

웹 서비스 선택 시 로컬 브로커를 이용한 실행 단계 결정

김태원⁰ 송영재
경희대학교 컴퓨터공학과
woni@se.khu.ac.kr

Execution Path Selection Using

Local Broker in Web Service Selection

Tae-Won Kim⁰ Young-Jae Song
Dept of Computer Engineering, Kyung-Hee University

요 약

컴포넌트 지향 개발 방법이 도입되면서 많은 분산환경하에 컴포넌트들이 상호연동하여 사용되어지기 시작했다. 그러나 분산환경에서의 상호연동은 많은 어려움과 과제를 우리에게 안겨주었다. 이를 보완하여 상호운영성을 고려한 웹 서비스의 개념이 나오기 시작하고 있다. 웹 서비스는 이기종간의 컴포넌트 서비스를 쉽게 구현할 수 있게 하며 컴포넌트들을 이용에 블록을 쌓듯이 애플리케이션을 만들 수 있다. 그러나 웹 서비스는 아직 QoS 문제가 남아있다. 웹 서비스의 품질을 보장하고 사용자가 웹 서비스를 안전하게 사용하기 위해서는 QoS 기반 컴포넌트 조립이 필수적이다. 본 논문은 로컬 브로커와 XML-QoS 스키마를 사용하여 사용자가 QoS를 고려한 실행 계획을 세울 수 있게 한다.

1. 서론

현재 웹 환경은 운영 환경이 다양하며 다양한 플랫폼에서 다양한 언어로 구현되어져 있는 분산 컴퓨팅 환경이 주를 이루고 있다. 이러한 이기종 간의 분산 환경에서는 통합을 위해서 많은 비용과 노력이 요구되며, 소프트웨어의 확장성·통합성·상호운영성과 같은 소프트웨어의 기본적인 요구 사항을 만족시키지 못 한다[1].

분산 컴퓨팅의 문제점은 네 가지 정도로 요약할 수 있다. 첫째, 연결된 상태에서 클라이언트 또는 서버의 갑작스런 연결 끊김이 발생했을 경우에 적절히 대응하지 못한다는 것이다. 둘째, 수천 개의 클라이언트가 서버로 연결되어 있는 경우, 이미 비 활동 중인 클라이언트로 인해 서버 쪽의 귀중한 자원이 낭비될 수 있다. 셋째, 분산 컴퓨팅의 프로토콜은 대부분의 방화벽을 통과하지 못한다. 넷째, 프로토콜의 구조가 모두 다르기 때문에 같이 사용하기 힘들고, 프로토콜 자체가 복잡하다.

이와 같은 분산 컴퓨팅의 문제점으로 인해 이기종간의 상호 운영성에 문제가 발생한다. 우리가 원하는 웹은 언제 어디서나 우리가 원하는 장치를 통해 서비스를 제공받는 것이다. 이를 위해 Microsoft, Sun, IBM이 제시한 것이 웹 서비스이다[2].

웹 서비스는 서비스 이용자가 필요한 서비스를 찾고, 서비스 제공자가 서비스와 제품을 배포하는 형태를 가진다. 웹 서비스 이용자는 비즈니스 처리를 위해 자신의 비즈니스 행위를 수행하는 것뿐만 아니라, 다른 웹 서비스와 협업하

여 비즈니스 처리를 수행한다. 단순한 웹 서비스로는 신용 체크, 인증, 재고 확인과 같은 간단한 기능을 제공하는 것이다. 한편, 복잡한 웹 서비스는 보험 중개, 여행 계획, 자동차 대여 서비스와 같은 상이한 비즈니스 기능들을 통합하여 자동화된 처리를 제공하는 것이다.

그러나 현재 웹 서비스의 이용에서는 QoS를 만족시키는 데 많은 문제를 가지고 있다. 사용자가 중요한 서비스의 이용 시 웹 서비스 제공자의 가용성이 떨어져 그 서비스를 이용하지 못하는 경우 사용자는 다른 선택의 여지가 없다.

기존 논문[3][4]는 웹 서비스 브로커(Web Service Broker)와 QoS 기반 웹 서비스 합성을 제시하여 이를 극복하고자 한다. 그러나 웹 서비스 브로커는 성능의 저하가 문제가 된다. 따라서 본 논문은 로컬 브로커를 래퍼형식으로 제시하며 QoS를 정의하기위해 XML을 사용한 QoS-XML 스키마를 제시한다.

2. 관련 연구

2.1 웹 서비스 브로커(Web Service Broker)

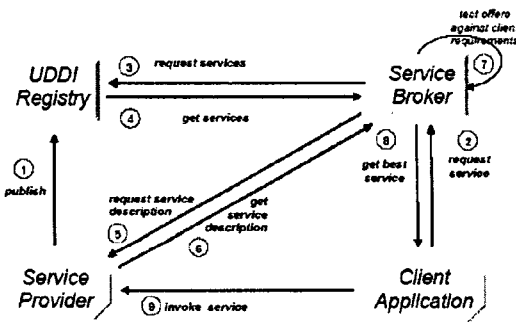
WSB[3]는 서비스 사용자의 QoS 요구사항을 고려하여 최적의 웹 서비스를 선택하는 메커니즘을 제공한다. WSB는 웹 서비스가 제공하는 서비스를 미리가지고 있을 수도 있고 없을 수도 있다. 이 경우에는, UDDI로부터 서비스에 대한 정보는 가지고 온다.

웹 서비스의 사용자는 UDDI에 접속하는 대신 서비스를 찾기 위해 WSB에 접속한다. WSB에 현재 이용 가능한 서

비스에 대한 정보가 없을 시에는, 하나이상의 UDDI에 관련 되어진 서비스에 대한 정보를 검색한다. 결과로 전달되어진 WSDL 파일은 WS-QoS의 검사에 사용되어지고 새로운 WSB의 리스트가 생성되어진다. 이 리스트를 통해 사용자를 위한 최상의 웹 서비스를 추출하여 준다.

이 리스트를 유지하기 위해 WSB는 UDDI에 정기적으로 검사를 수행한다. 웹 서비스가 소멸되었을 경우 WSB의 레지스트리에서도 삭제된다. 만약 웹 서비스가 확장되어 졌다면 다음 검사시 확장되어진 서비스의 내용이 추가되어진다.

만약, 사용자가 가장 저렴하게 이용할 수 있는 서비스를 요청했을때 사용자는 QoS 요구사항을 WSB에게 보낸다. WSB는 이용가능한 서비스들을 테스트하고 사용자의 요구 사항에 일치하는 서비스를 선택하고 사용자에게 결과를 돌려준다. 이러한 WSB 기반 웹 서비스의 선택은 QoS 인자를 통한 전략을 구현하고 최적의 서비스를 선택하는데 있어서 가치 있는 것이다.



(그림 1) WSB를 이용한 웹 서비스 검색

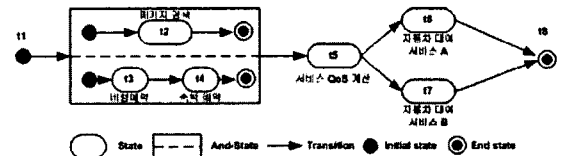
WSB의 구현에는 두 가지 경우가 있다. 첫째는, 사용자의 애플리케이션 안에서 수행되어지는 WSB이다. 둘째는, 다중 사용자를 위한 원격의 WSB의 구현이다. 원격의 WSB의 경우 UDDI의 확장이라고 할 수 있다. 또한 성능이 떨어지며 불필요한 많은 웹 서비스에 대한 정보를 유지하여야 하며, 정보가 세부적으로 기술되지 못 한다. 로컬의 WSB는 웹 서비스의 많은 정보를 가지고 있지는 않지만 사용자에게 필요한 정보를 세부적으로 정의하여 유지할 수 있으며 성능면에서도 뛰어나다. 기존 논문에서는 원격의 WSB의 구현을 사용하였다.

그림3은 서비스 제공자, 사용자, UDDI와 WSB 사이의 상호작용을 나타낸다. 일반적인 웹 서비스 모델[2]에서는 WSB가 존재하지 않는다. WSB는 서비스간의 효과적인 비교를 위해 웹 서비스 QoS XML 스키마(Web Service-QoS XML Schema)를 사용한다.

2.2 QoS 기반 실행 단계와 계획
2.2.1 실행 단계와 계획

그림2 상태도(statechart)[4]의 실행 단계(Execution path)는 상태(state)[t_1, t_2, \dots, t_n]의 흐름이다. t_1 은 초기 상태이고 t_n 은 최종 상태이다. 모든 상태 t_i 의 i 는 $1 < i < n$ 이다.

$P = \{ \langle t_1, s_{i2} \rangle, \langle t_2, s_{i2} \rangle, \dots, \langle t_n, s_{in} \rangle \}$ 은 여행 서비스 실행 단계의 실행 계획을 나타낸다.



(그림 2) "여행 서비스"를 위한 합성 서비스의 상태도

2.2.2 실행 계획을 위해 고려되어야할 QoS

① 실행 비용(Execution price)

실행 계획 p의 실행 비용 $Q_{price}(p)$ 는 모든 서비스 s_i 의 실행 비용의 합이다.

② 실행 시간(Execution duration)

실행 계획 p의 실행 시간 $Q_{du}(p)$ 는 Critical Path Algorithm(CPA)를 사용하여 처리된다. critical path는 초기 상태에서 최종 상태까지 각각의 노드에서 발생하는 최대 시간들의 총합이다.

③ 신뢰성(Reliability)

신뢰성은 서비스의 질을 유지할 수 있는 정도를 나타내는 질적인 면이다. 실행 계획 p의 신뢰성 $Q_{rel}(p)$ 는 $e^{Q_{rel}(s_i) \times z_i}$ 의 곱이다. 만약 서비스 s_i 가 critical service이면 z_i 는 1이다. 그렇지 않고 서비스 s_i critical service가 아닐 경우 z_i 는 0 이고 $e^{Q_{rel}(s_i) \times z_i} = 1$ 이다. 따라서 서비스 s_i 의 신뢰성은 실행 계획의 신뢰성에 영향을 주지 않는다.

④ 가용성(Availavility)

가용성은 서비스가 현재 존재하고 즉시 사용할 수 있는지에 대한 질적인 면을 의미한다. 실행 계획 p의 가용성 $Q_{av}(p)$ 는 $e^{Q_{av}(s_i) \times z_i}$ 의 곱이다. $Q_{av}(s_i)$ 는 서비스 s_i 의 가용성이다.

합성 서비스에서 실행 계획의 QoS는

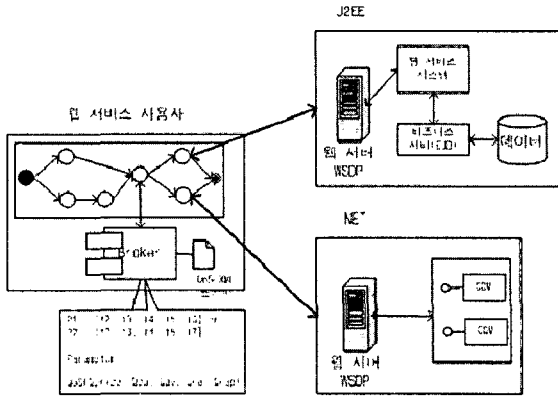
$$Q(p) = (Q_{price}(p), Q_{du}(p), Q_{re}(p), Q_{rep}(p))$$

으로 나타낸다.

3. QoS를 고려한 실행 단계 결정

3.1 로컬 브로커

기존 논문[3]에서는 웹 서비스 브로커를 원격으로 두어 UDDI와 상호 연동하는 아키텍처를 지니고 있다. 그러나 이는 성능의 저하와 사용자의 QoS 요구사항을 세부적으로 만족시키지 못 하는 단점을 지니고 있다. 이를 보완하기 위해 본 논문은 웹 서비스 사용자의 애플리케이션 안에 래퍼로



(그림 3) 로컬 브로커를 사용한 실행 계획 결정

로컬 브로커를 두어 성능의 향상과 사용자의 QoS를 만족시킨다.

그림3은 로컬 브로커를 두어 QoS를 측정하여 최적의 실행 계획을 동적으로 결정하는 아키텍처를 보여준다. 로컬 브로커는 이미 정해진 계획에 따라 QoS를 측정하여 웹 애플리케이션 실행시 웹 서비스를 연결하여 준다. 로컬 브로커에는 인자로 사용자 정의 QoS가 넘겨진다.

사용자 정의 QoS는 XML 데이터 형식으로 구현되어진다. 이를 위해 QoS-XML 스키마를 이용한다.

3.2 QoS-XML 스키마

QoS-XML 스키마를 나타내기 위해 기존 논문[4]에서 제시한 웹 서비스의 QoS를 이용한다. 로컬 브로커는 QoS-XML 데이터를 가지고 있으며 애플리케이션의 실행시 데이터를 이용한다. 또한 실행 계획의 내용을 저장하여 데이터를 갱신한다.

```

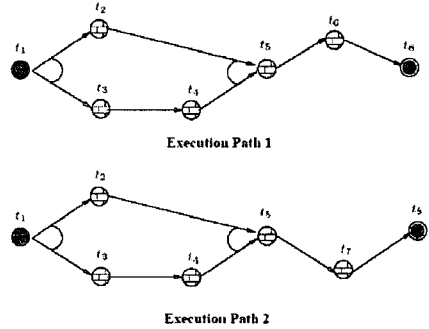
<?xml version="1.0" encoding="utf-1" >
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
<xs:element name="서비스요청" >
<xs:complexType >
<xs:sequence >
<xs:element name="데스크" minOccurs="0" maxOccurs="unbounded" >
<xs:complexType >
<xs:sequence >
<xs:element name="price" type="xsd:int" >
<xs:element name="rating" type="xsd:int" >
<xs:element name="reputation" type="xsd:int" >
<xs:element name="reliability" type="xsd:int" >
<xs:element name="availability" type="xsd:int" >
</xs:sequence >
<xs:attribute name="기타" type="xsd:string"/ >
</xs:complexType >
</xs:element >
</xs:sequence >
</xs:complexType >
</xs:element >
</xs:schema >
    
```

(그림 4) QoS-XML 스키마

3.3 실행 단계 결정

실행 단계 $p1 = \{ \langle t_2, s_{23} \rangle, \langle t_3, s_{38} \rangle, \langle t_4, s_{45} \rangle, \langle t_5, s_{42} \rangle, \langle t_7, s_{62} \rangle \}$ 이다. 그리고 $p2 = \{ \langle t_2, s_{23} \rangle, \langle t_3, s_{38} \rangle, \langle t_4, s_{45} \rangle, \langle t_5, s_{42} \rangle, \langle t_6, s_{47} \rangle \}$ 이다.

QoS의 실행 시간을 고려한 실행 단계를 결정할 때 단계2인 $\langle t_3, t_4, t_5, t_6 \rangle$ 의 수행 시간이 47초이고 단계1인 $\langle t_3, t_4, t_6, t_7 \rangle$ 의 수행 시간이 62초일때 critical path는 단계



(그림 5) "여행 서비스"의 실행 단계

2이다.

4. 결론 및 향후 과제

본 논문은 웹 서비스의 이용 시 QoS를 보장하는 실행 계획에 중점을 둔다. 이를 위해, 로컬 브로커와 QoS-XML 스키마를 이용한다. 이는 현재 웹 서비스의 이용 시 문제가되는 QoS를 보장하며 원격의 웹 서비스 브로커에 비해 향상된 성능을 기대할 수 있다.

향후 연구 과제로는 로컬 브로커와 QoS-XML 스키마의 세부적인 설계와 구현이다. 또한 구현 단계에서 미처 생각하지 못한 서비스의 동적인 추가를 고려한 합성 메커니즘의 구현이다. 구현은 브로커를 비주요하게 관리할 수 있으며 사용자 정의 QoS를 변환할 수 있는 툴의 개발로 이루어질 것이다.

참고문헌

- [1] "Web service Architecture Requirement", W3C, <http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211/>.
- [2] "Web Service Architecture", W3C <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
- [3] Tian, M. Gramm, A. Naumowicz, T. Ritter, H. Schiller, J., "A Concept for QoS Integration in Web Service", Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on , 13 Dec. 2003, pp149 - 155.
- [4] Liangzhao Zeng. Boualem Benattallah. Marlon Dumas., "Quality Driven Web Service Composition", The Twelfth International World Wide Web Conference.