ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 6 June 2016, Page No. 16855-16859

QOS Ranking Prediction For Cloud Brokerage Services

Muthulakshmi.J[1], Mr.Rama Doss.P[2]

[1]M.E. Student, Department of CSE, Parisutham Institute of Technology and Science, Tamil Nadu, India [2]Asst. Professor, Department of CSE, Parisutham Institute of Technology and Science, Tamil Nadu, India

[1]mlakshmi.jpks@gmail.com, [2]vp.ramadoss@gmail.com

***Abstract*-Cloud computing is becoming popular. Build high-quality cloud applications is a critical research problem. QoS rankings provide valuable information for make optimal cloud service selection from a set of functionally equivalent service candidates. To obtain Qos values real-world invocations the service candidates are usually required based on the Cloud Broker. To avoid the time consuming and expensive real-world service invocations, It proposes a QoS ranking prediction framework for cloud services by taking an advantage of the past service usage experiences of other consumers. Our proposed framework requires no need additional invocations of cloud services when making QoS ranking prediction by cloud broker service provider. Two personalized QoS ranking prediction approaches are proposed to predict the QoS rankings directly based on cost and ranking. Comprehensive experiments are conducted employing real-world QoS data, including 300 distributed users and 500 real world web services to all over the world. The experimental results show that our approaches outperform other competing approaches.**

***Keywords*—Cloud computing, cloud brokerage, cost management, instance reservation, approximation algorithm**

Cloud computing is Internet-based computing Cloud computing is Internet-based computing whereby shared configurable resources are provided to computers and other devices as services Strongly promoted by the leading industrial companies cloud computing is quickly becoming popular in recent years Applications deployed in the
Cloud environment are typically large scale and complex With the rising popularity of cloud computing, how to build high-quality cloud applications becomes an urgently required research problem. Similar to traditional component-based systems cloud applications typically involve multiple cloud components communicating with each other over application programming interfaces, such as through web services. The business process of this cloud application is composed by a number of software components, where each component fulfills a specified functionality. To outsource part of business server side refers to the cloud services.

II.PROBLEM STATEMENT

Nonfunctional performance of cloud services is usually described by quality-of-service. QoS is an important research topic in cloud computing. When making optimal cloud service selection from a set of functionally equivalent services, QoS values of cloud services provide valuable information to assist decision making. In
traditional component-based systems, software components are invoked locally, while in cloud applications, cloud services are invoked remotely by Internet connections. Client-side performance of cloud services is thus greatly influenced by the unpredictable Internet connections. Therefore, different cloud applications may receive different levels of quality for the same cloud service. In other words, the QoS ranking of cloud services for a user cannot be transferred directly to another user, since the locations of the cloud applications are quite different. Personalized cloud service QoS ranking is thus required for different cloud applications to other companies, some of these components invoke other cloud services. These cloud services are provided and deployed in the cloud by other companies by other cloud applications. Since there are a number of functionally equivalent services in the cloud, optimal service selection becomes important.Service users refer to cloud applications that use/ invoke the cloud services. In the context of a service invocation, the user-side refers to applications and Cloud Services Brokerage assesses the individual needs of the organization and recommends Cloud strategy, usage and management using the vendors other words, a Cloud Services Brokerage is a third party company that adds value to Cloud services on party company that adds value to Cloud services on behalf of Cloud service consumers. A cloud broker may not provide any granted the rights to negotiate
contracts with cloud providers on behalf of the customer. On-demand instances are economically inefficient to users, not only because of the higher rates, but also because there is a fundamental limit on how small the billing cycle can be made. For example, Amazon Elastic Compute Cloud (EC2) charges on-demand instances based on running hours. In this case, an instance running for only 10 minutes is billed as if it were running for a full hour. Such billing inefficiency becomes more salient for cloud providers adopting longer billing cycles (e. g., in VPS.NET.Even a single hour is charged at a daily rate), and for sporadic demands with a substantial amount of partial usage. In general, to what extent a cloud user can enjoy cost savings due to reservation, while avoiding its inefficiency due to coarse-grained billing cycles, is limited by its own demand pattern

III.PROPOSED SYSTEM

The propose a personalized ranking prediction

DOI: 10.18535/ijecs/v5i6.22

framework, named Cloud Rank, predict the QoS ranking of a set of cloud services without requiring additional real-world service invocations from the intended users. Our approach takes advantage of the past usage experiences of other users for making personalized ranking prediction for the current user based on the cloud broker. This approach takes gain of the past usage experiences of other users for building personalized ranking prediction and cost migration alert for the Active user in the cloud. It uses the two algorithms namely cloudrank1 and cloudrank2. This paper overcomes the existing system and it consists of following pros:. It takes the advantage of past usage experiences from other users. Identify the risky problem of personalized QoS ranking for cloud services and proposes a QoS ranking prediction framework to tackle the problem.

1) A PROFITABLE CLOUD BROKER

Most IaaS clouds provide users with multiple purchasing options, including on-demand instances, reserved instances, and other instance types [3], [4], [5], [6], [7], [8]. On-demand instances allow users to pay a fixed rate in every billing cycle(e.g., an hour) with no commitment. For example, if the hourly rate of an on-demand instance is p, an instance that has run for n hours is charged np.

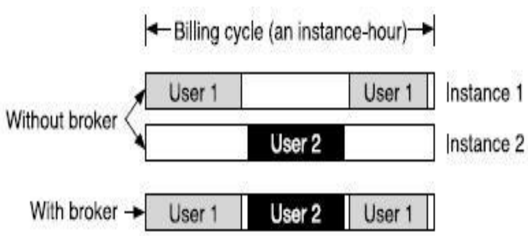

Fig. 1. The broker can time-multiplex partial usage from different users in the same instance-hour. In this case, serving two users only takes one instance-hour, instead of two.

On the other hand, a reserved instance allows a user to pay a one-time fee to reserve an instance for a certain amount of time, with reservation pricing policies subtly different across cloud providers. In most cases, the cost of a reserved instance is fixed. For example,in [4], [5], [6], [7], [8], [9], the cost of a reserved instance is equal to the reservation fee. As another example, in Amazon EC2 [3], the cost of a Heavy Utilization Reserved Instance is computed as a reservation fee plus a heavily discounted hourly rate charged over the entire reservation period, irrespective of the actual instance usage. EC2 also offers othe reservation options (e.g., light/medium utilization reserved instances), with cost linearly dependent on the actual usage time of the reserved instance. Throughout the paper, we limit our discussions to reservations with fixed costs, the most common cases in IaaS clouds.We propose a cloud broker that can save expenses for cloud users. As illustrated in Fig. 1, the broker reserves a large pool of instances from the cloud providers to serve a major part of incoming user demand, while accommodating request bursts by launching on-demand instances. The broker pays IaaS clouds to retrieve instances while collecting revenue from users through its own pricing policy. From the perspective of users, their behavior resembles launching instances “on demand” provided by the broker, yet at a lower price. The broker can reduce the total service cost and reward the savings to users mainly through demand aggregation, with the following benefits:Better exploiting reservation options. The broker aggregates the demand from a large number of users for service, smoothing out individual bursts in the aggregated demand curve, which is more stable and suitable for service through reservation. In contrast, individual users usually have bursty and sporadic demands, which are not friendly to thereservation option.

2) CLOUD SERVICE REGISTRATION

The design schemes consist of the n number of clients and cloud servers. In this module the client may collect information and give login detail such as the user name and password to register the cloud services. Before the registration of cloud services to ensure whether the client is an authenticated or not to access cloud server. We can ensure the information stored in the cloud is used judiciously by the responsible stakeholders as per the service level agreements. The module with an aim of accountability among users like cloud service providers who store and manage the information after registration completes, the cloud services are provided to individual users.

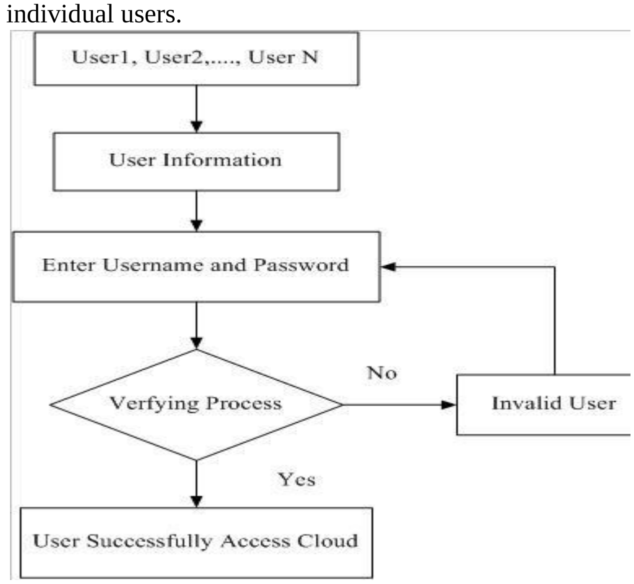

Figure 2: Service Registration

3) INDEXING THE CLOUD SERVICES

After completing the registration the user may communicate to the cloud network. First determining the clients are splitting into the categorized by depending on the quality of cloud services and then the cloud users are indexing to the based on the cloud services. The design with an aim of distributed accountability among key stakeholders like cloud service providers who store and manage the information the design with an aim of distributed accountability among key stakeholders like cloud service providers who store and manage the information Thus the indexing method may using the tree based indexing to produce the efficient quality of cloud services to the client.

SYSTEM ARCHITECTURE

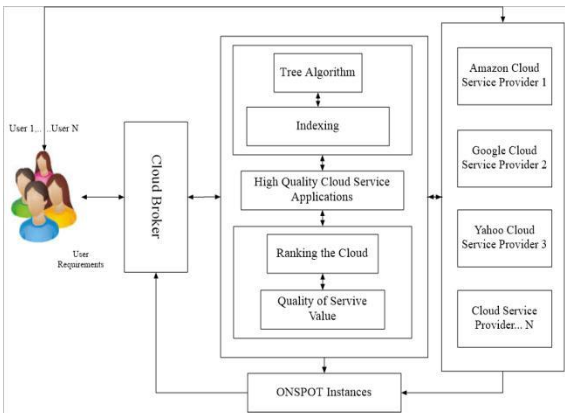

Figure 3: System architecture

4) PERFORMANCE EVALUATION

In this section, we conduct simulations driven by a large volume of real-world traces to evaluate the performance of the proposed brokerage service and reservation strategies, under an extensive range of scenarios.

Data Set Description and Preprocessing

Workload traces in public clouds are often confidential: no IaaS cloud has released its usage data so far. For this reason, we use the recently released Google cluster-usage traces [10], [14] in our evaluation. Although Google cluster is not a public IaaS cloud, its usage traces reflect the computing demands of Google engineers and services, which can rep-resent demands of public cloud users to some degree. The data set contains 180 GB of resource usage information of 933 users over 29 days in May 2011, on a cluster of 12,583 physical machines. In the traces, a user submits work in the form of jobs. A job consists of several tasks, each requesting a set of resources such as CPU, disk, memory, etc.

Instance scheduling. We take such a data set as input, and ask the question: How many computing instances would each user require if she were to run the same workload in a public IaaS cloud? It is worth noting that in Google cluster, tasks of different users may be scheduled onto the same machine, whereas in IaaS clouds each user will run tasks only on her own computing instances. Therefore, we reschedule the tasks of each user onto instances that are exclusively used by this user. We set the instances to have the same computing capacity as Google cluster machines, which enables us to accurately estimate the

task run time by learning from the original traces. For each user, we use a simple algorithm to

schedule her tasks onto available instances that have sufficient resources to accommodate their resource requirements. Tasks that cannot share the same machine (e.g., tasks of MapReduce) are scheduled onto different instances. (For simplicity, we ignore other complicated task placement constraints such as on OS versions and machine types.) A new instance will be launched if none of the available instances can accommo-date a submitted task. Note that tasks of one user cannot be scheduled onto another user's instances. In the end, we obtain a demand curve for each user, indicating how many instances the user requires in each hour. Fig. 6 illustrates the demand curves of three typical users in the first 200 hours. For the broker, it simply adds up all users' demands for instances as the aggregate demand. This preserves the instance isolations among users as no user shares instances with one another.

Pricing. Unless explicitly mentioned, we set the on-demand hourly rate to $0.044, the same as Amazon EC2 small instances.[2] Since the Google traces only spans one

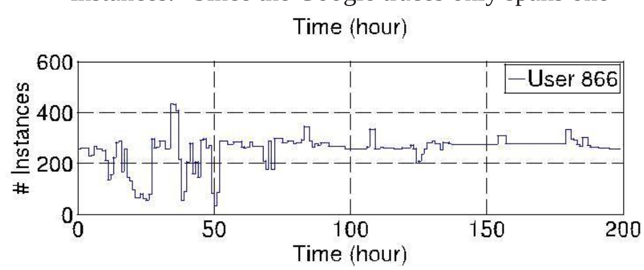

Fig. 4. The demand curves of three typical users. month, we assume that each reservation is effective for one week, with a full-usage discount of 50 percent: the reservation fee is equal to running an on-demand instance for half a res-ervation period, roughly matching the prevalent pricing policy in most IaaS clouds [3], [5], [6], [9].

Group division. To further understand the demand statis-tics of users, we compute the demand mean and standard deviation for each user and illustrate the results in Fig. 4. As has been mentioned, to what extent a user can benefit from reservations critically depends on its demand pattern: the more fluctuating the demand is, the less is the benefit from using reserved instances. We hence classify all 933 users into the following three groups based on the demand fluctua-tion level measured as the ratio between the demand stan-dard deviation and mean

IV.RELATED WORK

Three types of pricing options are currently adopted in IaaS clouds. Besides the on-demand and reserved instances introduced in Section 2, we note that some cloud providers charge dynamic prices that fluctuate over time, e.g., the Spot Instances in Amazon EC2 [3]. Some existing works dis-cuss how to leverage these pricing options to reduce instance running costs for an individual user. For example, Chohan et al. [17] investigate the use of Spot Instances as accelerators of the MapReduce process to speed up the overall MapReduce time while significantly reducing mone-tary costs. Zhao et al. [18] propose resource rental planning with EC2 spot price predictions to reduce the operational cost of cloud applications. Hong et al. [19] design an instance purchasing strategy to reduce the “margin cost” of over-provisioning. Hong et al. [19] also presents a strategy to combine the use of on-demand and reserved instances, which is essentially a special case of our Heuristic strategy when all demands are given in one reservation period. Chaisiri et al. [20] investigate a similar problem and propose an algorithm by solving a stochastic integer programming problem. Their algorithm limits the reservation decisions to be made at some specific time phases. The recent work of [13] proposes optimal online strategies to reserve instances without any a priori knowledge of future demands. Ver-meersch [21] implements a prototype software that dynami-cally retrieves instances from Amazon EC2 based on the user workload. All these works offer a consulting service, e.g., [22], [23], [24], that helps an individual user make instance purchasing decisions.

IaaS cloud brokers have recently emerged as intermedia-tors connecting buyers and sellers of computing resources. For example, SpotCloud [25] offers a “clearinghouse” in which companies can buy and sell unused cloud comput-ing capacity. Buyya et al. [26] discuss the engineering aspects of using brokerage to interconnect clouds into a global cloud market. Song et al. [27], on the other hand, propose a broker that predicts EC2 spot price, bids for spot instances, and uses them to serve cloud users. Unlike existing brokerage services that accommodate individual user requests separately, our broker serves the aggregated demands by leveraging instance multiplexing gains and instance reservation, and is a general framework not lim-ited to a specific cloud.

We note that the idea of resource multiplexing has also been extensively studied, though none of them relates to computing instance provisioning. For example, [28] makes use of bandwidth burstable billing and proposes a coopera-tive framework in which multiple ISPs jointly purchase IP transit in bulk to reduce individual costs. In [29], the anti-correlation between the demands of different cloud tenants is exploited to save bandwidth reservation cost in the cloud. [30] empirically evaluates the idea of statistical multiplexing and resource overbooking in a shared hosting platform. Compared with these applications, exploiting multiplexing gains in cloud instance provisioning poses new challenges, mainly due to the newly emerged complex cloud pricing options. It remains nontrivial to design instance purchasing strategies that can optimally combine different pricing options to reduce cloud usage cost.

5).CONCLUSION

In cloud computing environment are scalable and reliable event matching service for content-based pub/sub systems. It connects the brokers through a distributed overlay Cloud, which ensures reliable connectivity among brokers through its multi-level clusters and brings a low routing latency through a EIRQ algorithm. Through a hybrid multi-dimensional space partitioning technique, reaches scalable and balanced clustering of high dimensional skewed subscriptions, and each event is allowed to be matched on any of its candidate servers. Extensive experiments with real deployment based on a Cloud Stack tested are conducted, producing results which demonstrate that algorithm is effective and practical, and also presents good workload balance, scalability and reliability under various parameter settings.

A cloud services brokerage enables customer organizations to consume cloud resources easier, facilitating strategy around sourcing services and the decision to build vs. buy services. Utilizing automation, analytics, forecasting and real time reporting tools, an IT organization can make decisions on how to best broker internal services and third party cloud services seamlessly to its customers and also Increasing Maturity & Capability, Rapidly evolving arena within the cloud ecosystem with an array of service capabilities. Market shifting to cloud service brokers to solve cloud consumption and complexity issues. Focused on meeting the need for multi-cloud management, centralization and governance generation cloud service management software is already offering brokering capabilities.

REFERENCES

[1] W. Wang, D. Niu, B. Li, and B. Liang, “Dynamic cloud resource reservation via cloud brokerage,” in Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst., 2013, pp. 400–409.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” Commun. ACM, vol. 53, pp. 50–58, 2010.

[3] Amazon EC2 Pricing. (2014). [Online]. Available: http://aws.amazon.com/ec2/pricing/

[4] BitRefinery. (2013). [Online]. Available: http://bitrefinery.com

[5] ElasticHosts. (2013). [Online]. Available: http://www.elastichosts.com/

[6] GoGrid Cloud Hosting. (2013). [Online]. Available: http://www.gogrid.com

[7] Ninefold. (2013). [Online]. Available: http://www.ninefold.com

[8] OpSource. (2013). [Online]. Available: http://www.opsource.net

[9] VPS.NET. (2013). [Online]. Available: http://vps.net.

[10] Google Cluster-Usage Traces. (2012). [Online]. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion2

[11] W. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality. Hoboken, NJ, USA: Wiley, 2011.

[12] R. Fleischer, “On the bahncard problem,” Theoretical Comput. Sci., vol. 268, no. 1, pp. 161–174, 2001.

[13] W. Wang, B. Li, and B. Liang, “To reserve or not to reserve: Optimal online multi-instance acquisition in IaaS clouds,” in Proc. USENIX Int. Conf. Auton. Comput., 2010, pp. 13–22.

[14] C. Reiss, A. Tumanov, G. Ganger, R. Katz, and M. Kozuch, “Heterogeneity and dynamicity of clouds at scale: Google trace analysis,” in Proc. ACM 3rd Symp. Cloud Comput., 2012, pp. 1–13.

[15] A. E. Roth, Ed., The Shapley Value, Essays in Honor of Lloyd

S.

Shapley. Cambridge, U.K.: Cambridge Univ. Press, 1988. [16] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Algorithmic Game Theory, Cambridge, U.K.: Cambridge Univ. Press, 2007.

[17] N. Chohan, C. Castillo, M. Spreitzer, M. Steinder, A. Tantawi, and

C. Krintz, "See spot run: Using spot instances for MapReduce workflows," in Proc. Conf. Hot Topics Cloud Comput., 2010, p. 7.

[18] H. Zhao, M. Pan, X. Liu, X. Li, and Y. Fang, "Optimal resource rental planning for elastic applications in cloud market," in Proc.

IEEE 26th Int. Parallel Distrib. Process. Symp., 2012, pp. 808– 819.

[19] Y. Hong, M. Thottethodi, and J. Xue, "Dynamic server provisioning to minimize cost in an IaaS cloud," in Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Model. Comput. Syst., 2011, pp.147–148.

[20] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," IEEE Trans. Serv. Comput.,

vol. 5, no. 2, pp. 164–177, Apr.–Jun. 2010.

[21] K. Vermeersch, "A broker for cost-efficient qos aware resource allocation in EC2," Master's thesis, Dept. Comput. Sci., Univ. Antwerp, Antwerpen, Belgium, 2011

22 Cloudability. (2014). [Online]. Available: http://cloudability.com

23 Cloudyn. (2014). [Online]. Available: http://www.cloudyn.com

24 Cloud Express. (2014). [Online]. Available: https://www. cloudexpress.com

[22] SpotCloud. (2014). [Online]. Available: http://spotcloud.com/

[23] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comput. Syst., vol. 25, no. 6, pp. 599–619, 2009.

[24] Y. Song, M. Zafer, and K.-W. Lee, "Optimal bidding in spot instance market," in Proc. IEEE Conf. Comput. Commun, 2012, pp.190–198.

[25] 1592 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 26, NO. 6, JUNE 2015

[26] R. Stanojevic, I. Castro, and S. Gorinsky, "CIPT: Using tuangou to reduce IP transit costs," in Proc. ACM 7th COnf. Emerging Netw. EXp. Technol., 2011, pp. 1–12.

[27] D. Niu, H. Xu, and B. Li, "Quality-assured cloud bandwidth autoscaling for video-on-demand applications," in Proc. IEEE Conf. Comput. Commun, 2012, pp. 460–468.

[28] B. Urgaonkar, P. Shenoy, and T. Roscoe, "Resource overbooking and application profiling in shared hosting platforms," in Proc.

USENIX 5th Symp. Operating Syst. Des. Implementation, 2002, pp.239–254.

DOI: 10.18535/ijecs/v5i6.22