To Enhance Job Execution and Resource Management in Map Reduce Using Heterogeneous Phase Level Scheduling

R. Rubana, P.G Scholar
Department of Computer Science and Engineering
A.V.C College of Engineering
Mannampandal-609 305
rubu.93@gmail.com

M. S.VinayagaMoorthy, Associate Professor
Department of Computer Science and Engineering
A.V.C College of Engineering
Mannampandal-609 305
ssvmoorthy@gmail.com

***Abstract* - In big data Map Reduce is the technique that helps to process query from user to server in an efficient way. The Map Reduce is used to process large amount of servers in a parallel way. So the parallel processing is done in map reduce to retrieve results. In achieving parallel processing the Jobs are split into phase level into 3 phases. Each phase is to provide with resources for parallel and fast execution of jobs. The resources are provided in a homogeneous way. In this Paper present a Heterogeneous phase-level scheduling algorithm with Jobs Execution Scheduling that split resources into heterogeneous way this helps to achieve jobs to be execute greater with effective use of resources which improves speed and showing the resource usage variability within the lifetime of a task using a wide-range of Map Reduce jobs. This Scheduler improves execution parallelism and resource utilization without introducing stragglers. Energy-Efficient Algorithm provide the flow time of a job is the length of the time interval between the release time and the completion time of the job with work efficiency.**

I. INTRODUCTION

In MapReduce, a job is a collection of Map and Reduce tasks that can be scheduled concurrently on multiple machines, resulting in significant reduction in job running time. Many large companies, such as Google, Facebook, and Yahoo!, routinely use MapReduce to process large volumes of data problem becomes significantly easier to solve if we can assume that all map tasks (and similarly, all reduce tasks) have on a daily basis. A central component to a MapReduce system is its job scheduler. Its role is to create a schedule of Map and Reduce tasks, spanning one or more jobs, that minimizes job completion time and maximizes resource utilization. The job scheduling homogenous resource requirements in terms of CPU, memory, disk and network bandwidth. Indeed, current MapReduce systems, such as Hadoop Map- Reduce Version 1:x, make this assumption to simplify the scheduling problem. These systems use a simple slot-based resource allocation scheme, where physical resources on each machine are captured by the number of identical slots that can be assigned to tasks. several recent proposals, such as resource-aware adaptive scheduling (RAS) [15] and Hadoop MapReduce Version 2 (also known as HadoopNextGen and Hadoop Yarn) [7], have introduced resource- aware job schedulers to the MapReduce framework. How- ever, these schedulers specify a fixed size for each task in terms of required resources (e.g. CPU and memory), thus assuming the run-time resource consumption of the task is stable over its life time.In this paper, we present PRISM, a **P**hase and **R**esource **I**nformation-aware **S**cheduler for **M**apReduce clusters that performs resource-aware scheduling at the level of task phases. Specifically , we show that for most MapReduce applications, the run-time task resource consumption can vary significantly from phase to phase. Therefore, by considering the resource demand at the phase level, it is possible for the scheduler to achieve higher degrees of parallelism while avoiding resource contention. To this end, we have developed a phase-level scheduling algorithm with the aim of achieving high job performance and resource utilization. Through experiments using a real MapReduce cluster running a wide-range of workloads, we show PRISM delivers up to 18 percent improvement in resource utilization while allowing jobs to complete up to 1:3 faster than current Hadoop schedulers.

II. RELATED WORK

This section provides an overview of Hadoop MapReduce and various phases in a MapReduce job.

Hadoop MapReduce

Map Reduce is a parallel computing model for large- scale data-intensive computations. A MapReduce job consists of two types of tasks, namely map and reduce tasks. A map task takes as input a key-value block stored in the underlying distributed file system and runs a user specified map function to generate intermediary key-value output. Subsequently, a reduce task is responsible for collecting and applying a user specified reduce function on the collected key-value pairs to produce the final output. A Hadoop cluster consists of a large number of commodity machines with one node serving as the master and the others acting as slaves. The master node runs a resource manager (also known as a job tracker) that is responsible for scheduling tasks on slave nodes. Each slave node runs a local node manager (also known as a task tracker)

that is responsible for launching and allocating resources for each task.

MapReduce Job Phases

A map task can be divided into two main phases: Map and Merge. The input of a Map- Reduce job is stored as data blocks where data blocks are stored across multiple slave nodes. In the map phase, a mapper fetches a input data block from the Hadoop Distributed File System [4] and applies the user-defined map function on each record. The map function generates records that are serialized and collected into a buffer.

When the buffer becomes full (i.e., content size exceeds a pre-specified threshold), the content of the buffer will be written to the local disk. Lastly, the mapper executes a merge phase to group the output records based on the intermediary keys, and store the records in multiple files so that each file can be fetched a corresponding reducer. Similarly, the execution of a reduce task can be divided into three phases: shuffle, sort, and reduce. In the shuffle phase, the reducer fetches the output file from the localstorage of each map task and then places it in a storage buffer that can be either in memory or on disk depending on the size of the content.

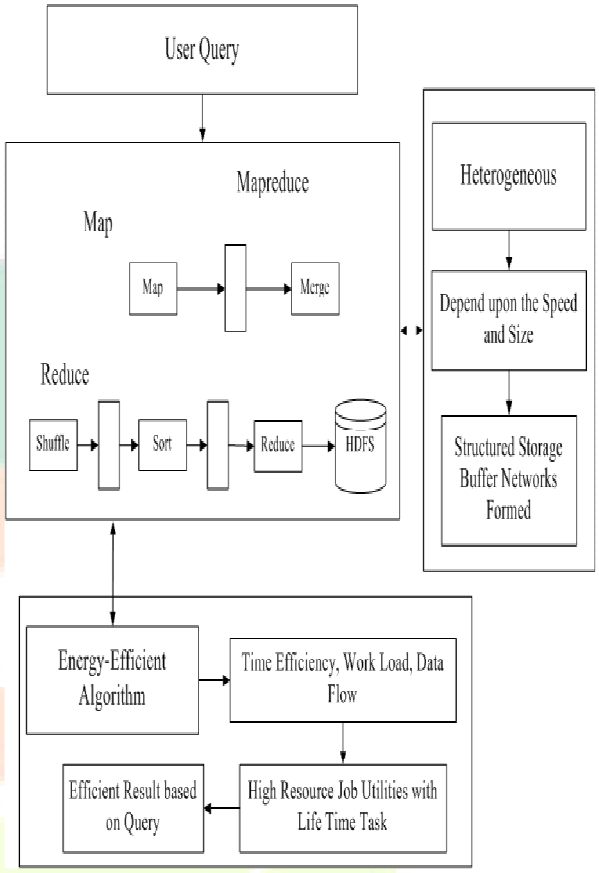
III. EXISTING SYSTEM

In an existing Map Reduce schedulers define a static number of slots to represent the capacity of a cluster, creating a fixed number of execution slots per machine. This abstraction works for homogeneous workloads, task-level schedulers to effectively utilize available resource requirements of individual jobs in multi-user environments. Homogeneous short interactive queries are submitted to the same Map Reduce cluster based on requirements in terms of CPU, memory, disk and network bandwidth. Map Reduce cluster scheduler is critical to providing the desired quality of service like time efficient in high resource schedules.

Disadvantages

- Ineffective for tasks with unpredictable behavior.
- Decrease of performance in very large clusters.
- Serial processing, so Time efficient for extracting in a high resources.

IV. SYSTEM ARCHITECTURE



V. PROPOSED SYSTEM

Map Reduce has emerged as a popular paradigm for processing large information in parallel over a cluster. Hadoop has been successfully used in a variety of applications, such as HDFS, search indexing, recommendation systems, etc. The propose concept of metric in a heterogeneous cluster to realize a scheduling scheme, the number of slots might not be an appropriate metric of the share because all the slots are not the same. But the phase level scheduling showing the resource usage variability within the lifetime of a task. And the concept heterogeneous in Map Reduce that achieves high performance and fairness. Energy-Efficient Algorithm that can maintain the load balancing and provides better improved strategies through efficient job scheduling and modified resource allocation techniques. Our research efforts on extracting in a concept of file path dependency for automatic text extraction techniques to resource allocation.

Advantages

- Killing and duplicating tasks results in wasted resources.
- Prevent stragglers resulting from resource contention.
- Completion time of their corresponding jobs is mitigated by allocating time.
- Improve scheduling delay, scheduling skew, system utilization, and parallelism.

- Heterogeneous reduce network traffic and increase of performance.

User Query Processing

- User search the information from large number of data, and query in extracting using map reduce.
- Suppose the user search something file on a complex in data intensive computing.
- It retrieve the result from collection of related data's from server.

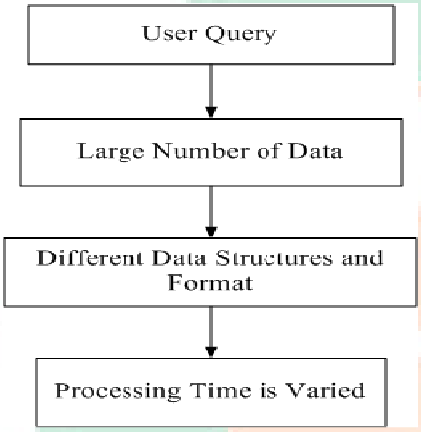
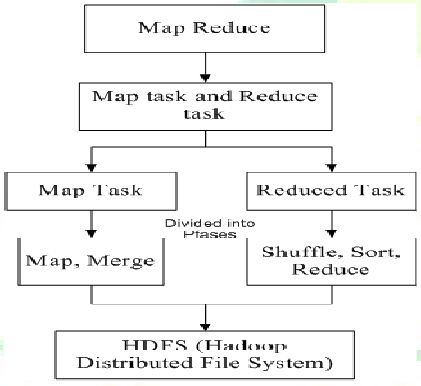


Figure 5.1(a)User Query Processing

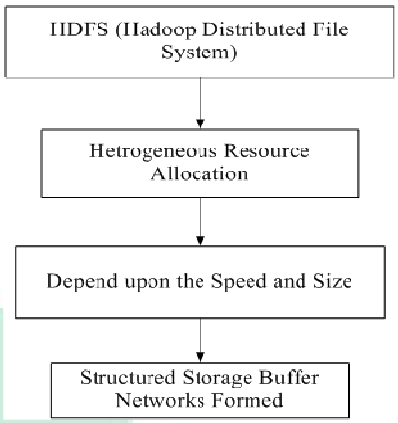
Execution of Parallel jobs in Map reduce Indexing

- The map reduce perform parallel processing that is collect user searchable query information from a large complex data base in simultaneously.
- The map reduce using shortest path of information retrieval from a collection of large file information.



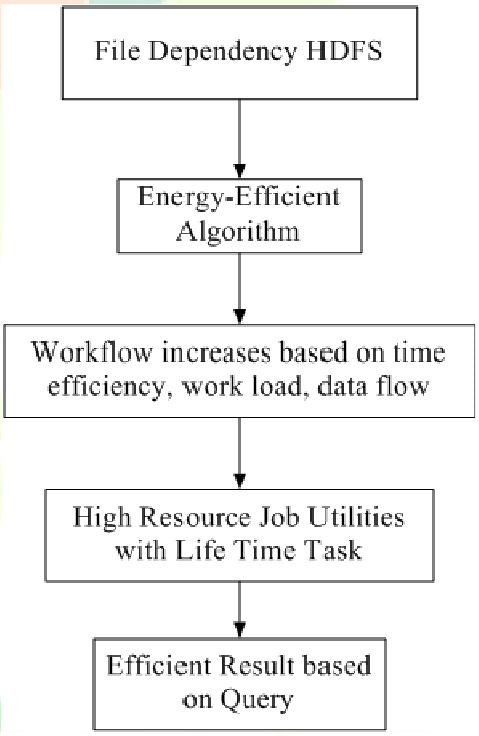
Heterogeneous Efficient Resource Allocation

- Heterogeneous process is used for parallel processing and resource usage variability within the lifetime of a task using a wide-range of Map Reduce jobs.



Energy-Efficient Algorithm Improves Workflow Efficiency

- Energy-Efficient Algorithm predict the extracting result based on file path dependency information System.
- It improves execution parallelism and resource utilization without introducing stragglers and Flow time minimizations.



screenshots

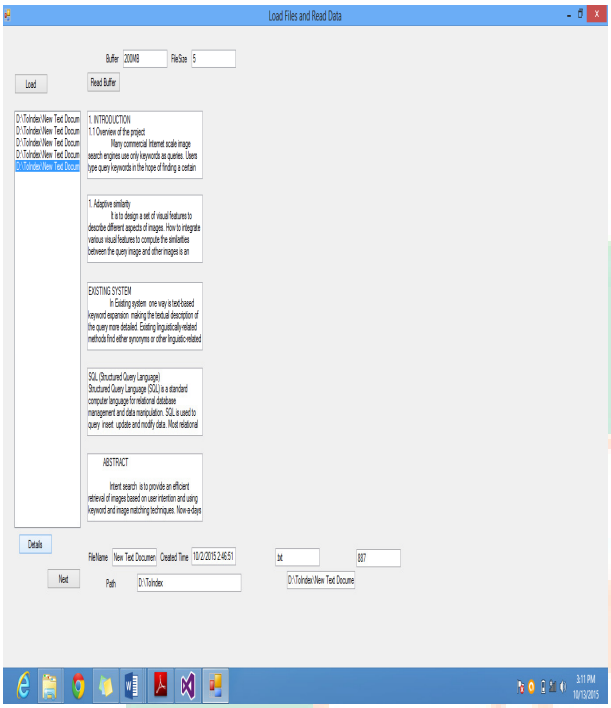


Figure 6.1 User Query Processing

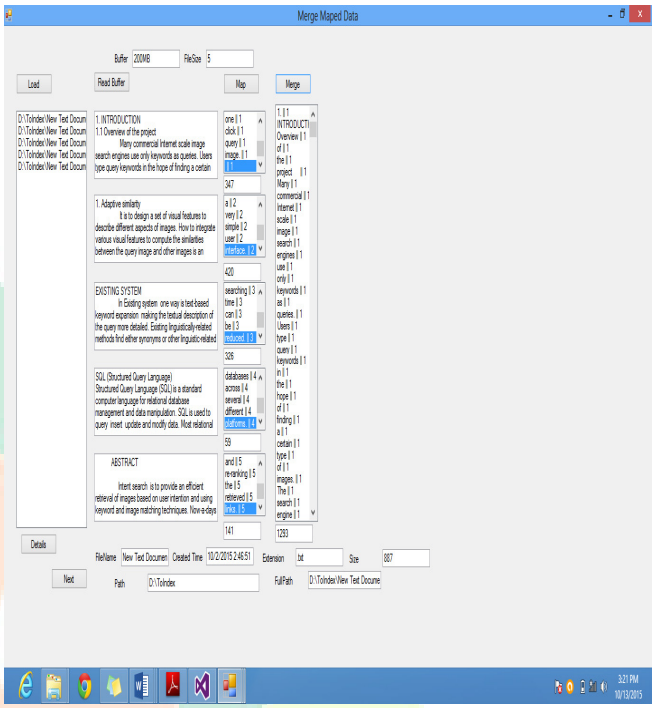


Figure 6.3Merging phase

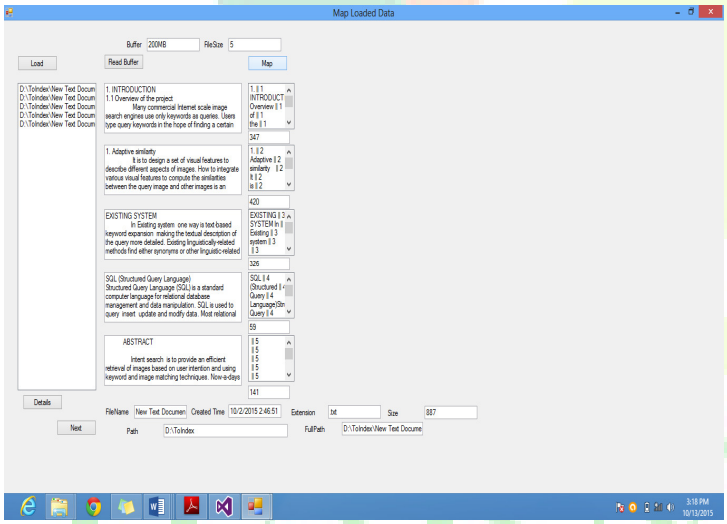


Figure 6.2 Mapping phase

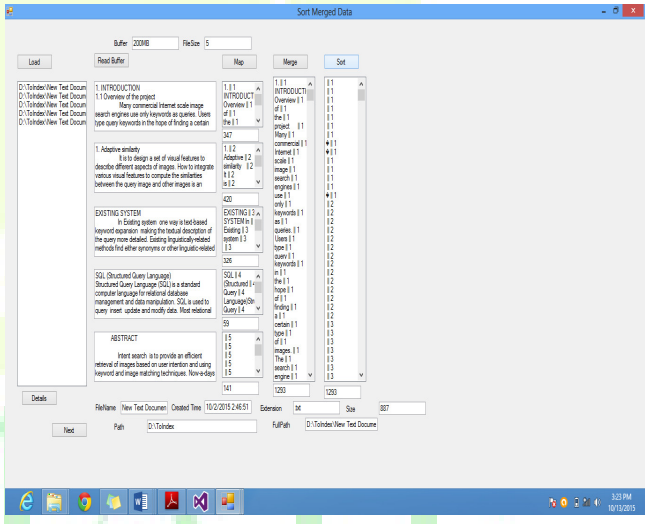


Figure 6.4 Sorting phase

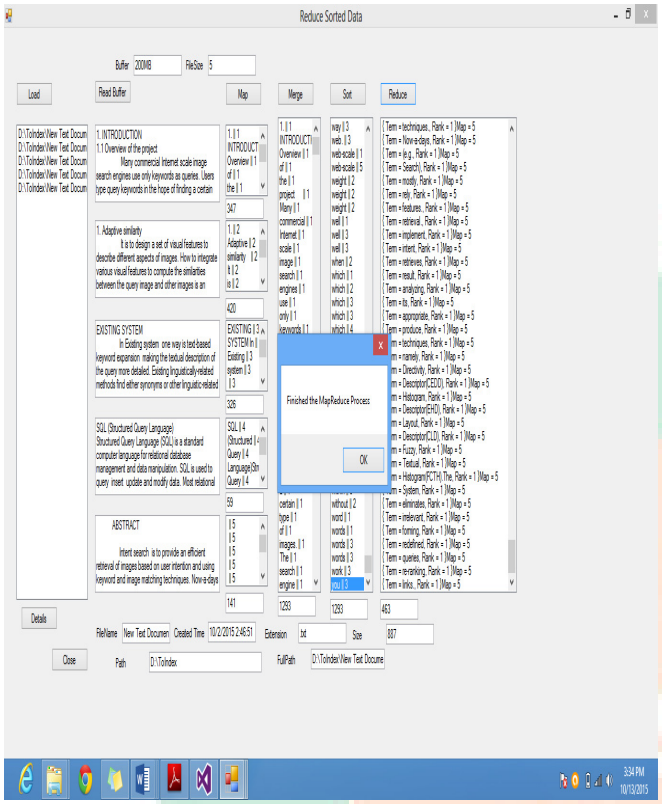


Figure 6.5 Reduce phase

VI. CONCLUSION

Map-reducing is the popular for the task scheduling and resource allocation based on the intensive computing programming. Resource based map task processing time estimation is satisfactory Resource scheduler did not manage to outperform scheduler in resource-homogenous environment and most cases of resource heterogeneous environment due to extra concurrent reduce task. However we verified that resource scheduler is indeed resource aware. Heterogeneity-Aware Dynamic Capacity Provisioning in data centres. Fine-grained Resource-Aware Map-Reduce Scheduling. It performs the prediction based on the energy efficient algorithm, when moved from a resource-homogeneous environment to a resource- heterogeneous environment.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Opear. Syst. Des. Implementation, 2004, p. 10.

[2] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for, in-memory cluster computing," in Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation, 2012, p. 2.

[3] S. Ewen, K. Tzoumas, M. Kaufmann, and V. Markl, "Spinning fast iterative data flows," in Proc. VLDB Endowment, 2012, vol. 5, no. 11, pp. 1268–1279.

[4] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "Haloop: Efficient iterative data processing on large clusters," in Proc. VLDB Endowment, 2010, vol. 3, no. 1–2, pp. 285–296.

[5] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: A runtime for iterative mapreduce," in Proc. 19th ACM Symp. High Performance Distributed Comput., 2010, pp. 810–818.

[6] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "imapreduce: A distributed computing framework for iterative computation,"J.GridComput.,vol.10,no.1,pp.47-68,2012.

[7] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce forincremental computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 7:1–7:14.