ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 3 (July - Sept. 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Market Place Model for Cloud Computing

Kamta Giri
Ph.D in CSE, Research Scholar of Sai Nath University, Ranchi, Jharkhand, India

Abstract
The cloud is designated as a three-tier structure, specifically Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) from low-layer to high-layer. In recent. In this paper we aim to propose a marketplace mechanism for cloud computing services, which permits users to reserve capricious combination of services at wished timeslots, prices and quality of service.

Keywords
Proponents, Optimization, Heuristic, Nomenclature, Leveraged, Taxonomy.

I. Introduction

Cloud computing is the employment of computing resources (hardware and software) that are delivered as a service over a network (characteristically the Internet). The name comes from the common use of a cloud-shaped symbol as an pensiveness for the multifaceted infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

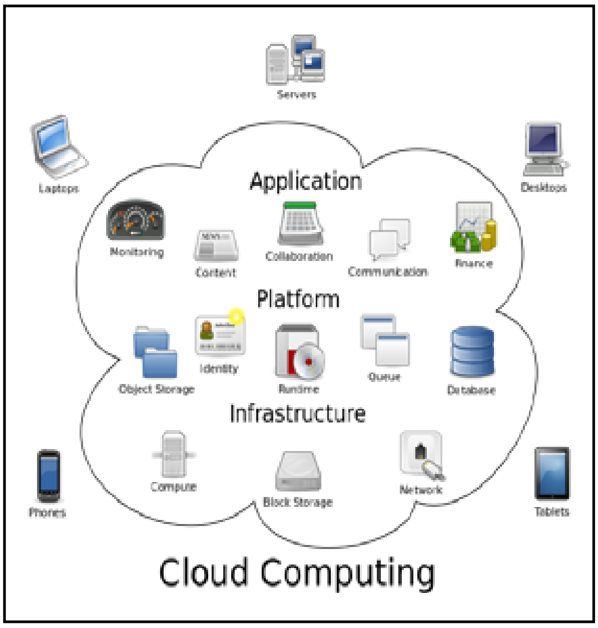

Fig.1: logical diagram of Cloud computing

End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

II. Enterprise System

An enterprise system generally consists of multiple subsystems running in parallel and/or sequentially, each of which requires a guaranteed quality of service (QoS) at a predictable price. Each enterprise system, or each user's request, is assumed to be represented by workflow. An example of business workflows is a payroll system [1]. It consists of a payroll calculation task on Java service along with an employee database task on SQL service, followed by reporting task on PDF/Email service as shown in fig. 2. Another example of engineering workflow is a CAE system [2]. It consists of a mesh generation task and a CFD analysis task on a HPC service, controlled by an optimization task on a general purpose optimization service. Every task needs to reserve the specified type of service within an appropriate timeslots to meet a deadline. The overall cost should also be restricted by the user's total budget. Each task in the workflow is implemented using PaaS; thus, the user needs combination of PaaS services to organize the user's workflow.

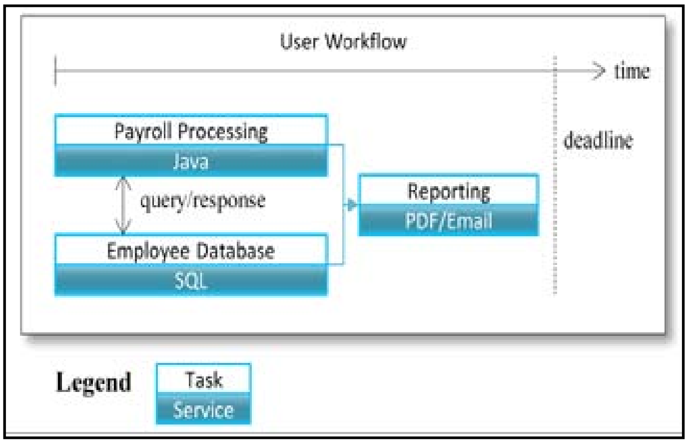

Fig. 2 : Enterprise system model

Fig. 3 illustrates another story of a company depends on SaaSes to run their business. The sales division uses a supply chain management service at the midnight to aggregate their daily records. The investment division uses a risk assessment service for several hours at the weekend to evaluate their assets. The accounting division uses a human resource management service for three days at the end of month to calculate the payroll.The design division uses a computer aided engineering service to meet a deadline of a product. The research division has their own computers currently not in use; so they can host other applications and earn from its user. The former three divisions need scheduled allocation of services forregular tasks, whereas the latter two divisions need immediate allocation of servicesfor irregular tasks. Consequently, the marketplace should support these scheduled and immediate allocations to meet the requirements of enterprise usage.

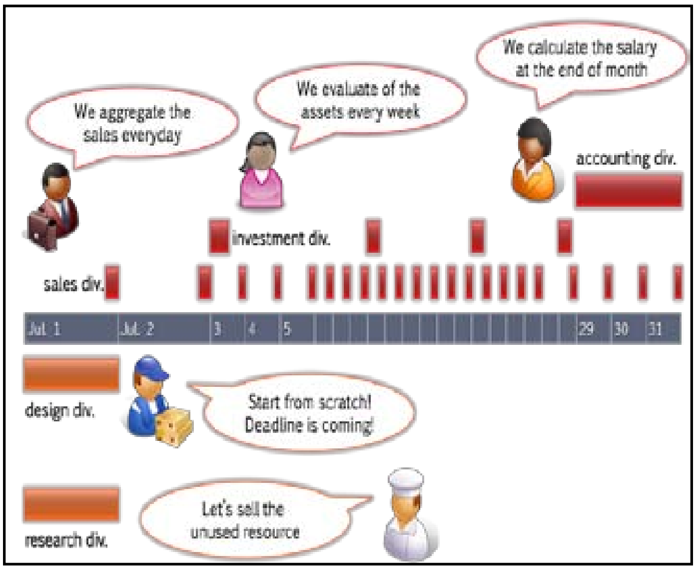

Fig. 3 : Example SaaS usage

A. Requirements

The service allocation decided by the marketplace must be fair and efficient; other wise the user will have no incentive to take part in. Hence, the PaaS marketplace is assumed to have to support the following requirements:

• Combination of services

Each user needs to bundle multiple services with different start/finish times as mentioned above. The cloud marketplace should allow users to express complementary requirements for an arbitrary combination of services.

• Predictability andflexibility

Since supply and demand in the cloud computing environment changes dynamically over time, users may desire predictable allocation in advance and adjustment at runtime.

• Economic efficiency

Every user and provider desires a fair and efficient allocation of services. The cloud marketplace should maximize the benefit of the participants and should not waste any resource. To this end, it is preferable for the marketplace to adopt an exact optimization approach rather than a heuristic approach.

• Double-sided competition

To encourage a fair exchange between providers and users, the prices should only depend on supply-demand condition, giving no structural advantage on a seller's (provider's) side or a buyer's (user's) side. The cloud marketplace should be designed after the double-sided auction model, meaning that the providers and the users compete with each other.

III. Marketplace Model

Figure 4 illustrates an ideal market place model that fits to the enterprise usage and requirements mentioned above. It should support multiple types of services, an application composed of multiple services as a workflow or as co-allocation, price bidding by both providers and users, and a fair outcome satisfying economic efficiency. The next chapter describes the proposed market model in detail.

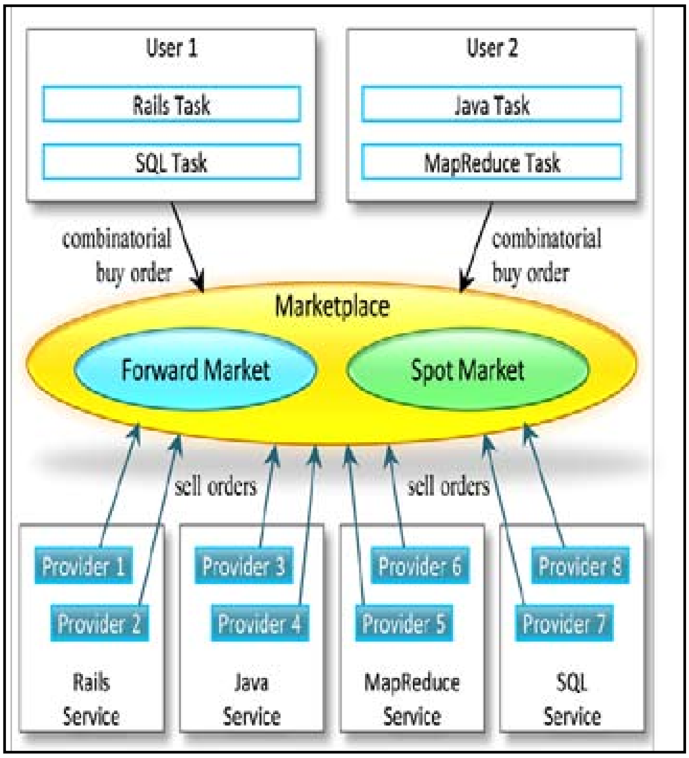

Fig. 4: Market place model

Fig. 4 outlines an abstract model for the market-based cluster Resource Management System (RMS). The purpose of the abstract model is to identify generic components that are fundamental and essential in a practical market-basedcluster Resource Management System and portray the interactions between these components. Thus, the abstract model can be used to study how existing cluster Resource Management Systemarchitectures can be leveraged and extended to incorporate market-based mechanisms to support utility-driven cluster computing in practice.

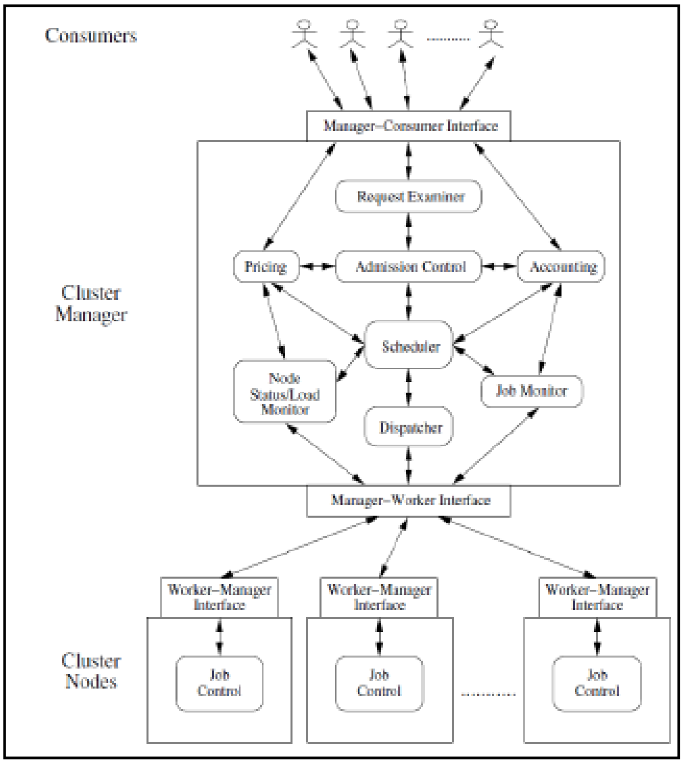

Fig. 5: The abstract model for market-based cluster Resource Management Systems.

The market-based cluster Resource Management System consists of two primary entities: a cluster manager; and a cluster node. For implementations within cluster systems, the machine that functions as the cluster managercan be recognized as the manager, server or dominant node and the machine that functions as the cluster node can be recognized as the worker or execution node. The real number of cluster manager and cluster nodes depends on the implemented management control. For example, a simple and common configuration for cluster systems is to support central management control where a single cluster manager collates manifold cluster nodes into a pool of resources as shown in Fig. 5.

The cluster manager assists as the front-end for users and provides the scheduling engine accountablefor allocating cluster resources to user applications. Thus, it supports two interfaces: the manager–consumer interface to receive requests from consumers; and the manager–worker interface to execute requests on selected cluster nodes. The consumers can be real user applications, service brokers that work on the behalf of user applications or other cluster Resource Management Systems such as those operating in multi-clustering or Grid federation environments where requests that cannot be fulfilled locally are forwarded to other cooperative clusters.

When a service request is first acquiesced, the request examiner interprets the submitted request for QoS requirements such as deadline and budget. The admission control then determines whether to accept or reject the request in order to ensure that the cluster system is not overloaded whereby many requests cannot be fulfilled successfully. The scheduler selects appropriate worker nodes to satisfy the request and the dispatcher starts the execution on the selected worker nodes. The node status/load monitor keeps track of the availability of the nodes and their workload, while the job monitor maintains the execution progress of requests.

It is vital for a market-based cluster Resource Management System to support pricing and accounting mechanisms.

The pricing mechanism decides how requests are charged. For example, requests can be charged based on submission time (peak/off-peak), pricing rates (fixed/changing) or availability of resources (supply/demand). Pricing serves as a basis for managing the supply and demand of cluster resources and facilitates in prioritizing resource allocations effectively. The accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to the consumers. Furthermore, the maintained historical usage information can be used by the scheduler to improve resource allocation decisions.

The cluster nodes deliver the resources for the cluster system to execute service requests via the worker–manager interface. The job control guarantees that requests are fulfilled by monitoring execution progress and enforcing resource assignment for executing requests.

A. Nomenclature

The taxonomy highlights on the practical facets of market-based cluster Resource Management Systems that are vital to achieve utility-driven cluster computing in practice. It recognizes key design factors and issues based on five major perspectives, namely the Market Model, the Resource Model, the Job Model, the Resource Allocation Model and the Evaluation Model.

Market Model nomenclature

The Market Model nomenclature examines how market concepts current in real-world human economies are incorporated into market-based cluster Resource Management Systems. This allows developers to understand what market- related attributes need to be considered, and in particular, to deliver utility. The Market Model nomenclature comprises four sub-taxonomies: the economic model, the participant focus, the trading environment and QoS attributes (see Figure 6).

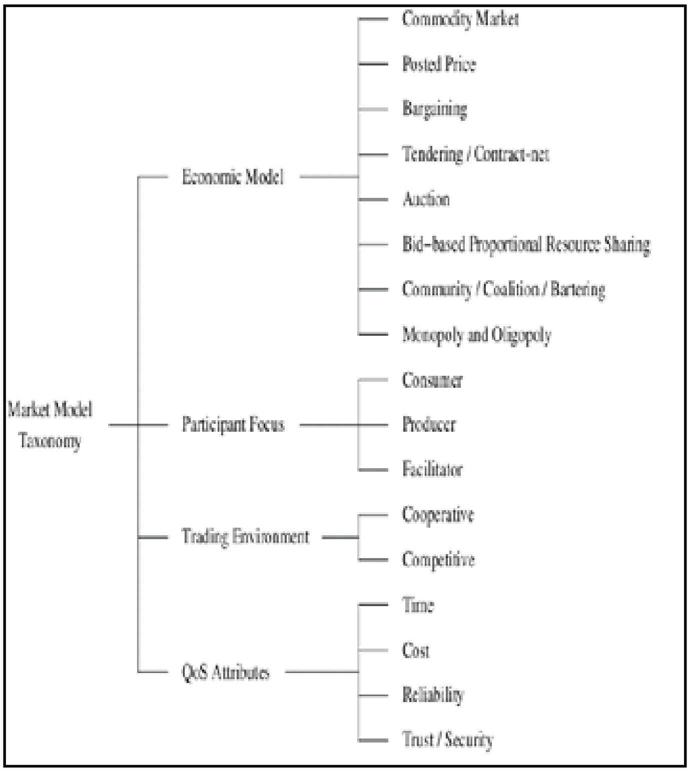

Fig. 6: The Market Model nomenclature.

IV. Conclusion

The proposed market mechanism is so generic that any kind of service can be traded equally upon it. In a realistic scenario, however, the marketplace will be used hierarchically. For instance, a bundle of a low-level "storage service" and a high-level "corporate management consulting service" is not likely to be ordered, but that of a "storage service" and another low-level "networking service" are likely to be ordered. There appears a hierarchical structure within the marketplace.

References

[1]. *D. Amrhein, P. Anderson, A. de Andrade, J. Armstrong, E.A. B, R. Bruklis, K.Cameron, R. Cohen, A. Easton, R. Flores, G. Fourcade, T. Freund, B. Hosseinzadeh, W.J. Huie, P. Isom, S. Johnston, R. Kulkarni, A. Kunjunny, T. Lukasik, G. Mazzaferro, C. McClanahan, W. Melo, A. Monroy-Hernandez, D. Nicol, L. Noon, S. Padhy, G. Pfister, T. Plunkett, L. Qian, B. Ramachandran, J. Reed, G. Retana, D. Russell, K. Sankar, A.O. Sanz, W. Sinclair, E. Sliman, P. Stingley, R. Syputa, D. Tidwell, K. Walker, K. Williams, J.M. Willis, Y. Sasaki, E. Windisch, and F. Zappert, Cloud Computing Use Cases White Paper Version 2.0,2009.*

[2]. *S. Weston, D. Green, G. Katsaros, T. Seed, N. Mc Donnell, and F. Scharinger,GridCAE Grid for Computer Aided Engineering, 2009.*

[3]. *H. Kita, H. Sato, N. Mori, and I. Ono, "U-Mart system, software for open experiments of artificial market," Computational Intelligence in Robotics and Automation 2003 Proceedings 2003 IEEE International Symposium on, Ieee,2003, pp. 1328*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

- 1333 vol.3.

[4]. T. Eymann, M. Reinicke, W. Streitberger, O. Rana, L. Joita, D. Neumann, B. Schnizler, D. Veit, O. Ardaiz, P. Chacin, I. Chao, F. Freitag, L. Navarro, M. Catalano, M. Gallegati, G. Giulioni, R.C. Schiaffino, and F. Zini, "Catallaxy-based Grid markets," Multiagent and Grid Systems, vol. 1, Dec.2005, pp. 297-307.

[5]. R. Wolski, J.S. Plank, T. Bryan, and J. Brevik, "G-commerce: market formulations controlling resource allocation on the computational grid," Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001, IEEE Comput.Soc, 2001, p. 8.

[6]. R. Buyya, D. Abramson, and J. Giddy, "Nimrod/G: an architecture for aresource management and scheduling system in a global computational grid,"Computer, vol. 1, 2000, pp. 283-289.

[7]. T. Eymann, M. Reinicke, W. Streitberger, O. Rana, L. Joita, D. Neumann, B. Schnizler, D. Veit, O. Ardaiz, P. Chacin, I. Chao, F. Freitag, L. Navarro, M. Catalano, M. Gallegati, G. Giulioni, R.C. Schiaffino, and F. Zini, "Catallaxy-based Grid markets," Multiagent and Grid Systems, vol. 1, Dec.2005, pp. 297-307.

[8]. R. Wolski, J. Brevik, J.S. Plank, and T. Bryan, "Grid Resource Allocation and Control Using Computational Economies," GRID COMPUTING: MAKING THE GLOBAL INFRASTRUCTURE A REALITY, 2003, pp. 747 - 772.

[9]. T.D. Cordeiro, D.B. Damalio, P.T. Endo, A.V. De Almeida Palhares, G.E.Gonçalves, D.F.H. Sadok, J. Kelner, B. Melander, V. Souza, and J.-E. Mångs, "Open Source Cloud Computing Platforms," 2010 Ninth International Conference on Grid and Cloud Computing, 2010, pp. 366-371.

Author's Profile

I, Kamta Giri, Research scholar in Computer Science and Engineering, for Sai Nath University, Rachi, Jharkhand. I, Played key role in network establishment of a joint enterprise cloud computing service provided by state-owned telecom services provider Bharat Sanchar Nigam Limited (BSNL) and IT solutions provider Dimension Data . Also had been instrumental in accept testing process of storage services for IDC (Internet Data Centre) of BSNL. Working as a Sr. Sub Divisional Engineer at Pune Data Center of BSNL.