# Resource Provision for Services Workloads based on (RPOA)

**Noha El.Attar[1], Wael Awad[2] and Fatma Omara[3]**

**[1] Ras-Elbar High Institute for Specific Studies and Computer, Ras El- Bar, Damietta, Egypt.**

**[2]Department of Math. and Comp. Science, Faculty of Science, Port Said University, Port Said, Egypt.**

**[3]Department of Comp. Science, Faculty of Computers and Information, Cairo University, Cairo, Egypt**

**Abstract**

Fulfilling the requirements of different applications and services in a real Cloud environment is extremely a big challenge. In this challenge the provision policies have to achieve the availability by allocating the appropriate resource to the customer services without any conflict in resource demands and with determining the right amount of required resources for the execution of services. According to the work in this paper, a Resource Provision Optimal Algorithm (RPOA) based on Particle Swarm Optimization PSO has been introduced and implemented to find the near optimal solution of resource allocation with minimizing both time and cost.

***Keywords****: Cloud Computing, Resources Provision, Particle Swarm Optimization.*

## 1. Introduction

There are many ways by which computational power and data storage facilities are provided to users, sometimes it is necessary to them to ask for additional resource when the local resources are not sufficient to meet their requirements. The user may need to locate a large number of computational resources that oblige him to contact several different resource providers in order to satisfy his requirements, but when the pool of resources is delivered, it is often heterogeneous[1]. Cloud computing environment solved this problem where Cloud computing is a new paradigm for hosting and delivering services on demand over the internet where users access services depending on their Quality of services (QoS) requirements regardless to where these services are hosted[2], or with the heterogeneous resources, only he cares about how much he will pay, and how much time is expected to provide the required hardware and software resources.

In a cloud computing environment, service providers are responsible for locating resources to user. Their traditional role is divided into two parts: the infrastructure providers who manage cloud platforms and lease resources according to a usage-based pricing model, and service providers who rent resources from one or many infrastructure providers to serve the end users.

Cloud computing delivers three application layers as services that are infrastructure, platform, and software as services, which are made available as subscription-based services in a pay-as-you-go model to consumers as follows:

1- ***Infrastructure as a Service (IaaS):*** where the providers rent the hardware and networking equipment used in supporting the customer's operations or business services. The client then pays on a per-use basis. With IaaS, the type of application the client runs does not matter – any application can be created or any shrink-wrapped software can be deployed on an IaaS platform[3].

2- ***Platform as a Service (PaaS):*** is providing an existent managed higher-level software infrastructure for building particular classes of applications and services. The platform includes the use of underlying computing resources, typically billed similar to IaaS products, although the infrastructure is abstracted away below the platform.

3- ***Software as a Service (SaaS):*** is providing specific, already-created applications as fully or partially remote services. Sometimes it is in the form of web-based applications and other times it consists of standard non remote applications with Internet-based storage or other network interactions[4].

Cloud computing system has main characteristics such as, hardware virtualization, dynamic provision, web service negotiation and economies of scale[5] which distinguish it from other distributed systems, some of these characteristics can be concluded as follows:

- Shared resource pooling: The infrastructure provider offers a pool of computing resources that can be dynamically assigned to multiple resource customers. Such dynamic resource assignment capability provides much flexibility to infrastructure providers to manage their own resource usage and operating costs.

- Geo-distribution and ubiquitous network access: Clouds are generally accessible through the Internet and using the Internet as a service delivery network. Hence any device with Internet connectivity is able to access cloud services.
- Service oriented: In a cloud, each IaaS, PaaS and SaaS provider offers his service according to the Service Level Agreement (SLA) which is a deal between providers and customers.
- Dynamic resource provisioning: Computing resources can be obtained and released on the fly. Compared to the traditional model that provisions resources according to peak demand, dynamic resource provisioning allows service providers to acquire resources based on the current demand, which can considerably lower the operating cost.
- Self-organizing: Since resources can be allocated or deallocated on-demand, service providers are empowered to manage their resource consumption according to their own needs. Furthermore, the automated resource management feature yields high agility that enables service providers to respond quickly to rapid changes in service demand such as the flash crowd effect.
- Utility-based pricing: Cloud computing adopts a pay-per-use pricing model. Utility-based pricing lowers service operating cost as it charges customers on a per-use basis[6].

This paper's main problem is the need for over provisioning of services to meet potential peaks in demand. These peaks can be considerably reduced in favor of providing resource allocations dynamically according to the overall application workload. However, it is still needed to define rules by which the service should be scaled. These rules must depend on user requirements which is stated as QoS conditions to enforce the rules accordingly. So, the resource provisioning system controls how multiple services share the platform of cloud system. Some challenges face a resource provision system as, 1) the different scheduling needs which are based on the requests and available resources, 2) the solution must be highly scalable, as the framework contains thousands of nodes and there are hundreds of jobs with millions of tasks active at a time, 3) the scheduling system must be fault-tolerant and highly available, as all the applications in the cluster depend on it, so the main challenge is to minimize user response time and minimize resource usage cost.

Finally the rest of this paper is organized as follows. Section 2 presents the related work on resource provision problem in cloud computing, section 3 handles and discusses the provision problem in more details. Section 4 states the problem definition, and section 5 represents the model implementation and evaluation.

## 2. Related Work

In order to handle huge numbers of users' applications all over the world, Cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to achieve availability and ensure reliability in case of site failures. For example, Amazon has data centers in the US (e.g., one in the East Coast and another in the West Coast) and Europe [7]. However, these applications have some constraints as (1) the Cloud customers (i.e., SaaS providers) have to decide where they prefer the location of services to be hosted but it is difficult for them to determine in advance the best location for hosting them. (2) they don't provide automatic mechanisms for scheduling customer services across multiple geographically distributed data centers, so Cloud providers may not be able to meet QoS expectations of their service- customers originating from multiple geographical locations. This necessitates building mechanisms for seamless federation of data centers of a Cloud providers to support dynamic scaling of applications across multiple domains in order to meet QoS targets of Cloud customers[7].

Byun; E., et. al, have suggested an architecture for the automatic execution of large scale workflow-based applications on dynamically and elastically provisioned computing resources. Especially, this research focuses on an algorithm named PBTS (Partitioned Balanced Time Scheduling) which estimates the minimum number of computing hosts required to execute a workflow within a user-specified finishing time. The main goal of this research is to minimize the resource cost, not the makespan of workflow [8]. Another trend in scheduling resource is to satisfy a minimum response time. Iqbal; W., et. al, have proposed a methodology and presented a working prototype system for automatic detection and resolution of bottlenecks in a multi-tier web application hosted on a Cloud in order to satisfy specific maximum response time requirements. Automatic bottleneck detection and resolution under dynamic resource management has the ability to enable Cloud infrastructure providers to provide SLAs for web applications that guarantee specific response time requirements. There are some limitations to this work. They only address scaling of the web server tier and a read-only database tier. This system did not address software configuration management [9].

Stillwell; M., et. al., have defined the resource allocation problem for a static workload of services that are fully contained in a single VM instance; this definition accounts for multiple resource dimensions, supports a mix of best-effort and QoS scenarios trying to promote performance, fairness and high resource. Algorithms that are used for solving this base problem are Exact solution, Greedy algorithms, Genetic algorithm, Vector packing algorithms [10].

Almeida et. al., have presented a self-managing technique that jointly addresses the resource allocation and admission control optimization problems in virtualized servers. Resource allocation and admission control represent key components of an autonomic infrastructure and are responsible for the fulfillment of service level agreements. The solution is designed considering the provider's revenues and the cost of resource utilization, and customers' QoS requirements and specified in terms of the response time of individual requests. Results show that this solution can satisfy QoS constraints while still yielding a significant gain in terms of profits for the provider, especially under high workload conditions, if compared to the alternative methods. Moreover, it is robust to service time variance, resource usage cost and workload mispredictions [11].

Other researchers such as , Islam et. al., have developed prediction-based resource measurement and provisioning strategies using Neural Network and Linear Regression to satisfy upcoming resource demands. This prediction framework uses statistical models which are able to speculate the future surge in resource requirement; thus enables proactive scaling to handle temporal bursty workload in a controllable way [12].

Chen and Tsai have presented a version of Discrete Particle Swarm Optimization (DPSO) algorithm for tasks allocation. They use the heuristic to minimize the total cost of application tasks execution on Cloud Computing environments. Chen and Tsai claim that their proposed DPSO algorithm is faster than mathematical methods [4].

Pandey et. al, have focused on minimizing the total execution cost of applications on Cloud service providers' resources, such as Amazon and GoGrid3. This research presents a Particle Swarm Optimization (PSO) based heuristic for scheduling applications services of Cloud resources that try to consider both computation cost and data transmission cost. By comparing the cost savings when using PSO and existing 'Best Resource Selection' (BRS) algorithm, the results show that PSO can achieve as much as 3 times cost savings as compared to BRS, and good distribution of workload onto resources [13].

## 3. Resource Provision Problem

The job scheduling to the computing resources is an NP-complete problem, even in two simple cases: (1) scheduling jobs with uniform weights to an arbitrary number of processors and (2) scheduling jobs with weights equal to one or two units to two processors [13].

The resources in the Cloud infrastructure layer have different types;

1-Hardware resources, e.g. computing power, storage, and machine provisioning.

2-Software resources, e.g. middleware and development resources.

3-Application resources. For example, Google has used Cloud Computing platform to offer Web applications for communication and collaboration [14].

Every required application in Cloud has different configuration and requirements. These requirements are scheduled in QoS that must be met by the resource allocation policies and application scheduling algorithms.

Fulfilling the requirements of different applications and services in a real Cloud environment is extremely a big challenge because of some reasons as: (i) Clouds exhibit varying demand, supply patterns, and system size; and (ii) users have heterogeneous and competing QoS requirements [15]. (iii) Cloud providers have to achieve the availability by allocating the appropriate resource to the services without any conflicting in resource demands and with determine the right amount of resources required for the execution of services to minimize the cost from the perspective of users and maximize the resource utilization from the perspective of resource providers[8].

Another shape of services requirements is Service-Level Agreements (SLAs) that is a deal between customer and Cloud providers, that is provided for availability or other quality attributes [16].

Guaranteeing response time is another difficult problem faces the provision polices because of the highly dynamic of application traffics and difficultly of accurate prediction [9]. and also it not addressed by SLA.

For this reason, the dynamic coordination and provision of distributed resources rapidly draw attraction from scientists. Some notable achievements are the resource virtualization and provisioning technologies such as CoD (Cluster on-demand) [17], Virtual Grid [18], Eucalyptus [19], and IaaS Cloud such as Amazon's EC2 (Elastic Compute Cloud). [8]

## 4. Problem Definition

The main goal of the provisioning policy is how to spread the application load on convenient Cloud resources to achieve the optimization objective of satisfying customers' QoS requirements (i.e. minimizing both response time and cost of resource utilization and, in the same time, maximizing the provider profit). The Cloud computing service provider's profit is achieved by providing high-quality services to the users through the efficient allocating of the resources on demand [20].

The main processes of the resources provision in the cloud computing is depicted in figure 1. the allocation algorithm will consider the decision, scheduling, and the allocation processes. According to the work in this paper, a Resource Provision Optimal Algorithm (RPOA) has been proposed based on particle swarm optimization to minimize user response time, as well as the resource usage cost which are considered contracted to each other

(i.e., user response time can be decreased by increasing the computing resources while the cost may be reduced by using fewer resources that will increase the time).

PSO has become popular due to its simplicity and its effectiveness in wide range of application with low computational cost [13]. Some of the applications that have used PSO are: data mining [21], pattern recognition and environmental engineering [19]. The PSO has also been applied to solve NP-Hard problems like Scheduling [22] and task allocation [23]. According to RPOA, defining the proper way to determine the suitable amount of required resources will be considered as discussed in the next section.

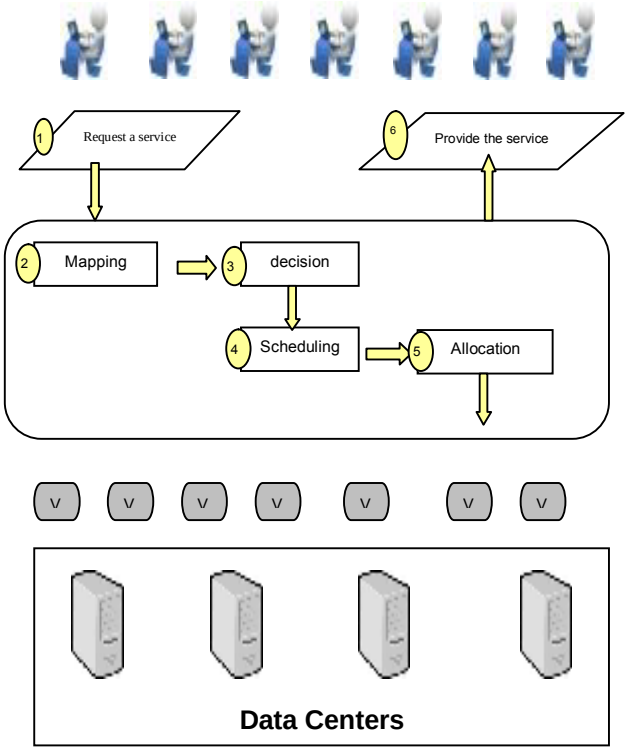


Fig. 1: Abstracted diagram to resource provision process

## 5. The RPOA Principles

The main principle of scheduling policies is to consider resources' prices as well as user's available budget and deadline time which is known as market-oriented scheduling policies[24].

According to the RPOA algorithm, the data centers are distributed over different resources pools. Each pool includes a specific resource type (e.g., computing and storage resources). Each computing resource is associated with the available number of hosts and a certain size of memory and has a certain power consumption of computing. Similarly each task workload is associated with the number of subtasks which require a certain amount of computing power or storage size depending on the workload type. According to the RPOA algorithm implementation, we consider that the work loads have the same priority, and in the same time, are independent. By considering independency of workloads, the measurement of the resources cost will be more accurate. Another important consideration is to allow customers to decide how much they can pay according to the amount of usage they need depending on their available budget.

### 5.1 The RPOA Algorithm Environment

Assuming that there are 'm' number of available resources, and 'n' workloads that contain '*j*' of subtasks. We consider some principle assumptions; fixed quantity of resources '$Q$', and each computing resource has a definite price '$p_j$' denoted by dollar and has default execution time '$t_j$' denoted by seconds. Every task *i* has a set of subtasks *j* that need a specific resource quantity denoted by '$q$'. Every customer can decide the price of each task that can be paid '$bp_j^i$'. Another constraint is to ensure that for all workloads, the available resources must not be less than the total amount of required demands.

Generally, the resources allocation problem can be stated as there are '*n*' workloads that have to be allocated to '*m*' different compute resources. So, the PSO particles will be represented as '*n*' dimensional vector that present by the position and the velocity vectors denoted orderly by $x_{ij}^k = (x_{i1}, x_{i2,}.....,x_{in})$ to denote the position, and $v_{ij}^k = (v_{i1}, v_{i2,}.....,v_{in})$ denoting the velocity of $j^{th}$ dimension of $i^{th}$ particle in $k^{th}$ generation where $j^{th}$ dimension of the particles are the number of tasks in a workflow.

Updating position and velocity is the way to move toward the best position of the particle, (see Figure. 2) [25], these updates are calculated by the following equations 1, 2[26].

$$v_{ij}(k+1) =$$

$$wv_{ij} + w_1 r_1[pbest - x_{ij}(k)] + w_2 r_2[gbest - x_{ij}(k)] \quad (1)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1) \quad (2)$$

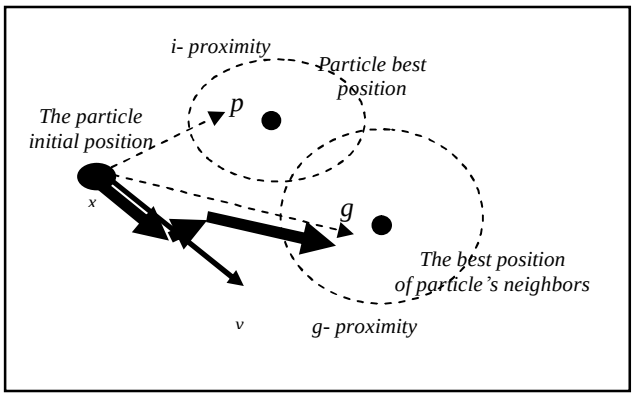


Fig. 2: the particle movement in PSO

The fitness function will be used to evaluate each particle by deciding which resource is suitable for every task. According to the RPOA algorithm, the utilization function has been introduced to define the fitness function.

## 5.2 Utilization Function

Assuming that N users ask for services and each service has a sequence of subtasks that have to be completed, a vector $Q_m$ of resources' fixed capacity $Q=[Q_1,Q_2,Q_3,\ldots\ldots Q_m]$ and a matrix $q_{nm}$ of amount, which is required by every task, are needed to be defined. Two calculation processes are needed, one for calculating spent time of every subtask on each resource according to equation 3, and the other is for calculating every paid and total cost for every subtask according to equation 4,5.

$$t_j^i = \frac{q_i}{Q_m} * t_i \quad (3)$$

$$c_j^i = bp_j^i * t_j^i \quad (4)$$

$$C_i^* = \sum_{i=m} p_j * t_j^i \quad (5)$$

Where $c_j^i$ is the paid cost for every subtask, and $C_i^*$ is the main total cost of each individual task.

Therefore, the utilization function is a relational equation between time and cost which is can be calculated based on elasticity parameters between cost and time as defined in equation 6 [27]:

$$\ell_c = \frac{\Delta c}{\Delta t}, \text{ and } \ell_t = \frac{\Delta t}{\Delta c}. \quad (6)$$

The pseudo code of the RPOA algorithm is depicted as follows:

Set n as number of tasks
Set m as number of resources
**Initialization of resources
For each resource j to m
Set resource's quantity vector to $Q_j = [Q_1, Q_2, \ldots\ldots, Q_m]$
Set resource's price vector to $p_j = [p_1, p_2, \ldots\ldots\ldots p_m]$
Set execution time vector $t_j = [t_1, t_2, \ldots\ldots\ldots t_m]$
Next
**PSO Algorithm
Initialize first particle to first task
Initialize position vector and velocity vector randomly
Set pbest vector to first particle position
Set gbest vector of the population
Set i=2, k=2
Repeat
For each particle i in the population
Update $v_{ij}$ by equation 1
Update $x_{ij}$ by equation 2
Call fitness function(fitness_value)
If fitness_value of $k^{th}$ generation is better than fitness_value of $k^{th-1}$ iteration then
Set pbest to the new position of particle i
Else
Ignore the new fitness_value
Next
Update gbest of $k^{th}$ generation
K=k+1
Update $w = w * 0.01$
Until (termination by reach best fitness value)

Fitness function( )
For each task i
For each resource j
Read the required quantity $q_{ij}$ of every subtask
Read the main price of every resource $bp_{ij}$
Calculate spent time by Eq. 3
Calculate demand cost by user Eq. 4
Next j
Calculate main total cost of each task by Eq. 5
Calculate elasticity parameters to find fitness function by Eq. 6
Next i

## 6. RPOA Algorithm Implementation

Suppose that a task contains four subtasks all these subtasks need 260 GB storage memory. The available resources are 500 GB of storage memory. Four vectors are used in storing the values for: 1) fixed quantity of

each resource (RQ-vector), 2) fixed computation cost of each resource (RP-vector), 3) resource computation processing time (RT-vector), and 4) user available budget for every task (BP-vector). The values for (Tq – matrix) resemble the quantity required by every task (see table [1]).

Table1. available resources and required tasks

| Resources | Memory (GB) | Executed Main Cost | Computation Processing Time | Required computation quantity (GB) | Available budget |
|---|---|---|---|---|---|
| R1 | 150 | 2 $ | 10 seconds | 50 | 1.9 |
| R2 | 100 | 1.5 $ | 15 seconds | 80 | 0.8 |
| R3 | 100 | 1.3 $ | 13 seconds | 30 | 0.4 |
| R4 | 150 | 1.7 $ | 9 seconds | 100 | 1.3 |

## 7. Performance Evaluation

The total cost and time of the workflow execution, that is randomly distributed 16 times on the available resources to find the optimal distribution map, are computed using two algorithms. The first algorithm is to be achieved by applying utilization function only while the other is to be achieved by applying PSO with fitness function based on the utilization function.

Table 2 shows the randomly sixteenth distribution attempts over resources with their utilization values.

Table 2. distribution of a workflow on available resources

| Number of distribution | Randomly distribution | Utilization value |
|---|---|---|
| 1 | 100 50 30 80 | 3.23 $/sec |
| 2 | 80 100 50 30 | 3.64 $/sec |
| 3 | 50 80 30 100 | 3.36 $/sec |
| 4 | 30 50 100 80 | 3.7 $/sec |
| 5 | 100 80 50 30 | 3.47 $/sec |
| 6 | 30 100 80 50 | 3.75 $/sec |
| 7 | 50 100 80 30 | 3.75 $/sec |
| 8 | 80 30 50 100 | 3.2 $/sec |
| 9 | 100 30 80 50 | 3.4 $/sec |
| 10 | 50 30 100 80 | 4.6 $/sec |
| 11 | 50 100 30 80 | 3.6 $/sec |
| 12 | 100 50 80 30 | 3.49 $/sec |
| 13 | 80 100 30 50 | 3.62 $/sec |
| 14 | 80 30 100 50 | 3.3 $/sec |
| 15 | 30 100 50 80 | 3.32 $/sec |
| 16 | 30 50 80 100 | 3.31 $/sec |

As shown in table 2- the highlighted rows are the minimum utilization values of distribution that satisfy both minimum cost and time using utilization function. By using PSO with utilization fitness function, the number of iterations has been decreased up to four to reach the nearest best distribution map, as shown in Figure 3.

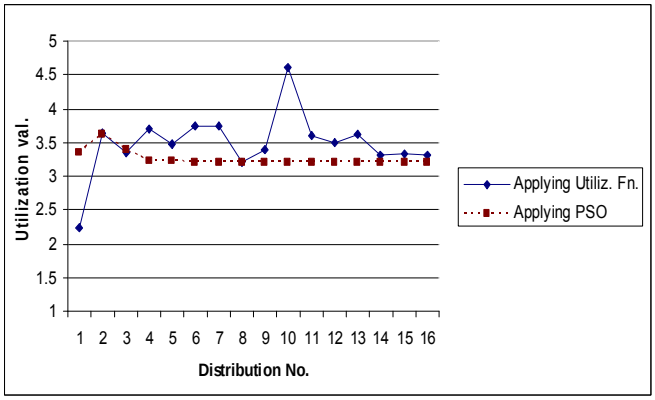


Fig. 3: Performance Evaluation for applying PSO with Utilization fn.

## Conclusions

The goal of this paper is to find a Workload- Resource map WM that is commensurate with customer budget and suitable for deadline time. Accordingly, maximizing the performance of computing resource can be achieved by allocating its capacity for the maximum number of workloads. To achieve this the Particle Swarm Optimization(PSO) algorithm is used with the utilization function in order to find the nearest optimal solution to our resource allocation problem. The results show that using PSO provides better distribution maps than that using utilization function only, because the number of iterations that reach the nearest best optimal value has been decreased.

## References

[1] Nurmi; D., et.al, "The Eucalyptus Open-source Cloud-computing System", CCGRID '09 Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society Washington, DC, USA, 2009.

[2] Zhang; Q. , "Cloud computing: State-Of-The-Art and Research Challenges", J of Internet Services and Applications, Vol. 1, No. 1, 2010, pp 7–18.

[3] Arinze; B., Anandarajan; M., "Factors that Determine the Adoption of Cloud Computing:A global Perspective", International Journal of Enterprise Information Systems, Vol. 6, No. 4, 2010, pp. 55-68.

[4] Chen; Y., Tai; S., "Optimal Provisioning of Resource in a Cloud Service", IJCSI, Vol. 7, No. 6, 2010, pp. 95-99.

[5] Teng; F., Magoules; F., "Future of Grids Resources Management. In: Fundamentals of Grid Computing: Theory, Algorithms and Technologies", Chapman & Hall/CRC, Boca Raton,2009, pp. 133–153,.
[6] Silva, F. et al., "Application Execution Management on the InteGrade Opportunistic Grid Middleware", Journal of Parallel and Distributed Computing, Vol. 70, No. 5, 2010, pp. 573- 583.
[7] Buyya; R., et.al., "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services", 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010, Busan, South Korea, LNCS, Springer, Germany, May 21-23, 2010, LNCS 6081, pp. 13–31.
[8] Byun; E., et. al, "Cost Optimized Provisioning of Elastic Resources for Application Workflows", Future Generation Computer System, Vol. 27, No. 8, 2011, pp. 1011-1026.
[9] Iqbal; W., et. al, "Adaptive Resource Provisioning for Read Intensive Multi-tier Applications in the Cloud", Future Generation Computer Systems, Future Generation Computer Systems, Vol. 7, No. 6, 2011, pp. 871-879.
[10] Stillwell; M., et. al, "Resource Allocation Algorithms for Virtualized Service Hosting Platforms", Journal of Parallel and Distributed Computing, Vol. 70, No. 9 , 2010, pp. 962-974.
[11] Almeida; J., et..al. , "Joint Admission Control And Resource Allocation In Virtualized Servers", Journal of Parallel and Distributed Computing, Vol. 70, No. 4, 2010, pp. 344-362.
[12] Islam; S., et. al, "Empirical Prediction Models for Adaptive Resource Provisioning in the Cloud", Future Generation Computer Systems, Vol. 28, No. 1, 2012, pp. 155–162.
[13] Pandey; S., et. al, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", AINA '10 Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, IEEE computer Society, Perth, Australia, 20-13 April 2010, pp. 400-407.
[14] Zhang; L., et.al., "CCOA: Cloud Computing Open Architecture", IEEE International Conference on Web Services, IEEE computer Society, 2009, Vol. 20, No. 1520, pp. 607-616.
[15] Buyya; R, et. al., "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities", proceeding of the 7th High Performance Computing and Simulation Conference (HPCS 2009), Leipzig, Germany, June 21-23, 2009.
[16] Ardagna; D., et. al., "Service Provisioning on the Cloud: Distributed Algorithms for Joint Capacity Allocation and Admission Control", ServiceWave 2010, LNCS 6481, 2010, p.p 1-12.
[17] Irwin; D., et. al., "Sharing Networked Resources with Brokered Leases", Proceedings of the annual conference on USENIX '06 Annual Technical Conference, USENIX Association Berkeley, CA, USA, 2006, pp. 199-212.
[18] Kee; Y., et. al, "Efficient Resource Description and High Quality Selection for Virtual Grids", Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid, Vol. 1, IEEE Computer Society Washington, DC, USA, 2005, pp. 598-606.
[19] Lu; W., et. al., "Analysis of Pollutant Levels in Central Hong Kong Applying Neural Network Method with Particle Swarm Optimization". Environmental Monitoring and Assessment, Vol. 79, No. 3, 2002, pp. 217–230.
[20] Zhou; H, "Dynamic Resource Provisioning for Interactive Workflow Applications on Cloud Computing Platform", proceeding of: Methods and Tools of Parallel Programming Multicomputers - Second Russia-Taiwan Symposium, MTPP 2010, Vladivostok, Russia, May 16-19, 2010, pp. 116-125.
[21] Sousa; T., et. al., "Particle Swarm based Data Mining Algorithms for Classification Tasks", Parallel Computing, Vol. 30, No. 5-6, 2004, pp. 767–783.
[22] Yu; B., et. al., "Short-Term Hydro-Thermal Scheduling using Particle Swarm Optimisation Method", Energy Conversion and Management, Vol. 48, No. 7, 2007, pp.1902–1908.
[23] Zavala; A., et. al., "Constrained Optimisation with An Improved Particle Swarm Optimisation Algorithm", Intl. Journal of Intelligent Computing and Cyvernetics, Vol. 1, No. 3, pp. 425–453, 2008.
[24] Salehi; M., Buyya; R., "Adapting Market-Oriented Scheduling Policies for Cloud Computing",
Proceedings of the 10th international conference on Algorithms and Architectures for Parallel Processing, May 21-23, 2010, Vol. Part I, pp. 351-362.
[25] Eberhart; R., Kennedy; J., "Particle Swarm Optimisation:AMiniTutorial",http://www12.informatik.uni-erlangen.de/edu/OC/SS09/pso-tutorial.pdf, visited on may -2-2012, 11 pm., 2004.
[26] Omnwunalu; J., Durlofsky; L, "Application of Particle Swarm Optimization Algorithm for Determining Optimum Well Location and Type" , Compute Geosci, Vol. 14, No. 1, 2010, pp. 183-198.
[27] Png; I., Cheng; C., Managerial economics, http://www.comp.nus.edu.sg /~ipng/mecon/sg/03elas_sg.pdf, 2001, visited on may- 3-2012, 12:30 am.
[28] Li; C., "Cloud Computing System Management Under Flat Rate Pricing", Journal of Network and Systems Management, Vol. 19, No. 3, 2011, pp. 305-318.

[29] Song; N, Jim; C., "Adaptable Scheduling Schemes for Scientific Applications on Science Cloud", IEEE International conference on cluster computing 2010, Heraklion, Crete, Greece, 20-24 September 2010.
[30] Tayal; S., et. al., "Tasks Scheduling Optimization for the Cloud Computing Systems", International Journal Of Advanced Engineering Sciences And Technologies (IJAEST), Vol. 5, No. 2, 2011, pp. 111-115.
[31] Varvarigou; T., et. al., "A Study on the Effect of Application and Resource Characteristics on the QoS in Service Provisioning Environments", International Journal of Distributed Systems and Technologies, Vol. 1, No. 1, 2010, pp. 55-75.
[32] Zhan; J., et. al., "PhoenixCloud: Provisioning Resources for Heterogeneous Workloads in Cloud Computing", The First Workshop of Cloud Computing and its Application, CCA08, Chicago 2008, Vol. 2, published in 2010.