Achieving Best Job Performance by Increasing the Virtual Map Reduce Clusters

Matla Himagireshwar Rao[1], Jomma Prathap[2]
[1]*M.Tech, Assistant Professor, Dept of CSE, Vidya Jyothi Institute Of Technology, Aziz nagar, Hyderabad , Telangana*
[2]*M.Tech, Dept of CSE, Vishwa Bharathi College of Engineering, Kukatpally, Hyderabad, Telangana*

Abstract— **MapReduce job as a map function and a reduce function, and provides a runtime system to divide the job into multiple map tasks and reduce tasks and perform these tasks on a MapReduce cluster in parallel. In order to provide high map and reduce data locality, we proposed an efficient and suitable scheduling scheme named as hybrid job-driven scheduling scheme (JoSS) for the users. But, in this existing scheduling scheme, virtual MapReduce workload problem is occurred. So, in this paper we enhance this JoSS scheme work with heterogeneous virtual MapReduce clusters by providing flexibility for JoSS. In this proposed work, we are providing individual servers for individual jobs to reduce the MapReduce workload. We can achieve the high map and reduce data locality and also we can achieve the best job performance through the heterogeneous virtual Mapreduce clusters.**

I. INTRODUCTION

In present years, MapReduce has come to be a famous version for data-extensive computation. The schedulers are important in improving the performance of MapReduce/Hadoop in presence of multiple jobs with specific traits and overall performance goals. The endorse enhance the resource aware scheduling approach for Hadoop map-reduce multiple jobs jogging that pursuits to improving aid usage across a couple of digital machines whilst watching of completion time goals. The present algorithm impacts activity profiling data to dynamically alter the range of slots allocation based totally on process profile and aid utilization on every system, in addition to workload placement across them, to maximize the useful resource usage of the cluster.

MapReduce jobs are typically executed on clusters of commodity PCs, which require huge funding in difficult-ware and management. Since a cluster has to be provisioned for peak utilization to keep away from overload, it is underutilized on common. Thus, cloud turns into a promising platform for MapReduce jobs because of its flexibility and pay-as-you-go enterprise version. For every MapReduce activity, a virtual cluster is created by way of employing some of Virtual Machines (VMs). The cluster size may be dynamically adjusted consistent with jobrequirements. However, the services furnished through an individual cloud issuer are generally restricted to positive geographic regions, making it not possible to method data from all around the globe. To really satisfy the promise of cloud computing for big data utility, an rising scheme is to save and technique data in a geographically dispersed cloud surroundings, in which a couple of clouds locate at extraordinary places within the international and they may be related by way of inter-cloud networks.

A cloud scheduler performs a main role in distributing sources for unique jobs executing in cloud environment. Virtual machines are created and managed at the fly in cloud to create surroundings for task execution. Map Reduce is a simple and effective programming version which has been extensively used for processing large scale records extensive programs on a cluster of physical machines. Now a day’s many groups, researchers, authorities businesses are strolling Map Reduce packages on public cloud. Running Map Reduce on cloud has many benefits like on-call for establishment of cluster, scalability.

Today’s data centers provide specific modes of computing structures - local and virtual clusters. Both those environments are having their very own strengths and weaknesses. For example, a local cluster is higher for batch workloads like MapReduce from the overall performance angle, lowers SLA violations, and however generally suffers from poor utilization, and excessive hardware and electricity cost. A virtual cluster, then again, is appealing for interactive workloads from consolidation and price standpoints, but won't provide aggressive performance like a local cluster, and incurs higher SLA infringements. Intuitively, a hybrid platform along with native and virtualized cluster must be able to take advantage of the blessings of both environments for providing a better cost-effective platform. In this paper, we discover this layout alternative, which we call hybrid data center, and demonstrate its benefits for helping each interactive and batch workloads, and attaining the right stability between these types of layout standards, making it a suitable cluster configuration alternative.

2. RELATED WORK

Jongse Park, Daewoo Lee, Bokyeong Kim, Jaehyuk Huh, Seungryoul Maeng proposed and evaluated a dynamic VM reconfiguration mechanism for distributed data-intensive platforms on virtualized cloud environments, called Dynamic Resource Reconfiguration (DRR). DRR improves the input data locality of a virtual MapReduce cluster, by temporarily increasing cores to VMs to run local tasks. DRR schedules tasks based on data locality, and adjust the computational capability of the virtual nodes to accommodate the scheduled tasks. This approach differs from prior approaches assuming a cluster which always has a fixed amount of computational resource in each node. Using dynamic VM reconfiguration for distributed data-intensive platforms can be extended to different types of load imbalance. Different resource requirements by different tasks or jobs may cause each virtual node to under-utilize its resource. With VM reconfiguration, each node can be adjusted to provide only the necessary amount of resource demanded for the node. Such a generalized framework with dynamic VM reconfiguration will be their extension work.

Balaji Palanisamy, Aameek Singh, Ling Liu and Bhushan Jain proposed a system model, the cloud provider faces two key questions – (1) Data Placement: Which physical machines should be used for each dataset? and (2) VM Placement: Where should the VMs be provisioned to process these data blocks? Poor placement of data or VMs may result in poor performance. They presented system architecture for the MapReduce cloud service and describe how existing data and virtual machine placement techniques lead to longer job execution times and large amounts of network traffic in the data center. We identify data locality as the key principle which if exploited can alleviate these problems and develop a unique coupled data and VM placement technique that achieves high data locality. Uniquely, Purlieus's proposed placement techniques optimize for data locality during both map and reduce phases of the job by considering VM placement, MapReduce job characteristics and load on the physical cloud infrastructure at the time of data placement.

Large scale data processing is increasingly common in Cloud Computing systems like Hadoop, MapReduce etc. In these systems, files are split into many small blocks and all blocks are replicated over several servers. To process files efficiently, each job is divided into many tasks and each task is allocated to a server to deal with a file block. Enhancing task data locality (placing tasks on servers that contain their input blocks) is crucial for the job completion time. Although there have been many approaches on improving data locality, most of them either are greedy and ignore global optimization, or suffer from high computation complexity. To address these problems, Vaishali W. Thawari, Sachin D. Babar, Nitin A. Dhawas propose a heuristic task scheduling algorithm in which an initial task allocation will be produced at first, and then the job completion time can be reduced gradually by tuning the initial task allocation.

3. FRAMEWORK

A. Extended JoSS Scheme

Previously, we advise JoSS to correctly schedule MapReduce jobs in a digital MapReduce cluster by using addressing both map-data locality and reduce-data locality from the attitude of a user. In this proposed JoSS, we can do the job classification and it based on the ratio of predefined block size of reduce and Map jab, job classification can be classified into either a Map-Heavy (MH) or Reduce-Heavy (RH) job.

The Hybrid Job-Driven Scheduling Scheme (JoSS) has two variations such as

1. Task-driven Task Assigner (TTA)
2. Job-driven Task Assigner (JTA)

Task-driven Task Assigner

Whenever VPS has an idle Map slot, TTA preferentially assigns a Map task from MQ to VPS based on the Hadoop FIFO algorithm. The aim is to preferentially execute all newly submitted jobs one by one and obtain their filtering percentage values to determine their job classifications. However, if MQ_{FIFO} is empty, TTA assigns one of the first Map tasks from all the other map-task queues of data center in a round-robin fashion such that tasks can be assigned quickly and job starvation can be avoided.

Job-driven Task Assigner

JTA, which in fact is very similar to that of TTA. The only difference is that JTA always uses the Hadoop FIFO algorithm to assign a Map task from each map-task queue so as to further improve the VPS-locality.

We can improve the job performance as well as we can increase the data locality in virtual MapReduce clusters by classifying jobs into Map-Heavy (MH) & Reduce-Heavy (RH) jobs as well as designing the corresponding rules to agenda each glory of jobs in JoSS. Furthermore, with the aid of classifying jobs into large and small jobs and scheduling them in a round-robin model, JoSS avoids task starvation and improves activity performance. However, the JoSS scheme is not flexible for load balancing.

In this paper we are implementing enhanced JoSS which means heterogeneous virtual MapReduce clusters into consideration so as to increase the flexibility of JoSS. By this proposed scheme, we can balance the load of the virtual MapReduce clusters. For load balancing, we are generating the number of virtual MapReduce clusters equals to the number of jobs.

B. JoSS Scheduling Policies

Based on Job classification, JoSS used three types of scheduling policies. Those are;

Policy A:

This policy is designed for a small Reduce-Heavy (RH) job.

Policy B:

This policy is designed for a small Map-Heavy (MH) job.

Policy C:

This policy is designed for a large job.

Whenever receiving a MapReduce process from a person, the task scheduler decides the sort of the job after which schedules the process primarily based on one in all policies A, B, and C. The task assigner then decides a way to assign a task to a VPS every time the VPS has an idle slot.

C. Load Balancing

Load balancing is beneficial in spreading the load equally throughout the free nodes while a node is loaded above its threshold degree. Though load balancing isn't so substantial in execution of a MapReduce set of rules, it turns into essential while handling massive documents for processing and while hardware resources use is vital. As a attention, it complements hardware utilization in resource-important situations with a moderate improvement in overall performance. A module become implemented to balance the disk area utilization on a Hadoop Distributed File System cluster when a few data nodes have become full or while new empty nodes joined the cluster. The balancer was started with a threshold value; this parameter is a fragment among 0 and a 100 percent with a default value of 10 percent. This sets the goal for whether the cluster is balanced; the smaller the brink value, the greater balanced a cluster can be. Also, the longer it takes to run the balancer. A cluster is considered balanced if for every records node, the ratio of used area on the node to the total capability of node (referred to as the usage of the node) differs from the ratio of used space at the cluster to the total capability of the cluster (usage of the cluster) by no extra than the threshold value.

4. CONCLUSION

To achieve high data locality as well as better job performance, in this paper, we are enhancing the traditional Hybrid Job-Driven Scheduling Scheme (JoSS) work. In this JoSS, the virtual MapReduce clusters are homogeneous. So, we cannot balance the workloads and we need to improve the JoSS flexibility. For that in this paper, we are implementing heterogeneous virtual MapReduce clusters and through this extension we can improve the flexibility of the JoSS.

REFERENCES

[1] S. Chen and S. Schlosser, “Map-Reduce meets wider varieties ofapplications,” Intel Res., Santa Clara, CA, USA, Tech. Rep. IRPTR-08-05, 2008

[2] B. White, T. Yeh, J. Lin, and L. Davis, “Web-scale computer visionusing mapreduce for multimedia data mining,” in Proc. 10th Int.Workshop Multimedia Data Mining, Jul. 2010, pp. 1–10.

[3] Z. Guo, G. Fox, and M. Zhou, “Investigation of data locality inmapreduce,” in Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud GridComput., May 2012, pp. 419–426.

[4] C. He, Y. Lu, and D. Swanson, "Matchmaking: A new mapreducescheduling technique," in Proc. IEEE 3rd Int. Conf. Cloud Comput.Technol. Sci., Nov. 2011, pp. 40–47.

[5] T. White, Hadoop: The Definitive Guide. Sebastopol, CA, USA:O'Reilly Media, Jun. 5, 2009.

[6] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker,and I. Stoica, "Delay scheduling: A simple technique for achievinglocality and fairness in cluster scheduling," in Proc. 5th Eur. Conf.Comput. Syst., Apr. 2010, pp. 265–278.

[7] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong, "BAR: An efficientdata locality driven task scheduling algorithm for cloudcomputing," in Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud GridComput., May 2011, pp. 295–304.

[8] M. Ehsan, and R. Sion, "LiPS: A cost-efficient data and taskco-scheduler for MapReduce," in Proc. IEEE 27th Int. Symp.Parallel Distrib. Process. Workshops PhD Forum, May 2013,pp. 2230–2233.

[9] B. Palanisamy, A. Singh, L. Liu, and B. Jain, "Purlieus: Localityaware resource allocation for MapReduce in a cloud," in Proc. Int.Conf. High Perform. Comput., Netw., Storage Anal., Nov. 2011, pp. 58.

[10] J. Park, D. Lee, B. Kim, J. Huh, and S. Maeng, "Locality-awaredynamic VM reconfiguration on MapReduce clouds," in Proc. 21stInt. Symp. High-Perform. Parallel Distrib. Comput., Jun. 2012,pp. 27–36.