International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at : www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET3214668

Survey on Resource Management Solutions to Speed up Processing Small Files in Hadoop Cluster

Prof. Shwetha K S[1], Dr. Chandramouli H[2]

[1]Ph.D Research Scholar, Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bengaluru, Karnataka, India

[2]Professor, Department of Computer Science and Engineering East Point College of Engineering and Technology, Bengaluru, Karnataka, India

ARTICLEINFO

Article History :

Accepted: 02 Nov 2023
Published: 20 Nov 2023

Publication Issue :
Volume 10, Issue 6
November-December-2023
Page Number :
164-172

ABSTRACT

High performance data analytics is a computing paradigm involving optimal placement of data, analytics and other computational resources such that superior performance is achieved with lesser resource consumption. Resource allocation and scheduling are the two major functionalities to be addressed in Hadoop clusters to satisfy the service level agreements of users for High performance data analytics applications. Though many solutions have been proposed for optimal resource allocation and scheduling, those schemes are designed for large Hadoop files. Recently with Internet of Things (IoT) convergence with big data, there is need to process large volumes of small files whose size is lower than block size of Hadoop. This creates huge storage overhead and exhausts Hadoop clusters computational resources. This survey analyzes the existing works on resource allocation and scheduling in Hadoop clusters and their suitability for small files. The aim is to identify the problems in existing resource allocation and scheduling approaches while handling small files. Based on the problems identified, prospective solution architecture is proposed.

Keywords : Internet of Things, Big data analytics, resource allocation and scheduling

I. INTRODUCTION

Big data analytics is revolutionizing the business across all sectors with its inherent ability to identify hidden knowledge from large volumes of data generated by enterprises. It has also contributed immensely to data driven machine/deep learning and artificial intelligence technologies which are rapidly transforming the ways business are conducted. High performance data analytics is the enabler technology for big data analytics. It involves optimal data placement, analytics, resource allocation such that desired performance is achieved with minimal resources. Hadoop is a big data processing platform designed in line with requirements of high performance data analytics. It is designed specifically to process data intensive applications optimally. Hadoop collocates data and computing together to reduce the data transfer overhead and increase the computational speed. The data is processed in form of blocks which is typically more than 64 MB. Each data block is allocated a map task. The allocation of data block to map task is managed by Namenode of Hadoop. Over the years many resource allocation and scheduling algorithms have been proposed to achieve higher performance over Hadoop clusters. These algorithms increased the performance gain, reduced the overhead in spawning map tasks and reduced the storage overhead at Namenode's.

With onset of IoT big data convergence, there is a need to process large volume of smaller files in Hadoop clusters. These smaller files have files size which are very marginal (<2KB) compared to block size of Hadoop. Processing of small files brings many performance bottlenecks to Hadoop. It increases the storage overhead at Namenode's, increases computational overhead due to short lived map tasks[1-2] and increases bookkeeping overhead [3]. Many solutions have been proposed to solve the small file problem (discussed in related works), but their applicability to Hadoop clusters has not been tested. Also works addressing small file problem to Hadoop clusters in the background of high performance data analytics are very meager. The resource allocation, data placement and scheduling algorithms of high performance data analytics has to be further improvised to handle small files processing requirement of IoT big data convergence. In this background, this survey analyzes the exiting resource allocation, data placement and scheduling algorithms of high performance data analytics in Hadoop platform. The aim is to identify the problems in existing works in handling small files. Based on the problems identified, an prospective solution architecture to handle the problems is proposed.

II. SURVEY

Lim et al [4] modeled resource allocation of scheduling on Hadoop as a optimization problem and proposed a constraint programming based resource allocation algorithm for Hadoop map reduce jobs. Each job has a service level agreement (SLA) in terms of deadline time for execution. A batch of jobs is considered for resource allocation and they are scheduled in such way to minimize the

SLA miss ratio. The proposed scheduling algorithm was data locality aware and it placed jobs proximal to data. In case of small files, the number of jobs shoots up and the computational complexity for resource allocation using constraint programming becomes NP hard in this approach. Yao et al [5] improvised the Hadoop's YARN scheduler to address the problems in fairness and efficiency while processing inter related tasks in Hadoop. The goal of the scheduler was to reduce the makespan of the batch of jobs by using the information on requested resources, resource capacities and dependency between tasks. Though authors considered maximizing the CPU and memory usage, they did not consider data nature and localization of tasks in their scheduling decision. Chen et al [6] proposed a Hadoop scheduler based on data locality. The Datanode's storage capability is set based on its execution capability. Blocks are given importance to be stored in the fastest Data node. By this way the performance of the map tasks were improved. But the scheme did not consider application and data characteristics in data block allocation and scheduling of tasks. Hammoud et al [7] proposed a locality aware scheduler to improve the map reduce tasks performance in Hadoop clusters. In addition to scheduling map based on data locality, reduce tasks are also scheduled based on data locality in this scheme. The early merge of map outputs are delayed and once after a sufficient time, the reduce task is scheduled on the node which is close to maximum size of map outputs. This improves the performance of reduce tasks by minimizing the data shuffling volume. But the scheme not does fully exploits the nature of data and its semantics in its scheduling decision. Kousiouris et al [8] experimented with CPU percentage allocated to tasks and their impact on application performance in clusters. Authors found that collocating inter related tasks on same node improves the performance. Thus identifying the inter related based on data characteristics and scheduling them on same node increases the performance. But the authors did not propose any scheme for identifying inter related tasks. Xie et al [9] proposed a data placement scheme for heterogeneous Hadoop clusters. A new metric called computing ratio was proposed to profile the nodes and data is placed in proportion to computing ratio. Authors also redistributed the data based on node utilization. The data placement strategy did not consider the data semantics and tasks correlation. Also the redistribution scheme proposed in this work has higher communication overhead. Wang et al [10] exploited both data locality and load balancing to increase the throughput in Hadoop clusters. A two stage scheduling algorithm was proposed with separate queue for each node and a common queue for all nodes. Nodes process task from its corresponding local queue when it is free, it process task from global queue using MaxWeight policy. Task processing did not consider data localization. Lim et al [11] improvised the constraint programming based resource management in Hadoop clusters to achieve higher performance. The job execution times were estimated using a presheduling error handling technique. But the error estimation technique proposed in this work did not consider the localization and proximity of data. Yang et al [12] proposed two techniques to optimize the internal overlap between map and reduce jobs to reduce the map reduce execution time in Hadoop clusters. Lazy start of reduce tasks and batch finish of map tasks are done for better alignment of map and reduce tasks. This alignment improved node throughput. But the proposed scheduling did not consider node localization in allocation of reduce tasks to node. Liu et al [13] proposed a Fair Sojourn Protocol in YARN scheduler to improve the responsiveness and ensure fairness in Hadoop clusters. It is a size based scheduler where job size is predicted and based on job size, resources are allocated to it. The efficiency of the scheduling depends on the accuracy of the job prediction. But prediction is based on moving average. But job size prediction did not consider data semantics. Bader et al [14] proposed a resource allocation algorithm called Tarema. The resource allocation algorithm was designed to process scientific workflows in Hadoop clusters effectively. The nodes are clustered based on profile similarity and task were grouped based on semantics. The task groups are allocated to node clusters. But the scheme did not consider the data semantics while grouping the tasks. Marquez et al [15] addressed the problems of data placement and resource allocation in heterogeneous Hadoop clusters using genetic programming. The data placement was optimized based on reduction of data write time using genetic algorithm. Resource allocation was optimized based on maximum utilization of physical machine. But the approach did not consider data semantics and task relation for data placement and allocation. Zhang et al [16] proposed a data locality aware scheduling for heterogeneous Hadoop clusters. The tasks are allocated to node based on objective of minimizing the waiting time and transmission time for data. A metric based on predicted waiting time and transmission time if formulated. For batch of jobs, the metric is calculated and the jobs are allocated to node with overall goal of minimizing the metric value. Task correlation and data correlation was not exploited to maximize the performance in this work. But the idea of evaluating the tasks for allocation using a metric is novel contribution in this work. Jeyaraj et al [17] proposed a fine grained data locality aware scheduling algorithm for reducing the makespan of map reduce jobs in Hadoop clusters. This scheduler attempts to minimize the amount of intermediate data in the shuffle phase by using multi level per node combiner. This scheme reduced the intermediate data by executing group of map jobs belonging to same application together. But the scheme did not consider data proximity for grouping map jobs. Kao et al [18] proposed a data locality aware real time scheduling technique for Hadoop clusters. The tasks were categorized to data intensive and CPU intensive. Data intensive tasks are scheduled to data proximal nodes and CPU intensive tasks are scheduled to nodes with minimal load interference. Authors did not consider collocating interrelated tasks to same node to maximize data proximity. Wei et al [19] extended the default first in first out (FIFO) scheduler of Hadoop with data locality awareness and sharing. Tasks requesting same data are grouped as batches and processed in same node, so that data to be shared across tasks is maximized. The scheduling algorithm was designed only for homogenous clusters. The proposed tasks grouping mechanism works did not consider the granularity of data and semantics of the data for grouping the tasks. Thaha et al [20] proposed a data provisioning strategy to localize Hadoop clusters proximal to data in high performance cloud computing environment. The strategy minimizes the data access delay in the environment by placing VM's in Hadoop cluster close to the storage node. By this way, the file transfer time and hence the MapReduce job completion time is reduced. The provisioning scheme does not consider collocating inter related data and tasks together to achieve maximal performance. Tao et al [21] proposed a dynamic Hadoop cluster on cloud infrastructure. The clusters were scaled based on load by varying the number of virtual machine. The VM were split to two categories of data storage and computing node. Computing nodes and storage node are placed proximal to minimize access delay. The tasks requesting same data are placed in computing node and data is placed in storage node. But the authors did not consider grouping data based on semantics and placing them on storage node. Due to this, the make span of Map reduce jobs increased in this work. Li et al [22] proposed two different data locality optimization for geo distributed clouds. In the first optimization scheme, tasks are assigned according to node locality, and access data of

non-node-locality tasks are migrated in advance by using the idle network bandwidth. In the second optimization scheme, hot files are predicted and synchronized among the nodes. The tasks were not grouped based on similarity on data access and thus the data transfer cost is higher in this approach. Gandomi et al [23] proposed a hybrid scheduling algorithm combining dynamic priority and localization. The algorithm aimed to increase the data locality rate and reduce completion time. With dynamic priority and proportional share assignment, tasks requesting same data are processed in batches. This reduces the makespan for related tasks. But the proposed scheme did not exploit the data collocation and grouping data based on semantics to maximize the performance gain. Choi et al [24] proposed a task scheduling algorithm to solve the performance problem due to input split consisting of multiple data blocks. The algorithm operates in two stages. In the first stage, tasks are classified to three types based on data proximity. In the second stage, classified tasks are assigned to nodes based on priority criteria. Task classification method proposed in this paper classifies task only based on data location in rack and neglects the data semantics. The performance of this method can be still improved by data grouping based on semantics. Convolbo et al [25] proposed a heuristic scheduling algorithm called GeoDis to optimize the makespan for data intensive jobs in geo distributed clouds. Authors formulated the task placement and data access as a linear programming problem and used heuristics linear problem solver to find optimal task placement schedule. But the solution is not scalable and computing complexity is higher. Xie et al [26] proposed a stochastic delay optimal algorithm called Pandas to reduce the makespan of data intensive tasks. It is task-level algorithm that specifies the priority among tasks of any data-processing phase by considering data locality. Pandas predict the contents for data blocks and create replicas to avoid contention. By this way task processing time is reduced at cost of minor storage overhead. The contention blocks are predicted only based on access frequency and performance can be still improved by using data semantic correlation to predict zero day blocks for replication. Li et al [27] proposed a performance aware scheduler (PAS) to schedule jobs in Hadoop clusters. The proposed solution automatically adjusts the scheduling policies to improve application performance and resource utilization. Multiple concurrent tasks are scheduled using different policies based on predicted job completion time. Greedy policy based adjustment of policy is done to maximize average job performance frequently. Task grouping did not consider data access similarity and data location. If these are considered, the makespan of tasks can be still reduced.

III. OPEN ISSUES

From the survey, following three gaps are identified in reducing the makespan of map reduce jobs in Hadoop clusters.

(i) Data placement

(ii) Data semantics based task grouping

(iii) Data locality based optimization

Data placement: Most of resource allocation algorithms assumed data were placed optimally and exploited data locality to optimize the task scheduling process. When the data placement is not optimal, the performance gain due to scheduling policies is limited. Traditional way of grouping small data by merging to files results in non optimal data placement. To solve this problem, the data placement process on storage node must be improved in the high performance data analytics environment. The data placement process must consider the access frequency and inherent data semantics to decide the proximity of data and number of replicas to be placed across storage nodes. The data placement process must also be dynamic to task arrival rates for streaming tasks in a geo distributed environment.

Data semantics based task grouping: In most of scheduling algorithms, the tasks were grouped and executed in batches. The grouping considered the storage node commonness property and did not consider fine grained data semantic correlation. By considering data semantics correlation in grouping, the intermediate data sharing across the tasks in same batch can be increased and the make span can be reduced further.

Data locality based optimization: The data locality based optimizations in the existing works replicated data based on popularity. They also predicted the data to be migrated to reduces the tasks access latency. These schemes were based only on history of data access. The performance of these schemes can be still improved by considering further factors like tasks similarity and data semantics correlation.

IV. RESEARCH DIRECTIONS

Based on the three research gaps identified a prospective solution architecture for improving the performance of Hadoop clusters in high performance data analytics environment is framed. The solution architecture is given in Figure 1.

The solution architecture has three layers. The first layer is data placement. In this layer, the incoming small files are analyzed for semantic correlation. Machine learning based semantic correlation analysis is employed to categorize the data. The categorization can be flat or hierarchical. Data is collocated in the storage based on their categorization at first level. Storage nodes are also grouped based on the data correlation across them at second level. The task considered for processing for batches are grouped based on data collocation from level of within storage node to across storage node at the second layer. By this grouping, the tasks has large amount of sharable intermediate data and this reduces the execution time of reduce jobs and hence the make span of map reduce jobs. The third layer decides the replication of data blocks or storage node as altogether based on the history of access latency and semantic relation between the access tasks and data semantics. The adaption to application access characteristics on data is made dynamic to achieve higher performance for applications.

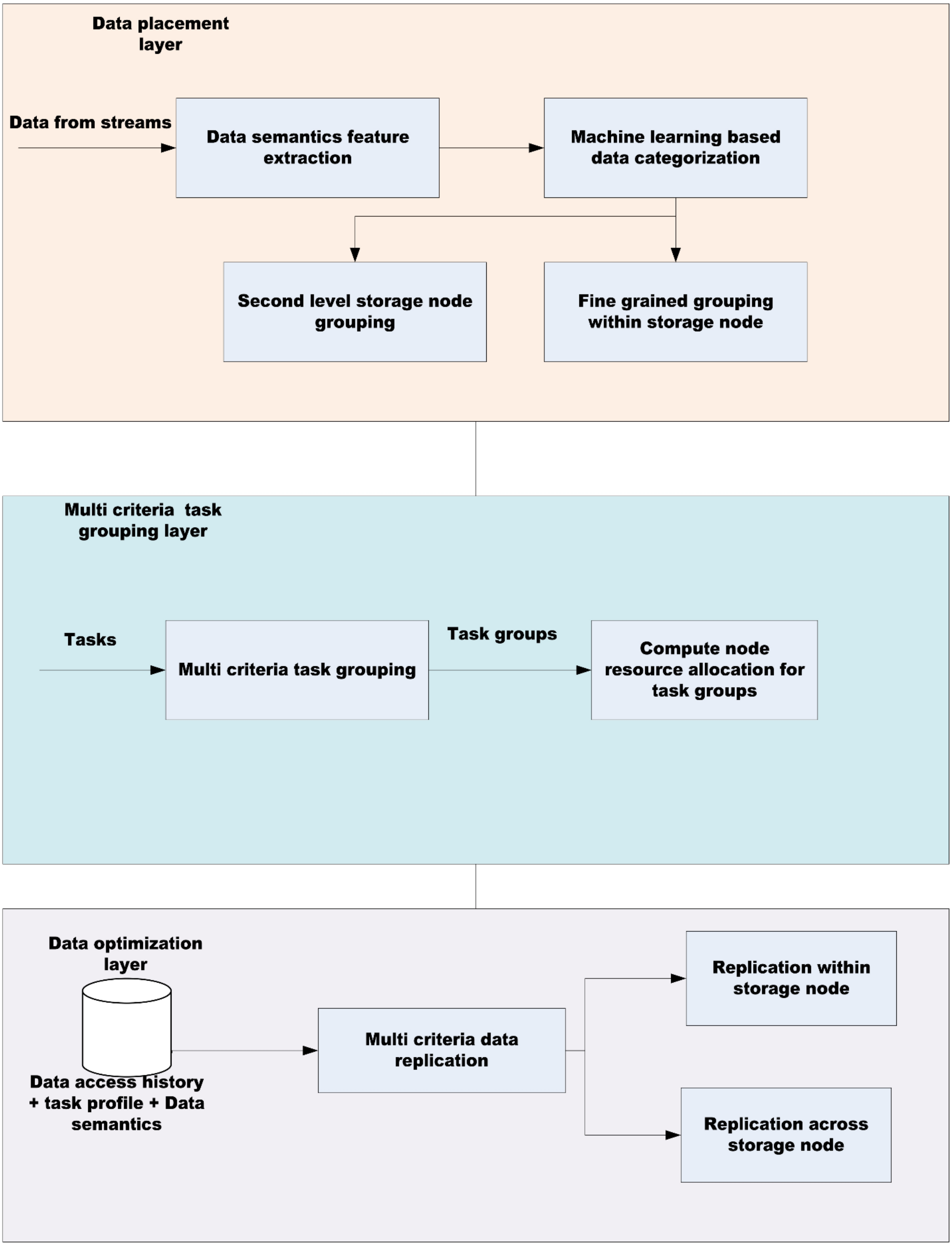
Data placement layer
Data from streams
Data semantics feature extraction
Machine learning based data categorization
Second level storage node grouping
Fine grained grouping within storage node
Multi criteria task grouping layer
Tasks
Multi criteria task grouping
Task groups
Compute node resource allocation for task groups
Data optimization layer
Data access history + task profile + Data semantics
Multi criteria data replication
Replication within storage node
Replication across storage node

Figure 1) Solution framework

V. CONCLUSION

A critical analysis of existing resource allocation and scheduling algorithms for Hadoop in high performance data analytics environment is presented in this work. Three research gaps in handling of data were identified to reduce the makespan of Hadoop map reduce maps. Based on the research gaps, solution framework is designed and its components were detailed. The salient advantages of the proposed framework were presented. The framework can be implemented and tested in high performance data analytics environment as part of future work.

VI. REFERENCES

[1]. Small size problem in Hadoop: http://blog.cloudera.com/blog/2009/02/the-small-files-problem/

[2]. Solving Small size problem in Hadoop https://pastiaro.wordpress.com/2013/06/05/solving-the-small-files-problem-in-apache-hadoop-appending-and-merging-in-hdfs/

[3]. Bo Dong , Qinghua Zheng, Feng Tian , Kuo-Ming Chao , Rui Ma, Rachid Anane.(2012), An optimized approach for storing and accessing small files on cloud storage, Journal of Network and Computer Applications, 35 (2012) 1847-1862, Elsevier

[4]. N. Lim, S. Majumdar and P. Ashwood-Smith, "MRCP-RM: A Technique for Resource Allocation and Scheduling of MapReduce Jobs with Deadlines," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 5, pp. 1375-1389, 1 May 2017

[5]. Y. Yao, H. Gao, J. Wang, B. Sheng and N. Mi, "New Scheduling Algorithms for Improving Performance and Resource Utilization in Hadoop YARN Clusters," in IEEE Transactions on Cloud Computing, vol. 9, no. 3, pp. 1158-1171, 1 July-Sept. 2021

[6]. C. -T. Chen, L. -J. Hung, S. -Y. Hsieh, R. Buyya and A. Y. Zomaya, "Heterogeneous Job Allocation Scheduler for Hadoop MapReduce Using Dynamic Grouping Integrated Neighboring Search," in IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 193-206, 1 Jan.-March 202

[7]. Hammoud, M., and Sakr, M. F., “Locality-aware reduce task scheduling for MapReduce,” In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CLOUDCOM’11), pp. 570–576, 2011.

[8]. Kousiouris, G., Cucinotta, T., and Varvarigou, T., “The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks,” Journal of Systems and Software, vol. 84, pp. 1270–1291, August 2011.

[9]. Xie, J., Yin, S., Ruan, X., Ding, Z., Tian, Y., Majors, J., Manzanares,A., and Qin., X., “Improving MapReduce performance through data placement in heterogeneous Hadoop clusters,” In Proceedings of 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1–9, 2010

[10]. Wang, W., Zhu, K., Ying, L., Tan, J., and Zhang, L., “MapTask scheduling in mapreduce with data locality: throughput and heavytraffic optimality,” IEEE/ACM Transactions on Networking, vol. 24, pp. 190–203, February 2016

[11]. N. Lim, S. Majumdar and P. Ashwood-Smith, "Techniques for Handling Error in User-Estimated Execution Times During Resource Management on Systems Processing MapReduce Jobs," 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, Spain, 2017, pp. 788-793

[12]. Yang, Allen & Wang, Jiayin & Mao, Ying & Yao, Yi & Mi, Ningfang & Sheng, Bo. (2021). Optimizing Internal Overlaps by Self-Adjusting Resource Allocation in Multi-Stage Computing Systems. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3089907.

[13]. Y. Liu, Y. Zeng and X. Piao, "High-Responsive Scheduling with MapReduce Performance Prediction on Hadoop YARN," 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Daegu, Korea (South), 2016, pp. 238-247

[14]. J. Bader, L. Thamsen, S. Kulagina, J. Will, H. Meyerhenke and O. Kao, "Tarema: Adaptive Resource Allocation for Scalable Scientific Workflows in Heterogeneous Clusters," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 65-75

[15]. Marquez, Jack, Oscar H. Mondragon, and Juan D. Gonzalez. 2021. "An Intelligent Approach to Resource Allocation on Heterogeneous Cloud Infrastructures" Applied Sciences 11, no. 21: 9940

[16]. X. Zhang, Y. Feng, S. Feng, J. Fan and Z. Ming, "An effective data locality aware task scheduling method for MapReduce framework in heterogeneous environments," 2011 International Conference on Cloud and Service Computing, Hong Kong, China, 2011, pp. 235-242

[17]. Jeyaraj, R., Ananthanarayana, V.S. & Paul, A. Fine-grained data-locality aware MapReduce job scheduler in a virtualized environment. J Ambient Intell Human Comput 11, 4261–4272 (2020).

[18]. T. -Y. Chen, H. -W. Wei, M. -F. Wei, Y. -J. Chen, T. -s. Hsu and W. -K. Shih, "LaSA: A locality-aware scheduling algorithm for Hadoop-MapReduce resource assignment," 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 2013, pp. 342-346

[19]. Wei, H., Wu, T.Y., Lee, W., & Hsu, C. (2015). Shareability and Locality Aware Scheduling Algorithm in Hadoop for Mobile Cloud Computing. J. Inf. Hiding Multim. Signal Process., 6, 1215-1230.

[20]. Muhamad Amin, Anang Hudaya & Ahmad, Nazrul & Kannan, Subarmaniam. (2016). Data location aware scheduling for virtual Hadoop cluster deployment on private cloud computing environment. 10.1109/APCC.2016.7581422.

[21]. Tao D, Lin Z, Wang B. Load Feedback-Based Resource Scheduling and Dynamic Migration-Based Data Locality for Virtual Hadoop Clusters in OpenStack-Based Clouds. Tsinghua Science and Technology, 2017, 22(2): 149-159

[22]. Li, Chunlin & Zhang, Jing & Tao, Ma & Tang, Hengliang & Lei, Zhang & Luo, Youlong. (2018). Data locality optimization based on data migration and hotspots prediction in geo-distributed cloud environment. Knowledge-Based Systems. 165. 10.1016/j.knosys.2018.12.002.

[23]. Gandomi, A., Reshadi, M., Movaghar, A. et al. HybSMRP: a hybrid scheduling algorithm in Hadoop MapReduce framework. J Big Data 6, 106 (2019).

[24]. D. Choi, M. Jeon, N. Kim and B. -D. Lee, "An Enhanced Data-Locality-Aware Task Scheduling Algorithm for Hadoop Applications," in IEEE Systems Journal, vol. 12, no. 4, pp. 3346-3357, Dec. 2018

[25]. Convolbo, M.W., Chou, J., Hsu, CH. et al. GEODIS: towards the optimization of data locality-aware job scheduling in geo-distributed data centers. Computing 100, 21–46 (2018).

[26]. Xie, Q., Pundir, M., Lu, Y., Abad, C. L., & Campbell, R. H. (2017). Pandas: Robust Locality-Aware Scheduling with Stochastic Delay Optimality. IEEE/ACM Transactions on Networking, 25(2), 662-675,2017

[27]. Yiren Li, Tieke Li, Pei Shen, Liang Hao, Jin Yang, Zhengtong Zhang, Junhao Chen, Liang Bao, "PAS: Performance-Aware Job Scheduling for Big Data Processing Systems", Security and Communication Networks, vol. 2022, Article ID 8598305, 14 pages, 2022.

Cite this article as :

Prof. Shwetha K S, Dr. Chandramouli H, "Survey on Resource Management Solutions to Speed up Processing Small Files in Hadoop Cluster", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 6, pp. 164-172, November-December 2023. Available at doi : https://doi.org/10.32628/IJSRSET3214668
Journal URL : https://ijsrset.com/IJSRSET3214668