Self-Slot Configurations for Dynamic Hadoop Clusters – A Review

Ahmed Zuber Behar[1] . Mohammed Zain Khalid[2]

[1]Faculty of Engineering, Al-Azhar University, Egypt
[2]Faculty of Information Science, King Faisal University, Saudi Arabia

Received: 09 March 2022 / Revised: 23 April 2022 / Published Online: 06 June 2022
©Milestone Research Publications, Part of CLOCKSS archiving

Abstract – For analyzing the large set of scalable data by using map reduce framework and Hadoop has become popular. The major concentration of this paper is to produce the review on static slot configuration of Hadoop clusters under a dynamic job reduce approach. As the static slot of the cluster shall deal with only a similar pattern of data sets. By this paper, we shall present a brief survey on the Hadoop slot configuration and hence a clear agenda is being maintained as a clear comparison.

Index Terms – *Map reduce, Job Reduce, slot configuration, Dynamic Scheduling of MapReduce functions.*

I. INTRODUCTION

MapReduce has become a major research topic in current trends. The major focus of today's development in big data is towards the Hadoop management. This system of data generation shall enormous and cases large unused data paradigms for processing. In current technological era, a major concentration and focus is laid on how exactly a data is generated and analyzed under a positive environment.

As the demand of data has increased in recent times a major contribution has been done by various communities in understanding and analyzing the data clusters. Many academicians and industrial professionalism has initiated this process. With large data sets a distinct environment is created under big data warehouse. A Hadoop cluster is been configured and produced to deal with such a complex and immersed data under data mining techniques.

In this paper a brief survey has been conducted to reveal a self-configuring data slots to perform a job under a minimal job reduce and slot making time. Hence the entire system module developed is well featured with a performance analysis and thus fetches a high performance ratio.

The paper is organized under the preliminary standards of survey with Introduction in session I and followed by a brief literature survey on the two papers related to the subject under Session II and thus followed by a summary of Comparisons in Session III and concluded under session IV

II. LITERATURE SURVEY

Many authors such as Mr. Bikash and Ramya in the overall presentation of Hadoop they describe the slots as a resource for clustering a multiple resources and thus a big data maps are reduced under an optimal MapReduce approach. These slots are

programmed in a static manner, in this paper a deep abbreviation of how a job can be handled in such a complex environment.

Due to lack of coordination of management of multiple slots between resources and nodes of the environment, motivated dynamic slots are programmed to achieve a grater a deeper understanding on how these datasets are configured under a bigdata slots and with an algorithmic approach. As from this paper, we understand the disadvantages of how we failed to configure the bigdata slots under continues and prolonged datasets such as heart signals and ECG graph nodes for analyzing a feed approach.

From the paper of Y.Yao and J.Wang: they overcoming the problem from above paper by B.sharma and Ramya they have implemented the YARN. As from this paper a deeper approach is made on how to perform a detailed MapReduce approach with distinct dynamic slots. The performance ration depends on how we shall append an efficient resource scheduling for a Hadoop environment.

These clusters of resources slots perform the overall efficiency of the system. This paper also focus on how to eliminate the dependencies under slots and jobs. This paper shall remain the overall best scheme for resource utilization and performance assurance. Hence the entite the paper in this range shall depend on the performance. With this we can summer up the paper.

III. COMPARATIVE STUDY

The trivial terminology of system configuration of Hadoop clusters are based on static slot configuration. In this approach the system is programmed with a constant slots for a particular job under a resource v/s task and thus the system performance in lowed as the slots needs to wait until the available slots are released from the task. Later on time this technique is been faded with many new techniques.

The latest technique to overcome this is the new dynamic slot configuration approach. This technique shall dynamically configure the slots on based on task and thus reduces the overall head load. In this terminology, the system is been released with the allocated resource and the pool is been updated for future job scheduling.

The overall comparison is made with respect to the performance ratio. As the performance of the Hadoop framework with static slot configuration was been comparatively low and inefficient. Thus when the new terminology of Dynamic Slot is been studied, I have seen a hike on performance ratio and the clearance time of any job under the Hadoop with an effective time and resource sharing for faster and better performance in the overall framework.

IV. CONCLUSION

As of two techniques in general has been compared to fetch a clear idea on which of the two technique is better on performance and efficiency. In this survey paper, I shall claim no rights on the survey done as it was a primary requirement to conclude the better performing technique to move my work a step ahead. In this paper a brief overview from the different authors has been compared and review for the same is been projected.

References

1. Sharma, B., Prabhakar, R., Lim, S. H., Kandemir, M. T., & Das, C. R. (2012, June). Mrorchestrator: A fine-grained resource orchestration framework for mapreduce clusters. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 1-8). IEEE.
2. Yao, Y., Wang, J., Sheng, B., Lin, J., & Mi, N. (2014, June). Haste: Hadoop yarn scheduling based on task-dependency and resource-demand. In *2014 IEEE 7th international conference on cloud computing* (pp. 184-191). IEEE.

3. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.
4. Ahmed, S. T., Basha, S. M., Arumugam, S. R., & Kodabagi, M. M. (2021). *Pattern Recognition: An Introduction*. MileStone Research Publications.
5. Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S., & Stoica, I. (2010, April). Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In *Proceedings of the 5th European conference on Computer systems* (pp. 265-278).
6. Verma, A., Cherkasova, L., & Campbell, R. H. (2012, August). Two sides of a coin: Optimizing the schedule of mapreduce jobs to minimize their makespan and improve cluster performance. In *2012 IEEE 20th international symposium on modeling, analysis and simulation of computer and telecommunication systems* (pp. 11-18). IEEE.
7. Isard, M., Prabhakaran, V., Currey, J., Wieder, U., Talwar, K., & Goldberg, A. (2009, October). Quincy: fair scheduling for distributed computing clusters. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles* (pp. 261-276).
8. Verma, A., Cherkasova, L., & Campbell, R. H. (2011, June). Aria: automatic resource inference and allocation for mapreduce environments. In *Proceedings of the 8th ACM international conference on Autonomic computing* (pp. 235-244).
9. Polo, J., Carrera, D., Becerra, Y., Torres, J., Ayguadé, E., Steinder, M., & Whalley, I. (2010, April). Performance-driven task co-scheduling for mapreduce environments. In *2010 IEEE Network Operations and Management Symposium-NOMS 2010* (pp. 373-380). IEEE.
10. Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., ... & Baldeschwieler, E. (2013, October). Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th annual Symposium on Cloud Computing* (pp. 1-16).
11. Wang, X. W., Zhang, J., Liao, H. M., & Zha, L. (2011, November). Dynamic split model of resource utilization in mapreduce. In *Proceedings of the second international workshop on Data intensive computing in the clouds* (pp. 21-30).