ISSN 2348 – 9928
Doi:01.0401/ijaict.2014.07.27 Published on 05 (4) 2017

COST MINIMIZATION OF RUNNING MAPREDUCE ACROSS GEOGRAPHICALLY DISTRIBUTED DATA CENTERS

Ms. T. Cowsalya
PG Scholar,
SVS College of Engineering,
Coimbatore, Tamilnadu, India

Dr. S. Senthamarai Kannan
Assistant Professor (SG)
SVS College of Engineering,
Coimbatore, Tamilnadu, India

Abstract— ***Information is increasingly important in our daily lives. We need information when and where it is required. We access the internet everyday to perform searches, to participate in social network, send and receive email, share pictures and videos and other applications. Therefore cost minimization for processing data has become an important issue in big data era. Here considered three factors like Data assignment, data placement and data movement which manipulate the operational expenses of data centers. In many scenarios, data are, however, geographically distributed across multiple data centers, and easily moving all data to a single data center before processing it can be expensive. We evaluate possible ways of executing such jobs, and propose Two Dimensional Markov Chain Models that can be used to resolve schedules for job sequences which are optimized either with respect to execution time or financial cost.***

Keywords— Cost Minimization, Mapreduce, Data Centers.

I. INTRODUCTION

BIG data analysis is one of the key challenges of current era. The restrictions to what able to be done are often times due to how much amount of data can be processed in a given period of time. Big data sets innately occur due to applications generating more information to get better operation, performance; general applications like social networks supports every individual users in producing massive amounts of data.

MapReduce is a programming model and an connected implementation for processing and producing massive data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is collected of a Map () method that performs filtering and sorting and a Reduce () method that performs a summary operation. The MapReduce System arrange the processing by running the various tasks in parallel, administrating all data transfers between the variety of parts of the system. It provides redundancy and fault tolerance. Hadoop is an important tool used in Big Data with its two important components as MapReduce and Hadoop Distributed File System. MapReduce is used to process the data whereas Hadoop Distributed File System is used to store the data nodes in the cluster. MapReduce is used to process the data whereas Hadoop Distributed File System is used to store the data nodes in the cluster.

A. *1.1 Geo-Distributed data center*

1. Data centers that are spread at different geographic regions, e.g., Google's 13 data centers over 8 countries in 4 continents[1].the main objective of this paper is toto place these data chunks in the server.to distribute tasks onto servers.
2. to move data between data centres.
3.

In step 1 Volley system [2] is used that chooses the datacenter to place the data chunks into the server. Step 2 chooses the optimized server to process the data that is stored in the data center. In step 3 two dimensional Markov Chain model is used i.e. a data chunk is processed in the first data center and the output is passed as an input to the next data center. Likewise all the data chunks are processed in different geographically distributed data center.

Many hard works have been made to minimize the computation or data movement cost of data centers. Data center resizing (DCR) has been projected to minimize the processing cost by adjusting the number of activated servers through task placement [3].Although there are some advantage, it is far from achieving the cost efficient big data processing because of the following weaknesses like data locality may result in waste of resources.

1. The link in network may vary on transmission rate and costs.

To overcome above problems, we study the cost minimization problem of big data computation through joint optimization of data assignment, data placement, and data movement in geographically distributed data centers. Servers are equipped with limited storage and computation resources. Our objective is to optimize the big data placement, task assignment, routing

© 2017 IJAICT **(www.ijaict.com)**

ISSN 2348 – 9928
Doi:01.0401/ijaict.2014.07.27 Published on 05 (4) 2017

and DCR such that the overall computation and communication cost is minimized.

II. RELATED WORK

2.1 Minimizing Server cost

Numerous of large-scale data centers are deployed providing services to large users. As suggested in [11], a data center may contain big servers and guzzle high power. Millions of dollars cost on electricity have cause a serious trouble on the operating cost to data center providers. Therefore, minimizing the electricity cost has established major attention from both academia and industry [4], [5]–[7]. Data Center Resizing and data placement are generally jointly measured to match the processing requirement. [8] Suggest the best workload control by taking account of latency, energy expenditure and electricity cost.

2.2 Managing Big Data

To undertake the challenges of successfully managing big data, many decisions have been proposed to recover the storage and processing cost. The advantage in managing big data is reliable and efficient data placement. [9] use of flexibility in the data placement policy to boost energy efficiency in data centers and propose a scheduling algorithm. In addition, allocation of computer resources to task has also strained much concentration. Cohen et al. [10] developed new design attitude, techniques and knowledge providing a new magnetic, agile and deep data analytics for one of the world's major marketing networks at Fox Audience Network, by using Greenplum parallel database system.

III. HADOOP DISTRIBUTED FILE SYSTEM

The Hadoop Distributed File System (HDFS)—a subproject of the Apache Hadoop project—is a distributed, highly fault-tolerant file system designed to run on low-cost commodity hardware. HDFS provides high-throughput access to application data and is suitable for applications with large data sets. This article explores the primary features of HDFS and provides a high-level view of the HDFS architecture.

3.1 Name Node and Data Node

Within HDFS, a given name node manages file system namespace operations like opening, closing, and renaming files and directories. A name node also maps data blocks to data nodes, which handle read and write requests from HDFS clients. Data nodes also create, delete, and replicate data blocks according to instructions from the governing name node. Name Node and Data Node sends message each other to prove their identity

3.2 Data Replication

An application can specify the number of replicas of a file at the time it is created, and this number can be changed any time after that. The name node makes all decisions concerning block replication. HDFS uses an intelligent replica placement model for reliability and performance. Optimizing replica placement makes HDFS unique from most other distributed file systems, and is facilitated by a rack-aware replica placement policy that uses network bandwidth efficiently.

IV. SYSTEM MODEL

4.1 Network Model

Geo Distributed data center topology is considered, in which all the available servers of the same data center (DC) are related to their local switch, while the different data centers will communicate through switches. A set I data centers, and each data center i € I consists of a set of servers Ji that are connected to a switch mi € M with a local transmission cost of CL. The entire system can be represented as a directed graph G = (N,E). The vertex set N = M U J includes the set of all switches and the set of all servers, and E is the directional edge set. All servers are connected to their local switch. while the switches are connected via inter-data center links. The weight of individual link $W^{(u,v)}$ can be defined as

$$W^{(u,v)} = \begin{cases} C_R, \text{ if } u,v \in M \\ C_L, \text{ otherwise} \end{cases}$$

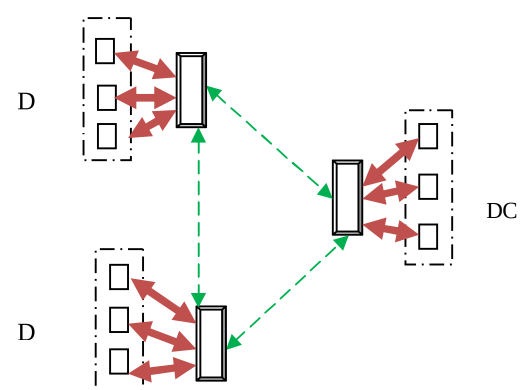

Fig 1: Data Center Topology

4.2 Task Placement

Suggest an automated task placement method Volley for geographically distributed cloud services with the reflection of WAN bandwidth cost, capacity limits of data center and inter-dependencies between data. Cloud computing services use

© 2017 IJAICT **(www.ijaict.com)**

Corresponding Author: Ms. T. Cowsalya, SVS College of Engineering, Coimbatore, Tamilnadu, India.

ISSN 2348 – 9928
Doi:01.0401/ijaict.2014.07.27 Published on 05 (4) 2017

Volley by submitting datacenter requests as logs. Volley system analyzes such logs by using an iterative algorithm.

4.3 Task Model

Big data tasks targeting on data stored in a distributed file system that is built on geographically distributed data centers. The given data is divided into a set of N chunks. Each chunk n € N has the size of Φn(Φn<=1) which are then normalized to the server storage ability. Z-way replica is used that is, for each chunk, Z copies will be stored in the distributed file system for fault-tolerance.

V. CONSTRAINTS ON BIG DATA PROCESSING

There are three constraints on big data processing they are constraints on task placement, data loading and QOS constraints.

5.1 Constraints of task Placement

A binary variable Yjk is defined to denote whether data chunk n is placed on server j as follows

$$y_{jk} = \begin{cases} 1; \text{ if chunk k is placed on server j;} \\ 0; \text{ Otherwise} \end{cases}$$

5.2 Constraints on remote data loading

When a chunk is required by the server there may be a internal or external data transmission. The routing information is done by the switch. All the nodes in the graph can be separated into three categories like source node, intermediate node and a destination node.

Source nodes are the servers in which the data chunks are stored. Intermediate nodes receive the data from the source node and forwards to the destination node based on t routing strategies. When the required data chunk is not stored in the destination node, it must receive the data chunks in the rate the request rate for chunk on server and the CPU usage of the chunk on the server.

5.3 Constraints on QOS

The processing rate and loading rate for data chunks on the server be µjk and γjn. The processing can be based on Two Dimensional Markov Chain model, where each state (u,v) represents u as pending task and v as available task. Let θjk be the computation resources. The processing rate of the task proportional to processing resource usage and it is represented as µjn = αj .θjn.

The given graph is divided into three regions; Region 1: all the states in the first line except the state (0,0). State (p,0),(p>1) transmits the data to its two neighboring nodes. In region 2: all the states in the diagonal line except (0,0). In region 3: all remaining states in the central region.

VI. PERFORMANCE EVALUATION

We analyzed the performance of joint-optimization algorithm and also we compare it with another optimization scheme algorithm ("Non-joint"), which it finds minimum number of servers to be activated and the data routing scheme using the network model.

The given data is preprocessed first and the preprocessed data is given into the geographically distributed data center. we consider |J| = 6 data centers, each of which is with the same number of servers. The data center link communication cost are set as CL = 1 and CR = 4, respectively. The cost Pj on each activated server j is set to 1. The data size, storage requirement, and task arrival rate are all randomly generated.

A. *6.1 Performance based on number of servers*

When the total number of servers increases as shown in fig. 2 the communication cost of both joint and non-joint optimization algorithm will decreases significantly. This is because more tasks and data chunks are placed in the same data center when more servers are placed in the data centers. Hence the communication cost greatly reduced.

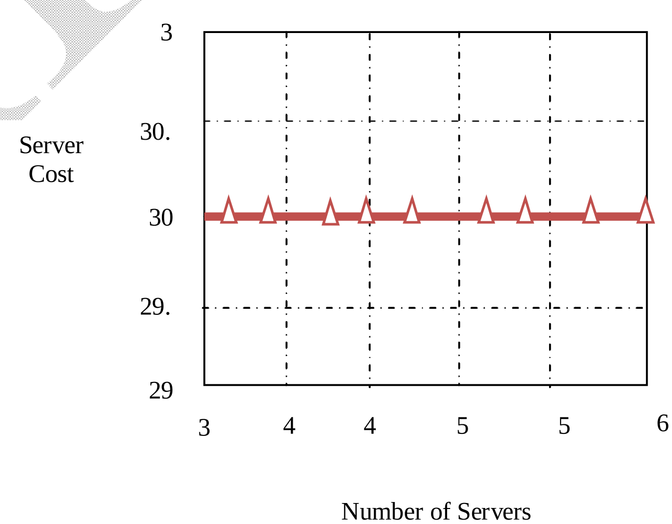

Fig.2. server Cost

© 2017 IJAICT (www.ijaict.com)

Corresponding Author: Ms. T. Cowsalya, SVS College of Engineering, Coimbatore, Tamilnadu, India.

ISSN 2348 – 9928
Doi:01.0401/ijaict.2014.07.27 Published on 05 (4) 2017

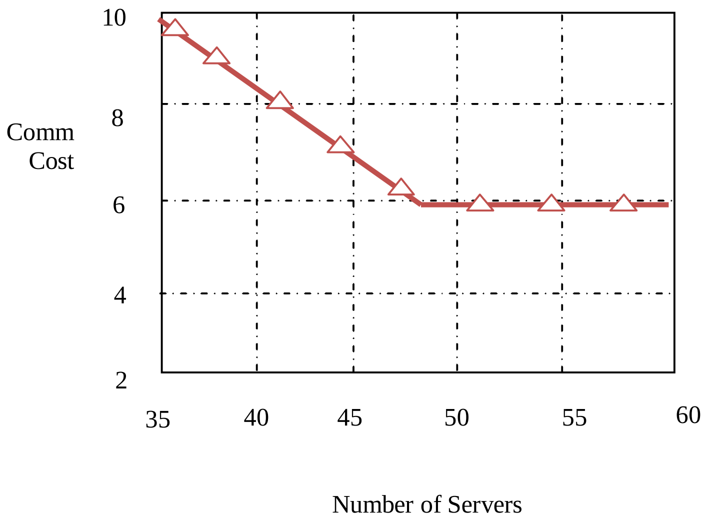

Fig.3. Communication Cost

More tasks and their corresponding data chunks can be placed in the same data center, or even in the same server. Further increasing the number of servers will not affect the distributions of tasks or data chunks any more.

6.2 Performance Based on Data Size

Fig. 4 shows the cost as a function of the total data lump size from 8.4 to 19. Larger chunk size leads to activating new servers with increased server cost. At the same time, fig. 5 shows more resulting traffic over the links creates higher communic

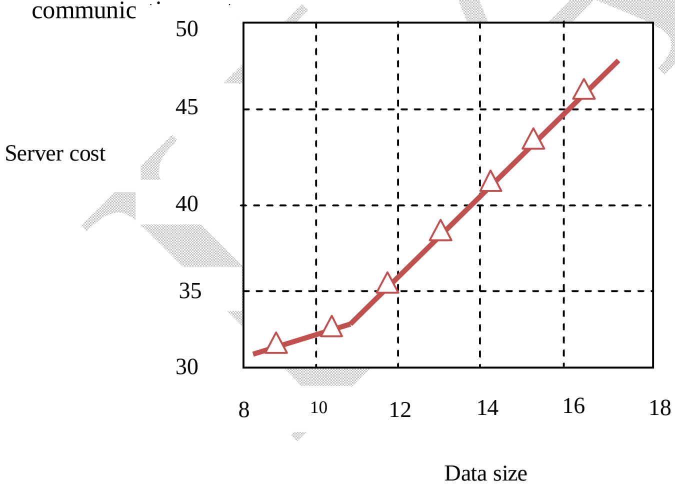

Fig.4. Server Cost

The data chunks are placed in multiple servers and each data chunk is processed separately. when the delay requirement is very small, more servers will be activated to guarantee the QoS. Therefore, the server costs of both algorithms decrease as the delay constraint

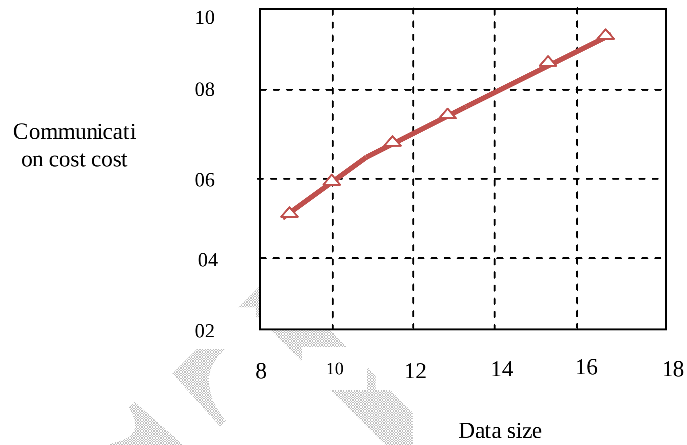

Fig.5. Communication Cost

VII FAULT TOLERANT

Here fault tolerant plays an important role. Data loss and node failure can be easily identified using heartbeat messages. A heartbeat message is a message sent from an originator to a destination that enables the destination to identify if and when the originator fails or is no longer available. Heartbeat messages are typically sent non-stop on a periodic or recurring basis from the originator's start-up until the originator's shutdown.

When the destination identifies a lack of heartbeat messages during an anticipated arrival period, the destination may determine that the originator has failed, shutdown, or is generally no longer available. Heartbeat messages may be used for high-availability and fault tolerance purposes

VIII CONCLUSION

In this paper, we jointly study the data placement, data Assignment and data center resizing and data movement to minimize the operational cost in geographically distributed data centers. We first describe about the data processing process using a two-dimensional Markov chain and derive the expected completion time in closed-form. In future the nodes in the cluster can be arranged in hierarchal order by using tree data structure. It reduces the processing time and increases performance.

© 2017 IJAICT **(www.ijaict.com)**

Corresponding Author: Ms. T. Cowsalya, SVS College of Engineering, Coimbatore, Tamilnadu, India.

ISSN 2348 – 9928
Doi:01.0401/ijaict.2014.07.27 Published on 05 (4) 2017

References

[1] Data Center Locations," http://www.google.com/about/data centers/inside/locations/index.html.

[2] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated Data Placement for Geo-Distributed Cloud Services," in The 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2010, pp. 17–32.

[3] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment," in Proceedings of the 29th International Conference on Computer Communications (INFOCOM). IEEE, 2010, pp. 1–9.

[4] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal Power Cost Management Using Stored Energy in Data Centers," in Proceedings of International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS). ACM, 2011, pp. 221–232.

[5] X. Fan, W.-D. Weber, and L. A. Barroso, "Power Provisioning for A Warehouse-sized Computer," in Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA). ACM, 2007, pp. 13–23.

[6] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar, "Benefits and Limitations of Tapping Into Stored Energy for Datacenters,"in Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA). ACM, 2011, pp. 341–352

[7] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's Not Easy Being Green," in Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM). ACM, 2012, pp. 211–222.

[8] S. A. Yazd, S. Venkatesan, and N. Mittal, "Boosting energy efficiency with mirrored data block replication policy and energy scheduler," SIGOPS Oper. Syst. Rev., vol. 47, no. 2, pp. 33–40, 2013.

[9] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, and C. Welton, "Mad skills: new analysis practices for big data," Proc. VLDB Endow., vol. 2, no. 2, pp. 1481–1492, 2009.

[10] R. Kaushik and K. Nahrstedt, "T*: A data-centric cooling energy costs reduction approach for Big Data analytics cloud," in 2012 International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2012, pp. 1–11.

IJAICT

© 2017 IJAICT **(www.ijaict.com)**

Corresponding Author: Ms. T. Cowsalya, SVS College of Engineering, Coimbatore, Tamilnadu, India.