# Automatic Service Composition Using User-Centric Approach in Cloud Computing

Bhuvnesh Kumar[1], Hardeep Singh[2], Nancy[3], Navroop Kaur[4]
Department of Computer Science and Engineering
Guru Nanak Dev University, Amritsar (Punjab), India.
Email: [1]bhuvnesh.gsp@gmail.com
[2]hardeep_gndu@rediffmail.com
[3]nancy.k1307034@yahoo.co.in
[4]navonline@yahoo.co.in

***Abstract*—Service oriented architecture emerging as important service of cloud computing. It is a producer-centric approach in which the service provider published their services and service consumers must search available services to compose their applications. In the proposed work we follow User-Centric SOA (that allows end users to compose applications) in which some parameters are added like total number of requested services, number of new services requested, ratio between them, efficiency of resources and determine the formation time (the sum total of requested time, discovery time and composition time).**

**Keywords—** ***Service-Oriented Architecture, User-centric Approach, Cloud computing.***

## I. INTRODUCTION

The basic principle of cloud computing is to distribute the computing tasks to many distributed computers, not local computer or remote servers. The services of cloud computing are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Infrastructure-as-a-Service is the delivery of huge computing resources such as the capacity of processing, storage and network. Platform-as-a-Service generally abstracts the infrastructures and supports a set of application program interface to cloud applications. Software-as-a-Service aims at replacing the applications running on PC.

From all the layers of cloud computing shown in the figure1, this paper focuses on the third layer (Component as a service, SOA). The SOA is producer-centric because service providers publish their services and service consumers must search available services to compose their applications. The Service-Oriented Architecture (SOA) provides a set of principles to create service oriented systems, by defining how services can be created, composed, published, discovered and invoked. CCSOA focuses on consumers' publishing the services they need and even the applications they need. The service providers must produce services that are in need. This new paradigm extends the design and code sharing, and thus further improves the software productivity. User-centric service oriented architecture (UCSOA) that allows end users to compose applications. UCSOA is an extension of consumer-centric service-oriented architecture (CCSOA), which is an extension of conventional SOA.

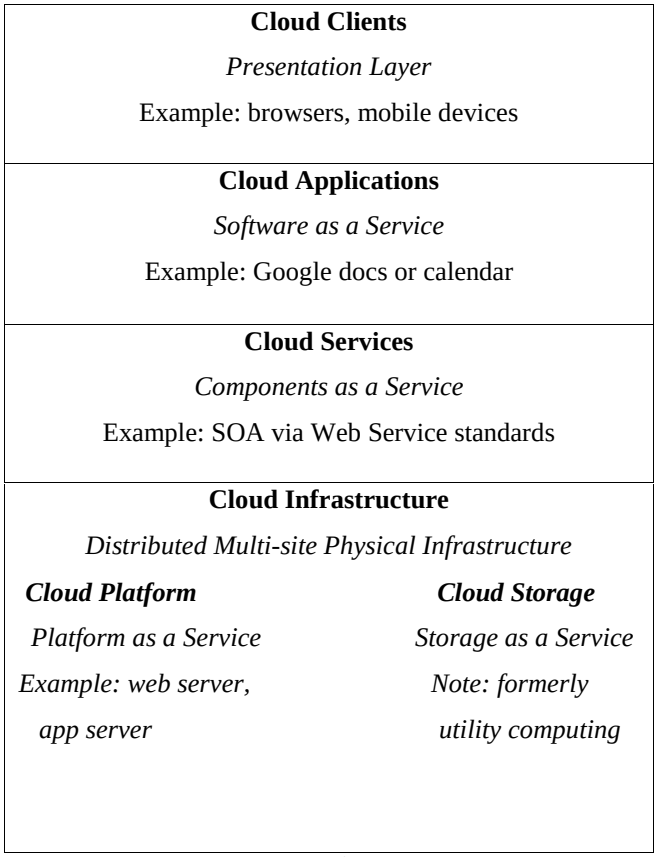



Figure1: Layer's of Cloud Computing

Automatic service composition is the automatically composing services that satisfy a given service request

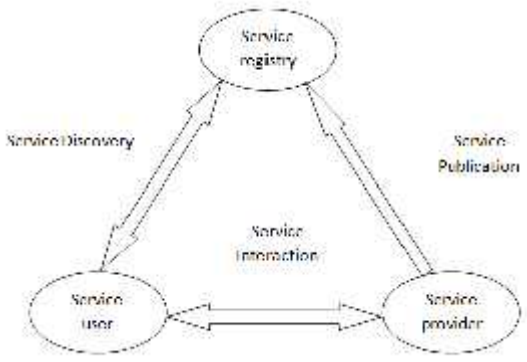



Fig. 2. Service-Oriented Architecture elements and interactions

from an end-user or service developer.
Services are composed in terms of already available atomic services, which are orchestrated in the service composition. Service requests are used for service discovery, matching and composition.

Service requests allow end-users or service developers to specify what they want the service to do for them, abstracting from the way this service is implemented, possibly in terms of a composition of atomic services.

The previous works on Service Composition discussed how to create, composed, publish and discover the services requested by the end users. This paper adds some more parameters to the automatic service composition such as to calculate the total number of requested services, number of new services requested, ratio between them, efficiency of resources and to determine the formation time (the sum total of requested time, discovery time and composition time).

## II. RELATED WORK

Many service composition approaches and solutions have been proposed and developed in recent years. However, more effort has to be spent on their evaluation and comparison.

W.T. Tsai, et.al. [2] introduced a Consumer-Centric Service-Oriented Architecture (CCSOA) paradigm over Producer-Centric Service-Oriented architecture. This new paradigm extends the design and code sharing, and thus further improves the software productivity.

Mark Chang, et.al. [3] proposed the paper which introduces a new user-centric service oriented architecture (UCSOA) that allows end users to compose applications. UCSOA is an extension of consumer-centric service-oriented architecture (CCSOA).

Xuanzhe Liu, et.al. [1] proposed some user-centric mining algorithms which uses Direct Acyclic Graph to built up potential composition opportunities. This approach allowed users to achieve service composition in a heuristic manner.

Eduardo Silva, et.al. [5][6][7][8] proposed the paper which address the challenge of performing dynamic service composition. This paper defines a life-cycle for dynamic service composition, which defines the required phases and stakeholders. In this paper authors present our prototype in which the different phases of the dynamic service composition lifecycle are being implemented. They also developed a framework named DynamiCoS, which aims at supporting the different phases required to provide end-users with automatic service discovery, selection and composition process and also present the developed prototype and its evaluation.

Eduardo Silva, Lu´ıs Ferreira Pires, Marten van Sinderen [9] proposed the paper which present framework for semantics-based service composition approaches which use a collection of existing services, and define a set of evaluation metrics, confusion matrix-based and time-based. Furthermore, we present how composition evaluation scenarios are generated from the collection of services and specify the strategy to be used in the evaluation process.

Eduardo Gon, et.al. [10] proposed the paper which defines the DynamiCoS framework based on a service composition life-cycle. Semantic services are used in our framework to enable reasoning on the service requests issued by end users, making it possible to automate service discovery, selection and composition. This paper also shows the benefits of semantic based frameworks for service composition, particularly for end-users who will be able to have more control on the service composition process.

This paper is focused on User-centric approach for service composition.

## III. PROPOSED POLICY

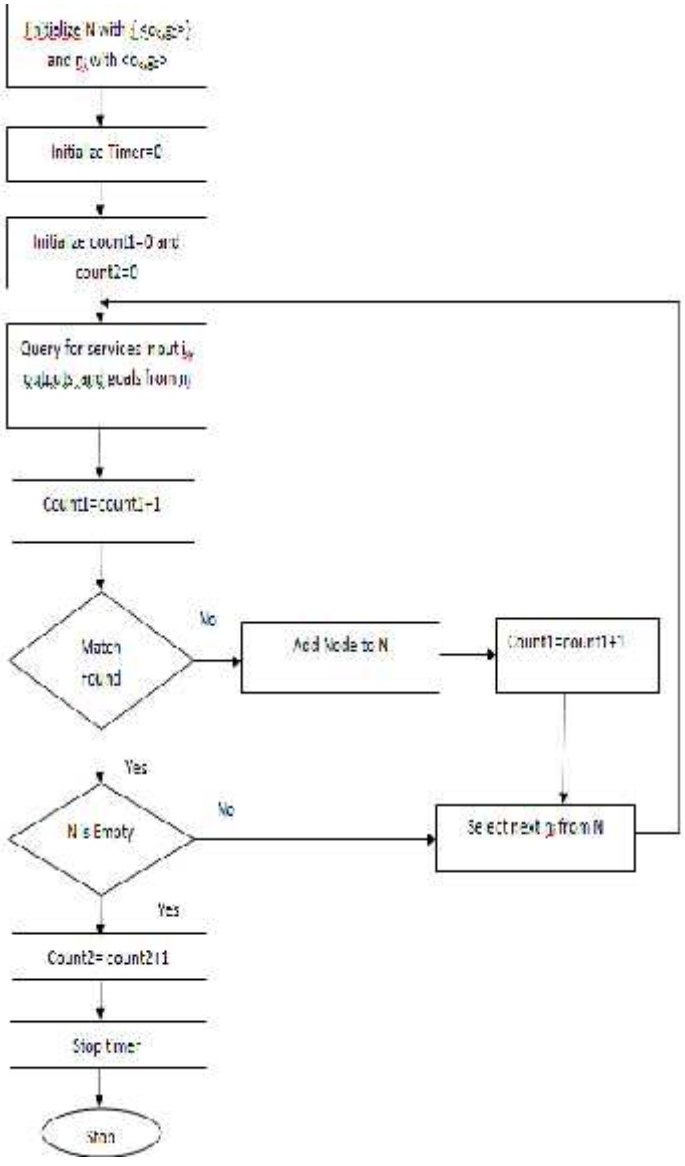



Figure 3: Flowchart of service composition with some parameters

In the proposed policy, some new are parameters are added which calculate the total number of requested services, number of new services requested, ratio between them, efficiency of resources and determine the formation time (the sum total of requested time, discovery time and composition time).

*Ratio= count1/count2*

Where, count1 is number of requested new services and count2 is number of requested services.

*Formation time = Service request Time + Discovery Time+ Composition Time.*

Timer is used for calculating the formation time through which we can calculate the efficiency of resources.

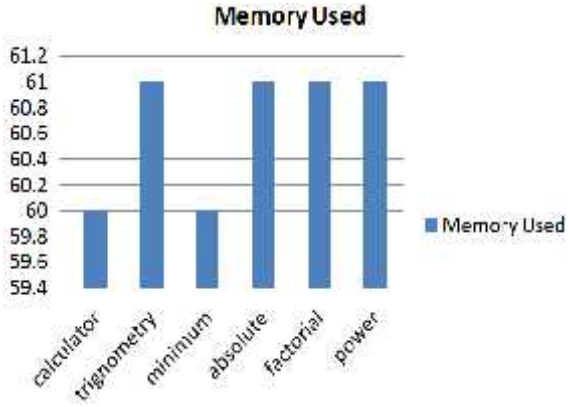
Efficiency of the resources mostly depends upon the formation time. If the formation time is short, then efficiency of resources more. Otherwise, efficiency of resources is less.

*Efficiency=1 / Formation time*

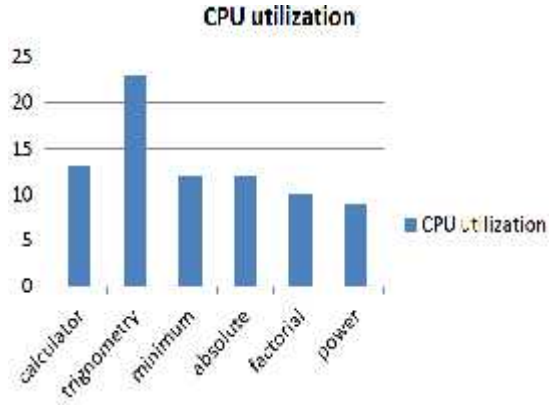
Efficiency means the CPU utilization, memory used and response time of service.

## IV RESULTS

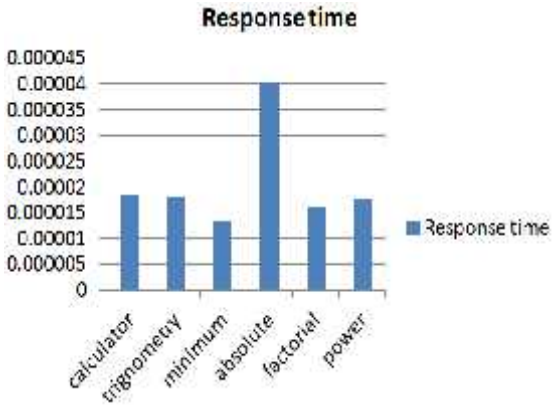
A. Graph of memory used by services



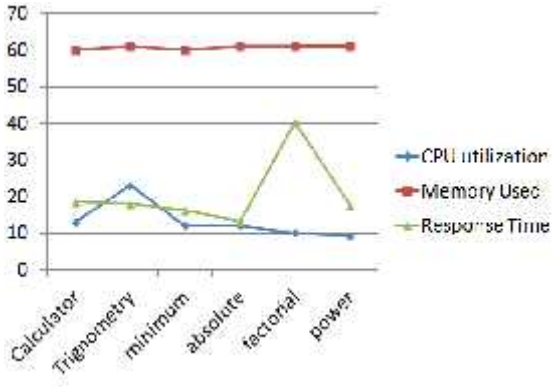

B. Graph of CPU utilization of services




C. Graph of Response Time of services




D. Graph of efficiency of services




From the above graph of efficiency, if the CPU utilization of service is more, memory used is less and response time is small then service is more efficient.

## V CONCLUSIONS

Many works have been done on user-centric service oriented architecture. This paper extends the previous papers by adding some new parameters like total number of requested services, number of new services requested, ratio between them, efficiency of resources and determine the formation time.

## REFERENCES

1. Xuanzhe Liu, Gang Huang, Hong Mei, 2008. A User-Oriented Approach to Automated Service Composition, School of Electronics Engineering and Computer Science, Peking University, DOI = 10.1109/ICWS.2008.139,773-776.
2. W.T. Tsai, Bingnan Xiao, Raymond A. Paul, Yinong Chen, 2006. Consumer-Centric Service-Oriented Architecture: A New Approach, Arizona State University, Tempe, AZ 85287-8809, USA Department of Defense, Washington, USA, 2006.
3. Mark Chang, Jackson He, W.T. Tsai, Bingnan Xiao, Yinong Chen," UCSOA: User-Centric Service-Oriented Architecture"Intel Incorporation, USA ,Arizona State University, Tempe, AZ 85287-8809, USA 2006 IEEE.
4. J. Eduardo Silva, Luís Ferreira Pires, Marten van Sinderen, 2008. An Algorithm for Automatic Service Composition, Centre for Telematics and Information Technology, University of Twente, Enschede, the Netherlands,2008.
5. Eduardo Silva, Jorge Mart´ınez L´opez, Lu´ıs Ferreira Pires, Marten van Sinderen, 2008. Defining and Prototyping a Life-cycle for Dynamic Service Composition, Centre for Telematics and Information Technology University of Twente, The Netherlands,2008.

6. Eduardo Silva, Lu´ıs Ferreira Pires, Marten van Sinderen, 2008. Dynamic Service Composition: Why, Where and How, Centre for Telematics and Information Technology University of Twente, The Netherlands, 2008.
7. Eduardo Silva, Lu´ıs Ferreira Pires, Marten van Sinderen, 2009. On the Support of Dynamic Service Composition at Runtime, Centre for Telematics and Information Technology University of Twente, The Netherlands, 2009.
8. Eduardo Silva, Lu´ıs Ferreira Pires, Marten van Sinderen, 2009. Supporting Dynamic Service Composition at Runtime based on End-user Requirements, Centre for Telematics and Information Technology University of Twente, The Netherlands, 2009.
9. Eduardo Silva, Lu´ıs Ferreira Pires, Marten van Sinderen, 2009. A Framework for the Evaluation of Semantics-based Service Composition Approaches, Centre for Telematics and Information Technology University of Twente, The Netherlands, 2009.
10. Eduardo Gon, calves da Silva, Lu´ıs Ferreira Pires, Marten van Sinderen, 2010. Towards Runtime Discovery, Selection and Composition of Semantic Services Centre for Telematics and Information Technology University of Twente, The Netherlands, March 26 , 2010.