Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

K-MEANS METHOD FOR GROUPING IN HYBRID MAPREDUCE CLUSTERS

[1]YANG YANG, XIANG LONG, BO JIANG†, YU LIU,
[1]School of Computer Science and Engineering, Beihang University, Haidian 100191, Beijing, China
†:Corresponding author

ABSTRACT

In hybrid cloud computing era, hybrid clusters which are consisted of virtual machines and physical machines become more and more Popular? . MapReduce is a good weapon in this big data era where social computing and multimedia computing are emerging. One of the biggest challenges in hybrid mapreduce cluster is I/O bottleneck which would be aggravated under big data computing. In this paper, we take data locality into consideration and group slave nodes with low intra-communication and high intra-communication. After introducing the architecture and implementation of our grouped hybrid mapreduce cluster (GHMC), we give our k-means algorithm in GHMC and evaluate it with reality environments. The results show that there is a nearly 34.9% performance improvement in our system achieved by the K-means algorithm. Moreover, GHMC system also shows good scalability.

Keywords: *MapReduce, Hybrid Cloud Computing, K-means Cluster*

1. INTRODUCTION

MapReduce program model is emerging technology resurgence in 2006[1]. With the development of social network and multimedia, the size of unstructured data and semi-structured data increased at a fantastic speed. There are remarkable features in MapReduce, such as simplicity, fault tolerance and scalability. It is by far the most successful realization of data intensive cloud computing platform[2]. Since Amazon EC2 has spanned up Hadoop framework on Amazon EC2 instances supplying Elastic MapReduce services [3], MapReduce program model tightly combines with cloud computing and virtualization technology.

In cloud computing era, virtualization technology[5] becomes a foundation technology with the features of easy-deployed, highly-utilized and well-isolated[6]. In Gartner' report[15] he trends of hybrid cloud computing would keep in mainstream in next 5 to 10 years. It means more and more applications may be deployed in hybrid cluster consisting of virtual machine(VM) and physical machine(PM). In hybrid mapreduce cluster, virtual machines, which "slice" a single physical machine to multiple relatively separated VMs. These VMs are all assigned with their own resources, co-exist with physical machines.

The performance of a large scale of hybrid mapreduce clusters may degrade because of I/O bottleneck[7]. The integration of mapreduce and hybrid cloud computing has bad performance of I/O. The reasons are as follows:

- MapReduce applications have clear bounds among the processing stages. In I/O-intensive stages, large numbers of I/O requests are sent and data transfer performance degrades sharply.
- The architecture of MapReduce is master/slave which limits the I/O performance on the space. All of the slaves communicate with the master. It means that the more slaves are, the more packets should be dealt with. The performance of the master node reduces when it is overload[14].

It was pointed out[1] that locality is an important issue affecting performance in a shared clusters environment, due to limited network bisection bandwidth. The current research [2][4][8][11][12][14] on reducing I/O communication costs both take use of data locality. Grouped Virtual Mapreduce Cluster(GVMC)[14] group slave nodes and select local-masters which manage the nodes in every group. However, all of the researches are based on homogeneous mapreduce clusters.

In a hybrid mapreduce cluster, we take use of the way which groups low intra-communication nodes together and selects local master. We name this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

hybrid mapreduce cluster as grouped hybrid mapreduce cluster(GHMC). GVMC is one part of GHMC. Using the minimum-weight spanning tree method to construct GVMC minimum communication tree is not suitable for our GHMC because of the more complex situations.

To take fully use of data locality and alleviate the I/O bottleneck in hybrid mapreduce cluster, we make use of clustering method to group the slave nodes with low intra-communication and high inter-communication. Considering the characteristic of hybrid mapreduce cluster, the advantage of clustering method is that the scale of the problem is relatively small and has low-dimension.

The rest of the paper is organized as follows. Section II is related work. Section III gives a introduction of grouped hybrid mapreduce cluster(GHMC) overview and formalize the question. In section IV, we propose our K-means Method Algorithm for Clustering. We verify our algorithm in real GHMC in Section V. Section VI discusses our evaluation results and some related issues.

2. REALTED WORK

2.1 MapReduce Overview

MapReduce is a parallel program model. The input data stores in <key, value> which is split into map, shuffle and reduce consecutive phase. The most popular open-source implementation of MapReduce is Hadoop which is designed to scale up from single server to thousands of machines, each offering local computation and storage. Hadoop follows the architecture of Google MapReduce[1]. The basic MapReduce framework consists of HDFS and MapReduce. HDFS is a distributed file system supplying high-throughput access to application data, which makes of Namenode and Datanodes. Jobtracker has responsibilities for MapReduce task scheduling and tasktrackers management. This typical master/slaves architecture aggravates I/O bottleneck.

There are two types of communications in MapReduce working flow:

- Master-to-slaves: at the beginning of the process, the master node assigns tasks to slaves and pings slaves to get the slaves status periodically during the working process.
- Slave-to-master: slaves send their locations and intermediate files to the master

In order to reduce the communication costs and alleviate I/O performance degradation, data locality is fully used. Studies[2][8] make Hadoop's reduce task scheduler aware of partitions' network locations and sizes to reduce the communication costs and execution time. BAR[9] allocates tasks dynamically according to network state and clusters workload. Studies[4][11][12] take VMs data locality to improve the performance. Study[4] has built a model that defines metrics to analyze the data allocation problem. Study[12] proposes a MapReduce framework on virtual machine which takes full advantage of data locality, virtual machine live migration and checkpoint. Study[11] explores the effect of I/O scheduling on performance of MapReduce running on VMs.

All of these works take use of data locality and reduce the communication costs in either pure physical mapreduce cluster or pure virtual mapreduce cluster. These researches on their mapreduce cluster are based on the assumption that mapreduce cluster are in homogeneous environments. It's obviously that mapreduce clusters are in heterogeneous environments when they deploy physical and virtual hybrid machines. Making use of the idea of GVMC, we group the slave nodes with low intra-communication costs and high inter-communication costs.

2.2 Clustering Algorithm Overview

Cluster is the process to collect similar data objects to one another within the same cluster and dissimilar to the objects in other clusters. The popular methods for clustering are classified by partitioning methods, hierarchical methods, density-based methods and grid-based methods.

K-means algorithm is one the most popular algorithms. It's a partitioning method to construct a partition of a database D of n objects into a set of k cluster[13]. It's efficient algorithm for clustering. The process is implemented in 4 steps: firstly, it partition objects into k subsets; secondly, k seed points as the centroids of the cluster; thirdly, each object is assigned to the cluster with the nearest seed point. The process lasts until no more new assignment happen.

3. DESIGN OF GHMC

3.1 Grouped Hybrid Mapreduce Cluster

At the original mapreduce cluster, every slave connects to master nodes and sends/receives messages to/from it. We take advantages of data locality; group the low intra-communication slave nodes to reduce data transfer costs in mapreduce

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

process. Local-masters are also introduced to manage group members.

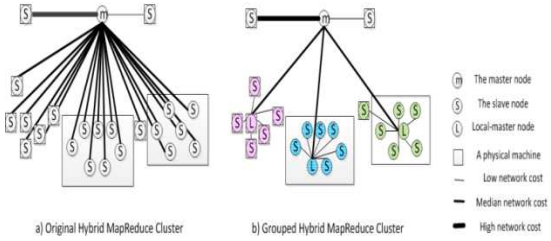

Figure 1. A Case Of Original Mapreduce Cluster

Figure 1 shows a case of GHMC topology. Figure 1.a) shows original hybrid mapreduce cluster. b) shows GHMC. The black line with HH Type describes the high communication costs of the slave node when it connects with both the master and the other slaves. The finer line means the connection of HL Type slave and the master. The HL Type slaves represent the slave nodes with high-communication costs connecting the master and low-communication costs connecting other slaves. The finest line means the lowest communication costs with the master which we call LO type communication. Since HH type nodes are hard to optimal by grouping and LO type nodes have no need to optimal, we take the method of grouping to optimal on HL type slaves communication costs.

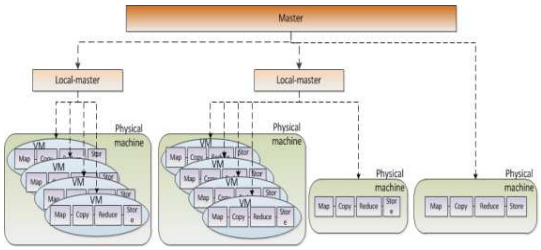

Figure 2. The Architecture of Hybrid Mapreduce

Figure 2 shows the architecture of GHMC. In this structure, the most important implementation is the local-master.

HH type nodes or LO type nodes are directly connected with the master. Others are connected with local-masters which have ability of local-NameNode and local-JobTracker are simplified original master. The local-NameNode is the master of local DataNodes and the local JobTracker is the master of local TaskTrackers. The local-JobTracker only receives the local-TaskTrackers heartbeats, updates and monitors task status. If JobTracker finds out that the local-JobTracker is dead or local-JobTracker sends the group status report which shows all the status of the grouped TaskTrackers are dead, JobTracker would put this group into waiting-for-restart queue until the physical machine has been reboot and re-added into cluster. The local-NameNode stores the information of filename and block locations to take advantages of data locality. The realized protocols are LocalClientProtocol and LocalDatanodeProtocol communication protocols. The LocalClientProtocol defines local namespace operations, including the operations of adding, creating, deleting blocks and getting block locations. The LocalDatanodeProtocol defines the interface between local-NameNode and NameNode.

4. K-MEANS METHOD ALGORITHM

4.1 Algorithm Description

In order to complete the operation of grouping and selecting the local-master, we need to build a Low and Flow Tree(LFTree). The conditions are satisfied that T is a connected , undirected graph $T = (V, E)$ where V is its vertex set which represents nodes and E is its edge set which stands for the connection between nodes. For each edge $(i, j) \in E$, we have weight $w(i, j)$ specifying the cost to connect i and j and put $w(i, j)$ into the matrix D_{ij} .

The input of the algorithm is D_{ij} matrix and the number of groups specified by the administrator k . The output is the LFTree built by our algorithm.

K-MEANS-IN-GROUPEDHMC(D_{ij} , k)

1: init $n = length(D_{ij})$;

2: init *CQueue* which stores the centroid V_x ;

3: init *NCQueue* which stores the non-centroid V_y ;

4: init combination number of $l = C(n, k)$;

5: initial control variety of loop:

$$S_0 = S_1 = S_2 = tmp = 0$$

6: let $HLMatrix = w(i, j)$ which

$$i \in \{x \mid x \in HL\}$$

7: **while** $S_0 < l$

8: let $CQueue^{s0} = combination(length(HLMatrix), k)$;

9: let $V_i \in CQueue^{s0}$;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10: let $V_j \in NCQueue^{s0}$;

11: **while** $S_2 < length\left(CQueue^{s0}\right)$

12: **while** $S_1 < length\left(NCQueue^{s0}\right)$

13: let $V_o \in Group_p^{s0}$ when $D_{po}^{s0} = min(D_{ij})$;

14: let $S_1 = S_1 + 1$;

15: **end**;

16: let $S_2 = S_2 + 1$;

17: **end**;

18: let Weight(T^{s0}) $= \sum_{p=0}^{k-1}\sum_{o=k}^{n-1} D_{po}^{s0} + \sum_{i=0}^{k-1} D_{ij}^{s0}$;

19: record the β th minWeight(T^{s0});

20: record the group $situationGroup\_\beta_\alpha^{s0}$ where $a \in CQueue^{s0}$;

21: let $S_0 = S_0 + 1$;

22: **end**;

23: Choose suitable centroid based on CPU cap;

24: Restore LFTREE;

We firstly initialize the temporary variables. Lines 1-5 show this procedure. Then, choose the HL Type nodes by scanning MNC matrix and build *HLMatrix* consisted by edges' weight connecting HL type nodes. It shows in Line 6. Lines 7 - 22 are designed for improving k-means clustering algorithm to find out the clusters and centroids which are groups and local-master in our GHMC environment. Lines 23-24 find out the relatively optimal group method for the cluster. Line 8 is the combination function which produces l initial centroid way. Lines 7-22 have triple loop. The outermost loop is the l different initial centroid. The inner loops are the processes which group the non-centroid nodes into the minimum distance centroid group. Lines 18-20 are choose the β th minimum sums of weight in certain situation. Then, takes use of group function to complete the process of group operation and produce LFTree.

5. EVALUATION

5.1 Experiments environment

The hardware configuration of our test systems are physical machines with Intel Q9400 quad-core CPU, DDRII-800 2GB memory. The software of Hadoop 0.20.2 cluster configurations are 2000MB Hadoop heapsize, 3 replications of files, maximum of map or reduce number on the same slot is 2 and speculative execution is on. The virtual node configurations are 256M memory, 1 vcpu, unpinned.

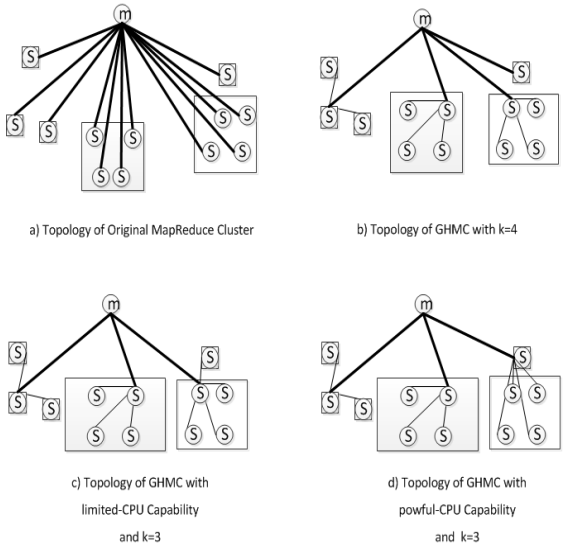

Figure 3. The Experiments Environments of Group Hybrid Mapreduce Cluster

We use hybrid cluster to verify our GHMC performance. The cluster is made of 12 slave nodes and 1 master as Figure 3 shows. 8 slave nodes are put on 8 virtual machines which allocate on 2 physical machines equally. We adjust the network bandwidth and simulate the different network condition in reality situation. The network bandwidth which is set 5MB/s in thick black lines and 200 MB/s thin lines in Figure 3. The network bandwidth of disconnected nodes is set to 10 KB/s. Figure 3 shows the results after running our GHMC algorithm and reconstructing the cluster. We also run Wordcount, Grep and BBP to measure the performance.

5.2 Experiments and Discussion

Figure 4 shows the experiments results under the four environments as Figure 3 shows. To show more visually appealing way, we take the average value on 50 times execution time of each applications and normalize the last results. The results tell that the improvement of our grouped virtual mapreduce cluster is up to ~34.9%. It's obviously that set the powerful CPU machines as

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

local-masters have shorter execution time than that

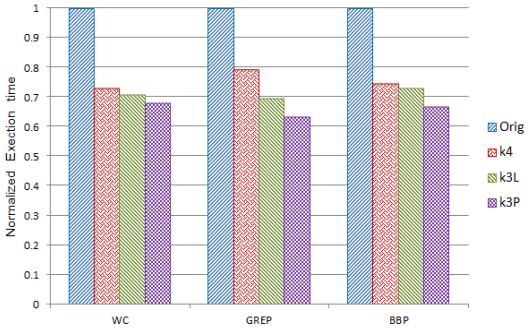

Figure 4. The Normalized Execution Time on four structures

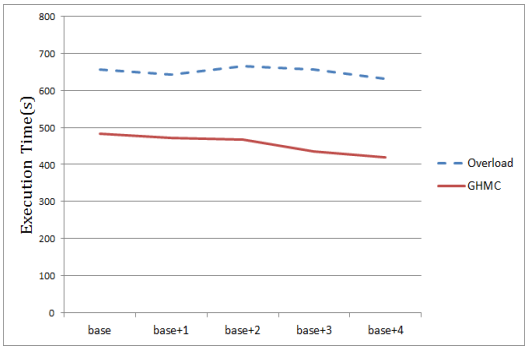

Figure 5. Scalability comparison on Overload original

of selection with weaker local-master. The parameter of k value also effects the execution time, it needs more study on further research.

In GHMC scalability tests, we add four physical machines and use GHMC algorithm to compute the topology each time. Figure 5 shows the results. Comparing the instability of overloaded original mapreduce cluster, the execution time of GHMC is shorter when the nodes number increased. It says that our GHMC algorithm has good scalability.

6. CONCLUSTION AND FURTHER WORK

In this work, we propose a novel method to group the hybrid mapreduce cluster which improves the performance by fully taking advantages of data locality. We use K-means method as the clustering algorithm in GHMC. The experimental results show that our GHMC which uses K-means algorithm has better performance than original hybrid mapreduce cluster and shows good scalability as well.

ACKNOWLEDGEMENTS

This work is supported by grants of the National Natural Science Foundation of China (project no. 61202077) and the Fundamental Research Fund for Central Universities in China (project no.YWF-12-LXGY-008).

REFERENCES:

[1] Jeffrey Dean and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters". *Commun.* ACM 51, 1, 2008, 107-113.

[2] Shadi Ibrahim, Hai Jin, Lu Lu, Song Wu, Bingsheng He, and Li Qi, "LEEN: Locality/Fairness-Aware Key Partitioning for MapReduce in the Cloud," Cloud Computing Technology and Science (CloudCom), *2010 IEEE Second International Conference on* , vol., no., pp.17-24, Nov. 30 2010-Dec. 3 2010

[3] http://aws.amazon.com/elasticmapreduce/

[4] Yifeng Geng, Shimin Chen, YongWei Wu, Wu, R., Guangwen Yang, Weimin Zheng, "Location-Aware MapReduce in Virtual Cloud," Parallel Processing (ICPP), 2011 International Conference on , 13-16 Sept. 2011, pp.275-284,

[5] Cong Xu, Sahan Gamage, Pawan N. Rao, Ardalan Kangarlou, Ramana Rao Kompella, and Dongyan Xu. "vSlicer: latency-aware virtual machine scheduling via differentiated-frequency CPU slicing." In Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing (HPDC '12). ACM, New York, NY, USA, 3-14.

[6] Jin Hai, "From Grid Computing to Cloud Computing: Experiences on Virtualization Technology." Future Generation Information Technology: Second International Conference, Lecture Notes in Computer Science, 2010, Volume 6485/2010, 41.

[7] Yanyan Hu, Xiang Long, Jiong Zhang, Jun He, and Li Xia, "I/O scheduling model of virtual machine based on multi-core dynamic partitioning." In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 142-154.

[8] Mohammad Hammoud, M. Suhail Rehman, Majd F. Sakr，"Center-of-Gravity Reduce Task Scheduling to Lower MapReduce Network Traffic." 2012 IEEE Fifth International Conference on Cloud Computing, pp.49-58.

[9] Jiahui Jin, Junzhou Luo, Aibo Song, Fang Dong, Runqun Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing," Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on , vol., no., pp.295-304, 23-26 May 2011.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi, "Evaluating MapReduce on Virtual Machines: The Hadoop Case", Proc. Conf. Cloud Computing (CloudCom 2009), Springer LNCS, Dec 2009, pp.519-528.

[11] Jun Fang, Shoubao Yang, Wenyu Zhou, Hu Song, "Evaluating I/O Scheduler in Virtual Machines for Mapreduce application," Grid and Cooperative Computing (GCC), 2010 9th International Conference on , vol., no., pp.64-69, 1-5 Nov. 2010.

[12] Shadi Ibrahim, Hai Jin, Bin Cheng, Haijun Cao, Song Wu, and Li Qi. "CLOUDLET: towards mapreduce implementation on virtual machines." In Proceedings of the 18th ACM international symposium on High performance distributed computing (HPDC '09). ACM, New York, NY, USA, 65-66.

[13] MacQueen, J. B, "Some Methods for classification and Analysis of Multivariate Observations," 1. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201. Retrieved 2009-04-07.

[14] Yang Yang, Xiang Long, Bo Jiang. "An Efficient Grouped Virtual Mapreduce Cluster" . the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013)

[15] Drue Reeves, Richard Watson, Kyle Hilgendorf, Dan Blum, "2012 Cloud Computing Planning Guide: From Hybrid IT to Hybrid Clouds." http://www.gartner.com/DisplayDocument?id=1837017