TELKOMNIKA, Vol. 11, No. 8, August 2013, pp. 4587~4593
e-ISSN: 2087-278X

A Privacy Data-oriented Hierarchical MapReduce Programming Model

Haiwen Han*[1,2], Weiping Zheng[3]
[1]College of economic and management, South China Normal University, China
[2]Research Institute of Computer Systems, South China University of Technology, China
[3]College of computer science and Engineer, South China Normal University, China
*Corresponding author, e-mail: hanhw@scnu.edu.cn, iamzhengweiping@126.com

Abstract

To realize privacy data protection efficiently in hybrid cloud service, a hierarchical control architecture based multi-cluster MapReduce programming model (the Hierarchical MapReduce Model, HMR) is presented. Under this hierarchical control architecture, data isolation and placement among private cloud and public clouds according to the data privacy characteristic is implemented by the control center in private cloud. And then, to perform the corresponding distributed parallel computation correctly under the multi-clusters mode that is different to the conventional single-cluster mode, the Map-Reduce-GlobalReduce three stage scheduling process is designed. Limiting the computation about privacy data in private cloud while outsourcing the computation about non-privacy data to public clouds as much as possible, HMR reaches the performance of both security and low cost.

Keywords: *hierarchical architecture, map reduce programming, hybrid cloud, privacy data*

1. Introduction

Even though the cost-effective shared computing resources in public cloud of cloud computing technology provides pretty good business solution to meet the growing social demand for massive data processing, it still leads inevitably to security risk [1-3]. Consequently the private cloud which characterizing in spending expensive self-build construction cost but owning the total control become the only choice of many application with emphasis of privacy data protection. Grew hot in recent years, hybrid cloud technology [4] provides possibility to distribute computation among the public and private clouds to achieve both absolutely limiting the privacy data computation in private cloud and fully outsourcing the non-privacy data computation to public cloud. Currently, although the public cloud computation resource could be used to support the private cloud computation scale out on-demand [4], the effective technology for privacy data oriented computation splitting and distributed executing has still not been well developed. Recently, Kehuan Zhang [8] designed a Sedic model based on MapReduce [5], the most widely used cloud computing programming model, to isolate the privacy data in private cloud by the data distinguish mechanism on different data characteristic (privacy and non-privacy) and the task scheduling mechanism on different computation resource(public cloud and private cloud). But the detail modification directly on the original execution framework would reduce the execution efficiency of the conventional computation tasks, while the directly scheduler of public cloud resource without using control node would bring the risk of chaotic management. The deeply reason is that Sedic keep the original MapReduce working architecture of single cluster, which is neither compatible to the hybrid cloud naturally composed of multiple clusters nor good at data and corresponding computation isolation. About the improvement of MapReduce's architecture, Jin Jing [6] developed a control architecture of multi-control nodes instead of the original single node, while Yuan Luo [7] proposed a hierarchical MapReduce architecture to implement the flexible scheduling for multi clusters resource. All these researched are benefit to the development of MapReduce's architecture for designing distinguish mechanism and achieving the privacy data protection.

Summarizing the previous research and focusing on the hybrid cloud's multi-clusters characteristic, a hierarchical control architecture MapReduce programming model (Hierarchical MapReduce Model, HMR) is presented in this paper. The major development based on the

Received January 15, 2013; Revised May 17, 2013; Accepted May 27, 2013

original MapReduce programming model is composed of the centralized hierarchical architecture to manage multi clusters' resource, the data processing mechanism to achieve the privacy data oriented automatic isolation of data and computation, and the MapReduce-GlobalReduce three stages scheduling process to correctly carry out the parallel distributed computation tasks in multi-cluster environment. The detail implementation is based on Hadoop MapReduce system [9].

2. The Hierarchical Architecture
2.1. Architecture

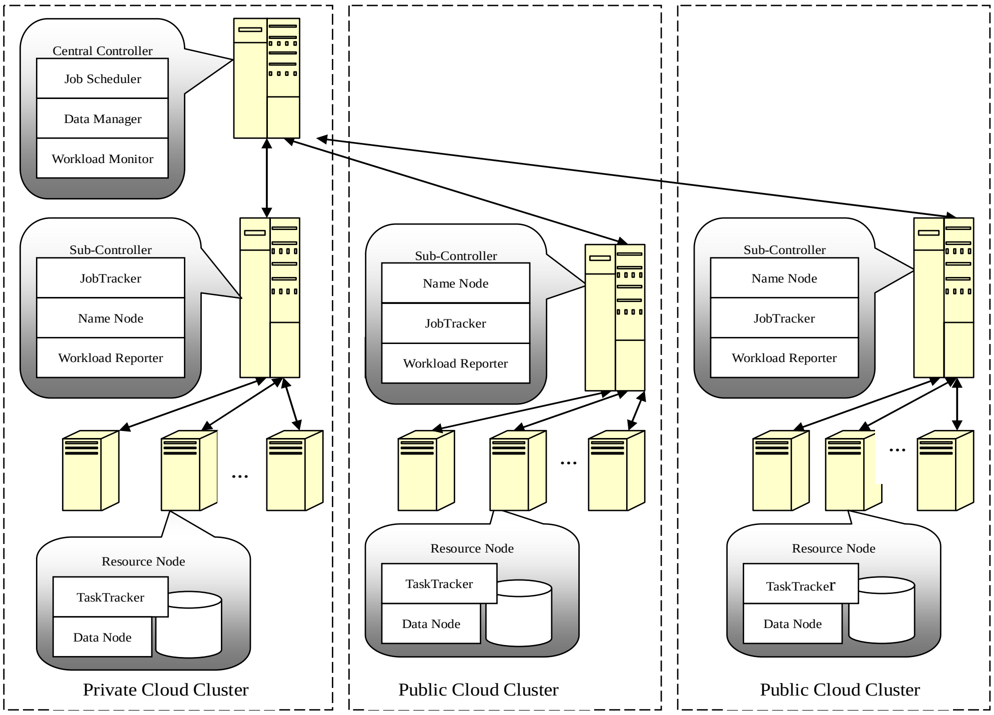

Figure 1. The HMR Architecture

As showed in Figure 1, HMR's hierarchical architecture is composed of three layers of central controller, sub-controller and resource node. Ranging about one private cloud and multi public clouds, the only one central controller of HMR's hierarchical architecture is located in private cloud to lead the whole execution, while every cluster's only one sub-controller is leaded by central controller to manage every cluster's resource accordingly. In HMR system, the conventional single control node in single cluster is extended to two-layer control architecture to unified manage the resource from different source (private and public clouds) through the central controller and effectively implement the isolation of data and computation under the multi-clusters mode.

2.2. The Components

As mentioned above, in HMR system, there are three types of nodes. The first one, central control node, is a brand new type to the original MapReduce model. Being located in private cloud, central control node is composed of job scheduler which, data manager and workload monitor. The job scheduler manage the whole process including user's submission acception, computation tasks assign to sub-control nodes, and outcome data return to user. The data manager would be invoked by the job scheduler to implement the data isolation according to privacy characteristic. The workload monitor is major in monitoring and recoding the public cloud's current workload which would be used by job scheduler to do the sub-control node selection.

The second one, sub-control node, is composed of namenode, jobtracker and workload reporter. As sub-control node is almost the same as the control node in Hadoop, the workload reporter is the new component which reporting the cluster's current workload to central control node in fixed time. As working in conventional way as original MapReduce, the namenode and the job tracker also would work in a new data privacy oriented way if needed. Besides, namenode must manage the data transfer process between clusters and accept job from central control node instead of from user in original Hadoop cluster.

The third one, resource node, is composed of datanode and tasktracker. As working in conventional way as original MapReduce, the datanode and tasktracker also would work in a new data privacy oriented way if needed.

3. Programming Model

To achieve the privacy data oriented isolation on data and computation, HMR develops the original MapReduce in these two ways: 1) the privacy data oriented data process mechanism to isolate the data in two categories of privacy and non-privacy under the hierarchical architecture; 2) the Map-Reduce-GlobleReduce Three Stage Scheduling process to carry out the privacy data oriented parallel distributed computation under the hierarchical architecture.

3.1. Privacy Data Oriented Data Process Mechanism

The privacy data-oriented data process mechanism for HMR system is about user data submission and data placement.

3.1.1. User Data Submission

Without loss of generality, it is assumed that the user data submitted in file. And besides submitting application job by Job Client in Hadoop’s user client, users should also submits their data in three types: 1) Public, which means that this file is composed without privacy data at all and could be passed to public cloud entirely. 2) Sensitive, which means that this file is composed of privacy data and must be passed to private cloud entirely. 3) Semipub, which means that the file is mixture of privacy data and public data, should be firstly split off according to the specific privacy data strings provided by users, such as some specific ID card number, customer name, social security number and credit card number, and so on. These specific privacy data strings would be submitted together.

3.1.2. Data Process and Placement

The job scheduler receives the user submission including computing job and user data firstly, and then passes it to cloud or clouds according to the three data submission types as follow.

1) Sensitive

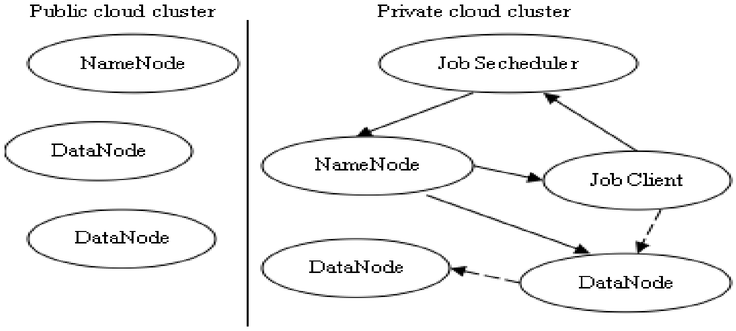

Figure 2 Mechanism for sensitive data
→Control Stream
--->Data Stream

If the submission type is sensitive, the user submission including computing job and user data would be entirely passed to private cloud cluster's sub-control node by Job Scheduler. NameNode in this node receives the submission and do the data placement with the same approach as original MR system. Specifically, as is shown in Figure 2, NameNode creates meta data as INode structure for this file which records all the information about this file, such as filename, the latest modified time, and so on. After saving this INode in sub-control node, NameNode finds out a free storage resource node in cluster to save this file by local DataNode in fixed size (64M) data block. Datanode would find out next free storage resource node to save the next datablock if needed. In addition, HDFS distributed file system would save three replica for each data block in three nodes respectively (according to the default process in hadoop system).

2) Public

If the submission type is public, the user submission including computing job and user data would be entirely passed to sub-control node(s) in (a) public cloud cluster(s) according to some scheduling strategies, such as the cluster's current workload or the bandwith between clusters. NameNode in this node. NameNode in this node receives the submission and do the data placement with the same approach as before. The only difference is that NameNode in public cloud cluster need to transfer data form private cloud, which means the data transfer between clouds, as is shown in Figure 3.

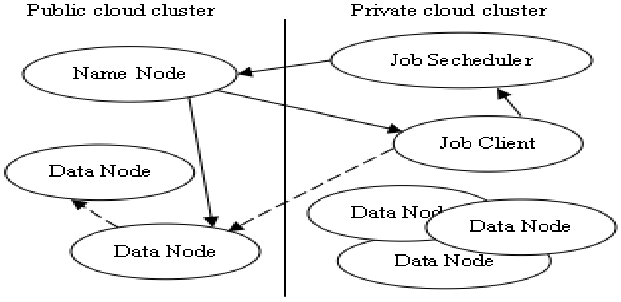

Figure 3 Mechanism for Public data
→Control Stream
--->Data Stream

3) Semipub

If the submission type is semipublic, Data Manager would be executed to get and process data in these steps.

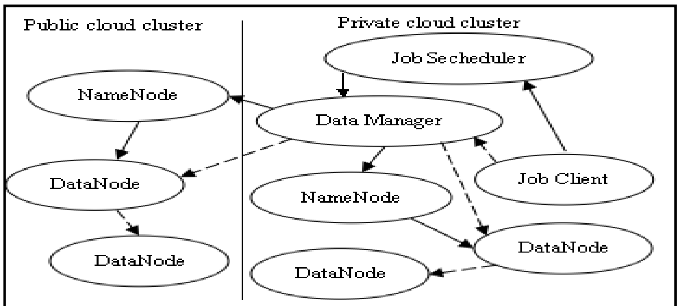

Figure 4 Mechanism for semipub data
→Control stream
--->Data stream

Firstly, a privacy label table is founded by Data Manager. Scanning the whole file according to the specific privacy data strings provided by users, Data Manager creates a new privacy tuple (filename, offset, length) for every matched string. As the location information about all privacy data strings being recorded in this table, a sanitized replica of this file is created by zeroing out all the privacy data strings in original file.

Secondly, Data Manager passes the submission including computation job, original file and the privacy label table to private cloud cluster's sub-control node, and NameNode in which would do the data placement in private cloud cluster. As is shown in Figure 3, the procedure of data placement is the same as previous except: 1) Instead of from Job Client, DataNode gets data from Data Manager. 2) NameNode saves not only the INode structure but also the privacy label table in sub-control node.

Finally，Data Manager passes the submission including computation job, sanitized replica and the privacy label table to public cloud cluster's sub-control node, and NameNode in which would do the data placement in public cloud cluster. As is shown in Figure 4, the procedure of data placement is the same as previous except: 1) Instead of from Job Client, DataNode gets data from Data Manager. 2) NameNode saves not only the INode structure but also the privacy label table in sub-control node. 3) DataNode saves instead of the original file, but the sanitized replica.

3.2. The Map-Reduce-GlobleReduce Three Stage Scheduling Process

The conventional Map-Reduce two stages computation task scheduler process in original MapReduce must be changed because of the characteristics of multi-cluster and privacy data in HMR system. The characteristic of multi-cluster means that the outcome data from multi map fuctions would be distributed among multi clusters instead of single cluster in original MapReduce. The characteristic of privacy data means that the computation tasks on privacy data must be executed in private cloud. The directly solution, which is to transfer all the inter media outcome data back to private cloud cluster, owns the deficiencies of great data transfer between clusters and great computation in private cloud. To utilize the cloud resource effectively, a Map-Reduce-GlobalReduce three stage computation tasks scheduling process is presented in this paper. While the Map stage of scheduling process is the same as in original MapReduce, The Reduce stage is being changed to the Reduce-GlobalReduce two stages, in which the Reduce stage is to implement the local reduce in each cluster and transfer the outcome data back to private cloud while the GlobalReduce is to being performed in private cloud to finished the reduce operation on those data. The specifically work is showed in Figure 5.

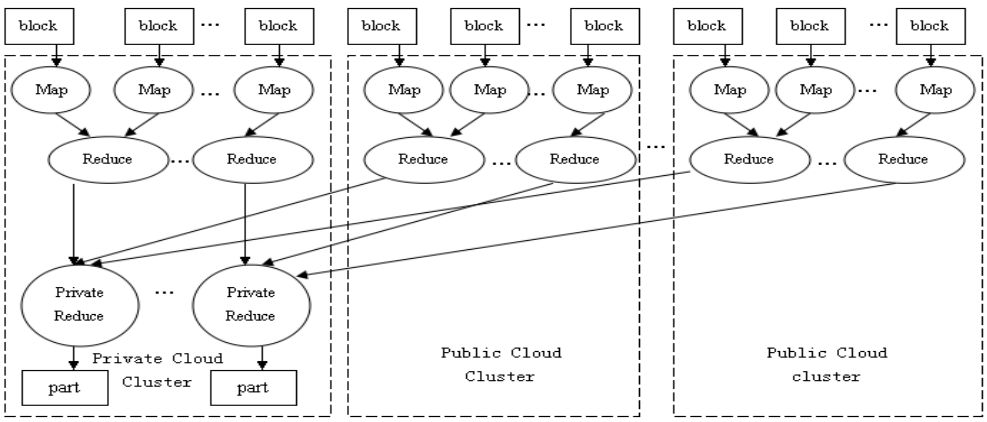

Figure 5. The Map-Reduce-GlobalReduce Three Stage Scheduling Process

3.2.1. Map stage: $(k^i, v^i) \rightarrow (k^m, v^m)$ List

In Map stage, distributed map functions are performed on distributed data blocks in parallel in every cluster. After a job is submitted into HMR system, the input data set is partitioned into blocks. JobTracker first creates TaskInProgress object according to map function for each block in resource node and assembles these objects into a JobInProgress queue. Whenever a "heartbeat" signal comes, indicating that a TaskTracker of a computing resource node is ready to run a new task, JobTracker looks up the queue and assigns the most appropriate task for the node, the one whose data block is stored on the node, for example.

For the data block composed of uniform data that totally public or private, TaskTracker reads a record from the block through a RecordReader object and performs the process defined in map function. For the data block mixture of public data and private data, there are original block in private cloud and sanitized replica in public cloud but with privacy label table both. So, TaskTracker reads the privacy data only according to the privacy label table and performs the process defined in map function in private cloud cluster, while reads the public data only according to the privacy label table in public cloud clusters.

Each map task consumes a data block, no matter uniform or mixture, as an input key/value pair named (k^i, v^i) and produces a set of intermediate key/value pairs named (k^m, v^m). The (k^m, v^m) list is stored as the output data of map functions in local node.

3.2.2. Reduce Stage: $(k^m, [v_1^m, \cdots, v_n^m]) \rightarrow (k^m, v^r)$

In Reduce stage, reduce functions are performed on distributed data output from map functions in every cluster. Similar to the scheduling process for map functions, JobTracker creates TaskInProgress object according to reduce function and place it into JobInProgress queue. When a “heartbeat” signal comes and the previous map operation being completed, this object would be assigned to a TaskTracker of a computing resource node. Each reduce task consumes the intermediate (k^m, v^m) list might needed to be dragged from other resource nodes. Taking in same key like k^m, a set of corresponding values like $(k^m, [v_1^m, \cdots, v_n^m])$ would be classified and values of $[v_1^m, \cdots, v_n^m]$ would be merged according to the specific reduce operation. When set of key/value pairs as (k^m, v^r) list for different k^m are produced, NameNode in sub-control node’s would send a submission to central control node’s JobTracker for transferring these output data back to private cloud. If this submission comes from public cloud cluster, it will be passed to sub-control node to locate a free resource node in private cloud cluster and schedule this node’s DataNode to finish this transfer and store these data. Al the local Reduce results would be sent back to private cloud cluster to perform Private Reduce task.

3.2.3. GlobalReduce Stage: $(k^m, [v_1^r, \cdots, v_n^r]) \rightarrow (k^m, v^o)$

In PrivateReduce stage, the final reduce operation, being the same as reduce function in Reduce stage, are performed on data output from reduce functions executed in Reduce stage. After JobTracker receives all the submissions from all clusters, it passes the submission about reduce to sub-control node. NameNode in this node creates TaskInProgress object according to reduce function and place it into JobInProgress queue. When a “heartbeat” signal comes, this object would be assigned to a TaskTracker of a computing resource node. Each reduce task consumes the intermediate (k^m, v^r) list might needed to be dragged from other resource node. Taking in same key like k^m, a set of corresponding values like $(k^m, [v_1^r, \cdots, v_n^r])$ would be classified and values of $[v_1^r, \cdots, v_n^r]$ would be merged according to the specific reduce operation. When the set of key/value pairs as (k^m, v^o) list for different k^m are produced, this final output data would be submitted to user by central control node’s JobTracker.

If all the data are distributed in only one cluster, such as the data submission type is private and all the data are limited in private cloud cluster, the three-stage scheduling process becomes the same as two-stage scheduling process.

The three-stage Map-Rduce-GlobalReduce scheduling process brings about these advantages to HMR: 1) Achieving the target of privacy data protection. This scheduling process provides the assurance by isolating storage of privacy data and corresponding intermediate data in private cloud cluster and scheduling only the private cloud computing resource to perform the corresponding computation include map tasks and reduce tasks. 2) Minimizing the communication overhead between clouds. Huge computing amount of reduce operation occurs in Reduce stage in public cloud cluster decreases the output data that is needed to transferred from public cloud to private cloud. 3) Maximizing the outsourced computation. In both the Map stage and Reduce stage, all the computation tasks about non-privacy data are executed only using public cloud resource.

4. Conclusion

A privacy data-oriented hierarchical MapReduce programming model, HMR, is presented in this paper. HMR provides not only the solution for the privacy data-oriented application but also the scalability for scale out the application in large scale hybrid cloud. In addition, the scenarios composed of one private cloud cluster and multi public cloud clusters is coincided with the requirement of actual applications. In summarize, HMR has great application prospects.

References

[1] Walker E. Benchmarking Amazon EC2 for high-performance scientific computing. *Usenix Login*. 2008; 33(5): 18-23.
[2] Su H. Distribution Grid Reactive Power Optimization Based on Improved Cloud Particle Swarm Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 11(1): 468-475
[3] S Jha, L Kruger, V Shmatikov. *Towards practical privacy for genomic computation.* In 2008 IEEE Symposium on Security and Privacy. Madison. 2008: 216-230.
[4] Xue SJ, Wu W. Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm. T*ELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(7): 1560-1566.
[5] J Dean, S Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*. 2008; 51(1): 107–113.
[6] Jin Jing, Wang Yan, LiXin, et al. A New Multi-Master Framework of MapReduce. *Journal of BeijingUniversity of Posts and Telecommunications.* 2012; 35(4): 89-93.
[7] Yuan Luo, Beth Plale. *Hierarchical MapReduce Programming Model and Scheduling Alorithms.* Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster. Washington. 2012: 769-774.
[8] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, et al. *Sedic: Privacy-Aware Data Intensive Computing on Hybrid Clouds.* Proceedings of the 18th ACM conference on Computer and communications security. Chicago. 2011: 515-526.
[9] Taylor RC. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC bioinformatics.* 2010; 11(Suppl 12): S1.
[10] TWhite T. Hadoop: The definitive guide. Sebastopol: O'Reilly Media Inc. 2012: 203-243.