

# 웹 서비스를 통한 그리드의 진화

함재균\* 명훈주\*\* 김형진\*\*\* 이종숙\*\*\*\*

## ◆ 목 차 ◆

1. 서론
2. 그리드의 동향
3. 그리드 서비스의 표준 아키텍처
4. 웹 서비스와 그리드의 수렴
5. 결론

## 1. 서론

그리드 컴퓨팅은 그 출현 이후 많은 관심을 모으며 차세대 컴퓨팅의 중요한 몫을 감당할 것으로 기대가 되었다. 원래 그리드 컴퓨팅은 슈퍼컴퓨터와 같은 고성능 계산 자원을 가상의 조직으로 묶어 자유롭게 사용하게 함으로써 엄청난 양의 계산 자원을 필요로 하는 과학 분야를 지원하기 위한 것이었다. 그러나 서로 다른 조직간의 자원의 공유와 컴퓨팅, 스토리지, 데이터, 네트워크 등의 자원의 가상화를 가능하게 하는 그리드의 기본 철학은 기존 IT 환경의 변화를 모색하던 대형 IT 벤더들의 관심을 끌었고, 많은 기업들이 그리드 커뮤니티에 적극적으로 참여하기 시작하였다. 뿐만 아니라 그리드 시스템에서의 표준화가 중요한 문제로 대두되면서, 현재 분산 시스템의 가장 중요한 표준인 웹 서비스를 도입하게 되었다.

본 고에서는 웹 서비스를 도입한 그리드 서비스 표준에 대해 설명하기 위해, 그리드의 시작에서부터 미들웨어 및 시스템 측면에서 어떻게 발전되어 왔는지를 1장에서 소개하고, 그리드 표준화 기구인

GGF (Global Grid Forum)에서 정의한 그리드 서비스의 표준 아키텍처인OGSA (Open Grid Service Architecture)에 대해서 3장에서 설명할 것이다. 마지막으로 4장에서는 그리드에서 출발한 웹 서비스 표준의 하나인 WS-RF (Web Service Resource Framework)과 WS-N (Web Service Notification)에 대해서 소개할 것이다.

## 2. 그리드의 동향

그리드에 관한 연구개발은, 1995년 슈퍼컴퓨터 컨퍼런스에서 I-WAY의 시연을 기점으로 확산되기 시작했다. I-WAY는 OC-3 네트워크를 이용하여 연결된 슈퍼컴퓨터들과 가시화 장비 상에서 60여개의 어플리케이션을 수행함으로써 고성능 네트워크 기반 컴퓨팅의 가능성을 보여 주었다. 이를 기점으로 미국 내에서는 NSF PACI, IPG, ASCI DISCOM과 같은 여러 정부 주도 그리드 개발 과제가 시작되었다.

NSF PACI는 1998년 NCSA와 SDSC중심으로 시작된 연구로서 국가 차원의 그리드 인프라를 위한 미들웨어 및 알고리즘을 개발하는 것을 주요 목표로 삼았다. IPG는 1999년 NASA Ames, Glenn, Langley의 장비들을 초고속망을 통해 연결하는 통합된 연구환경을 구축하는 프로젝트이며, ASCI DISCOM은 1999년 에너지성이 중심이 되어 무기

\* 한국과학기술정보연구원 연구원

\*\* 한국과학기술정보연구원 연구원

\*\*\* 과학기술연합대학원대학교 그리드슈퍼컴퓨팅 석사과정

\*\*\*\* 한국기술연합대학원대학교(UST) 그리드슈퍼컴퓨팅 전공부문 겸임교원 (조교수)

관련 연구를 위해 테라플롭스급 계산 기술을 확보하기 위해 시작된 연구이다.

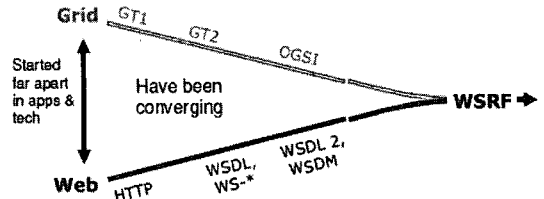
이러한 그리드 연구의 초창기에 많은 그리드 미들웨어가 등장하게 되었으며, 그 중 주목할 만한 미들웨어가 ANL (Argonne National Laboratory) 과 USC (University of Southern California)의 ISI (Information Science Institute), 그리고 시카고 대학교 (University of Chicago)에서 공동으로 개발한 글로버스 툴킷 (globus toolkit)이다. 글로버스 툴킷은 PPDG, NEESGrid, DOE Science Grid, GriPhyn, 영국 e-Science 그리드 프로젝트(UK e-Science Grid Project) 등 세계 여러 그리드 프로젝트에서 사용되고 있다.

이 시기 당시 다발적으로 세계 각국에 그리드 연구 프로젝트들이 많이 생겨나게 되었는데, 각 프로젝트에서 사용하는 그리드 시스템은 상호 연동이 어려우며, 기술 재사용의 어려움 등의 여러 가지 문제점이 제기되었다. 이를 해결하기 위해 GGF (Global Grid Forum)에서는 OGSA (Open Grid Services Architecture)라는 그리드 기반 서비스에 대한 표준 아키텍처를 제정하기에 이르렀다. OGSA의 목표는 그리드 환경의 어플리케이션이 공통적으로 제공하는 모든 서비스에 대하여 공통된 인터페이스를 규정함으로써 서로 다른 프로젝트 간의 상호 연동성, 코드의 재활용성 등을 높이는 데에 있다.

OGSA가 기반으로 하는 미들웨어 아키텍처는 웹 서비스이다. 이는CORBA, RMI, RPC, EJB와 같은 기존의 미들웨어 아키텍처와 비교했을 때 웹 서비스가 느슨하게 결합된 (loosely coupled) 시스템이라는 점이 그리드 환경에서의 이기종의 시스템들 간의 통신에 보다 적합하기 때문이다.

하지만 웹 서비스는 상태를 가지고 있지 않고 일시적이기 때문에, GGF에서는 웹 서비스를 확장한 그리드 서비스를 정의하였으며, 이에 관한 명세가 OGSi (Open Grid Services Infrastructure)이다. OGSi는 상태를 가지는 웹 서비스, WSDL를 확장한 GWSDL (Grid Web Services Description Language), 통보(notification) 매커니즘 등 웹 서비스 기반으로 부터 확장된 기능을 제공하고 있다.

GT3(Globus Toolkit 3)는 이러한 OGSi 명세를



(그림 1) 웹 서비스와 그리드의 진화

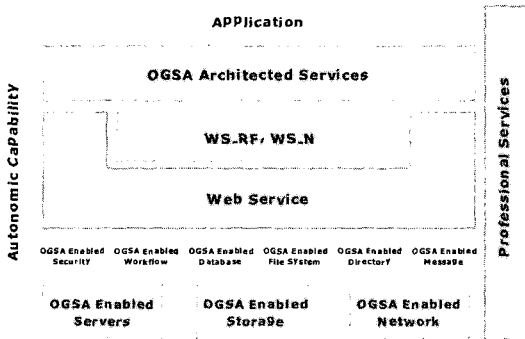
자바로 구현하였으며, OGSA에서 규정한 서비스들을 일부 제공하였다. GGF에서는 이렇게 구현한 OGSi 명세가 웹 서비스의 한 표준으로 받아들여질 바랬으나, OGSi 명세의 방대한 양과 복잡성, 기존 웹 서비스 툴과의 비호환성 등의 이유로 받아들여지지 않았다.

위의 문제는 웹 서비스와 그리드 서비스의 통합을 위해서, 2004년 1월에 개최된 글로버스 월드 (Globus World) 2004에서 WS-RF (Web Services Resource Framework)를 발표함으로써 해결점이 보이기 시작했다. WS-RF는 기존 OGSi를 개념을 계승하며, 그의 문제점을 보완한 새로운 표준이다. 또한 웹 서비스와 호환성이 개선되었으며, IBM과 HP 외에도 대다수의 IT 기업들이 지원하기 때문에 표준으로 정착하기가 수월할 것으로 예상되고 있다.

그리고 WSRF를 적용한 첫 구현물인 GT4가 2005년 4월 말에 정식 발표 되었으며 향후 GT4가 어떻게 활용될 것인지 주목되고 있다.

### 3. 그리드 서비스의 표준 아키텍처

앞장에서 언급된 대로, 그리드의 개념이 분산된 이기종(heterogeneous)의 자원들을 네트워크로 연동하여 하나의 거대한 가상 컴퓨터처럼 사용할 수 있다는 것임에도 불구하고 OGSA 이전의 그리드 미들웨어 툴킷이나 솔루션들은 상호 작용(interaction)과 상호 이용성(inter-operability)에 대한 표준과 프로토콜 등을 고려하지 않는 한계를 안고 있었다. 이 문제는 또한, 구현물을 재사용하기 어렵다는 단점이 떠오르게 했다. 이러한 문제 해결을 위해 공개 표준이 필요하다는 것이 OGSA가 나오게 된 동기가 되었으며, 2002년 6월 GGF의 OGSi 워킹



(그림 2) OGSA의 구조

그룹(working group)에서 명세가 작성되었다.

그림 2는 OGSA의 구조를 나타낸 것이다.

위의 그림을 살펴보면 기존의 그리드의 구조에 비해서 크게 달라진 점들을 확인할 수가 있는데, 그 중 하나가 OGSA 구조 한 가운데를 웹 서비스가 차지하고 있다는 점이다. 다시 말하면, OGSA는 웹 서비스를 이용하는 SOA (Service Oriented-Architecture)[7]를 채택하고 있다.

(표 1) SOA의 주요 특징

- 서비스는 자기 설명적인 인터페이스를 가진다.
- 서비스를 찾을 수 있는 방법을 제공한다.
- 메시지는 형식을 갖추어서 정의된다.
- 서비스는 XML과 같은 플랫폼 독립적인 표준을 사용해 기술되고 동작한다.
- 서비스는 네트워크의 어떤 플랫폼에서도 제공될 수 있다.

SOA는 (표 1과 같은 특징들을 가지고 있는데, 이러한 SOA의 특징과 SOA가 추구하는 목표들 상호 운용성, 느슨한 연결, 기존에 있는 시스템의 재사용성, 기존의 서비스들을 이용해 새로운 서비스를 생성할 수 있는 능력(composability) - 이 새로운 그리드 공개 표준의 필요성에 정확하게 부합하고 있다는 것을 알 수 있다. 한편, 그리드 기술이 SOA를 채택함으로써 현재의 IT 시스템들 (J2EE, .NET 등)과 연동하여 쓰일 수 있는 기초를 마련했다는 점에서도 큰 의미가 있다고 하겠다.

다음 절부터는 OGSA의 구조를 각 계층별로 간

단히 살펴본다.

### 3.1 자원(Resource) 계층

자원은 물리적 자원과 논리적 자원으로 나누어서 생각할 수 있다. 보통 물리적 자원은 서버, 스토리지, 네트워크 등을 일컫는다. 한편, 논리적 자원은 물리적 자원 위에 있으며, 물리적 자원들을 가상화 혹은 집합화시키는 기능들을 제공하는데, 파일 시스템, 디렉토리, 데이터베이스 관리 시스템 등과 같은 것들이 논리적 자원에 해당된다.

### 3.2 웹 서비스 계층

OGSA에서 모든 그리드 자원들은 “서비스”화된다. 그렇다면, 웹 서비스 기술만으로는 불충분한가? 실제 그리드 상에서는 다음과 같은 요구사항들이 존재한다.

- 그리드에서의 서비스들은 동적이고, 잠재적으로 일시적인 경향을 보인다. 즉, 그리드 서비스는 서비스의 생성, 파괴, 생명 주기 관리할 인터페이스를 필요로 한다.
- 그리드 서비스는 서비스와 관련된 속성과 데이터를 가질 수 있다. 즉, 그리드 서비스는 상태 정보를 필요로 한다.

현재 웹 서비스의 기술만으로는 아우르지 못하는 이러한 그리드 서비스의 요구사항 때문에, OGSA가 제안된 초창기에는 표준 웹 서비스를 확장한OGSI 명세에서 이러한 그리드 서비스의 요구사항을 다루고 있다.

그러나, 앞 장에서 언급한 바와 같이, 이러한 식의 접근 방법은 웹 서비스의 변종을 유발시키는 일일 뿐 아니라, 그리드와 웹 서비스를 인터페이스로 사용하는 IT 시스템들과 연동을 어렵게 한다. 이러한 문제를 해결하고자 OGSI 명세의 주요 부분들을 웹 서비스의 표준으로 만들려는 움직임이 그리드 커뮤니티에서 일기 시작하여 현재 OASIS의 WS-RF, WS-N 등의 TC (technical community)들

이 활동 중에 있다. WS-RF와 WS-N에 관한 사항은 다음 장에서 다루도록 한다.

### 3.3 OGSA 기반 그리드 서비스 계층

- 핵심 서비스: 핵심서비스는 크게 4가지로 이야기 될 수 있다. 그리드 서비스 간의 상호 통신할 수 있는 기본 방법을 제공하는 서비스 통신 (Service communication), 그리드 상에서 설치, 유지보수, 감시, 문제해결 작업 등을 다루는 서비스 관리 (Service management), 다양한 시스템들이 안전하게 상호작용 할 수 있도록 여러 보안 모델, 메커니즘, 프로토콜을 지원, 통합하는 보안 서비스 (Security Service), 시스템 작동에 위한 정책과 협약의 생성, 운영, 관리에 대한 프레임워크를 제공하는 정책 서비스 (Policy Service)들이 바로 그것이다.
- EMS (Execution Management Service)[5]: EMS는 태스크를 수행하고, 그 태스크를 관리하는 전반에 관련된 문제를 다루고 있다. EMS 서비스를 3개의 계층으로 나누어 생각할 수 있다.
  - \* 자원 (데이터 처리, 저장, 자원관리 등)
  - \* 작업 관리 및 감시 서비스: 작업 관리자는 작업을 수행시키는 일을 하기 위해서 여러 서비스들을 조정하는 역할을 하며, 워크 플로우 관리자, 포탈, 큐 (queue) 등의 여러 형태가 있을 수 있다.
  - \* 자원 선택 서비스 (Resource selection Service): 태스크를 어떤 자원에서 수행시킬지 결정하는 서비스이다. 여기서는 작업을 수행시킬 후보군(群)을 만들어 내고, 실제 스케줄링을 하고, 자원을 예약하는 일을 다루게 된다.
- 데이터서비스[5]: 데이터 서비스는 데이터를 원하는 곳으로 이동, 복제 및 복제본 관리, 갱신, 데이터를 새로운 포맷으로 변경 등의 일을 다루고 있다. 이 서비스의 인터페이스는 데이터 가상화의 개념을 지원하고 있으며, 다양한

타입의 정보들의 분산 접근과 관련된 메커니즘을 제공한다.

- 도메인에 특화된 서비스

### 3.4 그리드 응용 계층

OGSA 기반 그리드 서비스들이 많이 개발됨에 따라 이 서비스들을 이용하는 새로운 그리드 응용들이 나타날 것으로 생각된다.

## 4. 웹 서비스와 그리드의 수렴

웹 서비스는 앞 장에서 설명한 대로 OGSA 구조 안에서 서비스들의 근간을 이루는 분산 컴퓨팅 모델이다. 하지만 기존의 웹 서비스에서는 서비스에서의 상태를 나타낼 수 있는 많은 방법들이 존재하기는 하지만, 웹 서비스간이 호환이 가능하도록 표준화된 방법이 아니라 각 서비스의 구현에 따르는 것이었다. 그러던 중 그리드 진영에서 기존 웹 서비스를 확장한 OGSi 명세에서 서비스의 상태를 다루는 표준이 WS-RF (Web Service Resource Framework)이라는 이름으로 제안되었고, 결국 OASIS 내에서 WSRF-TC와 WSN-TC를 통해 상태 보유 웹 서비스 (stateful Web service)에 관한 표준을 제정하기에 이르렀다. 현재 OASIS의 두 TC에서 표준화가 진행중인 명세는 표 2와 같다.

### 4.1 웹 서비스 안에서의 상태정보 모델링

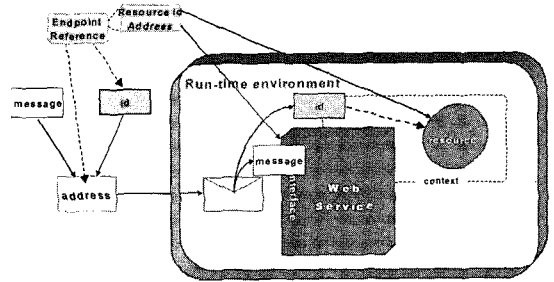
WS-RF에서는 웹 서비스 안에 상태를 표현하고 그 상태 정보를 웹 서비스 인터페이스를 통해 접근할 수 있도록 하기 위해 WS-Resource라는 개념을 도입했다. WS-Resource는 웹 서비스와 상태 보유 자원(stateful resource)과의 결합으로 이루어진다. 상태 보유 자원이란 일반적인 하드웨어뿐 아니라, 작업, 구독(subscription), 논리적인 데이터셋 등에 대한 XML 문서로 표현된 특정 상태 데이터이다. 상태 보유 자원은 생명 주기(well-defined lifecycle)를 가지며, 하나 이상의 웹 서비스에 의해 접근

(표 2) WSRF 및 WSN 명세

WS-RF 스펙	설 명
WS-Resource	WS-RF에서 상태 정보를 표현하는 기본 모델인 WS-Resource에 대한 기본 개념
WS-ResourceProperties	리소스의 상태 정보에 대한 접근 매커니즘에 대한 정의
WS-ResourceLifetime	WS-Resource의 생성 및 제거에 대한 매커니즘 정의
WS-ServiceGroup	웹 서비스들의 그룹을 생성하고 관리할 수 있는 매커니즘 정의
WS-BaseFaults	WS-Resource에 대한 요구의 결과로 발생할 수 있는 오류 타입 정의
WS-BaseNotification	웹 서비스로부터 관심 있는 정보에 대한 통보(notification)를 제공 받는 전반적인 매커니즘에 대한 정의
WS-Topics	웹 서비스에서 제공되는 통보의 타입 등 통보 메시지 생성에 필요한 토픽 및 메타 데이터 정의
WS-Brokered-Notification	통보 중개자(notification broker)의 인터페이스를 정의

되고 관리된다. 예를 들어 도서 관리 시스템을 WS-Resource로 표현해보면, 하나의 도서에 대한 상태 정보, 예를 들어 대출 여부, 대출자, 반납 예정일 등이 상태 보유 자원이 되고, 이에 대한 접근을 웹 서비스로 하게 된다.

각 상태 보유 자원은 같은 웹 서비스에 의해서 구별되는 식별자를 가지고 있는데, 클라이언트가 자원에 접근할 때 이 식별자는 메시지 안에 내재되는 방식으로 전달된다. 이때 자원에 대한 주소를 정의하는 방식이 WS-Addressing이다. WS-Addressing EPR(Endpoint Reference)는 웹 서비스의 주소와 자원의 식별자 등을 포함하는 메타데이터로 구성되어 있다. (그림 3에서 WS-RF에서 상태 보유 자원에 접근하는 매커니즘이 설명하고 있다. 클라이언트는 서비스에 전달하려고 하는 메시지를 SOAP 바디 안에 넣고 웹 서비스의 주소(EPR)을 명시한다. 이때 자원의 식별자가 SOAP 헤더 안에 포함되고, 이 메시지가 웹 서비스에 도착하게 되면 내재된 식별자를 가진 자원과 관련된 작업이 실행된다.



(그림 3) WS-Resource 접근 매커니즘

## 4.2 WS-Resource 생명 주기 (Lifecycle)

WS-Resource의 생명 주기는 그 생성에서부터 소멸에 이르기까지의 기간을 말하는데 본 절에서는 WS-Resource의 생성과 소멸 매커니즘에 대해서 설명한다.

### 4.2.1 WS-Resource의 생성

WS-Resource는 다른 방법으로도 생성이 가능하지만, 여기서 이야기하고자 하는 것은 WS-Resource 팩토리를 통한 생성이다. WS-Resource 팩토리는 새로운 WS-Resource를 만들어내는 웹 서비스로서 그 과정은 다음과 같다.

- ① 새로운 상태 보유 자원 생성
- ② 생성된 상태 보유 자원에 식별자 부여
- ③ 자원과 연관된 웹 서비스와의 결합

새로 생성된 상태 보유 자원에 부여된 식별자는 연관된 웹 서비스 내에서 유일하고, 이 식별자는 웹 서비스의 EPR의 참조 속성(reference properties) 부분에 추가되어 WS-Resource-qualified EPR을 만든다. WS-Resource 팩토리로 WS-Resource 생성을 요청한 클라이언트는 이 EPR을 응답 메시지를 통해 받게 되며, 이후 외부에서 WS-Resource에 접근할 때 주소로서 사용한다.

### 4.2.2 WS-Resource의 소멸

WS-Resource의 소멸은 두 가지 방식으로 되는데, 클라이언트에 의해 소멸 요구를 받아 이루어진다.

어지거나(immediate destruction) 각 자원이 가지고 있는 수명이 다하면 자동으로 소멸될 수 있다 (scheduled destruction). 팩토리를 통해 자원을 생성하였던 클라이언트는 생성시 받은 WS-Resource-qualified EPR을 이용해 해당 자원에 소멸 요구를 보내게 된다. 그러면 EPR안에 명시된 상태 보유 자원을 제거함으로써 WS-Resource는 소멸된다. 두 번째 방식에서는 상대 보유 자원의 생성시 수명을 부여함으로써 일정 시간 이후에 자원이 소멸되도록 하는데, WS-ResourceLifetime 명세 안에는 자원에 수명을 부여하거나 수정하거나 함으로써 WS-Resource의 소멸 시간을 조절할 수 있는 메시지 형식 등이 정의되어 있다.

### 4.3 WS-Resource 속성

WS-ResourceProperties 명세에는 웹 서비스 인터페이스를 통해 접근되는 WS-Resource의 상태 정보에 대한 타입과 값 등에 대해 정의되어 있다. 각 WS-Resource는 자원 속성 문서 (resource property document)라는 XML 스키마를 가지고 있고, 이 스키마를 통해 정의된 상태 정보는 웹 서비스의 WSDL 포트타입에 의해 공개되고 접근 가능하게 된다. 또한 WS-ResourceProperties 명세에서는 자원 속성 값을 읽어오거나 입력/수정/삭제할 수 있는 오퍼레이션을 정의하고 있다. 이 밖에도 자원 속성 문서에 대해 XPath 쿼리 등을 실행할 수도 있다. 또 속성에 대한 읽기 접근은 통보(notification)방식을 통해서도 이루어지는데, 이에 대해서는 별도의 OASIS TC에서 다루어지고 있다.

### 4.4 WS-Notification

WS-Notification은 웹 서비스에 대한 발행/구독 (publish/subscribe) 매커니즘을 정의하고 있는 명세이다. 좀 더 자세히 설명하자면, 웹 서비스의 특정 토픽에 관심이 있는 구독자 (subscriber)는 해당 웹 서비스 안에 구독 신청 할 수 있고, 이 구독은 통보 중개자 (Notification Broker)가 관리한다. 통보 중개자는 구독자와 발행자를 연결시키고 중간에서

통보를 중개하는 웹 서비스이다. 발행자측에서 구독된 토픽에 변화가 일어나면 그 메시지는 통보 중개자에게 전달되고, 통보 중개자는 이 정보에 연관된 모든 구독자에게 통보한다. 여기서 기억할 것은 구독 역시 하나의 WS-Resource이기 때문에 생명 주기를 가지고 있고, 속성을 가지며 WS-Resource-qualified EPR 역시 가지고 있다.

## 5. 결론

그리드는 그 출발은 어떠한데 간에 이제 많은 IT 기업들이 관심을 기울이는 중요한 IT기술이 되었고, 그 결과로 웹 서비스를 채택한 표준 인터페이스와 아키텍처를 가지게 되었다.

OGSA는 그리드 서비스의 표준 아키텍처로서 그리드 서비스의 모델을 제시하고 있고, 각 서비스의 기반 구조로서 웹 서비스를 채택하였다. 이러한 결과로 그리드 시스템은 상호 운용성이나 재사용성이 향상된 서비스들로 구성된다.

뿐만 아니라 그리드 진영에서 정의된 OGSI 스펙은 웹 서비스에도 영향을 미쳐서 결국 OGSI를 통해 확장된 개념들이 웹 서비스 표준에 포함되도록 논의가 진행되고 있다. 이미 WS-RF기반의 그리드 미들웨어인 GT4가 출시되었고, 머지 않아 WS-RF 및 WS-N 표준도 완성될 것으로 보여 웹 서비스 기반의 그리드 기술을 적용한 IT 인프라의 활용이 더욱 확산될 것이다.

## 참고 문헌

- [1] Borja Sotomayor, "Toward a service-oriented Grid", 2004
- [2] Ian Foster, Carl Kesselman, Jeffrey M, Nick/Steve Tuecke, "The Physiology of the Grid"
- [3] Borja Sotomayor, "Globus Toolkit 4 Programmer's Tutorial"
- [4] S.Tuecke, K. Czajkowski, I. Foster, "Open Grid Services Infrastructure Version 1.0", 2003

- [5] I. Foster, H. Kishimoto, A. Savva, "Open Grid Services Architecture, Version 1.0", 2005
- [6] "A visual tour of Open Grid Service Architecture," see <http://www-106.ibm.com/developerworks/grid/library/gr-visual/>
- [7] "서비스 지향 아키텍처의 이해", [http://www.it-solutions.co.kr/lecture/lecture\\_print.asp?code=01&num=45](http://www.it-solutions.co.kr/lecture/lecture_print.asp?code=01&num=45) 참조 .
- [8] Ian Foster, Jeffrey Frey, Steve Graham, et al., "Modeling Stateful Resource with Web Services", 2004
- [9] Steve Graham, Anish Karmarkar, Jeff Mischkinsky, Ian Robinson, Igor Sedukhin, "Web Services Resource 1.2", OASIS WSRF-TC, 2005
- [10] Latha Srinivasan, Tim Banks, "Web Service Resource Lifetime 1.2", OASIS WSRF-TC, 2005
- [11] Steve Graham, Jem Treadwell, "Web Service Resource Properties 1.2", OASIS WSRF-TC, 2005

◎ 저자 소개 ◎



**한재균**

1999년 전남대학교 전산학과 학사  
2002년 한국과학기술원 전산학과 석사  
2002년~현재 한국과학기술정보연구원 연구원



**명훈주**

1998년 서강대학교 전자계산학과 학사  
2000년 서강대학교 대학원 컴퓨터학과 석사  
2000년~2002년 동양시스템즈 기술연구소 주임연구원  
2002년~현재 한국과학기술정보연구원 연구원



**김형진**

2004년 항공대학교 기계설계학과 학사  
2005년~현재 과학기술연합대학원대학교 그리드슈퍼컴퓨팅 석사과정



**이종숙**

1992년 충남대학교 전산학과 석사  
2001년 University of Canterbury (New Zealand) 전산학과 공학박사  
1992년~1993년 한국전자통신연구원 연구원  
1999년~2002년 University of Canterbury (New Zealand) 연구원  
2002년~현재 한국과학기술정보연구원실 선임연구원  
2004년~현재 한국인터넷정보학회논문지 편집위원  
2005년~현재 한국기술연합대학원대학교(UST) 그리드슈퍼컴퓨팅 전공부문 겸임교원 (조교수)