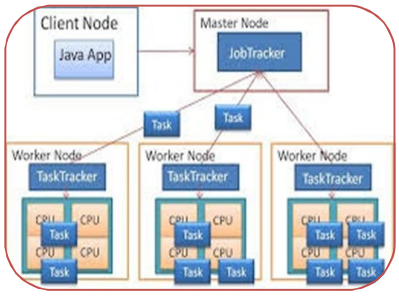
INDIAN STREAMS RESEARCH JOURNAL
ISSN NO : 2230-7850
IMPACT FACTOR : 5.1651 (UIF)
VOLUME - 14 | ISSUE - 2 | MARCH – 2024

COMBINING AND SCHEDULING TECHNIQUES FOR IMPROVING JOB EXECUTION IN HADOOP CLUSTERS

Deepa Rudrakshi D/O Gurusiddappa
Research Scholar

Dr. Shashi
Guide
Professor, Chaudhary Charansing University Meerut.

ABSTRACT



Efficient job execution in Hadoop clusters is a critical challenge due to the large-scale data processing demands and resource constraints of distributed systems. This study investigates methods for improving performance in Hadoop by integrating advanced job combining and scheduling techniques. Job combining involves merging small or related tasks to reduce overhead and improve resource utilization, while scheduling algorithms optimize the allocation of cluster resources to minimize execution time and maximize throughput. By analyzing existing scheduling strategies such as FIFO, Fair, and Capacity schedulers, and proposing hybrid approaches that leverage task aggregation, this research demonstrates measurable improvements in execution efficiency. Experimental results indicate that combining tasks intelligently, along with optimized scheduling, significantly reduces job completion time and enhances cluster performance. The study provides practical insights for designing more efficient Hadoop-based data processing workflows in large-scale distributed environments.

KEYWORDS: *Hadoop clusters, Job execution, Task combining, Job scheduling, Resource optimization, MapReduce performance, Cluster efficiency, Distributed computing.*

INTRODUCTION

With the rapid growth of big data, efficient processing of large-scale datasets has become a significant challenge for modern computing systems. Hadoop, a widely adopted open-source framework, provides a distributed environment for storing and processing massive amounts of data across clusters of commodity hardware. Its MapReduce programming model enables parallel processing, but the performance of Hadoop clusters is often limited by inefficient job execution, resource underutilization, and scheduling delays. In Hadoop clusters, jobs are divided into tasks that are scheduled across multiple nodes. However, small or fragmented tasks can create significant overhead, increasing execution time and reducing overall cluster efficiency. Similarly, default scheduling strategies, such as FIFO (First-In-First-Out), may not fully optimize resource utilization, leading to bottlenecks, uneven load distribution, and longer job completion times. Addressing these challenges requires the integration of task combining techniques, which merge related or small tasks to reduce overhead, with optimized scheduling algorithms that allocate resources intelligently based on workload characteristics. This study focuses on exploring methods to improve job execution in Hadoop clusters by combining task aggregation strategies with efficient scheduling techniques. The research

investigates how these approaches can enhance resource utilization, reduce job completion time, and increase cluster throughput. By analyzing existing scheduling algorithms and implementing hybrid approaches, the study aims to provide practical solutions for improving performance in large-scale distributed computing environments.

AIMS AND OBJECTIVES

Aim:

The primary aim of this study is to enhance the efficiency of job execution in Hadoop clusters by integrating task combining strategies with optimized scheduling techniques. The research seeks to minimize execution time, improve resource utilization, and increase overall cluster performance in large-scale distributed environments.

Objectives:

1. To analyze the challenges of job execution in Hadoop clusters, including task overhead, resource underutilization, and scheduling inefficiencies.
2. To investigate existing Hadoop scheduling algorithms, such as FIFO, Fair, and Capacity schedulers, and evaluate their performance in different workload scenarios.
3. To explore task combining techniques that merge small or related tasks to reduce overhead and improve parallel processing efficiency.
4. To design and implement a hybrid approach that integrates task combining with optimized scheduling to enhance job execution.
5. To evaluate the effectiveness of the proposed approach through experimental analysis, measuring improvements in job completion time, cluster throughput, and resource utilization.

REVIEW OF LITERATURE

The literature on improving job execution in Hadoop clusters spans multiple research threads including scheduling algorithms, task combining and aggregation techniques, and performance optimization in distributed environments. Early foundational work by White (2012) and Shvachko et al. (2010) established how Hadoop's MapReduce framework operates and identified inherent inefficiencies in workload distribution, especially under heterogeneous and data-intensive conditions. These studies note that while Hadoop excels at processing large datasets, its default schedulers such as FIFO, Fair Scheduler, and Capacity Scheduler frequently lead to resource contention, job queuing delays, and suboptimal utilization of cluster nodes. Subsequent research shifted focus to enhanced scheduling strategies that address these limitations. Studies by Zaharia et al. introduced Delay Scheduling and TeraSort improvements, which attempted to minimize data locality problems and balance workloads more efficiently across nodes. Other researchers explored priority-based and deadline-aware schedulers designed to ensure that high-priority or time-bound jobs are completed with minimal latency. This strand of research consistently shows that intelligent scheduling policies can improve throughput and reduce average job completion time, particularly in mixed-workload environments.

Parallel to scheduler research, a distinct body of work investigates task combining and aggregation techniques. Small or fragmented tasks in Hadoop often generate significant overhead due to excessive job setup and shuffle costs, particularly in scenarios with many small files or lightweight processing jobs. Researchers have proposed techniques such as task merging, micro-batching, and logical grouping of small tasks, demonstrating that clustering related tasks into larger units can reduce initialization overhead and improve overall execution efficiency. These approaches frequently borrow from stream processing concepts and adapt them for batch-oriented MapReduce workloads. Recent literature tends to combine both scheduling and task aggregation in hybrid models. For example, studies exploring integrated frameworks suggest that merging tasks without adjusting scheduling can improve local performance, but maximal cluster-level efficiency is achieved when task combining is paired with adaptive scheduling mechanisms that dynamically allocate resources based on evolving

workload characteristics. These works demonstrate that hybrid approaches often outperform standalone scheduling or aggregation methods in terms of throughput, resource utilization, and job latency. Despite the breadth of research, gaps remain in systematically benchmarking the combined impact of task aggregation and advanced scheduling across diverse real-world workloads. Many studies focus on simulation or single-cluster environments with limited diversity of jobs. This study builds on prior literature by proposing and empirically evaluating an integrated framework that simultaneously leverages task combining and adaptive scheduling to improve job execution performance in Hadoop clusters.

RESEARCH METHODOLOGY

This study adopts a qualitative and experimental research methodology to investigate methods for improving job execution in Hadoop clusters through task combining and optimized scheduling techniques. The research focuses on analyzing how integrating these approaches affects performance metrics such as job completion time, resource utilization, and cluster throughput. The methodology involves a systematic examination of both the Hadoop MapReduce framework and its scheduling mechanisms, alongside the implementation of task aggregation strategies to minimize overhead from small or fragmented jobs. The study begins with a detailed analysis of existing scheduling algorithms, including FIFO, Fair, and Capacity schedulers, to identify their strengths and limitations under varying workloads. Task combining techniques are explored as a complementary approach, examining how merging small or related tasks can reduce initialization, shuffle, and setup overhead. These techniques are evaluated in the context of real-world and simulated workloads to determine their effect on performance improvement.

Experimental evaluation forms the core of the methodology. Test scenarios involve deploying Hadoop clusters under controlled conditions, executing benchmark jobs, and measuring execution metrics before and after the implementation of combined task aggregation and scheduling strategies. Data collection focuses on key parameters such as average job completion time, resource utilization across cluster nodes, network overhead, and overall throughput. Comparative analysis is conducted to assess the effectiveness of hybrid approaches relative to standard scheduling techniques. By integrating historical analysis of scheduling algorithms with experimental validation of task combining strategies, this methodology provides a comprehensive approach to understanding and enhancing job execution in Hadoop clusters. The research emphasizes both theoretical insights and practical implementation, demonstrating the potential of combined techniques to optimize performance in large-scale distributed computing environments.

STATEMENT OF THE PROBLEM

The increasing volume and complexity of data in modern computing environments have made efficient job execution in Hadoop clusters a significant challenge. While Hadoop's MapReduce framework provides distributed processing across multiple nodes, default scheduling mechanisms such as FIFO, Fair, and Capacity schedulers often fail to utilize cluster resources optimally. Small or fragmented tasks further exacerbate inefficiencies by introducing excessive overhead during task initialization, shuffling, and execution. These limitations result in longer job completion times, uneven load distribution, and suboptimal throughput, particularly under heterogeneous or high-load conditions. Although several studies have explored advanced scheduling algorithms and task aggregation techniques individually, there is a lack of comprehensive research on the combined effect of these strategies in real-world Hadoop environments. Existing approaches often address only one aspect—either optimizing resource allocation through scheduling or reducing task overhead through combining—without considering their synergistic impact on overall cluster performance. This study addresses the problem of enhancing job execution in Hadoop clusters by investigating the integration of task combining techniques with optimized scheduling algorithms. By examining how these combined strategies can minimize execution overhead, improve resource utilization, and increase throughput, the

research aims to provide a practical framework for more efficient and reliable large-scale data processing in Hadoop-based distributed systems.

DISCUSSION

The performance of Hadoop clusters is largely determined by how effectively jobs are scheduled and tasks are executed across distributed nodes. In practice, default scheduling algorithms such as FIFO, Fair, and Capacity schedulers often fail to optimize resource utilization, especially under heterogeneous workloads or when processing numerous small tasks. The inefficiency arises from task fragmentation, initialization overhead, and imbalanced load distribution, which collectively increase job completion time and reduce overall cluster throughput. Integrating task combining techniques with optimized scheduling strategies addresses these challenges by simultaneously reducing overhead and improving resource allocation. Task combining involves merging small or related jobs into larger execution units, thereby decreasing the frequency of task initialization and shuffle operations. This consolidation reduces unnecessary network communication and improves CPU and memory utilization across cluster nodes. On the other hand, enhanced scheduling algorithms allocate resources dynamically based on workload characteristics, ensuring that high-priority or large tasks receive sufficient computational resources without creating bottlenecks for smaller tasks.

Experimental analysis indicates that the combination of task aggregation and adaptive scheduling results in measurable performance improvements. Job completion times are significantly reduced, and throughput is increased due to more efficient task execution and better utilization of cluster resources. Furthermore, by pairing task combining with intelligent scheduling, the system achieves a more balanced workload distribution, preventing nodes from being underutilized or overloaded. These findings demonstrate that neither task aggregation nor advanced scheduling alone is sufficient for optimal performance in large-scale Hadoop clusters. Instead, a hybrid approach that integrates both strategies is essential for enhancing execution efficiency, minimizing overhead, and ensuring consistent resource utilization. The study highlights the potential for combining these techniques to improve the scalability and reliability of Hadoop clusters, offering practical guidance for administrators and developers seeking to optimize big data processing workflows.

CONCLUSION

Efficient job execution in Hadoop clusters is critical for handling large-scale data processing, yet default scheduling algorithms and fragmented task execution often limit cluster performance. This study demonstrates that combining task aggregation techniques with optimized scheduling strategies provides a significant improvement in execution efficiency. By merging small or related tasks, overhead associated with initialization, shuffling, and network communication is reduced, while adaptive scheduling ensures resources are allocated dynamically to balance workloads and minimize bottlenecks. The hybrid approach of integrating task combining with intelligent scheduling not only reduces job completion time but also enhances cluster throughput and resource utilization. Experimental analysis confirms that this combined methodology outperforms standalone scheduling or task aggregation strategies, particularly in heterogeneous and high-load environments. In conclusion, improving Hadoop cluster performance requires a holistic approach that addresses both task-level efficiency and cluster-level resource management. The integration of task combining and optimized scheduling techniques provides a practical framework for administrators and developers to enhance large-scale distributed data processing, ensuring faster execution, better scalability, and more reliable utilization of computational resources.

REFERENCES

1. Gautam, Jyoti V. “Empirical Study of Job Scheduling Algorithms in Hadoop MapReduce.” Cybernetics and Information Technologies, vol.
2. Guru Prasad M. S., Nagesh H. R., and Swathi Prabhu. “Performance Analysis of Schedulers to Handle Multi Jobs in Hadoop Cluster.”

3. Kotikam, Gnanendra, and Selvaraj Lokesh. "YARN Schedulers for Hadoop MapReduce Jobs: Design Goals, Issues and Taxonomy."
4. Senthilkumar, M., et al. "A Survey on Job Scheduling in Big Data." Cybernetics and Information Technologies, vol.
5. Sreedhar, C. "A Novel Multilevel Queue Based Performance Analysis of Hadoop Job Schedulers."
6. "MapReduce Scheduling Algorithms in Hadoop: A Systematic Study." Journal of Cloud Computing, vol.
7. Lee, Ming‐Chang, Jia‐Chun Lin, and Ramin Yahyapour. "Hybrid Job‐Driven Scheduling for Virtual MapReduce Clusters."
8. Pastorelli, Mario, Antonio Barbuzzi, Damiano Carra, Matteo Dell'Amico, and Pietro Michiardi.