

웹 서비스를 적용한 그리드 서비스

승실대학교 최재영* · 이상근

1. 그리드란?

90년대 중반에 등장한 그리드는 기존의 병렬 및 분산 컴퓨팅이 가지고 있는 성능과 확장성, 그리고 지리적인 한계를 극복하여, 네트워크로 상호연동된 컴퓨팅 자원들을, 각 자원을 보유하고 있는 기관의 자율성을 침해하지 않으면서, 온라인상에서 안전하게 사용할 수 있는 환경을 말한다[1]. 기존의 병렬 및 분산 시스템 분야는 작은 컴퓨팅 자원을 여러 개 연결하여 보다 큰 컴퓨팅 성능을 제공하기 위한 방법으로 발전해 왔지만, 범용성이나 단일 시스템 이미지 등을 제공하기에는 많은 한계가 있었다. 그리드 컴퓨팅은 병렬/분산 컴퓨팅의 연장선상에서 이러한 한계를 극복하기 위하여 태동하고 발전해 온 기술이다.

“그리드”라는 용어는 전력망 (Electric Power Grid)에서 유래하였다. 전력망에는 수만 개의 발전기와 수십억 개의 컨센트를 연결하여 안정적이며 저렴하고 효율적으로 단일한 개체로 통합되어 있다. 컴퓨팅 자원에 대해서도 이러한 통합을 가능하게 하려는 노력이 바로 그리드 컴퓨팅이다[2]. 또한 전력망이 전기를 언제 어디서나 손쉽게 사용할 수 있는 수단을 제공함으로써 인류에 지대한 영향을 미쳤듯이, 그리드는 정보기술의 기반을 제공하는 컴퓨팅 프로세서, 데이터베이스, 센서와 같은 모든 요소들을 진정한 협력 도구로서 공유하게 함으로써, 인류에게 새로운 응용의 장을 제공할 것이다[3].

그리드 컴퓨팅에 대한 연구는 GGF (Global Grid Forum)[4]의 주도로 진행되고 있다. GGF는 그리드 컴퓨팅의 표준화를 위한 목적으로 그리드 커뮤니티에 의해 시작된 포럼으로, 산업계와 연구소, 대학으로부터 수천명 이상이 참여하고 있다. GGF의 주된 목적은 기술 명세, 사용자 경험, 구현 가이드라인 등의 모범 사례들을 생성하고 문서화하여 그리드 기술의 개발, 설치, 구현 및 응용을 장려하고 지원하는 것이다.

그리드 컴퓨팅은 주로 과학 포털, 분산 컴퓨팅, 대용량 데이터 분석 등에 응용되고 있다. 그리드를 적용한 과학 포털은 웹 브라우저나 혹은 단순하면서 쉽게 내려받을 수 있는 신플라이언트를 이용하여 과학 연구에 사용되는 복잡한 패키지들을 손쉽게 사용할 수 있도록 하였다. 패키지들은 그리드 안에 있는 적절한 컴퓨터에서 원격으로 스스로 작동될 수 있다. 이러한 과학 포털들은 주로 생물학, 핵융합, 계산화학 분야에서 주로 이용되며, 대표적인 연구로는 Biology Workbench Portal [5], myGrid[6], NASA Launchpad Portal[7] 등이 있다.

고성능의 PC나 워크스테이션과 네트워크를 묶어 하나의 거대한 컴퓨팅 자원을 제공할 수 있다. 대표적인 분산 컴퓨팅으로 30,000대 이상의 컴퓨터를 연결하여 AIDS 치료제의 후보 물질을 분석하는 미국 Entropia사의 FightAIDSAtHome과 같은 시스템이 있다[8]. 또한 2001년에는 미국과 이탈리아에 있는 수학자들이 Nug30이라는 최적해 문제를 해결하려고 많은 컴퓨터들을 연결하여, 당시 가장 성능이 좋은 워크스테이션에서 7년 이상 걸려서 해결될 수 있는 문제를 일주일 만에 해결하였다[9].

한편 많은 과학적인 연구 주제들은 엄청난 분량의 데이터 분석을 요구한다. 이러한 연구에는 분산 컴퓨팅과 저장 자원들을 함께 묶어서 사용하며, 그러한 데이터를 분석하는 과정은 본질적으로 병렬성을 가지고 있으므로 분산된 자원들을 효과적으로 이용할 수 있다. 예를 들어, 하드론 입자 가속기나 고에너지 물리학에서는 수 페타(peta=10¹⁵) 바이트의 데이터를 분석해야 하는데, 임시 결과를 저장하여 계산하는데만 수 만개의 프로세서와 수백 테라(tera=10¹²) 바이트의 디스크 공간을 필요로 한다. 그러한 막대한 자원을 한 곳으로 모으는 것은 기술적으로나 정략적으로 실현성이 없다. 그러나 연구에 참여하고 있는 기관과 국가가 보유하고 있는 자원들을 합치면 필요한 자원을 제공할 수 있다. 그러한 연구팀들은 단순히 컴퓨터와 저장 공간만을 공유할 뿐 아

* 종신회원

나라, 분석하는 과정과 결과를 공유할 수도 있다. 대표적인 연구로는 Griphyn (Grid Physics Network)[10], NEES (Network for Earthquake Engineering Simulation) Grid[11] 등이 있다.

망원경, 싱크로트론, 전자현미경과 같은 과학 장비 등은 계속해서 일괄적으로 작업을 진행하면서 원시 데이터를 생성해낸다. 그러나 의사(quasi) 실시간적인 데이터의 분석은 장비의 효율성을 크게 향상시킬 수 있다. 예를 들어, 천문학자가 태양 표면의 폭발을 천체 망원경으로 관찰한다면, 생성된 데이터를 처리하고 분석하는데 많은 컴퓨팅 자원을 필요로 한다. 하지만 폭발은 순간적이고 간헐적이므로 계속해서 알고리즘을 수행시킬 필요는 없다. 만일 천문학자가 필요한 때에 막대한 컴퓨팅 자원과 효과적인 알고리즘을 주문형(On-demand)으로 사용할 수 있다면, 태양의 표면에서 폭발이 일어났을 때 자동적으로 탐지할 수 있는 기술을 개발할 수 있을 것이다. 그러한 대표적인 연구로는 International Virtual Data Grid Laboratory[12]와 Particle Physics Data Grid [13]가 있다.

2000년대 초반부터 미국, 영국, EU, 일본, 중국 등 각 나라에서는 연구인력, 시설장비, 연구 소프트웨어 등을 하나로 통합하여 동시 활용할 수 있는 차세대 과학기술 연구 개발 환경인 e-Science 프로그램을 수립하여 추진하고 있다[14]. 첨단과학기술 및 차세대 성장동력 분야를 중심으로 e-Science를 집중적으로 도입하여 응용하는 한편, e-Science를 기반으로 e-Business, e-Learning, e-Health 등으로 확대 적용함으로써 과학기술의 향상을 선도하고 있다. e-Science의 핵심에는 그리드 기술이 자리하고 있다.

2. Globus Toolkit

Globus Toolkit[15]은 그리드 환경을 구축하는데 사용되는 대표적인 그리드 구축 도구로, 그리드 컴퓨팅에서 사실상의(de facto) 표준으로 인정받고 있다. Globus Toolkit은 슈퍼컴퓨터들을 연결하여 활용하기 위한 목적으로 미국 아르곤 연구소, 남가주 대학교의 정보과학센터(ISI), 시카고 대학 등이 참가하여, 1996년부터 개발이 진행되었다.

Globus Toolkit은 그리드와 그리드 어플리케이션을 지원하는 서비스 및 소프트웨어 라이브러리로서 보안, 정보 발견, 자원 관리, 데이터 관리, 통신 오류 감지, 이식성 등 그리드에서 필요한 서비스들을 독립적인 요소로 제공한다. Globus Toolkit에서 제공하는 주요 서비스로는 TLS 기반의 인증 기능인 GSI (Grid Security Infrastructure), 원격에서 작업을 제출할 수 있게 해주는

GRAM, 대용량의 데이터를 빠르고 안정적으로 제공해주는 GridFTP, 시스템의 정보를 검색할 수 있는 MDS 등이 있다. 1998년에 처음으로 공개된 Globus Toolkit 1.0 버전에서는 분산 컴퓨팅 기능을 손쉽게 구현할 수 있는 기본적인 기능들을 제공하였으며, 2002년에 공개된 Globus Toolkit 2.0 버전(GT2)은 대규모 데이터를 처리할 수 있도록 성능과 안정성을 크게 개선하였다.

W3C (World Wide Web Consortium)의 정의에 따르면, 웹 서비스는 웹에서 컴퓨터간의 상호작용이 가능하도록 만들어진 컴퓨터 시스템이다[16]. 웹 서비스는 WSDL이라는 컴퓨터가 처리할 수 있는 형식으로 기술된 인터페이스를 가지며, 여러 컴퓨터들이 XML, SOAP, HTTP 등의 표준화된 방식을 사용하여 메시지를 전달하고 상호작용하면서 WSDL로 기술된 서비스를 처리한다.

그리드 서비스는 그리드에서 제공하던 서비스를 웹 서비스에서 사용하는 인터페이스를 이용하여 웹 서비스 프레임워크에 통합시킨 개념이다[17]. 웹 서비스를 이용하면 이질적인 환경에서 동작하는 다양한 그리드 서비스들을 표준화하고 통합할 수 있을 뿐만 아니라, 기존에 웹 서비스로 구현된 많은 다른 서비스들을 그리드 환경에서 사용할 수 있는 이점이 있다. 웹 서비스에 관한 원천 기술을 보유하고 있는 IBM이 주도적으로 Globus의 개발에 참여하면서, 2003년에 공개된 Globus Toolkit 3.0 버전(GT3)부터 그리드 서비스의 개념이 적용되기 시작하였다. IBM은 Globus Toolkit을 무료로 사용할 수 있는 오픈소스 소프트웨어로 지원한다고 발표하여, 그리드 서비스가 보다 활성화될 수 있는 전기를 마련하였다.

GT3은 OGSA (Open Grid System Architecture)[17]와 OGSI (Open Grid System Infrastructure)[16]를 기반으로 구현되었는데, OGSA는 그리드 서비스를 제공하기 위한 프레임워크를 정의하며, OGSI는 그리드 서비스의 생명 주기 관리 등의 기반 서비스를 정의한다. OGSA와 OGSI의 Open이라는 단어가 의미하듯이, 그리드 서비스는 웹 서비스를 이용하여 그리드 기술을 보다 개방된 형태로 발전시킨 것이다. 3.0 이전 버전의 Globus Toolkit에서 유닉스 데몬으로 제공하였던 GRAM, MDS 등의 서비스를 그리드 서비스로 제공함으로써 보다 용이하게 서비스를 통합할 수 있으며, 또한 보다 유연한 시스템을 구성할 수 있다. OGSA와 OGSI에 대해서는 4장에서 자세히 설명한다.

현재 개발이 진행되고 있는 Globus Toolkit 4.0 버전에서는 그리드 서비스를 웹 서비스에 통합한 WSRF (Web Service Resource Framework)[18]를 기반으로 하고 있다. 웹 서비스와 그리드 서비스의 상호운용성

을 제공하기 위해서는 광범위한 웹 서비스와 그리드 서비스의 표준화가 필요한데, 이를 위한 프레임워크가 바로 WSRF이다. 그림 1에서 보듯이 고성능 컴퓨팅 및 자원의 공유를 목적으로 하는 그리드와 상업적인 응용을 목적으로 하는 웹이 WSRF라는 하나의 통합된 프레임워크 위에서 지원되며[19], WSRF에서 바로 그리드 기능을 지원함으로써 보다 유연하게 그리드 시스템을 구성할 수 있다. 자세한 기술적인 내용은 4장에서 기술한다. 2004년 9월 현재에 안정된 최신 버전은 OGSA를 기반으로 하고 있는 Globus Toolkit 3.2.1 버전이다. Globus Toolkit 4.0 베타 버전은 2004년 후반기에 출시될 예정에 있고, 정식 버전은 2005년에 출시될 예정이다.

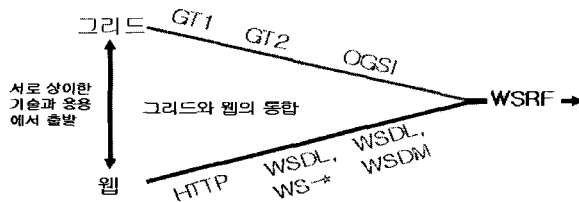


그림 1 그리드 기술과 웹 서비스의 통합

3. Globus 2.0의 그리드 기술

그리드는 분산 컴퓨팅을 위한 인프라의 개념에서 연구가 시작되었으며, 기존의 인터넷을 대체할 차세대 네트워크 인프라로서도 많은 관심을 끌고 있다. 따라서 그리드 시스템의 계층 구조는 인터넷 프로토콜의 계층 구조와 비교되는 경우가 많다. 그림 2는 그리드 프로토콜의 계층 구조와 인터넷 프로토콜의 계층 구조를 비교한 그림이다[1].

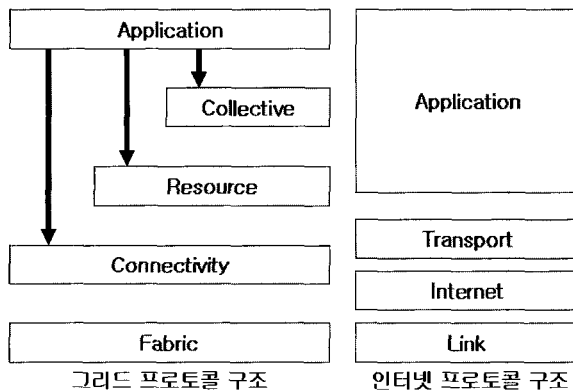


그림 2 그리드 프로토콜과 인터넷 프로토콜의 비교

각각의 계층을 살펴보면 Fabric 계층은 자원을 컴퓨팅 자원, 스토리지 자원, 네트워크 자원 등의 자원으로 추상화하고, 자원에 접근하기 위한 저수준 기능을 제공한다. Connectivity 계층은 전송, 라우팅 등의 통신 기능과 인증 기능을 제공한다. Resource 계층은 현재 설

정, 사용자 정책 등 자원에 관한 정보를 제공하고, 자원을 관리할 수 있는 기능을 제공한다. Collective 계층은 스케줄링, 디렉토리 서비스, 모니터링 등 자원을 집합으로 묶어서 관리할 수 있는 기능을 제공한다. 그리드 프로토콜이 인터넷 프로토콜과 가장 큰 차이점은, 인터넷 프로토콜은 통신할 때 서로 인접한 계층들 사이에서만 직접적으로 통신하지만, 그리드 프로토콜에서는 필요에 따라 응용 프로그램이 상위 계층을 거치지 않고 Collective나 Resource 계층으로 직접 접근할 수 있다는 점이다.

GT2의 기본적인 구조는 그림 3과 같다[17]. TLS/GSI는 Connectivity 계층의 기능을 제공하고, 그 위의 GRAM, GridFTP, MDS 등은 Resource 계층의 기능을 제공한다. Collective 계층에서 제공하는 스케줄링 기능은 Globus Toolkit에서 직접 제공하지는 않지만, Condor [21]와 같은 시스템을 연동하여 사용한다.

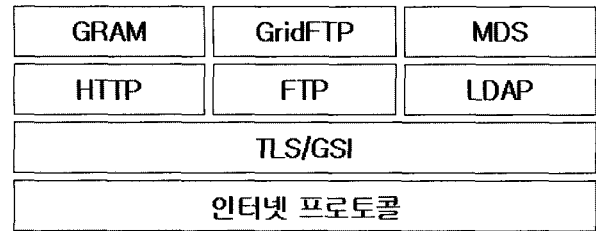


그림 3 GT2의 구조

4. Globus Toolkit 3.0의 그리드 기술

웹 서비스를 적용하기 이전의 서비스는 유닉스 데몬의 형태로 제공되었고, 그림 3에서 보듯이 GRAM은 HTTP 기반, GridFTP는 FTP 기반, MDS는 LDAP에 기반하는 등 서비스 별로 다른 프로토콜에 의존하도록 구현되었다. 웹 서비스의 개념이 적용된 GT3의 구조는 그림 4와 같다[17]. GT2에서는 서비스에 따라 다른 프로토콜을 사용하였지만, GT3에서는 모든 서비스를 그리드 서비스 인터페이스에 따라 구현하고 통합하여, 보다 유연한 시스템을 구성할 수 있다. 그리드에 웹 서비스 기술을 적용한다면 Connectivity, Resource, Collective 계층 모두 웹 서비스 프로토콜로 통합할 수 있으며, 서비스 간의 연결이나 프로토콜 변환을 보다 손쉽게 수행할 수 있다.

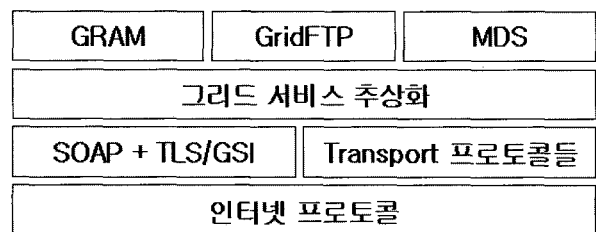


그림 4 GT3 구조

한편 기존의 웹 서비스에서는 서비스의 상태(state)를 유지할 수 없으며, 서비스 간의 통신이 동기적으로 이루어진다. 동적인 자원을 공유하기 위한 그리드 서비스에서는 자원의 상태를 필수적으로 유지하여야 하며, 높은 성능을 제공하기 위해서는 비동기 통신이 지원되어야 한다. OGSA는 이러한 특징을 지원하는 그리드 서비스의 개념을 따로 정의하였다.

OGSA에서 정의된 서비스 모델은 다음과 같다[16].

- 시스템은 소수의 정적인(persistent) 서비스와 다수의 동적인(transient) 서비스로 구성된다.
- 모든 서비스는 신뢰성 있는 서비스 호출, 생명 주기 관리, 서비스 발견, 인증, 메시지 통지, 소프트웨어 업그레이드, 동시성 등 각각의 정의된 그리드 서비스의 관리(Factory, registry, discovery, manageability)는 그리드 서비스 인터페이스와 동작에 의해 정의된다.
- 그리드 서비스 인터페이스들을 통해 안정적이고 안전하게 분산된 상태를 관리한다.

OGSI는 OGSA를 구현하기 위한 인프라를 제공하기 위한 표준이다. OGSI는 그리드 서비스를 위한 포트타입을 정의하여 WSDL을 확장한 GWSDL을 이용하여 서비스를 기술하였으며, XML 스키마를 이용해 웹 서비스의 상태를 저장하는 변수를 정의하는 방법으로 OGSA의 개념을 구현하였다[17].

OGSI가 설계될 당시의 WSDL 1.1 표준에서는 그리드 서비스의 상태 유지와 같은 개념이 지원되지 않았으며, 개발 중이었던 WSDL 2.0은 이러한 기능을 제공하는 하였지만 명세가 확정되지 않은 상태였기 때문에, 비표준적인 방법으로 웹 서비스를 확장하였다.

OGSI 1.0 버전에서 정의하고 있는 전반적인 내용들을 살펴보면 다음과 같다[22].

- 상태(state)를 가지는 웹 서비스
- 웹 서비스 인터페이스의 확장
- 상태 변화의 비동기적인 통지
- 서비스의 인스턴스에 대한 레퍼런스
- 서비스 인스턴스의 집합
- XML 스키마를 사용하여 정의된 서비스 상태 데이터

OGSI에 정의된 그리드 서비스를 위한 표준 포트 타입들을 살펴보면 표 1과 같다[17]. GridService 인터페이스는 반드시 구현되어야 하며 그 이외의 서비스는 필요에 따라 구현할 수 있다.

OGSI에서는 그리드 포트 타입 이외에도 서비스를 참조하기 위해서 그리드 서비스 핸들(GSH)과 그리드 서비스 레퍼런스(GSR)를 정의하고 있다[22]. 그리드 서

비스 레퍼런스는 그리드 서비스 인스턴스를 참조하는데 사용된다. 그러나 그리드 서비스 레퍼런스에 의해 지정된 인스턴스는 다른 컨테이너로 이동하는 경우에 무효화될 수 있다. 반면에 그리드 서비스 핸들은 특정 그리드 서비스 인스턴스에 대해 전역적으로 고유한 핸들을 지정한다. 그리드 서비스 핸들은 HandleResolver 서비스에 의해 그리드 서비스 레퍼런스로 변환되어 사용된다.

표 1 그리드 서비스를 정의하는 표준 포트 타입

포트타입	연산	설명
GridService	FindServiceData	핸들, 레퍼런스 등 그리드 서비스 인스턴스에 관한 다양한 정보를 질의
	SetTerminationTime	그리드 서비스 인스턴스의 종료 시간을 확인 및 설정
	Destroy	그리드 서비스 인스턴스를 종료
Notification-Source	SubscribeTo-NotificationTopic	메시지 타입과 관심 여부에 기반하여 서비스에 관련된 이벤트 통지를 구독
Notification-Sink	DeliverNotification	통보 메시지의 비동기적인 전달을 가능하게 함
Registry	RegisterService	상태가 수시로 변화하는 그리드 서비스 핸들의 등록
	UnregisterService	그리드 서비스 핸들의 등록을 취소
Factory	CreateService	새로운 그리드 인스턴스를 생성
HandleMap	FindByHandle	인자로 제공된 그리드 서비스 핸들과 현재 연결된 그리드 서비스 레퍼런스를 리턴

위에서 보듯이 OGSI는 웹 서비스 표준을 벗어난 방법으로 확장하여, 그리드 서비스에서 필요한 기능은 충족할 수 있었지만 웹 서비스와의 호환성을 저해하였다. 웹 서비스 진영에서는 그리드 서비스에 대해서 다음과 같이 비판하였다.

- 너무 많은 기능을 하나의 표준안에 담고 있다.
- 기존의 웹 서비스 및 웹 서비스 관련 도구들과 잘 호환되지 않는다.
- 지나치게 객체 지향적이다.

이러한 비판은 서비스의 상태를 유지하는 그리드 서비스의 기능을 웹 서비스에서도 유용하게 사용될 수 있음에도 불구하고, OGSI에서는 비표준적인 방법으로 웹 서비스를 확장함으로써 웹 서비스와의 호환성을 저해한 것에 대한 아쉬움의 표현이라고 할 수 있으며, OGSI에 대한 재검토와 웹 서비스의 발전과 보조를 맞춘 그리드 서비스의 발전의 필요성을 제기하였다.

5. WSRF (WS-Resource-Framework)

WSRF는 GGF 산하의 OGSI 워킹 그룹의 활동에 영향을 받아 제안되었다. 실제로 이것은 WS-Addressing과 같은 최근의 웹 서비스 연구 결과들을 활용하여,

OGSI 1.0 표준안에 제시된 개념과 인터페이스들에 따라 재구성한 것으로 볼 수 있다[20]. 앞에서 기술한대로 OGSI는 그리드 서비스에서 필요한 기능들을 제공하기 위하여 따로 GWSDL (Grid WSDL)을 정의하였지만, 이러한 확장은 그리드 서비스에서는 웹 서비스를 호출할 수 있지만 표준 웹 서비스에서는 그리드 서비스를 호출할 수 없는 호환성 문제를 야기시켰다. 이런 이유로 OGSI를 웹 서비스 표준을 기반으로 재구성하고 확장한 것이 WSRF이다.

WSRF는 웹 서비스에서의 상태 정보를 관리하는 개념인 WS-Resource를 정의하는 5개의 표준안들과 WS-Notification으로 구성된다. WSRF를 구성하는 표준안들의 내용을 살펴보면 표 2와 같다[16].

표 2 WSRF를 구성하는 표준안들 WSRF + WS-Notification

표준안	설명
WS-Resource Lifetime	WS-Resource를 즉시 혹은 시간 예약에 의해 제거하는 것을 가능하게 하는 메커니즘
WS-Resource Properties	WS-Resource 및 WS-Resource의 속성들을 검색, 변경, 삭제하기 위한 메커니즘을 정의
WS-Renewable References	WS-Addressing으로 지정되는 웹 서비스의 중단 레퍼런스가 무효화 되어, 새로운 중단 레퍼런스를 획득할 경우에 사용되는 정책 정보들을 이용해 WS-Addressing을 확장
WS-ServiceGroup	이질적인 웹 서비스들의 레퍼런스를 담고 있는 집합에 대한 인터페이스
WS-BaseFaults	웹 서비스 메시지 교환 과정에서 발생하는 오류를 리턴하는데 사용되는 기본적인 오류를 정의한 XML 타입
WS-Notification	Topic 기반의 Publish/Subscribe 패턴을 이용한 이벤트 구독 및 통지 메커니즘을 정의

OGSI에서 정의된 포트 타입과 WSRF의 차이점을 정리하면 표 3과 같다. 그리드 서비스 핸들과 그리드 서비스 레퍼런스를 표현하는 URL은 WS-Addressing 표준을 이용하여 표현한다. 그리드 서비스 인터페이스를 정의하는 각종 포트 타입은 기능별로 WS-Resource Lifetime, WS-Resource Properties, WS-Renewable References, WS-ServiceGroup, WS-BaseFaults 표준과 WS-Notification 표준으로 재정의되었다. 그리고 OGSI에서 팩토리 포트 타입을 이용해 정의되었던 서비스의 생성이 디자인 패턴을 이용해 처리한다.

표 3에서 알 수 있듯이 기능적으로는 OGSI와 큰 차이가 없다[20]. 그러나 WSRF를 구성하는 표준안들을 여러 개의 표준안으로 분할하였으며, 표준안들의 이름이 WS-로 시작하는 점에서 짐작할 수 있듯이 그리드 서비스 이외에 다른 웹 서비스에서도 원활히 사용할 수 있도록 웹 서비스 표준에 충실하게 정의하였다.

WSRF는 그리드 서비스를 웹 서비스의 표준에 적합

하도록 재구성하고 웹 서비스에 통합하여, 웹 서비스의 발전과 함께 그리드 서비스를 발전시킬 수 있는 토대를 마련하였다.

표 3 OGSI와 WSRF의 비교

OGSI	WSRF
그리드 서비스 레퍼런스	WS-Addressing Endpoint Reference
그리드 서비스 핸들	
Handle Resolver 포트 타입	WS-RenewableReference
XML 스키마로 정의된 서비스 데이터 및 데이터 접근을 위한 인터페이스	WS-ResourceProperties
그리드 서비스 생명 주기 관리	WS-ResourceLifetime
Notification 포트 타입	WS-Notification
Factory 포트 타입	디자인 패턴으로 처리
Service Group 포트 타입	WS-ServiceGroup
Base Fault 포트 타입	WS-BaseFaults

6. 결 론

그리드 서비스는 그리드 컴퓨팅에 필요한 자원을 공유하기 위한 수단으로 웹 서비스를 적용하는 목적에서 출발하였다. 웹 서비스를 확장하여 그리드 서비스를 구현한 결과 여러 계층에 존재하는 그리드 서비스와 프로토콜을 단일한 프로토콜로 통합하는 것이 가능하였다. 그러나 초기의 OGSI로 구현된 그리드 서비스는 그리드 환경에만 초점을 두고 연구되었기 때문에 표준적인 웹 서비스와의 호환에 문제가 발생하였다. 이러한 단점에도 불구하고 OGSI의 구현을 통해 그리드 서비스 개념의 유용성이 검증되었고, 그리드 서비스에서 제공하는 상태가 유지되는 서비스 등의 기능과 그리드 자체의 자원 공유 기능은 웹 서비스에 대해서도 유용성을 가진다.

그리드 서비스를 웹 서비스 표준에 통합하기 위해 웹 서비스 진영과 그리드 진영의 협력이 진행되었고, 그 결과로 웹 서비스 표준에 기반한 WSRF가 등장하였다. WSRF를 통해 그리드 서비스는 웹 서비스 표준의 일부로 완전히 통합되었고, 웹 서비스의 발전과 궤를 같이하면서 그리드 서비스를 발전시킬 수 있는 계기가 되었다. WSRF의 등장으로 그리드 서비스를 바로 웹 서비스에서 제공받을 수 있으며, 또한 웹 서비스에서 바로 그리드 서비스의 기능들을 손쉽게 사용할 수 있다. 따라서 앞으로 WSRF는 응용 범위를 넓혀가면서 폭넓게 사용될 것이다.

참고문헌

- [1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual

Organizations," International J. Supercomputer Applications, 15(3), 2001.

[2] Ian Foster and Carl Kesselman, "The Grid: Blue Print for a New Computing Infrastructure," Morgan Kaufman Publishers, 1999.

[3] Ian Foster and Carl Kesselman, "The Grid2: Blue Print for a New Computing Infrastructure," 2nd Edition, Morgan Kaufman Publishers, 2004.

[4] Global Grid Forum, <http://www.ggf.org>.

[5] Biology Workbench Portal, <http://workbench.dsc.edu>.

[6] My Grid <http://www.mygrid.org.uk>.

[7] NASA Launchpad Portal, <http://portal.ipg.nasa.gov>.

[8] FightAIDSAtHome, <http://www.fightaidsathome.org>.

[9] <http://www-unix.mcs.anl.gov/metaneos/nug30/>.

[10] The Grid Physics Network, <http://www.griphyn.org>.

[11] NEESgrid, <http://www.neesgrid.org>.

[12] iVDGL - International Virtual Data Grid Laboratory, <http://www.ivdgl.org>.

[13] Particle Physics Data Grid, <http://www.ppdg.net>.

[14] 한국과학기술정보연구원, "e-Science 국내 연구 환경 영향 평가 및 추진 타당성 조사연구", 2004.

[15] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," Intl J. Supercomputer Applications, Vol.11, No.2, pp. 115-128, 1997.

[16] Web Services Glossary, <http://www.w3.org/TR/ws-gloss/>.

[17] I. Foster, C. Kesselman, J. Nick, and 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.

[18] I. Foster, J. Frey, S. Graham, S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, I. Sedukhin, D. Snelling, T.

Storey, W. Vambenepe, and S. Weerawarana, "Modeling Stateful Resources with Web Services v. 1.1," 2004. 3.

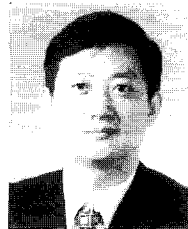
[19] Ian Foster's presentation, "WS-Resource Framework: Globus Alliance Perspectives," http://www.globus.org/wsrf/foster_wsrf.pdf.

[20] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke, "From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution," 2004. 3.

[21] Condor, <http://www.cs.wisc.edu/condor/>.

[22] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, and D. Snelling, "Open Grid Services Infrastructure (OGSI) Version 1.0," Global Grid Forum Draft Recommendation, 2003. 6.

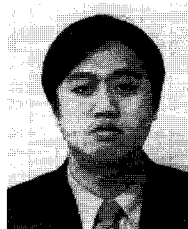
최재영



1984 서울대학교 제어계측공학과(학사)
 1986 미국 남가주대학교 전기공학과(석사)
 1991 미국 코넬대학교 전기공학과(박사)
 1992~1994 미국 국립 오크리지연구소 연구원
 1994~1995 미국 테네시 주립대학교 연구교수
 2001~2002 미국 국립 슈퍼컴퓨팅 응용 센터 (NCSA) 초빙연구원

1995~현재 숭실대학교 정보과학대학 컴퓨터학부 부교수
 관심분야: 그리드컴퓨팅, 병렬/분산컴퓨팅, 유비쿼터스컴퓨팅
 E-mail: choi@ssu.ac.kr

이상근



2001 숭실대학교 컴퓨터학부(학사)
 2003 숭실대학교 컴퓨터학과(석사)
 2003~현재 숭실대학교 컴퓨터학과(박사과정)
 관심분야: 그리드컴퓨팅, 초고속컴퓨팅(HPC), 전자상거래
 E-mail: sglee@ss.ssu.ac.kr