Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 10, October 2013, pg.36 – 42

RESEARCH ARTICLE

Enhancing MapReduce Functionality for Optimizing Workloads on Data Centers

Gowtham Krishna Kumar S[1], Ramesh Ragala[2]

[1]M.Tech (CSE), ASCET, Gudur, India
[2]Associate Professor, ASCET, Gudur India

[1] gowtham369@hotmail.com; [2] rameshragala@gmail.com

Abstract— In cloud computing environment, data centers are used to provide the services to storage computation. Most of the applications storing the data in data centers. Now a day's even terabytes of data are supposed to be stored in data centers of cloud. The input datasets are geographically distributed on data centers. In many of the real applications, the data centers need to handle more requests. In order to handle more requests by the data centers, it uses more resources. So to reduce the resources used by the data centers, we designed a framework which is using the agents with MapReduce functionality. The MapReduce mechanism is commonly used for processing large datasets. In this paper, we analyse the possible ways of executing jobs and used to determine the scheduling of job sequences with respect to the execution time and monetary cost by the MapReduce functionality. Our evaluation shows that using MapReduce functionality with agents improves the processing time and cost of geographical distribution of datasets across data centers.

Keywords: - Cloud; Data Centers; Distributed Workloads; Scheduling; Cost & Time; Agents; MapReduce

I. INTRODUCTION

The growth of internet access and internet speed has a result to cause vast amount of data available online. In addition, many applications have created large volume of structured, semi-structured and unstructured data called big data, which needs to be processed and analyzed. Big data analysis represents one of the major challenges of this era. In cloud computing environment data centers are used to store data being collected from various levels of resources. Data centers [5] are increased their number and their volume size in the cloud environment. Much number of applications can use this type of environments for computation of tasks easily. Data centers mainly consist of servers, data storage and LANs. Data centers are connected to the rest of the network through a gateway. Servers are responsible for handling the user requests based on their computation and as well as allocation of resources based on their need of computation. The server performance depends on the computational tasks held by the server. Data centers mainly concerns with server management, power consumption, security, data management and performance. Virtualized environments can simplify the cluster management. In public cloud or private cloud, virtual machines are running on many interactive applications like web servers. Number of users can share a single server and increases the utilization of resources through server virtualization [6]. These will help to increase the efficiency, but is challenging to analysing the workloads. Massive computing clusters are increases for computation of workloads. The cost of these clusters makes it critical to improve their operating efficiency and also energy. Consequently, there is an effort to improve the energy efficiency and workload efficiency for data centers at both academic and industry level.

A. Data Geographical Distribution

More and more applications data on cloud data centers are geographical distributed for any of the subsequent reasons:

1. Data is stored in nearest respective sources or frequently accessed components which can be distributed globally [3], but the data is analyzed globally. Suppose a regional based organization produces data sets being managed to provide localized fast access for common operations (e.g., sales). All data sets are analysed as a whole.
2. Data is gathered [4] and stored by different data centers shared towards a common goal. Example (2) data stored collected and stored (thousands of GBs respectively) for global analytics.
3. Data is replicated across data centers for availability. Example (3) is used to overcome failures but don't replicate all the data items on data centers this will leads to waste the resources [7].

Although how to analyze such geographical distributed datasets most efficiently with a MapReduce functionality? Suppose administrator to execute multiple datasets across data centers if support all [2]. Gathering all datasets to one place called one data center agent from all data centers. Data center agent is responsible to gathering all data sets across all data centers. After gathering perform MapReduce operations on all datasets.

In brief, MapReduce functionality works by divides the input datasets into smaller chunks and these processing in a parallelize steps. In MapReduce job, *mapper* and *reducer* are essential functions. In mapper function, set of mapper process reads respective input datasets chunks and produces <*key*, *value*> pairs called "intermediate values". In reducer function, each reducer process applies the reduction based on aggregate functions to all values of each *key* assigned to it. The concerns in data centers are server management, data management, and performance are easily achieved through the MapReduce functionality. MapReduce enhances the computational resources in the cloud computing environment.

B. Job Scheduling Sequences

MapReduce jobs are executed on a given input datasets by applying the first job on the given input dataset, applying the second job on the output and so on like this. Page Rank [8] is an algorithm which constructs a graph for handling the large web caches through intra- page relationships. Hash tree generation [9] is used to perform the integrity of data. In hash tree each level can be generated by using MapReduce jobs from leaves in a graph, which represents the original input dataset. Repeatedly, distinct MapReduce jobs are applied in sequence order performing the same job than the iterative order. Using Pig to explain complex data analysis tasks with data-flow graphs from MapReduce jobs are generated automatically. When MapReduce jobs are performed in sequence the possible number of schedules increases significantly. We use the DTG (Data Transformation Graph) algorithm [10] is optimized schedule for performing the sequence of MapReduce jobs and also data center infrastructure. DTG algorithm schedule is also used to optimize the execution time and execution cost. So our solution is to determine the execution time and execution cost of MapReduce jobs across data centers and also identifies the data center loading per hour also calculates the data transfer cost of data center. MapReduce operations are performed within one or more data centers using clusters available in each of the available data centers.

C. Agents

Software agents [11] can be used for implementing the intelligence in cloud computing. These are flexible in the resource management and also running in large scale applications. In this paper we are using the agents for gathering the data across all data centers. So we implement the data center agent for that concern. Data center agent monitors the all data centers which are connected to data center agent. Data center agent can act as a single data center.

The remainder of the paper is represented as Section 2 describes background information on MapReduce Functionality. Section 3 describes the System Model Implementation. Section 4 describes Performance Calculation. Section 5 represents Conclusion of paper.

II. BACKGROUND

This section describes the trendy Hadoop open source Apache Project also its MapReduce functionality.

1. Hadoop

Hadoop [1] is a free open source frame work for distributed applications. It provides reliability, scalability and also provides new methodology to storing and processing data in distributed environments. It implements MapReduce model using Hadoop Distributed File System (HDFS). HDFS is having high availability and unlimited storage. It is store the input and output files but not storing the intermediate results on the *mapper* phase. Hadoop provides a feature called Replication. By default Hadoop provides Replication for the data is 3.

HDFS is highly fault-tolerance and is used to deploy on low cost hardware. It provides high throughput access to application data and is suitable for applications that have large datasets. Hadoop architecture follows master-slave model. Job Tracker is a master node divides the works into tasks. Also take decisions on task assignments. Name Node manages the Name Spaces. It contains all information about data. There is only one Name Node in each cluster. Secondary Name Node to overcome the failure of Name Node. Data Node holds the actual file system data. One or more Data Nodes in each cluster. Task Tracker accepts tasks from Job Tracker and processes the data.

2. MapReduce Functionality

MapReduce functionality [12] consists of two functions called *mapper* and *reducer* functions. The execution of these two functions in phases. A MapReduce job divides the input dataset into individual chunks which are processed with *mapper* function in parallel. The framework shuffle and sorts the outputs of the *mapper* phase, which are then input to the *reducer* phase. Typically both the input and output of the jobs are stored in a file system called HDFS. The framework takes care of scheduling, monitoring of tasks and re-executes of failed tasks. The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types. Input and Output types of a MapReduce job:

(Input) <k1, v1> -> **map** -> <k2, v2> -> **combine** -> <k2, v2> -> **reduce** -> <k3, v3> (Output).

III. IMPLEMENTATION

This section describes the implementation process of the model.

A. Data and System

Suppose we have n data centers, named by DC_1, DC_2, DC_3... DC_n with input datasets ds1, ds2, ds_3... ds_n respectively. The total amount of input dataset is $|ds|=\sum_{i=1}^{n}|ds_i|$. The total bandwidth between the data centers is DC_a and DC_b (a! =b) is given by $B_{i,j}$ and the cost of transferring a unit of data between the same two data centers is $C_{i,j}$.

User requests can transfers to the data centers by using service broker policy. Service broker policy [24] [12] setup with closest data center and optimize response time policies. Based on requirement of user requests, the requests are moved to data centers by service broker policy.

User requests are not directly sends to the data centers. User requests are handled by the service broker policy. The service broker policy has setup with mainly nearest data center and optimizes response time. Based on the user request the service broker policy transfers their requests to appropriate data centers. After that assigns user requests with scheduling to execution of user requests.

Scheduling also key factor is in this work. In hadoop, fair [13] and FIFO [14] are the common used schedulers. And most advanced scheduling strategies available for Hadoop framework are dynamic priority scheduler [15], Hadoop capacity [16]. In this work, we used the Round robin [17] and throttled [18] scheduling mechanisms to handling the tasks. Internet characteristics plays vital role in this work. Delay Matrix [19] and Bandwidth Matrix [20]. Transmission delay is defined between regions. It is calculated in milliseconds. Bandwidth is allocated to each region in terms of Mbps. Data centers are configured with these attributes are operating system, hardware architecture type, VM cost, memory cost, storage cost, data storage cost, physical hardware units. User requests are configure with requests per hour, data size per hour.

B. Operations

Data Transform Graph algorithm performing on given input dataset. The sequence of MapReduce jobs mr_1, mr_2, ..., mr_n have to be executed on two phases called *mapper* and *reducer* phases.

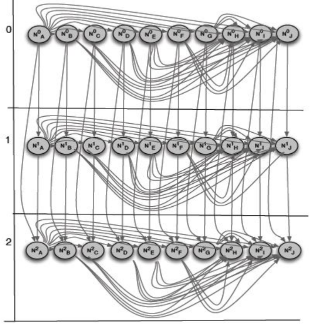

A= <1,2,3>
B= <2,2,3>
C= <3,2,3>
D= <1,1,3>
E= <1,3,3>
F= <1,2,1>
G= <1,2,2>
H= <3,3,3>
I= <1,1,1>
J= <2,2,2>
0
1
2
N^0_A N^0_B N^0_C N^0_D N^0_E N^0_F N^0_G N^0_H N^0_I N^0_J
N^1_A N^1_B N^1_C N^1_D N^1_E N^1_F N^1_G N^1_H N^1_I N^1_J
N^2_A N^2_B N^2_C N^2_D N^2_E N^2_F N^2_G N^2_H N^2_I N^2_J

Fig. 1: Example of DTG for MapReduce Sequence for single job executing on data centers

Before performing MapReduce phase, Data center agent gathers all the information from across all data centers and hosts a MapReduce cluster. After completion of *mapper* phase, the data center agent sends the *mapper* phase intermediate results to appropriate data centers for *reducer* operation. After completion of reducer phase the data center agent retrieves all reduce phase results for results.

Data center Agent

Data center agent is responsible for receiving and sending the processing requests from across all data centers. A Data center agent receives all information from all data centers and submits inputs to VM. *Reduce* phase starts only when *map* phase completed. Data center agent is constructed with ABLE (Agent Building and Learning Environment) [21], is a JAVA framework and toolkit for developing agents.

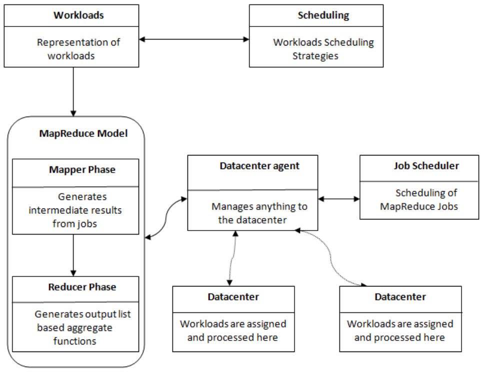

Workloads
Representation of workloads
Scheduling
Workloads Scheduling Strategies
MapReduce Model
Mapper Phase
Generates intermediate results from jobs
Reducer Phase
Generates output list based aggregate functions
Datacenter agent
Manages anything to the datacenter
Job Scheduler
Scheduling of MapReduce Jobs
Datacenter
Workloads are assigned and processed here
Datacenter
Workloads are assigned and processed here

Fig 2: Execution Flow of MapReduce Functionality on distributed Data Centers

Data centers are monitoring with these steps: *M*onitoring, *A*nalyzing, *P*lanning, and *E*xecuting steps.

Monitoring: First, data centers are dynamic. New equipment is continuously added and existing equipment is often reconfigured or removed, the both kinds of changes to the supporting power, cooling, and network infrastructure. Similarly new equipment offers new capabilities and functions. Second, data centers and consolidation, the number of objects and measurement points within the data centers continues to grow. Data gathered from many sources. Data with different formats, published through several interfaces. The data collection engine that implements the elements of the interface communication associated with the data source. The communication interfaces gather a time stamp when corresponding item was recorded. Different timestamp sensors data is available at the data sources. The data collection and filtering agents is to filter the all incoming data streams already in the database. Filtering can reduces the amount of data is stored in database that leads to improving scalability.

Analyzing: Data analysis with mainly two functions is correlation and attributes behaviour. Attribute behaviour describes the values change over time and as well as the value of observed readings. Attribute behaviour analysis is useful to provide information to establish the filtering policies. Data correlation methods examine the other attributes and strength of correlations. This analysis is to determine the minimum set of control points to control the attributes.

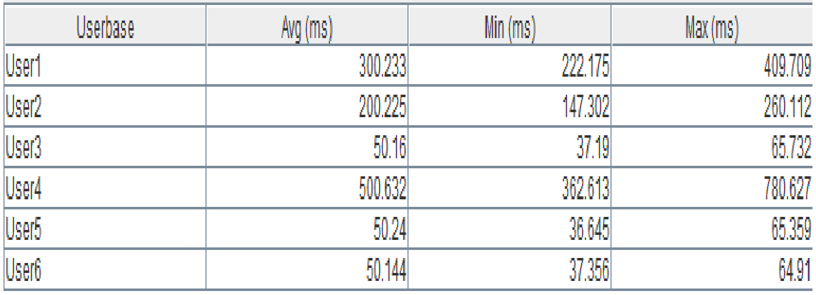
Planning: Planning process [22] assigns the responsibilities, scheduling, configuring project management, and budgeting. Planning can be evaluated in five steps are defining IT parameters, developing system concept, determining user requirements, generating specification, and generate detailed design.

Executing: Executing process [23] assigns acquire and implement steps. In acquire step, gathering the equipments to execute like shipment and site co-ordination. In implement step, complete the execution by the assembly, installation of sub contracts, start-up, network integration and orientation and training phases.

IV. PERFORMANCE EVALUATION

This section is to evaluate the effectiveness of our analysis of workloads with MapReduce functionality across six data centers was mainly (1) calculated the overall response time of data center to processing the workloads. (2) Calculated the region response time of user requests in hourly base. (3) Calculated the data center request servicing times. (4) Finally calculated the total cost of Data center transferring data.

A. Response Time by Location



Userbase	Avg (ms)	Min (ms)	Max (ms)
User1	300.233	222.175	409.709
User2	200.225	147.302	260.112
User3	50.16	37.19	65.732
User4	500.632	362.613	780.627
User5	50.24	36.645	65.359
User6	50.144	37.356	64.91

Fig 3: Response Time of User Requests in Locations

B. User's hourly Response Time

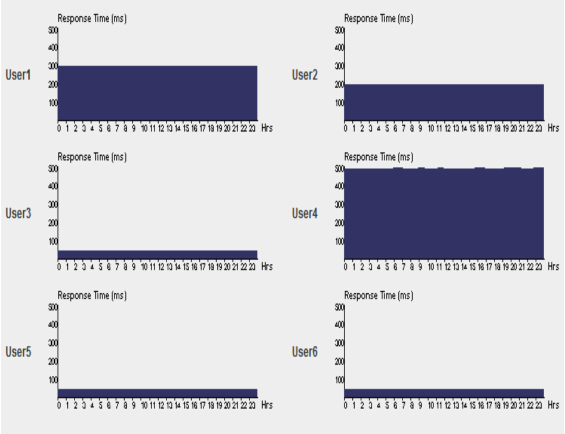
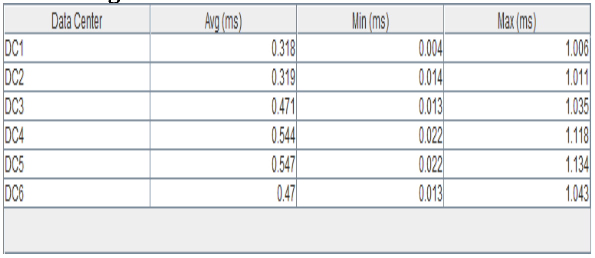


Fig 4: Response Time of User Requests in hourly base

C. Data center request servicing time



Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.318	0.004	1.006
DC2	0.319	0.014	1.011
DC3	0.471	0.013	1.035
DC4	0.544	0.022	1.118
DC5	0.547	0.022	1.134
DC6	0.47	0.013	1.043

Fig 5: User Request Servicing Time of Data center

D. Data center Loading

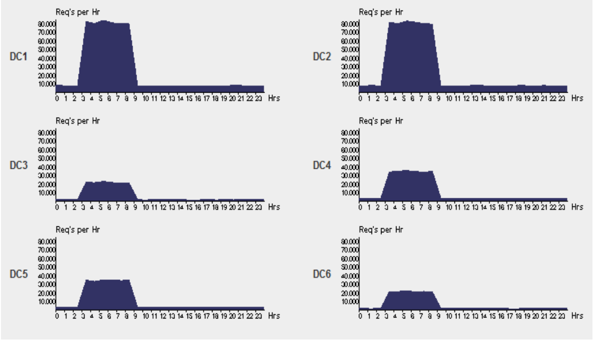
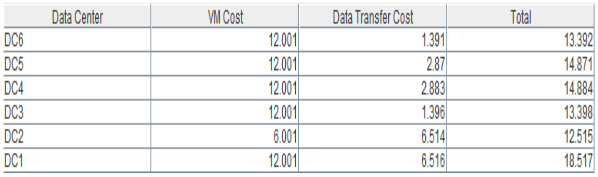


Fig 6: Data center Loading in hourly base

E. Cost of Data center



Data Center	VM Cost	Data Transfer Cost	Total
DC6	12.001	1.391	13.392
DC5	12.001	2.87	14.871
DC4	12.001	2.883	14.884
DC3	12.001	1.396	13.398
DC2	6.001	6.514	12.515
DC1	12.001	6.516	18.517

Fig 7: Cost of Data center to handle User Requests

F. Total cost:

Total virtual machine cost	: $66.1
Total data transfer cost	: $21.57
Grand total	: $87.67

V. Conclusion

This paper shows the MapReduce framework can be used to execute the sequence of MapReduce jobs on distributed datasets across Data centers efficiently. MapReduce framework uses the DTG algorithm is to perform the MapReduce jobs to minimize the cost and execution time. We believe our framework is also applicable to single Data centers also. In our framework, we can achieve the better performance results with MapReduce framework compared to earlier to execute the sequences of MapReduce Jobs.

References

[1] Apache Hadoop

[2] Across Data centers http://mailarchives.apache.org/mod_mbox/hadoopgeneral/201001.mbox/%3C4B4B6AC7.6020801@lifeless.net%3E.

[3] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan. Volley: Automated Data Placement for Geo-Distributed Cloud Services. In NSDI 2010.

[4] Data from Japan Census Data http://aws.amazon.com/datasets/Economics/2285

[5] Data center evolution - A tutorial on state of the art, issues, and challenges, Krishna Kant, Intel Corporation, Hillsboro, Oregon, USA.

[6] Introduction to virtualization Morty Eisen april, 2011.

[7] V. Ramasubramanian, T.L. Rodeheffer, D.B. Terry, M. Walraed- Sullivan, T. Wobber, C.C. Marshall, and A. Vahdat. Cimbiosys: a Platform for Content-based Partial Replication. In NSDI 2009.

[8] PageRankAlgorithm. http://en.wikipedia.org/wiki/PageRank.

[9] Optimally-balanced Hash Tree Generation in Ad Hoc Networks, V. R. Ghorpade.

[10] Transforming Graph Data for Statistical Relational Learning, Ryan A. Rossi.

[11] Software Agents: An Overview, Hyacinth S. Nwana.

[12] A SOAP- based infrastructure for service broker, “Shin Nakajima”, Japan.

[13] Ross Benjamin Clay. Enabling MapReduce to Harness Idle Cycles in Interactive-User Clouds. Master’s thesis, NC State University, 2011.

[14] Schedulig algorithms, “Frédéric Haziza”, Department of Computer Systems, Uppsala University, spring 2007.

[15] Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments, International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011.

[16] Rong-En Fan (rafan) Yahoo! Search Engineering, Hadoop Taiwan User Group meeting 2009.

[17] Round Robin Scheduling http://siber.cankaya.edu.tr/OperatingSystems/ceng328/node125.html

[18] Throttled Scheduling approach: http://docs.oracle.com/cd/E19182-01/821-1075/cnfg_sch-bc-qos-features_c/index.html

[19] www.ittc.ku.edu/~jslites/723/.../the%20Transmission%20Matrix.pdf “Jim Stiles The Univ. of Kansas Dept. of EECS”

[20] bandwidth matrix: http://mathworld.wolfram.com/Bandwidth.html.

[21] A SOAP- based infrastructure for service broker, “Shin Nakajima”, japan.
[22] Data center projects: System Planning, “Neil Rasmussen, and Suzanne Niles”.
[23] Data Center Projects: Standardized Process, “Neil Rasmussen and Suzanne.
[24] Dhaval limbani and Bhavesh Oza, “ A proposed service broker strategy in cloud for cost effective data center selection” International Journal of Engineering Research and Applications, India, Vol.2, Issue 1, jan-12, pp. 793-797.