INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

EFFICIENT RESOURCE ALLOCATION AND SCHEDULING IN CLOUD COMPUTING ENVIRONMENT

R. Dharani[1], Dr. M. Kalaiarasu[2]

[1]PG Scholar, Computer Science and Engineering,
[2]Associate Professor, Department of Information Technology,
Sri Ramakrishna Engineering College, Coimbatore, Tamilnadu, India.

Abstract: - The cloud architecture is usually composed of several XaaS layers—including Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The previous work studies efficient resource allocation to optimize objectives of cloud users, IaaS provider and SaaS provider in cloud computing. This work proposes the composition of different layers in the cloud, such as IaaS and SaaS, and its joint optimization for efficient resource allocation. The efficient resource allocation optimization problem is conducted by sub problems. Our proposed work mainly concentrates on efficient scheduling and resource allocation to optimize objectives of cloud users, IaaS provider and SaaS provider. Early task scheduling algorithms are focused on minimizing make span, without mechanisms to reduce the monetary cost incurred in the setting of clouds. A cost-efficient task-scheduling algorithm using two heuristic strategies. The first strategy dynamically maps tasks to the most cost-efficient VMs based on the concept of Pareto dominance.

Keywords: Virtualization, Resource management, Distributed storage system, Scheduling, Resource utilization.

1 INTRODUCTION

There are many considerations for cloud computing architects to make when moving from a standard enterprise application deployment model to one based on cloud computing [2]. There are public and private clouds that offer complementary benefits, there are three basic service models to consider, and there is the value of open APIs versus proprietary ones.

Cloud computing is transforming business by offering new options for businesses to increase efficiencies while reducing costs. It lets user can access all applications and documents from anywhere in the world, freeing from the confines of the desktop and making it easier for group members in different locations to collaborate. It is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a metered service over a network (typically the Internet) [12]. Cloud computing provides computation, software, data access, and storage resources without requiring cloud users to know the location and other details of the computing infrastructure. End users access cloud based applications through a web browser or a light weight desktop or mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance as if the software programs were installed locally on end-user computers.

Vol.4 Issue 3, Pg.: 48-55
March 2016

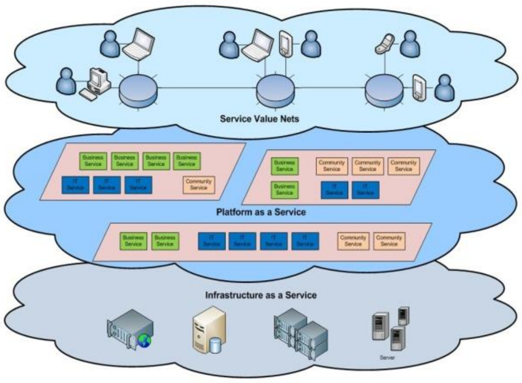

Fig 1. Cloud Architecture

Cloud computing is transforming business by offering new options for businesses to increase efficiencies while reducing costs [10]. These problems include:

- High operational costs, typically associated with implementing and managing desktop and server infrastructures
- Low system utilization, often associated with non-virtualized server workloads in enterprise environments
- Inconsistent availability due to the high cost of providing hardware redundancy
- Poor agility, which makes it difficult for businesses to meet evolving market demands

Efficient resource allocation to optimize objectives of cloud users, IaaS provider and SaaS provider in cloud computing. This work proposes the composition of different layers in the cloud such as IaaS and SaaS and its joint optimization for efficient resource allocation [11]. The efficient resource allocation optimization problem is conducted by sub problems.

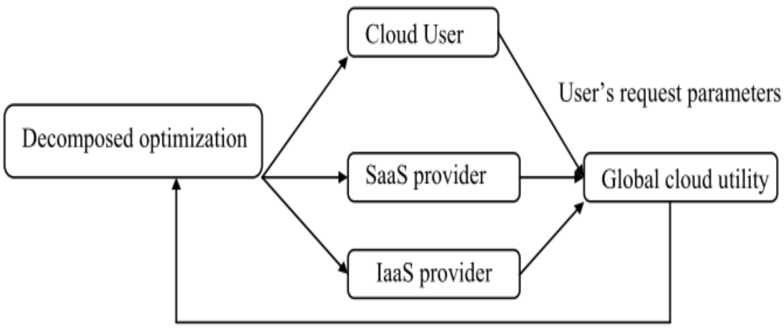

Fig 2. Resource allocation considering cloud users, IaaS provider and SaaS provider

The proposed cloud resource allocation optimization algorithm is achieved through an iterative algorithm. The proposed efficient resource allocation for optimizing the objectives of cloud users, IaaS provider and SaaS provider in cloud computing is conducted by sub problems. In each iteration, the cloud users compute the unique optimal payment to SaaS provider under the deadline constraint to maximize the cloud user's satisfaction. The cloud users individually solves its fees to pay for SaaS services to complete its all jobs, adjusts its SaaS service demand and notifies the SaaS provider about this change.

Formula for optimization problem of task scheduling with multiple VMs and different pricing models as a convex combination of minimizing makespan and monetary costs. These heuristics can be further Classified into four groups: (1) List Scheduling Algorithm (2) Task Duplication-based Scheduling Algorithm (3) Clustering Algorithm and (4) Guided Random Search Algorithm. It was shown that for task scheduling, HEFT outperforms the other scheduling heuristics. All these algorithms intend to minimize makespan, and none of them deals with the problem of dispatching large tasks in a cloud setting, or consider monetary costs.

Minimizing makespan and minimizing monetary costs are competing objectives. The key component of our algorithm is two task-scheduling heuristics. The first heuristic uses the concept of Pareto dominance to generate a cost-efficient schedule based on the execution time of the tasks and the monetary charges of the VMs. The second heuristic complements the first heuristic–it attempts to minimize the monetary costs of non-critical tasks by extending their execution time. It show that our algorithm can substantially reduce monetary costs while producing makespan as good as the best known task-scheduling algorithm can provide.

2 RELATED WORKS

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. Above the clouds: a Berkeley view of cloud computing by M. Armbrust [1].The services themselves have long been referred to as Software as a Service (SaaS), so the term was used.

The datacentre hardware and software is what will be called as a Cloud. When a Cloud is made available in a pay-as-you-go manner to the public, so call it a Public Cloud; the service being sold is Utility Computing. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility [2] Computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to traditional utilities such as water, electricity, gas, and telephony. In such a model, users access services based on their requirements without regard to where the services are hosted or how they are delivered..Several computing paradigms have promised to deliver this utility computing vision and these include cluster computing, Grid computing, and more recently Cloud computing.

The latter term denotes the infrastructure as a “Cloud” from which businesses and users are able to access applications from anywhere in the world on demand. Thus, the computing world is rapidly transforming towards developing software for millions to consume as a service, rather than to run on their individual computers. . Underutilizing resources for HPC on clouds [3].The cloud computing model emphasizes the ability to scale compute resources on demand. The advantages for users are numerous. Unlike conventional cluster systems, there is no significant upfront monetary or time investment in infrastructure or people.

Instead of allocating resources according to average or peak load, the cloud user can pay costs directly proportional to current need. When resources are not in use, total cost can be close to zero. Individuals can quickly create and scale-up a custom compute cluster, paying only for sporadic usage. However, there are also disadvantages to cloud computing services. Costs can be divided into different categories that are billed separately: for example, network, storage, and CPU usage. This model can be complex when attempting to minimize costs. Resource provisioning for cloud computing [4] Static resource allocation based on peak demand is not cost-effective because of poor resource utilization during off-peak periods.

In contrast, autonomic resource management could lead to efficient resource utilization and fast response in the presence changing workloads. This paper is concerned with resource allocation strategies that are relevant to autonomic resource management. The two-level resource management architecture provides a framework for our investigation. At the lower level, there are multiple application environments (AEs). Each AE consists of a set of computing resources that are shared by one or more applications. At the higher level, a global arbiter performs resource allocation across AEs. Efficient Resource Provisioning in Compute Clouds via VM Multiplexing [5] In both static and dynamic provisioning, VM sizing is perhaps the most vital step. VM sizing refers to the estimation of the amount of resources that should be allocated to a VM. The objective of VM sizing is to ensure that VM capacity is commensurate with the workload. While over-provisioning wastes costly resources, under-provisioning degrades application performance and may lose customers.

Traditionally, VM sizing is done on a VM-by-VM basis, i.e., each VM has an estimated size based on its workload pattern. In a significant departure from such an individual-VM based approach, a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned based on an estimate of their aggregate capacity needs. SaaS (Software as a Service) On Cloud [6] Cloud computing is fast growing as an alternative to conventional computing. However, the paradigm is same as cluster computing, distributed computing, utility computing and grid computing in general.

Cloud computing creates a virtual paradigm for sharing data and computations over a scalable network of nodes. Examples of such nodes include end user computers, data centers, and web services. Such a scalable network of nodes is called cloud. An application based on such clouds is taken as a cloud application. Cloud computing is modern TCP/IP integrations of computer and network technologies such as fast micro processor, gigantic memory, high-speed network and reliable system architecture.

3 PROPOSED SYSTEM

A new task-scheduling algorithm for running large programs in the cloud. Most conventional task scheduling algorithms do not consider monetary costs, and so they cannot be directly applied in a cloud setting. Existing system algorithm computes scheduling plans that produce make span as good as the best known algorithm of while significantly reducing monetary costs. The concept of Pareto dominance was used to devise a cost-efficient scheduling algorithm to process multiple tasks in the cloud setting.
Pareto optimality incorporating monetary cost into task scheduling adds a layer of complexity. Since cannot directly compare different scheduling plans with competing objectives using the concept of Pareto dominance to select VMs and carry out comparisons.

Pareto optimal scheduling heuristic (POSH) describe a heuristic to dispatch tasks in a DAG to the cost-conscious VMs based on Pareto dominance, and call it Pareto Optimal Scheduling Heuristic (POSH). POSH is an extension of the Heterogeneous Earliest Finish Time (HEFT) heuristic. Developed for scheduling tasks on heterogeneous dedicated multiprocessing systems, HEFT is better than other scheduling heuristics. HEFT assigns a priority to each task in the DAG and then maps the task with the highest priority to the VM that minimizes the earliest finish time. POSH uses both the running time and the monetary cost to modify the last step to map the task with the highest priority to the most cost-efficient VM based on Pareto dominance. The advantages are

- Our algorithm can substantially reduce monetary costs while producing makespan as good as the best known task-scheduling algorithm can provide.
- To find a schedule of these tasks so that the monetary costs and the makespan are both minimized.

4 SYSTEM IMPLEMENTATION

4.1 Cloud Setup

A simulation toolkit enables modeling and simulation of Cloud computing systems and application provisioning environments [23]. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies [27]. It implements generic application provisioning techniques that can be extended with ease and limited effort. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios. In this module creating cloud users and datacenters and cloud virtual machines as per our requirement.

4.2 Resource allocation algorithm for cloud users, IaaS Provider and SaaS Provider

The proposed global optimization strategy for resource allocation mechanisms enables IaaS providers and SaaS providers to partition their resource and services based on some criteria such as profit and resource efficiency. Global optimization based cloud resource allocation is to allocate cloud resources and services such that the cloud system utility *U*Cloud be maximized subject to resource constraints of IaaS provider, QoS constraints of cloud users, and SaaS service constraints of SaaS provider. The parameters are resource utilization, resource cost, execution success ratio. In the experiment, some metrics are designed to test the performance of cloud users, such as execution time, resource utilization and cost.

The resource allocation method for optimizing the objectives of cloud users, IaaS provider and SaaS provider in cloud computing is formulated as follows

----

Resource allocation algorithm

Step 1: cloud users receive from the SaaS provider *I* the price$\tau_i^{(n)}$;

Step 2: cloud users calculate its optimal payments to SaaS provider to maximize the Utility of the users;

$Q_m^{i*}=Max\{(B_m-\sum q_m^i)+(T_m-\sum t_m^n)+(E_m-\sum e_m^n)\}$;

Step 3: cloud users compute new cloud SaaS service

If $B_m \geq \sum_i q^i_m$

Then $v_m^{i(n+1)}=q_m^{i*(n)}/\tau_i^n$;

Else Return Null;

Step 4: send SaaS service demand to SaaS providers

Step 5: SaaS provider receives SaaS service demands v^i_m from cloud users m;

Step 6: SaaS provider receives the price of CPU and memory of the IaaS provider j, r^{cpu}_j, r^{ram}_j;

Step 7: SaaS provider calculates new price of SaaS service

If $S_i \geq \sum_m v^i_m$

Then

$\tau_i^{(n+1)}=\max\{\varepsilon,\tau_i^{(n)}+\eta(\sum_m v_m-s_i)\}$;

$\eta>0$ is a small step size parameter , n is iteration number

Else Return Null;

Step 8: SaaS providers calculate optimal payment to maximize the benefit of SaaS provider; $P_i^{j(cpu)*},P_i^{j(ram)*}=Max\{(SB_i-\sum P_i^{j(cpu)}-\sum P_i^{j(ram)}-\eta\sum D_i^{j(cpu)})\}$;

Step 9: Send SaaS service price to all cloud users;

Step 10: Send the payment to IaaS providers

The resource allocation method in cloud computing aims to maximize U_{Cloud} subject to the constraints of IaaS provider, SaaS provider, and cloud users. In this problem, the first type of the constraints is related with IaaS provider. Further, $x_i^{j(cpu)}$ is CPU required by a VM for SaaS provider *i* from the IaaS provider *j*, $x_i^{j(ram)}$is the memory required by a VM for SaaS provider *i* from the IaaS provider *j*.

Vol.4 Issue 3, Pg.: 48-55
March 2016

The constraint implies that the aggregate CPU does not exceed the total capacity C_j^{cpu} of CPU of IaaS provider *j*, aggregate memory units do not exceed the total resource C_j^{ram} of memory of IaaS provider *j*. The second type of constraints is related with cloud users. Cloud users should complete all its jobs under time limits and certain payment [20]. Cloud users needs to complete a sequence of jobs in a specified amount of time, T_m, while the payment overhead accrued cannot exceed the budget B_m,q_m^i Being the payment of the cloud users *m* to the SaaS provider *i*. $P_i^{j(cpu)}$ *i* , $P_i^{j(ram)}$ are the payments of the SaaS provider *i* to the IaaS provider *j* for CPU and memory required by a VM, respectively. SB_i is the budget of SaaS provider *i*. $D_i^{j(ram)}$, $D_i^{j(cpu)}$ represent resource allocation delays from IaaS provider *j* for memory and CPU required by SaaS provider *i* supporting a VM, respectively. $L_i^{j(ram)}$, $L_i^{j(cpu)}$are upper limits of allocation delay of memory and CPU required by SaaS provider *i* for supporting VM.

Problem *F*1 is conducted by the IaaS provider at resource layer; different IaaS providers compute optimal resource allocation for maximizing the revenue of their own [24]. Problem *F*2 is conducted by SaaS provider at the SaaS service layer, the SaaS provider pays IaaS provider for available cloud resources to run VMs and also provides software services for cloud users to maximize the benefits [15].

The objective of Problem *F*2 is to maximize the surplus of SaaS providers that pays IaaS provider at the resource layer for available resources to run VMs and also revenue that is obtained by providing software services for cloud users. Problem *F*3 is conducted by cloud users; the cloud users give the unique optimal payment to SaaS provider under the deadline constraint to maximize the cloud user's satisfaction.

Optimization-based cloud resource allocation problem can be decomposed into a sequence of three subproblems at three layers [17]. Interactions between the three subproblems are through optimal variables for capacities of cloud computing resources and service demand. Resource allocation problem is conducted by IaaS provider. Different IaaS providers compute optimal resource allocation for maximizing the revenue of their own under constrains of cloud resource capacity.

Cost-efficient task scheduling algorithm

Step 1: IaaS provider receives the payments

$P_i^{j(cpu)*}$,$P_i^{j(ram)*}$from SaaS provider i;

Step 2: IaaS provider calculates optimal resource allocation to provide VM for SaaS provider

$x_i^{j(cpu)*}$,$x_i^{j(ram)*}$=Max{U_{IaaS}};

Step 3: Iaas provider computes a new resource price for SaaS providers

If $C_j^{cpu} \geq \sum_i x_i^{j(cpu)}$,$C_j^{ram} \geq \sum_i x_i^{j(ram)}$

Then

$r_j^{cpu(n+1)}=\max\{\varepsilon, r_j^{cpu(n)}+\eta(\sum x_i^{j(cpu)}-C_j^{cpu})\};$

$r_j^{ram(n+1)}=\max\{\varepsilon, r_j^{ram(n)}+\eta(\sum x_i^{j(ram)}-C_j^{ram})\};$

Else

Return Null;

Step 4: send new prices $r_j^{cpu(n+1)}$, $r_j^{ram(n+1)}$to SaaS providers

----

Assume a fine-grained pricing model based on the actual allocation of CPU cycles. In other words, a VM with more processing power is associated with a higher monetary cost, and vice versa. In particular, use a linear pricing model and an exponential pricing model. In the linear pricing model, the cost of using a VM is linearly correlated with the number of CPU cycles [22]. Let ca_{slow} denote the CPU cycle for the slowest VM m_{slow}. If Vc_{base} is the base price charged to m_{slow}, then the cost incurred to execute task v_i on VM m_j can be calculated as follows:

$$C(v_i,m_j) = \sigma \times t(v_i,m_j) \times Vc_{base} \times \left(\frac{ca_{mj}}{ca_{m\ slow}}\right)$$

Where σ is a random variable used to generate different combinations of VM pricing and capacity. In the exponential pricing model, the cost of VM allocation is calculated as follows:

$$C(v_i,m_j) = \sigma \times t(v_i,m_j) \times Vc_{base} \times exp^{\frac{ca_{mj}}{ca_{m\ slow}}}$$

The total monetary cost is computed by

$$C=\sum_{j\epsilon select} C(v_i,m_j)$$

Incorporating monetary cost into task scheduling adds a layer of complexity. Since cannot directly compare different scheduling plans with competing objectives will use the concept of Pareto dominance to select VMs and carry out comparisons.

Vol.4 Issue 3, Pg.: 48-55
March 2016

The formula for the following minimization problem with an objective function for each node $v_i \epsilon V$ as a convex combination of running time and monetary cost:

Minimize:$\alpha \times T(i.j)+(1-\alpha)\times C(i,j)$for all $m_j \epsilon M$,

Subjectto :$T(i,j)=\frac{t(vi,mj)-tmin}{tmax-tmin}$

$$C(i,j)=\frac{c(vi,mj)-cmin}{cmax-cmin}\alpha \text{ €}[0,1],$$

where α is a cost-efficient factor that represents the user's preference for the execution time and the monetary cost; T (i,j) and C (i,j) represent cost-efficiency ratios of time and costs, respectively; $t_{min(max)}$ and $c_{min(max)}$ are, respectively, the minimum (maximum) execution time and the minimum (maximum) monetary cost of any task scheduling plan.

POSH involves the following three phases:

(a)Weighting Phase: Assign weight to the nodes and edges in the workflow [19]. The weights assigned to nodes are calculated based on the predicted execution time of the tasks and the weights assigned to edges are calculated based on predicted time of the data transferred between the VMs.

(b) Prioritizing Phase: Create a sorted list of tasks organized in the order how they should be executed [22]. The priority of each task is to be set with the upward priority value, which is equal to the weight of the node plus the execution time of the successors. The task list is generated by sorting the tasks by the descending order of priority.

(c)Mapping Phase: Assign the tasks to the resources based on Pareto dominance. Consecutive tasks are mapped to the resources based on the priority queue [25]. For each task, choose the VM that favors scheduling tasks with low monetary cost to run it. This is done by the pre-defined objective function.

Pareto Optimal Scheduling Heuristic (POSH) algorithm

Step 1: Compute priority for all nodes v_i € V by traversing graph upward;
Step 2: Sort v_i in the descending order by its priority value;
Step 3: **for** each v_i € V **do**
Step 4: **for** each m_i € M **do**
Step 5: Compute α T (i , j)+(1-α)*C (i ,j);
Step 6: **end**
Step 7: **end**
Step 8: **for** each task v_i in the ready queue **do**
Step 9: Assign task v_i to the VM m_j that minimizes the objective function of task;
Step 10: **end**

POSH produces a scheduling plan with time slots for task executions and data communications on VMs.

5 PERFORMANCE EVALUATION

Normalize the makespan and monetary cost, and call them the schedule length ratio (SLR) and monetary cost ratio (MCR). In order to compare the efficiency of the algorithms which use several metrics to evaluate their performance [21]. The parameters are resource utilization, resource cost, execution success ratio.

In the experiment, some metrics are designed to test the performance of cloud users, such as execution time, resource utilization and cost.

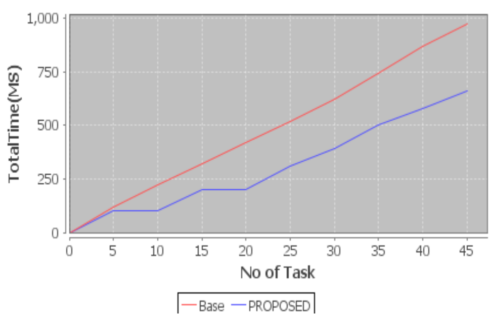

Fig 3. Total Time Graph

Using Total Time as the main metric for the comparisons so that the results will not be sensitive to the size of the DAG. Conceptually, the TT is a normalization of the makespan to an estimate of the best possible schedule length of a given DAG in a given environment [26]. In a perfect world use an optimal schedule for this estimate; however, since finding the optimal makespan is NP-complete, instead use the estimated critical path length. Because the costs of nodes depend on where they are mapped, in this calculation approximate the computation cost of a DAG node by its average cost over all possible processors. fig 3 represents the performance of both existing and proposed system based on the task execution time. In that graph X-axis represents the task varies from 0 to 45 and the Y-axis represents the execution time of task rang of 0 to 1000ms. From the results show that our proposed algorithm has less time compared to existing algorithm.

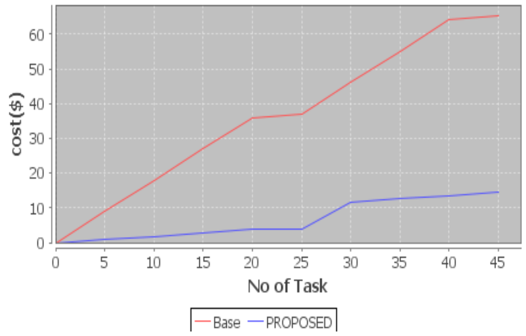

Fig 4. Monetry Cost

Dynamically change the input cost-efficient factor to measure the make span and the monetary cost. In fig 4, X-axis represents the task varies from 0 to 45 and the Y-axis represents the cost of task rang of 0 to 60. Cost depends on two attributes acquisition and on-going [15]. It is not easy to compare different prices of services as they offer different features and thus have many dimensions. Even the same provider offers different VMs which may satisfy users' requirements. Defined a volume based metric, i.e. the cost of one unit of CPU, storage, RAM, and network bandwidth. Therefore, if a VM is priced at p for cpu units, net network units, data unit and RAM memory units, then the cost of the VM. From the results show that our proposed algorithm has less cost compared to existing algorithm

To meet the task requirements of users and improve the utilization of resources while minimizing the make span of given task sets. This proposed is used to improve the resource utilization while try to find optimal schedule for given task set in the cloud environment. In fig 5, resource utilization using proposed algorithm was varying from 0 to 99 percent while existing utilized the resources ranges from 0 to 60 percent. Proposed algorithm increased the resource utilization up to 99 percent when number of tasks was increased, then indicating that proposed is able to work better in the case of more tasks.

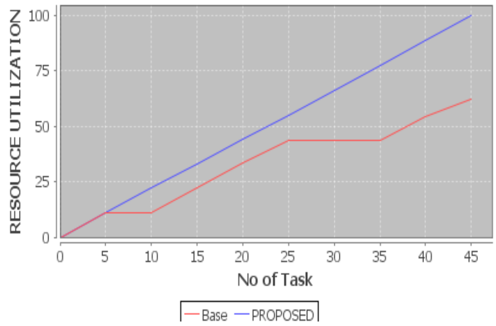

Fig 5. Resource Utilization Graph

CONCLUSION

The previous work study efficient resource allocation to optimize objectives of cloud users, IaaS provider and SaaS provider in cloud computing. The work proposes the composition of different layers in the cloud such as IaaS and SaaS and its joint optimization for efficient resource allocation. The efficient resource allocation optimization problem is conducted by sub problems. The proposed cloud resource allocation optimization algorithm is achieved through an iterative algorithm. A new task-scheduling algorithm for running large programs in the cloud. Most conventional task scheduling algorithms do not consider monetary costs, and so they cannot be directly applied in a cloud setting. In this work, Our algorithm computes scheduling plans that produce make span as good as the best known algorithm of while significantly reducing monetary costs. For the future Virtualization-based full-system measurement and monitoring tools are also included to aid in using the proposed system for co-design of high-performance computing system software and architectural features.

REFERENCES

[1] Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and SaaS provider in cloud environment Chunlin Li · Layuan Li Published online: 7 ebruary 2013 © Springer Science+Business Media New York 2013.
[2] Chunlin Li · Layuan Li Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and SaaS provider in cloud environment Published online: 7 February 2013 © Springer Science+Business Media New York 2013.
[3] SaaS (Software as a Service) On Cloud by Sanjay Chaudhary International Conference on Modern Trends in Science, Engineering and Technology 2013.
[4] Chaisiri S, Lee B, Niyato D (2012) Optimization of resource provisioning cost in cloud. IEEE Trans Serv Comput 5(2):164–177.
[5] Rimal BP, Choi E (2012) A service-oriented taxonomical spectrum, cloudy challenges and opportu-nities of cloud computing. Int J Commun Syst 25(6):796–819
[6] Younge AJ, von Laszewski G, Wang L (2012) Efficient resource management for cloud computing environments. In: IEEE international green computing conference (IGCC), pp 357–364
[7] Garg SK, Yeo CS, Anandasivam A, Buyya R (2011) Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers. J Parallel Distrib Comput 71(6):732–749
[8] Gulati A, Shanmuganathan, G, Holler A (2011) Cloud scale resource management: challenges and techniques. In: USENIX HotCloud, Portland.
[9] Kouki Y, Ledoux T, Sharrock R (2011) Cross-layer SLA selection for cloud services. In: First international symposium on network cloud computing and applications, pp 143–147
[10] Wu L, Garg SK, Buyya R (2011) SLA-based resource allocation for software as a service provider in cloud computing environments. In: Proceedings of the 11th IEEE/ACM international symposium on cluster computing and the grid (CCGrid 2011), Los Angeles, USA, 23–26 May 2011
[11] Addis B, Ardagna D, Panicucci B (2010) Autonomic management of cloud service centers with availability guarantees. In: IEEE 3rd international conference on cloud computing, pp 220–227
[12] Chang F, Ren J, Viswanathan R (2010) Optimal resource allocation in clouds. In: IEEE 3rd international conference on cloud computing, pp 418–425
[13] Ferretti S, Ghini V, Panzieri F, Pellegrini M, Turrini E (2010) QoS–aware clouds. In: IEEE 3rd international conference on cloud computing, pp 321–328
[14] Lee YC, Wang C, Zomaya AY, Zhou BB (2010) Profit-driven service request scheduling in clouds. In: 10th IEEE/ACM international conference on cluster, cloud and grid computing, pp 15–249
[15] Mohd Yusoh Z, Tang M(2010) A cooperative co evolutionary algorithm for the composite SaaS placement problem in the cloud. In: Proceedings of the neural information processing, theory and algorithms, pp 618–625
[16] Underutilizing resources for HPC on clouds by J. Napper, P. Bientinesi, R. Iakymchuk Aachen Institute for Advanced Study in Computational Engineering Science Preprint: AICES-2010/06-1 10/June/2010
[17] Randles M, Lamb D, Taleb-Bendiab A (2010) A comparative study into distributed load balancing algorithms for cloud computing. In: IEEE 24th international conference on advanced information networking and applications workshops, pp 551–556
[18] Rodero I, Jaramillo J, Quiroz A, Parashar M, Guim F (2010) Energy-efficient application-aware online provisioning for virtualized clouds and data centers. In: International conference on green computing (GREENCOMP '10)
[19] Saure D, Sheopuri A, Qu H, Jamjoom H, Zeevi A (2010) Time-of-use pricing policies for offering cloud computing as service. In: IEEE SOLI 2010, pp 300–305
[20] Teng F, Magoulès F (2010) Resource pricing and equilibrium allocation policy in cloud computing. In: 10th IEEE international conference on computer and information technology (CIT 2010), pp 195–202
[21] Efficient Resource Provisioning in Compute Clouds via VM Multiplexing by Xiaoqiao Meng, Canturk Isci, Jeffrey Kephart, Li Zhang, Eric Bouillet Dimitrios Pendarakis IBM T. J. Watson Research Center Hawthorne *ICAC'10,* June 7–11, 2010, Washington, DC, USA.
[22] Yazir YO, Matthews C, Farahbod R (2010) Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. In: IEEE 3rd international conference on cloud computing, pp 91–98
[23] [Above the clouds: a Berkeley view of cloud computing by M. Armbrust, et al. Tech. Rep. CB/EECS-2009-28, EECS Department, U.C. Berkeley, Feb 2009.
[24] Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility by R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg, I. Brandic R. Buyya et al. / Future Generation Computer Systems 25 (2009) 599_616.
[25] Cloud Computing Resource Management through a Grid Middleware: A Case Study with DIET and Eucalyptus by Eddy Caron ,Frédéric Desprez ,David Loureiro and Adrian Muresan CLOUD '09 Proceedings of the 2009 IEEE International Conference on Cloud Computing.
[26] Resource provisioning for cloud computing by J. Wong, G. Iszlai, M.L. Ye Hu, Copyright © 2009 Ye Hu, Johnny Wong, Marin Litoiu and IBM Canada Ltd.
[27] Supporting Database Application as a Service by Zhou Yuan Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on March 29 2009-April 2 2009