QoS and Cost Aware Brokerage System for Selecting Green CDCs

by

Abdullah Al Noman

ID: 12101004

Farzana Noman Tamanna

ID: 12101041

A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BRAC UNIVERSITY

August 2016

Declaration

We, hereby, declare that the work presented in this thesis work is the outcome of the investigation performed by us under the supervision of Tamal Adhikary, Lecturer, Department of Computer Science and Engineering, BRAC University and Md. Shamsul Kaonain, Lecturer, Department of Computer Science and Engineering, BRAC University. We also declare that no part of this thesis work has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

. . . .

(Tamal Adhikary)

Supervisor

. . . .

(Md. Shamsul Kaonain)

Co-Supervisor

Signature

. . . .

(Abdullah Al Noman)

. . . .

(Farzana Noman Tamanna)

Abstract

Cloud service brokerage (CSB) systems are getting increasingly popular day by day to manage the performance and utility between cloud service providers (CSP) and cloud service consumers (CSC). An optimized CSB is supposed to be cost effective, energy efficient and should maintain the quality of service (QoS) according to the service level agreement (SLA). Additionally, a CSB needs to have the capability to select the optimal CSP which reflects the requirements of user requests. The state-of-the-art works on establishing a CSB system did not consider all the three objectives that a smart CSB system demands. In this work, we have defined a CSB model based on all of these three necessary criteria to achieve an optimized CSB which are energy efficiency, cost effectiveness and certainty of achieving QoS. The Mixed-Integer Linear Program (MILP) is used to formulate the objective function that will select a CSB. We simulate our proposed model in Matlab, compare our model with existing working models and the results show that significant performance improvement is achieved by our newly defined CSB.

Acknowledgements

First of all, we are thankful and expressing our gratefulness to the Almighty who offers us His divine blessings, patient, mental and psychical strength to complete this project work.

We are deeply indebted to our project supervisor Tamal Adhikary, Lecturer, Department of Computer Science and Engineering, BRAC University and co-supervisor Md. Shamsul Kaonain, Lecturer, Department of Computer Science and Engineering, BRAC University. Their scholarly guidance, important suggestions, endless patience, constant supervision, valuable criticism, and enormous amount of work for going through our drafts and correcting them, and generating courage from the beginning to the end of the research work has made the completion of the project possible.

Last but not the least; we are highly grateful to our parents and family members for their support and constant encouragement, which have always been a source of great inspiration for us.

Abddullah Al Noman

Farzana Noman Tamanna

August, 2016

Contents

4. .Problem Formulation

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Cloud computing is a distributed computing environment that is revolutionizing the way of computation over the last few years. Instead of keeping data on user's hard drive or updating applications for user needs, a cloud service can be used over the Internet, at another remote location. In recent days, cloud computing has become one of the most essential technology trend. As a result, different commercial cloud platforms are getting popular day by day to store information or to use their applications [1], [2], [3]. There are a large number of services that can be provided by Cloud Computing. These can be categorized as: (a) Platform as a service (PaaS), e.g., AppSclae, Google App Engine, Microsoft's Azure, Amazon Elastic Compute Cloud (Amazon EC2) etc. (b) Software as a service (SaaS), e.g., Amazon Web Services (AWS), Google Apps, Salesforce, impels CRM, Oracle etc. (c) Infrastructure as a service (IaaS), e.g., flexiscale, AWS Database as a service (DBaaS) etc. [4], [5].

Now-a-days, cloud computing has become one of the mostly used computing environment and one of the major forces changing the IT landscape. It has become an essential technology computation environment and commercial cloud platforms have begun to be deployed for performing massive amount of processing and calculation. Cloud computing has facilitatedthe consumers to use computation services in a pay-as-you-go model by delivering infrastructure, platform and software as services.

1.2 Cloud Brokerage System

With the immense growth of cloud computing, the network traffic has increased significantly. At the same time the energy consumed by the huge infrastructure of cloud service providers (CSPs) and the cost to operate these physical servers in the cloud data centers (CDC) has also increased [6]. Besides, the CSPs require to respond quickly and effectively to user requests. the cloud users face a great challenge in selecting the appropriate cloud services and resources that fit their requirements. A cloud broker is a third-party individual or business that acts as an intermediary between the consumer of a cloud computing service (CCS) and the provider of that service. A cloud broker, also known as Cloud Agent (CA), is a software application that facilitates the distribution of work between different CSPs. According to NIST [7], cloud broker is an entity that manages the usage, performance and delivery of cloud services and maintains the relationship between cloud service providers (CSP) and cloud service consumers (CSC).

1.3State-of-the-Art Works

Because of the increased growth in this field, many research works have been conducted on cloud services and on cloud service brokerage (CSB) system. Researchers of [8] ~ [10] tried to achieve an energy efficient optimized CSB and proposed different models. On the other hand, in [11] and [12] the researchers proposed a CSB in respect of quality of service (QoS). In [13] and [14], authors tried to achieve a cost effective brokerage system. None of the existing works has considered all the three important features of an effective brokerage system like cost, energy and QoS together.

1.4 Thesis Contributions

In our work, we propose an energy efficient and cost effective CSB system model that will select a CSP which will render optimal services in respect of energy, cost and QoS. To select the optimal service provider, we have formulated the problem as a Mixed-Integer Linear Programming (MILP). We have solved the MILP considering certain performance and QoS constraints. We have simulated our work and the result shows that our proposed model provides significant performance improvement compared to the existing state-of-the-art works.

1.5 Organization of the Report

The rest of the thesis report is organized as follows:

Chapter 2 presents a literature review of related problems and models. Chapter 3 describes our proposed system model. In Chapter 4 we showed our problem formulation in details and the performance evaluation of our proposed model is given in Chapter 5. Finally, the summed up conclusion of our work along with directions for future improvements are given in Chapter 6.

Chapter 2

Motivation and Background Studies

2.1 Introduction

With the vast growth of cloud computing, researchers started to work in this field. As a result, there are a large number of works for facilitating easy and efficient way for resource provisioning has been carried out. A number of researchers tried to improve the resource provisioning in cloud environment while others developed efficient cloud brokerage system that selects cloud services on behalf of the users. However, the state-of-the-art works did not consider all three factors; energy, cost and QoS, simultaneously.

2.2 Energy Efficient Cloud Computing Propositions

The researchers in [8] tried to calculate the energy consumptionof the datacenters and the energy needed to transport data to the datacenters using optimal network and virtualization in IP-over-WDM architecture. They proposed two models: Delay-Minimized Provisioning

(DeMiP), The Power-Minimized Provisioning (PoMiP). Satoh et al [9] mainly focused on reducing energy consumed by IT resources at multiple data centers and developed a cloud energy management system by using VM allocation tool and sensor management function. Goudarzi et al [10] proposed an approach that creates multiple VMs in a single server at the beginning. They used an optimization procedure of the VM controller in active servers to minimize energy cost. However, they did not take the Cost and QoS into consideration.

2.3 Cost efficiency in Cloud Computing

Researchers in [13] provide an MILP for cost efficient cloud resource allocation. They consider price, VM configuration, network latency and provisioning time of cloud providers as parameters to form the MILP where the price and VM configurations are real time value. They mainly tried to minimize the cost related to deployment of cloud infrastructure and proposed three policies among which the best is supposed to minimize the cost. However the cost per unit execution, cost per unit size of data storage etc. are not considered here. In [14], the authors presented a "Smart Cloud Broker" which compared and evaluated the performance of different CSPs and selected the best fit CSP. Centralized storage of data, reduced time and cost are supposed to be achieved through this approach with Infrastructure as a Service (IaaS) platform. However, they did not consider the specific cost of per unit execution and the size of data storage. Besides the other two important components to propose an efficient CBS system, energy and QoS are not consideredhere.

2.4 QoS aware models in Cloud Computing

Based on the QoS requirement analysis, a research shown in [11] proposed a CSB framework that can select the best cloud service provider (CSP). They defined an algorithm to get users QoS requirements based on two criteria: response time and throughput. Yet, the Cost of the CSP and required Energy, which are two vital objectives of an optimal CSB system, have not been taken care of in this work. Authors in [12] introduced a brokerage system that is supposed to ensure desired service level agreement (SLA) using their proposed framework which will maintain theQoS. In this study they mainly focus on two cloud services: Infrastructure as a Service (IaaS) and Network as a Service (NaaS). However, the detailed procedure by which the broker monitors and ensures SLAs in multiple clouds is not mentioned here.

2.5 State-of-the-Art Works

None of the existing works on the literature on cloud brokerage system have considered the energy, cost and qoS all together. Some research works are based on energy efficiency only while other tried to achieve QoS only. Moreover, in [8] ~ [10] the authors focused on reducing energy consumption based on either execution energy or transmission energy instead of taking care of both. On the other hand, the researchers of [11] and [12] tried to obtain QoS but they limited their study based on few components of QoS and for selective cloud services only. Similarly, the authors of [13] and [14] focused on creating a CSB that is cost effective. Yet they did not consider the cost of per unit execution and the size of data storage which are needed to estimate the total cost. In our work, we collaborate all of these three necessary parameters. We have also developed an optimal framework of cloud broker. We introduce a cloud brokerage system that can select the optimal cloud data center (CDC) among many CDCs based on these parameters. The broker selects the best green cloud based on the energy needed for transmission and execution of a job. On the other hand, the job execution rate, job size, per unit cost for execution and storage etc. are measured to obtain a cost effective CDC. Similarly, to assure the QoS, the proposed broker considers the reputation, latency, data loss, accuracy etc. of a CDC to choose the best. Therefore, the normalized form of these three parts gives the optimal green CDC that is cost and QoS aware. The detailed procedure of CDC selection is given in Chapter 4.

2.6 Summary

All the works in the area of selecting optimal CDC din not consider the energy, cost and QoS all together. Some research works are based on energy efficiency only, some on QoS only. Moreover, in [8] ~ [10] the authors focused on reducing energy.

Chapter 3

Proposed System Model

3.1 Introduction

In this chapter, a system model of cloud service brokerage system is presented. The model consists of three major components along with few more sub components. When a user sends a request for a CDC to perform a job, our model process the request and select the best fit data center to serve the client according to the requirements. It considers the energy consumption rate, cost and QoS of CDCs while selecting the best fit data center.

3.2Proposed System Model

The components of our proposed system model of cloud broker are shown in Fig. 1.2. The broker receives user requests, sorts them according to the priority of the requests and selects appropriate data center. The working modules of the broker model is described here in brief.

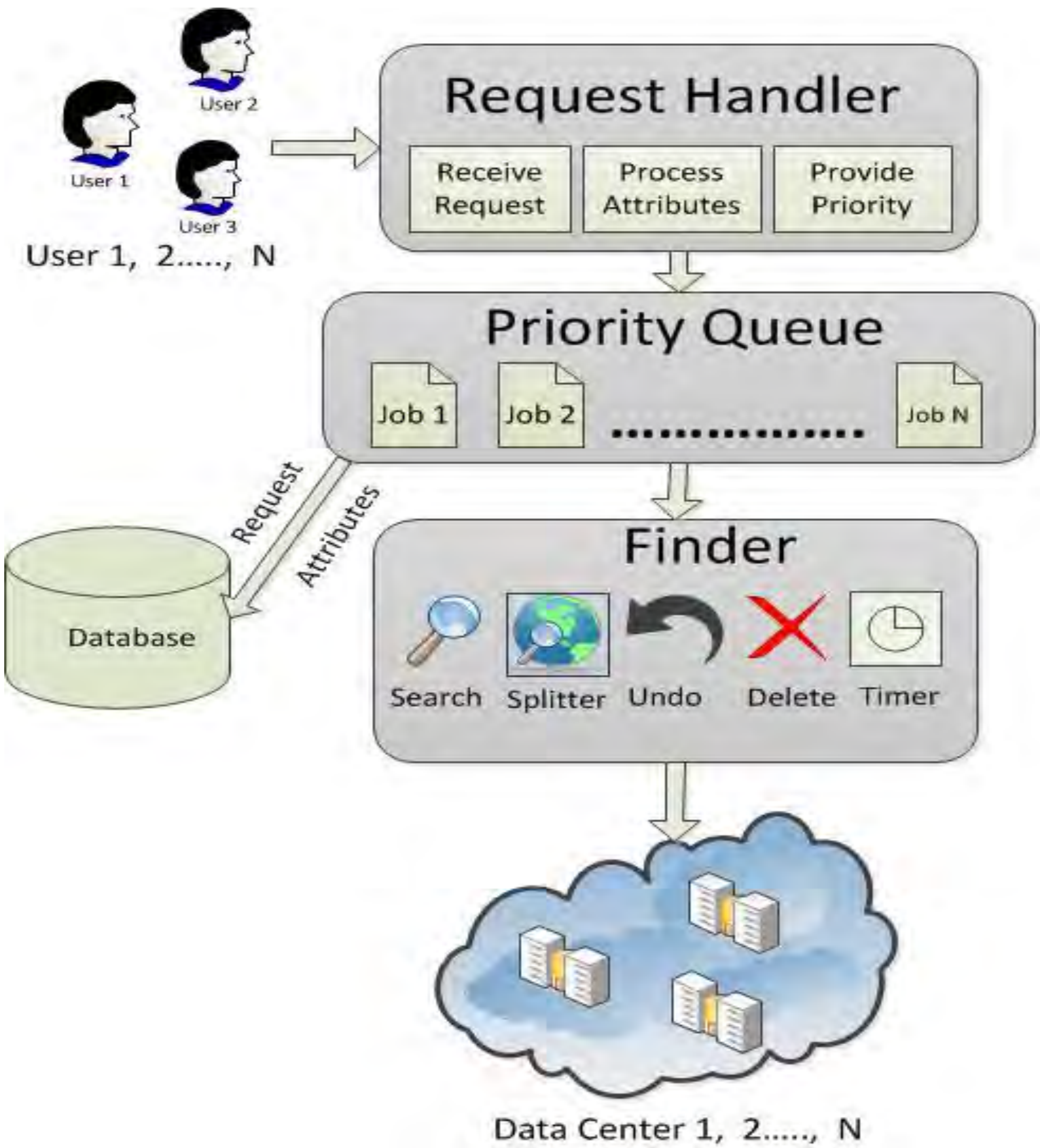


Figure 1.2System Model

3.3Request handler

One of the crucial components of the broker model is a request handler. It serves some basic functionalities. Firstly, it receives bulk of requests from users after they submit the requirements according to their demand. Also request handler process the attributes based on the user's request and provides priority by taking some factors into consideration. The basic functionalities done by the request handler is briefly discussed in the following section.

3.3.1Receive request

The broker accepts jobs from the user through the receive request module. User specifies different type of requirements while submitting requests. Sometimes the user requests for a data center that provides storage. Some users request for applications or platform for their works. A request can be made using a spreadsheet, document or form object for certain purpose. When the requests come to the request handler, the request handler receives the request with receive request function.

3.3.2 Process attributes

Upon receiving requests, all the demands according to the requirements of the users are processed into simple attributes. The request for platform service, application service, software service, database service, storage service etc. is processed and sorted with process attribute function.Attributes help a cloud service provider for prioritizing a process over another and selection of an appropriate CDC.

3.3.3Provide priority

Request handler prioritizes the submitted jobs based on some criteria. The criteria can be size of the job, region and profit etc. There might be other factors that could be taken into consideration. For example, if a user wants to have more quality of service and willing to pay higher for that, then more priority is added for QoS and less priority in cost is considered. Thus all the requests are prioritized based on the user request and user defined priority.

3.4 Priority Queue

The proposed model uses a database system to store user requests in a priority queue before processing. The priority queue is responsible for keeping highest priority job to the front. After sending each job to the finder component, priority queue automatically reshuffles and put the next highest priority job to the front. It takes job attributes into consideration and consider the attributes of the jobs in the queue to place a new job in the priority queue. Priority queue is also responsible to send the highest priority job to the finder for searching an appropriate data center for the request.

3.5 Finder

Among all the jobs in priority queue, the task from the front is selected and sent to finder module. The finder module is primarily responsible for the allocation of each task to suited data center. There are few basic functionalities of the finder module. They are categorized as search, region splitter, undo, delete and timer.

3.5.1 Search

While searching for the best cloud service provider, some parameters are taken into consideration. These are Energy, reputation, cost and delay. The select in of an optimal CDC for a job is elaborated in Chapter 4. The search functionality is responsible to this job. It is the most important part of our system model.

3.5.2 Region Splitter

For reducing the search space, dividing the data center based on the region might be considered as a vital task. Otherwise, search space would be a lot bigger. Even it can become troublesome if there are huge amount of data center. That is why our system model has a region splitter for splitting the data centers based on different regions.

3.5.3 Undo

The undo function is responsible for reversing the process in case of any hitch happened to the system. Besides, failure in choosing a proper data center can easily be turned. Sometimes it is not possible to find an appropriate data center for the requests of the user. The request can be re-processed totally by the undo function.

3.5.4 Delete

Delete function is used for deleting any unwanted search or search which is taking long to execute. Besides, it automatically erases the job which has been passed to the appropriate data center. For some request it is difficult to find an appropriate data center and for that reason the search function takes a longer time. Our proposed system can easily delete these types of unnecessary requests.

3.5.5 Timer

Timer function is often used for testing purpose. In order to debug the correct time to search and execute, cloud brokerage system can easily turn the timer on and check how much time has been elapsed. Timer can also be used for the purpose of choosing the appropriate data center for the future if any improvement needs to be performed.

3.6 Summary

The proposed system model consists of three main components which are dedicated to select the data center based on energy efficiency, cost effectiveness and QoS awareness. The three main part of the system model is the request handler, priority queue and the finder. Our proposed model also has a data center to store all the requests. By the request handler our system gives some priority to the requests considering some parameters. Then the priority queue prioritizes the requests and send these requests to the finder. Finder is responsible for finding an appropriate data center for the user's request by doing search operation. While searching our model will consider the QoS, energy and cost of the data centers. The finder also has option for undoing and deleting any request.

Chapter 4

Problem Formulation

4.1 Introduction

In this chapter, we have represented the cloud provider selection problem as an MILP optimization. A cloud broker is connected with at least one cloud service provider. The set of all cloud provider is given by N. Whenever a new job arrives, the cloud broker selects an appropriate data center for it. The size of a job is given by D. First we provide the calculation for cost, energy and QoS. Then, we develop the optimization problem and provide constraints for optimal solution. Notations used in this paper are summarized in table 01.

4.2 Calculation of Parameters

To obtain our desired broker, all of the three important components i.e. energy, quality of service and cost arecalculated separately. The breakdown calculation for each part is given in detail in the following section.

Variable Name	**Identifier**
Energy	E_i
Cost	C_i
Job Size	D
Data Transmission Rate for a CDC	γ_i
Energy Consumption Rate for Execution	R_c^i
Cost Per Unit of Execution	R_p^i
Cost Per Unit Size of Data	R_s^i
Energy Consumption Rate for Transmission	R_t
Job Execution Rate	R_e^i
Quality Observation	Q_O^{ij}
Quality for Accuracy	Q_A^{ij}
Quality for Data Loss	Q_D^{ij}
Quality for Reliability	Q_R^{ij}
Quality Expectation	Q_E^j
Maximum Delay	δ_i
Quality of Service (QOS)	Q^i
Rating for Accuracy from User	R_{AU}
Maximum Accuracy Rating	R_{AM}
Rating for Data Loss from User	R_{DU}
Maximum Data Loss Rating	R_{DM}
Objective Function	O_F

Table 1.1: List of Notations

4.2.1 Calculation of Parameters for Evaluating Energy Consumption

Energy consumption for a job will depend on two components; energy required for execution of a job and for transmission of the job to the data center.For calculating the transmission energyand execution energy we need to consider the size of the job.

4.2.1.1 Calculation for Transmission Energy

When the broker sends a request to the data-center some energy will need for sending the request. This required energy is defined as Transmission energy. Transmission energy can be achieved by dividing job size, D with data transmission rate, γ_i for data center i ∈ N and then multiplying the result with energy consumption rate R_t.

$$E_{trans}^{i} = \frac{D}{\gamma_i} \times R_t \text{.................(01)}$$

In the above equation, the ration of data size and data transmission rate gives the total time required to send the job to a data center. When it is multiplied by the energy consumption rate for transmission the total transmission energy for that particular job is achieved.

4.2.1.2 Calculation for Execution Energy

The energy needed by the data-centers to process the whole job is referred as execution energy. Execution energy can be estimated by the ratio of job size, D and the task execution rate, R_e^i multiplied by the energy consumption rate for execution, R_c^i for data center i ∈ N.

$$E_{exe}^{i} = \frac{D}{R_e^i} \times R_c^i \text{.............(02)}$$

Here in the equation the data size and execution rate for a job ration gives the time needed to process the job. After multiplying this with the energy consumption rate for execution, the total execution energy for that particular job can be achieved.

4.2.1.3 Calculation for Total Energy Consumption

For data center i∈ N the amount of energy that is required for transmitting a job is obtained by adding the transmission energy and the execution energy together.

$$E_i = E_{trans}^i + E_{exe}^i$$

$$\text{Or, } E_i = D\left(\frac{R_t}{\gamma_i} + \frac{R_c^i}{R_e^i}\right)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(03)$$

The above equation (eq. 01) presents the final equation to calculate the energy consumed by a data center to process a particular job. Here we also see if the data transmission rate, job execution rate, energy consumption rate for transmission and execution are constants for a particular data center, the total energy consumed is proportional to the data size.

4.2.2Calculation of Parameters for Cost

The price for unit resource of the CDCs connected with the CSB are different from one another. The total cost for the services provided to a job depends on the rate of job execution,R_e^i , size of job, D, cost per unit execution,R_p^i and the rate of cost for storage, R_s^i for data center i∈N.

4.2.2.1Calculation for Execution Cost

The cost needed by the data-centers to process the whole job is referred as execution cost. Execution cost can be estimated by the ratio of job size, D and the task execution rate, R_e^i multiplied by the cost per unit execution,$R_p^i$for data center i∈ N.

$$C_{exe}^i = \frac{D}{R_e^i} \times R_p^i$$

In the above equation, the ratio of data size and the job execution rate gives the total (???). When this is multiplied by cost per unit of execution the total cost to execute the job is achieved.

4.2.2.2Calculation for Storage Cost

For data center i∈ N, the cost for storage, R_s^i multiplied by data size, D gives the value of storage cost, C_{str}^i .

$$C_{\text{str}}^i = D \times R_s^i \text{.............(04)}$$

Here, the storage cost is actually the cost needed to store data inside a data center which is proportional to the cost of per unit data size for a fix length of data size.

4.2.2.3Calculation for Total Cost

The total cost is estimated by adding the execution cost with the storage cost. The mathematical expression for cost calculation is expressed as:

$$C_i = C_{exe}^i + C_{\text{str}}^i$$

$$\text{Or,} C_i = D\left(\frac{R_p^i}{R_e^i} + R_s^i\right) \ldots\ldots \text{(05)}$$

From eq. 02, it is clear that the cost necessary for processing a job is directly proportional to the data size when all other rates are constant.

4.2.3Calculation of Parameters for QoS

When the quality of service is calculated for a data center, several factors need to be considered. The observed quality for that data center for previous set of jobs, the delay that is made by the data center to serve a job in the previous times, all are considered to finally calculate the QoS fora data center.The Detailed formulation for calculating QoS is described below.

4.2.3.1Evaluation for Quality of Accuracy

To obtain the quality of accuracy, Q_A^{ij} by data center i∈N.after execution of job j, we get an accuracy from the user, R_{AU}in a scale of 1 to 10 and divide the value with maximum accuracy obtained from any job before, R_{AM}. As a result, Q_A^{ij} gets normalized having value in range from 0 to 1.

$$Q_A^{ij} = \frac{R_{AU}}{R_{AM}}\text{.............(06)}$$

Here, the accuracy is taken from the user for a particular data center and normalized the accuracy by dividing with the maximum accuracy of that particular data center.

4.2.3.2Evaluation for Quality of Reliability

Quality of reliability for job j is obtained from data center i∈ N denoted as Q_R^{ij} is measured from the weighted average of reliability of last j jobs. The equation for calculating reliability can be expressed as:

$$Q_R^{ij} = \frac{2}{j(j+1)}\{jQ_R^{ij} + (j-1)Q_R^{(ij-1)} + \ldots + Q_R^{(ij-j+1)}\}\text{......(07)}$$

4.2.3.3 Evaluation for Quality of Data Loss

Quality of data loss is also measured from the rating of user. A value from 1 to 0 is given by the user for a job j from the data center i∈N and is denoted by R_{DU}. The value is then divided by maximum data loss, R_{DM} to calculate the overall data loss, Q_D^{ij}.

$$Q_D^{ij} = \frac{R_{DU}}{R_{DM}} \text{.............(08)}$$

Here, the rating for data loss is taken from the user for a particular data center and normalized by dividing with the maximum data loss of that particular data center.

4.2.3.4 Evaluation for Latency

The total delay experienced by a user, δ_i is formed of delay for transmission, given by, $\frac{D}{\gamma_i}$ and delay for execution, given by $\frac{1}{R_e^i}$ for data center i∈ N.

$$\delta_i = max(1, D \times (\frac{1}{min\, \gamma_i} + \frac{1}{min R_e^i})) \text{ ....(09)}$$

In the delay calculation, the proposed broker considers the minimum of data transmission rate for a data center and when maximum delay for data transmission is achieved by dividing the data size with it. Similarly, the delay to execute a job can be calculated by the same procedure by taking the execution rate of job instead.

4.2.3.5Calculation for Quality of Observation

For calculating the quality observed, the performance of data center i∈ N for serving previous jobs is considered. Quality measured for data center i to serve the job j is Q_O^{ij}. This is measured by the addition of quality of accuracy Q_A^{ij} and reliability Q_R^{ij} of the data center. In addition, we add the delay Q_D^{ij} from the addition mentioned above for the job described as j.

$$Q_O^{ij} = Q_A^{ij} + Q_D^{ij} + Q_R^{ij} \text{.............(10)}$$

We will get 0 as the lowest value and 3 as the highest value from the equation described above.

4.2.3.6Calculation for Quality of Expectation

The Quality of expectation Q_E^j of each data center, we will take weight α into consideration and run exponentially weighted moving average (EWMA) process.

$$Q_E^j = \alpha Q_E^{(j-1)} + (1-\alpha) Q_O^{(j-1)} \text{.............(11)}$$

We put the maximum weight for the immediate job beforehand. Others get less priority considering the fact that we are putting most emphasis on recent one.

4.2.3.7Evaluating Quality of Service (QoS)

Quality of service $Q^i$for data center i∈ N is calculated by adding the expectation Q_E^j and the maximum delay δ_i. We divide the whole result by 4 for normalization.

$$Q^i = \frac{1}{4}\left(Q_E^j + \frac{1}{\delta_i}\right) \text{.................(12)}$$

The Eq. 03 shows the calculation for quality of service which is mainly dependent on quality of expectation and the delay. To normalized the equation, the right hand side is divided by 4.

4.3 Objective Function

For optimization process the value of Quality of service Q^i needs to be increased. In addition to that, we need lower energy and cost as well. For obtaining the objective function, we have to normalize the energy and cost factors. Normalization is done by dividing each factor with their highest possible values. Finally, the objective function is calculated by deducting the normalized energy and cost factors from the Quality of service Q^iConsidering each data centers we are multiplying each factors with selection variable, $x_i$and their relative weightsβ_1 , β_2 , β_3Data center which provides the maximum value, shall be selected for our purpose and creates the objective function O_F.

$$O_F = \max(\beta_1 Q^i x_i - \beta_2 \frac{E_i}{E_{max}} x_i - \beta_3 \frac{C_i}{C_{imax}} x_i) \dots\dots\dots.(13)$$

Where,

$$\beta_1 + \beta_2 + \beta_3 = 1 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots....(14)$$

Here exponentially weighted moving average (EWMA) process is used and the weight is added based on the priority. To get the normalized equation, the summation of weights, β_1 , β_2 , $\beta_3$need to be 1.

4.3.1 Constraints

4.3.1.1Delay Constraint

In our case, no data center should be exposed to such delay which exceeds the tolerable delay where d_t is the maximum tolerable delay provided by the user.

$$d_t \geq \delta_i, \quad \forall_i \in N \dots\dots\dots\dots\dots\dots\dots....(15)$$

If the delay for a data center exceed user provided tolerable delay, that data center will not get selected for that particular job so that least latency will be ensured.

4.3.1.2Single Collaboration

A job will be allowed to be executed simultaneously by a single CDC. No copy of a job is allowed to run simultaneously in different CDCs.

$$\sum x_i = 1, \quad \forall_i \in N \dots\dots\dots\dots\dots\dots\dots....(16)$$

4.3.1.3Fractionalization

Each data center does a complete job rather than serving a fraction of a job. In this case, the one full job will be completed by at most one data center.

$$x_i \in \{0,1\},\ \forall_i \in N \text{.........................}(17)$$

4.3.1.4Cost Reduction Constraint

The maximum cost user willing to pay must not exceed the total cost where C_t is the maximum expectable cost by the user.

$$C_i \leq C_t \text{..................................}(18)$$

4.3.1.5Normalized Quality

All the Quality factors which value will be in the range between 0-1.

$$Q_A^i \in [0,1] \text{.......................}(19)$$

$$Q_D^i \in [0,1] \text{.......................}(20)$$

$$Q_R^i \in [0,1] \text{.......................}(21)$$

4.3.1.6Degree of Expectation

Quality of expectation of all selected data center must maintain a minimum quality and can have a maximum value of 3.

$$Q_E^U \leq Q_E^i \leq 3\ldots\ldots\ldots\ldots\ldots\ldots\ldots(22)$$

Quality of expectation consists of three parameters: accuracy, data loss and reliability. So the maximum value of quality of expectation can be 3 and it must fulfil the user requirement of quality, Q_E^U.

Chapter 5

Performance Evaluation

5.1 Introduction

In this chapter, we have represented the cloud provider selection problem as an MILP optimization. A cloud broker is connected with at least one cloud service provider. The set of all cloud provider is given by N. Whenever a new job arrives, the cloud broker selects an appropriate data center for it. The size of a job is given by D. First we provide the calculation for cost, energy and QoS. Then, we develop the optimization problem and provide constraints for optimal solution. Notations used in this paper are summarized in table 01.

5.2 Simulation Environment Setup

MATLAB optimization toolbox provides some function to minimize or maximize the objective function while it also considers some of the constraints. The tool box has the facility to solve the linear programming, mixed-integer linear programming (MILP), quadratic programming, non-linear optimization and also non-linear least squares. For our simulation we solve mixed-integer linear programming.

To evaluate the performance of our proposed system we have considered five data centers with different capacity, cost and energy consumption rate. The specific volume of different parameters used in performance evaluation are summarized in the following table.

Parameters	**Value**
Number of Data Centers	5
Alpha	0.6
B1	0.4
B2	0.2
B3	0.4
Data Size	Uniform (1Gb, 2.5 Gb)
Data Rate	Uniform (3,10)
Data Execution Rate	Uniform (1,3)

Table1.2: Evaluation Parameters

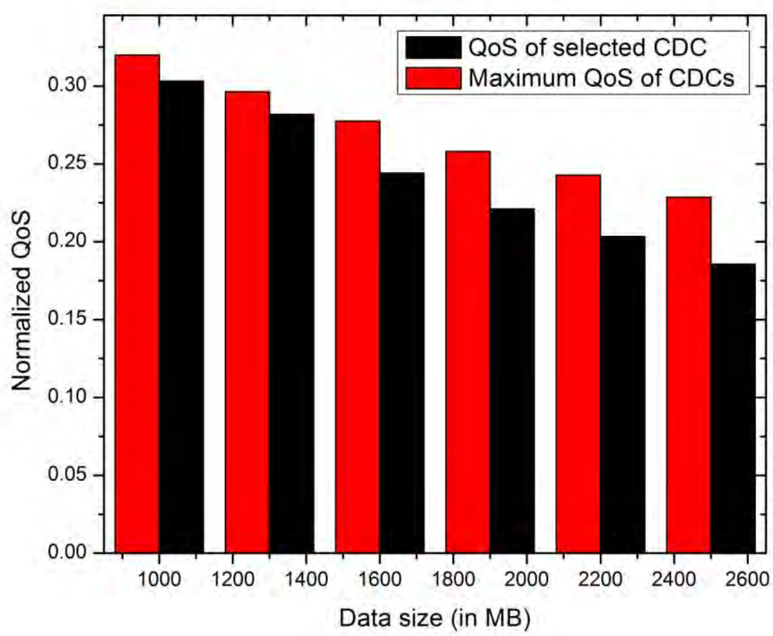
Considering these parameters our experiments show that the proposed model rightly selects the optimal valued data center for each tasks provided by the broker. In the time of experiment, we consider various data size to see our expected result is correct or not. There is significant amount of effects over QOS, energy and costs when data size varies. From our experiment it is clearly observed that while data size increases the QOS decreases, energy and costs increases.

5.3Simulation Result

From the simulation, we have found that by using our proposed broker model a significant performance improvement is being achieved. The comparative study using graph is discussed below. The experiments are conducted several times and the obtained results are then averaged to get the final result. The results calculated can help to avoid the random skew because of the poor selection of the input data space will follow the real trend of data and will better represent the actual system environment.

5.3.1Impacts on QoS with respect to Data Size

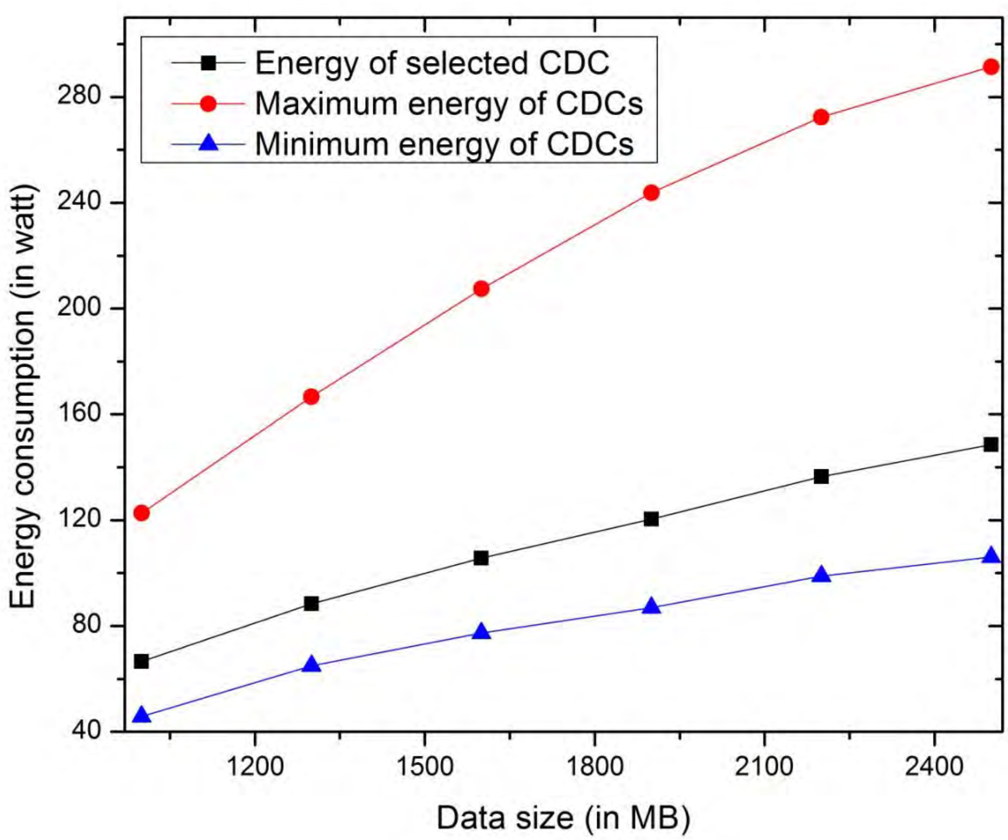
Graph 01 shows the relation between QOS and the data size. The graph shows that by the increment of data size QOS decreases but our simulation does not take the best QOS because weconsider few constraints and use optimal equation for selecting the data center.



Graph 01: Data Size Vs QoS

5.3.2Impacts on Energy with respect to Data Size

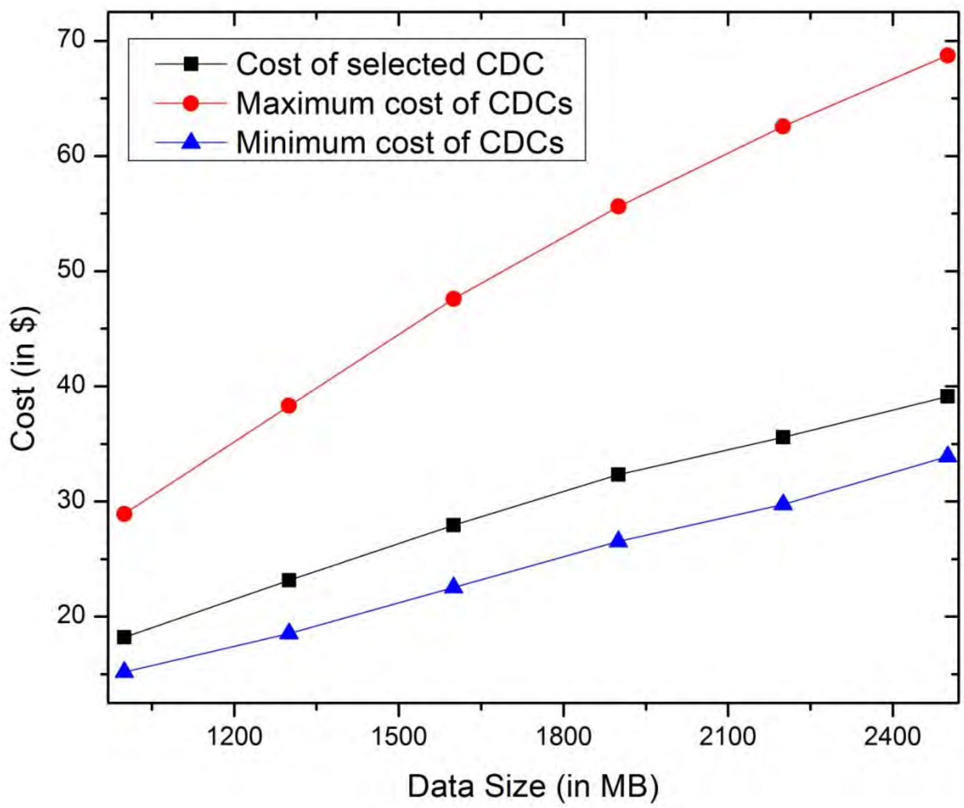
From graph02 it is clearly observed that by the increment of data size, energy decreases. Another important observation of our solution is the broker will select energy which are nearly to the lowest energy. To find the optimal energy we use the proposed optimal equation with someconstraints. In the graph three lines (highest, lowest and actual) are showed to represent energy for different data sizes.



Graph 02: Data Size Vs Energy Consumption

5.3.3 Impacts on Cost with respect to Data Size

Similarly, for graph03 it is observed that if the data size decreases cost of the task is increased. That means the higher the data size the lower cost will be obtained. From our simulation we found that the broker will select the cost which is nearly to the lowest cost. For our optimization we also consider energy and QoS of the data centers. For that particular reason it is not possible for us to select exactly the lowest cost for different data centers. In this graph we also showed three lines (highest, lowest and actual) to represent cost for different data sizes.



Graph03: Data Size Vs Cost

Chapter 6

Discussion and Conclusion

6.1 Summary of Research

In this work we have presented a Cloud Service Brokerage (CSB) system which is energy efficient, cost effective and guarantees the Quality of Service (QoS) to achieve an optimized Cloud Agent. We have simulated our proposed model. The result gives an optimized solution for the given task compared to the state-of-the-art works. In the system model section we proposed our own model by which a request can be send to the proper data center. In our system model broker will receive the request by a request handler then by priority queue the broker model will serve the requests. For problem formulation part we solve our problem as an MILP optimization. In this part our job is to calculate energy, cost and the QoS of the data center. We also provide some constraints for our optimal solution. For getting the energy we consider both the transmission energy and the execution energy. When we calculate the cost we have considered the execution cost and the storage cost. For measuring the quality of service we considered two main parts- quality of expectation and also the delay. Quality of observation obtained from accuracy, data loss and reliability. After finding the main three parameters we developed our objective function. For optimization process we need higher QoS, lower energy and the cost. We also developed delay, single collaboration, fractionalization and cost reduction constraints to simulate our result. The performance evaluation part is done in the Matlab optimization tool. To evaluate the performance of our proposed system we have considered five data centers with different capacity, cost and energy consumption rate. From our result we observed that the proposed model selects the optimal valued data center for each task. We showed three graphs which showed the relation of QoS, energy and cost for the various data size. If the data size increases QoS decreases. On the other hand energy consumption and the cost will increase by the increment of the data size.

6.2 Discussion&Future Works

Here we have focused on data center selection process only. Besides, we are not considering multiple collaboration. In future we want to implement this model with multi-cloud system and to create an actual cloud brokerage system.

Bibliography

1. Google App Engine, "https://console.cloud.google.com/projectselector/app-engine?src=ac," accessed on 5th August, 2016

2. Microsoft Azure Services Platform, "https://azure.microsoft.com/en-us/overview/what-is-azure/," accessed on 5th August, 2016

3. Amazon Elastic Compute Cloud, "https://aws.amazon.com/ec2/," accessed on 5th August, 2016

4. A. K. Das, T. Adhikary, M. A. Razzaque and C. S. Hong, "An intelligent approach for virtual machine and QoS provisioning in cloud computing," The International Conference on Information Networking 2013 (ICOIN), Bangkok, 2013, pp. 462-467.

5. T. Zain, M. Aslam, M. R. Imran and A. M. Martinez-Enriquez, "Cloud service recommender system using clustering," Electrical Engineering, Computing Science and Automatic Control (CSRS), 2014 11th International Conference on, Campeche, 2014, pp. 1-6.

6. T. Adhikary, A. K. Das, M. A. Razzaque and A. M. J. Sarkar, "Energy-Efficient Scheduling Algorithms for Data Center Resources in Cloud Computing," High Performance Computing and Communications and 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC), 2013 IEEE 10th International Conference on, Zhangjiajie, 2013, pp. 1715-1720.

7. F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, "NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and Technology," Cloud Comput. Program, Inf. Technol. Lab., 2011.

8. B. Kantarci and H. T. Mouftah, "Optimal Reconfiguration of the Cloud Network for Maximum Energy Savings," 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), IEEE Computer Society, Washington, DC, USA, pp. 835-840.

9. F. Satoh, H. Yanagisawa, H. Takahashi, and T. Kushida, "Total Energy Management System for Cloud Computing," IEEE International Conference on Cloud Engineering (IC2E), 2013, pp. 233-240.

10. H. Goudarzi and M. Pedram, "Energy-Efficient Virtual Machine Replication and Placement in a Cloud Computing System," IEEE Fifth International Conference on Cloud Computing, 2012, pp. 750-757.

11. M. Usha, J. Akilandeswari and A. S. S. Fiaz, "An Efficient QoS Framework for Cloud Brokerage Services," Cloud and Services Computing (ISCOS), 2012 International Symposium on, Mangalore, 2012, pp. 76-79.

12. M. Hamze, N. Mbarek and O. Togni, "Autonomic Brokerage Service for an End-to-End Cloud Networking Service Level Agreement," Network Cloud Computing and Applications (NCCA), 2014 IEEE 3rd Symposium on, Rome, 2014, pp. 54-61.

13. F. Díaz-Sánchez, S. Al Zahr and M. Gagnaire, "An Exact Placement Approach for Optimizing Cost and Recovery Time under Faulty Multi-cloud Environments," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 138-143.

14. M. BaruwalChhetri, S. Chichin, Q. Bao Vo and R. Kowalczyk, "Smart Cloud Broker: Finding your home in the clouds," Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on, Silicon Valley, CA, 2013, pp. 698-701.

List of Acronyms

- CSB- Cloud Service Brokerage
- CSP- Cloud Service Provider
- CSC- Cloud Service Consumer
- QoS- Quality of Service
- SLA- Service Level Agreement
- MILP- Mixed Integer Linear Program
- PaaS- Platform as a Service
- SaaS- Software as a Service
- IaaS- Infrastructure as a Service
- CA- Cloud Agent
- CDC- Cloud Data Center