

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 5, May 2018

Survey on PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce

Nilima P. Sonawane

S.S.V.P.S's B.S.Deore College of Engineering, Dhule, India

ABSTRACT: Map Reduce has become a popular model for intensive data counting in recent years. By dividing each job into a small map and reducing tasks and running them in parallel on machines, Map Reduction can significantly reduce the execution time of jobs that require a lot of data. However, despite recent efforts to design map reducers with efficient use of resources, existing solutions focused on activity-level programming still offer suboptimal work performance. This is due to the fact that the activities may have very varied resource requirements during their useful life, making it difficult for task-level programmers to effectively use the resources available to reduce work execution time. To overcome this limitation, we present PRISM, an extremely accurate map reduction planner that divides tasks into phases, where each phase has a constant resource usage profile and performs phase-level programming. First, we demonstrate the importance of phase-level programming by showing the variability of resource use over the life of an activity using a wide range of Map Reduction jobs. Next, we present a phase programming algorithm that improves execution parallelism and the use of resources without introducing delays.

KEYWORDS: MapReduce, Hadoop, scheduling, resource allocation

I. INTRODUCTION

Today, companies depend entirely on large-scale data analysis, so they can make critical business decisions day by day. This is directed to the development of Map-Reduce, i.e. a parallel programming model that has become equivalent with large-scale and data-intensive calculations. Map-Reduce consists of a job, which is a collection of Map and Minimize activities. These activities can be synchronously programmed on several machines, with a substantial reduction in work time. An essential component of a Map-Reduce system is your task planner. The main role of the activity planning program is to create a mapping and reduction planning activity that includes one or more jobs, minimizes job completion time and maximizes resource utilization. In many situations, the containment of heavy resources and the time of completion of long processes take place due to a planning with too many tasks performed simultaneously on a single machine. On the contrary, hunger occurs because of the improper use of resources and also because of a planning with very few simultaneous activities in a single machine.

The problem of job scheduling becomes significantly simpler to solve, assuming that all map activities (and all reduction activities) have consistent resource requirements, such as CPU, memory, disk, and network bandwidth. However, this hypothesis is used to simplify the programming problem with current systems of Map Reduction, such as Hadoop Map-Reduce Version 1.x. This system uses a simple slot-based resource allocation scheme, in which the physical resources of each machine take on the amount of indistinguishable slots that can be allocated to activities.

This document offers PRISM, which is a fine-stage programmer and resource information for map reduction clusters. PRISM realizes the conscious planning of resources at the level of the phases. Specifically, this document shows that, for Map-Reduce applications, the consumption of resources of the activity during the execution time can vary considerably from one phase to another. Therefore, it is possible that the planner has a greater degree of parallelism even if it avoids the containment of resources, only when it comes to the demand for resources at the phase level. Therefore, in the end, this document has developed a phase-level programming algorithm with the aim of obtaining high work performance together with the appropriate use of resources.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 5, May 2018

II. LITERATURE SURVEY

The application works on MapReduce a large cluster of raw materials machines and is highly scalable. A typical MapReduce computer processes a large amount of data in thousands of machinery. Many programs have been implemented in MapReduce Google cluster every day. Hadoop MapReduce implements resource allocation scheme based on slots, which does not consider consumption of resources time of execution of the task. As a result, many recent ones newspapers have reported the inefficiency introduced because such a simple design and solutions proposed. For example, Polo et al. RAS has proposed [15], an adaptive scheduler resources uses a resource-conscious programming technique for MapReduce multi-job workloads aimed at improving resource utilization on all the machines it observes the objectives of the completion time. However, RAS still exhibits activity-level planning, not consider changes in task use of resources at runtime. Subsequently, Hadoop yarn [7] represents a great effort planning of resources in MapReduce cluster. Offers possibility to specify the size of each container of the activities in terms of requirements for each type of resource. In this context, a key challenge is to define the notion of equity when more resource types are considered Ghodsi et al. proposed key equity of resources as a measure of equity, in the presence of different types of resources, and provided simple programming algorithm for obtaining an almost optimal DRF However, the DRF The scheduling algorithm is still focused on activity level programming and does not consider the variation in the consumption of resources inside His individual tasks after model, that is to say. Dominant Resource Fair Queuing (diabetes FC), aims to achieve DRF planning of packages over time. However, the algorithm is CFRD It is designed primarily for programming packages, which is different from the type of activity-based "bin-packing" planning model Consider in this role. Therefore, it cannot be applied directly to MapReduce programming. Use profiles to improve MapReduce work performance has received considerable attention in the past years [12]. For example, Varma et al. [17] developed a framework that outlines the execution time of the activities and uses the work profiles to reach the expiry program in MapReduce cluster. Herodotus et al. newly developed starfish [12], a job profile generator that collects the characteristics of use of fine-grained tasks it can be used to adjust the configuration parameters of the job. However, the goal of the profiles in these studies is to optimize the work parameters, instead of optimizing work programs. Anotherthe related search direction is the MapReduce funnel.In particular, MapReduce Online [9] is a framework for processing based on the MapReduce workflow. It allows partial results of each phase that will be sent directly to the next stage, therefore, it allows the overlap of the phase execution. To minimize I / O, ThemisisMR [16] is another scheme that makes the design decisions are fundamentally different from the previous ones MapReduce implementations. Themis does an extended Variety of MapReduce jobs almost at the speed of Triton Sort record classification performance. However, both these Solutions do not concern programming. Moreover, they are not aware of the resources. Introducing the awareness of resources into MapReduce Online is another interesting alternative, the programming model for MapReduce online is very different from the current MapReduce. Further research will be needed identify programming problems for Map-Reduce online.

III. EXISTING SYSTEM

The original MapReduce work is to schedule the task in different levels using fair scheduler and Capacity scheduler. In a MapReduce technique, it is a collection of jobs and can be scheduled concurrently on multiple machines, resulting in reduction in job runningtime.Many companies such as Google, Facebook, and Yahoo, they refer MapReduce to process large volume of data. But they refer at task level to perform these data. At the task level, performance and effectiveness become critical to day-to-day life. Initially the task level performs two phases one is mapper phase and other is reducer phase. In mapper phase, it takes data blocks hadoop distributed file system and it maps, merge the data and stored in the multiple files. Then the second, reducer phase will fetch data from mapper output and shuffle, sort the data in a serialized manner.

Disadvantages of Existing System

- Varying resources at the task-level offer lower performance.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 5, May 2018

- It is difficult for task-level scheduler to utilize the run-time resources. So that it reduces job execution time while executing.

IV. PROPOSED SYSTEM

The main contribution of this paper is to demonstrate the importance of the phase level. At the phase level, we run an activity or process with heterogeneous resource requirements. We have a phase-level programming algorithm. This improves the parallelism of the execution and the performance of the task. The phase level that has these parameters with good working characters. Therefore, we present PRISM, i.e. Information on phases and resources: for the problem programmer MapReduce at the phase level. When running a task, you have a lot of runtime resources in the life cycle. During work planning, PRISM offers a greater degree of parallelism than the current hadoop cluster. It refers to the Phase level to improve the use and performance of resources.

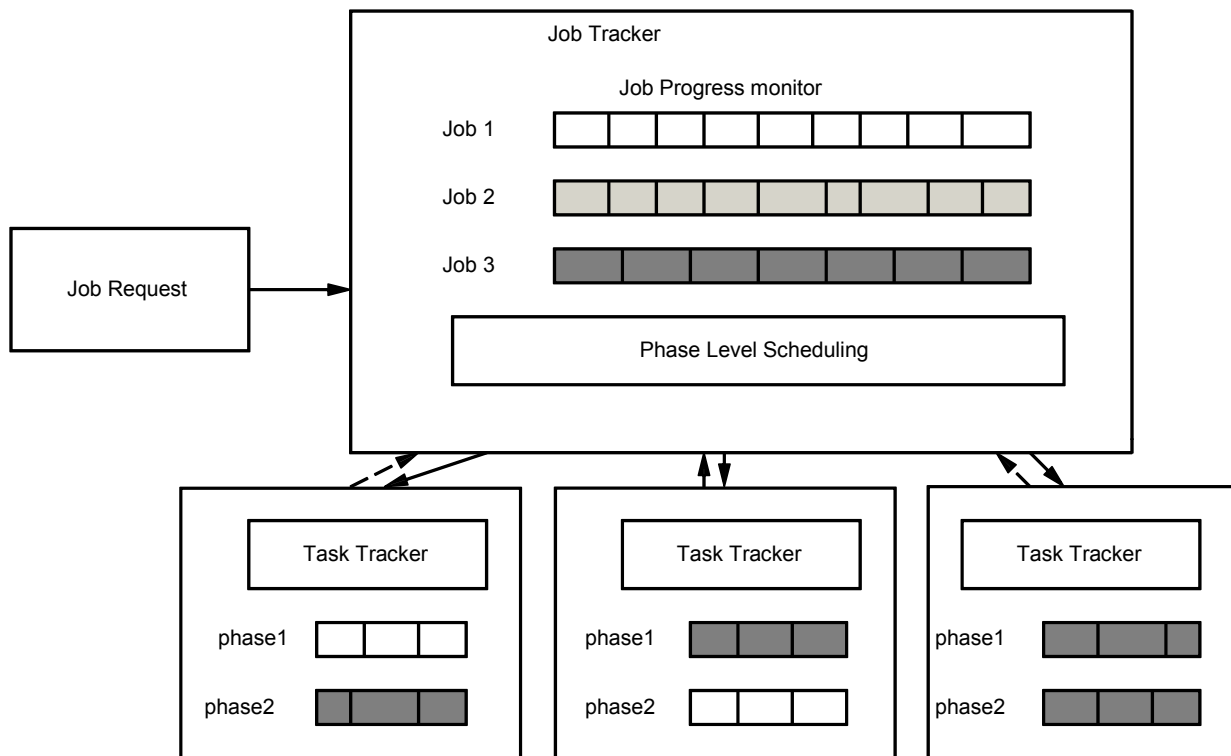


Fig. System Architecture

IV. CONCLUSION

In this Paper, Map-Reduce is used as a popular programming model to calculate data-intensive jobs. PRISM, which is a fine-grained resource allocation / reduction planner, divides tasks into phases and also performs phase-level programming. Due to the use of this phase-level programming, there is an improvement in the use of resources. The planning algorithm used by PRISM contributes to minimizing work execution time compared to current Hadoop programmers. In general, PRISM achieves high work performance. Finally, the future purpose of this document will be to improve the scalability of PRISM through the use of distributed programmers.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 5, May 2018

REFERENCES

- [1] Hadoop Map Reduce distribution [Online]. Available: <http://hadoop.apache.org>, 2015.
- [2] Hadoop Fair Scheduler [Online]. Available: http://hadoop.apache.org/docs/r0.20.2/fair_scheduler.html, 2015.
- [3] Hadoop Distributed File System [Online]. Available: hadoop.apache.org/docs/hdfs/current/, 2015.
- [4] The Next Generation of Apache Hadoop MapReduce [Online]. Available: <http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>, 2015.
- [5] R. Boutaba, L. Cheng, and Q. Zhang, "On cloud computational models and the heterogeneity challenge," *J. Internet Serv. Appl.*, vol. 3, no. 1, pp. 1–10, 2012.
- [6] T. Condie, N. Conway, P. Alvaro, J. Hellerstein, K. Elmeleegy, and R. Sears, "MapReduce online," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2010, p. 21.
- [7] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [8] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2011, pp. 323–336.
- [9] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. Cetin, and S. Babu, "Starfish: A self-tuning system for big data analytics," in *Proc. Conf. Innovative Data Syst. Res.*, 2011, pp. 261–272.
- [10] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, and K. Talwar, "Quincy: Fair scheduling for distributed computing clusters," in *Proc. ACM SIGOPS Symp. Oper. Syst. Principles*, 2009, pp. 261–276.
- [11] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multi-resource allocation: Flexible tradeoffs in a unifying framework," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1206–1214.
- [12] J. Polo, C. Castillo, D. Carrera, Y. Becerra, I. Whalley, M. Steinder, J. Torres, and E. Ayguad_e, "Resource-aware adaptive scheduling for MapReduce clusters," in *Proc. ACM/IFIP/USENIX Int. Conf. Middleware*, 2011, pp. 187–207.
- [13] A. Rasmussen, M. Conley, R. Kapoor, V. T. Lam, G. Porter, and A. Vahdat, "ThemisMR: An I/O-Efficient MapReduce," in *Proc. ACM Symp. Cloud Compute.* 2012, p. 13.
- [14] A. Verma, L. Cherkasova, and R. Campbell, "Resource provisioning framework for MapReduce jobs with performance goals," in *Proc. ACM/IFIP/USENIX Int. Conf. Middleware*, 2011, pp. 165–186.
- [15] D. Xie, N. Ding, Y. Hu, and R. Kompella, "The only constant is change: Incorporating time-varying network reservations in data centers," in *Proc. ACM SIGCOMM*, 2012, pp. 199–210.
- [16] Y. Yu, M. Isard, D. Fetterly, M. Budiu, _ U. Erlingsson, P. Gunda, and J. Currey, "DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language," in *Proc. USENIX Symp. Oper. Syst. Des. Implementation*, 2008, pp. 1–14.
- [17] M. Zaharia, D. Borthakur, J. SenSarma, 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.
- [18] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica, "Improving MapReduce performance in heterogeneous environments," in *Proc. USENIX Symp. Oper. Syst. Des. Implementation*, 2008, vol. 8, pp. 29–42.