Optimizing the Cloud Service Provider and its User in Cloud

[1] N. Kesava Rao, [2]P.K. Venkateswar Lal, [3]K..Venkatarathanam

[1]Assoc.Professor, [2] Assoc. Professor, [3]Assoc.Professor

[1]Dept. of CSE,

[1]Narayana Engineering College, Gudur, A.P, India

Abstract- In this paper, we are implementing a service mechanism for profit optimizations of both a cloud provider and its multiple users. We consider the problem from a game theoretic perspective and characterize the relationship between the cloud provider and its multiple users as a Stackelberg game, in which the strategies of all users are subject to that of the cloud provider. The cloud provider tries to select and provision appropriate servers and configure a proper request allocation strategy to reduce energy cost while satisfying its cloud users at the same time. For each user, we design a utility function which combines the net profit with time efficiency and try to maximize its value under the strategy of the cloud provider. We formulate the competitions among all users as a generalized Nash equilibrium problem (GNEP). We solve the problem by employing variation inequality (VI) theory and prove that there exists a generalized Nash equilibrium solution set for the formulated GNEP. Finally, we propose an iterative algorithm (IA), which characterizes the whole process of our proposed service mechanism. The experimental results show that our IA algorithm can benefit both of a cloud provider and its multiple users by configuring proper strategies.

Keywords- *Cloud, Utility Function, Iterative Algorithm*

I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing. The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.

Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Price: Pay for only the resources used.

Security: Cloud instances are isolated in the network from other instances for improved security. Performance: Instances can be added instantly for improved performance.

Clients have access to the total resources of the Cloud's core hardware.

Scalability: Auto-deploy cloud instances when needed.

Uptime: Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.

Control: Able to login from any location. Server snapshot and a software library lets you deploy custom instances.

Traffic: Deals with spike in traffic with quick deployment of additional instances to handle the load.

.

II. PROBLEM STATEMENT

The motivation behind the project is to provide cloud space to the users based on their requirement. The main aim of the project is to provide profit optimizations for cloud providers and its users. The main objective of the project is to provide service to the users by giving the time duration as input to the cloud space response. Based on the requirement of the user, cloud space response provides service to the users.

III. LITERATURE REVIEW

Cloud computing uses a network of remote servers hosted on the internet to store, manage and process data rather than a local server or a personal computer. Cloud provides the space to store the data i.e. the user can store his data in the cloud service. There are different types of clouds where the data can be store.

Public cloud: The data in the public cloud can be accessible by any person.

Private cloud: The data in a private cloud be accessible by a group of people.

Hybrid cloud: It is the combination of both public and the private cloud.

Community cloud: A group of similar organisations can access the data in this type of cloud.

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.

Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Payas you go (weekly, quarterly or yearly), based on demand.

Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.

Streamline processes. Get more work done in less time with less people.

Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.

Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.

Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.

It could bring hardware costs down. Cloud computing systems would reduce the need for advanced hardware on the client side. You wouldn't need to buy the fastest computer with the most memory, because the cloud system would take care of those needs for you. Instead, you could buy an inexpensive computer terminal. The terminal could include a monitor, input devices like a keyboard and mouse and just enough processing power to run the middleware necessary to connect to the cloud system. You wouldn't need a large hard drive because you'd store all your information on a remote computer.

ECONOMIC MODELS FOR CLOUD SERVICE MARKETS

Cloud computing is a paradigm that has the potential to transform and revolutionalize the next generation IT industry by making software available to end-users as a service. A cloud, also commonly known as a cloud network, typically comprises of hardware (network of servers) and a collection of softwares that is made available to end-users in a pay-as-you-go manner. Multiple public cloud providers (e.g.,

Amazon) co-existing in a cloud computing market provide similar services (software as a service) to its clients, both in terms of the nature of an application, as well as in quality of service (QoS) provision. The decision of whether a cloud hosts (or finds it profitable to host) a service in the long-term would depend jointly on the price it sets, the QoS guarantees it provides to its customers, and the satisfaction of the advertised guarantees. In the first part of the paper, we devise and analyze three inter-organizational economic models relevant to cloud networks. We formulate our problems as non cooperative price and QoS games between multiple cloud providers existing in a cloud market. We prove that a unique pure strategy Nash equilibrium (NE) exists in two of the three models. Our analysis paves the path for each cloud provider to know what prices and QoS level to set for end-users of a given service type, such that the provider could exist in the cloud market. A cloud provider services end-user requests on behalf of cloud customers, and due to the uncertainty in user demands over time, tend to over-provision resources like CPU, power, memory, storage, etc., in order to satisfy QoS guarantees. As a result of over-provisioning over long timescales, server utilization is very low and the cloud providers have to bear unnecessarily wasteful costs. In this regard, the price and QoS levels set by the CPs drive the end-user demand, which plays a major role in CPs estimating the minimal capacity to meet their advertised guarantees. By the term 'capacity', we imply the ability of a cloud to process user requests, i.e., number of user requests processed per unit of time, which in turn determine the amount of resources to be provisioned to achieve a required capacity. In the second part of this paper, we address the capacity planning/optimal resource provisioning problem in single-tiered and multi-tiered cloud networks using a techno-economic approach. We develop, analyze, and compare models that cloud providers can adopt to provision resources in a manner such that there is minimum amount of resources wasted, and at the same time the user service-level/QoS guarantees are satisfied.

OPTIMIZING ENERGY EFFICIENCY OF CLOUD

The energy consumption of cloud data centers has been growing drastically in recent years. In particular, CPUs are the most power hungry components in the data center. On the one hand, CPUs are not energy proportional with respect to their utilization levels because a cloud server's energy efficiency is much lower with limited CPU utilizations. On the other hand, current cloud computing applications usually, exhibit significant CPU idle time composed of idle intervals of variable lengths. The power consumption in these idle intervals is significant due to the prominent leakage current in recent technology nodes. There are a few existing schemes that transition a CPU into various low-power and sleep states to reduce its idle power. But none of them is optimal due to the fact that entering a sleep state may result in negative power savings if its wake-up latency is longer than the current idle interval. Therefore, intelligent sleep state entry is a key challenge in improving data centers' CPU energy efficiency. In this work, we propose a dynamic idle interval prediction scheme that can estimate future CPU idle interval lengths and thereby choose the most cost-effective sleep state to minimize power consumption at runtime. Experiments show that our proposed approach can significantly outperform other schemes, achieving 10% - 50% power savings compared to DVFS for a variety of CPU idle patterns. Of short and variable idle intervals. The power consumption in these idle intervals is significant due to leakage power being prominent in recent technologies. Therefore, we study a number of schemes that transition the CPU into various low powers and sleep states to reduce the CPU idle power. Entering a sleep state may result in negative power savings if its wakeup latency is longer than the current idle interval. Therefore, intelligent sleep state entry is a key challenge in improving data center CPU energy usage. In this work, we propose a dynamic idle interval prediction scheme that can forecast the current CPU idle interval length and thereby choose the most cost-effective sleep state for achieving the minimized power consumption during runtime. Our proposed approach largely outperforms other schemes examined, achieving 10% - 50% power savings compared to DVFS when using various CPU idle patterns. Our future work includes developing the real predictor in a Cloud simulation environment to provide a detailed and flexible evaluation platform for future studies.

The system is developed as three modules which are described in the following sections:
Set Alarm: This module is responsible to get inputs from user about the alarm – name of location, expiry date of alarm, reminder description if any. This module is responsible of converting the location name into actual geo co-ordinates and set the alarm for that location.
Alarm Generator: This module is responsible to ring the alarm if user is near to the location for which alarm is set.
Alarm Viewer: This module is responsible to display the already set alarms. Allows user to edit/delete/update alarms.
Location Selection and Characterization
In this module, for the first time users, the users need to configure the application using various options. The users are given options to configure the application in their mobile, such that options such as emergency numbers with two options, with the name which should be displayed in the messages, the location information, time information, pin information etc. Pin information is given to make the application secure. Such that no one can change the configuration files, to help in emergency. There may be chances of someone to change the configuration files, so as to protect in from these attacks, secure pin methodology is adopted.
Emergency Scenarios
Users having received notification of an emergency are unlikely to maintain normal usage patterns. In particular, users are likely to attempt to contact their friends and/or family soon after learning about such conditions. Here we considered emergency scenarios like Accident, heart attack, lost location and struck to thief. Alert message will be sent immediately to the emergency numbers like friends or relatives number, to whom ever configured initially in first module.

IV. SYSTEM ANALYSIS

Some works have been done for profit optimizations of cloud centers
In proposed a heuristic method to tackle profit maximization for a cloud provider. They focus on auction profit maximization in the context of multiple virtual machines (VMs).

Goudarzi and Pedram developed a heuristic to deal with profit maximization in cloud computing system with service level agreements. They try to reduce cost by powering off appropriate servers, i.e., selecting appropriate servers to provide services.

More recently, Cao et al. proposed an optimal method for energy saving under continuous dynamic voltage frequency scaling (DVFS) environment. Specifically, they try to configure appropriate speed for each server to save energy.

However, all these existing methods mainly consider from the perspective of the cloud provider.

To our knowledge, hardly any previous works investigate multiple users′ profit optimizations, let alone optimizing the profits of a cloud provider and its users at the same time.

In this project, we are implementing a new service mechanism for profit optimizations of both a cloud provider and its multiple users.

approximately characterize the request arrivals as a Poisson process. Since the payment and time efficiency of each of the cloud users are affected by the decisions of others, it is natural to analyze the behaviors of these users.

In our mechanism, the cloud provider tries to select appropriate servers and configure a proper request allocation strategy to reduce energy cost while satisfying its users at the same time. Experimental results show that our algorithm can benefit both of the cloud provider and its multiple users by configuring proper strategies.

MODULES

CLOUD SPACE REQUEST

In this module customer will have features like customer details, search and buy options. Customer will Search for type of services provided by cloud service and select option to purchase that service from cloud by viewing cost and select total number of day's customer want to user service.

USER ACTIVATION

Activation module can active or deactivate users view customer details , service details , feedback and it can view types of services cloud is providing and view feedbacks of each user for used services.

CLOUD SPACE RESPONSE

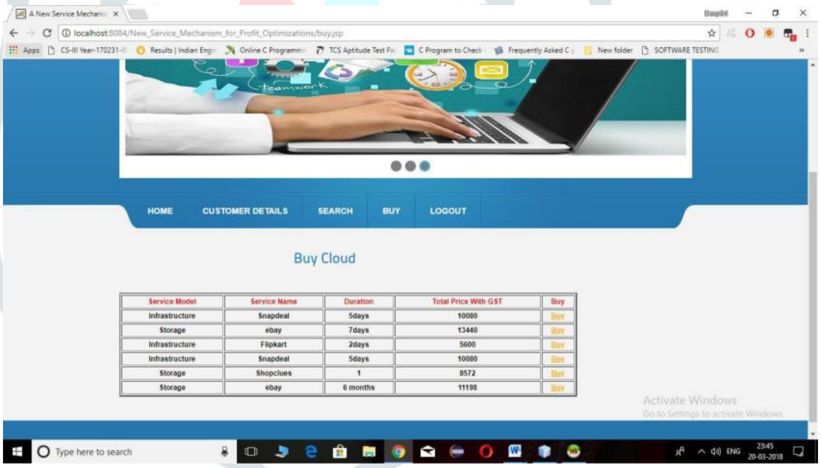
In this module service provider will provide different types of services or users like infrastructure, platform and storage.Service Type

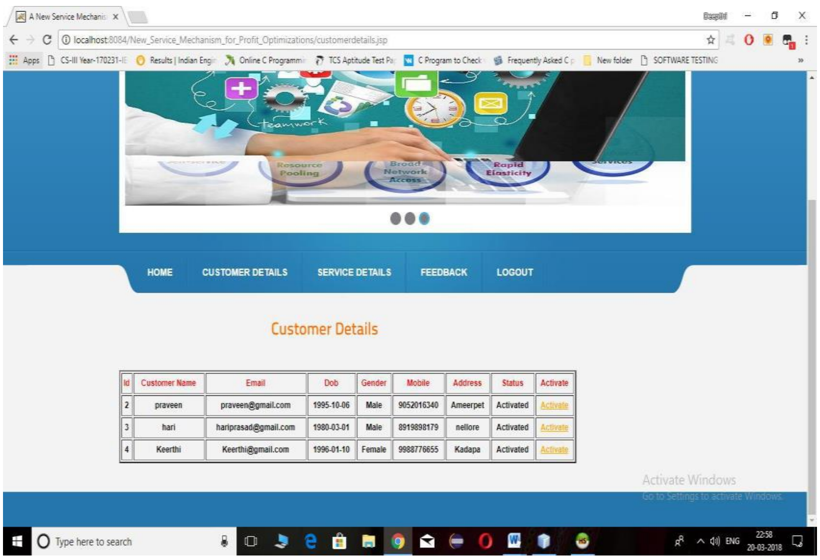
Service provider module adds type of service like ecommerce.etc. and type of implement drive public or private and allocate service type like windows, Linux, Unix and provide cost for this type of service with validation date.

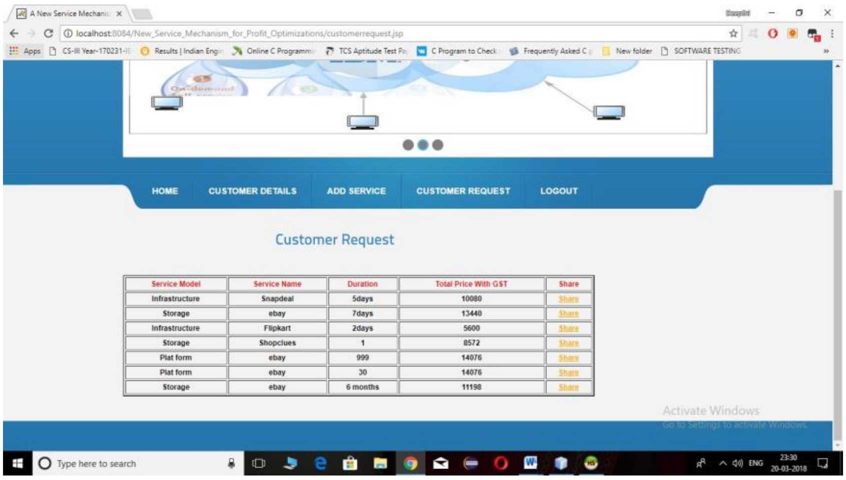
Customer Request

The module can view list of requests received from users to take available services along with total number of days he want cloud service and total cost for given days.

V. EXPERIMENTAL RESULTS





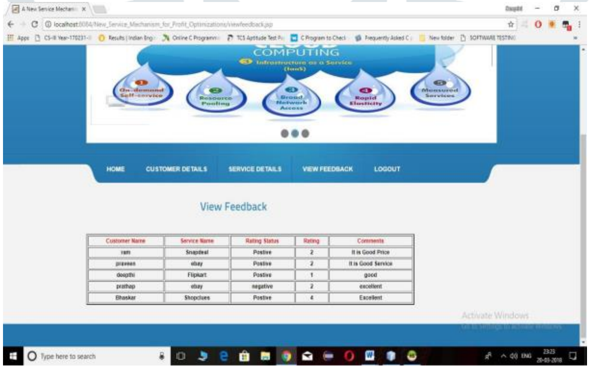


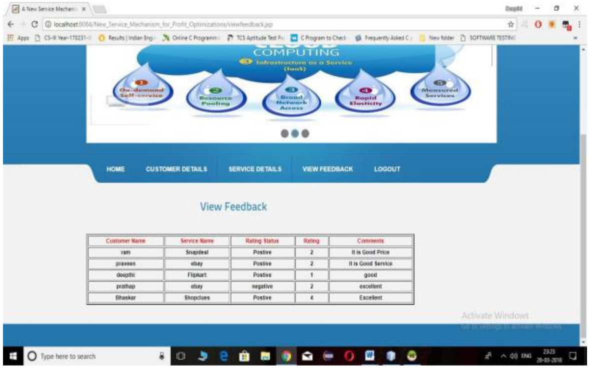
VI. CONCLUSION

With the popularization of cloud computing and its many advantages such as cost-effectiveness, elasticity, and scalability, more and more applications are moved from local computing environment to cloud center. In this work, we try to design a new service mechanism for profit optimizations of both a cloud provider and its multiple users.

We consider the problem from a game theoretic perspective and characterize the relationship between the cloud provider and its multiple users as a Stackelberg game, in which the strategies of all users are subject to that of the cloud provider. The cloud provider tries to select appropriate servers and configure a proper request allocation strategy to reduce energy cost while satisfying its cloud users at the same time. We approximate its server selection space by adding a controlling parameter and configure an optimal request allocation strategy. For each user, we design a utility function which combines the net profit with time efficiency and try to maximize its value under the strategy of the cloud provider. We formulate the competitions among all users as a generalized Nash equilibrium problem (GNEP).

We solve the problem by employing varational inequality (VI) theory and prove that there exists a generalized Nash equilibrium solution set for the formulated GNEP. Finally, we propose an iterative algorithm (IA), which characterizes the whole process of our proposed service mechanism. We conduct some numerical calculations to verify our theoretical analyses. The experimental results show that our IA algorithm can reduce energy cost and improve users utilities to certain extent by configuring proper strategies.





VII.REFERENCES

[1]D. Boneh and M. Franklin, “Identity- based encryption from the weil pairing,” in Advances in Cryptology–CRYPTO.
California, USA: Springer Berlin Heidelberg, Pages: 19-23, 213–229 Year: August 2001.

[2] Y. Zhu, G.-J. Ahn, H. Hu, and H.Wang, “Cryptographic role-based security mechanisms based on role-key hierarchy,” in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. Beijing, China: ACM, Pages: 13-16, 314–319, Year: April 2010

[3]Y. Zhu, H.-X. Hu, G.-J. Ahn, H.-X. Wang, and S.-B. Wang, “Provably secure role-based encryption with revocation mechanism,” Journal of Computer Science and Technology, vol. 26, no. 4, Pages: 697–710, Year: 2011.

[4]Y. Zhu, G. J. Ahn, H. Hu, D. Ma, and S. Wang, “Role-based cryptosystem: A new cryptographic rbac system based on role-key hierarchy,” IEEE Transactions on Information Forensics and Security, vol. 8, no. 12, Pages:2138–2153, Year:2013.

[5]A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Advances in Cryptology–EUROCRYPT 2005, vol. 3494. Aarhus, Denmark: Springer Berlin Heidelberg, Pages: 22-26 ,457– 473, Year: May 2005

[6]J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in IEEE Symposium on Security and Privacy. Berkeley, CA: IEEE, 20-23, Pages: 321–334, Year: May 2007