Yuke Yang, Raouf Boutaba and Bernard Wong, April/June (2015), "PRISM: Fine-grained resource-aware scheduling for MapReduce," in IEEE Transactions on Cloud Computing, Vol3,No.2.

[10] A. Rasmussen, M. Conley, R. Kapoor, V. T. Lam, G. Porter, and A. Vahdat, (2012), "ThemisMR: An I/O-Efficient MapReduce," in Proc. ACM Symp. Cloud Comput., p. 13.

[11] A. Verma, L. Cherkasova, and R. Campbell,(2011), "Resource provisioning framework for MapReduce jobs with performance goals," in Proc. ACM/IFIP/USENIX Int. Conf. Middleware, pp. 165–186.

[12] D. Xie, N. Ding, Y. Hu, and R. Kompella, (2012), "The only constant is change: Incorporating time-varying network reservations in data centers," in Proc. ACM SIGCOMM, pp. 199–210.

[13] Y. Yu et al (2008), "DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language," in Proc. USENIX Symp. Oper. Syst. Des. Implementation, pp. 1–14.

[14] M. Zaharia et al "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in Proc. Eur. Conf. Comput. Syst., 2010, pp. 265–278.

[15] M. Zaharia et al (2008), "Improving MapReduce performance in heterogeneous environments," in Proc. USENIX Symp. Oper. Syst. Des. Implementation, vol. 8, pp. 29–42.

[16] Hadoop MapReduce distribution [Online]. Available: <http://hadoop.apache.org>, 2015.

[17] Hadoop Capacity Scheduler [Online]. Available: [http://hadoop.apache.org/docs/stable/capacity\\_scheduler.html/](http://hadoop.apache.org/docs/stable/capacity_scheduler.html/), 2015.