ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Performance Analysis of Query Optimization for Hadoop Applications

Parul Chhabra[1], Surender Singh[2]

M.Tech Scholar, Dept. of CSE, OITM Juglan Hisar, India[1]

Assistant Professor, Dept. of CSE, OITM Juglan Hisar, India[2]

ABSTRACT: Big Data applications consume the resources at large scale due to huge volume of data. Researchers have developed some solutions to optimize the resource consumption but each scheme was designed to sort out a specific problem. In this paper, impact of query execution was investigated and a optimal solution for query processing was offered. Performance analysis includes various parameters i.e. execution time, battery life, number of processes used. For experimental purpose, MapR sandbox was used.

KEYWORDS: Hadoop, Big Data, Energy Consumption, HDFS, MAPReduce

I. INTRODUCTION

For experiment purpose, MapR sandbox was used which have following features:
It integrates the Hadoop framework with other open source frameworks i.e. Hive, Spark and Yarn etc.

Advantages:

- Optimized architecture for faster analysis of huge data
- Support for Multiple Clusters
- Automatic disaster recovery
- Real time data synchronize
- Distributed data extraction features
- Robust security policies
- Provision of real time data recovery from multiple Clusters
- Integration support of NoSQL
- Data & Event streaming for real-time applications[1][2]

Traditional Database

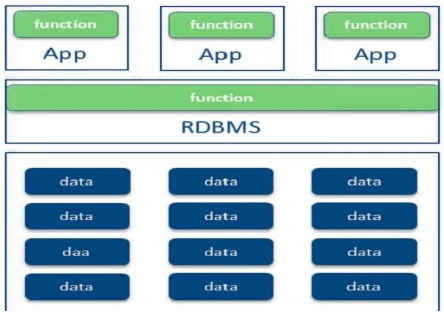


Figure: Architecture of Traditional Database[1][2]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

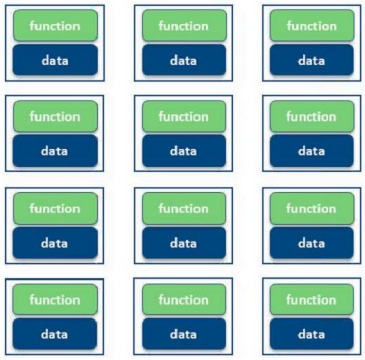


Figure: Hadoop Architecture

A. *MapR Sandbox Architecture*

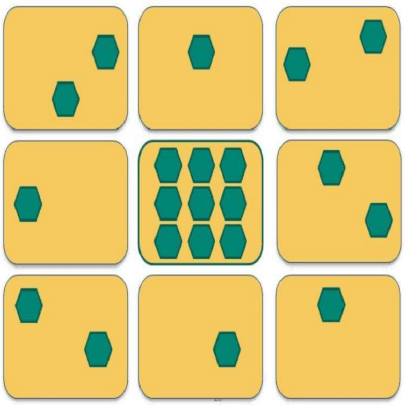


Figure: MapR Sandbox Architecture

It uses multiple cells to retain the image of a cluster and each cell contains some small amount of data. In case of server crash, it is ensured that replica must survive and immediately a new master node is sleeted

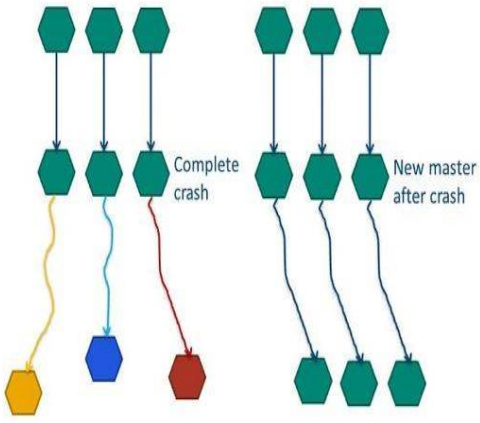


Figure: Master node selection

MapR is able to perform faster read write operations and enhances the overall efficiency by reducing the data access intervals. [1][2]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

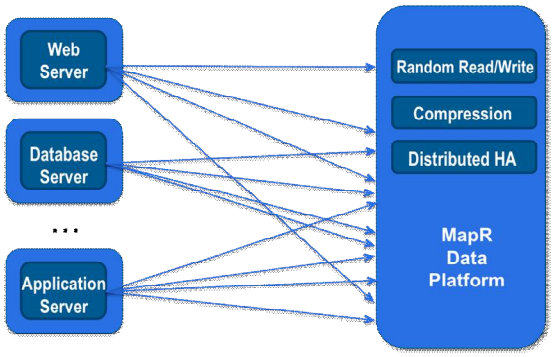


Figure: File System operation

Apache Hive

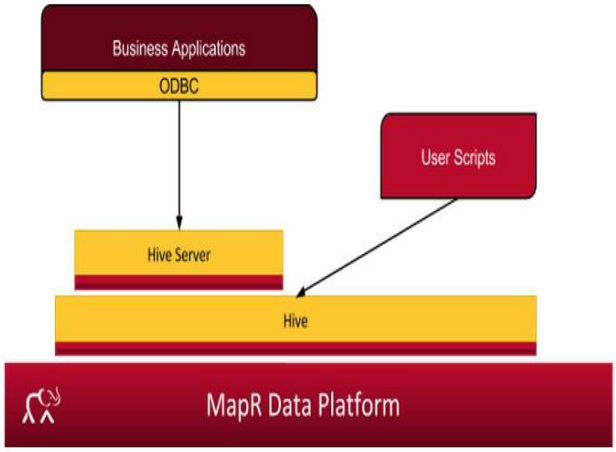


Figure: Apache Hive [3]

Apache Hive is a framework which can directly execute SQL queries using Hadoop applications. It support a query language which can be converted in to MAPreduce processes. Following are the few application domains of Hive:

- It can be used for Data Mining
- Data processing and Analysis

II. LITERATURE SURVEY

Eugen Feller et al. [12] analyzed Hadoop performance using traditional model of collocated data and compute services. Data Separation and compute services provides more stiffness in environments where data locality might not have a considerable impact such as virtualized environments and clusters with advanced networks. They also did analysis of energy efficiency of Hadoop on physical and virtual clusters using various specifications. Analysis results show that performance on physical clusters is significantly better as compared to virtual clusters. Performance is degraded due to separation of services depends on data to compute ratio. Application completion progress correlates with energy consumption which is application specific.

Z. Niu et al. [16] considered the resource aware solutions for Hadoop framework and identified that optimization of energy consumption and load balancing may result in fair utilization of resources. As per the study, it can be observed that performance of Hadoop applications suffers from the unfair utilization of resources and there is a need to have some sort of resource aware schemes. This study can be further extended for distributed Hadoop environment.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

M. Malik et al. [17] investigated the issues related to scalability support over Hadoop framework and found that inefficient energy consumption and variation in density, both are the major constraints for server performance. For analysis, different application were used i.e. Word Count, Sort, Grep and Tera Sort using Hadoop framework version 1.2.1 and experimental results show that server performance can be achieved under the above mentioned constraints by selecting the optimal configuration of parameters at architecture level.
Duy-Hung Phan et al. [18] explored the hadoop ecosystems and focused on inefficient query execution and communication of applications and developed a new ROLLUP operator for high level languages. It is self optimized and can perform automatically load balancing by estimating the suitable operating point and achieves the highest performance. For development purpose, they used Apache-PIG, high level language of Hadoop system and they used real data for analysis purpose and according to analysis results, performance of proposed scheme is upto 50% as compared to traditional Rollup operator.

Krish K. R et al. [19] proposed a hardware level scheduler for workflow optimization. It can operate in heterogeneous environment. Experimental results show that adopt can adopt the various hardware configurations and offers multiple energy consumption profiles. Energy profiler analyzes the power consumption ratio for each job at a particular cluster and maintains a optimal list of cluster.
Alok Kumbhare et al. [20] investigated some important issues i.e. load balancing, fault tolerance etc and offered a flexible streaming MapReduce model by introducing consistent hashing with the support of peer check pointing/peer backup. For load balancing and fault tolerance, it uses low latency and dynamic updates. Its performance can be measured in terms of efficient load-balancing, low-overhead fault-tolerance and parallel fault-recovery from multiple concurrent failures. Current research work can be extended to support the large scale real time applications and databases.

Demetrio Gomes Mestreyet al.[21] enhanced a load balancing method for distributed blocking-based entity matching Based on MapReduce framework. It uses entity matching process for large scale datasets and dose not depends over input partitions. It uses greedy optimization method for processing of data distributions and workloads and reduce the tasks using slicing of large blocks. They also compared the proposed scheme with BlockSplit and show its performance in terms of optimal load balancing. Current research work can be extended for data-intensive tasks and to resolve horizontal skew problem.

Xiaofei Hou et al. [22] developed a method which can analyze log files for load balance of various racks on a Hadoop cluster. Each rack is busy in execution but when any operation utilizes heterogeneous racks at same time, may results in degradation of performance which can be solved by shifting the current task to the idle rack/less busy rack. Decision to shift a task to a particular rack is taken after analyzing the current log file, in order to identify the idle/less busy racks and finally, task is moved to most appropriate rack for processing. Simulation results show its performance in terms of optimization of execution and completion time of the shifted tasks.

D. Cheng et al. [23] explored the issues related to the job scheduling and its impact over the availability of resources and their consumption. Proposed scheme estimates the execution time and predict the resource consumption. Resource predictor estimates the number of available resources by interacting with job tracker at regular intervals. Experimental results show its accuracy in terms of fair task scheduling w.r.t. available resources.

D. Cheng et al. [24] identified the requirements of resource optimization and proposed an ant algorithm based resource aware solution for heterogeneous Hadoop clusters. Application execution in heterogeneous environment with variations in workload may cause the unfair utilization of energy. Proposed scheme uses a task assignment method without depending upon the characteristics of current workload. Experiments show that it can conserve the available resources as compared to Tarazu job scheduler.

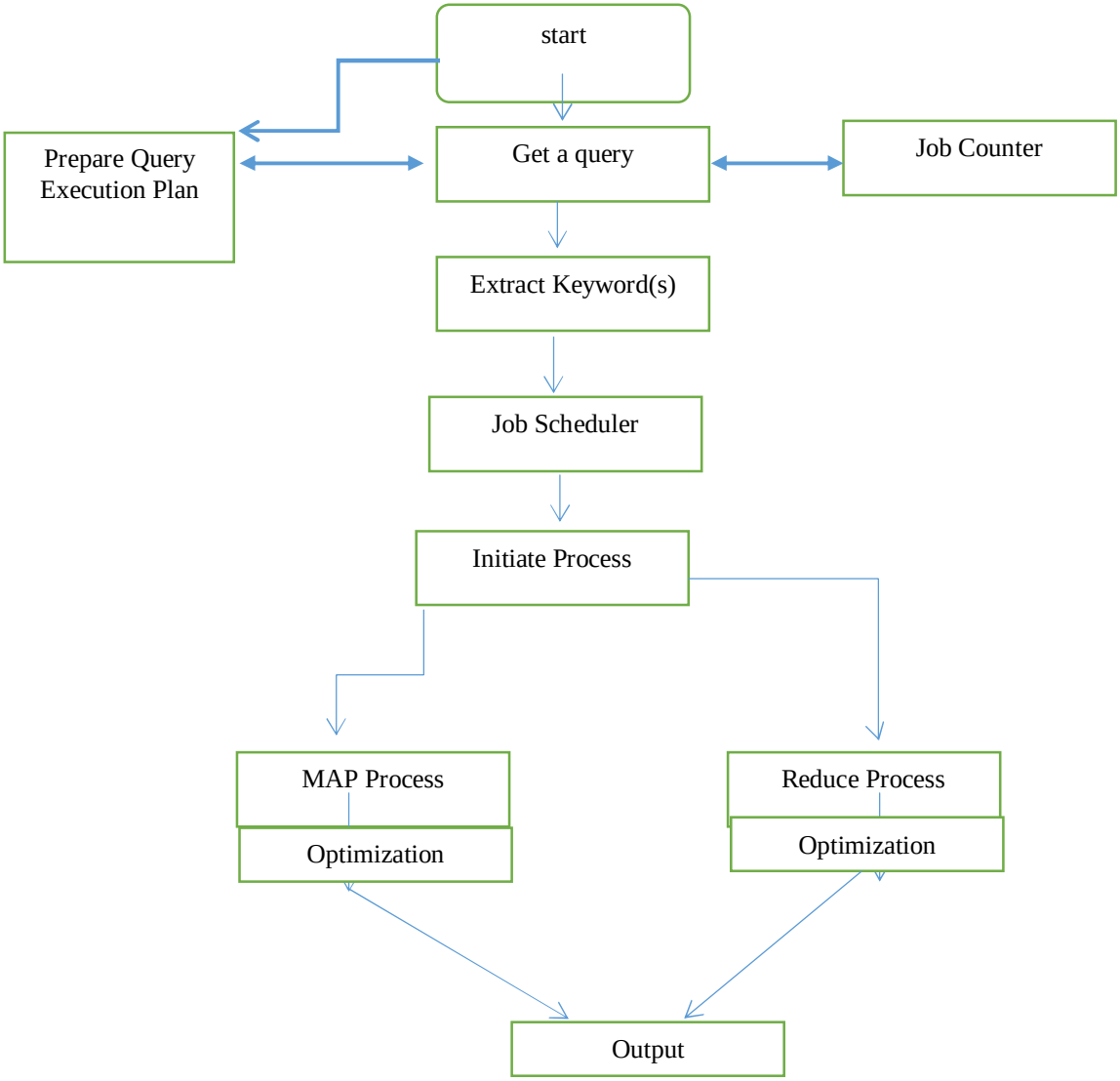
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

III. PROPOSED SCHEME



As per proposed scheme, a simple query is executed as per the execution plan defined by database but it may consumes more resources so there is need to optimize the query execution process, in order to save the available resources. Query execution can be modified in such a way that it will consume less resource by dividing the query execution plan to the two different processes in parallel execution environment. Job scheduler keeps the track of all jobs and process being executed and assigns a unique counter for each process. Single execution process is subdivided into MAP and Reduce Processes and their final output generates the actual query's output that is similar to the query which was not optimized, in less time.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

IV. PERFORMANCE ANALYSIS

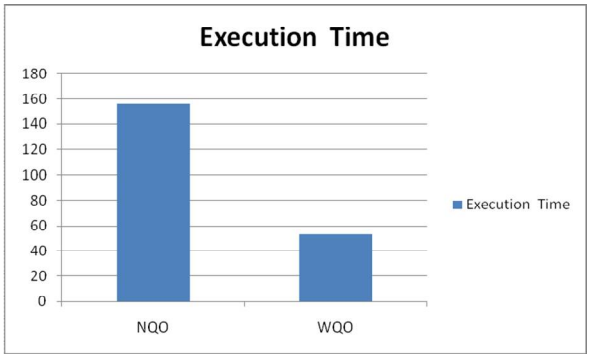


Figure:4.1 Execution Time

Figure:4.1 above shows the energy consmued by NQO and WQO and ti can be observed that WQM consumed less amount of energy as compared to NQO.

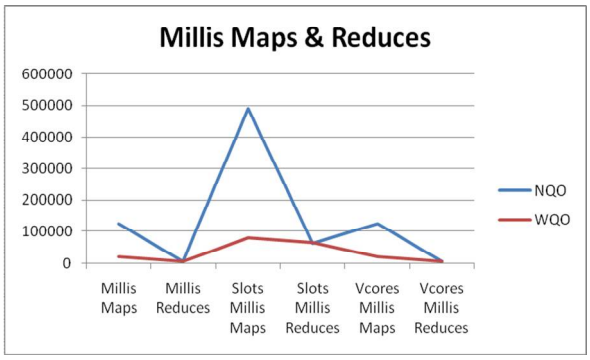


Figure:4.2 Millis Maps & Reduces

Figure:4.2 shows the number of Bytes written during file access operation. It can be observed that more Bytes are written during query optimization process as compared to the NQO.

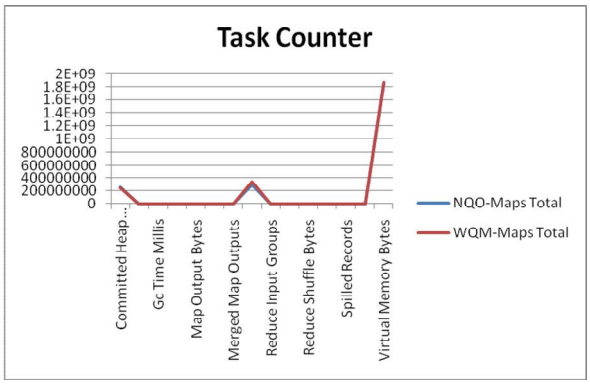


Figure: 4.3 Task Counter

Figure:4.3 shows the number of Bytes read during file access operation. It can be observed that NQO and WQM, both have attempts for read operation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

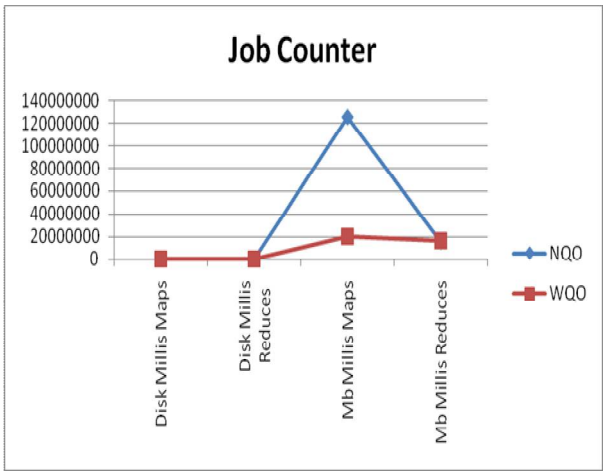


Figure: 4.4 Job Counter

Figure: 4.4 shows the Job counter for NQO and WQM. It can be observed that NQO used more Mb Millis Map and Reduce process as compared to WQO.

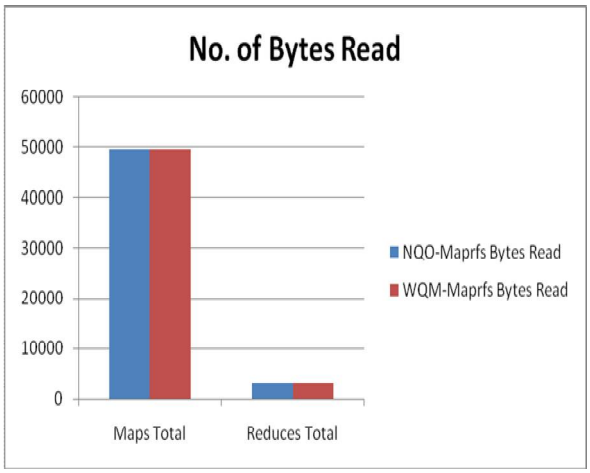


Figure:4.5 Number of Bytes Read

Figure: 4.5 shows that NQO and WQO, both have almost performed similar tasks but results show that WQO performed well as compared to NQO.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

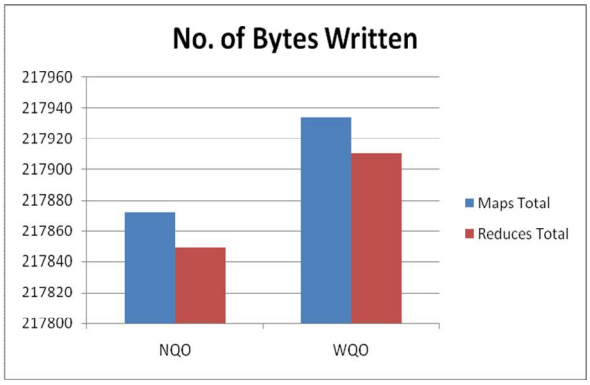
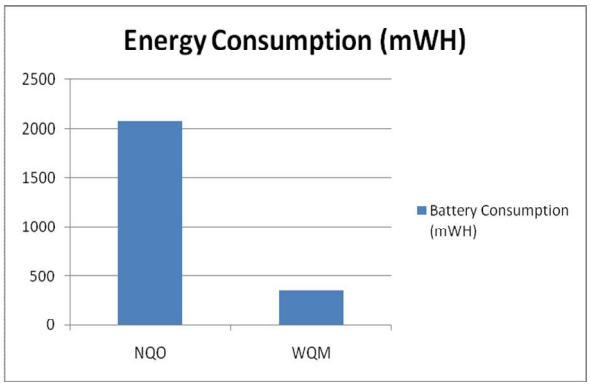


Figure:4.6 Number of Bytes Written

Figure: 4.6 shows that WQO requires less slots for MAP and Reduce processes. Results show that NQO consumed extra slots for Millis Map and Reduce processes.



FIgue:4.7 Energy consumption (mWH)

Figure: 4.7 shows the query execution time . NQO was completed 156s for completion as compared to WQO which is 53s only.

V. CONCLUSION

In this paper, we investigated the performance issues related to the resource optimization. For experiments purpose, we use MapR sandbox. Experimental results show that NQO and WQM, both have attempts for read operation and WQM wrote more Bytes are written during query optimization process as compared to the NQO.

Analysis show that NQO used more Mb Millis Map and Reduce process as compared to WQO and both have almost performed similar tasks but WQO performed well as compared to NQO.

WQO requires less slots for MAP and Reduce processes. NQO consumed extra slots for Millis Map and Reduce processes.

NQO was completed 156s for completion as compared to WQO which is 53s only. So it can be observed that NQO requires more time for job completion and consumed more energy as compared to WQO.

Finally, it can be concluded that proposed scheme is able to reduce the over all consumption of available resources and it can be extended to load balancing also.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

REFERENCES

[1]https://www.mapr.com/why-hadoop/why-mapr
[2] https://www.mapr.com/sites/default/files/mapr-datasheet-apache-hadoop-converged-enterprise-edition-020116.pdf
[3] https://www.mapr.com/products/product-overview/apache-hive
[4] Eugen Feller, Lavanya Ramakrishnan, Christine Morin, "On the Performance and Energy Efficiency of Hadoop Deployment Models", IEEE-2013, pp.131-136
[5] Z. Niu, Bingsheng He, "A Study of Big Data Computing Platforms: Fairness and Energy consumption", International Conference on Cloud Engineering Workshops, IEEE-2016, pp.207-209
[6] M. Malik, AvestaSasan, Rajiv Joshi, SetarehRafatirahHoumanHomayoun,"Characterizing Hadoop Applications on Microservers for Performance and Energy Efficiency Optimizations", IEEE-2016, pp.153-154
[7] Duy-Hung Phan, Quang-Nhat Hoang-Xuan, Matteo Dell'Amico, Pietro Michiardi, "Efficient and Self-Balanced ROLLUP Aggregates for Large-Scale Data Summarization", IEEE International Congress on Big Data, IEEE-2015, pp.158-165
[8] Krish K. R., M. Safdar Iqbal M. Mustafa Rafique, Ali R. Butt, "Towards Energy Awareness in Hadoop", Fourth International Workshop on Network-Aware Data Management, IEEE-2014, pp.16-22
[9] AlokKumbhare, Marc Frincu, Yogesh Simmhan, Viktor K. Prasanna, "Fault-Tolerant and Elastic Streaming MapReduce with Decentralized Coordination", International Conference on Distributed Computing Systems, IEEE-2015, pp.328-338
[10] Demetrio Gomes Mestrey, Carlos Eduardo Santos Pires, "Improving Load Balancing for MapReduce-based Entity Matching", ISCC, IEEE-2013, pp. 000618 - 000624
[11] XiaofeiHou, Ashwin Kumar T K, Vijay Varadharajan, "Dynamic Workload Balancing for Hadoop MapReduce", International Conference on Big Data and Cloud Computing, IEEE-2014, pp.56-62
[12] Dazhao Cheng, Jia Rao, Changjun Jiang, Xiaobo Zhou, "Resource and Deadline-aware Job Scheduling inDynamic Hadoop Clusters", 29th International Parallel and Distributed Processing Symposium, IEEE-2015, pp.956-959
[13] Dazhao Cheng, Palden Lama, Changjun Jiang, Xiaobo Zhou, "Towards Energy Efficiency in Heterogeneous Hadoop Clusters by Adaptive Task Assignment", 35th International Conference on Distributed Computing Systems, IEEE-2015, pp.359-368