Priority Based Service Composition Using Multi Agent in Cloud Environment

Khushbu D. Virani [1],Dhara D. Virani [2],Madhuri Vaghasia [3]

[1] ,[2] PG Student, Marwadi Education Foundation Group Of institutions, Rajkot, Gujarat.
[3]Asst. Prof.,Marwadi Education Foundation Group Of institutions, Rajkot, Gujarat.

ABSTRACT:

Cloud computing is a highly research area in the technical I.T field and economic world, and many of the software industry have entered the development of cloud services. It is the Preeminnt on-demand service system along with a "Pay-as-you-go" Policy. Agent-based Cloud computing consist of the design and development of software agents Cloud service discovery, finding the appropriate service and service composition. Distributed and constantly changing Cloud computing environments pretense new challenges to automated service composition such as: (i) Dynamically contract service providers, which set service fees on a supply-and-demand basis, and (ii) Dealing with incomplete information regarding Cloud resources.Sometimes it allows resource agents to take advantage of the delegation of tasks, e.g., when a resource agent delegates a task to another resource agent given a failure.Proposed work contains designing a mechanism like adjusts existent service compositions to changes in consumer requirements, for example, a Cloud consumer demanding for more service to its currently contracted Broker Agents. Proposed system assume that smaller no. of length service has higher priority because less collboration among service agent and average time is also reduced so priority is also consider at the time of service composition.

KeyWords: *Agent, Broker, Composition , Consumer, Multi-Agent, Priority, Provider*

I. INTRODUCTION

Imagine yourself in the world where the users of the computer of today's internet world don't have to run, install or store their application or data on their own computers, imagine the world where every piece of your information or data would reside on the Cloud (Internet). Some analysts and vendors define cloud computing narrowly as an updated version of utility computing: basically virtual servers available over the Internet.

It is a manner of computing in which dynamically scalable and virtualized resources are provided as a service over the Internet for the users. Consumers need not have knowledge of, proficiency in, or control over the technology infrastructure "in the cloud" that supports them. The concept incorporates infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) as well as Web 2.0 and other recent technology trends which have the common theme of confidence on the internet for satisfying the computing needs of the customers.

Cloud computing is an emerging paradigm in the computer industry where the computing is moved to a cloud of computers. It has become one of the buzz words of the industry. Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). Servers in the cloud can be physical machines or virtual machines. In the business model using software as a service, users are provided access to application software and databases. The cloud providers manage the infrastructure and platforms on which the applications run. 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.At a global level, Cloud architectures can benefit government to reduce duplicate efforts and increase effective utilization of resources.

The basis of cloud computing is to create a set of virtual servers on the available vast resource pool and give it to the clients. Any web enabled device can be used to access the resources through the virtual servers. The main criteria of research are related to following.

1. It gives flexible environment to the client.
2. Because of multi agent process will be easy.
3. Average time is also reduced at the time of service composition because of agent.

II. THEORETICAL BACKGROUND

Many people are confused as to exactly what cloud computing is, especially as the term can be used to mean almost anything. Roughly, it describes highly scalable computing resources provided as an external service via the internet on a pay-as-you-go basis. The cloud is simply a metaphor for the internet, based on the symbol used to represent the worldwide network in computer network diagrams.

Economically, the main appeal of cloud computing is that customers only use what they need, and only pay for what they actually use. Resources are available to be accessed from the cloud at any time, and from any location via the internet. There's no need to worry about how things are being maintained behind the scenes – you simply purchase the IT service you require as you would any other utility. Because of this, cloud computing has also been called utility computing, or 'IT on demand'.

2.1. Introduction to Service Composition

If an application or a client requires functionalities which can not be provided by a single service provider, one of the main functions of a service-oriented architecture (SOA) platform is to make it possible to composite several services in order to respond to the expressed needs. This function is referred to as service composition[14].

In Cloud service composition, collaboration between brokers and service providers is essential to promptly satisfy incoming Cloud consumer requirements. These requirements should be mapped to Cloud resources, which are accessed via web services, in an automated manner. However, distributed and constantly changing Cloud computing environments pose new challenges to automated service composition such as: (i) Dynamically contracting service providers, which set service fees on a supply-and-demand basis, and (ii) Dealing with incomplete information regarding Cloud resources (e.g., location and providers) [4].

Cloud computing provide elastic services, high performance and scalable data storage to a large and everyday increasing number of users. Here we are using agent based technique to provide effective service. An agent is a computational entity that acts on behalf of another entity (or entities) to perform a task or achieve a given goal [3]. Generally a cloud service lifecycle consists of service requirements, service discovery, service negotiation, service composition, and service consumption [1]. Generally no single web service can satisfy the functionally required by the user, so there is a need of combine existing service together in order to fulfil the requirement. So our main focus is on service composition [8].

Multi-Agent system represent computing paradigm based on multiple interacting agents that are capable to intelligent behaviour. Software agent used some AI approach based on some co-operation among several agents result in a solving large complex problem which keeps execution time low. Main focus is of cloud computing is efficient use of infrastructure in reduced cost. MAS can run on a cloud infrastructure or most compute-intensive part of it can be hosted in cloud whereas the light part can run on a local server or simply on client pc. So finally agent become more efficient and at the same time lighter & smarter [3].

On the other hand, in agent and multi-agent system (MAS) communities in order to model autonomous and interactive agents and models to enable their coordination. Indeed, analogously to service collaboration, coordination among autonomous agents is defined as the process of organizing and distributing actions of a (distributed) system in order to achieve a specific goal or to obtain better system flexibility[14].

2.2. Approach to Service Composition

First of all we have to define web service model. The web service model consists of three entities, the service provider, the service registry and the service consumer [13].

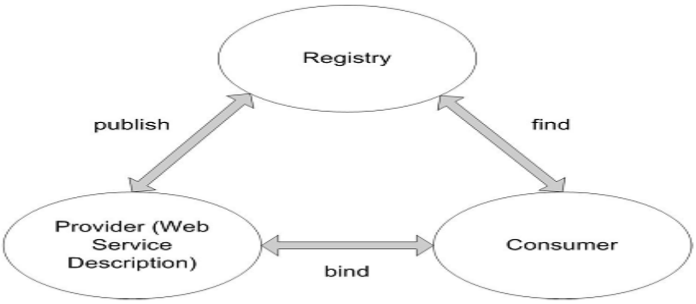

Figure 1.Web Service Model[13]

Basically, there is a 5 step model for service composition process. Each step requires different language as well as different platforms. Steps include service presentation, translation, process generation, evaluation & execution [5].

There are 4 different languages which are generally used for service composition. 1) UDDI (Universal Description Discovery Integration), 2) WSDL (Web Service Description Language), 3) SOAP (Simple Object Access Protocol) & 4) BPEL4WS (Business Process Execution Logic for Web Service) [8].

2.3. Framework for Service Composition

Figure 2 shows the framework for the service composition system. Service requestor will give requirement then translator will translate and convert this requirement in machine understandable form. Process generator will generate process according to service requestor. Evaluator will evaluate the best process among list of process. Finally execution engine will execute the result and delivered it to the service requestor.

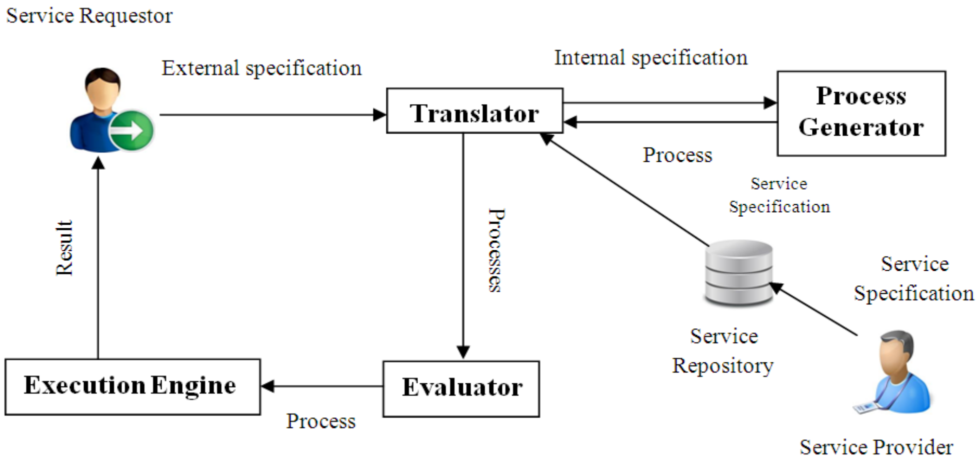

Figure 2. Framework of the Service Composition System [5][8]

III. PROPOSED ALGORITHM

3.1. Introduction to System

This system describes how service composition will execute at the time of dynamically changing the consumer requirement. First of all consumer will place their requirement. Then priority will be assigned to each service based on requirement. Smaller length service will execute first. Services are available with different provider and at the time of consumer requirement services are coming from different provider and at last all services are composite together and delivered to the consumer.

3.2. Proposed Algorithm

<u>Priority Based Dynamic Service Composition(PBDSC)</u>

The proposed algorithm is as follows.

```
Step 1: Read the consumer requirement and store all the  values in list.
Step 2: For each request assign the priority of different parameter and store the
           priority value.
 Step 3: For each parameter calculate the total of each priority .
Step 4: Sort the Sum of values.
Step 5: Send consumer request for execution & store consumer ID.
Step 6: For each Requirement do
            Send CFP (Call For Proposal) message to broker agent.
            check if broker agent has particular service then
send accept proposal message to consumer & store broker ID.
             else
send reject proposal message.

Step 7: Ask to consumer for extra service.
Step 8: Check if any extra service requirement then go to step 6.
Step 9: If broker agent has no service available then broker agent will contact   another broker agent for service
and store broker ID and service will provided by another broker agent.
Step 10: Integrate all service together and delivered to consumer.
```

3.3. Description of Algorithm

Step 1-5 describes that collect consumer requirement and how to assign the priority to job. First of all read the consumer requirement store this requirement into list. Now for each request assign the priority of different parameter and store the priority value. Then for each parameter calculate the total of each priority. At last sort the sum of priority values. And smallest priority job will be execute first so send it for execution.

Step 6 describes for each proposal send call for proposal message to each broker agent and if broker agent has particular service then send accept proposal message else send reject proposal message.

Meanwhile if consumer wants to change their requirements then consumer has to put their requirement in service requirement table after step 6 broker will check SRT if any extra requirement. If SRT table has any requirement then again searching for service from previously allocated broker agent and if broker agent has no more service then broker agent will contact another broker agent and service will be provided by another broker agent.

At last service will be integrated together and composite service will be delivered to the consumer.

3.4. Experiments and Results

Table 1 shows the average executime in seconds(Sec) with increasing no. of cloudlets(jobs). Here Results are compared with existing techniques.

Table 1. Results of Proposed Algorithm

Existing Service Composition Technique		Proposed Service Composition Technique	
No. of Cloudlets(Jobs)	Average Execution Time(Sec)	No. of Cloudlets(Jobs)	Average Execution Time(Sec)
10	5	10	4.7
20	10	20	7.68
30	14.99	30	10.26
40	19.58	40	11.58
50	24.99	50	18.56
60	29.99	60	19.94
70	34.96	70	20.91
80	39.56	80	22.02
90	44.96	90	25.77
100	49.99	100	29.42

Figure 3 shows the average execution time is less then the existing service composition technique.

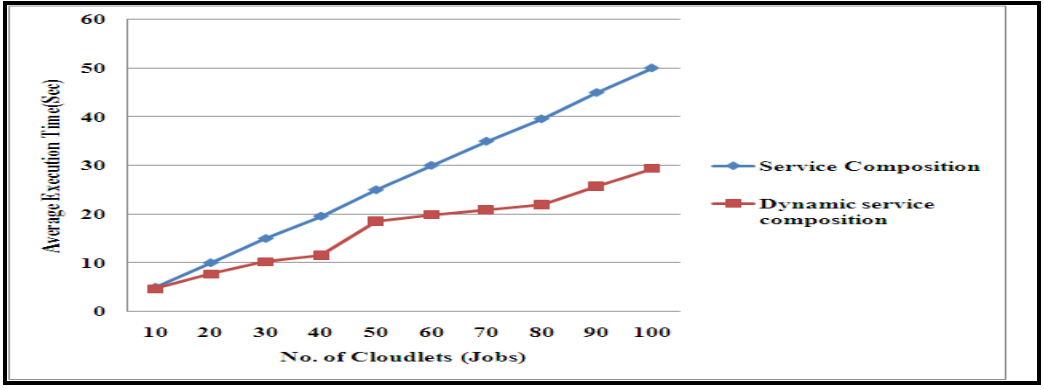

Figure 3. Result of Proposed Algorithm

CONCLUSION

In this wide and distributed environment we need service composition to answer different requests. It has aimed to give an overview of recent progress in automatic web services composition.At first, we propose a five-step model for web service composition process. The composition model consists of service presentation, translation, process generation, evaluation and execution. From the perspective of cloud computing, this work is related to the field of Cloud resource management by devising several approaches for facilitating Cloud service discovery, service negotiation, and service composition. When dynamic service composition is exploited for achieving better flexible result. Proposed algorithm is used for priority based web service composition as well as it is also works with changing consumer requirements. So flexibility at user side will be improved. Again average time to compose service together is also less because there is a less colloboration among service agent.

REFERENCES

[1.] K. M. Sim "Agent-based Cloud Computing" IEEE Transactions on Services Computing, Issue 99, 2011.

[2.] H. Tong, J. Cao, S. Zhang, and M. Li "A Distributed Algorithm for Web Service Composition Based on Service Agent Model" IEEE Transactions On Parallel And Distributed Systems, Vol. 22, No. 12, pp-2008-2021, December 2011.

[3.] D. Talia "Cloud Computing and Software Agents: Towards Cloud Intelligent Services", in proc. Of CEUR-WS, Vol.741, 2011, pp-2-6.

[4.] J. Octavio, G. Garcia and K.M. Sim" Self-Organizing Agents for Service Composition in Cloud Computing" 2nd IEEE International Conference on Cloud Computing Technology and Science, pp-59-66, 2010.

[5.] A.VishnuVardhan, M.S.SaleemBasha, P.Dhavachelvan "An Overview of Web Services Composition Approaches" international Journal of Computer Applications (0975 – 8887) Volume 29– No.8, September 2011.

[6.] W.T. Tsai, P. Zhong, J. Balasooriya, Y. Chen, X. Bai, J. Elston"An Approach for Service Composition and Testing for Cloud Computing "2011 Tenth International Symposium on Autonomous Decentralized Systems.

[7.] C. Seh Wu and I. Khoury "Tree Based Search Algorithm for Web Service Composition in SAAS", 2012 Ninth International Conference on Information Technology- New Generations.

[8.] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods" Proc. First Int'l Workshop Semantic Web Services and Web Process Composition (SWSWPC), pp. 43-54, 2004.

[9.] G. B. Chafle, S. Chandra, V. Mann, and M. G. Nanda, "Decentralized orchestration of composite web services" in Proc. of the 13th international World Wide Web Conference on Alternate Track Papers & Posters, ACM, New York, NY, pp. 134-143,2004.

[10.] R. Aggarwal et al., "Constraint Driven Web Service Composition in METEOR-S" Proc. IEEE Int'l Conf. Services Computing (SCC), pp. 23-30, 2004.

[11.] Z. Maamar, S.K. Mostefaoui, and H. Yahyaoui, "Toward an Agent-Based and Context-Oriented Approach for Web Services Composition" IEEE Trans. Knowledge and Data Eng., vol. 17, no. 5, pp. 686-697, May 2005.

[12.] J. Yan, J. Zhang, J. Lin, M. B. Chhetri, S. K. Goh, and R. Kowalczyk "Towards Autonomous Service Level Agreement Negotiation for Adaptive Service Composition" 10th International Conference on Computer Supported Cooperative Work in Design, Nanjing, China, 1-6, 2006.

[13.] E. Pejman, Y. Rastegari, P. MajlesiEsfahani, A. Salajegheh "Web Service Composition Methods: A Survey" proceeding of the international Multi Conference of engineers and computer scientists 2012 Vol I,IMECS 2012,March 14-16, 2012,HongKong.

[14.] Y. Charif, N. Sabouret "Dynamic service composition enabled by introspective agent coordination", Auton Agent Multi-Agent Syst., pp. 54-85, 2013.

[15.] http://www.dialogic.com/media/products/docs/whitepapers/12023-cloud computing wp.pdf.

[16.] http://www.thinkgrid.com/docs/computing-whitepaper.pdf.