An Improved Job Scheduling Plan for Hadoop Running in Clouds

Krishan Tuli
Chandigarh Group of Colleges, Landran, Mohali

Abstract- *Hadoop has become the most popular knowledge management framework for parallel data-intensive computing within the clouds. Hadoop is an open supply implementation of Map cut back, currently maintained by the Apache Foundation, and supported by leading IT companies such as Facebook and Yahoo.*

In sequence to get more allocation of load equalization, the MapReduce work mode and task scheduling algorithmic rule of Hadoop platform is calculated in this paper. This review paper explains the various jobs programming facet random job, smart and DW flooding. According to this method variety of tasks of an equivalent weight job is giant, while that of the larger weight job is less, this paper introduces the idea of weighted round-robin programming algorithmic rule into the task programming of Hadoop. This paper introduce algorithm for job programming mistreatment random job, smart, DW flooding. In an assignment theme Hadoop running in the cloud there square measure appropriate for same. We discuss the problems with the Hadoop task assignment theme, associated present an improved theme for heterogeneous computing environments, such as the general public clouds. The proposed theme is based mostly on interval, response time, and scalability.

Keywords- Cloud Computing, Hadoop, Map Reduce, Job Scheduling in Hadoop.

I. INTRODUCTION

Cloud computing is a big-scale distributed computing paradigm that is derived by economy of scale, in which a pool of abstracted, virtualized, dynamical scalable, managed computing power, storage, platforms and services are delivered on demand to external customers over the Internet. So far, Google, Microsoft, IBM, Amazon and other IT business giants launched their cloud computing platforms and viewed the cloud computing as the future development of one of the most main strategy. Therefore, the cloud computing research not only follows the trend of the development of the technology industry, also has a high application value. At present, most cloud computing systems use Hadoop to develop and schedule programs, which is the open source programming development platform based on MapReduce model and task scheduling tool. Hadoop is a distributed computing framework, and it can run applications in the cluster that consists of a large number of inexpensive hardware. Hadoop can provide a stable and reliable interface for the application and build a distributed system with high reliability and good scalability for users. Hadoop also provides a reliable shared storage and analysis system. Hadoop Distribute File System (HDFS) implements storage, while MapReduce realizes analysis processing, the two part are the core of Hadoop. However, Hadoop is still a new framework that needs to be improved in some aspects.Task Scheduling technology, one of the key technologies of Hadoop platform, mainly controls the order of task running and the allocation of computing resources, which is directly related with overall performance of the Hadoop platform and system resource utilization. Default scheduling algorithm that Hadoop platform provides is FIFO. The advantages of FIFO include simple idea and easy to be executed, light workload of job server etc. The disadvantages of FIFO lie in ignoring the different needs by different operations. For example, if a job analyzing massive data occupies computing resources for a long time, then subsequent interactive operations may not be processed timely. Therefore, this situation may lead to long response time and affect the users' experience. Through analyzing the MapReduce work mode and studying the Hadoop platform architecture and existing task scheduling algorithm, this paper proposes an improved weighted round-robin task scheduling algorithm (IWRR) which is easy to be understood and implemented.

II. BIG DATA

With the increase of technologies and services, the big amount of data is giving that can be structured and unstructured from the various sources. Big data handle the large amount of information used to uncover the hidden patterns

Big Data Parameters

Data is too large from various sources in various forms; it is represented by the 3Vs.

The three Vs of Big Data are:

- Variety
- Volume

- Velocity

Fig: 1 - Big data parameter

Volume: Many factors contribute to the increase in data volume. Transaction-based data stored through the years. Unstructured data streaming in from social media. Increasing amounts of sensor and machine-to-machine data being collected. In the past, excessive data volume was a storage issue. But with decreasing storage costs, other issues emerge, including how to determine relevance within large data volumes and how to use analytics to create value from relevant data.

Velocity: Data is streaming in at unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time. Reacting quickly enough to deal with data velocity is a challenge for most organizations.

Variety: Data today comes in all types of formats. Structured, numeric data in traditional databases. Information created from line-of-business applications. Unstructured text documents, email, video, audio, stock ticker data and financial transactions. Managing, merging and governing different varieties of data is something many organizations still grapple with.

III. HADOOP

This is representing if the applications is running on clusters. Apache Hadoop is an open source implementation of the Google's MapReduce parallel processing framework. Hadoop hides the details of parallel processing, including data distribution to processing nodes, restarting failed subtasks, and consolidation of results after computation.

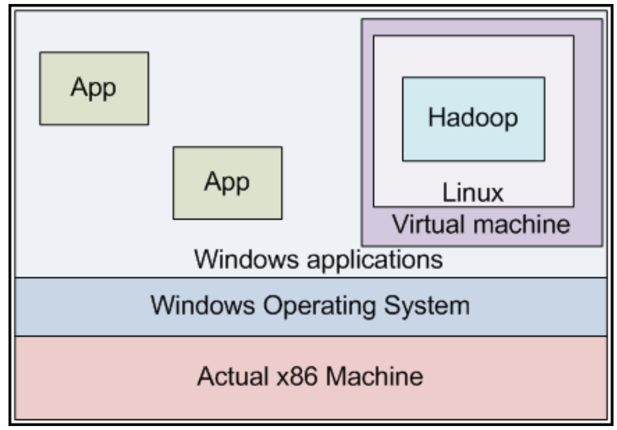

Fig: 2 - A virtual machine encapsulates one operating system within another

This framework allows developers to write parallel processing programs that focus on their computation problem, rather than parallelization issues.

Hadoop includes:-

1) **Hadoop Distributed File System (HDFS):** A distributed file system that store large amount of data with high throughput access to data on clusters.

The main components of HDFS are as described below:

- **NameNode** is the master of the system. It maintains the name system (directories and files) and manages the blocks which are present on the DataNodes.
- **DataNodes** are the slaves which are deployed on each machine and provide the actual storage. They are responsible for serving read and write requests for the clients.
- **Secondary NameNode** is responsible for performing periodic checkpoints. In the event of NameNode failure, you can restart the NameNode using the checkpoint.

Machine Type	Workload Pattern/ Cluster Type	Storage	Processor (# of Cores)	Memory (GB)	Network
Slaves	Balanced workload	Four to six 2 TB disks	One Quad	24	1 GB Ethernet all-to-all
	HBase cluster	Six 2 TB disks	Dual Quad	48	
Masters	Balanced and/or HBase cluster	Four to six 2 TB disks	Dual Quad	24	

Fig: 3-For small clusters (5-50 nodes)

2) **Hadoop Map Reduce:** A software framework for distributed processing of data on clusters. MapReduce provides an easy parallel programming interface in a distributed computing environment. Also MapReduce deals with fault tolerance issues for managing multiple processing nodes. The most powerful feature of MapReduce is its high scalability that allows user to process a vast amount of data in a short time. The structure of MapReduce is primarily based on the master-slave architecture. A single master node monitors the status of the slave nodes and assigns jobs to them. A task assigned to slave nodes has two phases of processing; map and reduce. From the map process, intermediate results are generated and transferred to the reduce process as input. The reduce process sorts the intermediate results by keys then merges them as one final output. There is a synchronization step between the map and reduce processes. The synchronization phase is a data communication step between the mapper and reducer nodes to launch the reduce process. The architecture of the MapReduce model follows a shared nothing structure where each task on a node has no knowledge of other tasks. This leads to simplicity of data processing and fault tolerance.

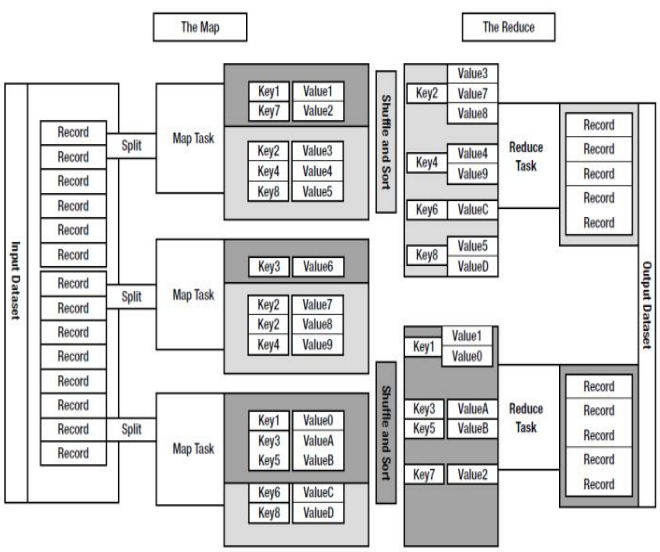

Fig: 4- Map Reduce Model

IV. OPTIONS FOR RUNNING HADOOP IN THE CLOUD

Map Reduce as a Service - Amazon's EMR (Elastic MapReduce) provides a quick and easy way to run MapReduce jobs without having to install a Hadoop cluster on its cloud.

Hadoop on S3 - Hadoop give the all option that is already stored in S3 for your business options. (e.g. Netflix uses a Hadoop cluster using S3).

Hadoop in private cloud - In case of a private cloud, you may have infrastructure that will enable you to provision bare-metal servers. Some of these private cloud solutions also provide a Paas layer deploying Hadoop clusters easily (e.g. IBM offers patterns for deploying Info Sphere Big Insights on their Smart Cloud Enterprise).

V. KEY THINGS TO CONSIDER BEFORE DEPLOYING A HADOOP CLUSTER IN THE CLOUD

Your enterprise should running the security for introduce workloads in public cloud. Hadoop cluster security is very small. There is no protection for data that will safe enterprise data security requirements around SOX, PII, HIPPA, etc. Implementation Hadoop distributions that you would want to use and the operating system standards of your enterprise. However, with this method you will be tied to VMware's Hypervisor condition to consider when choosing a cloud provider.

VI. SCHEDULING IN HADOOP

The default Scheduling algorithm is based on FIFO where jobs were executed in the order of their submission. Later on the ability to set the priority of a Job was added. Facebook and Yahoo contributed significant work in developing schedulers i.e. Fair Scheduler and Capacity Scheduler respectively which subsequently released to Hadoop Community.

- ❖ Default FIFO Scheduler: The default Hadoop scheduler operates using a FIFO queue. After a job is partitioned into individual tasks, they are loaded into the queue and assigned to free slots as they become available on Task Tracker nodes. Although there is support for assignment of priorities to jobs, this is not turned on by default.

- ❖ Fair Scheduler: The Fair Scheduler was developed at Facebook to manage access to their Hadoop cluster and subsequently released to the Hadoop community. The Fair Scheduler aims to give every user a fair share of the cluster capacity over time. Users may assign jobs to pools, with each pool allocated a guaranteed minimum number of Map and Reduce slots. Free slots in idle pools may be allocated to other pools, while excess capacity within a pool is shared among jobs. In addition, administrators may enforce priority settings on certain pools. Tasks are therefore scheduled in an interleaved manner, based on their priority within their pool, and the cluster capacity and usage of their pool. As jobs have their tasks allocated to Task Tracker slots for computation, the scheduler tracks the deficit between the amount of time actually used and the ideal fair allocation for that job. As slots become available for scheduling, the next task from the job with the highest time deficit is assigned to the next free slot. Over time, this has the effect of ensuring that jobs receive roughly equal amounts of resources. Shorter jobs are allocated sufficient resources to finish quickly. At the same time, longer jobs are guaranteed to not be starved of resources.

- Capacity Scheduler: Capacity Scheduler originally developed at Yahoo addresses a usage scenario where the number of users is large, and there is a need to ensure a fair allocation of computation resources amongst users. The Capacity Scheduler allocates jobs based on the submitting user to queues with configurable numbers of Map and Reduce slots. Queues that contain jobs are given their configured capacity, while free capacity in a queue is shared among other queues. Within a queue, scheduling operates on a modified priority queue basis with specific user limits, with priorities adjusted based on the time a job was submitted, and the priority setting allocated to that user and class of job. When a Task Tracker slot becomes free, the queue with the lowest load Is chosen, from which the oldest remaining job is chosen. A task is then scheduled from that job. Overall, this has the effect of enforcing cluster capacity sharing among users, rather than among jobs, as was the case in the Fair Scheduler

VII. CONCLUSION & FUTURE WORK

Ability to make Hadoop scheduler resource aware is one the emerging research problem that grabs the attention of most of the researchers as the current implementation is based on statically configured slots. This paper summarizes pros and cons of Scheduling policies of various Hadoop Schedulers developed by different communities. Each of the Scheduler considers the resources like CPU, Memory, Job deadlines and IO etc. All the schedulers discussed in this paper addresses one or more problem(s) in scheduling in Hadoop. Nevertheless all the schedulers discussed above assumes homogeneous Hadoop clusters. Future work will consider scheduling in Hadoop in Heterogeneous Clusters. In the past, a number of scheduler algorithms have been developed specifically to suit the dynamic cloud computing environments such as ECT algorithm, FIFO algorithm, capacity scheduling algorithm, and weighted round robin algorithm. We are going to use the job scheduling aspect random job, smart and DW flooding. In this research work, we proposed an algorithm for job scheduling using random job, smart, DW flooding.

REFERENCES

[1] L. Mashayekhy, M. Nejad, D. Grosu, Q. Zhang and W. Shi, "Energy-Aware Scheduling of MapReduce Jobs for Big Data Applications", IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 10, pp. 2720-2733, 2015.

[2] Y. Yao, J. Wang, B. Sheng, C. Tan and N. Mi, "Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters", IEEE Transactions on Cloud Computing, pp. 1-1, 2015.

[3] Y. Wang, R. Rao and Y. Wang, "A round robin with multiple feedback job scheduler in Hadoop", 2014 IEEE International Conference on Progress in Informatics and Computing, 2014.

[4] C. Chen, B. Lee and X. Tang, "Improving Hadoop Monetary Efficiency in the Cloud Using Spot Instances", 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, 2014.

[5] C. Tsai, W. Huang, M. Chiang, M. Chiang and C. Yang, "A Hyper-Heuristic Scheduling Algorithm for Cloud", IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 236-250, 2014.

[6] A. Raj, K. Kaur, U. Dutta, V. Sandeep and S. Rao, "Enhancement of Hadoop Clusters with Virtualization Using the Capacity Scheduler", 2012 Third International Conference on Services in Emerging Markets, 2012.

[7] Mehul Nalin Vora, "Hadoop-HBase for large-scale data", Proceedings of 2011 International Conference on Computer Science and Network Technology, 2011.

[8] Kaur, P.D., Chana, I. “Unfolding the distributed computing paradigm”, In: International Conference on Advances in Computer Engineering, pp. 339-342 (2010)

[9] Mei, L., Chan, W.K., Tse, T.H., “A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues”, In: APSCC 2008, pp. 464-469 (2008)