IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Web Service Scheduling in Multi-Cloud Environment

P. Sen[1*], D.Sarddar[2], S.K. Sinha[3], R. Pandit[4]

[1,2]Department of Computer Science and Engineering, University of Kalyani, Kalyani, W.B., India
[3]B.P.Poddar Institute of Management and Technology, Salt Lake, Sector-V, Kolkata, W.B., India
[4]Department of Computer Science, West Bengal State University, Barasat, kolkata, W.B., India

*****Corresponding Author: priyajit91@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 16/Jan/2019, Published: 31/Jan/2019

***Abstract*-**Cloud platforms are increasingly being used for hosting a broad diversity of services from traditional ecommerce applications to interactive web-based IDEs. However, we have noticed that the proliferation of offers by service providers promotes several challenges. Developers have to consider migrating services from one cloud to another and manage distributed applications spanning multiple clouds while specific service is not available in single cloud. In this paper, we have introduced a new methodology of web services scheduling in multi-cloud environment based on the request of preferred user services in different time slots of the day. A comparison has been made on the performance and cost of single and multi-cloud environments to show how multi-cloud is more efficient than single cloud and also to conclude that web service scheduling in multi-cloud environment is more efficient than single cloud during service migration.

Keyword- *Scheduling, Multi Cloud System, Web Services, Service Scheduling, Service Composition, Web Service Scheduling.*

I. INTRODUCTION

Cloud Network is a combination of commodity computers networked together in same or different geographical locations, working together to serve a number of customers with different requirements and workload on demand basis with the help of virtualization. Cloud services are provided to the cloud users as utility services like electricity, telephone using pay-as-you-use business model. These utility services are different services such as providing software, platform or infrastructure etc. based on the user demand. Cloud users use these services provided by the cloud service providers and develop their applications in the internet and thus deliver them to their end users. So the cloud users don't have to worry about installing, maintaining hardware and software needed in their personal computer. They also can take these services as they have to pay as much they use. This causes the reduction of expenditure as well as effort in the field of Information Technology using cloud based services instead of establishing IT based infrastructure.

1. Cloud Services

Most widely accepted services include three main services which would be- Software as a Service, Platform as a Service & Infrastructure as a Service. Different levels of abstraction are provided by these services to the cloud infrastructure.

1.1. SaaS (Software as a service):

It delivers a single application through the web browser to thousands of customers using a multi-tenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one application to maintain, cost is low compared to conventional hosting. The customer who uses SaaS will always access the required applications through the medium of internet. Examples of such type of services are Google Apps, Zoho Office.

1.2. PaaS (Platform as a service)

It delivers development environment as a service. A person can easily build his own applications which would run on the service provider's infrastructure which will then support transactions, uniform authentication, robust scalability and also availability. SaaS offers applications built by PaaS which can be consumed from the customer's web browsers. The ability to integrate and consume third-party services from other service platforms is available due to application of the above mentioned logic. Examples of such type of services are Google App Engine.

1.3. IaaS (Infrastructure as a Service)

IaaS service provides the users of the cloud greater flexibility to lower level than other services. It gives even CPU clocks with OS level control and also storage support to the users. Examples of such type of services are Amazon EC2 and S3.

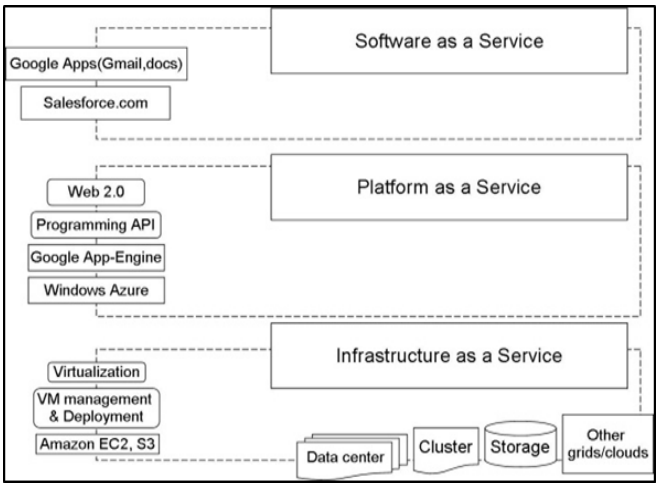

Figure 1: Cloud Service stack

- Major Challenges faced with Single Cloud Systems are,

i) In the recent years, the users of computer and information systems have been growing in a very rapid manner. Hence services existing in a single-cloud may not be properly provided, when requested by a huge number of users. Hence the multi-cloud environment can be used to a greater satisfactory level to serve user requests.

ii) In a single cloud system, adequate number and type of resources may not be available to serve user requests. In these cases, resources may be accumulated from different clouds to provide services to a huge number of people at a time.

iii) Any particular user requested service may have certain operations which presently cannot be accomplished from the cloud where other operations of this service are executing. Example: Railway Ticket Booking is a service which consists of two sections: providing reservation information and request, as well as paying amount through payment gateway. It may happen, ticket booking operation may be present/ available in a cloud, but at the same time, payment gateway option is not available, and may be needed to obtain from another cloud.

iv) Similar services may be present in different clouds, but the service may be selected from the one, where cost of obtaining the service is less and the network is reliable and secured.

v) Accessing information from a cloud may take several minutes depending on the type of data accessed from the cloud. Accessing multimedia data may require more time, as they must be stored in a compressed manner. It may be possible to have a cloud of huge space providing such content, thereby separating the textual contents that need not to be compressed and can be accessed fast from a relatively smaller storage cloud.

But there is a major drawback to combine services from multiple clouds, in time of need, due to the service integration factor for integrating services from multiple clouds especially with different protocols. But then it is beneficial, since the user always is provided with the service, which is the major goal of the distributed and cloud-based organizations.

1.4. Web Services

Web Services are the services that reside in web server and can be offered to users on demand. These services may reside in cloud and can be offered through providers known as agents.

1.5. Multi-cloud System

To use multiple clouds computing services in a single heterogeneous architecture is termed as Multi-cloud. For example, separate cloud providers for IaaS, SaaS services, or use multiple IaaS providers maybe used concurrently by an enterprise. Also, in another type of case they may use different types of infrastructure providers for different types of workloads or deploy a single workload load balanced across multiple providers (active-active), deploy a single workload on one provider, with a backup on another (active-passive). There are a number of reasons for deploying a multi-cloud architecture, including reducing reliance on any single vendor, increasing flexibility through choice, mitigating against disasters, etc. The use of best applications from multiple developers on a personal computer is similar to the above mentioned process, rather than just using the defaults offered by the operating system vendor. It is recognition of the fact that no one provider can be everything for everyone. It is different from hybrid cloud because it refers to multiple cloud services and not multiple deployment modes which would be (public, private, and legacy). The following diagram shows multi-cloud architecture.

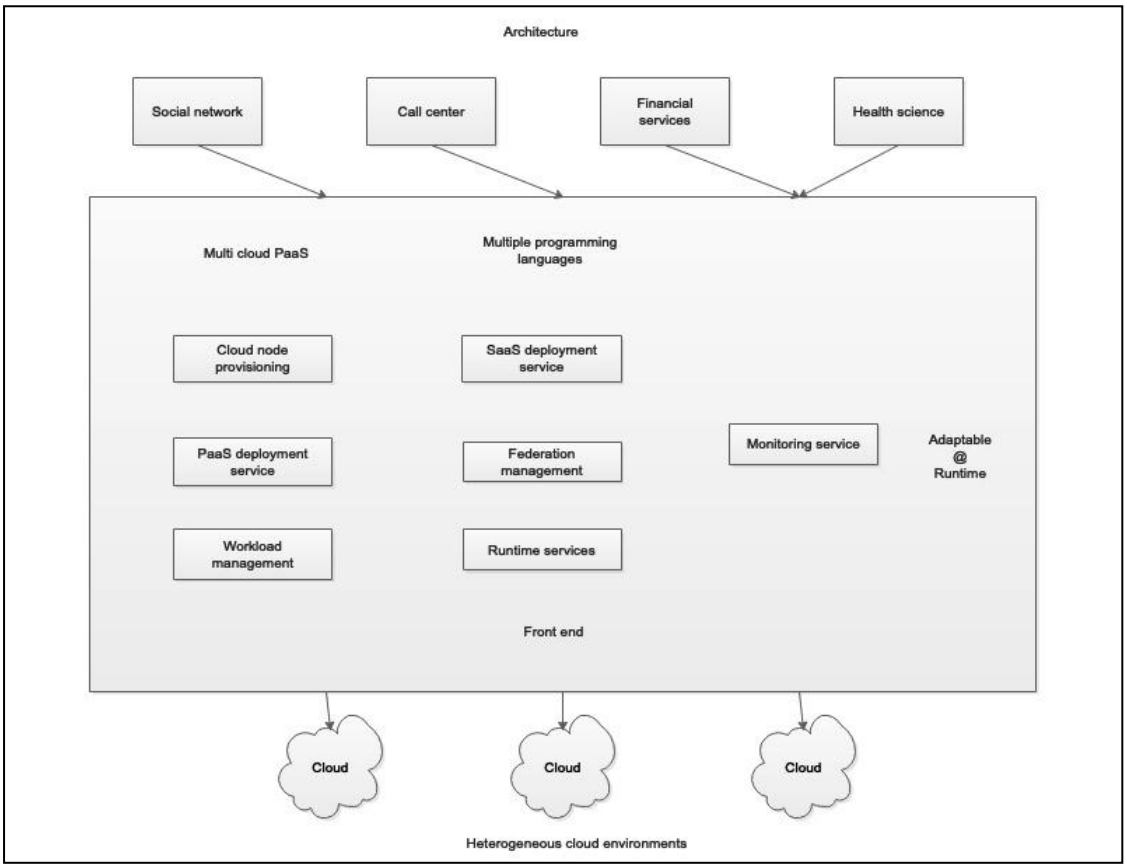

Figure 2: Multi-Cloud Architecture

As a last paragraph of the introduction should provide organization of the paper/article (Rest of the paper is organized as follows, Section I contains the introduction of Cloud Services, In Section 2 Background study is discussed, In Section 3 main proposed approach is discussed, In Section 4 Simulation result is mentioned, In Section 5 a detailed discussion on the proposed approach is revealed and Section 6 contains the conclusion.

II. RELATED WORKS

Optimal resource allocation and task scheduling in cloud will decide optimal number of systems required to be present in cloud so that the total cost will be minimized and the SLA would be easily upheld. Cloud computing is highly dynamic, so, resource allocation problems have to be continuously addressed, as servers become available/non-available while at the same time the customer demand fluctuates. Thus this study focuses on scheduling algorithms in cloud environment considering above mentioned characteristics, challenges and strategies.

Task Scheduling Types

1. **Cloud Service Scheduling**: Cloud service scheduling is categorized at user level and system level [1]. Scheduling at the user level always deals with problems which are raised by service provision in between service providers and the customers. The system level scheduling handles resource management within datacenter. Datacenter consists of many physical machines. Millions of tasks from users are received; assignment of these tasks to physical machine is done at datacenter. The performance of datacenter is significantly impacted by the assignment and scheduling. In addition to system utilization, requirements like QoS (Quality of Service), SLA, resource sharing, fault tolerance, reliability and real time satisfaction should be taken into consideration.

2. **User Level Scheduling:** The supply & demand of cloud resources are regulated by the Market based and Auction based schedulers because of their suitability for such. Market based resource allocation is effective in cloud computing environment where resources are virtualized and delivered to user as a service. A suite of market-oriented task scheduling algorithms to an Auction Net for heterogeneous distributed environments is proposed in [2]. Development of a pricing model using processor-sharing for clouds, the application of this pricing model to composite services with dependency consideration and the development of two sets of profit-driven scheduling

algorithms are proposed in [3]. Service provisioning in Clouds is based on Service Level Agreements (SLA). The contract signed between the customer and the service provider is called SLA which states the terms of the agreement and also the non- functional requirements of the service which is specified as QoS, obligations, and penalties in case of agreement violations. Considering multiple SLA parameters there is a urgent need of scheduling strategies which will result to efficient allocation of the resources. A novel scheduling heuristic considering multiple SLA parameters for deploying applications in cloud is presented in [4]. The scheduler algorithm that allows re-provisioning of resources on the cloud in the event of failures is introduced in [5]. Providing fair deal to the users and consumers is the main focus of the model, as well as providing enhanced quality of service which leads to the generation of optimal revenue. A novel cloud scheduling scheme [6] uses SLA along with trust monitor to provide a faster scheduling of the over flooding user request with secure processing of the request. An approach of request scheduling (heuristic) at each server, at each of the geographically distributed data-centers, which will minimize the penalty charged globally to cloud computing system is proposed in [7]. This approach considers two variants of heuristics, one based on the simulated annealing method of neighborhood searches and another based on gi-FIFO scheduling. Based on the queuing model and system cost function, considering the goals of both the cloud computing service users and providers, [8] proposes an algorithm to get the approximate optimistic value of service for each job in the no-preemptive priority M/G/1 queuing model. The QoS requirements of the user's and the maximum profits for the cloud computing service providers are guaranteed by this approach. To deal with dynamically fluctuating resource demands, market-driven resource allocation has been proposed and implemented by public Infrastructure-as-a-Service (IaaS) providers like Amazon EC2. Cloud resources are offered as various distinct types of virtual machines in this environment & an auction-based market for each VM type is run by the cloud service provider with the goal of achieving maximum revenue over time. A case study of single cloud provider and how to best match customer demand in terms of both supply and price in order to maximize the providers revenue and customer satisfactions while minimizing energy cost is proposed in [9]. Another auction-based mechanism for dynamic VM provisioning and allocation that takes into account the user demand for VMs when making VM provisioning decisions is proposed in [10].

3. Static and Dynamic Scheduling

Pipelining different stages of task execution & prefetching required data is allowed by static scheduling. Static scheduling imposes less runtime overhead. In case of dynamic scheduling information of the job components/task is not known beforehand. As the application executes the allocation of tasks is completed but the execution time of the task might not be known to us. Flextic which is a job execution environment than can exploit scalable static scheduling techniques & provide the user with a flexible pricing model and also reduce scheduling overhead for the cloud provider has been presented in [11].

The service request scheduling strategies in three-tier cloud structure, which consists of resource providers, service providers and consumers, should satisfy the objectives of the service providers and consumers. A dynamic priority scheduling algorithm (DPSA) is proposed to achieve above objectives.

Scheduling of Web Services

Linear Programming Technique is used to map web service calls to potential servers. Different business metrics along with a search heuristic is used for mapping single workflows [12]. Solution to a multiple composite web services resource allocation and scheduling problem having limited local resources from private clouds and multiple available resources from public cloud, has been carried out using a random-key genetic algorithm[13]. A predictive and reactive technique for multi-tiered internet application delivery is presented through a dynamic provisioning approach [14]. The concept of existence of local scheduler [15] providing decision for each individual resource along with a meta-scheduler for scheduling parallel applications is presented eliminating undesirable interactions [16].

III. PROPOSED APPROACH

It consists of two sections:

1. **Architecture Description:** The two-tier multi-cloud architecture on which the proposed methodology is designed.

2. **The Algorithm:** It describes the algorithm which illustrates of the proposed methodology.

1. Architecture Description

The action states of the different objects are:
Client: Request Services, Receive Services
Provider Agent: Invoke Client Request,
Publish Request, Provide Services
Scheduler Agent: Invoke Services,
Provide Services
Service Mapper: Map Constraints
Service Scheduling Log: Schedule Services,
Provide Scheduled Services
Cloud Agent: Collect Services, Publish Services
CESB: Provide Services

To carry out scheduling of composite web-services, we consider the two-tier ECB (Enterprise Cloud Bus) Architecture of a multi-cloud environment. The ECB is a common bus that connects all the CESB (Cloud Enterprise

Service Bus) of each cloud. Each cloud provides different web services through different nodes which provide a particular service. Each such node is connected to one another through Enterprise Service Bus (ESB). The user generates a service request through an agent (software) known as provider. The requests get registered in the HUDDI interface, which are scheduled through the scheduler to be executed. The mapper module is used to map the requests to the scheduler, which in turn use the proper resource from a chosen cloud. The three diagram of the ECB Architecture given below represents the working of the cloud-based service scheduling system. The first one represents the two-tier hierarchical architecture of the ECB which we have used throughout our project. The second one represents the Ontology-based service mapping depicting how the service is provided to the multi-tenants by the provider agent. The third one represents an integrated view of the ECB Architecture, Ontology-based Service mapping along with service composition and scheduling.

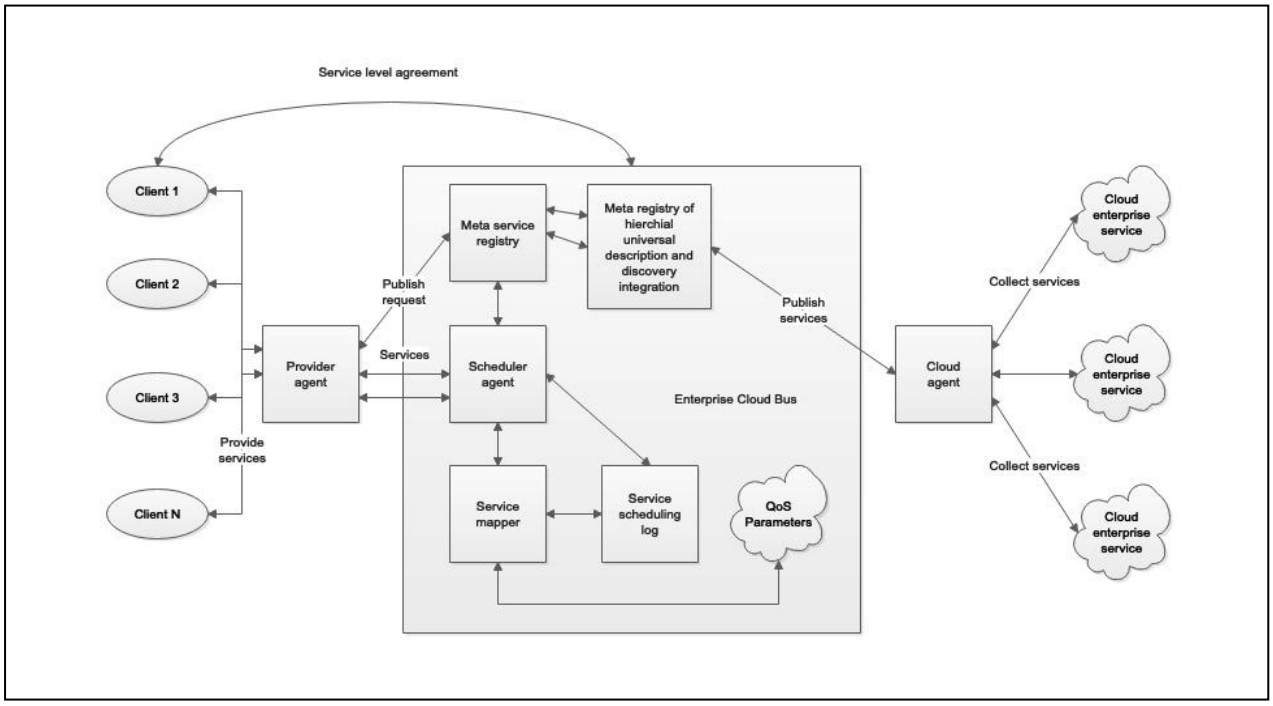

Figure 3: Multi-cloud Architecture

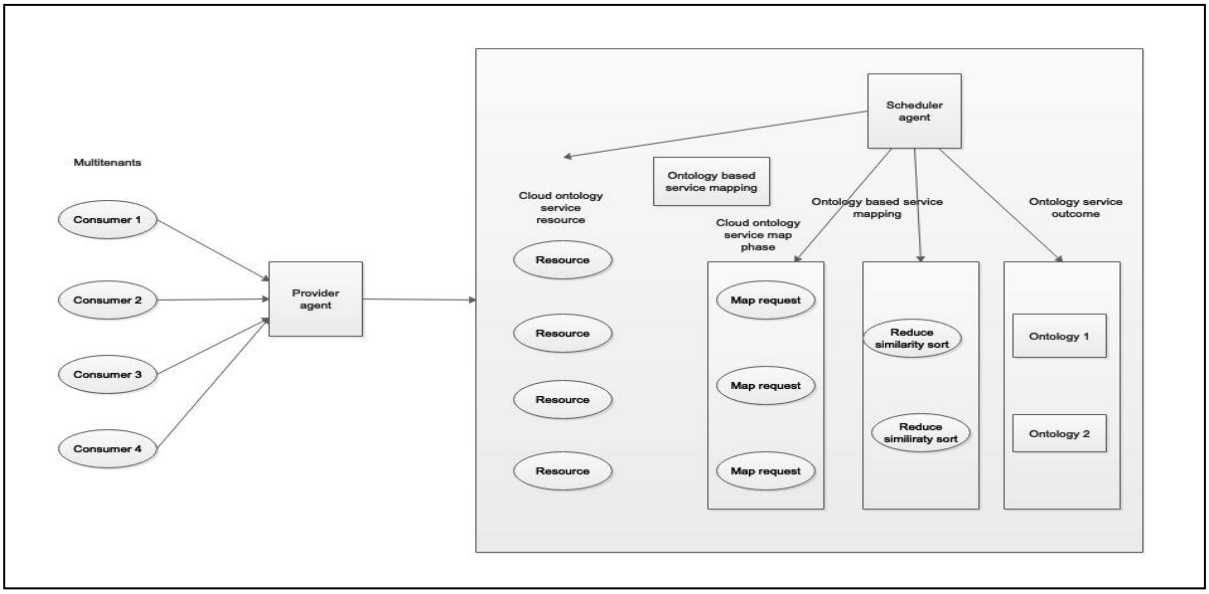

Figure 4: Architecture of multi-cloud along with Web Services

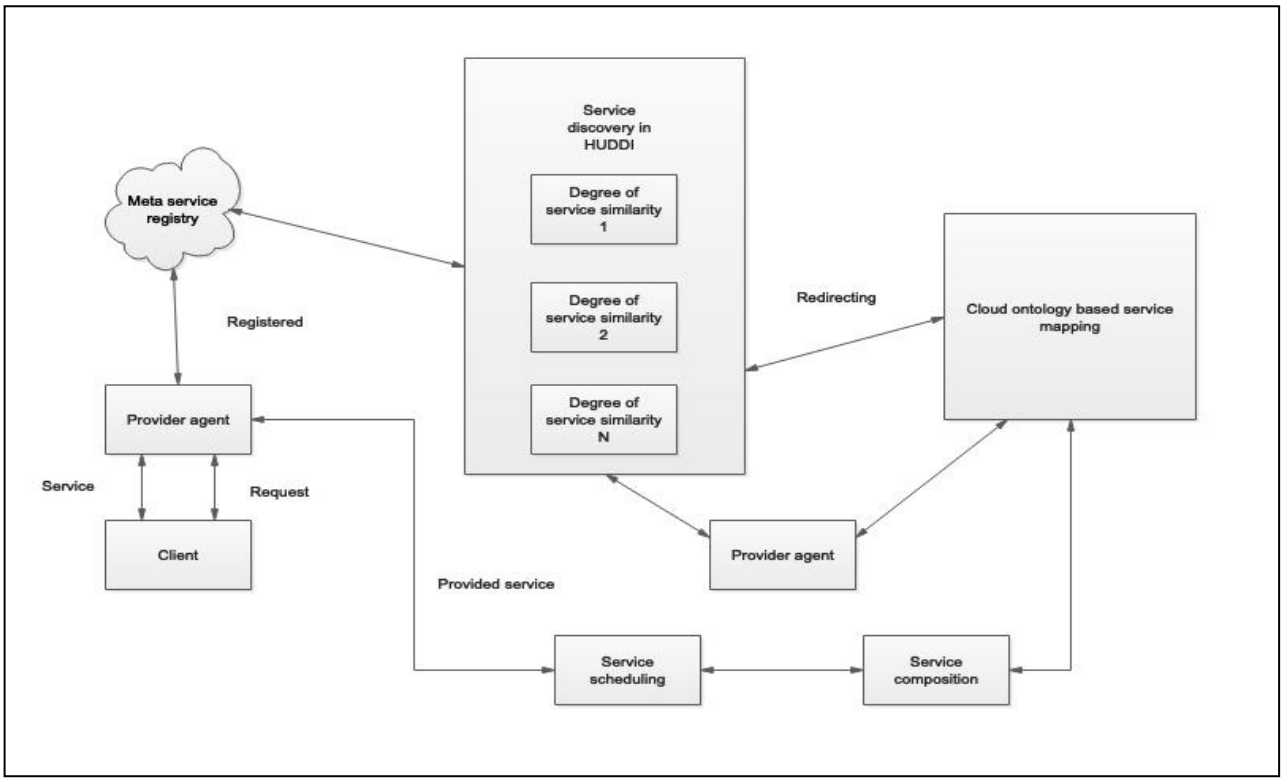

Figure 5: Service Composition and Scheduling

Service Preference Model:

In this model, there is a concept of the types of services for each user respective to the timing slot of a day. Services to get mapped to the cloud with maximum support for the services, remaining part may be obtained from other clouds.

2. Algorithm

Input: Service demanded at an instant of time by each user.

Output: Allocation of various services mapped to various clouds.

1. Let the number of users be *p* and *n* be the categories of services offered by CSP.
   Allocated time slot for each service=*t*
   Users ($u_1, u_2, u_3, \ldots\ldots, u_n$), Service_Category (category1, category2, category n)
2. For each time slot t, initialize service based on the category of the service. Time slot is based on per slot. Each category has different has different cost per time slot.
3. For each service, Service_Provider_Id is stored in the database. Status of the CSP will be stored in the database after a certain time interval(suppose t_1, $t_1 > t$)
4. For each user u_i and for Service_Category(j), i=(1,2,3….n) and j=(1,2,3,…n).
5. Service Scheduling will be done based on the following steps:
   5.1. Service (i) arrived on the cloud server is inserted in the service queue.

   5.2. Each service is moved to different service queue S[j] depending upon the mapping between the Service Category and service selected by the scheduler from the service queue.

   5.3. For every single service S[j] a server machine SM[j] is allocated.

   5.4. Server Machine SM[j] select the service from the service queue S[j] and allocate the virtual machine VM[i] (where i<j) depending on the mapping between the service and the virtual machine.

   5.5 Service (i) is executed by the selected VM.

5.6. Go to step 5.4 until the service queue is emptied.

6. Cost_Matrix will be obtained based on the category of the service.
   Cost=(No_of_Instances_of_each_type*Cost_Per_Instance*Time_Slots_Used),with respect to minimum cost.
7. If no resource is available or partial availability of resources then multiple cloud nodes containing the requested resource can be merged to form multi cloud network. Cost matrix will change according to the service and service used in it.

   7.1 For multi-cloud based resource allocation, the cost for a single resource type r_i is:

$e(r_i)=(a_1*d_1+a_2*d_2+\ldots+(req_j-d_n)*c_n)*\ t_j$, till $req=a_n$, $a_1,a_2,\ldots,a_n$ are the available number of instances of type $r_i$in cloud $c_1,c_2,\ldots c_n$ and $d_1,d_2,\ldots,d_n$ are the respective costs of a single instance, req_j is the resource requirement of service type s_j and the expected time for completion of that service is t_j

   7.2 For multi-cloud based resource allocation, the cost for a multiple resource type $r_i,r_j,\ldots r_n$ is: $q=e(r_i) + e(r_j) + \ldots + e(r_n)$.

   7.3 Considering integration cost $k=k_1, k_2, k_3\ldots k_n$ between n clouds $c_1{:}c_2, c_2{:}c_3\ldots c_{n-1}{:} c_n$, the total cost of service execution will be: q+k.

8. Service will be provided if the resource is available and the available resource to be updated.
9.

IV. SIMULATION RESULTS

Case 1: Resource Allocated from Single Cloud

Resource Information						
Cloud ID	Resource Type 1 Available	Resource Type 2 Available	Resource Type 3 Available	Cost of Resource Type1	Cost of Resource Type2	Cost of Resource Type3
101	20	10	15	200	100	150
102	15	12	18	180	144	216
103	18	15	26	160	150	170

Service_ID: 1

Resource Requirement Information			
ResReqdType1	ResReqdType2	ResReqdType3	Time(Hours)
10	15	20	10

Cloud ID : 103		
Per Unit Cost::		
Cost of Resource Type1	Cost of Resource Type2	Cost of Resource Type3
160	150	170
Total Cost:: 72500		

Case 2: Resource Allocated from Multi-Cloud

Resource Information						
Cloud ID	Resource Type 1 Available	Resource Type 2 Available	Resource Type 3 Available	Cost 1	Cost 2	Cost 3
101	20	10	15	200	100	150
102	15	12	18	180	144	216
103	18	15	26	160	150	170

Service_ID: 2

Resource Requirement Information			
ResReqdType1	ResReqdType2	ResReqdType3	Time(Hours)
20	25	20	8

Resource unavailable in single cloud ...>multi cloud
Available resource for the service in other cloud

ResReqdType1	ResReqdType2	ResReqdType3
20	10	15
15	12	18
18	15	26

Cloud ID : 103		
Per Unit Cost::		
Cost of Resource Type1	Cost of Resource Type2	Cost of Resource Type3
160	150	170
Allocated Resource Type 1	Allocated Resource Type 2	Allocated Resource Type 3
101: 20	101: 10 103:15	101:15 102:5
Cost:: 20*160=3200	10*150+15*150=3750	15*170+5*170=3400
Total Cost: 10350*8=82800		

V. DISCUSSION

This algorithm is a simple yet easy and efficient technique, since services requested at a particular time quarter does not differ for most of the users. This algorithm provides an efficient way to schedule services requested by user, in a simple way. Resource utilization is done efficiently to provide prompt services to the user. Probability of denying service is almost nil, since multi-cloud environment provides an approach to combine smaller components of services to form a composite service. This project is applicable to all fixed devices, not the mobile devices. When combining service components from different clouds, the cost of integration needs to be considered.

VI. CONCLUSION

Cloud computing now-a-days is a very important field of study and promises its users receive the best possible services either free or at low cost. Service Scheduling promises this pretty well, but our algorithm is of use for mainly fixed devices, not efficient for mobile devices. This mapping between the device ID and the Mobile IP make this algorithm work efficiently for mobile devices. Reliability is achieved through replication of data and other resources required for providing the service promptly. However in today's world, cloud computing plays an important role in providing web services, and our algorithm will prove to be a well-defined approach for providing services to the users.

ACKNOWLEDGMENTS

We would like to convey our hearty respect and sincerest gratitude to our respected research advisor Dr. Debabrata Sarddar for sparing his valuable time and assimilating new flawless ideas at every stage of our work. Without his kind co-operation, motivation, and careful guidance, we would not have been able to carry out this research work successfully.

REFERENCES

[1] FeiTeng, *"Resource allocation and scheduling models for cloud computing"*,EcoleCentrale Paris, 2011.

[2] Han Zhao, Xiaolin Li, *"AuctionNet: Market oriented task scheduling in heterogeneous distributedenvironments"*, Research Gate,IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd forum, IEEE, pp.1-5, 2010.

[3] Lee, Young Choon, Wang, Chen, Zomaya, Albert Y. and Zhou, Bing Bing, *"Profit-Driven Service Request Scheduling in Clouds"*, 10th IEEE/ACM International Conference on Cluster,Cloud and Grid Computing, IEEE, pp. 1-13, 2010.

[4] Emeakaroha, V.C., Brandic, I., Maurer, M.andBreskovic, I., "*SLA-Aware Application Deployment and Resource Allocation in Clouds"*,IEEE 31st Annual Computer Software and Applications Conference Workshops, IEEE, pp.1-7, 2011.

[5] Ahuja, R., De, A., Gabrani, G., *"SLA Based Scheduler for Cloud for Storage & Computational Services"*, International Conference on Computational Science and its Applications.IEEE , 2011.

[6] Daniel, D., Lovesum, S.P.J. *"A novel approach for scheduling service request in cloud with trust monitor"*, International Conference on Signal Processing, IEEE , 2011.

[7] Boloor, K., Chirkova, R., Salo, T., Viniotis, Y., *"Heuristic-Based Request Scheduling Subject to a Percentile Response Time SLA in a Distributed Cloud"*,GLOBECOM , IEEE, 2010.

[8] Luqun Li, *"An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers"*,2009 Third International Conference on Multimedia and Ubiquitous Engineering, IEEE, pp.1-5, 2009.

[9] Qi Zhang, Quanyan Zhu, Boutaba, R., *"Dynamic Resource Allocation for Spot Markets in Cloud Computing Environments"*, 2011 Fourth IEEE International Conference on Utility and Cloud Computing, IEEE, pp.1-6, 2012.

[10] Thomas A. Henzinger, Anmol V. Singh, Vasu Singh, Thomas Wies, Damien Zufferey, *"Static Schedling in Clouds"*, Austria,IST Austria, pp.1-6.

[11] Zhongyuan Lee, Ying Wang, Wen Zhou, *"A dynamic priority scheduling algorithm on service request scheduling in cloud computing"*,Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, IEEE, 2011.

[12] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q.Sheng, *"Quality driven web services composition,"* In Proc. of the 12th WWW Conference, pp.411 –421, 2003.

[13] Lifeng Ai, Maolin Tang, Colin Fidge., "*Resource allocation and scheduling of multiple composite web services in cloud computing using cooperative coevolution genetic algorithm*", Neurak Information Processing, Lectures Notes in Computer Science, Volume 7063, pp.258 –267, 2011.

[14] B. Urgaonkar, P. Shenoy, A. Chandra, and P. Goyal., *"Dynamic provisioning of multi-tier internet applications,"* In Proc. of the IEEE Int'l Conference on Autonomic Computing, pp.217 –228, 2005.

[15] Weissman, J. B., Grimshaw, A. S, A Federated model for scheduling in wide Area systems, Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing , pp.542, 1996.

[16] R. Raman, MironLivny, and M. Solomon, *"Resource Management through Multilateral Matchmaking"*, Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), Pittsburgh, Pennsylvania, August, pp.290-291, 2000.

