

QoS 보장을 위한 시뮬레이션 기반 웹 서비스 조합 도구

장희정*, 이강선**

Simulation-Based Web Service Composition for Guaranteed QoS (Quality of Service)

Heejung Chang, Kangsun Lee

Abstract

최근 웹 서비스(Web Service)와 웹 프로세스(Web Process)는 이기종 시스템의 통합과 웹 애플리케이션 배포를 위한 대안으로 주목받고 있으며, 이에 따라 웹 서비스의 QoS (Quality of Services)의 중요성도 증가하고 있다. 본 논문에서는 웹 서비스의 QoS 분석 방법을 제안하고, QoS 중심의 웹 서비스 조합 도구인 SimQom4W (Simulation-based Qos-aware Composer for Web Services)를 소개한다. SimQom4W는 첫째, 웹 서비스와 웹 프로세스의 구성을 시각으로 기술하고, 둘째, 웹 프로세스에 대한 테스트 코드와 시뮬레이션 코드를 자동 생성하여 실행시킴으로써 QoS를 분석한다. 셋째, 다양한 QoS 요구 사항을 만족하는 최적의 웹 프로세스를 선택할 수 있도록 하며, 넷째, 웹 서비스의 QoS 분석 및 평가 데이터의 보고서를 생성하여 제공한다.

Key Words: Web Services, Web-based modeling and simulation, QoS analysis and estimation

* 명지대학교 컴퓨터공학과 대학원

** 명지대학교 컴퓨터공학과

1. 서론

웹 서비스의 활용이 증가됨에 따라, QoS (Quality of Services)는 웹 서비스의 조합과 선택에 필수적인 요소로, 점차 그 중요성이 증가하고 있다[1]. 웹 프로세스의 QoS는 이를 구성하는 웹 서비스들의 가용성, 보안, 신뢰성, 성능 등의 여러 품질 요소들의 조합으로 표현된다[2]. 이러한 QoS 요소들은 웹 서비스들의 실제 실행 환경에 따라 동적으로 변화하게 된다. 따라서 웹 서비스의 설계 및 조합 시에 QoS를 측정하는 작업에는 어려움이 따른다.

웹 서비스의 QoS 분석법으로 테스트 기반의 분석법, 시뮬레이션 기반의 분석법 등이 있다. 시뮬레이션 기반 분석법은 웹 프로세스에 대한 시뮬레이션 모델을 작성하고 실제 실행 없이 시뮬레이션을 통해 성능을 분석하는 방법이다. 시뮬레이션 기법은 분석에 소요되는 시간과 비용을 절감하고 다양한 환경에서의 성능분석이 가능하다[3]. 그러나 시뮬레이션 모델과 환경 변수에 대한 타당성이 검증되지 않으면 분석 결과의 정확성을 보증하기 어렵다는 단점이 있다. 테스트 기반 분석법은 실제 환경에서 웹 서비스를 호출하고 결과를 분석하는 방법이다[4]. 이는 분석 결과에 대한 높은 신뢰성을 제공하나 분석에 소요되는 시간과 비용이 높다. 본 연구에서는 시뮬레이션 기반의 분석법에 테스트 기법을 적용하여 분석에 소요되는 비용 및 시간을 절감하고, 분석 결과의 정확성을 높이고자 한다.

본 논문에서는 QoS 중심의 웹 서비스 조합 도구인 SimQom4W (Simulation-based Qos-aware Composer for Web Services)를 소개하고, 웹 서비스 조합 시 QoS의 만족 여부를 미리 고려할 수 있음을 보인다. 본 논문

의 구성은 다음과 같다. 2장에서 SimQom4W의 QoS 분석 방법과 시스템 아키텍처를 소개하고 3장에서는 도구를 활용한 예를 보인 뒤 4장에서 결론을 맺는다.

2. 웹 프로세스 조합 지원 도구

2.1 QoS 품질 요소

2.1절에서는 SimQom4W의 QoS 요소에 대해 소개한다. SimQom4W는 웹 서비스 서버의 로그 파일(log file)을 분석하여 가용성과 신뢰성을 평가한다. 웹 서버는 2xx, 5xx 등의 HTTP 상태코드를 통해 요청에 대한 웹 서버의 처리 결과를 로그 파일에 기록한다. 2.1.1절과 2.1.2절에서 웹 서버의 로그 파일로부터 가용성과 신뢰성을 평가하는 과정을 설명한다. 또한, 성능은 테스트와 시뮬레이션 기반의 분석법을 통해 분석되며, 2.1.3절에서 설명하도록 한다.

2.1.1 가용성

웹 서비스의 가용성은 웹 서비스 전체 가동 시간 중 실제로 사용자가 사용할 수 있는 시간의 비율을 나타내며 식(1)과 같이 정의된다.

$$\text{Availability} = 1 - \frac{\text{SER}(s)}{\Delta} \quad (1)$$

이때, SER(s)는 서버 에러를 나타내며 5xx 상태 코드를 측정하여 도출할 수 있다. 델타(Δ)는 전체 요청 수를 나타낸다. 5xx 상태 코드는 HTTP 1.1에 정의되어 있으며 서버가 요구를 처리할 능력이 없음을 인지한 경우를 표시한다.

2.1.2 신뢰성

신뢰성은 사용자가 요청한 전체 요청 중 올바른 응답의 비율을 나타내며 식(2)와 같이 정의된다.

$$\text{Reliability} = \frac{\text{SR}(s)}{\Delta} \quad (2)$$

SR(s)는 전체 요청 중 성공적으로 처리된 요청의 개수를 나타내며, 이는 2xx 상태 코드부터 도출 될 수 있다. 2xx 상태 코드는 사용자의 요구가 성공적으로 수신, 해석 및 접수 되었음을 표시한다.

2.1.3 성능

QoS 요소들 중 성능은 네트워크의 상태나 동시 사용자 접속 패턴의 갑작스러운 변화 등으로 인해 평가에 어려움이 따르는 품질 요소이다. 성능 분석 결과에 대한 정확성을 높이고 분석에 소요되는 시간과 비용의 절감을 위해 SimQom4W는 시뮬레이션 기반 분석법에 테스트 기반 분석법을 적용하여 성능을 평가한다. 본 논문에서는 DRT(Dissected Response Time)와 TRT(Traced Response Time)의 두 가지 측정 기준을 통한 성능 평가를 제안한다. DRT는 *Network Time(N)*, *Messaging Time(M)*, *Service Time(S)*의 세 가지 요소로 구성되며 단일 웹 서비스 s에 대하여 식(3)과 같이 정의된다.

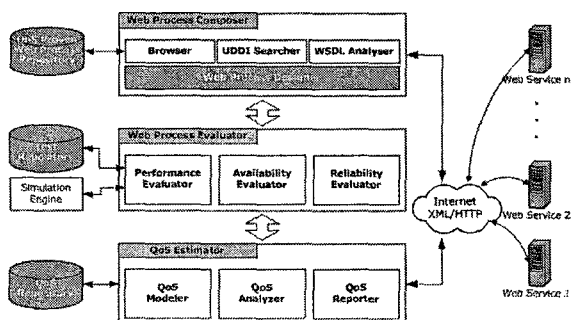
$$T(s) = N(s) + M(s) + S(s) \quad (3)$$

*Network Time*은 네트워크 대역폭, 트래픽, 장비의 성능 등으로 인한 지연을 나타내며, *Messaging Time*은 서비스 제공자가 SOAP 메시지를 처리하는 데 소요되는 시간을, *Service Time*은 웹 서비스가 요청을 처리하기 위해 소요되는 시간을 의미한다. 웹 프로세스가 웹을 통한 서비스를 제공하게 되면 각각의 웹 서비스에는 동시 사용자 증가에 따른 성능 저하가 발생하게 된다. TRT는 가상의 동시 사용자를 발생시켜 동시 사용자의 증가에 따른 웹 프로세스의 성능을 평가한다.

TRT에는 시간과 시스템 자원이 높게 요구되며, 메모리와 네트워크의 조건, 웹 서비스의 서버 프레임워크 등에 의해 테스트 가능한 최대 동시 사용자의 수에 영향을 받는다. 이러한 제약을 극복하기 위하여 시스템의 자원이나 웹 서비스의 호출 없이 시뮬레이션을 통해 TRT를 평가하며, 결과의 정확성을 높이기 위해 실제 서비스 호출을 통한 DRT 결과를 적용하여 시뮬레이션을 수행, 과부하 조건에서의 프로세스의 성능을 예측한다.

2.2 SimQom4W 시스템 아키텍처

SimQom4W를 구성하는 주요 구성 요소로는 *Web Process Composer*, *Web Process Evaluator*, *QoS Estimator*가 있다<그림 1>.



<그림 1> SimQom4W 시스템 아키텍처

*Web Process Composer*를 통해 사용자는 UML의 Activity Diagram을 사용하여 웹 프로세스를 명세하고, QoS 요구사항 및 테스트와 시뮬레이션에 필요한 환경 변수들을 결정한다. Activity Diagram은 각각의 서비스를 노드(node)로, 실행의 흐름을 링크(link)로 표현하며 분할(fork), 합병(join)등의 표현을 통해 다양한 실행 조건을 표현한다. *Web Process Evaluator*는 테스트와 시뮬레이션을 통해 QoS 요소들을 평가한다. 프로세스가 구성되면 동적으로 테스트 코드와 시뮬레이션 코드를 생성하고, 테스트를 수행하여 측정 결과를 히

스토리 데이터베이스에 저장한다. SimQom4W의 시뮬레이션 모델은 Simjava[5]를 기반으로 구성되며 Activity Diagram로 부터 자동 변환된다. 시뮬레이션 모델이 작성되면 앞서 수행된 테스트 결과를 분석하여 단일 요청에 대한 평균 응답시간 및 분산을 구하고 이를 모델에 적용하여 시뮬레이션을 수행한다. QoS Estimator는 사용자의 다양한 QoS 요구사항을 만족하는 최적의 프로세스 구성을 선택할 수 있도록 한다. QoS 요구사항은 다양한 QoS 요소로 표현될 수 있다. 예를 들어 어떤 사용자는 최대의 가용성과 신뢰성을 갖는 웹 프로세스를 구성하고자 할 수도 있고, 또 다른 사용자는 빠른 응답시간을 최우선 조건으로 하는 웹 프로세스를 요구할 수도 있다. 본 논문은 효과적인 사용자 요구사항의 표현을 위해 웹 프로세스의 구성 시 빈번히 제기되는 사용자의 QoS 요구사항을 다음의 3가지로 정형화하며 이에 대한 QoS를 다음의 절차에 따라 분석한다.

(유형1) 웹 프로세스 S가 사용자가 선택한 서비스 S_i 들로 구성되었다면 QoS 요소는 다음과 같이 도출될 수 있다.

$$\begin{aligned} \text{가용성} & : A(S) = \prod p_i A(S_i) & (4) \\ \text{신뢰성} & : R(S) = \prod p_i R(S_i) & (5) \\ \text{응답시간} & : T(S) = \sum p_i T(S_i) & (6) \end{aligned}$$

이때, p_i 는 서비스 S_i 가 선택되어 수행될 확률을 나타낸다.

(유형2) 동일 기능을 수행하는 다수의 서비스 S_i 가 존재할 경우 다음의 절차를 따른다.

Step 1: 서비스 S_j 가 j 개의 서비스 $\{S_{i1}, S_{i2}, \dots, S_{ij}\}$ 중 하나인 경우,

$$\text{가용성} : A(S) = \begin{cases} \prod p_i A(S_{ik}) & (A(S) \geq A_c), \text{ for } 1 \leq k \leq j \\ 0 & (\text{otherwise}) \end{cases} \quad (7)$$

$$\text{신뢰성} : R(S) = \begin{cases} \prod p_i R(S_{ik}) & (R(S) \geq R_c), \text{ for } 1 \leq k \leq j \\ 0 & (\text{otherwise}) \end{cases} \quad (8)$$

$$\text{응답시간} : T(S) = \sum p_i T(S_i), \text{ for } 1 \leq k \leq j \quad (9)$$

이때, A_c, R_c 는 각각 가용성과 신뢰성에 대한 한계치를 나타낸다.

Step 2: 모든 S_i 에 대해 Step1을 반복 적용한다.

Step 3: $A(S) \neq 0$ 이고 $R(S) \neq 0$ 인 프로세스의 조합을 선택하도록 추천한다.

(유형3) 가용성, 신뢰성 한계치를 만족하면서 최소의 응답시간을 갖는 서비스를 선택하여 프로세스를 구성하고자 한다면, 다음의 절차를 따른다.

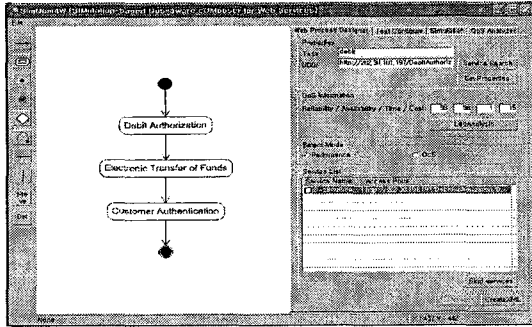
Step 1~2: 유형 2)의 경우와 동일

Step 3: $A(S) \neq 0$ 이고 $R(S) \neq 0$ 인 프로세스의 조합 중 최소의 $T(S)$ 를 갖는 프로세스를 추천한다.

3. 실험 및 결과

3장에서는 MyBillPayment 예제를 통해 제안된 도구의 활용 예를 보인다. MyBillPayment는 온라인 지불 프로세스로 전화요금, 신용카드 대금 결제 등의 서비스를 제공한다. *debit authorization, electronic transfer of funds, customer authentication* 서비스로 프로세스가 구성되며, *electronic transfer for funds* 서비스는 3개의 서비스 제공자 중에서, *customer authentication*은 2개의 서비스 제공자 중에서 선택하여 프로세스를 조합할 수 있다. 이때, 사용자는 프로세스가 최소의 응답시간과 90%이상의 신뢰성, 97%이상의 가용성을 만족하도록 서비스를 선

택하여 조합하고자 한다.



<그림 2> MyPaymentBill 프로세스 명세

사용자는 SimQom4W를 통해 <그림 2>와 같이 프로세스를 구성하고, 각 서비스들의 서버 로그 파일을 분석하여 가용성과 신뢰성을 평가한다. 평가된 결과를 바탕으로 사용자의 QoS 요구사항에 부합하는 조합을 찾아 <debit authorization>-<electronic transfer of funds(2)>-<customer authentication(1)>의 조합을 사용자의 요구사항을 만족하는 조합으로 추천한다<그림 3>.

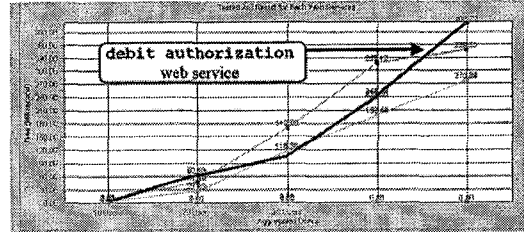
Constraint	Condition	Value
Availability	>=	97
Reliability	>=	90
Availability	>=	97

QoS Analysis Reports	
Average of Reliability	0.941894
Average of Availability	0.960498

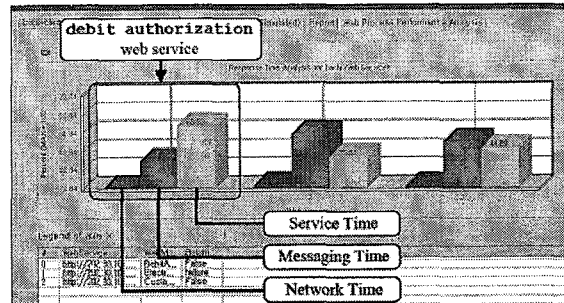
<그림 3> MyBillPayment의 QoS

<그림 4>, <그림 5>는 추천된 조합에 대한 성능 분석 결과이다. <그림 4>의 TRT 테스트 결과를 통해 동시 사용자가 증가되면 debit authorization 서비스에서 병목현상이 유발될 수 있음을 알 수 있다. 또한 <그림 5>의 DRT 테스트 결과를 통하여 debit authorization 서비스의 Service Time이 전체 응답시간에서 차지하는 비율이 상대적으로 높은 것을 확인할 수 있다. 따라서 debit authorization 서비스의 비즈니스 로직이나 하

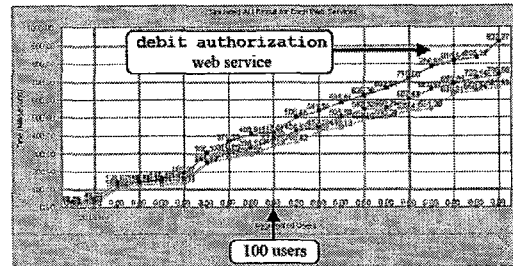
드웨어의 교체 등을 통해 전체 프로세스의 성능을 향상시킬 수 있음을 할 수 있다.



<그림 4> TRT 테스트 결과



<그림 5> DRT 테스트 결과



<그림 6> 시뮬레이션 기반 TRT 평가 결과

<그림 6>은 시뮬레이션 통해 10~200명의 동시 사용자 증가에 따른 성능을 예측한 결과를 보이고 있다. 동시 사용자가 100명 이상으로 증가할 경우, debit authorization 서비스에서 병목 현상이 발생할 수 있으며 전체 응답시간이 급격히 증가함을 알 수 있다. 사용자는 이러한 평가 자료를 통해 프로세스를 미리 예측하고 이를 통해 적절한 대응 방법을 수립할 수 있다.

4. 결론

본 논문에서는 QoS 보장을 위한 시뮬레이션 기반의 웹 서비스 조합 도구인 SimQom4W를 소개하였다. SimQom4W는 기존의 웹 서비스 표준 기술이나 웹 서비스 서버에 대한 수정 없이도 QoS를 평가할 수 있는 소프트웨어 아키텍처이다. 가용성과 신뢰성은 웹 서비스 서버의 로그 파일을 분석하여 평가하며, 성능 분석은 저부하 환경에서의 테스트 기반 분석법과 고부하 환경에서의 시뮬레이션 기반 분석법의 두 가지 분석법을 적용하여 분석에 소요되는 시간과 비용을 절감한다. 또한 구성된 웹 프로세스는 자동으로 시뮬레이션 모델로 변환되며, 테스트 기반 분석의 결과를 시뮬레이션의 파라미터로 적용하여 평가의 정확성을 높이게 된다.

Library", <http://www.dcs.ed.ac.uk/home/hase/simjava>, 1996

참고문헌

- [1] Eric Newcomer, "Understanding Web Services: XML, WSDL, SOAP and, UDDI", Addison-Wesley, 2002
- [2] IBM Korea, "웹 서비스 요소기술", <http://www-903.ibm.com/kr/software/kr/element/element.html>, 2003
- [3] Gregory Silver, John A. Miller, Jorge Gardoso, Amit P. Sheth, "Web service technologies and their synergy with simulation", In Proc. of the 2002 Winter Simulation Conference, pp.606 - 615
- [4] J.D. Meier, Srinath Vasireddy, "How To: Use ACT to Test Web Services Performance", Microsoft Developer Network, Microsoft Corporation, 2004
- [5] Fred Howell, Ross McNab, "Simjava