

**International Journal of Innovative Research in Science, Engineering and Technology**  
*An ISO 3297: 2007 Certified Organization* *Volume 4, Special Issue 5, April 2015*  
**International Conference On Emerging Trends in Engineering and Technology (ICETET'15)**  
**On 13<sup>th</sup> & 14<sup>th</sup> March 2015**

**Organized by**

**Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India**

# Resource Allocation Model in Cloud Computing Using Mixed Integer Linear Program

S. Akilandeswari<sup>1</sup>, R. Dhanalakshmi<sup>2</sup>

PG Student, P.G.P College of Engineering and Technology, Namakkal, India<sup>1</sup>

Assistant Professor, Department of CSE Department, P.G.P College of Engineering and Technology, Namakkal, India<sup>2</sup>

**ABSTRACT:** In base paper work a cost tradeoff between buyer and seller is achieved in an efficient way. In cost tradeoff, an obstacle lies in the uncertainty of computational resource cost. In spot market, the cost of the resource is fluctuating all the time depending on the resource supply and demand levels. To address this problem, develop an optimization model using a mixed integer linear program, to generate optimal rental decisions for given cost parameters. The systematically analyze the predictability of the time-varying spot instance prices and find that the best achievable prediction to provide a close approximation to the actual prices.

**KEYWORDS:** Resource allocation, auction, bundle, virtualization.

## I. INTRODUCTION

Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles. The main enabling technology for cloud computing is virtualization. With operating system level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on demand.

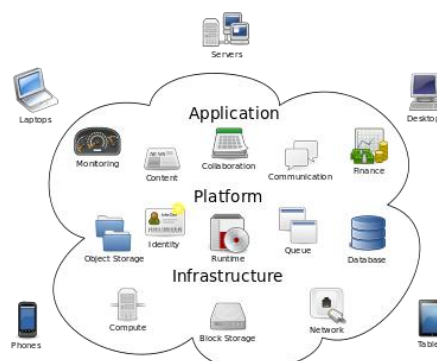


Fig.1 Cloud Computing

By minimizing user involvement, automation speeds up the process, reduces labour costs and reduces the possibility of human errors. Users routinely face difficult business problems. Cloud computing adopts concepts from

## International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization

Volume 4, Special Issue 5, April 2015

International Conference On Emerging Trends in Engineering and Technology (ICETET'15)

On 13<sup>th</sup> & 14<sup>th</sup> March 2015

Organized by

Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India

service oriented architecture (SOA) that can help the user break these problems into services that can be integrated to provide a solution. Such metrics are at the core of the public cloud pay-per-use models. In addition, measured services are an essential part of the feedback loop in autonomic computing, allowing services to scale on-demand and to perform automatic failure recovery.

### II. PROBLEM DESCRIPTION

Computational Grids and Cloud enable the coordinated and aggregated use of geographically distributed resources, often owned by autonomous organizations, for solving large-scale problems in science, engineering, and commerce. Resource management systems need to provide mechanisms and tools that realize the goals of both service providers and consumers. The resource consumers need a utility model, representing their resource demand and preferences, and brokers that automatically generate strategies for choosing providers based on this model. Further, the brokers need to manage as many issues associated with the execution of the underlying application as possible.

Grid and cloud computing have many similarities in terms of features and functions. Both paradigms are used primarily for enhancing the utilization of the available resources. Grid computing, coined in the 1990s by Ian Foster and Carl Kesselman, was briefly defined as a coordinated resource sharing and problem solving technology that spans many dynamic, 58 multi-institutional virtual organizations.

Although grid and cloud computing share similarities in terms of optimizing the availability of resources, they are different in the method of the tasks involved in the computing. Grid is mostly used for high-intensive computational requirements, while cloud can be used for a variety of services that involve high CPU usage, storage and bandwidth. Cloud computing has more diversity with respect to providing services. As a result, cloud computing appears to be more complex than grid computing.

### III. EXISTING SYSTEM

In a cloud environment, providers compete to offer their services, while users compete to receive resources to run their tasks on the basis of their Quality of Service (QoS) and pricing requirements. Usually, providers and users are independent in their policies and resource allocation strategies, using market-based techniques and economic models to regulate the supply and demand for resources, providing an incentive for providers and motivating users to select the least cost service provider based on their budget and level of QoS. The crucial aim of the resource allocation procedure is to find an allocation that is feasible and optimal for a given service. Due to the heterogeneous nature of cloud computing, resource allocation is often based on two approaches, namely, system centric and user centric.

The system-centric approach, which is the traditional approach, optimizes system-wide measures of performance, such as the overall throughput of the system. The user-centric approach concentrates mainly on providing maximum utilization for users on the basis of QoS requirements. The dynamic behavior in turn provokes challenges between users and service providers to maximize their required resources or tasks. Therefore, the resource allocation in this market environment must be optimal.

#### DISADVANTAGES

1. Providers have advantage over users.
2. Resources are limited.
3. Cost is high and resources allotted for paid cost is low.
4. Allocated resources are not efficient to get the job done.

### IV. PROJECT DESCRIPTIONS

#### A. Advertising and resource discovery

Advertising resources and services: cloud providers sent messages to CIS about their offered services and resources. Resource and service discovery: for each user task, a broker sent the message to CIS about the requirements

**International Journal of Innovative Research in Science, Engineering and Technology**

An ISO 3297: 2007 Certified Organization

Volume 4, Special Issue 5, April 2015

**International Conference On Emerging Trends in Engineering and Technology (ICETET'15)**

**On 13<sup>th</sup> & 14<sup>th</sup> March 2015**

**Organized by**

**Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India**

for resource discovery. The message requested CIS to create a list of resources and services that matched with the users requirements. The CIS returned a list of resources that match with the user's requirements.

**B. Generate bundles**

$b_i, i = 1, 2, \dots, m$ , was responsible to:

- Generate a combination of resources (a bundle).
- Generate a price value within the user's budget for each combination of resources.
- Generate a time value for each combination of resources.
- Send bids to an auctioneer.

$d_j, j = 1, 2, \dots, n$ , is responsible to:

- Generate a combination of resources (a bundle).
- Generate a price value for each combination of resources.
- Send bids to an auctioneer.

**C. Informing the end of an auction**

An auctioneer sent a message to the brokers and to the cloud providers to inform them about the close of the bidding.

**D. Winner determination**

Our model was generalized by redefining M for the users and for the cloud providers. Because was used only for the virtual machine, it redefined M to make it applicable to different types of resources.  $w_i$  was the number of CPUs. In this model,  $w_i$  was changed to a specific attribute of a requested item and was optional, such as a number that represents the CPU processing power of the CPU, the storage capacity, the bandwidth, or the memory. Because users/cloud providers request/offer different items with different quantities and attributes, the sort operation would be based on their bid, quantity and attribute of resources.

The allocation scheme with a bid density of  $M_p$  approximated the optimal allocation Therefore, users were ranked in descending order based on their bid density and time because the cloud providers were willing to sell their services to long-term users. The assumption was to consider "time" as a parameter for sorting the brokers by giving such users a higher priority. In fact, brokers who needed resources for a long time were placed at the top of the sorted list, and cloud providers were ranked in ascending order of their bid density.

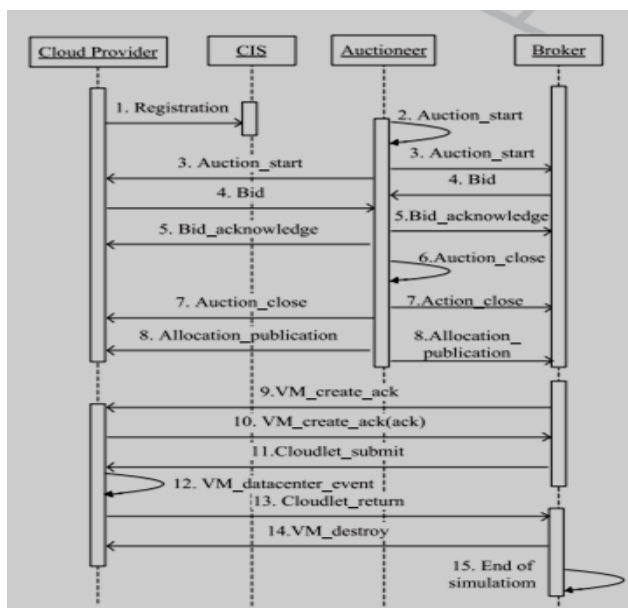


Fig. 2 The flow of communication among the entities

**International Journal of Innovative Research in Science, Engineering and Technology**

An ISO 3297: 2007 Certified Organization

Volume 4, Special Issue 5, April 2015

**International Conference On Emerging Trends in Engineering and Technology (ICETET'15)**

**On 13<sup>th</sup> & 14<sup>th</sup> March 2015**

**Organized by**

**Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India**

**E. Mixed Integer Problem**

In the context of cloud computing, it present the complete formulation for the Deterministic Resource Rental Planning (DRRP) as follows:

$$\text{Min} \sum_{i \in I} \sum_{t \in T} C^+ f(t) \Phi_i - ai; t + (Cs(t) + Cio(t)) \beta_i; t + C^- f(t) D(i, t) + Cp(i, t) \chi_i; t \tag{1}$$

$$\beta_i; t - 1 + ai; t - \beta_i; t = D(i, t), i \in I, t \in T \tag{2}$$

$$P(i) \Phi_i - ai; t \leq Q(i, t), i \in I, t \in T \tag{3}$$

$$ai; t \leq B \chi_i; t, i \in I, t \in T \tag{4}$$

$$\beta_i; 0 = \varepsilon, i \in I \tag{5}$$

$$ai; t, \beta_i; t \in \mathbb{R}^+, i \in I, t \in T \tag{6}$$

$$\chi_i; t \in \{0, 1\}, i \in I, t \in T \tag{7}$$

Note that it have not considered I/O and storage cost for input data in the objective function (1). This is because it choose to fetch input data into the cloud on the fly to complete the application. Another option is to transfer all input data once and store them in cloud throughout the planning horizon. The choice on which option is better depends on the specific data access pattern and the duration of planning. This transfer/storage tradeoff problem was noticed in their investigation on running scientific workflow applications. Its imply specifies that customer demand should be satisfied at any time slot.

At slot  $t$ , the data stored at the previous timeslot  $\beta_i; t-1$ , and the data generated in the current slot  $ai; t$ , are combined together to serve customer demand emerged in the current time slot, i.e.,  $\beta_i; t-1 + ai; t - D(i, t)$ . The over provisioning amount becomes the storage amount  $\beta_i; t$  at the end of  $t$  to serve future demand. The initial storage amount is set to be some constant  $\varepsilon$  in constraint (5), depending on the specific planning scenario.

Next, let  $P(i)$  be the average bottleneck resource consumption rate for one class- $i$  instance, and let  $Q(i, t)$  denote the bottleneck resource available for one class- $i$  instance in slot  $t$ , constraint (3) ensures that the data generation does not exceed the available bottleneck resource. For example, for data-intensive applications, I/O bandwidth becomes the bottleneck resource.

During slot  $t$ , the total output data at class- $i$  instance cannot exceed the aggregate throughput volume on that instance. Constraint (4) is often referred to as the forcing constraint. It states that there will be no data generated in slot  $t$  if no rental decision is made ( $\chi_i; t = 0$ ).  $B$  is set to be a very large constant that exceeds the maximum possible value of  $ai; t$ . Finally, constraints (6) and (7) specify domains of the variables. The formulation of DRRP is a mixed integer linear programming (MILP) problem that is NP-complete in nature. With reasonable input size, this problem can be solved using the branch-and-bound (B&B) method in most optimization software packages.

**F. Allocation of the task and payment**

A broker who won in an auction would send the tasks to the related cloud provider. A cloud provider performed these tasks and would send a message to the broker about the executed tasks. A broker sent the payment to a cloud provider.

**V. CONCLUSION**

The present system proposes mixed integer linear program approach for improving resource usage by utilizing cost-effective cloud services under the uncertain prices. This approach systematically analyze the predictability of the time-varying spot instance prices and find that the best achievable prediction to provide a close approximation to the actual prices. The proposed optimization model is based on a thorough rental cost analysis of running elastic applications in cloud. Considering the cost tradeoff between data generation and storage, it developed a deterministic optimization model that minimizes the unit rental cost of covering customer demand over a planning horizon.

This model works well with deterministic cost parameters but not suitable for the emerging spot instance market in cloud computing. Based on this observation, we further designed a stochastic optimization model that seeks to minimize the expected resource rental cost given the presence of spot price uncertainty. Simulations based on realistic settings clearly demonstrate the advantages of our proposed optimization solutions in rental cost reduction.

**International Journal of Innovative Research in Science, Engineering and Technology**

*An ISO 3297: 2007 Certified Organization*

*Volume 4, Special Issue 5, April 2015*

**International Conference On Emerging Trends in Engineering and Technology (ICETET'15)**

**On 13<sup>th</sup> & 14<sup>th</sup> March 2015**

**Organized by**

**Pandian Saraswathi Yadav Engineering College, Arasanoor, Sivagangai, Tamilnadu, India**

**REFERENCES**

- [1] Y. Li, W. Zhang, Y. Zhang, X. Zhang, X. Xiong, Calibration of the agent-based continuous double auction stock market by scaling analysis, *Inf. Sci.* (2012).
- [2] C. Jiang, L. Duan, C. Liu, J. Wan, L. Zhou, VRAA: virtualized resource auction and allocation based on incentive and penalty, *Cluster Comput.* (2012) 1–12.
- [3] X. Wang, J. Sun, M. Huang, C. Wu, A resource auction based allocation mechanism in the cloud computing environment, in: 26th IEEE International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012, pp. 2111–2115.
- [4] X.-W. Wang, X.-Y. Wang, M. Huang, A resource allocation method based on the limited English combinatorial auction under cloud computing environment, in: 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2012, pp. 905–909.
- [5] P. Samimi, A. Patel, Review of pricing models for grid & cloud computing, in: IEEE Symposium on Computers & Informatics (ISCI), 2011, pp. 634–639.
- [6] S. Zaman, D. Grosu, Combinatorial auction-based allocation of virtual machine instances in clouds, in: Second IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2010, pp. 127–134.
- [7] S. Shang, J. Jiang, Y. Wu, Z. Huang, G. Yang, W. Zheng, DABGPM: a double auction bayesian game-based pricing model in cloud market, *NetworkParallel Computing* (2010) 155–164.
- [8] W.-Y. Lin, G.-Y. Lin, H.-Y. Wei, Dynamic auction mechanism for cloud resource allocation, in: 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010, pp. 591–592.