

웹 서비스 시스템의 품질 기반 장애 극복 방안

이용표^o 신재동 한상용

중앙대학교 컴퓨터공학과

{drgnyp^o, mulli2}@archi.cse.cau.ac.kr, hansy@cau.ac.kr

Fault-Tolerant in the Web Services System

Yongpyo Lee^o, Jaedong Shin, Sangyong Han

Dept. of Computer Science & Engineering, Chung-Ang University

요 약

웹에서의 협업과 통합의 대표적인 기술이라 할 수 있는 웹 서비스의 사용은 날로 증가되고 있다. 웹 서비스를 이용하여 애플리케이션을 구축할 때, 사용한 웹 서비스 중에서 장애가 발생했을 경우 애플리케이션의 신뢰성 있는 실행을 위해서는 장애를 극복할 수 있는 방법이 필요하다. 본 논문에서는 클라이언트측 애플리케이션에서의 수정 사항 없이 장애를 극복할 수 있는 시스템을 제안한다. 이 시스템에서는 일반적인 장애 상황뿐만 아니라 품질 요소들도 고려하여 요구사항에 만족하지 못할 경우 같은 서비스를 지원하는 웹 서비스로 자동 전환이 되도록 개선하였으며, 같은 서비스를 찾는 방법의 개선도 제안하고 있다. 이를 바탕으로 품질 기반의 장애 극복 웹 서비스 시스템을 구축할 수 있는 방안을 제시한다.

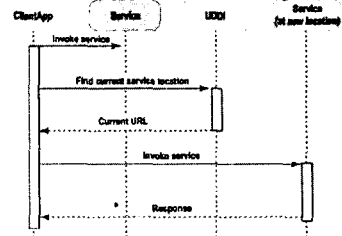
대부분의 스키마(Schema)이 이를 따르고 있다.

1. 서 론

컴퓨팅 환경에서 외부와의 연계를 통한 협업과 통합의 요구는 점점 더 커지고 있다. XML 기술이 개발되고, 웹을 통하여 상호 이질적인 환경에서도 데이터 교환이 가능하게 되면서, 협업과 통합으로의 속도는 더욱 빨라지게 되었다. XML은 단순히 데이터를 구조화 하여 표현하고 교환하는 수준을 넘어서 프로세스와 비즈니스를 공유하는 수준에까지 이르게 되었다. 웹 서비스(Web Services)는 이러한 통합의 가장 대표적인 기술로 자리 잡고 있다.

웹 서비스를 이용하여 애플리케이션이나 또 다른 서비스를 구축하는 경우가 많아질수록, 여기에 사용된 웹 서비스들이 원활하게 돌아가는 것은 구축한 애플리케이션에게 중요한 문제가 된다. 하나의 애플리케이션을 구축하는 경우라도 여러 개의 웹 서비스를 사용할 수 있는 상황에서, 이미 구축한 애플리케이션에 사용되고 있는 웹 서비스의 장애는 전체 기능에 적지 않은 영향을 미치게 된다. 즉, 웹 서비스를 이용한 애플리케이션의 안정성 있는 실행을 위해서는 사용하고 있는 웹 서비스에 장애가 발생하더라도 이를 극복할 수 있는 웹 서비스 장애 극복(Fault-Tolerant) 방법[1]이 필요하게 된다.

현재, SOAP(Simple Object Access Protocol)과 WSDL(Web Services Description Language), UDDI(Universal Description, Discovery and Integration)만을 이용한 기본적인 웹 서비스 구조에서는 이러한 장애를 극복할 수 있는 방법이 없다. 이를 해결하기 위해서는 장애 극복을 처리하는 부분이 추가되는 것이 필요한데, 이 부분은 클라이언트 측 또는 서버 측에 위치할 수 있다. 클라이언트 측에서 장애 극복을 처리하는 구조는 [그림 1]과 같으며 장애 극복을 지원하는



[그림 1] 클라이언트 측에서의 장애 극복 구조

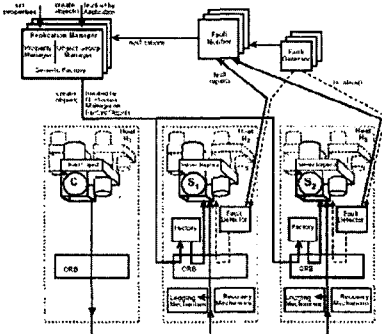
이러한 구조는 간단하기는 하지만 애플리케이션을 구축할 때마다 클라이언트 측에서 일일이 구현을 해야 하는 불편함이 따르게 된다. 따라서 본 연구에서는 클라이언트 측에서는 일반 웹 서비스를 사용하는 것과 동일하게 웹 서비스를 사용하면서도 장애 극복을 지원하는 시스템의 구조를 제안하고자 한다. 여기에서는 동일한 한 웹 서비스에 장애가 발생했을 경우 이를 대체할 수 있는 동일한 웹 서비스로 자동 전환 연결을 하는 방법을 사용한다. 제 2장에서는 관련 연구들을 살펴보고, 3장에서는 본 논문에서 제안하는 시스템의 구조와 각 요소들에 대해 살펴보고 4장에서는 본 논문의 결론을 제시한다.

2. 관련 연구

2.1. FT-CORBA[3]와 FT-SOAP[2]

장애 극복 시스템은 보통 다음의 세 가지 기본 기능을 포함한다. 첫 번째는 동일한 서비스에 대한 관리, 두 번째는 장애에 대한 관리, 세 번째는 저장 또는 복구 메커니즘이다. FT-CORBA(Fault tolerant CORBA)에서는 [그림 2]와 같이 RM(Replication Manager), Fault

Detector, Logging/Recovery Mechanism의 컴포넌트 들을 통해 이를 지원하고 있다.[2]



[그림 2] FT-CORBA의 시스템 구조 [2]

FT-CORBA를 웹 서비스로 구현한 FT-SOAP도 거의 같은 구조를 가지고 있다. 여기에서도 웹 서비스 AP를 통해 웹 서비스 그룹을 RM에 등록, 생성하며, 여기에서 어떤 웹 서비스를 Primary로 할 것인지 등도 설정을 한다. RM이 이러한 내용을 포함한 WSDL 문서를 UDDI에 등록함으로써, 클라이언트는 이 UDDI를 참조하여 확장된 WSDL 문서를 보고, 가능한 서비스에 연결을 하게 된다. WSDL에서 확장된 엘리먼트는 <WSG/> 엘리먼트로 이 엘리먼트를 통해 웹 서비스 그룹을 표시한다. 클라이언트의 FT-SOAP 엔진에서 이를 해석하여 <WSG/> 엘리먼트 안에 들어있는 웹 서비스들 중 Primary 서비스로 연결을 시도하게 된다.

2.2. UDDI의 tModel

웹 서비스는 SOAP과 WSDL만으로도 구현될 수 있으나, 누가 어떤 서비스를 제공하고 있는지에 대한 정보를 쉽게 얻기 위한 UDDI 표준이 생겨나게 되었다. UDDI 내부의 데이터는 businessEntity, businessService, bindingTemplate, tModel 등으로 구성된다. 이는 각각 서비스 제공자, 제공하는 웹 서비스, 실제 호출 정보를 나타낸다. 특히 tModel은 사용자가 잘 알지 못하는 웹 서비스와도 상호 작용할 수 있도록, 어떻게 동작하는지, 어떤 협약이나 스펙, 표준을 따르는지를 기술하기 위해 만들어졌다. tModel 정보는 크게 두 가지로 사용된다. 한 가지는 웹 서비스의 기술적인 호환성을 결정하는 것이고, 다른 한 가지는 고유 네임스페이스로서 사용되는 것이다.[4]

3. 시스템의 구조

3.1 기존 방법에서의 개선점

현존하는 대부분의 장애 극복 방안들은 서버의 장애를 감지하고, 이후의 클라이언트 요청은 백업 서버로 보내는 방식이다. 이러한 작업의 처리는 대부분 클라이언트 측에서 이루어진다. 즉, 장애 극복 기능을 사용하기 위해

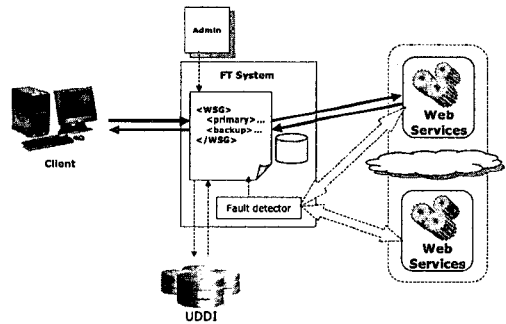
서는 클라이언트 측에서 어떤 형태로든 수정이 필요하게 된다. 본 논문에서는 클라이언트와 웹 서비스 호스트 사이에 장애 극복을 지원하는 시스템을 두어 클라이언트 투명성(Client Transparency)을 지원하도록 한다.

클라이언트 애플리케이션에서는 장애가 발생하는 경우 뿐만 아니라 응답 속도 등 다른 품질 정보를 바탕으로 서비스 자동 전환이 되는 것이 필요할 수 있다. 장애 극복 시스템을 개선하여 QoS(Quality of Service) 정보를 추가하여 품질 기반 장애 극복을 지원하도록 하였다.

마지막으로, 동일한 웹 서비스 그룹을 구성할 때 사용자가 서비스들을 다 알고 있어야 하는 문제가 있다. UDDI를 검색하여 정보를 얻을 수도 있지만, 불편함이 따르게 된다. 본 연구에서는 기존 UDDI에 들어있는 tModel 정보 등을 이용하여 이를 자동으로 지원할 수 있도록 한다.

3.2. 시나리오

이 논문에서 제안하는 품질 기반 장애 극복 시스템의 구조는 [그림 3]과 같다. 이 시스템은 크게 초기 설정 단계와 런타임 단계로 나누어 볼 수 있다.



[그림 3] 품질 기반 장애 극복 시스템의 구조

초기 설정 단계에서는 클라이언트 측에서 Admin 애플리케이션을 이용하여 원하는 웹 서비스의 서비스 포인트를 등록한다. 클라이언트는 동일한 서비스를 제공하는 웹 서비스들을 직접 등록할 수도 있다. 이때에 어느 웹 서비스를 주된 웹 서비스로 이용할 것인지, 그리고 어떤 품질 정보를 어느 수준에서 이용할 것인지 하는 품질 정보를 기록한다. 등록을 마치고 나면 장애 극복 시스템은 UDDI를 통해 클라이언트가 등록한 웹 서비스와 tModel key가 같은 값을 가진 웹 서비스를 찾아내어 네임스페이스를 검사, 동일한 서비스 그룹에 추가를 한다. 이를 통해 클라이언트 입장에서는 동일한 웹 서비스들 모두에 대한 정보를 가지고 있지 않아도 서비스 그룹 등록을 하는 것이 가능하게 된다.

웹 서비스를 제공하는 측에서도 백업용 웹 서비스 시스템을 구축해놓은 경우 tModel key를 등록하기만 하면 사용자에게 신뢰성 있는 서비스를 간단히 지원할 수 있

게 된다. 즉, 네트워크 등의 문제로 시스템에 장애가 발생하는 경우를 대비하여 도메인 외부에 백업 시스템을 구축해놓고 백업 시스템을 가동시킬 수 있게 된다.

런타임 단계에서는 Fault Detector에 의해 전송 속도가 장애 여부 등 웹 서비스의 상태를 체크하여 동일 서비스 그룹에 포함되어 있는 웹 서비스들의 품질을 기술한다. 시스템은 이를 바탕으로 어느 웹 서비스가 Primary가 될 것인지를 결정하며, Primary 웹 서비스에 연결을 하게 된다. 클라이언트가 보낸 SOAP 메시지는 시스템 내부의 저장 공간에 저장해 두어, 서비스 진행 중에 장애가 발생했을 때 SOAP 메시지를 재 전송하는데 사용한다.

추가적으로 고려되는 사항은, 서비스 인터페이스가 동일하지만 SOAP 인코딩 방식 등의 차이로 클라이언트로부터 온 SOAP 메시지를 그대로 처리할 수 없는 경우이다. 이러한 경우는 SOAP 메시지를 백업 시스템으로 전달하는 것 뿐 아니라, 메시지를 디코딩하여 백업 시스템에 맞는 형식으로 다시 디코딩을 해야 하기 때문에, 인코딩, 디코딩에 걸리는 시간으로 인하여 응답 속도 등은 보장할 수 없다. 하지만 본래 서비스의 장애를 해결하는 짧은 시간동안 응답 시간이 늘어나는 것을 제외하고는, 웹 서비스를 정상적으로 이용하는 것이 가능하다.

3.3 품질 요소

웹 서비스가 발전하면서 많은 관련 기술들이 나오고 있지만, 서비스 품질에 대한 기술은 아직 성숙되지 않은 상태이다. 기존 연구들[5][6] 중에서 본 논문이 제안하는 시스템에 사용될 수 있는 품질 요소를 뽑아내면 다음과 같다.

본 논문에서 사용하는 품질 요소는 크게 3가지로 나뉜다. 첫째는, 현재의 웹 서비스의 품질에 대한 실시간 정보, 둘째는, 사용자들이 해당 웹 서비스를 이용한 정보를 바탕으로 뽑아낸 통계 정보, 마지막으로, 해당 웹 서비스가 어떤 관련 표준(보안 등)을 지키고 있는지에 대한 정적인 정보이다.

실시간 정보에 해당하는 것으로는 응답 시간(Response Time)과 이용가능성이 포함된다. 응답 시간은 요청을 보내고 응답을 받는데 걸린 시간을 의미한다. 이를 식으로 나타내면 (1)과 같다. 이용가능성은 장애 극복의 가장 중요한 요소인 서비스가 잘 제공되고 있는가에 대한 여부를 나타낸다.

$$Response\ Time = \frac{TotalMonitoring\ Time}{TotalTransactionCount} \quad (1)$$

통계 정보에 해당하는 것으로는 신뢰성(Reliability)과 선호도(Reputation)가 있다. 신뢰성은 해당 서비스가 얼마나 안정적으로 동작 했었는가에 대한 통계 기록으로 일정 시간동안의 서비스 가용 시간을 바탕으로 측정한다. 이를 식으로 나타내면 (2)와 같다. 선호도는 동일한 서비스들 가운데 해당 서비스가 사용되는 비율이 어떠한지를 측정한다.

$$Reliability = 1 - \frac{ErrorEventTimeSeries}{TotalMonitoringTime} \quad (2)$$

정적 정보 중 고려할 표준은 대표적으로 보안 등의 경우가 있다. 예를 들어 동일 웹 서비스 그룹에 속해있는 웹 서비스들은 WS-Security를 지원해야 한다는 것을 기술할 수 있다. 이러한 표준 요소들은 네임스페이스를 통해 찾아내게 된다.

4. 결론 및 향후 연구

본 논문에서는 웹 서비스를 이용하여 애플리케이션을 구축할 때, 사용된 웹 서비스들에 장애가 발생했을 경우 이를 해결해 줄 수 있는 장애 극복 시스템을 제안하고 있다. 기존의 웹 서비스 장애 극복이 클라이언트나 엔진에서의 수정이 필요했던 반면에 본 논문에서는 클라이언트 측에서 기존의 시스템을 그대로 사용하면서도 장애 극복 기능을 지원받을 수 있도록 하였다. 여기에 품질에 대한 고려 사항을 추가하고 동일 웹 서비스를 찾는 방법들을 개선하여 전체적인 편의성을 증가시키는 방법을 제안하였다.

웹 서비스는 다양한 표준과 기술을 갖고 있기 때문에 같은 내용의 서비스를 제공하더라도 제공하는 웹 서비스의 구조가 완전히 동일하기는 어렵다. 그러나 웹 서비스 호출뿐만 아니라 비즈니스 등 진정한 협업을 위해서는 시멘틱 정보 등을 이용하여 같은 내용의 서비스를 제공하는 웹 서비스를 찾아내는 연구와 장애극복 처리시간을 향상시키는 연구가 더 필요할 것이다.

참고 문헌

- [1] Navid Aghdaie, Yuval Tamir "Client-Transparent Fault-Tolerant Web Service", Performance, Computing, and Communications, 2001. IEEE International Conference on, 4-6 April 2001, pp.209 - 216
- [2] Deron Liang, Chen-Liang Fang, Chyowha Chen, Fengyi Lin, "Fault tolerant web service", Proceedings of the Tenth Asia-Pacific Software Engineering Conference 2003(APSEC'03)
- [3] OMG, Common Object Request Broker Architecture(CORBA): Core Specification, v 3.0.3, OMG Technical Committee Document formal/2004-03-12, 2004
- [4] UDDI v3, "http://uddi.org/pubs/uddi-v3.htm"
- [5] 웹 서비스 품질관리 동향 및 도입전략 연구, 한국전산원, 2003.12
- [6] SeokHyun Yoon, DongJoon Kim, Sangyong Han, "WS-QDL containing static, dynamic, and statistical factors of Web services quality", 2004, IEEE International Conference on Web Service, July 6-9, pp.808-809