Optimized Resource-to-Task Scheduler for Cloud Service

R. A. Karthika and A. Shajin Nargunam

Computer Science & Engineering Department, Noorul Islam University, Tamilnadu, India-629180

Abstract

A key concern in cloud computing setting is to capitalize on turnover by accommodating all arriving needs and to diminish bad consequences for cloud providers. Attaining these purposes extremely depends on optimal usage of accessible resources in datacenters. To diminish the power and time consumption in cloud computing environment, the previous work evaluate the process of identifying level of each server peer consuming power and task execution time to perform Web requests from client peers. This work explains inadequacies caused by the absence of resource organization method and propose a geometric representation using Scheduler-based Optimal Resource Allocation (S-ORA). The proposed S-ORA scheme investigates the suitability of commercial cloud service using Amazon EC2 to hierarchical data exchange between multiple cloud instances with minimal resource utilization. The experimental evaluation shows that proposed performance of cloud computing services including Amazon EC2, with suitable management of resources result in an increment of profit by reducing rejected requests of the cloud.

Keywords Cloud instance, distributed networks, resource utilization, rejected requests.

1 Introduction

With the fast development of computing, storage space and system tools, dispersed calculating paradigms have undergone intense amends in the precedent decade. Such a mutiny allows request service providers (ASPs) to organize treacle or even pet scale requests for manufacture reason. For instance, employing cluster or grid calculating amenities, scientists are capable to sprint large-scale climate anticipate representations with a huge quantity of data engendered from the most superior scientific tools, and regularly issue the estimate outcome to the universal public. Nevertheless, owing to the vast computational intricacy and elevated quantity of data, such requests need demanding resource practice and place a serious financial weight on the association who systems, upholds and functions these possessions. In the kingdom of large-scale dispersed computing, the rising cloud computing notion, with its almost infifinite possessions and suppleness, is being accepted with the assurance to release ASPs from the upfront system and preservation cost of the communications.

As an efficient means of giving subtracting resources in a shape of usefulness, cloud computing has lately concerned a considerable quantity of concentration for both manufacturing and academic world. The example move to cloud computing is determined by strapping command, particularly from endeavors, to progress the general efficiency by means of and running computing possessions. Cloud repair providers employ datacenters to stipulation a common pool of calculation, storage space and bandwidth possessions, to be exercised by requests when the requirement arises. As resources at datacenters are common by using virtualization, requests are permissible to statistically complex such possessions in the structure of practical machines.

Completion of cloud arrangement has been probable with release source and profitable software letters. In the middle of them OpenNebula, Eucalyptus, and OpenQRM are finest option for making IaaS cloud communications as a release basis service. Bringing services in IaaS stage has been probable by means of datacenters as possessions with the assist of virtualization knowledge. Virtualization is equipment that intangibles absent the particulars of corporeal hardware and gives virtualized resources for high-level requests. Virtualization is a leading expertise in the countenance of resource operation sorting in huge kind of resources from mainframe, recall to storage and system. Leveraging virtualization creates it probable to combine, disconnect and evolutes resource operation.

Spanning and dimension of the cloud possessions demand arrangements for resource administration and transmitting requests. This subject is an inspiration to suggest algorithms and machines for allocation, harmonizing and organization of resources in cloud surroundings. Current years there have been planned different algorithms for organization and QoS provisioning in the cloud resources.

The objective of such a cloud is to supply competent good method questioning on the back-end data at a short cost with clever mode while being reasonably practical, and in addition, best advantageous and also receiving decrease of preparation cost on command changes. A worth over the commission charge for every structure can make sure proceeds for the cloud. And also inside cloud is occasionally inform his caches in turn as necessary on diverse cloud and other command will almost runs a diverse enthusiastic server at an instance on cloud.

2 Related work

Cloud computing is a hopeful commercial framework that promises to do away with the obligation for supporting pricey computing amenities. Nonetheless, the current cost-effective clouds created to continue web and minute database workloads tremendously vary from characteristic computing workloads.

While services related to current cloud computing are insufficient for scientific computing, Many-Task Computing [1] employing loosely coupled applications which did not provide with resources, required instantly. Our work differs in the problem that the address is towards the minimization of resource utilization obtained from the resource pool instantly. Subsequently, VM migration algorithm based on Nash equilibrium [2] solves resource utilization maximization problem in virtualized data centers.

Cloud computing is a deep uprising method in its calculation ability. The major purpose currently is to decrease the charge of organizing a service in the cloud and containing correct coordinative in among the models. Public, private, and mixture cloud [3] surroundings all facade the presentation confines intrinsic in todays requests and networks. In turn for endeavors to exploit the plasticity and cost investments of the Public, private, and cross cloud they have to conquer the similar latency and bandwidth restraints that confront dispersed IT communications environments [4].

The optimization trouble of reducing resource cost in cloud for meeting service necessities is analyzed in literature[5]. Social Clouds provide the possibility to share resources among clients. The benefit provided to clients within a social network community using different economic patterns and evolved technical metrics, through simulation [6]. A novel framework called, Network Flow based Resource Allocation NFRA, for reducing the energy consumption and increasing the profit was introduced in [7]. The OCRP [8] gave provisioning for computing resources to be used in multiple provisioning stages as well as a long-term plan in which cloud consumer minimize cost of provisioning in cloud environments. In literature[9], a novel highly decentralized information accountability framework to keep track of the behavior of the users in the cloud was maintained to provide the accountability.

The problem for assigning with a set of clients for certain demands towards a set of servers with capacities and degree constraints are presented in literature[10]. The goal of heterogeneous resource allocation was to find an allocation, in such a way that the number of clients allocated with a server is smaller than the degree of assigning to the server. At the same time, their overall demand of the client should be smaller than the server's capacity, while maximizing the overall throughput. However, achieving security in cloud was another major concern. The author presented FADE [11], a mechanism to achieve full security goals upon a set of cryptographic key functions supported by a quorum of key managers. In literature[12], the task of assigning a third party auditor was the main focus that works on behalf of the cloud and client to verify the integrity of data stored in client. Consequently, the choosing nodes for implementing a job in the cloud computing must be measured to develop the efficiency of the resources [13].

The author provided a method called, Nephele, a first data processing framework that uses dynamic resource allocation [14] for both the tasks, scheduling and execution. With CPU and memory as constraints, a protocol called Gossip

[15] was designed that minimizes the cost for adapting an allocation that executes on dynamic input and does not require global synchronization.

Scheduling algorithms for parallel jobs [16] make efficient use of the two tier VMs to improve the responsiveness which significantly outperforms commonly used algorithms such as extensible Argonne scheduling system in a data center setting. A method for the efficient mapping of resource requests with a heuristic methodology [17] was addressed. Routing and scheduling algorithm [18] for cloud architecture target minimal total energy consumption by switching off unused network and/or information technology (IT) resources. The author provided maximized resource utilization with optimal execution efficiency [19] using the proportional share model. Various workflow scheduling algorithms are listed and compared their characteristics and applicability for cloud scheduling and concluded that HCOC schedulers superior performance may partially be due to its multicore awareness, which is clearly a characteristic requiring consideration in hybrid cloud computing [20].

A mechanism for scheduling single tasks considering two objectives: monetary cost and completion time and dynamic scheduling of scientific workflows [21] was proposed. A cloud scheduler [22] considers both user requirements and infrastructure properties assured users that their virtual resources are hosted using physical resources that match their requirements without getting users about the details of the cloud infrastructure.

All the works mentioned above provide mechanisms for resource management in cloud environment. Certain other problems related to resource allocation for cloud in the existing P2P infrastructure has to be addressed. So, the problem specification is provided in depth, followed by the method adopted to solve the issue.

3 Problem specification

Normally, cloud consists of set of resources. To manage the set of resources in the cloud, it is necessary to provide a resource manager (RM) to control and set up. The request sent by the clients (C) for the requisition of resources is processed based on Dispatcher algorithm. Once the resources are allocated and released, resource monitoring unit (RMU) set up a list of resources which are free from the data centers on it. The resource monitoring unit also provides information about the unavailability of the resources in the data centers. The sequence diagram described in Fig.1 uses web service as an interface to enhance the communication between the clients and clouds.

The workflow of managing the resources proceeds with the clients' requests. The foremost step is that the client sends a resource request to the cloud, through the Web Interface (WI). The WI then passes onto the authorization unit

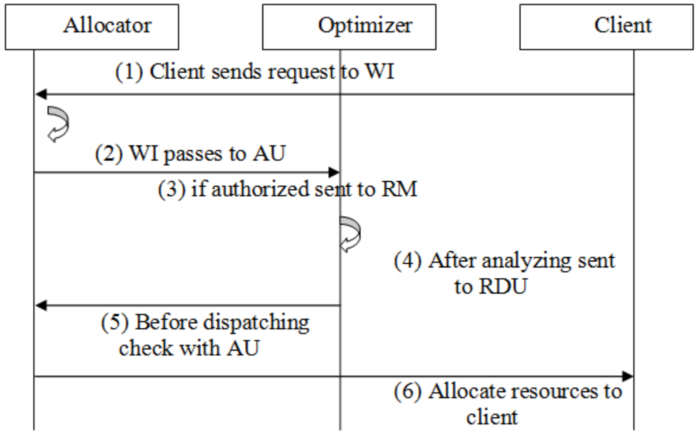

Fig.1 Resource allocation (without scheduler)

(AU). The AU check for authorization and proceeds with the next process upon successful completion of the authorization. It will pass the clients requests to the Resource Manager (RM). Upon analyzing the set of resources required by the client, the RM passes the requests to the resource dispatcher unit (RDU). Before dispatching the set of resources to the client, the RDU checks the clients authorization information from the authorization unit (AU). Then the AU allocates the resources to the clients who send requests. By this way, the resources are allocated to the clients from the clouds.

4 Scheduler-based optimal resource allocation in cloud environment

The previous section discussed about the process of managing the resources in the clouds in conventional manner. If allocation of resources is done improperly, then the resources which are free are simply assigned to the task accomplished. This results in the wastage of resources in the resource pool. To enhance and to further optimize the resource allocation in the cloud, in this section we describe a mathematical model using scheduler-based model with a set of constraints.

As a solution for resource utilization in hierarchical distributed peer networks, inward requests to cloud primarily are sent to an essential component. This vital component called the scheduler, judge quantity of present resources in cloud instances and employs an improvised dispatcher algorithm to choose which cloud instance swarm this request and propels essential instructions to its virtual machine formation. At first we design the architecture, further explain the scheduler-based optimal resource allocation using improvised dispatcher algorithm.

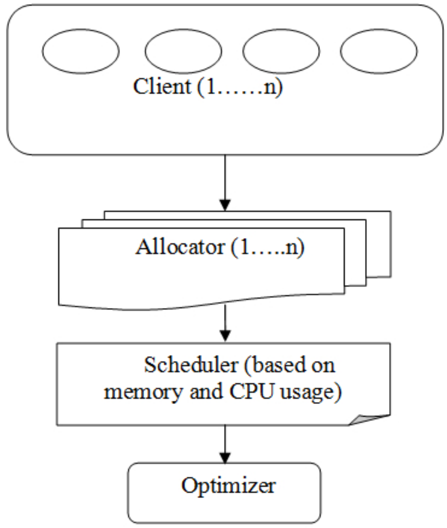

Fig.2 Architecture of S-ORA

4.1 System architecture of scheduler-based optimal resource allocation

The scheduler-based optimal resource allocation is designed based on sensible clouds like Amazon EC2. The architecture of scheduler-based resource allocation is shown in Fig.2. The process of allocating the resources is described individually based on the four components namely, the client, the allocator, the scheduler and finally the optimizer.

The process starts with the (i) Client (C), (ii) Allocator comprises of Web Interface (WI) and Authorization Unit (AU) (iii) Scheduler schedules the availability of resources according to the availability and finally (iv) Optimizer comprises of Resource Manager (RM), Resource Monitoring Unit (RMU) and Resource Dispatcher Unit (RDU). The notations will be used throughout the work.

4.2 Process of scheduler-based optimal resource allocation

To balance the set of utilized resources and free resources in the cloud, in this section, the resources are distributed among the clients to increase the availability with the help of scheduler. The scheduling of resources in the clouds are focused oriented towards the management of memory and CPU usage of every clients required. A mathematical model is presented here to introduce a parametric environment for better allocation of resources in cloud computing environment.

Before assigning the resources to the clients, the current stature of the client is to be noted. The stature includes the requirement of resources, memory and CPU usage. For this purpose, a parametric setting is initiated in a method to think about proximity and resource consumption on cloud instances as illustrated below using the following constraints. Let us consider the Scheduler-based Optimal Resource Allocation problem for processing and memory usage denoted by the equations (1) and (2) respectively.

$$\alpha = \sum_{i=1}^{n} \frac{(Proc_c[avail] - Proc_i[cons]}{Total\_Proc_c} \tag{1}$$

$$\beta = \sum_{i=1}^{n} \frac{(Mem_c[avail] - Mem_i[cons]}{Total\_Mem_c} \tag{2}$$

where $i = 1, 2, ..., n$ requests placed by the clients to n cloud instances denoted by $c = 1, 2, ..., m$. The scheduler-based optimal resource allocation using processing and memory are further denoted by objective function as given below:

$$x = \alpha \in Proc^n symbolizes\ x = \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix}, x_i \in Proc \tag{3}$$

Similarly,

$$y = \beta \in Mem^n symbolizes\ y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{pmatrix}, y_i \in Mem \tag{4}$$

where Proc^n denotes set of n representation or n requests placed by the clients. Proc$_c$[avail] and Mem$_c$[avail] represent the current processing and memory availability in cloud instances denoted by ‘c’ for a particular request with total processing and memory available in cloud ‘c’ represented by Total_Proc$_c$ and Total_Mem$_c$ respectively. Proc$_i$[cons] and Mem$_i$[cons] represents the processing and memory consumed for the corresponding i$^t h$ request. This design creates the model self-sufficient of quantity of resources in each cloud instance and creates a reliable representation for all cloud instances.

The two parameters, Proc$_i$ and Mem$_i$ show the ratio of resource utilization for every machine. The two parameters are evaluated for every request and for every clients present in the cloud. The process of Scheduler-based Optimal Resource Allocation in cloud computing environment is performed using (A) Cloud Interface Generation and (B) Allocating optimal set of resources which will be discussed in the forthcoming section.

4.3 Cloud interface generation

The Cloud Interface Generation (CIG) forms the foremost step in scheduler-based optimal resource allocation. The CIG generate, retrieve and modify the

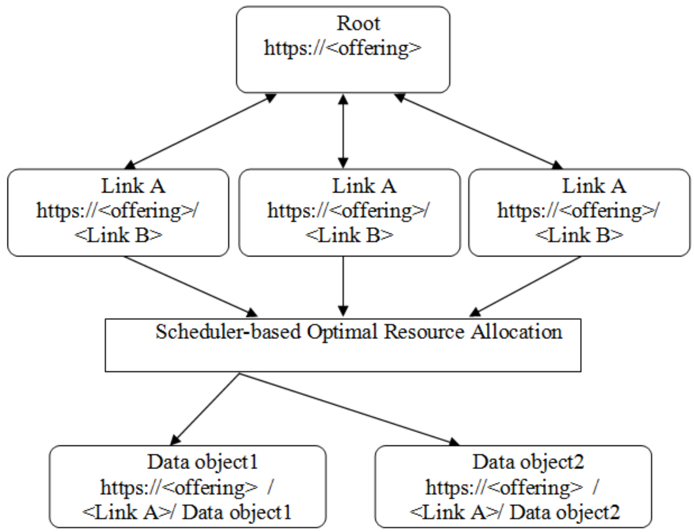

Fig.3 Model for scheduler-based cloud interface generation

data items from the cloud instances. With the interface generated, the clients in the clouds derive the capability of the cloud to manage, link and data placed in it. The cloud interface send and process the data items through the set of cloud instances. With the cloud instances, the client sent the request for resources. Most of the existing cloud process through these interfaces. A sample model for the cloud interface generation using scheduler-based model is shown below in Fig.3.

For the purpose of data storage operations, the client requires to know only about the link objects and data objects. As illustrated in Fig.3, the client performs a PUT to the link URL and creates a new link with the specified name. Once the link for the client is created, the scheduler does the job of scheduling the processing and memory of the respective client and finally performs a PUT to form a new data object URL. The subsequent GET then fetches the actual data object and its corresponding value section.

4.4 Allocating optimal set of resources

Once the cloud interfaces has been generated, the clients select the cloud instances from the set of cloud instances available in the network. Next, the optimal resource allocation is taken place for each client who sends requests to the scheduler as illustrated in Fig.4. The scheduler allocates the resources based on the resource availability. The optimal resource allocation is done based on the process of data exchange between the available cloud instances and assigning/releasing the resources based on the users tasks. The processes is based on two steps, one is task selection and node selection.

For the optimal set of allocating resources, the scheduler maintains resource pool and task pool. The resource pool consists of set of freed resources and the task pool consists of set of tasks which are not assigned to any resources. At first, the scheduler checks these two pools before assignment of resources. The scheduler identifies the rate of resource utilization and further identifies the task which is to be performed. Once the resource from the resource pool is selected, the particular resource is locked by the scheduler and it cannot be assigned to any other tasks requested by the clients. Once the resource is locked, the corresponding node which holds its task is ready to send the task to finish. Once the task is received by the node, the resource starts its process to accomplish the task.

Input: Client Request for resource
Output: Assign resource to client
Initialize Client (**C**), Resource Manager (**RM**), Resource Monitoring Unit (**RMU**), Web Interface (**WI**), Authorization Unit (**AU**), Resource Dispatcher Unit (**RDU**)

1. **Submit** set of resources to RM
2. **Submit** client (C) requisition of resources to RM
3. **Check** for client authorization by AU
4. **IF** (authorization successful) then RDU
    a. **Sent** C request to RM through WI
    b. **Process** through cloud instance
    c. **Select** cloud instance
5. **END IF**
6. **FOR** each client request
    a. **IF** task (is assigned) then **scheduler** check for resource pool and task pool **THEN**
        i. **Identify** task
        ii. **Select** resource from resource pool
        iii. **Lock** selected resource
        iv. **Accomplish** task
    b. **ELSE IF** (task is unassigned by **scheduler**) **THEN**
        i. **Identify** resource request by C
        ii. **Allocate** resource to C once resource gets freed
    c. **END IF**
7. **END FOR**

Fig.4 Scheduler-based optimal resource utilization using improvised dispatcher algorithm for distributed hierarchical peer networks

If any new task enters into the network, the scheduler checks the status of the task. If the task is unassigned, the scheduler determines the resource requirements of the corresponding task and further assigns it. Based on the utilization of resources and task, the nodes will adjust its corresponding resource allocation processes. The performance of the scheduler-based optimal resource allocation is analyzed in the forthcoming section.

5 Experimental evaluation

The Scheduler-based Optimal Resource Allocation (S-ORA) for HP2P networks with efficient data exchange among cloud instances are implemented in Java using Amazon EC2 a web service that provides resizable computing capacity in cloud. The S-ORA uses the Amazon EC2's simple to use web service interface to obtain and configure that reduces the time taken to obtain and boot new server instances to minute. The performance evaluation tests aimed at comparing the direct invocation for cloud computing services using hierarchical distribution process with challenging interactions through traditional scientific computing.

The cloud computing services at first recognizes the source and destination node for data transfer from one cloud instance to the other. The source nodes effectively chose the destination node based on hierarchical distribution process. So, the data transfer is successfully performed in cloud computing environment with Amazon relational database service. Then the fitness of commercial cloud service to hierarchical data exchange between multiple cloud instances is measured. Presenting an efficient commercial cloud structure for service provider with optimal resource utilization and fast and accurate data exchange between the cloud service users. The performance of scheduler-based optimal resource allocation is evaluated by, number of rejected requests, time consumption to exchange data between cloud instances and resource utilization by clients from different cloud instances.

In this section the experimental setup for designing S-ORA that is used in our experiments is explained. The experiments were conducted on Amazon's EC2 infrastructure due to the popularity, feature rich, and stable commercial cloud available. It offers distinct resource configurations for virtual machine instances. Amazon EC2's interface minimizes the time required for various instances according to the changes observed in computing requirements. S-ORA experiments with c1.medium, a compute optimized instance type, a 32-bit processor, 1.7 GB RAM and 350 GB local disk storage.

6 Results and discussion

In S-ORA, HP2P networks are designed for cloud computing service composition to capture the minimal resource allocation for service performance to other systems written in mainstream languages such as Java using Amazon EC2 web service. Independent tests are run with growing number of applications, and constant number of service requests sent by each user. The performance graph and table describes the evaluation of the performance of S-ORA.

6.1 Measure of number of rejected requests

The number of rejected requests [Client_{rejreq}] measures the number of request rejected as made by the client to the cloud instance through scheduler using Amazon EC2 with windows server. The rejected request for a client is evaluated based on the difference between the requests made by the client and rejected request made by the cloud for the specific client at a particular time period. A rejected request for a client is given as:

$$Client_{rejreq} = Client_{req} - Cloud_{rejreq} \tag{5}$$

Table 1 described the performance of S-ORA and compared the results with existing works like Bargaining Towards Maximized Resource Utilization in Video Streaming Datacenters [1], CCS using scientific computing [2] based on requests to be rejected.

Table 1 Number of requests vs. rejected requests

Number of requests (task/minute)	Number of rejected requests (task/minute)		
	Proposed S-ORA	Existing Nash bargaining solutions	Existing CCS using scientific computing
5	1	3	4
10	3	5	7
15	4	8	13
20	7	12	16
25	10	14	17
30	12	15	22

Fig.5 shows the ratio of requests made by the client to cloud instances, to the number of rejected requests from the cloud instances according to the availability of resources. Sending the client requests to cloud, either result in accept of the particular request, if the cloud has resource to be allocated from pool of resources or reject the clients request if the resource is not available. With the help of improved dispatcher algorithm, the number of rejected request is measured.

The algorithms effectiveness is measured using the rejection rate. Higher the number of rejection rate, lower the performance of algorithm. Since the resource is allocated with the help of scheduler in the proposed model, the performance of S-ORA is improved when compared to other algorithms. The number of rejected request using S-ORA is 40%, 50% using Nash bargaining solutions and 70% with CCS using scientific computing. When compared with the three algorithms, the rejected rate using S-ORA is comparatively less as the scheduler schedules the

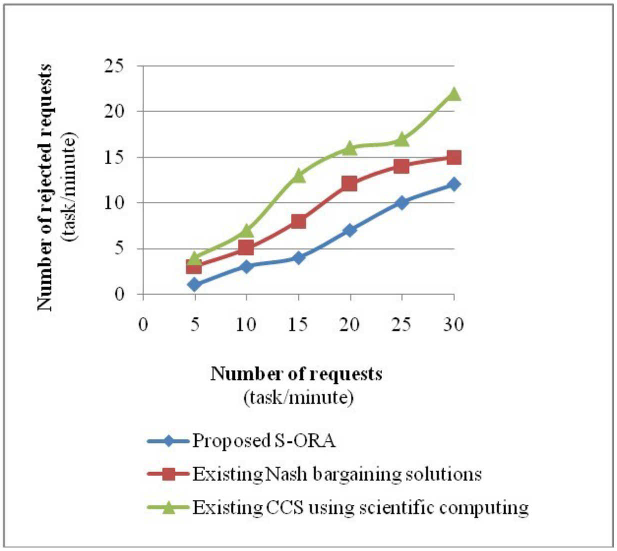

Fig.5 Number of requests vs. rejected requests

resource using the resource pool whereas the rejected rate using Nash bargaining solutions is 50% which is less when compared to CCS as it uses the Nash bargaining solution.

6.2 Measure of time consumption for data exchange

The time consumption [Time_{con}] in S-ORA measures the time consumed for data exchange between the cloud instances. The time consumption is evaluated on the basis of distance for particular data exchange by the cloud instances divided by the speed with which the operation is performed. The formula to evaluate time consumption is given below:

$$Time_{con} = Distance_{dataexc}/Speed \tag{6}$$

The time consumed to exchange the data by the cloud instances are illustrated in Table 2 for the S-ORA with the existing works like Bargaining Towards Maximized Resource Utilization in Video Streaming Datacenters, CCS using scientific computing.

Fig.6 describes the consumption of time taken to exchange the data among the cloud instances in the distributed peer networks. In the proposed S-ORA, the deployment of commercial cloud computing services is efficiently achieved using the scheduler-based cloud interface generation. The is due to clear separation between resource manager (RM) and resource dispatcher unit (RDU), where the scheduler maintains the set of resources and at the same time checks through the pool of resources and allocates accordingly. This in turn results in minimum

Table 2 Data size vs. time consumption

Data size (KB)	Time consumption for data exchange (Seconds)		
	Proposed S-ORA	Existing Nash bargaining solutions	Existing CCS using scientific computing
10	10	15	20
20	18	24	26
30	22	33	34
40	30	42	45
50	38	56	58
60	43	68	70

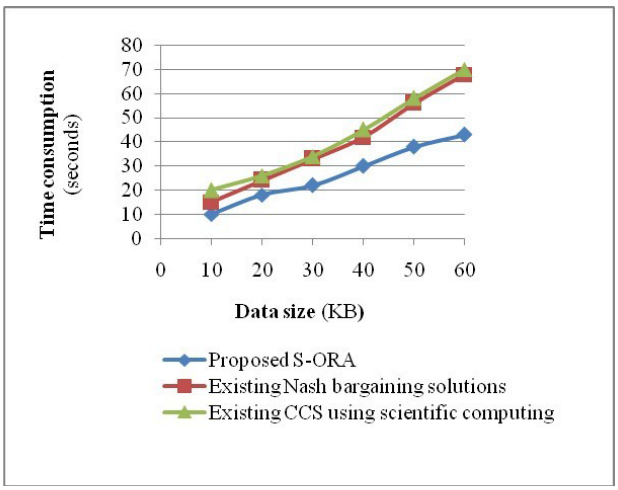

Fig.6 Data size vs. time consumption

time consumption to the hierarchical peer users. Rather than existing works, the S-ORA consume less time to exchange the data among the set of cloud instances. The variance in time consumption is 5-10% low in the S-ORA whereas using Nash bargaining solutions the time consumption is comparatively higher due to the use of VM migration algorithm and finally higher in CCS with many task computing.

6.3 Measure of resource utilization by clients

The resource utilization [Res_{util}] in S-ORA measures the utilization of resources by the clients from different cloud instances through scheduler. Resource utilization is the summation of resource consumption by the total resource availability.

Table 3 Algorithms vs. resource utilization.

Algorithms	Resource utilization (%)
Proposed S-ORA	50
Existing Nash Bargaining towards maximized resource utilization in video streaming datacenters	70
Existing CCS using scientific computing	80

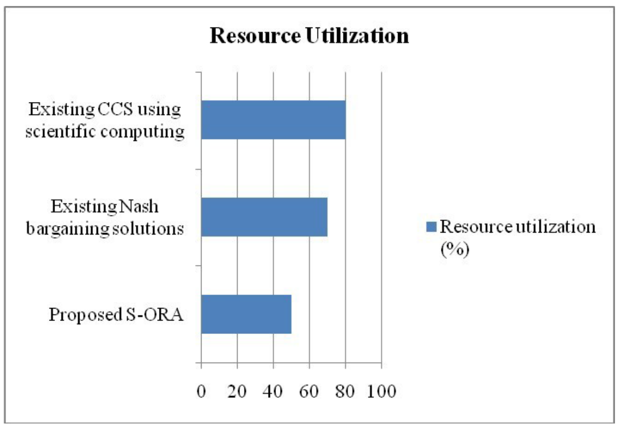

Fig.7 Algorithms vs. resource utilization

Resource utilization is evaluated using the formula given below:

$$\begin{aligned}\mathrm{Res}_{util} &= Actual\mathrm{Res}_{util}/Total\mathrm{Res}_{util} \\ &= Proc_i + Mem_i/Total\_Proc_i + Total\_Mem_i\end{aligned} \tag{7}$$

The comparison of utilization of resources for the purpose of exchange of data using three different algorithms is depicted in Table 3.

Fig.7 describes the utilization of resources for the efficient data exchange between the cloud instances in the hierarchical distributed peer networks. When a network resource, for instance a CPU or a meticulous disk, is engaged by an operation or query, it is occupied for handing out further requests. Awaiting requests have to stay for the resources to turn out until the resources have been freed by other clients. Existing woks such as Bargaining towards Maximized Resource Utilization in Video Streaming Datacenters and CCS using scientific computing has higher the percentage of time that the resource is occupied, the longer each operation must wait for its turn. So, rather than using existing schemes, the S-ORA utilizes minimal resources for transfer of data which is handled efficiently by the scheduler with the set of cloud instances. Further, from the Table 3 it is evident that S-ORA uses minimum resources (50%) whereas the Nash bargaining solutions uses (70%) that is minimum when compared to the CCS (80%) as the Nash bargaining solutions equips the pivot point which is determined by the user, but comparatively higher to the S-ORA as it uses the scheduler-based CIG resulting in minimum resource utilization.

Finally, it is being observed that the scheduler-based optimal resource allocation technique efficiently manage the resources and tasks of the nodes in the cloud instances. By generating the scheduler-based cloud interface, the allocation of resources is done in an optimal manner.

7 Conclusion

This paper efficiently analyzes and resolves the issue of resource management in cloud environment by adapting scheduler-based optimal resource allocation technique. To manage the resources in the clouds, the cloud interface is generated primarily with the help of scheduler. With the cloud interface, the resources and the unassigned tasks are balanced efficiently using the scheduler. Then the improved dispatcher algorithm is presented to manage the resources and tasks efficiently in the cloud instances. The scheduler-based optimal resource allocation using improved dispatcher algorithm is implemented on Amazon EC2 as a practical cloud. The obtained results are compared with the existing works like Bargaining towards Maximized Resource Utilization in Video Streaming Datacenters, CCS using scientific computing. The results had shown that improper resource allocation in the existing works leads to decline in resource utilization. Experimental results showed that using the optimal resource allocation algorithm provides at most 5-10% enhancement in resource consumption than unique works in commercial cloud instances.

References

[1] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer and D. H. J. Epema. (2011), "Performance analysis of cloud computing services for many-tasks scientific computing", *IEEE Transactions on Parallel and Distributed Systems*, Vol.22, pp.931-945.

[2] Yuan Feng, Baochun Li and Bo Li. (2012), "Bargaining towards maximized resource utilization in video streaming datacenters", *IEEE INFOCOM*, No.25-30, pp.1134-1142.

[3] V. Khoshdel, S. A. Motamedi, S. Sharifian and M. Farhadi (2011), "A new approach for optimum resource utilization in cloud computing environ-

ments", *International eConference on Computer & Knowledge Engineering*, No.13-14, pp.314-321.

[4] Deepak Mishra and Manish Shrivastava (2012), "Optimal service pricing for cloud based services", *Journal of Soft Computing & Engineering*, Vol.2, pp.531-540.

[5] Han Zhao, Miao Pan, Xinxin Liu, Xiaolin Li, Yuguang Fang. (2012), "Optimal resource rental planning for elastic applications in cloud market", *IEEE 26th IPDPS*, pp.808-819.

[6] M. Punceva, I. Rodero, M. Parashar, O. F. Rana, I. Petri. (2012), "Incentivising resource sharing in social clouds", *IEEE International Workshop on Infrastructure for Collaborative Enterprises*, pp.185-190.

[7] K. Patel, M. Annavaram, M. Pedram, NFRA. (2013), "generalized network flow based resource allocation for hosting centers", *IEEE Transactions on Computers*, Vol.62, pp.1772-1785.

[8] S. Chaisiri, Bu-Sung Lee, D. Niyato. (2012), "Optimization of resource provisioning cost in cloud computing", *IEEE Transactions on Services Computing*, Vol.5, pp.164-177.

[9] S. Sundareswaran, A. C. Squicciarini, D. Lin. (2012), "Ensuring distributed accountability for data sharing in the cloud", *IEEE Transactions on Dependable & Secure Computing*, Vol.9, pp.556-568.

[10] Beaumont Olivier, Eyraud-Dubois Lionel, Thraves Caro Christopher, Rejeb Hejer. (2013), "Heterogeneous resource allocation under degree constraints" *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, pp.926-937.

[11] Yang Tang, P. P. C. Lee, J. C. S. Lui, R. Perlman. (2012), "Secure overlay cloud storage with access control and assured deletion", *IEEE Transactions on Dependable & Secure Computing*, Vol.9, pp.903-916.

[12] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li. (2011), "Enabling public auditability and data dynamics for storage security in cloud computing", *IEEE Transactions on Parallel and Distributed Systems*, Vol.22, pp.847-859.

[13] Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng Wang, Ching-Wei Chen. (2011), "A three-phases scheduling in a hierarchical cloud computing network", *Third International Conference on Communications & Mobile Computing*, pp.114-117.

[14] D. Warneke, Odej Kao. (2011), "Exploiting dynamic resource allocation for efficient parallel data processing in the cloud", *IEEE Transactions on Parallel and Distributed Systems*, Vol.22, pp.985-997.

[15] F. Wuhib, R. Stadler, M. Spreitzer. (2012), "A gossip protocol for dynamic resource management in large cloud environments", *IEEE Transactions on Network & Service Management*, Vol.9, pp.213-225.

[16] Xiaocheng Liu, Chen Wang, Bing Bing Zhou, Junliang Chen, Ting Yang, A. Y. Zomaya. (2013), "Priority-based consolidation of parallel workloads in the cloud", *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, pp.1874-1883.

[17] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, A. Monje. (2013), "On the optimal allocation of virtual resources in cloud computing networks", *IEEE Transactions on Computers*, Vol.62, pp.1060-1071.

[18] J. Buysse, K. Georgakilas, A. Tzanakaki, M. De Leenheer, B. Dhoedt, C. Develder. (2013), "Energy-efficient resource-provisioning algorithms for optical clouds", *IEEE Journal of Optical Communications & Networking*, Vol.5, pp.226-239.

[19] Sheng Di, Cho-Li Wang. (2013), "Dynamic optimization of multiattribute resource allocation in self-organizing clouds", *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, pp.462-478.

[20] L. F. Bittencourt, E. R. M. Madeira, N. L. S. Da Fonseca. (2012), "Scheduling in hybrid clouds", *IEEE Communicatons Magazine*, Vol.50, pp.42-47.

[21] H. M. Fard, R. Prodan, T. Fahringer. (2013), "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments", *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, pp.1203-1212.

[22] I. M. Abbadi, Anbang Ruan. (2013), "Towards trustworthy resource scheduling in clouds", *IEEE Transactions on Information Forensics & Security*, Vol.8, pp.973-984.

Corresponding author
R. A. Karthika can be contacted at: Karthika78@gmail.com