

웹서비스 기반의 상태정보 관리 시스템의 설계 및 구현

박현길^o 문종배 김명호

충실파워 컴퓨터학과

{hgpark^o, comdoct}@ss.ssu.ac.kr, kmh@ssu.ac.kr

Design and Implementation of State Information Management System based on Web Services

HyunGil Park^o JongBae Moon MyungHo Kim

Department of Computer Science, Soongsil University

요약

그리드 구축 도구인 Globus Toolkit을 이용한 그리드 서비스의 상태정보 관리의 문제점을 해결한 새로운 그리드 서비스 표준인 WS-Resource Framework(WSRF)이 제안되었다. 그러나, WSRF에서는 웹서비스와 상태정보 관리 팩토리가 강하게 결합되어 있어 서비스 개발의 복잡도가 증가하고, 확장성이 떨어진다. 또한, 웹서비스에서 상태정보 관리 팩토리를 유연하게 선택할 수 없는 문제점이 있다. 본 논문에서는 WSRF의 문제점을 해결한 웹서비스 기반의 상태정보 관리 시스템을 제안한다. 웹서비스 기반의 상태정보 관리 시스템은 웹서비스와 상태정보를 관리하는 팩토리 서비스를 약하게 결합시키기 위해 웹서비스와 상태정보 팩토리를 분리시키고, 모든 웹서비스에서 공통된 인터페이스를 통해 상태정보 관리 팩토리를 사용할 수 있도록 하여, 복잡도와 확장성을 해결한다. 또한, 상태정보 이주 기능을 제공하여 웹서비스가 유연하게 상태정보 관리 팩토리를 선택할 수 있도록 한다.

1. 서론

웹서비스에서 상태정보는 웹서비스가 여러 번 호출되어도 유지되어야 하고, 여러 클라이언트를 위해 유지되어야 하는 내부 데이터 또는 속성이다[1]. 또한, 여러 웹서비스들 사이에 의존성을 유지하기 위한 내부 데이터이다. 일반적으로 웹서비스에서는 애플리케이션 실행 중에 유지되고 사용된다. 웹서비스의 오류가 발생하면 모든 상태정보는 사라지게 된다. 그러나 상태정보를 유지하고 있으면 오류가 발생한 시점부터 다시 서비스를 제공할 수 있게 된다. 이런 이유로 상태정보의 유지는 중요하다. Globus Toolkit 3(GT3)[2]와 WS-Resource Framework(WSRF)[3]에서는 상태정보 관리를 위한 메커니즘을 제공한다.

GT3의 그리드 서비스는 내부적으로 서비스의 상태정보를 가지고 있는 인스턴스의 개념과 서비스 인스턴스를 관리하는 서비스 팩토리의 개념을 도입하여 웹서비스를 확장시킨다. 그러나, 그리드 서비스의 상태정보 개념은 몇 가지의 문제점을 가지고 있다. GT3 환경에서 새로운 그리드 서비스를 배포할 때, GT3의 서비스 팩토리인 서비스 컨테이너를 재시작해야 한다. 서비스 컨테이너를 재시작하면 수행중인 모든 그리드 서비스의 실행이 중지되기 때문에, 상태정보도 없어지게 된다. 또한, 웹서비스의 표준을 따르는 서비스와 그리드 서비스의 상호운용성이 떨어지는 문제점이 있다.

이와 같은 GT3의 문제점을 해결하고, 웹서비스와 그리드 서비스의 상호 운용성을 위해 새로운 그리드 서비스 표준인 WSRF가 제안되었다. WSRF의 상태정보 관리는 웹서비스와 상태정보를 관리하는 상태정보 관리 팩토리를 분리하여 웹서비스의 오류가 발생해도 상태정보를 유지할 수 있다. 상태정보의 관리를 위해 웹서비스와 상태정보 관리 팩토리를 분리시켜 웹서비스의 상태에 관계없이 상태정보를 유지할 수 있다. 그러나 웹서비스와 상태정보 관리 팩토리가 강하게 결합

(tightly-coupled)되어 있어 서비스 개발의 복잡도가 증가하고, 상태정보 관리 팩토리의 확장성이 떨어진다. 또한, 웹서비스에서 상태정보 관리 팩토리를 유연하게 선택할 수 없는 문제점이 있다[4].

본 논문에서는 WSRF의 문제점을 해결한 새로운 형태의 웹서비스 기반의 상태정보 시스템을 제안한다. 서비스 팩토리의 확장성을 높이고 새로운 서비스를 만드는 복잡도를 낮추기 위해 상태정보 관리 팩토리의 인터페이스를 일반화시켜 모든 웹서비스가 공통된 인터페이스를 이용하여 상태정보 관리 팩토리를 사용할 수 있다. 또한 상태정보 이주(migration) 기능을 제공하여 상태정보 관리 팩토리를 유연하게 선택하여 사용할 수 있다.

본 논문의 구성을 다음과 같다. 2장에서는 그리드 서비스와 WSRF의 상태정보 관리의 문제점을 기술하고, 3장에서는 제안하는 상태정보 관리 시스템의 구성과 특성을 기술한다. 4장에서는 상태정보 관리 시스템의 구현에 대하여 기술한다. 마지막으로 5장에서는 결론 및 향후 연구과제를 기술한다.

2. 관련연구

2.1 그리드 서비스

GT3는 OGSA[5]와 OGSI[6]를 기반으로 구현된 그리드 서비스가 적용되었다. OGSA는 그리드 서비스를 제공하기 위한 프레임워크를 정의하며, OGSI는 그리드 서비스의 기본 서비스를 정의한다. 그리드 서비스는 내부적으로 서비스의 상태정보를 가지고 있는 인스턴스의 개념과 서비스 인스턴스를 관리하는 서비스 팩토리의 개념을 도입하여 웹서비스를 확장시켰다.

그리드 서비스는 새로운 서비스가 서비스 팩토리에 등록, 삭제될 때, 서비스 팩토리 기능을 하는 서비스 컨테이너가 재시작되어야 한다. 이 때, 서비스 컨테이너에 존재하는 모든 서

비스 인스턴스가 사라지게 되는 문제가 발생한다.

2.2 WS-Resource Framework(WSRF)

WSRF는 OGSI를 정제, 발전시켜 웹서비스의 상태정보를 관리하기 위한 표준을 정의하고 애플리케이션에서 표준적인 방법으로 상태정보를 사용할 수 있도록 하기 위한 프레임워크이다[3,7].

WSRF는 웹서비스와 상태정보를 관리하는 서비스 팩토리를 분리시켜, 새로운 서비스의 등록, 삭제할 때에도 다른 서비스에 영향을 미치지 않는다. WSRF의 상태정보는 Stateful Resource로 나타내며 다음과 같은 특징을 갖는다.

- 상태정보는 XML 형식으로 구성된다.
- 생명 주기를 가진다.
- 하나 이상의 웹서비스에서 사용될 수 있다[8].

WSRF는 웹서비스로부터 상태정보 관리를 분리했지만 여전히 몇 가지의 문제점이 있다. 첫째, 웹서비스에 특화된 서비스 팩토리가 존재하여 새로운 서비스의 개발 복잡도를 증가시킨다. 둘째, 웹서비스와 서비스 팩토리가 강하게 결합되어 있기 때문에 확장성을 제한한다. 셋째, 상태정보 저장소의 위치를 유연하게 선택하지 못하는 문제점이 있다[4]. 그래서, 서비스의 개발 복잡도를 낮추고, 서비스 팩토리 확장성을 가지는 시스템이 필요하다.

3. 제안하는 상태정보 관리 시스템

본 장에서는 WSRF의 상태정보 관리의 단점을 해결한 웹서비스에 기반을 둔 새로운 상태정보 시스템을 제안한다. 상태정보 시스템은 상태정보 관리 팩토리의 인터페이스를 일반화 시켜 모든 웹서비스가 공통된 인터페이스를 통해 상태정보 관리 팩토리를 사용한다. 이로 인해 상태정보 관리 팩토리의 확장성을 높이고 새로운 서비스를 만드는 복잡도를 낮춘다. 또한, 상태정보의 이주를 제공하여 상태정보 관리 팩토리를 유연하게 선택할 수 있다.

3.1 상태정보 관리 시스템의 구조

WSRF의 문제점을 해결하기 위한 상태정보 관리 시스템 모델은 그림 1과 같은 구성을 갖는다.

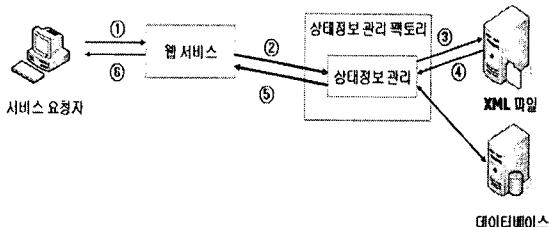


그림 1 상태정보 관리 시스템 구조

웹서비스의 상태정보를 상태정보 관리 팩토리가 관리를 하며, 상태정보 저장소는 XML 파일, 데이터베이스, 상태 서버[9]등 여러 모델을 사용한다. (1)-(2) 서비스 요청자가 웹서비스에 서비스를 요청하면, 웹서비스는 상태정보 관리 팩토리를 통해 상태정보 저장소를 요청한다. (3)-(4) 상태정보 관리 팩토리는 상태정보 저장소의 모델에 따라 웹서비스 인스턴스의 상태정보를 저장한다. (5) 상태정보의 인스턴스를 구분하기 위한 Global User ID(GUID)를 생성 후 웹서비스에 GUID를 전달한

다. (6) 웹서비스는 서비스 요청자에게 GUID를 전달한다.

3.2 상태정보 관리 시스템의 특성

상태정보 관리 시스템은 다음과 같은 특성을 갖는다.

- 웹서비스와 상태정보 관리 팩토리는 분리되어 있고, 약하게 결합(loosely-coupled)되어 있다.
- 상태정보 관리 팩토리는 웹서비스와 같은 호스트 또는 다른 호스트에 있을 수 있을 수 있다.
- 상태정보 관리 팩토리는 상태정보 관리를 위한 인터페이스를 제공하며, 모든 웹서비스는 상태정보 관리 팩토리의 인터페이스를 통해 상태정보를 관리한다.
- 상태정보는 XML 형식을 사용하여 생명주기를 가진다.
- 하나 이상의 웹서비스에서 사용할 수 있다.
- 상태정보 저장소의 여러 모델(파일, 데이터베이스, 상태서버 등)을 지원한다.
- 상태정보 생성 시 또는 웹서비스 인스턴스의 실행 중에 상태정보 관리 팩토리를 지정할 수 있다.

4. 설계 및 구현

본 논문에서는 상태정보 관리 시스템을 마이크로소프트닷넷 웹서비스(Microsoft.NET Web Services)와 마이크로소프트닷넷 리모팅(Microsoft.NET Remoting) 기반으로 구현한다. 상태정보 관리 팩토리는 닷넷 리모팅 서버로 구현하고, 상태정보를 XML 파일 또는 데이터베이스로 저장하는 리모팅 객체를 가지고 있도록 한다. 웹서비스는 상태정보 관리 팩토리를 사용하기 위한 닷넷 리모팅 클라이언트를 포함하는 C# 기반의 마이크로소프트 ASP.NET 웹서비스로 구현한다. 웹서버는 마이크로소프트 IIS 5.1을 사용한다. 상태정보 관리 시스템의 구조는 그림 2와 같다.

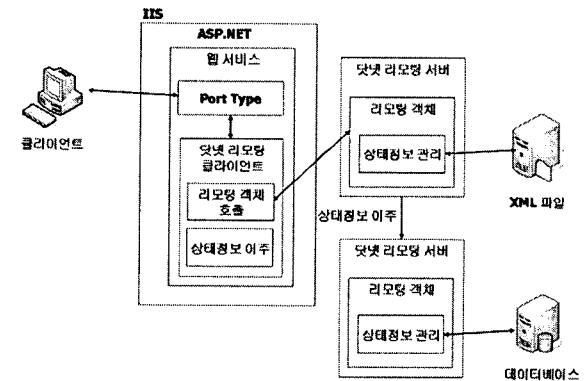


그림 2 웹서비스 기반 상태정보 관리 시스템 구조

클라이언트가 웹서비스의 PortType을 호출하면 닷넷 리모팅 클라이언트는 닷넷 리모팅 서버의 리모팅 객체를 호출한다. 리모팅 객체는 상태정보를 XML 형식으로 상태정보 저장소에 저장하고 GUID를 클라이언트에 전달한다. 클라이언트는 GUID를 통해서 상태정보의 저장, 복원 등의 관리를 할 수 있게 된다.

상태정보 관리 팩토리인 닷넷 리모팅 서버는 리모팅 객체를 가지고 상태정보를 관리하기 때문에, 리모팅 객체의 구현에 따라 여러 가지 형태의 상태정보 저장소 모델을 사용할 수 있다. 이로 인해, 상태정보 관리 팩토리의 확장성이 높아지게 된

다. 상태정보의 이주는 닷넷 리모팅 클라이언트에서 이루어진다. 클라이언트가 다른 상태정보 관리 팩토리로 상태정보 이주를 요청하면, 닷넷 리모팅 클라이언트는 상태정보 관리 팩토리에 접속하여 상태정보를 가져온 후 클라이언트가 요청한 상태정보 관리 팩토리에 저장한다. 그 후 새로운 GUID를 클라이언트에 전달하고 클라이언트는 새로운 GUID를 이용하여 상태정보를 관리할 수 있게 된다.

웹서비스의 상태 정보를 관리하는 상태정보 관리 팩토리 리모팅 객체의 인터페이스는 그림 3과 같다.

```
public class Factory : System.MarshalByRefObject
{
    public bool CreateResource(string guid);
    public bool SaveResource(string resource, string guid);
    public string GetResource(string guid);
    public bool DestoryResource(string guid);
}
```

그림 3 상태정보 관리 리모팅 객체의 인터페이스

모든 상태정보 관리 팩토리의 리모팅 객체는 그림 3과 같은 인터페이스를 가지고 있어야 하며, 상태정보 저장소 모델에 따라 구현이 다르게 된다.

5. 실험

상태정보 관리 시스템의 실험을 위해 계산기 웹서비스를 구현하였다. 계산기 서비스는 덧셈, 뺄셈의 기능을 가지고 있고, 상태정보로 계산된 결과 값을 가지고 있다. 그리고 상태정보 관리 팩토리의 리모팅 객체는 파일을 이용하여 상태정보를 저장한다.

클라이언트가 계산기 웹서비스의 PortType을 호출하면 서비스 팩토리는 GUID를 생성하고 계산된 값을 저장한다. 그림 4에서는 GUID를 파일명으로 하는 상태정보 XML 파일을 보여준다. 상태정보는 덧셈, 뺄셈 같은 PortType의 호출시마다 갱신된다. 상태정보의 삭제는 클라이언트에서 요청하거나, 상태정보를 사용하는 클라이언트가 없을 때 삭제된다.

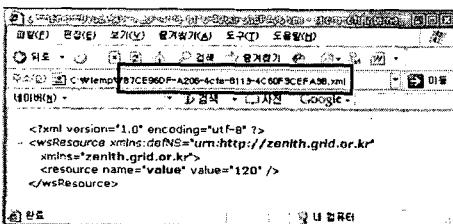


그림 4 상태정보 XML 파일

웹서비스의 오류가 발생하면, 클라이언트는 웹서비스를 다시 요청한다. 웹서비스는 클라이언트가 전달한 GUID를 확인하여 상태정보 관리 팩토리에 상태정보 복구를 요청한다. 서비스 팩토리는 GUID를 파일명으로 하는 XML 파일을 불러와서 웹서비스 내부의 변수에 값을 저장한다.

클라이언트가 상태정보의 이주를 요청하면, 닷넷 리모팅 클라이언트에서 GUID를 이용하여 상태정보 관리 팩토리의 리모팅 객체를 호출, 상태정보를 가져온다. 그리고 새로운 서비스 팩토리에 그림 5와 같이 상태정보를 저장하고, 새로운 GUID를 클라이언트에 전달한다. 클라이언트는 새로운 GUID를 이용하여 상태정보를 관리할 수 있게 된다.

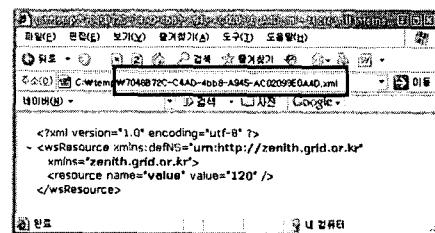


그림 7 이주된 상태정보 XML 파일

6. 결론 및 향후 연구과제

본 논문에서는 WSRF의 문제점을 제시하고 이를 해결한 웹서비스 기반의 상태정보 관리 시스템을 제안하였다. WSRF의 문제점인 웹서비스에 특화된 상태정보 관리 팩토리의 존재로 인한 새로운 서비스의 개발 복잡도의 증가, 웹서비스와 상태정보 관리 팩토리의 강한 결합으로 인한 확장성의 문제를 해결하기 위해 일반화된 인터페이스를 가지는 상태정보 관리 팩토리를 제공한다. 웹서비스와 상태정보 관리 팩토리를 분리시키고, 팩토리의 공통된 인터페이스를 사용해서 상태정보를 관리하도록 하여 문제를 해결하였다. 또한, 상태정보 저장소의 위치를 유연하게 선택하기 위해 상태정보 이주를 이용하여 해결하였다.

향후 여러 웹서비스가 하나의 상태정보를 사용할 때, 상태정보가 변경되면 모든 웹서비스에 상태정보 변경을 알려주는 WSRF의 WS-Resource Notification의 구현과 그리드 서비스, WSRF 서비스의 비교 분석이 필요하다.

7. 참고문헌

1. Hidayatulla H.Shaikh, Sonny Fulkerson, Implement and access stateful web services using websphere studio, 2004
2. Globus Toolkit 3, <http://www-unix.globus.org/toolkit/>
3. Karl Czajkowski, Don Ferguson, Ian Foster, Jeffrey Frey, Steve Graham, Igor Sedukhin, David Snelling, Steve Tuecke, William Vambenepe, WSRF : The ws-resource framework, 2004
4. Yong Xie,Yong-Meng Tei, State Management Issues and Grid Services, 2004
5. I. Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002
6. S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, D. Snelling, Open Grid Services Infrastructure (OGSI) Version 1.0, Grid Forum Draft Recommendation, 2003
7. Karl Czajkowski, Don Ferguson, Ian Foster, Jeffrey Frey, Steve Graham, Tom Maguire, David Snelling, Steve Tuecke, From Open Grid Services Infrastructure to WS-Resource Framework:Refactoring & Evolution,2004
8. Vladimir Silva, Globus Toolkit 4 Early Access : WSRF, 2004
9. Xiang Sona, Namgeun Jeong, Phillip W.Hutto, Umakishore Ramachandran, James M.Rehg, State Management in Web Services, 2004