ISSN:0975-9646

Scheduling of Parallel Applications Using Map Reduce On Cloud: A Literature Survey

A.Sree Lakshmi[#1], Dr.M.BalRaju[*2], Dr.N.Subhash Chandra[#3]

[#1] *Associate Professor, Geethanjali College of Engineering and Technology, Hyderabad, Telangana, India*

[*2] *Principal, Avanthi College of Engineering , Hyderabad , Telangana, India*

[#3] *Dean-Academics and Professor of CSE ,Vignan Bharathi College of Engineering, Hyderabad , Telangana, India*

***Abstract*— Most of the current day applications process large amounts of data. There were different trends in computing like mainframes, parallel computing, cluster computing, grid computing as per the requirement of the data size and execution speed. Cloud computing is the new era of computing where efficient utilization of resources can be done with no compromise on data size, execution time and cost of execution. Map Reduce is a programming model which is widely used for processing large scale data intensive applications in cluster, cloud environments. In this paper we have discussed various scheduling algorithms of map reduce tasks. The default schedulers available with Hadoop can be improved to make it more efficient for the cloud environments**

Keywords: **HDFS, hadoop, map reduce, virtual machine**

INTRODUCTION

A cloud scheduler plays a main role in distributing resources for different jobs executing in cloud environment. Virtual machines are created and managed on the fly in cloud to create an environment for job execution. Map Reduce is a simple and powerful programming model which has been widely used for processing large scale data intensive applications on a cluster of physical machines. Now a day's many organizations, researchers, government agencies are running Map Reduce applications on public cloud. Running Map Reduce on cloud has many advantages like on-demand establishment of cluster, scalability.

Many Map Reduce Frameworks like Google Map Reduce, Dryand, are available but the open source Hadoop Map Reduce is commonly used. But running a Hadoop cluster on a private cluster is different from running on public cloud .Public cloud enables to have virtual cluster where resources can be provisioned or released as per the requirement of the application in minutes. Executing Map Reduce applications on cloud enables user to execute jobs of different requirements without taking any pain of creating and maintaining a cluster.

Scheduling plays a major role in the performance of Map Reduce Applications. The default scheduler in Hadoop Map Reduce is FIFO Scheduler, Facebook uses Fair Scheduler, and Yahoo uses Capacity Scheduler. The above schedulers are typical examples of schedulers for Map Reduce application are best suitable for physical static clusters ,that can also serve the cloud systems with dynamic resource management, but these schedulers does not consider the features affected by virtualization used in cloud environments. Therefore, these is a need of dynamic scheduler which can schedule Map Reduce applications based on the features of the application , Virtual Machines and locality of input data to efficiently execute these applications in hybrid cloud environment.

BACKGROUND

Hadoop framework works in two layers : Hadoop Distributed File System(HDFS) which stores the data and Map Reduce framework which processes the data. HDFS is based on Master/Slave architecture where a single NameNode acts as Master and many DataNodes act as slaves. NameNode manages namespace and scheduling of jobs. DataNodes manage the data attached to them and performs the task given by the Name node.

Map Reduce is a distribute data analysis framework initially introduced by Google which provides very useful features like ease of programming, automatic parallelization, scalability, fault tolerance, data locality awareness. Map Reduce is well suited for large scale data processing in different environments like cluster, cloud. Map Reduce processing includes both sequential and parallel processing. It is divided into two phases: Map phase and Reduce phase. Reduce phase executes after Map phase, but many map and reduce tasks are executed in parallel. Map tasks run on Data Nodes on the input data chunks provided by the Master node (Name node) and produces key value pairs (K,V)which are written back to HDFS. The intermediate results generated by the map phase are sorted and merged using merge sort. Reducers receive the input corresponding to same key and reduce function is performed on these key value pairs as written by the user.

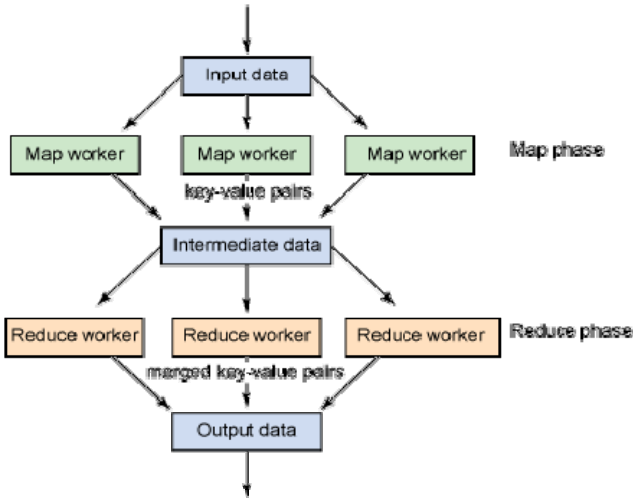

Fig1: Map Reduce Framework

RELATED WORK

A. *MAPREDUCE SCHEDULING IN HADOOP*

The most fundamental part of NameNode is job scheduling. FIFO (First In First Out) scheduling algorithm is the default scheduler used by Hadoop MapReduce and designed for running large batch jobs. However users are given a chance to change their scheduling algorithm from FIFO to Fair scheduler or Capacity Scheduler. Fair scheduler is included with delay scheduler from the release of Hadoop 0.21. In FIFO scheduler [1], jobs are scheduled based on their job submission time and their priorities. This approach schedules one job to use all task slots and other jobs cannot use it until the current job completes. This increases the execution time of the jobs that are waiting ahead in the queue.

Capacity Scheduler [2] uses multiple queues/pools where each pool is guaranteed some fraction of physical resources in the cluster which enables more jobs to be executed concurrently. Fair scheduler creates pools for multiple users where each pool is guaranteed a share of resources fairly which results in fair share of the resources in the cluster and more jobs can be processed concurrently. The draw backs of Capacity and Fair Scheduler are increased execution times and less sharing opportunities.

Delay Scheduler of Hadoop do not impose strict rule of queuing for the tasks in the scheduling process. If the scheduler does not find a data local task, it is delayed in its execution and the task next to it in the queue is scheduled. After sometime, the task may become data local and then be scheduled. It will run in non-data local manner if the scheduler cannot find any data local task
after certain time. The authors argue that reduce phase has to wait for all the tasks to complete which could considerable degrade the response time of the application. To overcome this, they have proposed a solution in which the reduce task would be split into two logical distinct type of tasks, copy and compute tasks with different ways of admission control. Copy tasks perform fetching and merging of map inputs, an operation which is usually network-I/O bound. Compute tasks perform user defined reduce function on the map outputs. Copy-compute splitting now become two separate processes for copy and compute tasks, and there by scheduling these tasks separately along with distinctive tasks increases overall performance.

Many scheduling strategies like Dynamic MR , MR Share , S3 Shared Scan Scheduler, Corona are proposed by different researches .Dynamic MR dynamically allocates unused map(Reduce) slots to overloaded reduce (map) slots to maximize slot utilization as much as possible. It also proposed an efficient speculative task scheduler. MRShare is a scheduling strategy which batches jobs that access same file and processes them as a single batch which reduces the execution time instead of reading the same file multiple times. S3(Shared Scan Scheduler) is a scheduler that shares scanning of a common file for multiple files Unlike MRShare , S3 can batch jobs that arrive at different times as it does not require a job to begin its processing from the starting segment of the file thus can schedule processing from any segment of the file. MRShare and S3 framework assumes that we know the query patterns and the jobs are batched before their execution. This assumption is not practical for cloud environment. All the above mentioned scheduling strategies are more suitable for physical cluster than cloud environments.

B. *VM SCHEDULING ON CLOUD*

Round Robin, Greedy and Power Save Round Robin, Greedy and Power Save algorithms are the virtual machine scheduling algorithms provided along with Eucalyptus open source cloud operating system distribution. Round Robin algorithm follows the basic mechanism of allocating the incoming virtual machine requests on to physical machines in a circular fashion. It is simple and starvation-free scheduling algorithm which is used in most of the private cloud infrastructures. The Greedy algorithm allocates the virtual machine to the first physical machine which has enough resources to satisfy the resources requested by it. In Power Save algorithm, physical machines are put to sleep when they are not running any virtual machines and are re-awakened when new resources are requested. First, the algorithm tries to allocate virtual machines on the physical machines that are running, followed by machines that are asleep. These algorithms have limited or no support for making scheduling decisions based on the resource usage statistics. Moreover these algorithms do not take into account SLA violations, energy consumed etc., which form very important factors in real time cloud environments.

2.1 Dynamic Round Robin

Ching-Chi Lin et. al in[3] presented an improved version of Round Robin algorithm used in Eucalyptus. According to Dynamic Round Robin algorithm, if a virtual machine has finished its execution and there are still other virtual machines running on the same physical machine, this

physical machine will not accept any new virtual machine requests. Such physical machines are referred to as being in 'retirement' state, meaning that after the execution of the remaining virtual machines, this physical machine could be shutdown. And if a physical machine is in the retirement' state for a sufficiently long period of time, the currently running virtual machines are forced to migrate on to other physical machines and shutdown after the migration operation is finished. This waiting time threshold is denoted as 'retirement threshold'. So, a physical machine which is in the retirement state beyond this threshold will be forced to migrate its virtual machines and shutdown. Even this algorithm has limited support for making scheduling decisions based on the resource usage statistics and does not take into account of SLA violations, energy consumed etc.

2.2 Single Threshold

Single Threshold algorithm[4] sorts all the VMs in decreasing order of their current utilization and allocates each VM to a physical machine that provides the least increase of power consumption due to this allocation. The algorithm does optimization of the current allocation of VMs by choosing the VMs to migrate based on CPU utilization threshold of a particular physical machine called 'Single Threshold'. The idea is to place VMs while keeping the total utilization of CPU of the physical machine below this threshold. The reason for limiting CPU usage below the threshold is to avoid SLA violation under a circumstance where there is a sudden increase in CPU utilization of a VM, which could be compensated with the reserve. Single Threshold algorithm works better in terms of energy conservation when compared to Dynamic Round Robin Algorithm discussed in 2.1. This algorithm is fairly improved one which takes into consideration of power consumption and CPU usage of physical machines.

C. MAP REDUCE SCHEDULING ON CLOUD

Wei and Tian [5] presented a genetic-algorithm-based (GA-based) scheduler for task level scheduling in Hadoop MapReduce. Ge and Wei pointed out that the performance can further be improved by considering a realistic view of all the tasks that are waiting for processing. Ge and Wei's algorithm considers the entire group of tasks in the job queue and makes a scheduling decision. A genetic algorithm is developed as the optimization method for scheduling. Unfortunately, Ge andWei's algorithm is not applicable to a hybrid cloud environment that supports task scheduling with QoS constraints.

Keke Chen, James Powers, Guo & Tian [6] has proposed a optimal resource provisioning for Map reduce computing in public clouds. They used combination of white box and machine learning techniques to build a cost function that models relationship among the available resources, amount of data, complexity of reduce function etc. Machine learning techniques are used to learn about model parameters.

Zhang , Feng, Ming [7] has proposed a data locality aware task scheduling method foe Map Reduce Framework in Heterogeneous environment. Data locality aware scheduler addresses the issue where the cloud is composed of different machines with different processing power. They obtained optimal task execution time by using tradeoff between transmission time and waiting time to schedule a task to a node. Their method schedules a tack if it is local to that node or else it selects a task whose input data is near to that node. The drawback of this method is it does not consider other parameters like priority, size of data sets.

Kim, Kang[8] has proposed an burstiness aware I/O scheduler for Map Reduce on virtualized environments. Their idea is to identify bursty virtual machines on-line and schedule them in a round robin fashion with relatively large time quantum, during which a scheduled VM can exploit most of the disk bandwidth in an isolated fashion, thereby decreasing I/O interferences. Their approach schedules the VM based only on burstiness irrespective of the other factors that affect the performance to a greater extent.

Hammoud, Sakrhas [9] modeled a center-of-gravity reduce task scheduling which improves the performance of reduce operation by considering the data locality and partitioning skew in Map Reduce task scheduling. But their model is based on static determination of sweep points which can be made to involve dynamic nature which is more apt for cloud environments.

Polo et al. [10] introduced online job completion time estimator which can be used for adjusting the resource allocations of different jobs. Their estimator tracks the progress of only the map stage and has does not use information of the reduce stage. Phan et al. [11] modeled an off-line optimal schedule for a batch of Map Reduce jobs with given deadlines by formulating the scheduling problem as a constraint satisfaction problem (CSP) with detailed task ordering of these jobs. Purlieus [12] improved the allocation of map and reduce tasks in Map Reduce platforms on the cloud by locality and load aware VM placement by exploiting prior knowledge about the characteristics of Map Reduce workloads and categorizing the jobs as Map intensive or Reduce intensive. Distributed Resource Scheduler (DRS) [13] places VMs to maximize the utilization, and performs live-migration of VMs across physical systems to avoid resource conflicts in a system. Though live migration provides improved performance and fault tolerance it causes a significant downtime in a oversubscribed system when serving many concurrent users.

To efficiently take the advantage of parallel execution of big data applications using Map Reduce on cloud, there is a requirement of designing a scheduler which provides high performance with no compromise on manageability, fault tolerance

CONCLUSION AND FUTURE WORK

In current day world as there is huge increase in volumes of data and big data has become an important point of research. This paper has discussed scheduling of Map Reduce parallel applications on cloud. There have been an active research in the area of scheduling of the map and reduce tasks to virtual machines to improve the performance of map reduce applications. Most of the scheduling algorithms concentrate on map tasks data locality.

Scheduling can be made efficient by using the knowledge of data locality of the intermediate data generated by the map tasks. This knowledge helps out to reduce the intermediate network traffic during the reduce phase and there by speeding the execution of map reduce applications.

REFERENCES

[1] http://hadoop.apache.org/docs/r1.2.1/fairscheduler.html

[2] http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html

[3] Ching-Chi Lin, Pangfeng Liu, and Jan-JanWu. Energy-*aware virtual machine dynamic provision and scheduling for cloud,*. In Cloud Computing (CLOUD), 2011 IEEE International Conference on, pages 736 –737, july 2011.

[4] Anton Beloglazov and Rajkumar Buyya. *Energy efficient allocation of virtual machines in cloud data centers*, In 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pages 577–578, 2010.

[5] Yibin Wei , Ling Tian , *Research on cloud design resources scheduling based on Genetic Algorithm*, 2012 International Conference on systems and informatics(ICSAI 2012)

[6] Chen, K. ; Powers, J. ; Guo, S. ; Tian, F. CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds , *IEEE Transactions on Parallel and Distributed Systems ,Volume: 25 , Issue: 6 Publication Year: 2014 , Page(s): 1403 – 1412*

[7] Xiaohong Zhang ; Yuhong Feng ; Shengzhong Feng ;Jianping Fan ; Zhong Ming *An effective data locality aware task scheduling method for MapReduce framework in heterogeneous environments*, 2011 International Conference on Cloud and Service Computing (CSC)Year: 2011 , Page(s): 235 - 242

[8] Sewoog Kim ; Dongwoo Kang ; Jongmoo Choi ; Junmo Kim Burstiness-aware I/O scheduler for MapReduce framework on virtualized environments , *2014 International Conference on Big Data and Smart Computing (BIGCOMP) Publication Year: 2014 , Page(s): 305 – 308*

[9] Hammoud, M. ; Rehman, M.S. ; Sakr, M.F. Center-of-Gravity Reduce Task Scheduling to Lower MapReduce Network Traffic , *2012 IEEE 5th International Conference on Cloud Computing (CLOUD) Publication Year: 2012 , Page(s): 49 - 58*

[10] J. Polo, D. Carrera, Y. Becerra, J. Torres, E. Ayguad´e, M. Steinder, and I. Whalley. *Performance-driven task co-scheduling for MapReduce environments.* In *12th IEEE/IFIP Network Operations and Management Symposium*. ACM, 2010.

[11] L. Phan, Z. Zhang, B. Loo, and I. Lee. *Real-time MapReduce Scheduling*. Tech. Report No. MS-CIS-10-32, UPenn, 2010.

[12] B. Palanisamy, A. Singh, L. Liu, and B. Jain. Purlieus: *locality-aware resource allocation for MapReduce in a cloud*. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2011.

[13] Resource management with VMware DRS http://www.vmware.com/pdf/vmware_drs_wp.pdf.