International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume - 2, Issue - II ISSN (Online): 2454- 1966

Research Manuscript Title

RESOURCE ALLOCATION IN PHASE-LEVEL USING MAPREDUCE IN HADOOP FRAMEWORK

S.Pavithra[1], Dr.G.Malathy[2]

PGScholar, Associate Professor

Dept of CSE, KSR Institute for Engineering and Technology, Tiruchengode, Tamilnadu, India

E-Mail: pavithrasenthil123@gmail.com, malathi.gurunathan@gmail.com

JUNE – 2016

www.istpublications.com

RESOURCE ALLOCATION IN PHASE-LEVEL USING MAPREDUCE IN HADOOP FRAMEWORK

S.Pavithra[1], Dr.G.Malathy[2]
PGScholar, Associate Professor
Dept of CSE, KSR Institute for Engineering and Technology,Tiruchengode,Tamilnadu,India
E-Mail: pavithrasenthil123@gmail.com, *malathi.gurunathan@gmail.com*

ABSTRACT

Map Reduce has a parallel programming model for large data-intensive computation.Map Reduce can significantly reduce the running time of data-intensive jobs by breaking down each job into small map and reduce phases and executing them in parallel across a large number of machines. The Phase-level Resource Information-aware Scheduling for Mapreduce (PRISM) used for fine-grained resource-aware scheduler that divides tasks into map and reduce phases, where a constant resource usage profile maintain for each phase and performs scheduling at the phase level.To overcome this limitation, it proposes an optimal reduce scheduling policy called Self Adaptive Reduce Scheduling(SARS)is proposed for reduce tasksin the Hadoop platform. It can decide the start time point of each reduce task dynamically according to each job context, including the task completion time and the size of map output by estimating job completion time, reduce completion time and system average response time.When comparing with other algorithms, the reduce completion time is decreased.It is also proved that the average response time is decreased by 11% to 29%.

Key Terms: Hadoop, HDFS, MapReduce, Scheduling

I INTRODUCTION

The data-driven from large volume of data in internet services such as facebook,Yahoo,Rackspace. To compute Cluster, some of systems like MapReduce use for optimizing the batch jobs. Its uses Map Reduce for processing a large data of size terabytes and petabytes of data. The large amount of data as increasing by leaps and bounds from the different sources such as individuals of organization, social media.

The data may be structured, semi-structured or unstructured. The primary advantage of all the organizations gaining knowledge from the large data set. While it creates problem on resources demands leads to poor performance. Scheduling tasks provides less running task on a single machine and cause poor resource utilization [1]. In a MapReduce, job scheduling problem is easy to solve when a map and reduce task have homogenous resources. Sometimes the job has heterogeneous resource requirements then it differs from one task-to-task and becomes lower performance.

There are many phases in single task with different resource constraints and different procedures thus job scheduling based on resource conflict or low utilization. PRISM algorithm performs resource allocation at the each phases in task.The Apache Hadoop is a framework that allows for the storing the huge amount of data sets in which distributed across clusters of computers using simple programming models.

The architecture of hadoop is small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a Job tracker, Task tracker, Name node and Data node.

The main idea of the MapReduce algorithm is denotes every Map and Reduce task independent of all other ongoing Maps and Reduces task, then the operation can be run on different lists of data and keys in parallel.

In a large cluster of machines, one step further and run their Map operations on servers where the data lives. Instead of copy the data over the network to the program, it gives the program to the machines. The output list can be saved to the HDFS and the reducers process for merge the results. Furthermore, it may be possible to run data in parallel for each reducing different keys.

HDFS spreads multiple copies of the data across different machines. Its offers reliability without the need for RAID-controlled disks and also offers multiple locations for running the map phase. If a machine with one copy of the data is busy or offline, then another machine used.

A job scheduler keeps track of MR jobs in which schedules individual Maps and Reduces or any intermediate merging operations for specific machines. It monitors the success and failures of individual *Tasks*and to complete the entire whole batch job.

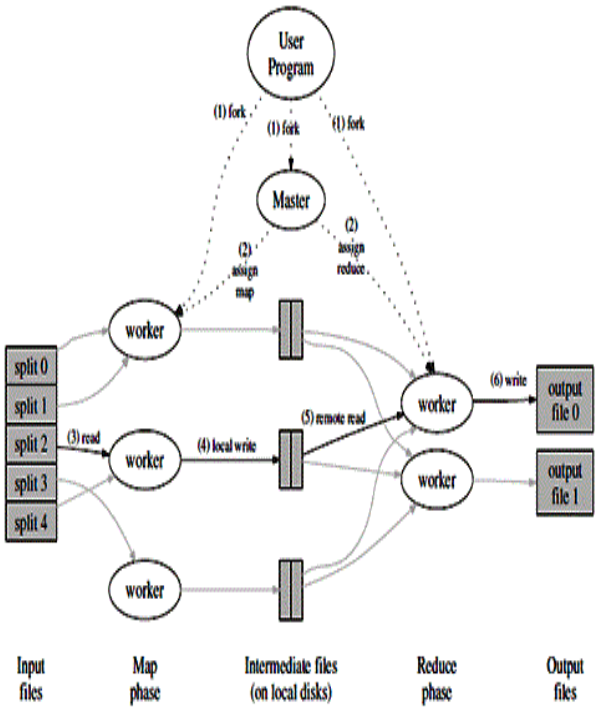

Fig.1: Map Reduce Overview

The HDFS and Job scheduler can accessed by the people and programs that requires to read and write data and to submit and monitor the MR jobs. It provides own distributed filesystem and jobs on servers near the data stored on the filesystem or any other supported filesystem while it use more than one.

The HDFS was designed to be a scalable, fault-tolerant,distributed storage system that works closely with MapReduce. HDFS will "just work" under a variety of physical and systemic circumstances. By distributing storage and computation across many servers, the combined storage resource can grow with demand while
remaining economical at every size.

IIRESEARCH OBJECTIVE

The FIFO is default scheduler of Hadoop and act as queue. Its works in the heterogeneity of Hadoop cluster components irrespectively. the Fair Scheduler to manage large amount of data with their Hadoop cluster. Fair scheduling aims to give a fair share of the resources to every user. In this the jobs are divided into pools and each pool is allocated a fair share of the resources i.e., the Map and Reduce slots. In the Capacity Scheduler the user jobs are submitted into queues and each queue is allocated a fraction of the total resource capacity. All the jobs assigned in a queue can use the entire capacity of resources given to the queue. The free capacity of a queue is shared by the other queues.

The Deadline Constraint Scheduler focuses on the deadlines for scheduling the jobs. It has two main proposed components:

- Job execution cost model- It deals with the distribution of data and handles different parameters like runtime of map and reduce tasks, input data sizes, etc.
- Constraint based scheduler- It takes the user deadlines as input and performs the scheduling.

The system works on the assumption that the reduce tasks begin only after the execution of all map tasks is finished and same amount of input data is given to each reduce node.In Hadoop the Job Tracker is responsible for taking all the scheduling decisions. It splits the input data and assigns the work to the Task Trackers.

The Task Trackers then execute the tasks. The Job tracker takes care of the running jobs and maintains a record of the tasks assigned to the Task Tracker and the state of each Task Tracker. Each Task Tracker has a fixed number of computation slots allocated to it. The idea of the Resource Aware Scheduling is that the Task Trackers keep a track of the utilization of the resources (CPU utilization, channel, I/O, etc).

There have been a number of proposals for task scheduling indistributed systems, which use various mathematical techniquesto achieve the MapReduce scheduling process. At present, theresearches on MapReduce scheduling algorithms focus on theoptimization of the job computation time, cluster workloads anddata communication.

Balanced-pools efficiently utilize performance properties ofMapReduce jobs in a given workload for constructing an optimizedjob schedule [10]. For the default method of Hadoop, whichcannot schedule the tasks to the nodes with the prefetched data,a prefetching technique was proposed in [11] to hide the remotedata access delay caused by the map tasks processed on thenodes without the input data. MTSD propose an extensional MapReduce task scheduling algorithm for deadline constraints in the Hadoop platform, which allows a user to specify a job'sdeadline and tries to make the job to be finished before the deadline.

Some of these proposals [9] presented the workload characteristicoriented scheduler, which strives for co-locating tasks ofpossibly different MapReduce jobs with complementing resourceusage characteristics. It presented a schedulingtechnique for multi-job MapReduce workloads that is able to dynamicallybuild performance models of the executing workloads and then use these models for scheduling purposes.

As the MapReducedistributed computations were analyzed as a divisible loadscheduling problem [1], several classes of algorithms were proposedand examined for scheduling divisible loads on a heterogeneoussystem with memory limits [2]. Some task schedulingalgorithms are to release the data communication among remoteslots, e.g., the center-of-gravity reduce scheduler is a localityaware
and skew-aware reduce task scheduler for saving MapReducenetwork traffic [23], and MaRCO employs eager reduce toprocess partial data from some map tasks while overlapping withother map tasks' communication [6].

A MapReduce-based frameworkfor HPC analytics was developed in [5] to eliminate themultiple scans and also reduce the number of data preprocessingMapReduce programs. Considering the dynamic resource allocationfor the IaaS cloud systems, the algorithms in [6] can adjustthe resource allocation dynamically based on the updated informationof the actual task executions. In these data processes, the utilitybecomes a considerable problem naturally. The authors of [10]discussed how to increase the utilization of MapReduce clusters tominimize their cost. They optimized the execution of MapReducejobs on the cluster through the design of a job schedule that minimizesthe completion time (makespan) of such a set of MapReducejobs. To achieve the optimal user utility, the goal of resource
provisioning in [7] is to minimize the cost of virtual machines forexecuting MapReduce applications. For the practical applications,some aspects of reality should be taken into account, such as faulttolerance [28] and energy efficiency [29,30].

Themain focus of most current works about MapReduce schedulingalgorithms appears to be job scheduling, less involving task delay,especially the consideration of the tasks with the same key for theinput data block. Through analysis of the intermediate data processin Hadoop, this paper indicates that the scheduling of reduce tasksis one of the key problems which affect the performance of asystem.

III EXISTING SYSTEM

In a phase-level, its perform a task or process with heterogeneous resource requirements. The phase-level scheduling algorithm which improves execution parallelism and performance of task. The phase-level which has these parameters with good working characters. PRISMat the phase-level. While scheduling the job, PRISM offers higher degree of parallelism than current hadoop cluster. [6]It refers at the phase-level to improve resource utilization and performance.

The PRISM allocates a fine-grained resources at the phase-level to perform jobscheduling. PRISM mainly consists of 3 components: first one is the phase based scheduler at master node, localnode manager at phase transaction with scheduler and job progress monitor to capture phase -level information. To achieve these three phases, it will perform a phase-level scheduling mechanism. When the task needs to scheduled from node manager, scheduler replies with task scheduling request. Then node manager launches atask.

After completion for execution of phase, then again next task will launches. While proceeding these phases, it will pause for some time to remove the resource conflict. While proceeding in a phase level, phase-based scheduler send message to node manager. Uponreceiving heartbeat message from node manager reporting resource availability on node, the scheduler mustselect which phase should be scheduled on node.

3.1Phase-level algorithm:

Upon receiving a status message from a machine 'n'.obtain its resource utilization

Step 1: Compute the set of candidate phases and select phases in an iterative manner.

Step 2: During each iteration, for each phase calculate the utility function U (i,n),

U(i,n) = Ufairness(I,n)+A.Upref (i.n)

Step 3: Select the phase with highest utility for scheduling.

Step 4: Repeat by recomputing the utility of all the phases in the candidate phase set and select the next phase to be scheduled.

Step 5: Ends when the candidate set is empty.

Upon receiving heart beat messages from a note manager reporting resource availability on the node, the scheduler must select which phase should be scheduled on the node.

The existing implementations may result in a block of reduce tasks. When the output of map tasks become large, the performance of a MapReduce scheduling algorithm will be influenced seriously.

IV PROPOSED SYSTEM

The proposes an optimal reduce scheduling policy called SARS (Self Adaptive Reduce Scheduling) for reduce tasks start times in the Hadoop platform. It can decide the start time point of each reduce task dynamically according to each job context, including the task completion time and the size of map output.

These method is to optimize the MapReduce tasks is to select an adaptive time to schedule the reduce tasks. By this means, we can avoid the reduce tasks' waitingaround and enhance the resource utilization rate. This section proposes a self-adaptive reduce task scheduling algorithm, which gives a method to estimate the start time of a task, instead of the traditional mechanism where the reduce tasks are started once any map task is completed.

The reduce process can be divided into the following several phases. Firstly, the reduce task requests to read each map output.Next, in the sort process, these intermediate data are output to an ordered data set by merging,which are divided into two types. One type is the data in memory.When the data are read from various maps at the same time, thedata should be merged as the same keys. The other is as like thecircle buffer. Because the memory belonging to the reduce task is limited, the data in the buffer should be written to disks regularly in advance.

The subsequent data need to be merged by the data which are written into the disk earlier, the so called external sorting. The external sorting need to be executed several timesif the number of map tasks are large in the practical works. The copy and sort are customarily called the shuffle phase. Finally, after finishing the copy and sort process, the subsequent functions start, and the reduce tasks can be scheduled to the compute nodes.

4.1Implementation

The reduce task start time is determined by this advanced algorithm SARS(Self-Adaptive Reduce Scheduling). In this method, the start times of the reduce tasks are delayed for a certain duration to lessen the utilization of system resources.

The SARS algorithm schedules the reduce tasks at a special moment, when some map tasks are finished but not all. By this means, how to select an optimal time point to start the reduce scheduling is the key problem of the algorithm. Distinctly, the optimum point can minimize the system delay and maximize the resource utilization.

The SARS algorithm works by delaying the reduce processes.The reduce tasks are scheduled when part but not all of the map tasks are finished. For a special key value, if we assume that there are s map slots and m map tasks in the current system, the completion time and the size of output data of each map task are denoted as t_mapi and m_outjrespectively, where i, j ∈ [1,m].Then, we can know the amount of the map tasks data can be calculated as:

$$N\_m=\sum_{i=0}^{m} m\_out_j,\ j \in [1,m] \rightarrow (1)$$

In order to predict the time required to transmit the data, it defines the speed of data transmission from the map tasks to the reduce tasks as transSpeed in the cluster environment, and the number of concurrent copy threads with reduce tasks is denoted as copyThread. We denote the start time of the first map and reduce task as startmap and start reduce respectively.

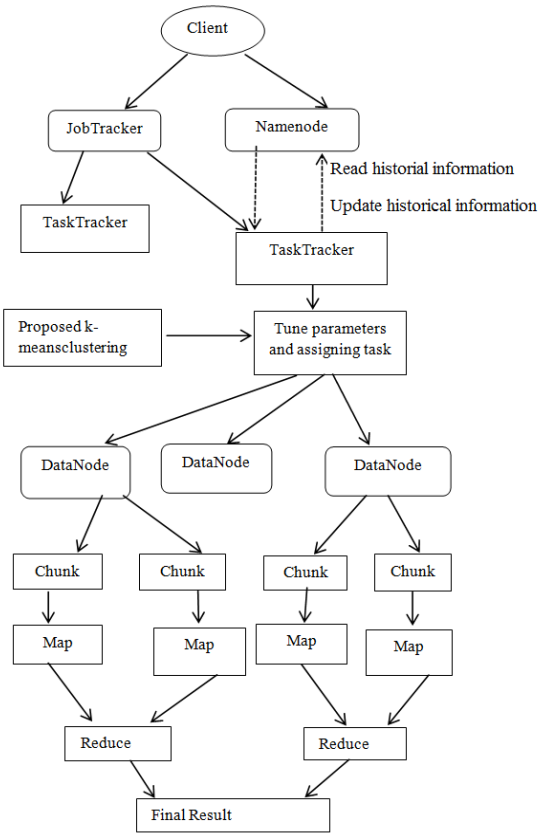

Fig 2: Reduce-phase implementation

Therefore, the optimalstart time of reduce tasks can be determined by the followingequation:

Start_reduce= startmap+1/s($\sum_{i=1}^{m}$ t_mapi –(N_mtransSpeed× copyThread))→(2)

The most appropriate start timeof a reduce task is when all the map tasks about the same key arefinished, which is between the times when the first map is startedand when the last map is finished. On the other hand, for mapand copy processes start almost at the same time, which is calledintermittent copy (once map has an output, copying followed). The scheduling method for reduce task in SARS causes the massive waste of slot resources. For the reduce tasks willbe started before the map task finished, the time cost should be cutfrom the map tasks completion time. Therefore, ituse the averagecompletion time of map tasks to minus data transmission timewhich produced by map task in Eq. (2), the value can be roughlyconsidered as the starting time of the no-intermission copy, whichis reduce tasks starting time in this paper. The waiting around ofthe reduce tasks may make the jobs in need of the slot resources notable to work normally. Through adjusting the reduce schedulingtime, this method can decrease the time waste for data replicationprocess and advance the utilization of the reduce slot resourceseffectively.

4.2SARS Scheduling Algorithm

Step 1: Obtain the MapTask from the queue(M, Q), start the map task in MT.
Step 2: Obtain KVS as the middle key-value set and compare with key values for mapping completing the map jobs.
Step 3: When map phases completes by using KVM,get the starting time of reduce task
Step 4: By using transSpeed and copyThreadvalues, if job exists waiting reduce task then 16: remove a reduce slot from R.
Step 5: Ends when the candidate set is empty.

Upon receiving heart beat messages from a note manager reporting resource availability on the node, the scheduler must select which phase should be scheduled on the node.

Using the job's own characteristics to determine the reducescheduling time can use the slot resources effectively. The improvementof these policies is especially important for the CPUtypejobs. For these jobs which need more CPU computing, the dataI/O of the tasks are less, so more slot resources will be wasted in thedefault schedule algorithm.

VCONCLUSION:

The goal of the improved algorithm proposed in the paper is to decrease the completion time of reduce tasks in the MapReduce framework. This method can decide reduce task start time through running situation of the jobs, and to avoid waiting around of the reduce function. In this paper, the performance of this algorithm is estimated from the job completion time, reduce completion time, and the system average response time.

The experimental results illustrate that when comparing with the FIFO, the reduce completion time is decreased sharply. It is also proved that the average response time is decreased by 11% to 29% when the SARS algorithm is applied to traditional job scheduling algorithms FIFO, FairScheduler, and CapacityScheduler,Phase-level scheduler.

However, there are also some difficulties in the experiments.The first is the speed of the network transmission. When multiplereduce tasks from one TaskTracker run at the same time, they may lead to the network I/O competition and lower the transmission speed during the copy stage.

REFERENCES

[1] Hadoop MapReduce distribution.Available:http://hadoop.apache.org, 2015.
[2] HadoopCapacityScheduler. Available: http://hadoop.apache.org/docs/stable/ capacity_scheduler.html/, 2015.
[3] Hadoop Fair Scheduler [Online]. Available:http://hadoop.apache.org/docs/r0.20.2/fair_scheduler.html, 2015.
[4] Hadoop Distributed File System [Online]. Available: hadoop. apache.org/docs/hdfs/current/, 2015.
[5] The Next Generation of Apache HadoopMapReduce[Online].Available:http://hadoop.apache. org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html, 2015.
[6] R. Boutaba, L. Cheng, and Q. Zhang, "On cloud computational models and the heterogeneity challenge," J. Internet Serv. Appl., vol. 3, no. 1, pp. 1–10, 2012.
[7] T. Condie, N. Conway, P. Alvaro, J. Hellerstein, K. Elmeleegy, and R. Sears, "MapReduce online," in Proc. USENIX Symp. Netw. Syst. Des. Implementation, 2010, p. 21.
[8] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
[9] Xiang Gao, Qing Chen, Yurong Chen, Qingwei Sun, Yan Liu, Mingzhu Li, Adispatching-rule-based task scheduling policy for MapReduce with multitypejobs in heterogeneous environments, in: 2012 Seventh ChinaGrid AnnualConference (ChinaGrid), pp. 17–24.
[10] Jiaqi Zhao, Lizhe Wang, Jie Tao, Jinjun Chen, Weiye Sun, Rajiv Ranjan, Joanna,G-Hadoop for big data computing across distributed Cloud data centres,J. Comput. System Sci. 80 (5) (2014) 994–1007.Qin, Research on scheduling scheme for hadoop clusters, Proc. Comput. Sci. 18(2013) 2468–2471.
[11] Zhuo Tang, Min Liu, Kenli Li, Yuming Xu, A MapReduce-enabledscientificworkflow framework with optimization scheduling algorithm, in: 2012 13thInternational Conference on Parallel and Distributed Computing.
[12] Joanna Kolodziej, Samee Ullah Khan, Lizhe Wang, AleksanderByrski, NasroMin-Allah, Sajjad Ahmad Madani, Hierarchical genetic-based grid schedulingwith energy optimization, Cluster Comput. 16 (3) (2013) 591–609.