Biomedical Research 2016; Special Issue: S392-S399

ISSN 0970-938X
www.biomedres.info

Online vaccines and immunizations service based on resource management techniques in cloud computing.

Tamil Vizhi[1]*, Parvatha Varthini B[2]

[1]Faculty of Computing, Sathyabama University, Chennai, Tamil Nadu, India

[2]St. Joseph's College of Engineering, Chennai, Tamil Nadu, India

Abstract

Online Vaccines and Immunizations service use for all kind of peoples through application software (service), Short Messaging Service (SMS), and Email Notifications. This research work working based on Resource Management Techniques in Cloud computing. Cloud computing is sharing the cooperative resources and distributed services in a large-scale computing network, in secure, seamless and transparent manner throughout the world. Existing Cloud technologies not have efficient resource allocation, scheduling only use the user run-time estimations with a limited number of Quality of Service (QoS) and hence not yielding accurate resource matching. This proposed work develops a Cloud portal (online service) for Online Vaccines and Immunizations service. It provide optimal Resource Management Technique (RMT) to discover the best fault tolerant resource and to enable a balanced scheduling based on system generated predictions for efficient resource match making. In this service, all people's health details are stored in the cloud with protected and authorized manner. Intelligent Broker based resource management techniques are proposed as an innovative idea in this research work. Here Broker (experts) gives an important and trustworthy suggestion for each job (health case) for requester (patients). Intelligent Broker based Resource Management Technique provides an enhanced Health service for peoples request from anywhere and anytime.

Keywords: Cloud portal, Resource management technique, Cloud scheduling, Cloud resource prediction, Intelligent broker, Cloud balancing.

Accepted on November 18, 2016

Introduction

Cloud computing provides cost effective utilization of available resources that are managed by different organizations and distributed anywhere in the world, to solve complex problems in quick time using parallel computing. The critical issue for achieving high performance in Cloud Computing is resource co-allocation of different distributed resources. Cloud resource co-allocation is an advanced technique used to execute multiple site jobs in large-scale computing environments. It generally uses the basic information derived from user run time estimations. Resource co-allocation includes following enhanced features like advance reservations, priority, fault detection, load balancing and etc. The worldwide Cloud network connects multiple resources through high speed networks. The resources are available anywhere in the world and it is possible to access many resources at the same time. Cloud Computing actually shares its resources and services in a large-scale computing organization. Brokers, a kind of super-independent agents hands the resource request over to the multi agent who in turn optimizes the distributed services. This work follows Data sets of World Health Organization (WHO) and Centers for Disease Control and Prevention (CDC). Vaccination is the more helpful tool to avoid disease and severe outcomes caused by infection viruses. The dominance of Cloud computing is spread over the world for computing solutions to the problems through Resource Management Technique (RMT). Cloud resource prediction pattern is developed for finding the best resource across the Cloud for any task submitted by a user. Cloud Resource Prediction pattern is develops a prediction model to evaluate the capability of each Cloud resource. The cloud selection is done based on the n number of packets to be stored in n number of cloudlets [1].

The proposed Cloud resource prediction algorithm requires less time and less number of iteration process compare with existing techniques for finding a best resource in a Cloud. The proposed system develops an elastic Cloud Balancing and Job Shop Scheduling [2]. This proposed Cloud resource matchmaking algorithm is based on evaluation and classification technique. It evaluates each resource's quality of service parameters to predict and balance the workload.

Material and Methods

The role of proposed Resource Management Technique (RMT) is resource selection, resource match making, schedule, resource allocation, agent based Cloud service. The RMT receives and resolves the request from the Cloud end-user. The existing load balancing is time consuming [3].

Objective

- To design a Cloud portal called as for Online Vaccines and Immunizations service. It provide extended resource management technique to enable efficient resource
- To develop an optimization algorithm for selecting the best Cloud resource for submitting job
- To develop elastic scheduling algorithm based on system generated prediction with resource matchmaking
- To apply this proposed Cloud scheduling and matchmaking in online vaccines and immunizations service
- Any recent vaccines introduced immediately all peoples get the notification through SMS and E-mail.

The proposed work improves the resource management technique operations in Cloud Computing to enhance the resource co-allocation. In the proposed model, the end users do not need to have the Cloud and resource knowledge, while submitting a job to the portal. The proposed work develops following improved resource optimization technique operations.

- Predicting the best Cloud resource for submitted job with fault
- Balanced resource matchmaking based on various QoS
- Elastic scheduling based on system generated prediction

Cloud manager and remote executor service architecture

Prediction of the best Cloud node for the submitted task is the major problem in Cloud computing. A novel optimization algorithm, namely, CRP is developed to predict the best Cloud node for the submitted task in Cloud computing. A novel Cloud Resource Prediction (CRP) pattern approach is developed to find the best resource across the Cloud. The CRP pattern uses the historical data and CRP model, to evaluate the prediction for each Resource. The historical data contains continuously updated information about any of the Cloud nodes and is more important to do prediction. CRP model selects the most available recent resource by checking the resource information using the queries to the historical database and prediction. System generated predictions are better parameters than user run time estimates, for resource co-allocation scheduling. This idea use to find best service for health problem by peoples. Based on peoples request about health problem, this service finds the best health solution from the cloud database in Figure 1.

This work carry out from Local Area Network of Sathyabama University's hospital Computer Lab is simulated as Cloud Network for analysing the performance of CRP algorithm for best resource. CLOUDSIM simulate to study the performance of the proposed CRP algorithm. On LAN, any node can give health problem request as user; Computing nodes are treated as resources. One node is selected as a prediction node with health detail database.

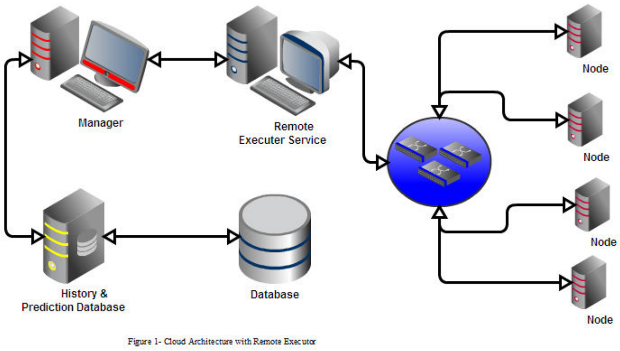

Figure 1. Cloud architecture with remote executor.

CRP algorithm uses the shortest path algorithm (Floyd's) to detect the best node based on QoS parameters [4], like, high-speed, previous history, customer‘s feedback, and distance. The CRP evaluates the distance of all resources for the task submitted. Suitable resources to the task submitted, are then found among all the resources based on using the minimal cost. Distance between resource and user job is considered as 'x' parameter and QoS parameter of resource as 'y' parameter. They are then used to find the best Cloud nodes from available resources. In Figure 2, the QoS rate y is calculated based on the listed attributes. The best node is a node with minimal distance x and high QoS rate y. The proposed CRP finds the available resources from the 150 number of active resources, and then finds the best resource.

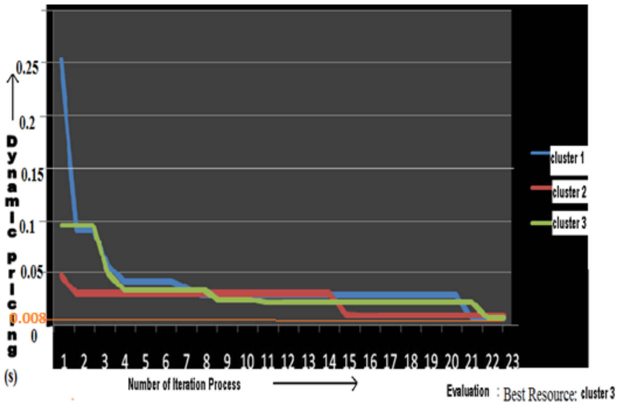

Figure 2. Performance analysis of CRP.

Balanced resource matchmaking and elastic scheduling

The proposed RMT does elastic scheduling with balanced resource matchmaking. The main role of a Cloud scheduler is resource matchmaking, job balancing and resource balancing. A Cloud scheduler does resource reservation, service-level agreement validation, enforcement, and job/resource policy management, to enforce the best turnaround times within the allowable budget constraints. The proposed RMT's Cloud Resource matchmaking algorithm matches suitable resource for the job submitted. It is carried out in two phases namely, Resource matchmaking based on Evaluation and Classification (EC) Technique and Elastic Scheduling based on system generated prediction. The proposed technique increases the quality of Cloud balancing, and detects the fault tolerance Cloud resource from the available resources.

Evaluation and classification consists of Cloud resource evaluation and classification, Job evaluation and classification and Cloud evaluation system by the system generated prediction scheduler. The QoS parameters considered for evaluation are dynamic costing, distance, bandwidth, virtual machine access ability, previous history, power capability, Cloud network support, cooperation facility, high-speed, policy of resource, customers previous feedback, CPU speed, bandwidth, and latency. Cloud resource's QoS parameters are stored in the master database of the resource hash table. After Cloud resource's QoS parameters evaluation, ranking is used to classify the Cloud resources based on the above said QoS parameters. The Resource specification details are stored in the resource matchmaker's database.

Jobs are classified with respect to the following factors:

Job execution time-Jobs with less execution time to be processed first. Priority value of the job-high priority job attains more preference. Situation adaptive of job processing: Run the task at a particular time when competitors would not monitor the output. Cloud evaluation collects the list of available resources prepared and evaluated by the system-generated prediction model based on resource's Quality of Service parameters, and balances the workload of each available best resource.

Cloud matching processes and resources are ranked in the following way. It provides a high priority rating for a local scheduling process to select local resources, because the local scheduler uses only a limited bandwidth and less transport time for completing the job. Cloud ranking provides a next priority rating for a Meta scheduling process to select the local and remote resources, because the Meta scheduler uses the advanced reservation based job computation. It provides a low priority rating for a remote scheduling process to select a remote resource because the remote scheduler takes more time for submitting and retrieving the job through a long distance network. Resource co-allocation and co-scheduling techniques are helpful to increase the priority value, compared to all other techniques in remote scheduling. A job is divided into multi tasks, and each task is evaluated for the matchmaking process.

Resource match making is then executed, using the topological sort technique [5]. Resources are placed in the host pool for the matchmaking and scheduling. If the resources are insufficient, the job is marked for the rescheduling process. Rescheduling is an unnecessary process from the angle of the worst resource selection. The resources can be accessed via the resource providers, simultaneously from the Cloud portal for a job. A Meta scheduler books the resources in advance. It assigns suitable resources (Matching) as well as the optimal order of task execution (Scheduling). The proposed system develops an elastic scheduling with balanced Cloud resource matchmaking. The proposed Cloud resource matchmaking algorithm is based on the Evaluation and Classification (EC) technique. It uses system generated predictions, instead of user run time estimates, to reduce the rescheduling cost [6]. It evaluates each resource's quality of service parameters to predict and balance the workload. It uses the topological sorting technique for ordering the vertices in a Directed Acyclic Graph (DAG), by considering the Cloud resources as vertices, and the connections between the resources as edges. This scheduler uses fuzzy optimization for dealing with the uncertainties in the application demands. Its advantage is to get a quick response for resource allocation request and high accuracy in scheduling.

The proximity of the resource, demand value, job type, service type, and resource type from the central Cloud system is checked for high priority jobs. For low and medium priority jobs, only the proximity value and service type value are checked. In the case of a low priority job, assign resources at a fixed minimum proximity, for a medium priority job at a fixed average proximity value, and for a high priority job at the maximum proximity value.

The proposed elastic scheduling does scheduling based on time, cost, priority, user history, system generated prediction and energy consumption. Two factors, namely, Discovery service and Priority play a vital role here. Discovery service is a conjunctive approval reprocessing for the resource co-allocation system [7,8]. The Discovery service automatically communicates all the processes through the Secondary Decision Point (SDP). It is used to collect and store all authorized user's information. The own proxy is created, and used for resource allocation through discovery service. The utilization of resource is very important in the Cloud network world, to avoid non-utilization of resources.

The advantages of the proposed system are a quick response for Resource Co-Allocation request, high accuracy in scheduling, system generated predictions, reducing the rescheduling time, and number of schedulers and cost. The Cloud network is mapped to a DAG with resources as vertices. One by one all the jobs are allocated simultaneously in a Cloud network based on a topological sort.

Indegree 0 vertex=Available (non-allocated and quality) resource → (1)

Equation 1 finds the available resource that is indegree 0 vertex from the directed Cloud network. Finding the indegree 0 vertex with a feasible price for the resource requester is repeated until all the jobs are allocated with respect to priority. Matching is to be done between the job and the Cloud balanced resource, based on QoS parameters. The closest Cloud resources are the perfect nodes in the Cloud network. Mapping a Cloud network to a network graph, the resources are named as vertex and the distance as edges. The shortest path algorithm is used to predict the closer node for a task. The data structure used in Cloud resource matchmaking is given in Table 1.

Table 1. *Data structure used in Cloud resource match making.*

Variables	Description
sminmid	Minimum size vales for medium priority job
smaxmid	Maximum size values for medium priority job
LoC	Job location
Lminmid	Minimum LoC *vs.* Ss for medium priority job
Lmaxmid	Maxim=LoC values for !midi= priority job
smin	Minion=Size for high priority job
Lmin	LoC size yam for high priority job
proximity	Hop distance of the resource from the central system
P_{low}; P_{mid}; P_{max}	Low, medium, high priority

The steps of Cloud resource matchmaking algorithm is explained below

Step 1. Update the resource detail like Resource ID, name, proximity, the type of service provider.

Step 2. Identify the attributes of the submitted job, namely, job size, LoC etc.

Step 3. Calculate the priority value as follows.

If Job size>sminmid and<smaxmid and LoC>Lminmid and<Lmaxmid

The priority is assigned as medium.

Else If Job size >smin and LoC> Lmin

The priority is assigned as high.

Else

The priority is assigned as low.

Step 4. Jobs and resources are evaluated and ranked based on EC Technique.

Step 5. Match the high ranked resource to the high ranked job with high priority.

If priority is low and proximity p_{low} and

Job type and resource service type match preference of remote scheduler,

Resource is matched for low priority Job.

Else If priority is medium and proximity p_{mid} and

Job type and resource service type match preference of Meta scheduler,

Resource is matched for medium priority Job

Else If priority is high and proximity p_{max} and

Job type and resource type match preference of local scheduler,

Resource is matched for high priority Job.

Else

No match found message is shown.

The elastic scheduling with balanced Cloud resource match making is implemented to get a quick response for Resource Co-Allocation request and high accuracy in scheduling. It uses system generated predictions, to reduce the rescheduling cost. The balanced scheduling using system generated prediction is carried out in the following manner verify the reservations and execute sub-jobs on the computing nodes. The system collects information about jobs that have run in the past that is the system history or previous observation of job execution. This information is used to generate a prediction model about the runtimes of newly submitted jobs. The Cloud resources are denoted as vertices, and the edges are denoted as Cloud network connection between the resources. This scheduler uses fuzzy optimization for dealing with uncertainties in the application demands. The communication demands, host capacity, cost constraints are used for fuzzy optimization which can determine the starting time of the task and data transfer time.

Here, the graph's vertex is the computing nodes, and the edges are LAN connection wires. Any node can raise service request. Multiple numbers of other energetic computing nodes can act as resources. The resources register their details with the matchmaker, based on the authorizing ID, name, proximity, type. This input is mapped to Directed Acyclic graph (DAG) with nodes and edges. The edges are used to represent the processing demand of the i^{th} and j^{th} task. Using the graphic mode the graphs are constructed, and the host is numbered from 0 to m. This constitutes the available resource updating interface. The admin enters the information about a resource through this interface.

All Cloud nodes and mapping the tasks are Initialized. In graphs, the related nodes are constructed, using the CLOUDSIM tool. By extracting the node, the related attributes such as the edges, are connected. The graphs represent Cloud topology and the dependencies among the task. The host capacity and the available bandwidth of the nodes are estimated. The data transfer time and the processing time of the task are computed.

The time and Quality of Information (QoI) index for single instruction is evaluated. The QoS parameters considered for performance analysis of the proposed one with existing scheduling techniques are shown in Table 2.

Table 2. OoS parameters.

Scheduling techniques	PaS Parameters
Parallel job scheduling	Time and cost
Adaptive scheduling	Time, cost and dynamic upgrade
Session scheduling	Time, Cost, priority and service session
Enemy consumption scheduling	Time, Cost, Priority, dynamic upgrade and energy consumption
Proposed system gemmed prediction scheduling	Time, cost, priority, user history, system generated prediction and energy consumption

From Table 2, it is seen that the proposed scheduling algorithm supports more number of QoS parameters, such as, time, cost, priority, user history, system generated prediction, and Energy consumption. The proposed system generated prediction scheduling with the existing energy consumption scheduling system with respect to the number of jobs computed and time. In this experiment, the given job services are floating point arithmetic operations, task management for office, social networking of sharing, online contact lists, collaborating gaming etc. System generated predictions reduce the rescheduling time and cost. The proposed elastic scheduling based on system generated prediction shows an improvement ranges from 20% to 30% than the existing energy consumption algorithm in Table 3.

Table 3. Performance analysis of elastic Scheduling based on system generated prediction.

	Time taken for execution of job (sec)		Improvement in completion time (%)
Number of jobs	Proposed elastic scheduling based on system generated prediction	Existing energy consumption scheduling based on user run time estimation	
8	2	4	20%
9	3	5	20%
10	4	6	20%
11	5	8	30%
12	6	9	30%
13	7	10	30%

The performance of these RMT operations are analysed for another scenario a Cloud network with three different processors/Resources (R1, R2, R3) and Nine tasks (T0, T1 …., T8) with execution time for each task: T0=1 s, T1=5 s, T2=3 s, T3=4 s, T4=2 s, T5=8 s, T6=7 s, T7=6 s, T8=8 s.

Matrix representation of tasks execution is represented with respect to the resource matchmaking process. Resource matchmaking process use to predict the suitable resources for the tasks.

Using proposed Cloud resource matchmaking algorithm, tasks and resources are matched as Tasks T1, T4, T5 are executed in the resource R1, Tasks T2, T6, T7 are executed in the resource R2, Tasks T0, T3 and T8 are executed in the resource R3.

When Scheduling is done by existing FIFO algorithm with user run time estimation, the time needed for completion of all the tasks is 21 s. Based on FIFO scheduling tasks are executed in the following order R1, R2, R3 in Figure 3.

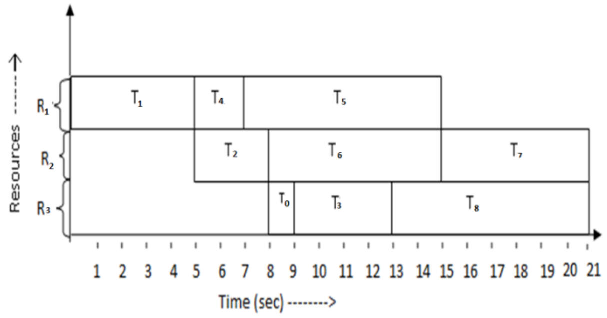

Figure 3. Scheduling by FIFO algorithm based on user run time estimations.

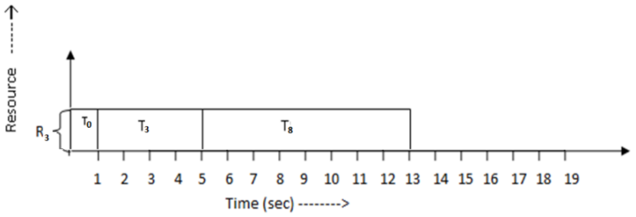

Figure 4. Elastic scheduling algorithm execution stage 1.

Table 4. Elastic scheduling algorithm execution stage 1.

Computing task	Computation time (sec)	Resource assigned
TO	I	R3
T3	4	R3
T8	S	R3

Min job scheduling algorithm applies for each resource. From the minimum job schedule, Tasks T4, T1, T5 are executed in the resource R1, Tasks T2, T7, T6 are executed in the resource R2, and Tasks T0, T3 and T8 are executed in the resource R3. Min job priority based matrix representation of tasks as follows.

$$Matrrix(M1) = \begin{pmatrix} T_0 & T_3 & T_8 \\ T_4 & T_1 & T_5 \\ T_2 & T_7 & T_6 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 8 \\ 2 & 5 & 8 \\ 3 & 6 & 7 \end{pmatrix}$$

From the minimum job schedule, Tasks T0, T3, T8 are executed in the resource R3 as shown in the Figure 4 and Table 4.

From the minimum job schedule [9], Tasks T4, T1, T5 are executed in the resource R1 as shown in the Figure 5 and Table 5.

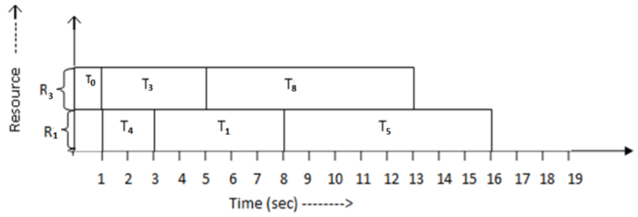

Figure 5. Elastic scheduling algorithm execution stage 2.

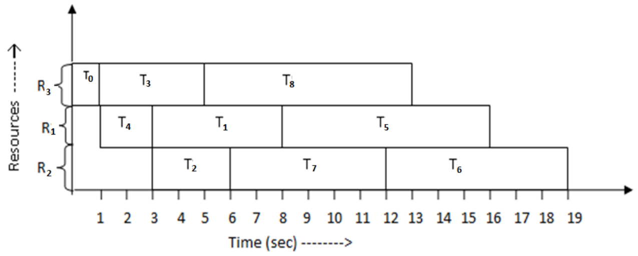

Figure 6. Elastic scheduling algorithm execution stage 3.

Table 5. Elastic scheduling algorithm execution stage 2.

Computing task	Computation time (sec)	Resource assigned
T4	2	RI
TI	5	RI
T5	8	RI

From the minimum job schedule, Tasks T2, T7, T6 are executed in the resource R2 with system generated prediction as shown in the Figure 6 and Table 6.

Based on processor remapping with help of flow shop scheduling tasks are executed in the following order R3, R1, R2 in Figure 6. When scheduling is done by proposed scheduling algorithm with system generated prediction parameter, the total time needed for completion of all the tasks is 19 s. The proposed elastic scheduling algorithm is use to get quick response for resource allocation request in Cloud scheduling. Resources can be filtered by the reservation manager and freely available matches can be considered for matchmaking. The design along with the application allocates resources based on system generated predictions based on priority. Evaluations are performed on the provided Cloud application. This result indicates the demand of the Cloud system for allocation of multiple resources to multiple clients without overlapping [10-12].

Load balancing based on greedy algorithm

The Cloud balancing problem is mapped to the Knapsack problem and solved using Greedy Algorithm. The data structure used for the proposed Load balancing based on Greedy Algorithm is given in Table 7.

$$\sum_{i=1}^{n} j_i \le M \to (2)$$

The load balancing based on Greedy Algorithm is described below:

Example, here Vaccines are considered as Cloud resources in the Cloud network. Intelligent broker based RMT maintains a shopping database containing valuable suggestions for every product based on following QoS parameters (Figure 7). Such as, available location of resource, knowledge up-gradation level, feedback of existing customers, and the expert's suggestion. Broker's every idea get it based on the buyer's requirement broker will suggest the product. Broker suggests a suitable product for user in Cloud based shopping network.

Table 6. Elastic scheduling algorithm execution stage 3.

Computing task	Computation time (sec)	Resource assigned
T2	3	R2
T7	6	R2
T6	7	R2

Table 7. Data structures used in load balancing based on Greedy algorithm.

Variables	Description
NR	Current job allocation resource
U	Dynamic capacity in Cloud
i	Loop index
N	Number of resources
M	Static capacity in Cloud
X []	Profit array of computation task
Ji	Jobs with identity

Table 8. Time complexity of intelligent broker based IRMTO.

Category	Time (ms) using IRMTO Time (ms) using RAT		
	Without intelligent broker	With intelligent broker	Improvement

Resource selection	8	6	20%
Resource=clinking	13	10	30%
Resource scheduling		5	20%
Load baffling	11		40%

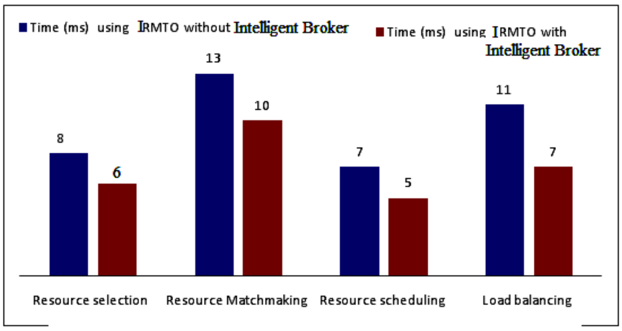

Figure 7. Performance analysis of intelligent broker based RMT.

Conclusion

A Cloud portal providing resource management technique to enable efficient resource allocation for users with no Cloud knowledge is developed. The objective of the research work to develop a single Cloud RMT framework has been achieved in the following manner. The proposed RMT does Cloud computing functions like resource selection, resource matchmaking, resource scheduling, resources access, resource co-allocation, and Cloud balancing in an efficient manner. A novel optimization algorithm, namely, CRP is developed to predict the best Cloud node for the task in Cloud computing. The best Cloud Resource is predicted using CRP optimization algorithm for the task in Cloud computing with the help of Cloud Resource Prediction Pattern (CRP). This CRP algorithm is also enhanced using genetic theory to find a best resource with fault tolerant across a hybrid network. Proposed RMT matches suitable resource for the job submitted using Evaluation and Classification (EC) technique and then schedules using system generated prediction model. Its performance is compared with existing algorithm using user run time estimation. Total time needed for completion of nine tasks with three resources by proposed elastic scheduling using generated prediction is 10% less than FIFO Algorithm with user run time estimation. In a Cloud scenario of 150 resources and 13 jobs, the total time taken for executing them, using the proposed resource matchmaking and scheduling based on system generated prediction is 20% less than the existing energy consumption algorithm.

The resource intelligent broker has been deployed between the service requester and the service provider to provide intelligent services for the service requester. This Intelligent broker based RMT is applied for online shopping Cloud portal and analysed. The Resource selection process takes 6 ms with the help of the intelligent broker in RMT. Without the intelligent broker's help, the resource selection process takes 8 ms in RMT. The Reason for the lesser time complexity is the resource search, based on hash search. The Resource matchmaking process takes 10 ms with the help of the intelligent broker in RMT. Without the intelligent broker's help the resource matchmaking process takes 13 ms in RMT. The Reason for the lesser time complexity is the resource matchmaking, based on the big data handling capability of the intelligent broker in RMT. The resource scheduling process takes 5 ms with the help of the intelligent broker in RMT. Without the intelligent broker's help, the resource scheduling process takes 7 ms in RMT. The Reason for the lesser time complexity is the resource scheduling, based on the trusty suggestion of the requester. The resource load balancing process takes 7 ms with the help of intelligent broker in RMT. Without intelligent broker's help, the resource load balancing process takes 11 ms in RMT (Figure 7). The Reason for the lesser time complexity is the resource load balancing based on the broker's historical discovery service. In intelligent broker based RMT, efficiency in job execution time improves from 20% to 40%. This research work proposes the use of Resource Management Technique (RMT). RMT is used to reduce the cost, save high-level energy, increase the resource availability, avoid failure, increase user satisfaction, and provide easy handling for all levels of users, high QoS, reduce the space and time complexity in the Cloud network (Table 8).

Future Enhancements

This work can be extended for future research in the following directions,

- The classification and evaluation of Cloud resources and Prediction can be extended for large user on demand Cloud computing.
- Providing high security backup solutions is also an important issue in system generated prediction model. Intermediate language can be used to create an application for authentication for the legacy machine database.
- Proposed RMT can be enhanced to provide interoperability between Cloud models
- Increase the QoS parameters like virtualization and negotiation.

Acknowledgments

The authors would like to thank Sathyabama University (India) for providing us with various resources and an unconditional support for carrying out this study.

References

1. Tamilvizhi T, Parvatha VB, Vinothini P. An innovative mechanism for proactive fault tolerance in cloud computing. international conference on mathematical sciences. Proc Elsevier 2014: 618-622.
2. Tamilvizhi T, Parvatha VB, Surendran R. An improved solution for resource management based on elastic cloud balancing and job shop scheduling. ARPN J Eng Appl Sci 2015; 10: 8205-8210.

3. Andres GG, Ignacio BE, Vicente HG. SLA-driven dynamic cloud resource management. Fut Gen Comp Sys 2014; 31: 1-11.
4. Dimitrios K, Jose MAR, Iraklis P. Semantic-based QoS management in cloud systems: Current status and future challenges. Fut Gen Comp Sys 2014; 32: 307-323.
5. Haopeng C, Wenyun D, Wenting W, Xi C, Yisheng W. A Cloud-federation-oriented mechanism of computing resource management. Int J Cloud Comp 2014; 2: 44-58.
6. Harpreet K, Amritpal K. A Survey on fault tolerance techniques in cloud computing. Int J Sci Eng Technol 2015; 3: 411-415.
7. Jiunn-Woei L, David CY, Yen TW. An exploratory study to understand the critical factors affecting the decision to adopt cloud computing in Taiwan hospital. Int J Inform Manag 2014; 34: 28-36.
8. Marc EF. Scheduling highly available applications on cloud environments. Fut Gen Comp Sys 2014; 32: 138-153.
9. Nabil S. Making use of cloud computing for healthcare provision: Opportunities and challenges. Int J Inform Manag 2014; 34: 177-184.
10. Kaur PD, Chana I. Cloud based intelligent system for delivering health care as a service. Comput Methods Programs Biomed 2014; 113: 346-359.
11. Surendran R, Parvatha VB. Inject an elastic grid computing techniques to optimal resource management technique operations. J Comp Sci 2013; 9: 1051-1060.
12. Ziqian D, Ning L, Roberto RC. Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers. J Cloud Comp Adv Sys Appl Springer J 2015: 1-14.

*Correspondence to

Tamil Vizhi

Faculty of Computing

Sathyabama University

Tamil Nadu

India