

## 워크플로우 분할을 이용한 복합 웹 서비스의 빠른 선택

장재호<sup>0</sup>, 이경호

연세대학교 컴퓨터과학과

jjang<sup>0</sup>@icl.yonsei.ac.kr khlee@cs.yonsei.ac.kr

### Fast Selection of Composite Web Services using Workflow Partitioning

Jae-Ho Jang<sup>0</sup>, Kyong-Ho Lee

Dept. of Computer Science, Yonsei University

#### 요 약

웹 서비스는 서비스 지향 컴퓨팅 환경으로의 변화를 주도하는 기술로써 많은 각광을 받고 있다. 본 논문에서는 사용자가 요구하는 품질의 서비스를 선택할 수 있는 방법을 제안한다. 제안된 방법은 워크플로우를 분할하는 휴리스틱에 기반하여 기존의 LP(Linear Programming) 기반의 서비스 선택 알고리즘의 수행시간을 단축한다.

#### 1. 서 론

웹 서비스(Web Services)는 HTTP와 SOAP과 같은 표준 프로토콜을 사용하여 접근 가능한 소프트웨어 컴포넌트이다. 기존의 웹 인프라를 그대로 이용한다는 점과 XML에 기반하고 있다는 점은 상호 운용성, 재사용성, 구축비용 등에 있어서 큰 이점을 제공할 수 있다. 현재의 웹 서비스 표준 기술인 WSDL 및 UDDI는 기능성에 따른 웹 서비스의 탐색을 지원한다. 하지만 사용자는 자신이 원하는 기능적인 요구사항뿐만 아니라 특정한 처리량을 보장하거나, 비용이 적은 서비스를 찾고자 할 수도 있다. 서비스 품질(Quality of Service)은 지연시간, 비용, 가용성, 신뢰성 등과 같은 특정 서비스가 가지는 비기능적 특성이며 개방된 비즈니스 환경에서는 필수적으로 고려되어야만 한다. <그림 1>은 웹 서비스의 기술 단계부터 실행 및 종료 단계에서 서비스의 품질을 지원하는 과정을 나타낸다. 특히 음영 처리된 부분은 서비스 품질의 도입을 위해 추가된 단계이다.

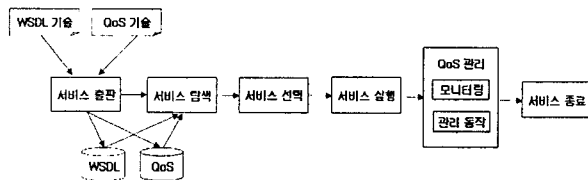


그림 1. 서비스 품질을 지원하는 웹 서비스의 처리 과정.

그동안 복합 웹 서비스의 서비스 선택에 관하여 빠른 수행시간과 최적의 품질 결과를 얻기 위한 연구가 계속적으로 이루어져 왔다. 그러나 실제 환경에 적용 가능한 수준의 방법에 대한 연구 결과는 미흡한 상태이다.

기존의 연구중에서 LP를 이용한 방법은 순차적 혹은 병렬 구조로만 구성된 워크플로우를 가지는 복합 서비스에 적합하다. 또한 큰 규모의 서비스 선택 문제에 대해서 수행시간이 느리다는 단점을 갖는다.

본 논문은 복합 서비스의 워크플로우 및 제약사항을 적절히 분할(partition)하는 휴리스틱에 기반한다. 이를 기존의 LP 기반의 선택 기법에 적용하여 분할된 각각의 문제를 독립적으로 해결한다. 실험을 통하여 기존 방법과 비교한 결과 제안된 방법이 서비스 품질의 저하는 적절한 수준으로 유지하는 반면 평균 수행속도를 크게 단축시킨다는 것을 확인하였다.

#### 2. 관련 연구

Zeng 등 [1]은 순차 및 병렬 구조의 복합 서비스를 대상으로 LP를 이용한 서비스 선택 방법을 제안하였다. 이 방법은 최적의 서비스를 선택하는 반면 수행시간이 기하급수적으로 증가한다는 단점이 있다.

Cardoso 등 [2]은 순차, 병렬, 조건, 루프 등의 구조를 가지는 워크플로우의 서비스 품질을 각 단일 서비스의 품질로부터 도출할 수 있는 방법을 제시하였다. 하지만 서비스 선택에 관한 연구는 이루어지지 않았다.

Yu 등 [3]은 서비스 선택 방법을 MKCP (Multiple Choice Knapsack Problem)으로 모델링한 뒤, 이를 효율적으로 해결할 수 있는 방법을 제안하였다. 하지만 이러한 방법은 하나의 제약사항이 주어진 경우에만 적용할 수 있다.

Jaeger [4]는 제약사항을 반영하기에 적합한 서비스 품질 모델과 서비스 품질 결합 방법을 제안하고, 워크플로우 패턴에 기반한 서비스 선택 방법을 제안하였다 [5]. 하지만 패턴 기반의 방법은 전체적인(end-to-end) 서비스 품질을 고려하지 못한다.

Canfora 등 [6]은 GA(Genetic Algorithm)를 이용한 웹 서비스 선택 방법을 제안하였다. GA를 이용한 방법은 제약사항을 만족하는 서비스가 존재하더라도 이를 찾지 못하는 경우가 발생할 수 있다.

\* 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (2004-041-D00613)

### 3. 제안된 방법

본 논문은 Zeng 등이 제안한 LP 기반의 웹 서비스 선택 기법을 기반으로 주어진 LP 문제를 두 부분으로 분할하여 해결한다. 제안된 방법은 <그림 2>와 같다.

```

입력 : 전체 워크플로우 W0, 후보 서비스 집합 S
      제약사항 C0, 사용자 선호도 We
1  {<W1, C1>, <W2, C2>} = 분할(W0, C0)
2  P1 = LP문제생성(W1, S, C1, We)
3  P2 = LP문제생성(W2, S, C2, We)
4  Result = 서비스선택(P1, P2)
5  If (Result == SUCCESS)
6    return Result
7  else // 기존 방법으로 시도
8    P0 = LP문제생성(W0, S, C0, We)
9    Result = LP(P0)
10 return Result
    
```

그림 2. 제안된 방법의 의사코드.

#### 3.1 복합 웹 서비스의 워크플로우 분할

워크플로우의 분할은 분할된 문제의 크기를 고려하여 이루어진다. 문제의 크기가 작으면 작을수록 개별 문제에 대한 수행시간은 급격히 줄어들지만, 적절하지 못한 제약사항의 분할에 따른 서비스 선택 실패 및 선택된 결과 서비스의 품질 저하가 심화될 가능성이 높다. 이러한 이유를 고려할 때, 후보 서비스의 개수가 1:1 비율에 근접하도록 이분하는 방법이 가장 적절하다고 판단된다. 분할 과정은 주어진 후보 서비스 개수가 m개일 때, O(m)의 시간복잡도를 가진다.

#### 3.2 복합 웹 서비스의 제약사항 분할

워크플로우 분할의 다음 과정으로 제약사항의 분할이 이루어진다. 실패 확률을 줄이는 동시에 서비스 품질을 높이기 위해서는 실제 최적의 서비스 선택 결과를 분할하였을 때의 값에 가깝게 하는 것이 유리하지만, 이러한 과정은 서비스 선택에 걸리는 시간만큼이나 오래 걸릴 수 있다. 따라서 제안된 방법은 단순히 실패확률을 줄이기 위한 방향으로 분할을 시도한다. 우선 각각의 서비스 품질 요소에 대해 분할된 워크플로우가 가질 수 있는 최상의 품질을 계산한다. 이 값들은 실제 만족 가능한 가장 엄격한 제약사항에 대한 지표가 된다. 직관적으로 각 작업에 대한 후보 서비스 집합 중 해당 품질 요소에 대해 최상의 품질을 가지는 서비스들로 복합서비스를 구성함으로써 해당 품질 요소에 대한 최상의 품질 값을 구할 수 있다. 분할된 제약사항은 식 (1)과 식 (2)를 이용하여 계산된다. 식 (1)은 지연시간 및 가격과 같은 덧셈에 기반한 결합 방식을 가지는 품질 요소에 적용하며, 식 (2)는 가용성과 신뢰성과 같은 곱셈에 기반한 결합 방식을 가지는 품질 요소에 적용한다.

$$C'_q = C_q \times \frac{Q_q^{best}}{Q_q} \quad (1)$$

$$C'_q = \sqrt{C_q \times \frac{Q_q^{best}}{Q_q}} \quad (2)$$

$C_q$ 는 품질 요소  $q$ 에 대한 제약사항을,  $Q_q^{best}$ 는 품질 요소  $q$ 에 대한 최상의 품질 값을 의미한다. '과' '으로' 표시된 변수는 분할된 워크플로우 각각에 대한 변수이다. 위의 식을 이용해 분할된 두 워크플로우 중 한쪽의 제약사항을 계산하면  $C'_q$ 와 앞서 계산된 결과 값을 이용해 다른 한쪽의 제약사항도 간단히 도출할 수 있다. 위의 식은 각 품질 요소를 개별적으로 고려하였기 때문에 모든 요소를 동시에 고려하였을 경우와 차이가 날 수 있지만 빠르고 단순하다는 장점이 있다. 또한 각 품질 요소에 대한 최상의 값을 구하는 과정은 기존 방법의 정규화 과정에서 사용되기 때문에 추가적인 시간을 요구하지 않는다고 볼 수 있다.

#### 3.3 웹 서비스 선택 과정

분할된 워크플로우와 제약사항은 LP를 이용한 Zeng 등의 방법과 동일하게 해결된다. 두 문제가 해결되는 순서는 상관없으며, 한 문제가 성공할 경우 원래 주어진 제약사항과 성공한 결과의 차이를 비교하여 그 차이만큼 다른 문제의 제약사항을 완화 (relax) 시킨다. 두 문제 모두 해결되지 못할 경우엔 실패를 반환한다. 이 경우 기존의 방법과 동일하게 전체 문제에 대한 해결을 시도하게 된다. 이러한 경우 기존 방법을 단독으로 사용하였을 때에 비하여 실패에 따른 추가적인 수행시간을 요구하지만 사용자가 요청한 제약사항을 만족하는 복합 서비스가 존재할 경우, 항상 찾을 수 있음을 보장한다. 만약 한 문제만이 성공할 경우, 성공한 문제에 대한 제약사항의 조정 과정을 거친 후 다시 시도한다. 이러한 재시도 과정은 실행 횟수나 시간제한 하에 이루어질 수 있다. 각 품질 요소의 제약사항은 서로 도메인이 다르기 때문에 실제 제약사항이 취할 수 있는 범위사이의 위치를 나타내는 값  $\alpha$ 를 계산하여 이를 통하여 조정한다. 품질 요소  $q$ 에 대한  $\alpha$ 는 다음의 식을 통해 계산된다.

$$\alpha_q = \left| \frac{Q_q - Q_q^{worst}}{Q_q^{best} - Q_q^{worst}} \right| RIGHT \quad (3)$$

$Q_q$ 는 실제 조정하려는 품질 요소  $q$ 의 값이며  $Q_q^{best}$ 와  $Q_q^{worst}$ 는 각각 품질 요소  $q$ 의 최상 및 최하 품질의 값이다. 제약사항의 조정 방법은 모든 품질 요소에 대한  $\alpha$ 값을 동일한 비율로 줄여나가는 방법과 하나의 요소에 대해서는  $\alpha$ 값을 증가시키는 반면 나머지 요소는  $\alpha$ 값을 감소시키는 두 가지 방법을 평가하였으며, 후자의 경우가 상대적으로 나은 결과를 보여주었다. 하지만 두 경우 모두 재시도를 통한 성공확률은 매우 낮았다.

4. 실험 및 결과분석

실험 방법은 각각 4, 8, 12, 16, 20, 24개의 순차적 작업을 가지는 복합 서비스를 대상으로 수행하였으며, 각 작업 당 후보 서비스 수는 75-80개를 생성하였다. 각 작업은 5개의 서비스 클래스 중 임의의 한 가지에 속하게 되며 동일 클래스의 후보 서비스들은 서로 유사한 특성의 품질을 가진다. <표 1>은 서비스 클래스와 품질 요소 설정 기준을 나타낸다. 요청으로 사용된 제약사항은 지연시간, 비용, 가용성, 신뢰성에 대하여  $Q^{worst}$  부터  $Q^{best}$  까지 동시에  $\alpha$ 를 0.01씩 증가시키는 방법으로 생성되었다. 실험 결과는 각 작업 크기 당 10개의 서로 다른 인스턴스에 대한 평균을 구하였으며, 실험은 Intel Xeon™ 3.0GHz, 1GB RAM, Microsoft Windows™ Server 2003 환경에서 수행되었다.

표 1. 실험에 사용된 서비스 클래스와 품질 요소 설정 기준

품질 요소	품질 클래스				
	1	2	3	4	5
지연 시간	30-50	30-50	100-200	150-200	200-300
가격	0-30	100-150	0-100	300-400	300-500
가용성	0.99-0.999	좌동	좌동	좌동	좌동
신뢰성	0.99-0.999	좌동	좌동	좌동	좌동

<그림 3>은 기존 Zeng 등의 방법과 제안된 방법의 평균 수행시간을 그래프로 나타낸 것이다. 두 결과를 비교한 결과 제안된 방법은 후보 서비스 수가 증가함에 따라 기존 방법에 비해 빠른 수행시간을 나타냈으며, 수행시간의 증가 비율 역시 크게 둔화되었다는 것을 알 수 있다. 한편 제안된 방법이 서비스 품질에 미치는 영향을 분석하기 위해 기존 방법과 제안된 방법의 서비스 품질 차이를 비교하였다.

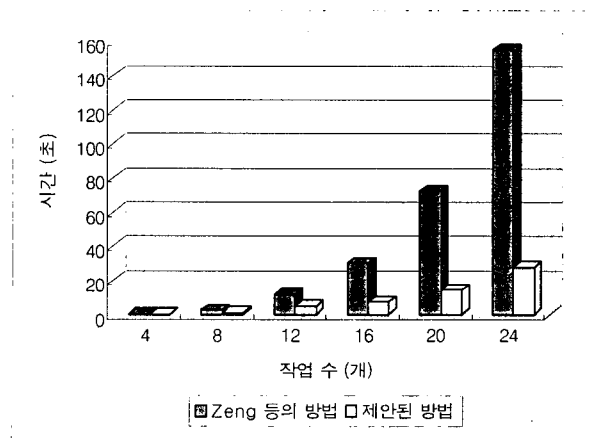


그림 3. 평균 수행시간 비교.

<표 2>는 제안된 방법을 사용했을 경우, 기존 방법과의 서비스 품질 차이를 백분율로 표시한 결과이다. 음수로 표시된 것은 품질 저하를 양수는 반대의 경우를 뜻한다. 이 결과는 작업 수가 8개(약 640개의 후보 서비스) 이상인 경우, 제안된 방법의 품질 저하가 기존 방법의 1% 미만임을 보여주고 있다. 지연시간과 비용에 비해 가용성과 신뢰성의 품질 저하가 적게 나타난 이유는 사용자의 기호를 전자에 두었기 때문이며, 제안된 결과가 기존 방법과 동일하게 사용자의 기호를 반영하고 있음을 보여준다.

표 2. Zeng 등의 방법과 제안된 방법의 평균 서비스 품질 차이

작업 수	지연시간	비용	가용성	신뢰성
4	-6.41%	-4.33%	0.13%	0.05%
8	-0.74%	-0.75%	0.00%	-0.03%
12	-0.44%	-0.61%	-0.02%	-0.04%
16	-0.35%	-0.60%	-0.05%	-0.05%
20	-0.28%	-0.19%	-0.03%	-0.04%
24	-0.20%	-0.36%	-0.03%	-0.03%

5. 결론

본 논문은 기존의 LP를 이용한 방법에 워크플로우와 제약사항을 분할하는 휴리스틱을 적용하여 수행시간을 단축시키는 방법을 제안하였다. 또한 실험을 통해 제안된 방법이 기존 방법에 비해 빠르며, 좋은 확장성을 가지는 것을 확인하였다. 향후 제약사항의 보다 정교한 분할이 이루어질 수 있도록 제안된 휴리스틱을 개선할 계획이다.

6. 참고문헌

- [1] Liangzhao Zeng, Boualem Benatallah and Marlon Dumas, "Quality Driven Web Services Composition," Proc. 12th Int'l Conf. World Wide Web, pp. 411-421, 2003.
- [2] Jorge Cardoso, Amit Sheth, John Miller, Jonathan Arnold, and Krys Kochut, "Quality of Service for Workflows and Web Service Processes," Web Semantics Journal, Vol. 1, No 3, pp. 281-308, 2004.
- [3] Tao Yu and Kwei-Jay 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.
- [4] Michael C. Jaeger, Gregor Rojec-Goldmann, and Gero Muehl, "QoS Aggregation in Web Service Compositions using workflow patterns," Proc. 8th Int'l Conf. EDOC, pp. 149-159, 2004.
- [5] Roy Gronmo and Michael C. Jaeger, "Model-Driven Methodology for Building QoS-Optimized Web Service Compositions," Proc. 5th IFIP WG 6.1 Int'l Conf. on DAIS (LNCS 3543), pp. 68-82, 2005.
- [6] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito and Maria Luisa Villani, "An Approach for QoS-aware Service Composition based on Genetic Algorithms," Proc. Conf. Genetic and Evolutionary Computation, pp. 1069-1075, 2005