# 사이트 자율성 보장을 위한 그리드 접근 제어 시스템 개발

김법균*, 안동언*, 정성종*, 장행진**
*전북대학교 컴퓨터공학과
**한국과학기술정보연구원
e-mail : kyun@duan.chonbuk.ac.kr

# Implementation of Grid Access Control System for Site Autonomy

Beobkyun Kim*, Dongun Ahn*, Seungjong Chung*, Haengjin Jang**
*Dept. of Computer Engineering, Chonbuk National University
**KISTI

요 약

지리적으로 분산된 이 기종의 유휴 자원들을 서로 연결하여 가상의 고성능 컴퓨팅 자원으로 사용하는 그리드에서 자원에 대한 접근 제어 시스템의 구축은 필수적이다. 본 논문에서는 그리드 환경 구축 시 전세계적으로 가장 많이 사용되는 Globus Toolkit 을 기반으로 하는 그리드 접근 제어 시스템을 설계 및 구현한다. 특히, 각 자원을 제공하는 사이트의 자율성을 보장하기 위해 각종 환경 설정 파일들을 이용하는 자원 접근 제어 시스템을 설계 및 구현하였다.

## 1. 서론

인터넷이 보편화되고 컴퓨터 및 네트워크 성능이 향상됨에 따라 분산 자원 기반의 고성능 어플리케이션들은 더 큰 컴퓨팅 파워를 요구하고 있다. 그리드는 지리적으로 분산된 고성능, 대용량의 자원들과 첨단 장비들을 원격에서 동시적으로 사용하여 단일 시스템처럼 사용하는 환경이다. 이러한 그리드 환경에서 어플리케이션을 수행하기 위해서는 그리드 환경에 적합한 자원 관리 시스템이 필수적이며, 특히 각 자원에 대한 접근 권한을 제어할 수 있는 시스템이 있어야 한다.

현재 구축된 대부분의 그리드 환경에서는 각 사이트의 자원 관리 정책이 상위 단계의 그리드 환경에서 설정한 방식으로 통일되어 있는 것이 일반적인 관례이다. 그러나 이러한 방식은 자원을 제공하고 있는 각 사이트의 자율성을 해치게 되며, 이로 인해 각 사이트의 그리드 환경에 대한 참여를 방해하는 요소가 될 수 있다. 그러므로, 각 사이트의 고유한 자원 관리 정책을 그리드 환경에서도 그대로 적용할 수 있도록 할 수 있는 그리드 접근 제어 시스템이 개발·적용될 필요가 있다.

본 논문에서는 그리드 환경에서의 각 자원에 대한 접근 권한을 사이트의 관리 정책에 따라 제어하는 그리드 자원 접근 제어 시스템을 설계 및 구현한다. 특히, 그리드 환경 구축 시 가장 많이 사용되는 Globus Toolkit 을 기반으로 하며, 각 사이트의 자율성을 최대한 보장할 수 있는 구조로 설계하였다.

## 2. 그리드 접근 제어 시스템

### 2.1 Globus Toolkit 에서의 접근 제어

Globus Toolkit 은 현재 진행되고 있는 그리드 환경 구축 프로젝트에서 가장 많이 사용되고 있는 미들웨어이다. 자원을 제공자 컴포넌트와 사용자 컴포넌트가 구분되어 용도에 맞게 설치하여 사용한다.

사용자의 해당 자원에 대한 접근 권한은 'grid-mapfile'에 사용자의 DN (Distinguished Name)과 로컬

자원의 계정을 함께 기입함으로써 부여된다. 기본적으로 다수의 DN 과 하나의 로컬 계정이 결합 가능하며, 각 DN 은 신뢰 가능한 CA (Certificate Authority)로부터 발급 받은 인증서로 확인된다.

```
"/O=Grid/OU=hi.ac.kr/CN=hdg" gw1
"/O=Grid/OU=hi.ac.kr/CN=how" gw2
"/O=Grid/OU=bye.com/CN=say" gw3
```

**그림 1. Globus Toolkit 의 grid-mapfile**

그림 1 에서는 '/O=Grid/OU=hi.ac.kr/CN=hdg'이라는 DN 을 가진 사용자가 작업을 제출하면 로컬 사이트에서는 'gw1'이라는 로컬 계정으로 실행된다는 의미이다. 그러므로, 외부의 그리드 사용자는 로컬 시스템 내에서 'gw1'이라는 로컬 계정이 가진 권한으로 모든 작업을 수행할 수 있게 된다.

이러한 방식은 다수의 DN 과 하나의 로컬 계정이 결합할 수 있도록 허용함으로써 로컬 시스템 내에서 발생하는 각종 기록들이 그리드 환경의 어느 사용자가 발생시킨 것인지 추적하기 힘들게 한다. 이를 추적하기 위해서는 작업을 제출할 때부터 사용자의 행위를 추적하는 별도의 모니터링 모듈을 필요로 하며, 로컬 시스템에 상당한 부하를 줄 수도 있다.

또한, 이를 관리하는 별도의 모듈 없이, 시스템 관리자가 요구가 있을 때마다 일일이 수작업으로 수정을 해주어야 하는 불편이 있다. 그리고 외부 사용자와 로컬 계정은 일시적으로 1:1 결합되는 것이 보안 및 시스템 관리 관점에서 가장 적절한 형태로 사료된다. 따라서 이러한 처리를 위한 모듈을 필수적이라 할 수 있다.

## 2.2 그리드 접근 제어 시스템의 설계

본 논문에서는 전술한 문제점을 해결하고 그리드 환경에서의 부가 서비스를 위해 다음과 같은 그리드 접근 제어 시스템을 설계하였다.

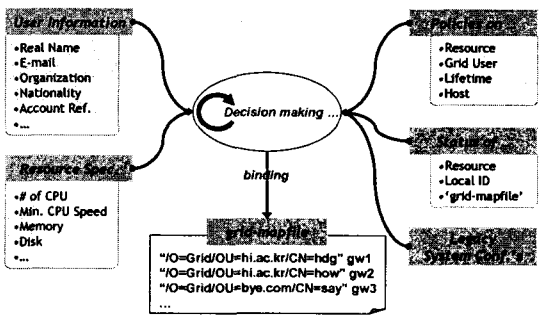


**그림 2. 그리드 접근 제어 시스템**

그리드 접근 제어 시스템은 사용자로부터 사용자의 신상 정보와 요구되는 자원의 명세를 전달 받아 시스템 관리자가 설정한 각종 설정 파일과 상태 및 시스템 설정 파일 등을 참조하여 'grid-mapfile' 상의 바인딩 가능 여부를 결정하여 통보한다.

그리드 사용자는 각 로컬 자원을 사용하기 이전에 각 로컬 자원에 위치한 접근 제어 시스템에 로컬 자원 사용을 위한 'grid-mapfile'상의 바인딩을 요청해야 한다. 이 때, 사용자는 자신의 신상 정보와 함께 자신이 원하는 자원의 명세를 보내야 한다.

그리드 접근 제어 시스템은 사이트 관리자가 수립한 자원, 사용자, 시간, 호스트 등에 대한 관리 정책과 자원, 로컬 계정, 'grid-mapfile' 등의 상태와 시스템에서 사용하는 각종 설정 파일 등을 참조하여 사용자가 제공한 신상 정보가 등록 가능한 사용자 인지, 사용자가 요구한 자원 명세가 수용 가능한 수준인지를 판별하게 된다. 만약, 등록 가능한 사용자이고 제공 가능한 수준의 자원 요구라면, 사용자에게 발급할 로컬 계정을 선택하여 사용자에게 그 정보를 제공한다.

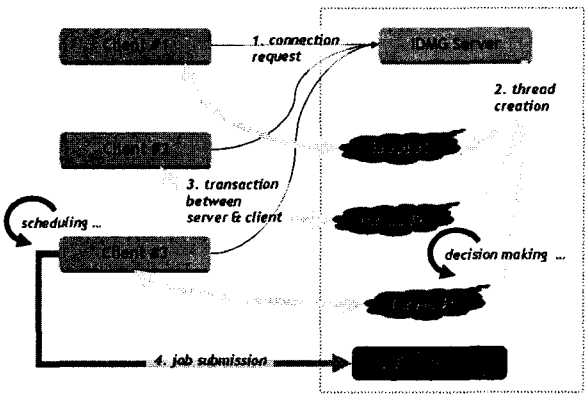


**그림 3. 사용자의 접근 권한 요청 처리 과정**

Globus Toolkit 의 경우, 사용자가 접근 권한 획득 후 작업을 제출하면, 자원의 gatekeeper 모듈이 이를 받아 처리하면서 'globus-gatekeeper.log' 라는 이름으로 그 기록을 남긴다. 대부분의 어카운팅 프로젝트의 경우 이 파일을 이용하여 로컬 자원 내에 각 프로세스를 실행시킨 실제 외부 그리드 사용자를 구분하고 있으나, gatekeeper 모듈에 폭주가 발생할 경우 파일 자체의 각 레코드가 파괴되는 문제점이 있으므로 실제 상황에 적용할 경우 심각한 장애가 발생할 가능성이 있다.

따라서, 본 논문에서는 각 클라이언트의 요청이 발생하면 각각 이 연결을 전담하는 쓰레드를 두고 각 연결을 위한 별도의 기록을 남긴 후, 연결이 종료되면 총괄 기록을 전담하는 프로세스에게 전달하여 일괄적으로 모든 연결 로그들을 한꺼번에 처리하도록 함으로써 'globus-gatekeeper.log' 파일에서 발생하는 것과 같은 문제가 발생하지 않도록 하였다. 그리고 발생하는 로그를 주기적으로 분할하여 관리가 쉽도록 할 수 있다.

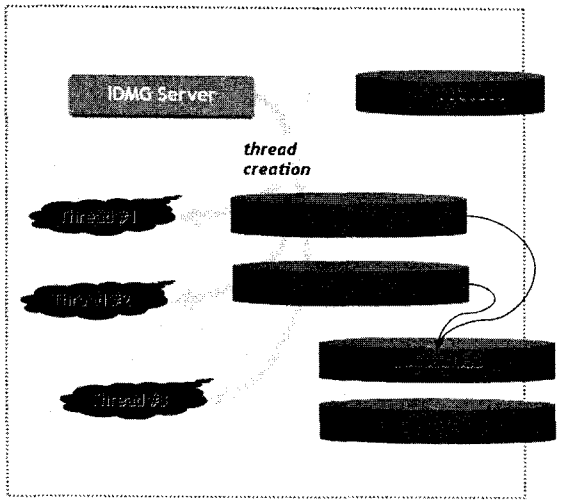


그림 4. 그리드 접근 제어 시스템의 로그 관리

2.3 사이트 자율성 보장을 위한 환경 설정

사이트의 운영 정책에 관한 자율성을 보장하기 위해 다수의 환경 설정 파일을 이용한다.

본 논문에서 설계한 그리드 접근 제어 시스템은 리눅스 운영체제의 limits.conf 나 디스크 쿼터 관련 설정 파일과 같이 기존의 시스템에서 활용하던 환경 설정 파일을 참조하고 그리드 환경에 적용시키기 위해 필요한 환경 설정 파일을 새로이 도입하여 사이트의 운영 정책에 관한 자율성을 최대한 보장할 수 있도록 하였다.

그림 5 와 6 은 각각 사용자와 호스트 관련 환경 설정 파일 및 로컬 계정관련 환경 설정 파일이다.

```
acceptable-user-DN
{
        /O=Grid/O=Globus/OU=ggf/CN=hjh
        /O=Grid/O=Globus/OU=cheonan/CN=KKY
        /O=Grid/O=Globus/OU=kisti/CN=grid
}
# give priority to all user-DN which starts with ...
acceptable-user-VO
{
        /O=Grid/O=Globus/OU=cbnu
}
acceptable-user-status
{
        worker,
        director
}
deniable-user-DN
{
        /O=Grid/O=Unary/OU=pznan/CN=polar
}
# deny all user-DN which starts with ...
deniable-user-VO
{
        /O=Grid/O=Legion/OU=cactus
}
deniable-user-status
{
        student
}
```

그림 5. policy-user.conf

```
# 'GRID_ID' is not user id. It is a general property.
GRID_ID
{
        max-binding = 5
}
gw01
{
        bindable = 1# bindable
}
gw02
{
        bindable = 0# not bindable
}
gw03
{
        bindable = 0
}
gw04
{
        bindable = 1
}
```

```
acceptable-host-ips
{
        210.117.187.133,
        210.117.131.70,
        210.117.131.71
}
acceptable-host-groups
{
        210.117.187.230.249
}
defaults
{
        max-connection = 5
        max-connection-per-host = 1
        max-connection-per-group = 3
}
max-connection-per-host
{
        210.117.187.133 = 2
}
max-connection-per-group
{
        210.117.187.230.249 = 4
}
```

그림 6. grid-id.conf 와 policy-host.conf

이외에도 다른 환경설정 파일들을 이용하며, 그리드 접근 제어 시스템은 이들 환경설정 파일들을 통해 그리드 환경내에서 사이트의 정책을 반영시킨다.

2.4 그리드 접근 제어 시스템의 활용

그리드 환경에서의 작업 제출이 포탈 또는 사용자별로 이루어지는데 이때 접근 권한 부여를 위한 사전 협약이 이루어져야 할 필요가 있다. 예를 들어, 포탈 A 에서 사이트 B 의 자원을 이용하고자 한다면, 포탈 A 는 사이트 B 에게 추후 자원 접근 권한 요구시 바인딩이 이루어질 수 있도록 포탈 A 의 어드레스, 사용자들의 DN 리스트 및 기타 정보를 사이트 B 에게 전달하여 사전에 환경 설정 파일에 등록될 수 있도록 하여야 한다. 이 과정에서 각 포탈과 각 자원 제공 사이트들 간에 사업적인 관계가 이루어져야 할 것이다.

2.5 그리드 접근 제어 시스템의 구현

본 논문에서 구현한 시스템은 그리드 환경 내에서 사용되기 위한 것이다. 그리고 그리드 환경에 포함된 각 시스템들은 서로 다른 운영체제와 하드웨어 들이며, 각각 다른 운영 방식을 가지고 있다. 따라서, 본 시스템이 각 시스템에 설치되어 사용되기 위해서는 설치가 용이하고 플랫폼에 독립적인 특성이 포함되어야 할 필요가 있어 Python 을 이용하였다. Python 은 플랫폼에 제한을 거의 받지 않는 언어로써, 자바 버전의 Jython 을 이용할 경우, 웹 어플리케이션 개발 시 자바와의 연동도 용이하다.

아래 그림은 클라이언트의 바인딩 요청에 대해 그리드 접근 제어 시스템의 동작을 보여준다.

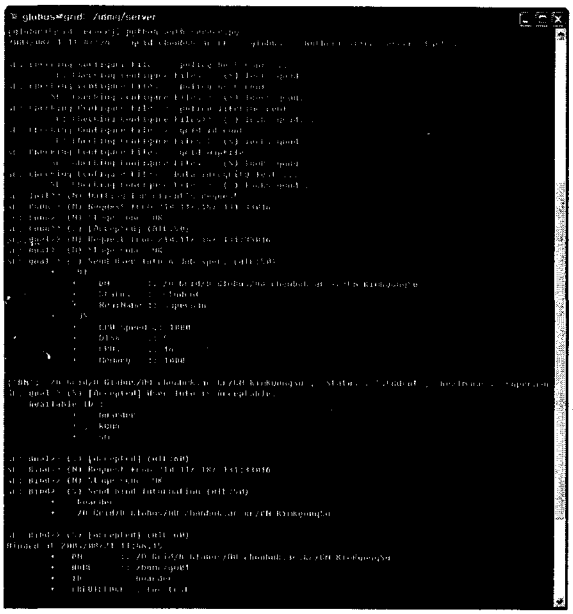

그림 7. 그리드 접근 제어 시스템의 동작

클라이언트로부터 사용자의 DN, Status, RealName 등의 신상정보와 사용자의 작업이 요구하는 CPU 의 수, 속도, Disk 용량, Memory 용량 등의 자원 명세를 전달받아 사이트 관리자가 설정한 환경 설정 파일 등을 참조하여 바인딩 가능 여부를 결정, 통보하게 되고 클라이언트는 이를 바탕으로 바인딩 요청을 함으로써 바인딩 과정이 이루어진다. 이후, 클라이언트는 Globus Toolkit 의 API 또는 커멘드를 이용하여 작업을 제출하게 된다.

## 3. 결론 및 향후 연구 방향

본 논문에서는 그리드 환경의 구축에 있어서 필수적인 그리드 접근 제어 시스템을 Globus Toolkit 을 기반으로 개발하였다. Globus Toolkit 의 접근 제어의 문제점을 해결하고 기존에 사용하던 각종 사용자 및 자원 관련 환경 설정 파일들과 그리드 환경에 적용하기 위해 필요한 새로운 환경 설정 파일들을 도입하여 자원 제공자의 관리 정책을 그리드 환경에 그대로 적용할 수 있도록 함으로써 사이트의 자율성을 보장하는 시스템을 개발하였다.

앞으로, 좀더 세밀한 제어를 위한 연구와 함께 이를 이용한 어카운팅 및 과금 서비스와 같은 부가 서비스를 위한 연구가 필요하다.

또한, 각 포탈과 자원 제공 사이트 간의 사업적인 관계를 설정하기 위한 연구도 따라야 할 것이며, 웹 기반 포탈에 적용하기 위해서는 API 형태로 제공하기 위한 노력이 필요할 것이다.

## 참고문헌


[1] Foster, C. Kesselman(eds), "The Grid : Blueprint for a New Computing Infrastructure" Mogan Kaufmann Publishers, 1998.

[2] Foster, C. Kesselman(eds), S. Tuecke "The Anatomy of the Grid: Enabling Scable Virtual Organizations", Intl. J. Supercomputer Applications, 2001.

[3] S. Mullen et al, "Grid Authentication, Authorization and Accounting Requirements Research Document", (draft), GGF8, 2003

[4] Sebastian Ho, "GridX System Design Documentation", (draft), Bioinformatics Institute, 2002

[5] A. Beardsmore et al, "GSAX (Grid Service Accounting Extensions)", (draft), GGF6, 2002

[6] R. Baker et al, "Conceptual Grid Authorization Framework and Classification", (draft), GGF8, 2003

[7] K. Czajkowski, I. Foster, et al, "A Resource Managment Architecture for Metacomputing Systems", Proc. of the 4th Workchop on Job Scheduling Strategies for Parallel Processing, 1998

[8] Thomas J. Haker, Brian D. Athey, "Account Allocations on the Grid", Center for Parallel Computing University of Michigan. 2000.

[9] http://www.girdforum.org

[10] http://www.globus.org

[11] http://www.gridforumkorea.org