A Review on Web Service Composition using Ant Colony Optimization with Agent based Approach

Jayjit J Gatha[1] Piyush V. Gohel[2]
[1]M.E. Scholar [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]Noble Group of Institutions, Junagadh

Abstract— The aim of the research work presented in this paper is to develop a system for automatic web service composition with agent based approach to provide QOS aware services to different users according to their requirement. Grid computing is increasingly considered as a promising next-generation computational platform that supports wide-area parallel and distributed computing. In this approach we provide web service composition through workflow model and use new version of ant colony optimization algorithm called multi objective ant colony optimization algorithm (MOACO) that is being used to decompose composite services into parallel execution path. In this we can get services through one or more agents from service composition algorithm according to users requirement. The main advantage is that in this system user only have to give higher level goals and they can get services.

Key words: Web Service, Web Service Composition, Quality of Service, Multi-Agent System, Service-oriented architecture, Multi Objective Ant Colony Optimization

I. Introduction

Web Services are based on distributed technology and provide standard means of interoperating between different software applications across and within organizational boundaries with the use of XML .Basically web services can be used to provide data to user level application from server side.

As with the growing number of alternative web services that provide the same functionality but differ in quality parameters, the service composition becomes a decision problem on which component services should be selected such that user's end-to-end QoS requirements (e.g. availability, response time) and preferences (e.g. price) are satisfied.

Web Services technologies allow interaction between applications. Sometimes a single service given alone does not meet user's needs. In this case, it is necessary to compose several services in order to achieve the user's goal. In web service composition, a repository of services is given in which input and output parameters of each service are annotated with a concept from ontology.

Ant colony optimization (ACO) is a population-based meta heuristic that can be used to find approximate solutions to difficult optimization problems.

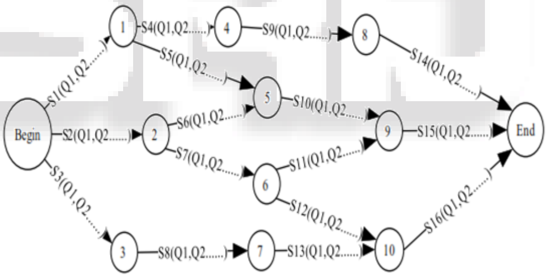
In ACO, a set of software agents called *artificial ants* search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph.ACO can be used in many composition optimization fields but there exist inherent limitation like slow conversion and poor performance. There is only one kind of pheromone in ACO, which cannot deal with the question of multiple attributes in web services composition.

So, to overcome these disadvantages DACO is put forward to fit for the dynamic services composition optimization and to promote the algorithm's effectiveness. In DACO After a round of optimization, we compare the weights of all paths, only updating the pheromone of the path with maximum pheromone. The experiment proves that the convergence rate and the probability of finding optimization of the DACO have increased greatly.

In previous system graph based model for web service composition is being done and in this QOS parameter can be achieved from users according to their requirement and then we have to apply DACO for dynamic web services composition optimization.

A. Model Building

The services composition graph is one direction and simple connected graph, each path in the graph denotes a complete services composition, each node denotes a services orchestration, each arch represents a service, the beginning point denotes the input of the services composition, and the ending point denotes the output of the service composition.



A service has multiple QOS and each user has different preference weight on each QOS .The preference value of each QOS will be obtained respectively after which the preferences of multiple QoS will be further compounded to express users' preference to this service. Different QoS have different attribute like

'>' attribute, which means that the bigger the QoS value, the greater the preference tendency.

'<' attribute, which means that the smaller the QoS value, the greater the preference tendency.

'=' attribute, which means that the closer the QoS value to the preference constraint value, the greater the preference level

Section attribute means that when the chosen value is in a certain scope, the user will show the preference.

'>' attribute operation: $p_{i=\frac{q_{i-l}}{range}}$

'<' attribute operation: $p_{i=\frac{l-q_i}{range}}$

Where p_i denotes the preference of q_i
l is the preference critical value of the user, and
Range is the range of corresponding QoS.

‘=’ attribute operation: if the value is equal to the preference critical value of user, $p_i = 1$, else $p_i = 0$

B. Use of DACO in Dynamic Web Service Composition

There are certain limitation in ACO like convergent to local optimization, slow convergence etc., we use the following improvement rules to improve performance of ACO

- After round of optimization, by comparing all paths we have to update pheromone of the path with maximum pheromone
- In order to avoid prematurity, stagnancy and converging to local optimization of the algorithm we set variable *L* as the optimization path list and *l* as the length of the list. In optimization proceeding, The *l* maximum pheromone paths generated by Comparing with the maximum pheromone path in each round are saved. When the algorithm reaches Maximum round count, we have to compare the maximum Pheromone path with the paths in *L* compute the QoS of the corresponding services again and get the QoS services composition with higher preference as Optimized findings.

II. LITERATURE SURVEY

I have studied many papers and done the following analysis for web service composition. There are mainly static web services and dynamic web service compositions have been done in previous research.

There are various approaches for web service composition like workflow-based, artificial intelligence (AI) planning-based, semantic- based, and graph-based.

The static means that the requester should build an abstract process model before beginning the composition the aggregation of the services is done at design time and composition is performed manually

So, there is a need of dynamic composition that automatically and efficiently generate composite services that exactly meet the expectation of requesters

The dynamic composition creates process model and selects atomic services automatically and requires the requester to specify several constraints including the dependency of atomic, the user’s preference

For automatic web service composition, Semantic web is proposed. An automatic web service composer should compose ‘‘right” services in a composition according to the user’s specification. Composite web services are dynamic that their components can be automatically selected at run-time based on specific requests

In previous research graph based service composition approach used with dynamic ant colony optimization (DACO) to provide QOS aware service composition.

But if we want to automate the service composition more efficiently there will be use of AWSQoSX (Agent based Web

Services Quality of Service Architectural Extension) that provide service selection by negotiating with agents to select services. In this we use MOACO for QOS aware service composition.

III. PROPOSED METHOD

There are many problems in previous approach like user has to give preference weight in each service, it doesn’t guaranties functionality, It is not used to manage composite web service for business process. So, To overcome these problems this proposed system have to be developed.

The main objective of this system is

- To design and analyze Multi-objective agent based Ant colony optimization QoS Model and service decomposition
- QOS performance comparison and analysis,
- Comparison of proposed composition with the best result reported in web service consortium,
- Implementation of proposed system using Grid Sim.

In this approach first workflow model is being used to select appropriate service and then service composition will be performed and in this new version of ant colony optimization is going to be used called MOACO(Multi Objective Ant Colony Optimization) for QOS aware service composition.

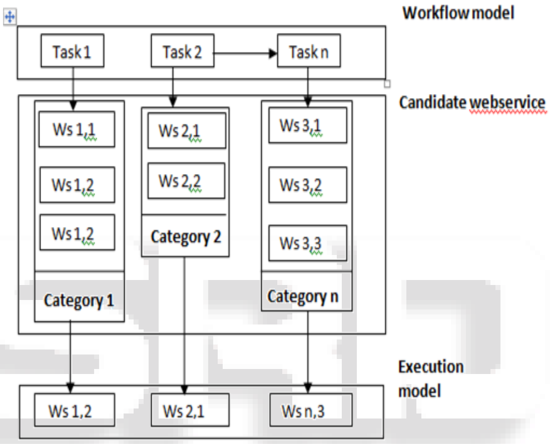


Fig: Web Service Composition

According to QOS parameters for composition

Web Service compositions can be seen as workflows based on Web Services. As shown in figure there is a workflow model that consists of abstract tasks describing the required functionality of a specific workflow step The functionality of one task can be provided by different candidate Web Services. So, this model provides service composition by selecting services from one or more candidate services

A. Web Service Composition with Software Agents

We proposed a framework to support Qos aware Web Service Composition. As shown in figure We add two layers between service requirement and web service candidates In the first layer, each Web Service candidate is linked to a home norm base, That can be used by home agent for negotiate with other home agents. Norms are set of rules and regulations and works like protocol governing the agent communications network Norms revolves around agents, which provides facility for the agents to execute a series of concerted actions to achieve a particular goal. And the second layer is composition algorithm which links with the home agent for choosing the best services for the composition.

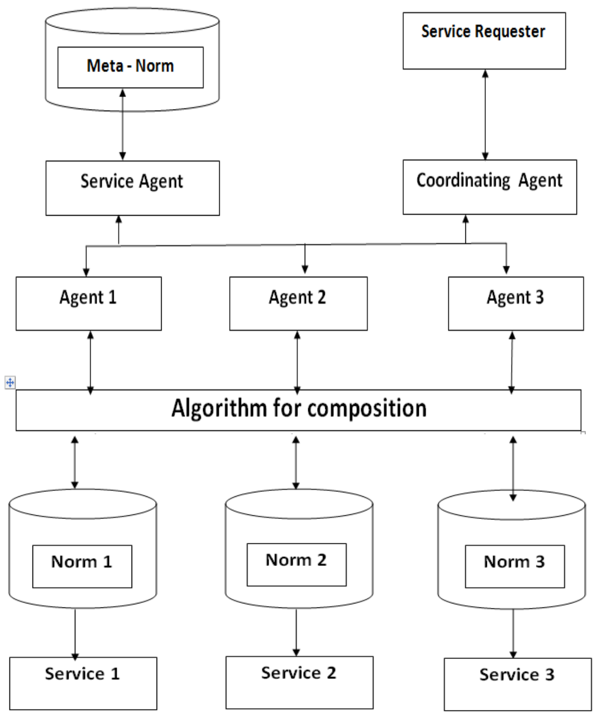


Fig: Web Service Composition with Agents.

1) *Agents:*

It is a piece of software that works automatically to undertake task on behalf of users.

The design of many Agents is based on the approach that the user only needs to specify a high-level goal

2) *Home Agents:*

The main functions of home agent can be divided into two parts

3) *Service equalizer model:*

Service matching is based on the Similarity between service descriptions that mainly contain service name and other related information.

4) *Service selection module:*

With the use of selection algorithm, Home Agent selects one optimum service or a group of services.

5) *Service Agents:*

They have four functional modules.

6) *Agent conversation:*

Agent conversation includes two parts One is the conversation between service agent and composition agent and second is the between service agents.

7) *Service invoker:*

Invoke the method of Web Services.

8) *Service adaptation:*

In the process of service invoking, service agent regulates Web Service method according service context.

9) *Coordinating Agents:*

The main function of coordinating Agents can be divided into four modules

10) *Divergence in Flow:*

The service composition flow is divided into some sub-flows that are going to be performed by service agent.

11) *Agent conversation:*

Service Agents that used to perform sub flow can be determined with the help of Conversation of composition agent.

12) *Execution:*

The module is responsible for to control and regulate service composition.

13) *Result Collection:*

Collecting the result of service composition and give the result to user application

After This Whole Agent based Service Selection the Multi Objective Ant Colony Algorithm have to be applied for QOS based Web service composition.

Divide and conquer is widely used to solve complex problems. This strategy can be performed by decomposing composite services with a general flow structure into parallel execution paths. The maximum number of parallel execution paths that can be determined by the number of AND split structure patterns in the composite service.

The main advantage of proposed approach is to recognize the feasibility of the composition process at any point of execution and produce better throughput and less consumption of memory to select composition of web services.

IV. CONCLUSION AND FUTURE WORK

We propose a strategy to decompose a composite service with into parallel execution paths. We then model dynamic service composition for each execution path as a multi-objective optimization problem and show that how agents are going to be used in service selection and for to apply QOS in service selection We have use the concept of MOACO.I have presented survey of this research model in future I will show whole working of this system and display MOACO with Agent based approach.

REFERENCES

[1] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Varela F, Bourgine P, eds. Proc. of the ECAL'91 European Conf. of Artificial Life. Paris: Elsevier, 1991. 134~144.

[2] Dortgo M, Gambardella L. M. "Ant colony system: Acooperative learning approach to the traveling salesman problem", IEEE Trans. on Evolutionary Computation, 1997, 1(l): 53~66.

[3] Zhongnan Shen & Jianwen Su. "On Automated Composition for Web Services". WWW 2007.

[4] John T. E. Timm and Gerald C. Gannod. "Specifying Semantic Web Service Compositions Using UML and OCL". WWW 2007.

[5] Gerardo Canfor, Massimiliano Di Penta, etc. "An Approach for QoSaware Service Composition Based on Genetic Algorithms". GECCO'05, 2005.

[6] MohammadAlrifai, ThomasRisse "combining Global optimization with local selection for efficient QOS-Aware Service composition" ICW3C2, april2024, 2009, Madrid Spain, ACM 978-1-60558

[7] Michael Matskin, sobah abbas Peterson “Enabling Web Service Composition with software agents” Norweign University, misha@idi.ntnu.no

[8] G Vadivelou, E. IIvarasan, S. Prasanna “ algorithm for web service composition using Multi-Agents” International Journal of computer Application(09758887) Volume13 No.8,January 2011

[9] M. Adel serhani, rachida Dssouli “A QoS broker based architecture for efficient web service selection “proceeding of IEEE international conference on web services ICWS’05-7695-2409-5/05

[10] Wei Zhang, Carl Chang “QoS based dynamic Web service Composition with ant colony optimization” 34th annual IEEE computer software and application conference-0730-3157/10-2010 IEEE

[11] M. Erdem Tijrsem, M hadi Gunes “performance analysis of mobile agents using simulation” Bogazici University, Istanbul, Turkey

[12] David Kotz and Robert S. Gray “Mobile Agents and the Future of the Internet” Dartmouth College Hanover, New Hampshire 03755 dfk@cs.dartmouth.edu,

[13] Eyhab Al-Masri, Quasy mohmad “QoS Based discovery and ranking of web services” 1-4244-12512007 IEEE

[14] Zakaria maamar, Hamdi Yahyaoui “Towards an agent based and context oriented approach for web services composition” IEEE transaction on knowledge and data engineering, Vol. 17, no. 5, may 2005.

[15] L. Ardissono, A. Goy and G. Petrone. “Enabling conversations with web services”. Proc. of the 2nd Int. Conf on Autonomous Agents and Multiple agent Systems, 2003, Melbourne, pp.819-826

[16] Gao, Yan, et al., "Optimal Web Services Selection Using Dynamic Programming." s.l.: Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06), 2006.

[17] Stefania Costantini1, “Agents and Web Services”, Dip. Di Informatica, Universit`a di L’Aquila, Coppito 67100, L’Aquila, Italy 10.