ISSN:2231-0711

Jamuna Rani M R et al | IJCSET |May 2013 | Vol 3, Issue 5, 183-185

Load Balancing Technique for Climate Data Analysis in Cloud Computing Environment

Jamuna Rani M R [#], K C Gouda [*], Nirmala H [#]

[#] *Department of Computer Science and Engineering, SJB Institute of Technology (VTU)*
BGS Health & Education City, Kengeri, Bangalore-60,India

[*] *CSIR Centre for Mathematical Modelling and Computer Simulation (C-MMACS)*
Wind Tunnel Road, Bangalore-37, India

***Abstract*— Climate data analysis requires huge computing as the data are multi-dimensional and of large size. In this paper, we aim at developing a prototype for large climate data analysis in cloud computing environment. A utility is being developed for executing the climate models in a cloud computing system, also big climate data analysis are carried out using multi-source data like satellite observation and various climate model output data. On the growing importance of climate studies and High Performance Computing, different users starting from a farmer to a scientist to a policy maker needs to interact and make the outreach and network strongly. Keeping this in mind a design is developed as cloud framework to provide cloud clustering, work flow scheduling, resource allocation & management of distributed data storage, data analytics and visualization service. The web contents have to dynamic and it has motivated large number of people to use it. Cloud's elasticity has helped to provide service for users who work on climate models and also make the used applications scalable. To reduce communication overhead; an algorithm called Join-Idle-Queue(JIQ) is been implemented in the present study. This framework helps perform better understanding and performance on multi-scale and multi-dimensional climate data analysis..**

Keywords*— *Cloud computing, climate data, load balancing, Join-Idle-Queue.

I. Introduction

Cloud is a collection of interconnected and virtualized computers that are dynamically provisioned and presented as more unified computing resources based on service-level agreements established between the service provider and consumer. The cloud service consumer is provided basically with 4 types of service i.e., SaaS, Paas, Iaas, DaaS. Cloud computing has characteristics of rapid developing resource pool which is unpredictable. It is supported by cloud's feature like scalability. This feature helps for extraction, analysis and processing of data. Data analytics is used to analyze raw data and provide summary and detect faults in the climate data provided by the user. When analyzing data the analyst may come across the problems like limited computing resources, huge datasets, larger complexity and less understanding with various data structures and usage of various tools. Designing a front end at cloud centers have become a challenge with respect to the rapid developments of web and it is also tedious to balance the load.

In the first section multi-scale climate data and the load balancing technique are discussed. The balancing techniques with different loads are also analyzed and presented.

II. Multiscale Climate Data

Generally the weather and climate model runs over different scenarios which depends on the type of grid, resolution, parameterization schemes etc. used in the model. The important points of consideration also depends on the different types of parameters like boundary condition (oceans, mountain, topography etc) or Initial conditions (atmospheric variables like rainfall, wind, humidity, temperature etc). These variables are covered spatio-temporally, which are represented in fixed resolutions and those can further interpolated to higher scale. The multi-scale data can be very from a station scale to the global scale (in space) and from hourly to decade (in time) and these data are multi-dimension in nature. These data are stored in various formats like binary, GRIB, netCDF, HDF etc.

To obtain the results at different spatial and temporal scales, three datasets are analysed in this study. They are thematic data, episodic data and trajectory data. Thematic data is a 2D representation which indicates both space and time. Episodic data relates to specific event at a particular co-ordinate. Trajectory data is associated with specific event and a spatio-temporal path is followed by these events.

III. Load Balancing Technique

A web server farm consists less servers whereas a cloud data centers consists massive number of processors. Based on the demand of cloud user the resources should scale horizontally by imposing elasticity in the data centers. But for this if we are using traditional methods then a single hardware must be used to serve several servers. In the previous works each dispatcher used to serve multiple servers in an ordered fashion by joining the shortest queue. It caused no communication overhead as the dispatcher knew about all the requests processed to the server. But this became infeasible with the increasing demands by increasing the complexity of the system as it required the information of queue length of all servers at router and lacked in the robustness of the system.

Load balancing with distributed dispatchers routed the requests to random servers instead of sending them to shortest one. Some requests may take less time and some takes more to get processed. But each job is balanced independently by dispatcher and hence all the dispatcher may not come to know about the actual number of jobs in the server. Fig. 1 illustrates design of dispatchers in cloud cluster.

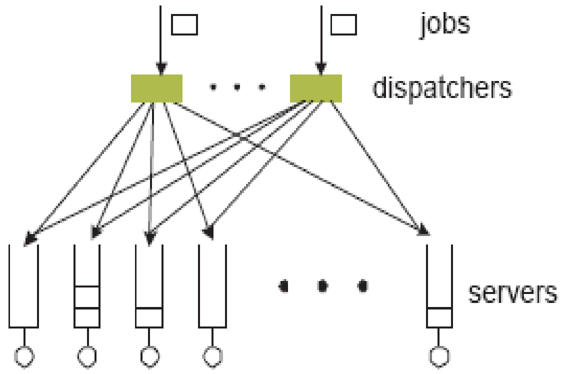

Fig. 1 Design of dispatchers in cloud cluster

To overcome the drawback of previous technique a new data structure is being used called as I-queue. This I-queue consists of subsets processors that are in idle state. This I-queue is placed at each dispatcher which helps to find the idle server in the process of job assignment. A novel algorithm called Join-Idle-Queue (JIQ) is being used with primary and secondary balancing technique which makes use of these I-queues. Hence each dispatcher is equipped with an I-queue and all processors can be access the dispatcher. Fig. 2 illustrates the overall system with an I-queue at each dispatcher.

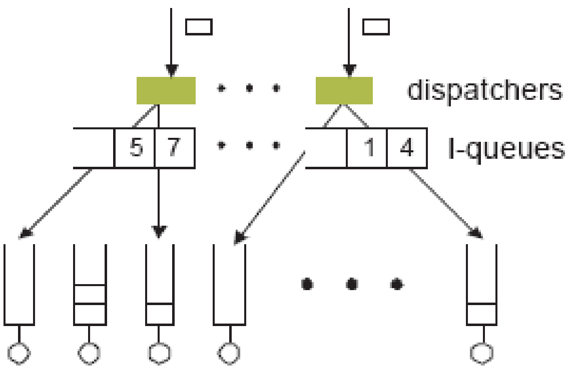

Fig. 2 Dispatchers which is equipped with an I-queue.

(a) Primary load balancing:

Here the system finds the information of idle servers/processors present in the I-queues and reduces the communication overhead which modifies the server loads. When a job arrives, the dispatcher consults the I-queues, if it is empty the jobs are directed randomly to the processor, if the processor is idle then it reports back to the I-queue about the idleness. Each I-queue is provided with information about idle processor only once to reduce communication overhead. If the I-queue is non-empty then the dispatcher removes the first idle processor from the I-queue and an idle processor is provided with the job.

(b) Secondary load balancing:

As soon as the processor becomes idle, it chooses one I-queue based on algorithm and informs about the idleness to the I-queue or joins the I-queue. So to provide information in reverse direction two load balancing algorithms are considered: random and SQ(d). Using random algorithm, the I-queue chooses the processor in random but uniformly, and, with SQ(d), d random i-queues are provided with idle processor and joins the one with shortest queue length. This algorithm provides an advantage of one-way communication where it doesn't require message to be sent from I-queues.

We discuss the performance of random algorithm along with its analysis. The implementation is done in HPC environment at CSIR Centre for Mathematical Modelling and Computer Simulation (C-MMACS).

(c) Analysis of secondary load balancing:

In a system it consists of n processors and m dispatchers, where m<<n. Requests arrive at rate of nλ and each request is sent to dispatchers randomly and then forwarded to processor.

The ratio for number of servers to numbers of I-queues is calculated as,

$$r \equiv \frac{n}{m}$$

I-queues occupied in the system when it is in equilibrium is calculated as follows: ratio for number of servers to numbers of I-queues is calculated as,

$$\frac{\rho}{1-\rho} = r(1-\lambda)$$

Proportion of empty I-queues are calculated as

$$1-\rho$$

Response time of the primary system can be calculated using the results of secondary load balancing system,

$$s = \lambda(1-\rho)$$

Where ρ is the calculated proportio of occupied I-queues.

Mean queue size at the servers is, $\bar{q} = \dfrac{q_s}{1-\rho}$

Mean response time, $\bar{T} = \dfrac{q_s}{s}$

IV. Results

The results are obtained for secondary load balancing algorithm using random technique. The results are analyzed for 100, 250 and 500 servers and dispatchers are varied as 50, 10 and 5. Figure 3-5 present the case study with 100, 250 and 500 servers respectively. The analysis with different dispatcher number (D) are presented in different colours in the graph, i.e. blue, red and green lines indicate the results for the experiment with values of D as 50,10 and 5 respectively.

The regression analysis is also done and linear equations and coefficients of slopes analyzed for JIQ algorithm and is tabulated as shown in table I. The first column in the table indicates the number of servers (Ns) and results from different D are presented in columns 2 to 4. The regression of load (x) and proportion of I-queues (y) and the correlation are presented in the table. These analysis shows that the proportio of empty I-queues increases with the load and becomes more at higher load. The analysis by changing the number of servers (like 100, 250 and 500) it is found that for the case of D (=50) the proportion is more compared to smaller values of D.

TABLE I

LINEAR EQUATIONS AND COEFFICIENTS OF SLOPES ANALYZED FOR JIQ ALGORITHM

Ns	D=50	D=10	D=5
100	y = 0.563x + 0.255 **R^2 = 0.94**	y = 0.43x - 0.001 **R^2 = 0.81**	y = 0.285x - 0.016 **R^2 = 0.75**
250	y = 0.55x + 0.062 **R^2 = 0.87**	y = 0.255x - 0.023 **R^2 = 0.74**	y = 0.148x - 0.017 **R^2 = 0.71**
500	y = 0.43x - 0.001 **R^2 = 0.81**	y = 0.148x - 0.017 **R^2 = 0.71**	y = 0.08x - 0.011 **R^2 = 0.69**

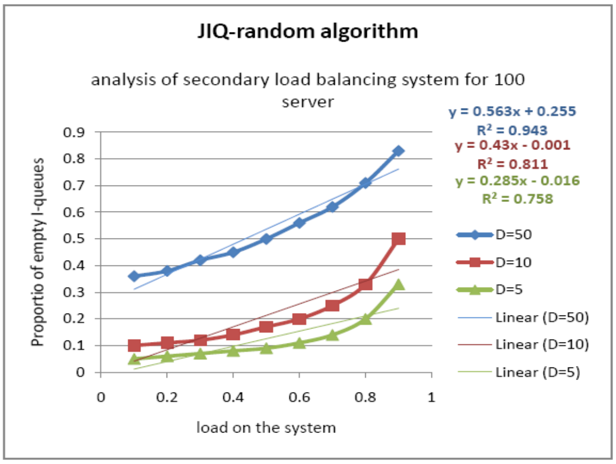

Fig. 3 JIQ random analysis for the case with 100 servers

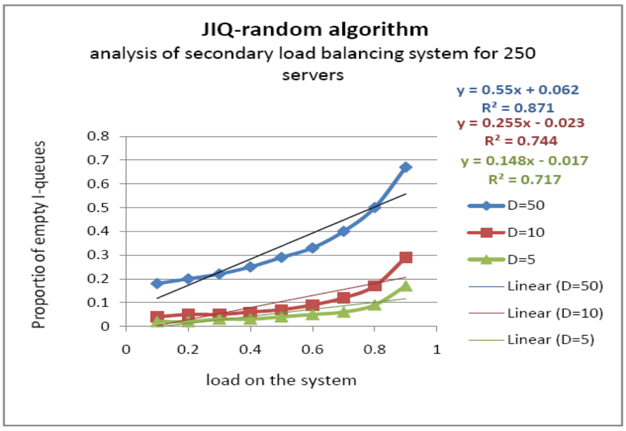

Fig. 4 JIQ random analysis for the case with 250 servers

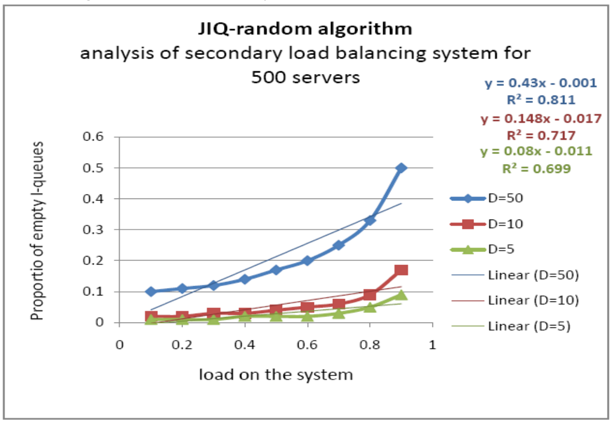

Fig. 5 JIQ random analysis for the case with 500 servers

V. RELATED WORK

In recent times several works related to cloud and grid computing and big data are carried out [1],[2]. The cost analysis of the cloud is also studied by several researchers and the efficient way of cloud cost is defined [3]. Resource allocation and scheduling work flow in the cloud is the important component as the users need on time services in the cloud [4] [5] [6]. Climate data in cloud are also touch upon by some works as explained by some authors [7]. A case study of tropical cyclone in cloud computing platform also analysed by some researchers [8].

VI. CONCLUSIONS

In this work a computing environment is created in cloud and load balancing technique is implemented, where loads are being varied and the performance is being analyzed for different types of climate data. The results obtained from different case studies indicate that the performance is improved and communication overhead is reduced when there is increase in number of servers. Hence in real time, data from different climate models are handled by using the JIQ algorithm and performance is optimized for different climate models. This work can be further extended to deal with different parameters (data) obtained from different servers and by multiple users.

ACKNOWLEDGMENT

The first author Jamuna acknowledges to SPARK program of CSIR C-MMACS, HOD Computer Science and Engineering & Principal, SJB Institute of Technology, Bangalore for providing the necessary infrastructures to carry out the work.

REFERENCES

[1] Naidila Sadashiv, S. M Dilip Kumar. Cluster, Grid and Cloud Computing: A Detailed Comparison. The 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. Super Star Virgo, Singapore.

[2] Divyakant Agrawal, Sudipto Das, Amr El Abbadi. Big Data and Cloud Computing: New Wine or just New Bottles.

[3] Sivadon Chaisiri. Optimization of Resource Provisioning Cost in Cloud Computing. Ieee transactions on services computing, vol. 5, no. 2, april-june 2012.

[4] Artin Avanes. Adaptive Workflow Scheduling Under Resource Allocation Constraints and Network Dynamics.

[5] Gouda K C, Radhika T V , Akshatha M, Priority based resource allocation model for cloud computing, International Journal of Science, Engineering and Technology Research (IJSETR), ISSN: 2278 – 7798, pp 215-219.

[6] Enda Barrett, Enda Howley, Jim Duggan. A Learning Architecture for Scheduling Workflow Applications in the Cloud. 2011 Ninth IEEE European Conference on Web Services.

[7] Sifei Lu. A Framework for Cloud-Based Large-Scale Data Analytics and Visualization: Case Study on Multiscale Climate Data. In 2011 Third IEEE International Conference on Cloud Computing Technology and Science.

[8] D. Hasenkamp, A. Sim, M. Wehner And K. Wu. Finding Tropical Cyclones on a Cloud Computing Cluster: Using Parallel Virtualization for Large-Scale Climate Simulation Analysis. 2nd IEEE International Conference on Cloud Computing Technology and Science.