IJIRIS: International Journal of Innovative Research in Information Security
Volume 10, Issue 04, May 2024
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives

Survey on MapReduce Scheduler Algorithms in Hadoop Framework

Kesavan MV
Assistant Professor, Department of CSE
East Point College of Engineering and Technology,
Bengaluru, INDIA
kesavan.mv@eastpoint.ac.in
Dr.Josephine PremKumar
Professor, Department of CSE
Cambridge Institute of Technology,
Bengaluru, INDIA
josephine.prem@gmail.com
Manimegalai A
Assistant Professor, Department of CSE
East Point College of Engineering and Technology,
Bengaluru, INDIA
manimegalai.a@eastpoint.ac.in

Publication History
Manuscript Reference No: IJIRIS/RS/Vol.10/Issue04/MYIS10117
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRIS/RS/Vol.10/Issue04/MYIS10117
Received: 07, April 2024 | Revised:16, April 2024 | Accepted: 27, April 2024 Published Online: 07, May 2024 Volume2024
Article ID MYIS10117 http://www.ijiris.com/volumes/Vol10/iss-04/38.MYIS10117.pdf
Article Citation: Kesavan,Josephine,Manimegalai(2024). Survey on MapReduce Scheduler Algorithms in Hadoop Framework. International Journal of Innovative Research in Information Security, Volume 10, Issue 04, Pages 314-319
doi:> https://doi.org/10.26562/ijiris.2024.v1004.38
BibTex key: Kesavan@2024Survey

Copyright: ©2024 This is an open access article distributed under the terms of the Creative Commons Attribution License; which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Hadoop serves as a robust framework tailored for the storage and processing of vast data volumes across clusters. Its foundation lies in the Hadoop Distributed File System (HDFS) for data storage, complemented by the MapReduce paradigm for data processing. MapReduce, functioning as a parallel computing framework, operates adeptly on distributed clusters to manage large-scale data processing tasks. Within this framework, scheduling emerges as a pivotal aspect, influencing the overall system's performance and efficiency. The essence of scheduling in MapReduce lies in enhancing performance, minimizing response times, and optimizing resource allocation. This paper undertakes a systematic exploration of existing scheduling algorithms, offering a fresh classification and detailed examination of each category. Furthermore, the analysis delves into the core principles, objectives, as well as the strengths and weaknesses of these scheduling algorithms.

Index Terms: Hadoop Map reduce, Scheduling Algorithms, Resource Allocation, HDFS, Parallel Computing

I. INTRODUCTION

The Explosion of data from social media, transactions, sensors, and mobile devices is overwhelming traditional data management tools [1, 2]. This surge, characterized by volume, velocity, and variety, creates complexities in processing data for accuracy, transformation, and analysis [3]. To tackle this challenge, new platforms are needed to efficiently store, transmit, and analyze these massive and often unstructured datasets. The answer lies in parallel and distributed processing, which breaks down large tasks across multiple computers (clusters) for faster results [4]. To make this complex system user-friendly, various frameworks have been developed, simplifying big data processing for businesses and organizations. MapReduce is a programming pattern that is popular among all frameworks. Using the Map and Reduce functions of MapReduce, users do not have to worry about the details of parallelism when defining parallel processes, such as data distribution, load balancing, and fault tolerance [6]. MapReduce is used to process high volumes of data concurrently. Map and Reduce are the two functions of MapReduce. In this framework, the first step in parallel computing is to assign map tasks to various nodes and perform them on input data. Then, the final results are generated by combining the map outputs and applying the reduce function [7, 8].MapReduce is in competition with other program paradigms like Spark and Data MPI. The choice of MapReduce for investigation is based on the fact that it is high-performance open source, utilized by many large companies to process batch jobs [9,10], and is our future research area. Scheduling is the process of assigning tasks to the nodes, which is a critical factor for improving system performance in MapReduce. There are many algorithms to solve scheduling issues with different techniques and approaches. The main goal of scheduling is to increase throughput while reducing response time and improving performance and resource utilization [11,12].

Our research provides a comprehensive and up-to-date survey of MapReduce scheduling algorithms, building upon the previous systematic review conducted [13]. We employed rigorous selection criteria to analyze relevant articles published between 2017 and June 2023. This survey fills the gap in existing literature. Key objectives of this study are review existing scheduling algorithms: We analyze current algorithms, compare their functionalities, and delve into specific details of several prominent ones and Identify trends and challenges: This survey explores the current landscape of MapReduce scheduling, pinpointing both established practices and areas needing further exploration. The paper is structured as Background and Research Method: This section provides an architectural overview of MapReduce and Hadoop (Part 1) followed by a detailed explanation of our research methodology (Part 2).Regular Surveys: This section discusses and categorizes existing schedulers within Hadoop MapReduce. We also present a comparative analysis of selected algorithms. Schedulers in Hadoop MapReduce: Here, we delve deeper into the discussed schedulers and analyze the obtained results. Discussion: This section concludes the paper by summarizing our findings and highlighting potential future directions for research.

2. BACKGROUND

Hadoop, an open-source project inspired by Google's MapReduce approach, is specifically designed for distributed processing of large datasets [14]. It comprises two key components: Hadoop Distributed File System (HDFS): This acts as the storage layer for Hadoop, using a master-slave architecture with a central Name Node managing data location and multiple Data Nodes storing the actual data in a distributed fashion [15, 16].MapReduce: This is the processing engine of Hadoop. It breaks down large jobs into smaller tasks (Map and Reduce) that can be executed in parallel across multiple machines (Task Trackers) overseen by a Job Tracker [17, 18]. Figure 1 illustrates this architecture. MapReduce follows a specific execution flow (detailed in Figure 2): Map Tasks: Input data is split and processed in parallel by Map tasks, generating intermediate key-value pairs. Shuffle and Sort: These key-value pairs are shuffled and sorted based on the key.

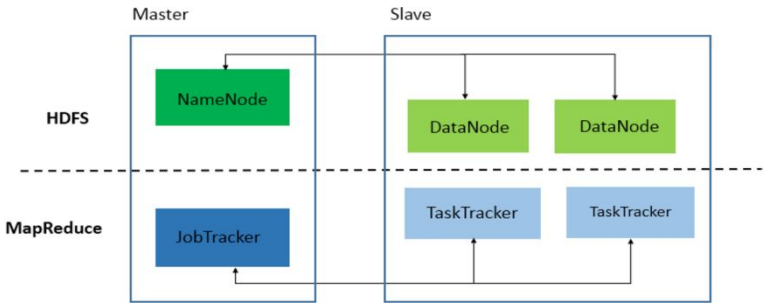

Fig 1. Hadoop Architecture

3. REGULAR SURVEYS

Our analysis aims to provide a holistic understanding of job scheduler surveys by comprehensively examining and summarizing existing literature. Ghazalietal.[23] presented an ovel classification sys-tem for job schedulers, categorizing them into three distinct groups: job schedulers for mitigating stragglers, job schedulers for enhancing data locality, and job schedulers for optimizing resource utilization. For each job scheduler within these groups, they provided a detailed explanation of their performance-enhancing approach and conducted evaluations to identify their strengths and weaknesses. Additionally, the impact of each scheduler was assessed, and recommendations were offered for selecting the best option in each category. Finally, the authors provided valuable guidelines for choosing the most suitable job scheduler based on specific environmental factors. However, the survey is not systematic and the process of selecting articles is not clear. Moreover, there is no information about the environment and the platform for implementation or simulation of the surveyed articles.

Abdallat et al. [24] focused on the topic of job scheduling algorithms in BigData Hadoop environment. The authors emphasize the importance of efficient job scheduling in processing large amounts of data in real-time, considering the limitations of traditional ecosystems. The paper provides background information on the Hadoop MapReduce framework and con-ducts a comparative analysis of different job scheduling algorithms based on various criteria such as cluster environment, job allocation strategy, optimization strategy, and quality metrics. The authors present use cases to illustrate the characteristics of selected algorithms and offer a comparative display of their details. The paper discusses popular scheduling considerations, including locality, synchronization, and fairness, and evaluates Hadoop schedulers based on metrics such as locality, response time, and fairness. However, the survey is not systematic and the process of selecting articles is not clear. Also, there is no comparison between these studies' advantages and disadvantages, and there is no information about the environment and the platform for implementation or simulation of the surveyed articles.

Hashem et al. [25] conducted a comprehensive comparison of resource scheduling mechanisms in three widely-used frameworks: Hadoop, Mesos, and Corona. The scheduling algorithms within MapReduce were systematically categorized based on strategies, resources, workload, optimization approaches, requirements, and speculative execution. The analysis encompassed two aspects: taxonomy and performance evaluation, where they thoroughly reviewed the advancements made in MapReduce scheduling algorithms.

Additionally, the authors highlighted existing limitations and identified potential areas for future research in this domain. However, the survey is not systematic and the process of selecting articles is not clear. Also, there is no information about the environment and the platform for implementation or simulation of the surveyed articles.

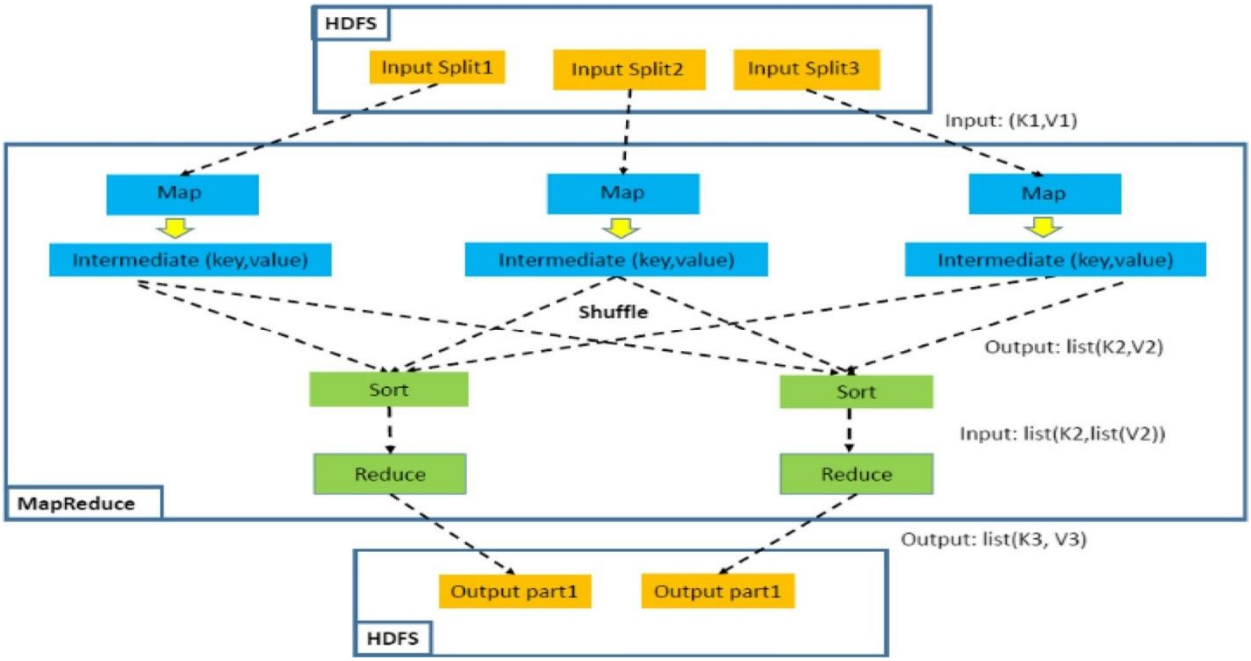

Fig 2 Map Reduce Job execution flow

Hashem et al. [25] conducted a comprehensive comparison of resource scheduling mechanisms in three widely used frameworks: Hadoop, Mesos, and Corona. The scheduling algorithms within MapReduce were systematically categorized based on strategies, resources, workload, optimization approaches, requirements, and speculative execution. The analysis encompassed two aspects: taxonomy and performance evaluation, where they thoroughly reviewed the advancements made in MapReduce scheduling algorithms. Additionally, the authors highlighted existing limitations and identified potential areas for future research in this domain

4. MAP REDUCE SCHEDULER ALGORITHMS

4.1 Schedulers in Hadoop MapReduce

We classified the schedulers into six categories:

1. Deadline-aware scfledulers;
2. DataLocality-aware scfledulers;
3. Cost-aware scfledulers;
4. Resource-aware scfledulers;
5. Makes pan-aware scfledulers;
6. Learning-aware Scfledulers

The idea of each paper has been validated by comparing the performance against existing solutions and benchmarks.

4.1.1 Deadline aware schedulers

Some Map Reduce jobs on big data platforms have dead-lines and need to be completed within those deadlines. When a job has a dead line, the proper resources must be allocated to the job; otherwise, the deadline cannot be satisfied. Therefore, meeting the job deadline is crucial in MapReduce clusters. We classify deadline-aware schedulers into two categories: deadline-aware schedulers in heterogeneous Hadoop Clusters and deadline-aware schedulers in homogeneous Hadoop clusters. In this section, we first survey and review the most popular deadline-aware schedulers, which minimize job deadline misses. Finally, the reviewed schedulers are compared and summarized.

4.1.2 Data locality aware schedulers

In data locality-aware schedulers, tasks are allocated to the node where the task's input data is stored; otherwise, they are assigned to the node closest to the data node[5]. Researchers proposed several scheduling algorithms to improve data locality because it minimizes data transfer over the network and mitigates the total execution time of tasks, thus improving the Hadoop performance[4]. Therefore, improving data locality is a crucial problem in MapReduce clusters. In this section, we review several important data locality-aware schedulers.

4.1.3 Cost aware schedulers

In big data platforms, data centers store a huge amount of data. Processing this data requires thousands of nodes in Hadoop clusters. Such large clusters consume enormous amounts of power and increase the cost of datacenters. Therefore, we face a big challenge in minimizing cost in MapReduce clusters[31].

In this section, we survey and review several important cost-aware schedulers that reduce the cost of MapReduce systems. Finally, the reviewed schedulers are compared and summarized.

4.1.4 Resource aware schedulers

In big data applications, datacenters are deployed on large Hadoop clusters. The nodes in these clusters receive a large number of jobs and require more resources to execute them. As a result, race condition arises among the jobs that demand resources like CPU, memory, and I/O [9]. Therefore, improving cluster resource utilization has become a major concern in MapReduce clusters. This section presents some of the most popular resource-aware schedulers which increase resource utilization

4.1.5 Make span aware schedulers

The make span (total completion time) of a set of jobs is the total amount of time it takes to complete jobs. In order to increase the cluster's performance, make span needs to be minimized by distributing the data across the nodes. Also, low make span is a major factor for any scheduler[4]. Therefore, minimizing the make span has become an important issue in MapReduce clusters. In this section, we first review several important make span-aware schedulers. Then, the reviewed schedulers are compared and summarized.

4.1.6 Learning aware schedulers

In this section, we first review several important learning-aware schedulers. Then, the reviewed schedulers are compared and summarized. Ghazali et al. [8] focused on the scheduling of Map Reduce jobs in Hadoop and specifically addresses the importance of data and cache locality in improving performance

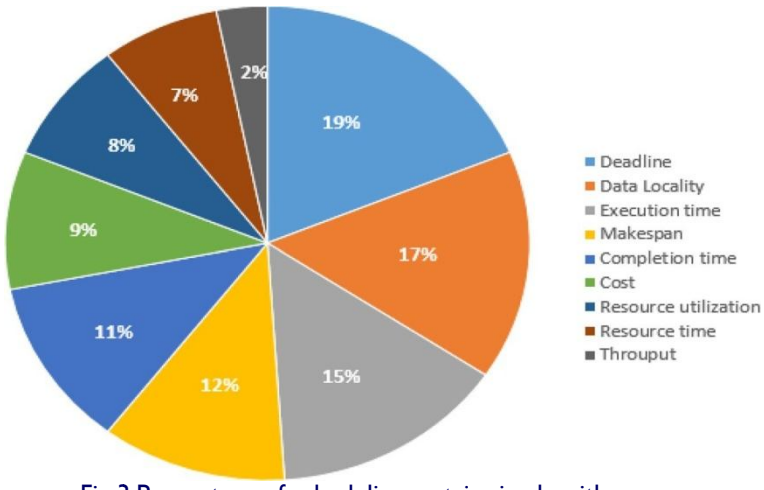

Fig 3.Percentage of scheduling metrics in algorithms

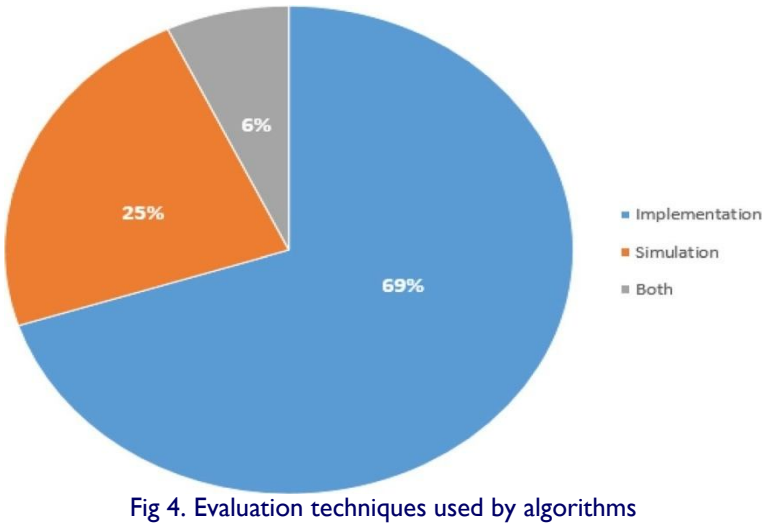

Fig 4. Evaluation techniques used by algorithms

5. DISCUSSION

A comparative analysis of different scheduling metrics in HadoopMap Reduce is presented in this section. We showed how the selected studies addressed the scheduling metrics. Figure3 demonstrates that 19% of the algorithms studied used the deadline metric,17% used the data locality metric, 15% addressed the execution time metric, 12% addressed the make span metric,11% used the completion time metric, 9% used the cost metric, 8% used the response time metric, 7% used the resource utilization metric, and 2% addressed the throughput. It is shown that the majority of algorithms have focused on the dead-line and data locality metrics.

Figure 4 shows the evaluation techniques used in the selected studies. A scan be seen,69% of the studies used implementation, which is the highest; 6% of them used simulation, and 25% of them used both implementation and simulation.

6. CONCLUSION

Scheduling in Hadoop MapReduce is an important challenge that Hadoop systems are facing. In this paper, we provided a comprehensive systematic study in HadoopMapReduce. First, an overview of Hadoop major components is presented. We thoroughly reviewed and analyzed individually the selected MapReduce scheduling algorithms. Based on our research method, we classify these into six categories: deadline-aware schedulers, data locality-aware schedulers, cost-aware schedulers, resource-aware schedulers, make spanaware schedulers, and learning-aware schedulers. We compared the studies in terms of key ideas, main objectives, advantages, disadvantages, comparison algorithms, and evaluation techniques. We concluded that Makespan aware scheduler algorithm is most suitable for parallel processing environment to process huge amount of data sets.

REFRENCES

[1]. AssunçãoM Detal (2015) Big Data computing and clouds: Trends and future directions. J Parallel Distributed Comput 79:3–15
[2]. Thusoo A et al (2010) "Hive-a peta byte scale data warehouse using hadoop."2010 IEEE26 th International conference on data engineering (ICDE2010).IEEE
[3]. Deshai N et al (2019) "Big data Hadoop MapReduce job scheduling: A short survey." Information Systems Design and Intelligent Applications: Proceedings of Fifth International Conference INDIA 2018 Volume 1.Springer,Singapore
[4]. HuHetal(2014) Toward scalable systems for big data analytics: A technology tutorial. IEEE Access2:652–687
[5]. ChenCP, Zhang C-Y(2014) Data intensive applications, challenges, techniques and technologies: A survey on Big Data. InfSci275:314–347
[6]. Chen M, MaoS,LiuY(2014) Bigdata: A survey.MobileNetwAppl 19:171–209
[7]. DeanJ,GhemawatS (2008)MapReduce: simplified dataprocessing on large clusters. CommunACM51(1):107–113
[8]. BakniN-E and IAssayad(2021) Survey on improving the performance of MapReduce in Hadoop. In: Proceedings of the 4th International Conference on Networking, Information Systems & Security
[9]. ZhangB,WangX,ZhengZ(2018) The optimization for recurring queries in big data analysis system with MapReduce. Futur Gener Comput Syst87:549–556
[10]. Kashgarani H,KotthoffL(2021)" Is algorithm selection worth it ? Comparing selecting single algorithms and parallel execution." AAAI Workshop on Meta-Learning and MetaDL Challege. PMLR
[11]. PakizeSR(2014) A comprehensive view of Hadoop MapReduce scheduling algorithms. Int J Comput Netw Commun Secur 2(9): 308–317
[12]. KangY,PanL,LiuS(2022) Jobscheduling for bigdata analytical applications in clouds: A taxonomy study. Futur Gener Comput Syst135:129–145
[13]. Bhosale HS,Gadekar DP(2014) Bigdata processing using hadoop: survey on scheduling. IntJSciRes3(10):272–277
[14]. ShvachkoKetal (2010)" The hadoop distributed file system."2010IEEE26th symposium on mass storage systems and technologies (MSST). Ieee
[15]. KhushbooK,GuptaN(2021)"Analysis of hadoop MapReduce scheduling in heterogeneous environment." Ain Shams Engineering Journal12(1):1101–1110
[16]. White T(2012) Hadoop: The definitive guide."O'ReillyMedia,Inc."
[17]. LuZetal (2018) IoT DeM: An IoT big data-oriented Map Reduce perfor-mance prediction extended model in multiple edge clouds. J Parallel Distributed Comput118:316–327
[18]. SinghR, KaurPJ (2016) Analyzing performance of ApacheTez and MapReduce with hadoop multi node cluster n Amazon cloud. J BigData3(1):1–10
[19]. WangHetal(2015)BeTL: MapReduce check point tactics beneath the task level .IEEE Trans Serv Comput9(1):84–95
[20]. Petersen K,Vakkalanka S,Kuzniarz L(2015) Guidelines for conducting systematic mapping studies in software engineering: An update. Inf SoftwTechnol64:1–18
[21]. Cruz-BenitoJ (2016) Systematic literature review & mapping
[22]. Lu Q et al (2015) "MapReduce job optimization: a mapping study." 2015 International Conference on Cloud Computing and BigData(CCBD).IEEE
[23]. GhazaliRetal(2021)A classification of Hadoop job schedulers based on performance optimization approaches. ClustComput24(4):3381–3403
[24]. AbdallatA A,AlahmadAI ,AlWidianJA(2019) Hadoopmap reduce job scheduling algorithms survey and use cases. ModApplSci13(7):1–38
[25]. HashemIA Tetal(2020) MapReduce scheduling algorithms: are view.J Supercomput76:4915–4945
[26]. SoualhiaM, KhomhF, TaharS(2017) Task scheduling in bigdata plat-forms: a systematic literature review. JSystSoftw134:170–189
[27]. KhezrSN,NavimipourNJ(2017) Map Reduce and its applications, challenges, and architecture: a comprehensive review and directions for future research.JGridComput15:295–321

[28]. Senthilkumar M,Ilango P (2016)A survey on job scheduling in bigdata. Cybern Inf Technol16(3):35–51
[29]. HashemIATetal(2016) Map Reduce: Review and open challenges.Scien-tometrics109:389–422
[30]. LiRetal(2016)MapReduce parallel programming model: a state-of-the-art survey. IntJParallelProg44:832–866
[31]. TiwariNetal(2015) Classification frame work of MapReduce scheduling algorithms. ACM Comput Surveys (CSUR)47(3):1–38
[32]. Polatoletal(2014) A comprehensive view of Hadoop research A systematic literature review. JNetw ComputAppl46:1–25
[33]. Gao Y, Zhang K (2022) "Deadline-aware preemptive job scheduling in hadoopy arn clusters."2022 IEEE25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE
[34]. Cheng D et al (2018) Deadline-aware MapReduce job scheduling with dynamic resource availability. IEEE Trans Parallel Distrib Syst 30(4):814–826
[35]. KaoY-C,ChenY-S(2016) Data-locality-aware map reduce real-time scheduling framework. JSystSoftw112:65–77
[36]. Verma A et al (2012) "Deadline-based workload management for MapReduce environments: Pieces of the performance puzzle."2012I EEE Network Operations and Management Symposium. IEEE
[37]. Phan LT et al (2011) "An empirical analysis of scheduling techniques for real-time cloud-based data processing." 2011 IEEE International Conference on Service-Oriented Computing and Applications(SOCA).IEEE
[38]. KcK,AnyanwuK(2010)" Scheduling hadoop jobs to meet deadlines."2010 IEEE Second International Conference on Cloud Computing Technology and Science. IEEE
[39]. Teng F et al (2014) A novel real-time scheduling algorithm and performance analysis of a MapReduce-based cloud. J Supercomput 69(2):739–765
[40]. WangXetal(2015)SAMES: deadline constraint scheduling in MapReduce. FrontCompSci9:128–141
[41]. DongX, WangY, LiaoH (2011)"Scheduling mixed realtime and non real-time applications in mapreduce environment." 2011 IEEE 17th International Conference on Parallel and Distributed Systems. IEEE
[42]. VermaA, CherkasovaL, CampbellRH(2011)" Resource provisioning frame work for mapreduce jobs with performance goals." Middleware 2011:ACM/IFIP/USENIX12th International Middleware Conference, Lisbon, Portugal ,December12-16,2011.Proceedings12.Springer Berlin Heidelberg
[43]. JabbariAetal(2021)" A Cost-Efficient Resource Provisioning and Scheduling Approach for Deadline-Sensitive MapReduce Computations in Cloud Environment. "2021 IEEE 14th International Conference on Cloud Computing (CLOUD).IEEE
[44]. ShaoYetal(2018) Efficient jobs scheduling approach for bigdata applications. ComputIndEng117:249–261
[45]. LinJ-W,ArulJM,LinC-Y(2019) Joint deadline constrained and influence-aware design for allocating MapReduce jobs in cloud computing systems. Clust Comput22:6963–6976
[46]. ChenC-H,LinJ-W,KuoS-Y(2015) MapReduce scheduling for dead line-constrained jobs in heterogeneous cloud computing systems. IEEE Trans Cloud Comput6(1):127–140
[47]. Tang Zetal(2013) A Map Reduce task scheduling algorithm for deadline constraints. Clust Comput16:651–662
[48]. Verma AL, Cherkasova, and RH Campbell (2011) Aria: automatic resource inference and allocation for map reduce environments. In: Proceedings of the 8th ACM international conference on Autonomic computing
[49]. PoloJetal(2013) Deadline-based MapReduce workload management. IEEE TransNetwServManage10(2):231–244
[50]. KaliaKetal(2022) Improving MapReduce heterogeneous performance using KNN fair share scheduling. RobotAutonSyst157:104228