

## 워크플로우 분할 기반의 빠른 복합 웹 서비스의 선택 방법

장재호<sup>0</sup>, 신동훈, 이경호  
연세대학교 컴퓨터학과

{jjang<sup>0</sup>, dhshin}@icl.yonsei.ac.kr khlee@cs.yonsei.ac.kr

### Fast Composite Web Services Selection Based on Workflow Partition

Jae-Ho, Jang<sup>0</sup>, Dong-Hoon Shin, Kyong-Ho Lee  
Dept. of Computer Science, Yonsei University

#### 요약

본 논문은 워크플로우 분할에 기반하여 복합 웹 서비스를 빠르게 선택하는 방법을 제안한다. 제안된 방법은 QoS 요구사항의 제약정도 계산, 워크플로우 분할, QoS 요구사항 분해, 그리고 서비스 선택의 네 단계로 구성된다. 특히 QoS 요구사항의 분해과정이 휴리스틱에 기반하기 때문에 워크플로우 분할 시 QoS 요구사항을 만족하는 서비스를 찾지 못할 가능성이 있다. 제안된 방법은 분할에 따른 서비스 선택 실패를 줄이기 위해 주어진 QoS 요구사항의 제약정도를 계산하여 워크플로우의 분할 여부를 판단한다. 실험 결과 제안된 방법은 분할 속도, 성공률, 품질 면에서 우수한 성능을 보였다.

#### 1. 서론

웹 서비스는 HTTP와 SOAP과 같은 표준 프로토콜을 기반으로 접근 가능한 소프트웨어 컴포넌트로서 이종 플랫폼 간의 상호운용성을 지원한다. 최근 고수준의 비즈니스 프로세스에 대한 요구가 증가함에 따라 단일 웹 서비스를 조합하여 복합 웹 서비스(Composite Web services)를 구성하는 방법에 대한 관심이 높아지고 있다. 일반적으로 복합 웹 서비스의 비즈니스 로직은 추상 워크플로우(abstract workflow)로 기술된다. 복합 웹 서비스 선택은 추상 워크플로우의 각 태스크에 웹 서비스를 바인딩하여 실행 가능한 워크플로우를 구성하는 과정이다. 한편 동일한 기능을 제공하지만 QoS는 서로 다른 다수의 후보 서비스들이 존재하기 때문에 서비스 선택을 통해 사용자가 요구하는 특정 QoS를 만족하는 복합 웹 서비스를 구성할 수 있다. <그림 1>은 QoS를 고려한 복합 웹 서비스의 선택 과정을 보여준다.

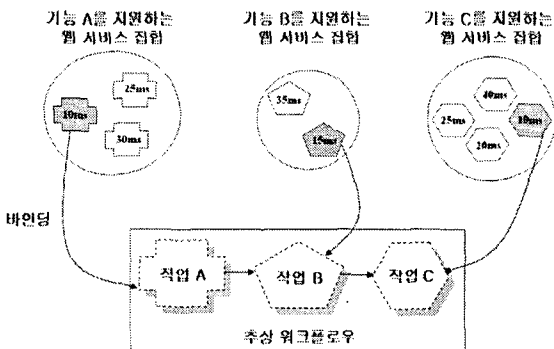


그림 1. QoS를 고려한 복합 웹 서비스의 선택 과정.

복합 웹 서비스 선택 문제는 모든 후보 서비스 집합의 조합을 고려해야 한다. 웹 서비스의 급격한 증가와 QoS가 동적으로 변하는 웹 서비스 환경을 고려할 때 주어진 QoS 요구사항을 만족하는 복합 웹 서비스의 빠르게 선택할 수 있는 방법이

필요하다. 본 논문은 워크플로우를 분할하여 복합 웹 서비스를 빠르게 선택할 수 있는 방법을 제안한다. 제안된 방법은 추상 워크플로우를 두 개의 서브 워크플로우로 분할한 후, 전체 워크플로우에 대한 QoS 요구사항이 만족될 수 있도록 QoS 요구사항을 분해하여 서브 워크플로우의 요구사항을 생성한다.

제안된 방법의 성능을 평가하기 위해 선택 속도와 선택된 서비스의 품질을 기존의 최적 서비스 선택 방법과 비교하였다. 실험 결과, 제안된 방법이 항상 빠르게 서비스를 선택하였으며, 이때 최적 서비스와의 품질 차이는 5% 미만이었다. 특히, 제안된 분할 방법은 99%이상의 성공률을 보였다.

#### 2. 관련 연구

복합 웹 서비스에 대한 관심이 급증함에 따라 복합 웹 서비스의 선택에 관한 다수의 연구 결과가 발표되었다. <표 1>은 QoS 기반 복합 웹 서비스 선택 관련 연구의 특징을 요약한다.

표 1. QoS 기반 복합 웹 서비스 선택에 관한 연구

관련논문	특징
[1]	서비스 선택 시 서비스 간 충돌을 고려
[2]	Multi-Object Programming 에 기반하여 선택
[3]	유전자 알고리즘을 이용하여 서비스를 선택
[4]	Transiently chaotic neural network를 이용하여 복합 웹 서비스를 선택
[5]	제약조건을 처리하는 적합도 함수를 사용하는 유전자 알고리즘에 기반하여 서비스를 선택
[6]	컴포지션 패턴을 기반으로 서비스를 선택
[7]	지연시간에 대한 제약조건을 가지는 서비스 선택 문제의 효율적인 해결 방법을 제안
[8]	워크플로우의 QoS에 대한 수학적 모델을 제안
[9]	integer linear programming 기반으로 복합 웹 서비스를 선택하는 미들웨어를 제안

위에 언급된 관련 연구의 대부분은 최적 QoS를 가지는 서비스의 선택에 중점을 두고 있으며 선택 속도의 개선을 위한 노력은 결여되어 있다. 한편 제안된 분할 방법은 선택 과정과 독립적으로 이루어지기 때문에 어떠한 선택 방법에도 적용가능하다는 장점을 갖는다.

※ 이 연구는 정보통신부(정보통신연구진흥원)에서 지원하는 2005년도 IT기초기술연구지원사업의 연구결과임.

3. 제안된 복합 웹 서비스 선택 방법

제안된 방법은 <그림 2>와 같이 QoS 요구사항의 제약정도 계산, 워크플로우 분할, QoS 요구사항 분해, 그리고 서비스 선택의 네 단계로 구성된다. QoS 요구사항의 분해과정이 휴리스틱에 기반하기 때문에 워크플로우 분할 시 QoS 요구사항을 만족하는 서비스를 찾기 못할 가능성이 있다. 제안된 방법은 분할에 따른 서비스 선택 실패를 줄이기 위해 주어진 QoS 요구사항의 제약정도를 계산하여 워크플로우의 분할 여부를 판단한다.

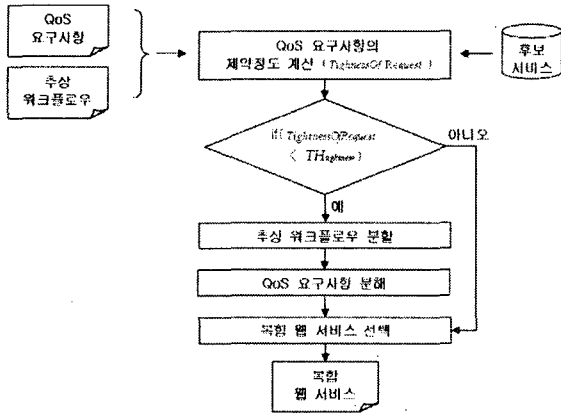


그림 2. 제안된 복합 웹 서비스 선택 방법.

3.1 QoS 요구사항의 제약정도 계산

QoS 요구사항의 제약정도는 후보 서비스들로 구성 가능한 최고 품질과 주어진 요구사항의 차이로 계산된다. 따라서 높은 제약 정도를 가지는 요구사항일수록 이를 만족하는 복합 웹 서비스를 구성 가능할 확률은 줄어들게 된다. QoS 요구사항을 구성하는 개별 품질 요구에 대한 제약 정도는 식 (1)을 이용하여 계산된다. 이를 기반으로 식 (2)를 이용하여 전체 요구사항에 대한 제약정도를 계산한다.

$$TightnessOfConstraint(q) = \frac{Constraint(q) - WorstQoS(q)}{BestQoS(q) - WorstQoS(q)} \quad (1)$$

$$TightnessOfRequirement = \frac{\sum_{i=1}^n Tightness_i}{\sum_{i=1}^n \frac{1}{i^2}} \quad (2)$$

단,  $Constraint(q)$ ,  $BestQoS(q)$ , 그리고  $WorstQoS(q)$ 는 각각 품질 요소  $q$ 에 대한 QoS 요구사항, 최고 및 최저의 품질이며  $Tightness_i$ 는  $TightnessOfConstraint(q)$  중  $i$ 번째로 큰 값이다. 계산된  $TightnessOfRequirement$ 가 임계값  $TH_{tightness}$ 보다 작은 경우에 한해 워크플로우 분할을 수행한다.  $TH_{tightness}$ 는 과거 실패하였던 요구사항들의 제약사항을 이용하여 설정한다.

3.2 추상 워크플로우 분할

제안된 방법은 순차, 병렬, 조건, 루프 구조로 구성된 워크플로우를 대상으로 한다. 단, 루프의 경우, 해당 루프의 최대 수행 횟수만큼 루프 구간을 복제하여 제거한다. 우선 분할 및 QoS 분해 과정을 단순화하기 위해 RSG(reduced sequential graph)를 생성한다. RSG는 추상 워크플로우에 속한 병렬 및 조건 구조를 순차구조로 환원한 그래프이다. 초기 RSG는 추상

워크플로우의 DAG (directed acyclic graph) 표현과 동일하며 각 정점(vertex)은 대응되는 태스크의 후보 서비스 개수를 저장한다. 초기 RSG는 환원 규칙을 적용하여 최종 RSG로 변환된다. <그림 3>은 RSG 생성의 예를 보여준다.

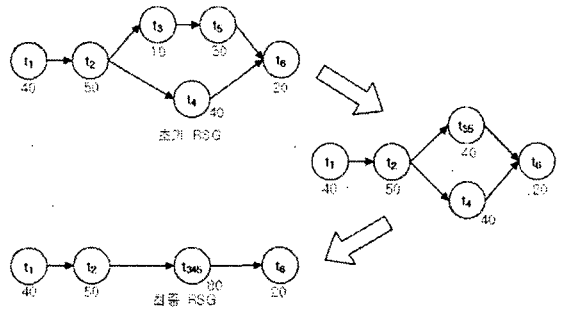


그림 3. RSG 생성의 예.

최종 RSG가 완성되면 RSG의 정점을 원소로 갖는 집합을 생성한 뒤, 근사 부분 합 (approximate subset sum) 알고리즘을 이용하여 전체 후보 서비스의 절반에 근사한 크기를 가지는 두 부분 집합으로 분할한다. 생성된 각 부분 집합의 원소를 순차적으로 연결하여 분할된 RSG를 생성한다. 최종적으로, 분할된 RSG의 각 정점을 대응하는 추상 워크플로우의 구조로 치환하여 분할된 서브 워크플로우를 생성한다.

3.3 QoS 요구사항 분해

서브 워크플로우로부터 선택된 서비스는 주어진 QoS 요구사항을 만족하여야 한다. 이를 위해, 전체 워크플로우의 QoS 요구사항을 분해하여 서브 워크플로우의 QoS 요구사항을 생성한다. 제안된 방법은 QoS를 시간, 비용, 확률, 용량의 네 가지 범주로 분류한다. 각 범주에 대한 QoS 요구사항의 분해는 식 (3)과 식(4)를 이용한다.

$$\begin{cases} Constraint'(q) = Constraint''(q) & \text{if } q \in \text{시간, 비용} \\ Constraint'(q) \times Constraint''(q) & \text{if } q \in \text{확률} \\ Constraint'(q) = Constraint''(q) & \text{if } q \in \text{용량} \end{cases} \quad (3)$$

$$\begin{cases} Constraint'(q) = Constraint(q) \times \frac{BestQoS'(q)}{BestQoS(q)} & \text{if } q \in \text{시간, 비용} \\ Constraint'(q) \times \frac{BestQoS'(q)}{BestQoS''(q)} & \text{if } q \in \text{확률} \end{cases} \quad (4)$$

단,  $Constraint'(q)$  (또는  $Constraint''(q)$ )는 품질 요소  $q$ 에 대한 서브 워크플로우의 QoS 요구사항이다.

3.4 서비스 선택

제안된 방법은 효율적인 서비스 선택을 위해 Mixed integer linear programming (MILP)를 사용한다. 복합 웹 서비스의 QoS는 목적함수를 통해 최적화되며 QoS 요구사항은 MILP 제약사항을 이용하여 표현한다. 분할된 워크플로우의 경우 두 문제가 모두 성공하지 않으면 전체 서비스 선택은 실패하게 된다. 한편, 먼저 시도된 문제가 해결될 경우 남은 문제의 QoS 요구사항은 식 (3)을 이용하여 다시 계산한다. 이때  $Constraint'(q)$ 는 앞서 선택된 복합 웹 서비스의 실제 QoS를 사용한다. 서비스 선택에 실패한 경우, 해당 QoS 요구사항의 제약정도를 로그에 기록한 뒤 임계값  $TH_{tightness}$ 를 갱신한다.

임계값  $TH_{tightness}$  는 과거 로그를 기반으로  $\epsilon$  ( $0 \leq \epsilon \leq 1$ )의 실패를 허용하도록 설정된다. 예를 들어, 100개의 로그 데이터가 존재하며  $\epsilon = 0.05$ 일 때, 임계값은 로그 데이터 중 5번째로 작은 제약정도가 된다.

4. 실험 및 결과분석

제한된 방법의 성능을 평가하기 위해 선택 속도와 선택된 서비스의 품질을 기존의 최적 서비스 선택 방법과 비교하였다. 실험은 <표 2>와 같이 고정된 크기의 후보 서비스 집합에 대하여 워크플로우의 태스크 수를 증가시키는 경우와 고정된 태스크 개수를 가지는 워크플로우에 대하여 후보 서비스 집합의 크기를 증가시키는 두 가지 경우에 대하여 수행되었다. 실험을 위해 2, 3, 그리고 4개의 품질에 대한 제약사항을 가지는 QoS 요구사항을 무작위로 생성하였으며 각 실험 당 5000개의 요구사항을 생성하여 실험의 일반성을 높이고자 하였다. 실험에 사용된  $TH_{partition}$  과  $\epsilon$ 은 각각 0.3과 0.05였으며, 각 품질 요소에 대한 가중치는 동일하게 설정하였다. 실험은 Intel Pentium™4 2.4GHz, 1GB RAM에서 수행되었다.

표 2. 실험에 사용된 워크플로우의 특징

	워크플로우 구조	태스크 수	후보 서비스 수
실험1	순차	단계별로 증가	변화 없음
실험2	순차, 병렬, 조건	변화 없음	단계별로 증가

<그림 4>과 <그림 5>는 각각 2개 제약사항을 가지는 QoS 요구사항에 대한 실험1과 실험2의 평균 서비스 선택 속도를 보여준다. 3개 및 4개 제약사항의 가지는 경우도 2개 제약사항의 경우와 유사한 결과를 보였다.

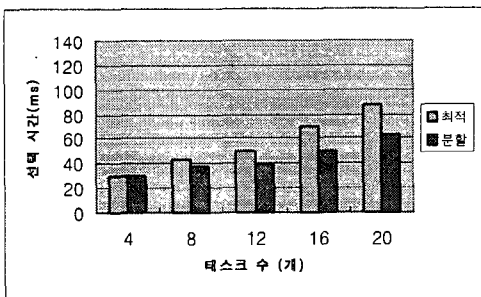


그림 4. 실험 1의 서비스 선택 속도. (2개 제약사항)

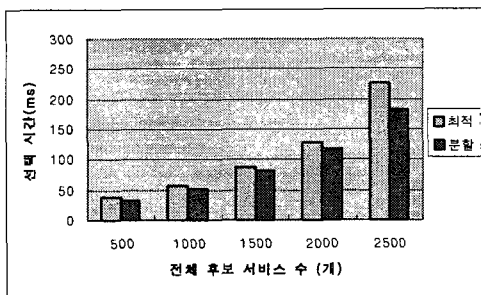


그림 5. 실험 2의 서비스 선택 속도. (2개 제약사항)

복합 웹 서비스의 품질을 최적 서비스 선택 결과와 비교한 결과 5% 미만의 미비한 차이를 보였다. 제안된 분할 방법은 99% 이상의 성공률을 보였다.

5. 결론

단일 웹 서비스로 제공할 수 없는 복잡한 비즈니스 프로세스에 대한 요구가 점차 증가함에 따라 사용자가 원하는 QoS를 만족하는 복합 웹 서비스를 선택하는 것이 중요하다. 한편 웹 서비스의 증가에 따른 선택 속도의 저하와 QoS가 동적으로 변하는 웹 서비스 환경을 고려할 때 주어진 QoS 요구사항을 만족하면서 적절한 QoS의 복합 웹 서비스를 빠르게 선택할 수 있는 방법이 요구된다.

본 논문은 워크플로우를 분할하여 복합 웹 서비스를 빠르게 선택할 수 있는 방법을 제안한다. 특히, 분할에 의한 서비스의 선택 실패를 줄이기 위해 주어진 QoS 요구사항의 제약정도를 계산하여 제약정도가 적절한 경우에 한해 선택적으로 분할한다. 한편, 실험을 통해 제안된 서비스 선택 방법의 성공률, 선택 속도, 그리고 복합 웹 서비스의 품질을 평가하였다. 실험 결과, 제안된 방법은 99% 이상의 성공률을 보였으며 분할을 적용하지 않았을 경우에 비해 언제나 빠른 선택 속도를 나타내었다. 결과 서비스의 QoS 측면에 있어서도 5% 미만의 차이를 보였다. 향후 더욱 정교한 제약정도 계산법을 개발하고, 후보 웹 서비스의 QoS가 동적으로 변하는 환경에 대한 추가 실험을 수행할 계획이다.

6. 참고문헌

- [1] A. Gao, D. Yang, S. Tang, and M. Zhang, "QoS-Driven Web Service Composition with Inter Service Conflicts," Proc. 8th APWeb Conf. (LNCS 3841), pp. 121 - 132, 2006.
- [2] Y. Chen, Z. Li, Q. Jin, and C. Wang, "Study on QoS Driven Web Services Composition," Proc. 8th APWeb Conf. (LNCS 3841), pp. 702 - 707, 2006.
- [3] S. Liu, Y. Liu, N. Jing, G. Tang, and Y. Tang, "A Dynamic Web Service Selection Strategy with QoS Global Optimization Based on Multi-objective Genetic Algorithm," Proc. 4th Int'l Conf on Grid and Cooperative Computing (LNCS 3795), pp. 84-89, 2005.
- [4] L. Yang, Y. Dai, B. Zhang, and Y. Gao, "Dynamic Selection of Composite Web Services Based on a New Structured TCNN," Proc. IEEE Int'l Workshop on Service-Oriented System Engineering, pp. 149 - 158, 2005.
- [5] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An Approach for QoS-aware Service Composition based on Genetic Algorithms," Proc. Conf. Genetic and Evolutionary Computation, pp. 1069-1075, 2005.
- [6] R. Grønmo and M. C. Jaeger, "Model-Driven Methodology for Building QoS-Optimized Web Service Compositions," Proc. 5th Int'l Conf. on DAIS (LNCS 3543), pp. 68-82, 2005.
- [7] T. Yu and K. Lin, "Service selection algorithms for web services with end-to-end QoS constraints," Proc. IEEE Int'l Conf. e-Commerce Technology, pp. 129-136, 2004.
- [8] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut, "Quality of Service for Workflows and Web Service Processes," Web Semantics Journal, Vol. 1, No. 3, pp. 281-308, 2004.
- [9] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition," IEEE Trans. on Software Engineering, Vol. 30, No. 5, pp. 311 - 327, 2004.