사용자 전역 QoS를 보장하기 위한 K-way 수퍼 노드 접근법

K-way Super Node Approach for Guaranteeing User's Global QoS

강 남 오*　　박 사 준**
NamOh Kang　　Park, Sajoon

요 약

요즈음, 웹 서비스의 확산으로 인하여, 등록된 웹 서비스를 합성함으로써 복잡한 응용프로그램을 구축하는 것이 가능하다. 이러한 추세는 오늘날 서비스 지향의 컴퓨팅 환경에 있어서 합성된 웹 서비스의 서비스 품질 보장을 중요한 문제로 대두시키고 있다. 하지만 주어진 서비스 품질 보장을 만족하는 전역적 최적화된 웹 서비스의 합성은 NP-hard한 문제로 분류된다. 본 논문에서는 이러한 문제를 해결하기 위하여, K-way Super node 접근법을 제시하고 실험 결과를 통하여 이의 효과를 보인다.

Abstract

Nowadays, with the proliferation of Web services, developer and user can implement a complex application by composing registered Web services. In this trend, the quality of service of a composite Web service is of crucial importance for today's Service Oriented Computing (SOC) environment. But the global optimization of a composite Web service satisfying given QoS is classified to NP-hard problem. To solve this problem, we propose K-way Super node approach and show the effect of it from experimental results.

☞ keywords: Web Service, QoS, K-way, Service Oriented Computing

1. Introduction

In service-oriented architecture (SOA), developers and users can implement a complex application to use the services which are supplied by many providers. Services are platform and network independent operations that clients or other services invoke. Remote ProcedureCall(RPC), Component Object Model(COM), and Common Object Request Broker Architecture(CORBA) etc. have been used in constructing and composing services, but they are notenough to offer three native capabilities for SOA: description, discovery, and communication. After appearance of Web services, which provides platform independence and a well standardized machine readable format, service providers can use Web Services Description Language (WSDL) for description, Universal Description, Discovery, and Integration (UDDI) for discovery and Simple Object Access Protocol (SOAP) for communication [1]. In this new model, Web service composition, the process of assembling composite Web services from collections of individual, interoperating Web services, has recently gained much attention to support business-to-business or enterprise application integration in SOA [2].

As Web services composition is accepted as a

* 정 회 원 : 중앙대학교 컴퓨터공학과
kang@archi.cse.cau.ac.kr
** 종신회원 : 대구한의대학교 모바일컨텐츠학부 교수
phdjoon@dhu.ac.kr(교신저자)
[2006/07/19 투고 - 2006/07/31 심사 - 2006/09/27 심사완료]

business solution to enterprise application integration, the quality of service of the composite Web service is emerging as crucial criteria on a SOA framework. And many researchers have achieved the planning scheme of guaranteeing a composite Web service. Liangzhao et al. [3, 4] and Danilo et al. [5] showed that optimizing the QoS of a given Web composite service is NP-hard problem. Liangzhao et al. [3, 4] proposed local optimization approach of selecting each Web service to achieve the given priority optimally and global optimization approach of selecting Web services to achieve the given global QoS. In case of composing the composite Web service for the given QoS is the Multiple choice Multiple dimension Knapsack Problem(MMKP). Liangzhao et al. solved this problem by using linear programming under the assumption that QoS values are linear. In this paper, we propose K-way Supernode approach to get Web services sets to guarantee user's global QoS.

The rest of this paper is organized as follows. In Section 2 we present some related work and in Section 3 we describe the K-way Super node approach. Section 4 presents experiments and comparison between local optimization and global optimization. Finally, Section 5 draws some conclusion.

2. Related Work

In this section, we review on the QoS of Web service composition. And our research starts from the work presented in [3, 4, 5]. For simplicity, we introduce their works briefly.

2.1 The QoS of a composite Web service

According to the W3C a Web serviceis a software system designed to support interoperable machine-to-machine interaction over a network. Because of the suitability as a service, the number of SOA framework supporting Web services is increasing. Providers register their Web services on a UDDI server, and users make a complex application by composing them. As composite Web service usages are increasing, the QoS of it - cost, response time, reliability etc. - is also becoming an issue. Even though the dynamic and unpredictable nature of the Web made a composite Web service guarantee the QoS given by user, the extension of Web service and SOA framework overcomes it.

Reliable HTTP(HTTPR), Blocks Extensible Exchange Protocol(BEEP), Direct Internet Message Encapsulation(DIME) etc. come out to exchange the unreliable HTTP. WSDL and Web service composing language - IBM's Web Service Flow Language (WSFL) and BEA Systems'Web Services Choreography Interface (WSCI), Business Process Execution Language for Web Services (BPEL4WS), Petri net etc. are designed to describe the functionality of a Web composite service and very lake of describing non-functional properties such as semantic or quality of service (QoS). W3C is standardizing a language to describing non-functional properties.In [6, 7], the enhanced UDDI registry, which provide information about quality parameters on the provider side, was proposed. Planning a composite Web service for guaranteeing user's QoS is also processed by many researchers. Liangzhao et al. [3, 4] proposed linear programming approach to solve the global QoS problem. Michael et al. [8] proposed QoS Aggregation using workflow patterns for efficient composite Web service.

2.2 Service Selection by Local Optimization and Global Optimization

Bellow is the formalization of the QoS optimization in the composing Web service and is summarized from [3, 4,5]. In the work presented in [3, 4, 5] two main approaches have been proposed: local and global optimization.

Registered Web services set in UDDI(RW) : Many Web service providers supply Web services, registered in UDDI, with different QoS.

$$RW = \{WS_1, WS_2, WS_3, \Lambda\ \Lambda\ \Lambda\ , WS_n\}$$
$$WS_i = \{ws_{i1}, ws_{i2}, ws_{i3}, \Lambda\ \Lambda\ \Lambda\ , ws_{im}\}$$

$ws_{ij}, ws_{ik} \in WS_i, j \neq k$ have the same functionality but different quality of service.

A composite service is specified as a collection of generic service tasks and it is possible to represent as a Directed Acyclic Graph (DAG). If it contains cycles, an unfolding technique makes it acyclic. [5]

Composite Service (CS): $CS = (T, E)$. Each node $t \in T$ is a task in a composite service, and is in one-to-one correspondence with a Web service. There is an edge $(x, y) \in E$, if the output of a task x is connected to an input of a task y.

Execution path (ep): An execution path of a composite service is a sequence of task. $ep = \{t_1, t_2, t_3, \Lambda\ \Lambda\ , t_n\}$, such that t_1 is the initial task, t_n is the final task. And for every task no t_i, t_j belong to alternative branches.

Sub path (sp): A sub path of an execution path ep_l is a sequence of tasks $\{t_1, t_2, t_3, \Lambda\ \Lambda\ , t_n\}$, $t_i \in ep_l : \forall i$, from the begin to the end task which does not contain any parallel sequence.

Execution plan ($exep$): An execution plan of an execution path $exep_l$ is a set of ordered pairs $< t, ws >$ $t \in ep, ws \in WS$. For each pair $< t, ws >$ in $exep$, Web service ws provides the operation required by task t.

QoS vector (Q) : It consist of quality values of a Web service. The criteria of QoS vector depends on the system model.

$$Q(ws_{ij}) = \{Q_{price}(ws_{ij}), Q_{response}(ws_{ij}), \Lambda\ , Q_t(ws_{ij})\}$$

In local optimization, when a task needs to be executed, the quality score of each of the Web services that can execute this task is calculated based on weight value of each QoS criteria. The weight value is assigned by user. After calculating quality score, the maximal score Web service which satisfies all the user constraints for that task is selected. If no service satisfies the user constraint for a given task, an exception is raised. Scoring a Web service is described bellow.

Before calculating fitness, scaling phase is necessary because some of the criteria such as execution time and execution price is negative, i.e. the higher the value, the lower the quality.

$V_k(ws_{ij})$ is the scaled value for criteria k of ws_{ij} at QoS vector.

For negative criteria,

If $Q_k^{\max}(W_i) - Q_k^{\min}(W_i) \neq 0$ $\quad V_k(ws_{ij}) = \dfrac{Q_k^{\max}(W_i) - Q_k(ws_{ij})}{Q_k^{\max}(W_i) - Q_k^{\min}(W_i)}$ else $V_{i,j} = 1$

For positive criteria,

If $Q_k^{\max}(W_i) - Q_k^{\min}(W_i) \neq 0$ $\quad V_k(ws_{ij}) = \dfrac{Q_k(ws_{ij}) - Q_k^{\min}(W_i)}{Q_k^{\min}(W_i) - Q_k^{\max}(W_i)}$ else $V_{i,j} = 1$

$Score(w_{ij}) = \sum_k (V_k(w_{ij}) * W_k)$, where W_k is the weight of criterion k.

In the local optimization approach, service selection is done for each task individually and optimized locally. But it can't guarantee to satisfy the global constraints which are given by user. In Web service environments, services composition could be hidden to the end user, therefore the global constraint is more important to the end user. When the global constrains are given, selecting Web services is described as below.

$$Maximize \sum_{l=1}^{t}(\sum_{i=1}^{n}\sum_{j=1}^{m} X_{l,ij} \cdot Score(w_{l,ij}))$$

Subject to: $\sum_{l=1}^{t}(\sum_{i=1}^{n}\sum_{j=1}^{m} X_{l,ij} \cdot Q_k(w_{l,ij}))$ satisfies the given constraint QC_k for all k.

$\sum_{j=1}^{m} X_{ij} = 1; \forall j$ and $X_{ij} \in \{0,1\}$

Where l is the task number in an execution plan and j is the size of Web service set i. QC is a QoS constraint given by user. It is same with Multiple choice Multiple dimension Knapsack Problem (MMKP). And it is NP-hard problem.

3. K-way Super Node Approach

In the pervious section, composing Web services to guarantee given global constraints is NP-hard problem. Liangzhao et al.[3, 4] solved this using a linear programming approach under assumption that Web services' QoS value is linear. In this paper, we propose K-way Super node approach that improves the local optimized composite Web service to guarantee user's global constraints.

Definition 1. (K-way Super node). A K-way Super node (SN) is a group of k tasks $\{t_1, t_2, t_3, \Lambda\Lambda, t_k\} : 1 \le k$ in a sub path, such that every task $t_i (1 < i < k)$ is a direct successor.

Definition 2. (Shortage). Let sp is a sub execution path $\{ws_1, ws_2, ws_3, \Lambda\Lambda, ws_n\}$ corresponding to a sub path $\{t_1, t_2, t_3, \Lambda\Lambda, t_n\}$ and CQ_k is a given QoS constraint of k. The path shortage $S_k(sp)$ is defined as $S_k(sp) = CQ_k - \sum_{ws_i \in sp} Q_k(ws_i)$.

Definition 3. (Satisfied). A sub path sp is *satisfied* if and only if its shortage is positive value for all QoSes.

K-way Super node approach calculates the shortage of each sub path for an execution plan, constructs K-way Super node DAG for each sub path, and finds a set of Web services which guarantees user's QoS by freely exchanging Web services in a Super node (sn) with candidate Web services. Optimizing a Super node is defined as below.

$$Maximize \sum_{l=1}^{t}(\sum_{i=1}^{n}\sum_{j=1}^{m} X_{l,ij} \cdot Score(w_{l,ij})),$$

$t = Size(sn)$

Subject to: $S_k(sn) \ge 0 : \forall k$

$\sum_{j=1}^{m} X_{ij} = 1; \forall j$ and $X_{ij} \in \{0,1\}$

Where l is the task number in a Super node and j is the size of Web service set i. QC is a QoS constraint given by user.

Example: Table 1 shows the registered Web services in a UDDI. Figure 1 shows a composite Web service. And the QoS weight for response time is 0.5 and Cost is also 0.5.

Table 1. Registered Web services in a UDDI (sec/dollar)

	Provider 1		Provider 2		Provider 3	
Web service	Response Time	Cost	Response Time	Cost	Response Time	Cost
A	2.0	0.7	0.5	0.6	0.8	0.5
B	1.2	0.8	0.9	0.6	0.8	0.7
C	2.5	1.7	2.2	2	3.2	2.1

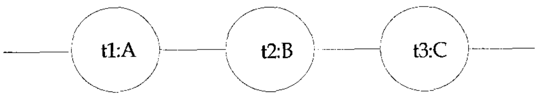


Figure 1. A composite Web service

Local optimization approach [3, 4] assigns Provider 3's A type Web service to task 1, Provider 2's B type to task 2, and Provider 1's C type to task 3. Total response time is 4.2 and total cost is 2.8 $. If response time and cost are given 3.8 sec and 3.2$ as the global constraint respectively, the local optimization can not guarantee the given constrains. To construct K-way Super node, the shortage value should be calculated (Figure 2). Figure 3 shows the K-way Super node DAG for the Figure 2 (here, K = 2).

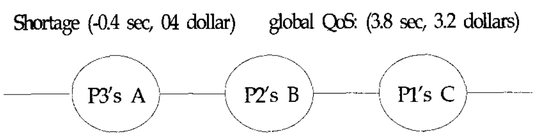


Figure 2. The shortage graph

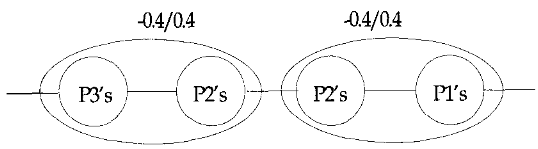
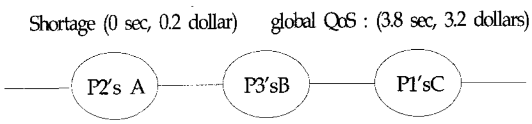


Figure 3. 2-way Super node DAG

Figure 3 shows that for making the composite Web service *satisfied*, response time should be decreased at 0.4 sec but it has 0.4 dollar cost margin. The goal is that the composite Web service is *satisfiedby* exchanging Web services in a Super node. In the first Super node, provider 2's Type A and provider 3's type B can make response time shortage zero in 0.4 $. And in the second Super node, provider 2's Type B and provider 3's type C can make it *satisfied*. But the first Super node's selection makes the higher composite service score. Figure 4 shows the *satisfied* composite Web service.



Firgure 4. a satisfied composite Web service

If an execution plan has a parallel execution path and it violates user's QoS constraint, it is necessary to be split into a sub path. Figure 5 shows an AND parallel execution which is optimized locally. (An OR parallel execution is not considered into, because it can not be contained in an execution path. Each path in an OR parallel execution becomes an execution path.)

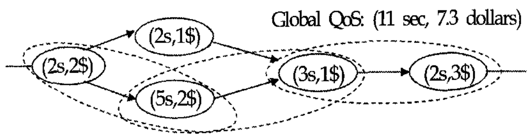


Figure 5. Response time centric 2-way Super node DAG

There are two kinds of QoS criteria; the one is commonly applying for each pathlike response time and the other is applying for total execution path like execution budget. For applying K-way Super node approach to an And parallel execution path, an additional policy is necessary. Common QoS criteria to a sub path are optimized first, and then remaining QoS criteria are optimized for total execution path under not changing the sum of common QoS criteria in a Super node; each sub paths in local optimized execution plan is optimized to response time by using K-way Super node. (Figure 5.) If any of sub paths violates response time, an exception is raised. After optimizing to response time, cost optimizing is applied to total execution path by using K-way Super node. (Figure 6.)

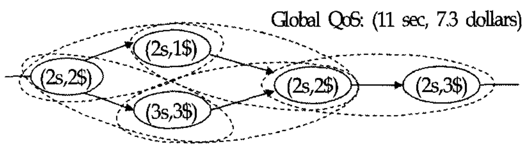


Figure 6. Cost centric 2-way Super node DAG

Algorithm: Optimizing a composite service by using K-way Super node

Input: A composite task service, a set of registered Web services, user's global constraints.

Output: A composite Web service which guarantee user's global constraints or an exception.

1: Calculate the local optimized composite service for input composite task service

2: Construct the transitive closure graph of the local optimized composite service
   Get the path information and execution plans

3: For all execution plans, do 4

4: If an execution plan has no parallel execution.
   Construct K-way Super node DAG
   Optimize Super nodes
   Else
   Construct K-way Super node DAG from every sub path for common QoS criteria.
   Optimize Super node in the point of common QoS criteria
   Construct K-way Super node DAG from total execution path for the remaining QoS criteria
   Optimize Super node in the point of the remaining QoS criteria

5: If all execution plans become *satisfied*, return a composite Web service
   Else an exception is raised, and ask user to loose the QoS constraints.

4. Experiments

In this section, we conduct experiments to illustrate the behavior of the K-way Super node scheme. We also compare its performance with local optimization and global optimization.

In our simulations, we consider 5 composite services with 15 tasks and an enhanced UDDI registry which has 10 operation types registered by 5 providers. Response time and execution cost are considered as QoS. QoS values have been random generated - response time gets from 2 to 5 and execution cost gets from 0.5 to 2.5. For measuring the performance of K-way Super node, we input the global optimized QoS as user's constraints.

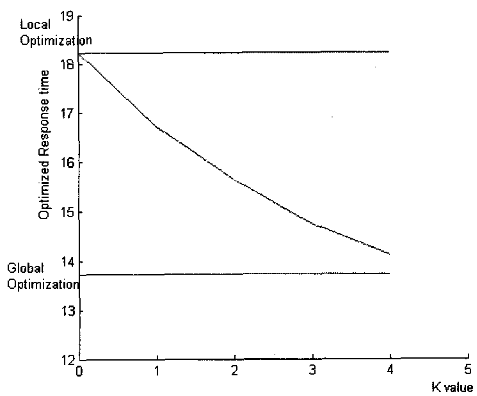


Figure 7. Effect of K value in optimization of response time

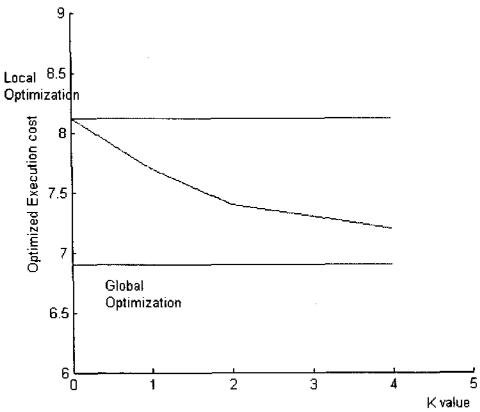


Figure 8. Effect of K value in optimization of execution cost

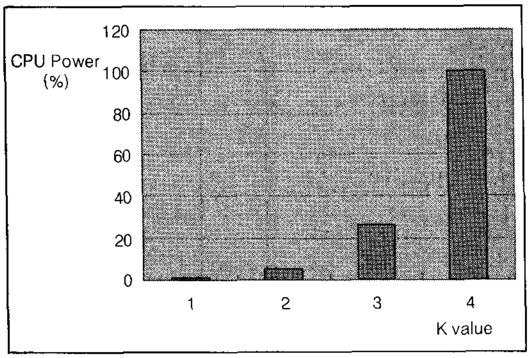


Figure 9. Effect of K value in performance

K value determines the size of Super node. The size of Super node affects the degree of optimization of QoS, but it incurs the degradation of performance. (Figure 7, 8, 9)

5. Conclusion

Optimizing a composite Web service for user's QoS constraints is very important operation in SOA. In this study, we proposed K-way Super node approach to improve a local optimized composite Web service and presented an optimizing algorithm using K-way Super node. Experimental results showed the effect of K value. We are currently working on a scheme to find an effective K value to a given composite Web service.

References

[1] Nikola Milanovic, Miroslaw Malek "Current Solutions for Web Service Composition." IEEE Internet Computing 8(6): 51-59 (2004)

[2] Maja Vukovic, Peter Robinson "GoalMorph: Partial Goal Satisfaction for Flexible Service Composition" International Journal of Web Services Practices, Vol. 1, No.1-2(2005), pp. 40-56

[3] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, Quan Z. Sheng "Quality driven web services composition" WWW 2003: 411-421

[4] Liangzhao Zeng, Boualem Benatallah, Anne H. H. Ngu, Marlon Dumas, Jayant Kalagnanam, Henry Chang "QoS-Aware Middleware for Web Services Composition" IEEE Trans. Software Eng. 30(5): 311-327 (2004)

[5] Danilo Ardagna, Barbara Pernici "Global and Local QoS Guarantee in Web Service Selection" Business Process Management Workshops 2005: 32-46

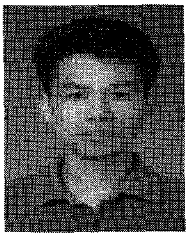
[6] S. Modafferi, E. Mussi, A. Maurino B. Pernici, "A framework for complex e-services provisioning"Proc. of IEEE International Conference on Services Computing(SCC 2004), Shanghai, China 81-90 , 2004

[7] Devis Bianchini, Valeria De Antonellis, Michele Melchiori "An Ontology-Based Architecture for Service Discovery and Advice System" DEXA Workshops 2005: 551-556

[8] Michael C. Jaeger, Gregor Rojec-Goldmann, Gero Mühl "QoS Aggregation for Web Service

Composition using Workflow Patterns" EDOC 2004: 149-159
[9] Min Tian, A. Gramm, Hartmut Ritter, Jochen H. Schiller "Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework" Web Intelligence 2004: 152-158
[10] Mohamed Adel Serhani, Rachida Dssouli, Abdelhakim Hafid, Houari A. Sahraoui "A QoS Broker Based Architecture for Efficient Web Services Selection" ICWS 2005: 113-120
[11] Daniel A. Menasce "Composing Web Services : A QoS View" IEEE Internet Computing, Volume 8, 2004, pp88-90
[12] Daniel A. Menasce "QoS Issues in Web Services" IEEE Internet Computing, vol. 6, no. 6, 2002, pp. 72 75

◐ 저 자 소 개 ◑



강 남 오(NamOh Kang)
1997년 중앙대학교 컴퓨터공학과 졸업(학사)
2000년 중앙대학교 대학원 컴퓨터공학과 졸업(석사)
2006년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
2006~현재 중앙대학교 CT 연구원 전임 연구원
관심분야 : 멀티미디어 처리, 웹 서비스, 시맨틱 기술, etc
E-mail : namohkang@gmail.com

박 사 준(Park, Sajoon)
1990년 중앙대학교 전자계산학과(학사)
1994년 중앙대학교 대학원 컴퓨터공학과(석사)
2004년 중앙대학교 대학원 컴퓨터공학과(박사)
2005~ 현재 대구한의대학교 모바일콘텐츠학부 교수
관심분야: 인공지능, 시맨틱 웹, 유비쿼터스, 모바일 콘텐츠, etc
E-mail : phdjoon@dhu.ac.kr