

# **GRS 기반 과학기술자원융합망 테스트베드 구축**

**곽재혁, 정용환, 함재균, 박동인**  
**한국과학기술정보연구원**

# 목 차

제 1 장 소개	1
제 2 장 융합망 테스트베드 구성도	2
제 3 장 GRS 서비스 구현 프로세스 소개	4
제 1 절 보안	4
제 2 절 네트워크 자원 예약	4
제 3 절 컴퓨팅 자원 예약	5
제 4 절 네트워크 및 컴퓨팅 자원 동시 예약	6
제 5 절 컴퓨팅 자원 모니터링	7
제 4 장 GRS 서비스 자원별 설치 및 설정	9
제 1 절 웹포털	9
제 2 절 CRM	11
1. Globus Toolkit 4 미들웨어 설치	11
2. GRS 서비스 디플로이	19
제 3 절 NRM	19
제 4 절 클러스터	20
제 5 장 결론	24

## 표 목 차

표 1 융합망 구성 요소	2
---------------	---

## 그림 목 차

그림 1 융합망 테스트베드 구성	2
그림 2 융합망 소프트웨어 아키텍처	3
그림 3 GRS 웹포털	9

## 제 1 장 소개

오늘날 많은 과학 응용 분야에서는 고성능 컴퓨팅 자원과 더불어 고 신뢰성, 고 대역폭의 네트워크 자원을 요구하고 있다. 특히, 대규모 대용량의 데이터를 실시간으로 처리해야 하는 과학 응용 분야에서 이러한 요구가 계속적으로 발생하고 있는데, 대표적인 과학 응용 분야로는 고에너지 물리 분야, 기후 분야, 천문 분야에서는 매년 처리해야 하는 데이터의 규모가 수PB에 이를 것으로 예상된다.

과학기술자원융합망은 이러한 과학 응용 분야의 요구에 맞추어 구축되고 있는 차세대 연구 환경으로서 사용자는 그리드 컴퓨팅 기술을 이용하여 과학기술자원융합망에 연동된 고성능의 컴퓨팅 자원과 고 대역폭의 네트워크 자원을 사전 예약 방식으로 이용할 수 있어서 과학 응용 연구자가 필요로 하는 연구 자원에 대한 가용성을 보장할 수 있게 된다.

과학기술자원융합망에서의 핵심 기술 요소는 GRS(Global Resource Scheduler)로서 GRS는 컴퓨팅 자원 관리 시스템(CRM)과 네트워크 자원 관리 시스템(NRM)과 협업하여 과학 응용 연구자에게 고품질의 연구 환경 서비스를 제공할 수 있으며, 본 보고서에서는 GRS를 이용하여 구축된 과학기술자원융합망 테스트베드에 대해서 소개하고 과학기술자원융합망 테스트베드를 구축하기 위한 기술적인 방법에 대해서 설명한다.

## 제 2 장 융합망 테스트베드 소개

다음은 현재 서울-대전을 중심으로 구축된 융합망 테스트베드 구성도이다. 아래 예는 융합망 테스트베드의 한 예로서 이것은 실제 서비스 환경에 따라서 변동될 수 있지만 GRS(Global Resource Scheduler)를 중심으로 다양한 구성 요소에 대한 의존성을 필요로 한다. 융합망 테스트베드를 구성하는 각 구성 요소에 대해 정리하면 다음과 같다.

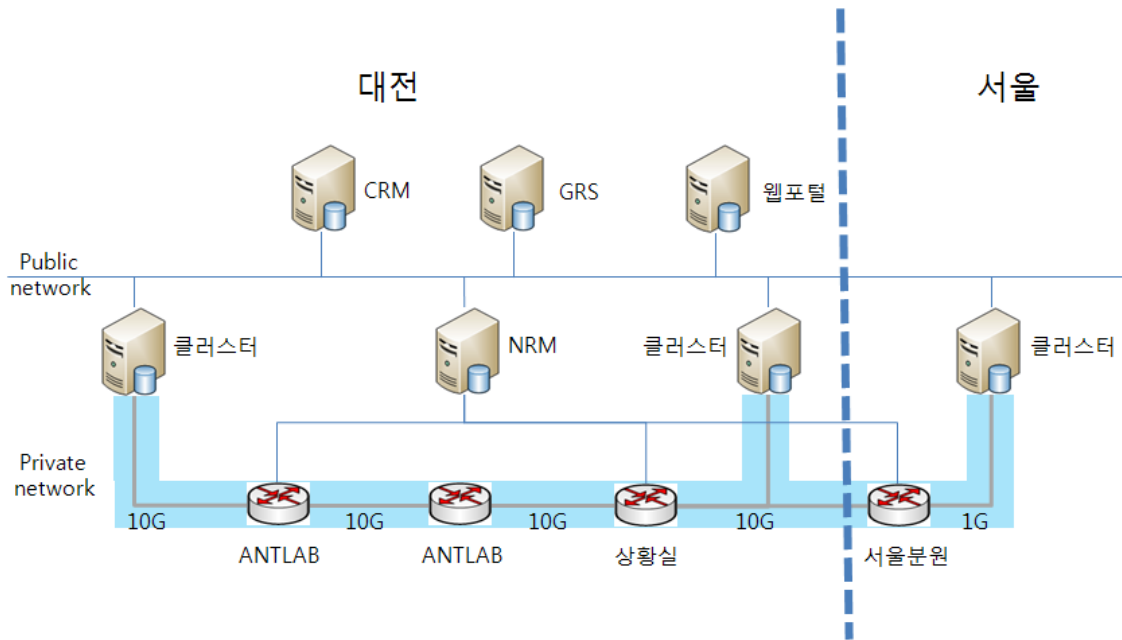


그림 1 융합망 테스트베드 구성도

표 1 융합망 구성 요소

구성 요소	기능
GRS (Global Resource Scheduler)	CRM과 NRM을 이용하여 융합망 사용자 작업에 대한 처리 및 관리를 조율
CRM (Computing Resource Management)	컴퓨팅 작업을 요구하는 사용자 작업에 대해서 컴퓨팅 자원을 예약, 관리, 모니터링하고 사용자 작업을 처리하는 역할
NRM (Network Resource Management)	선택된 네트워크 구간 간에 대역폭을 예약 및 관리, 모니터링하는 역할을 담당
웹포털	사용자에게 융합망 서비스를 제공하는 웹 기반 인터페이스
클러스터	사용자 작업을 실행하는 계산 자원

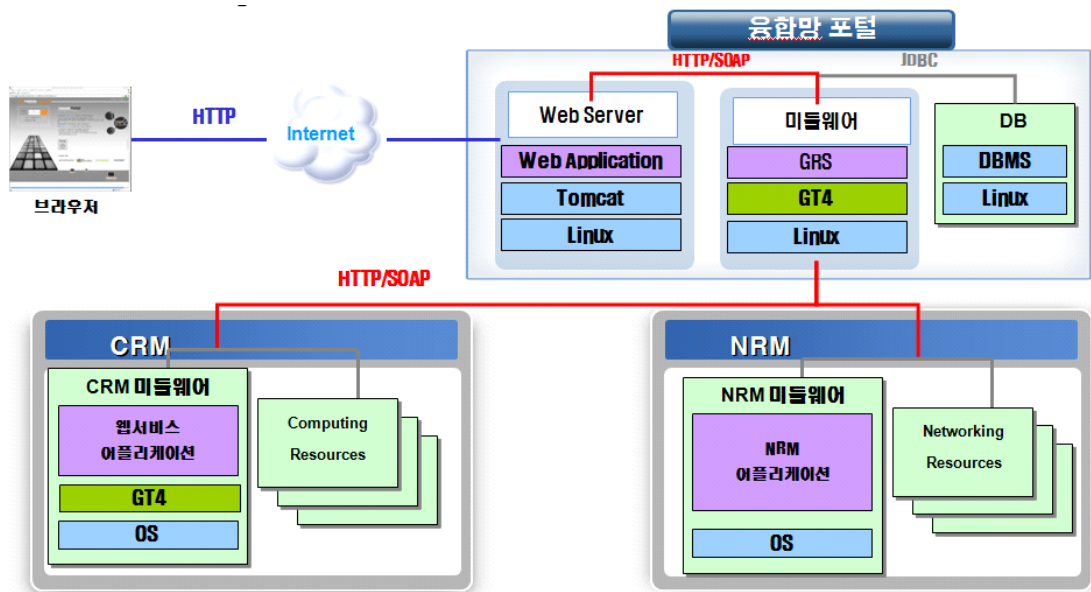


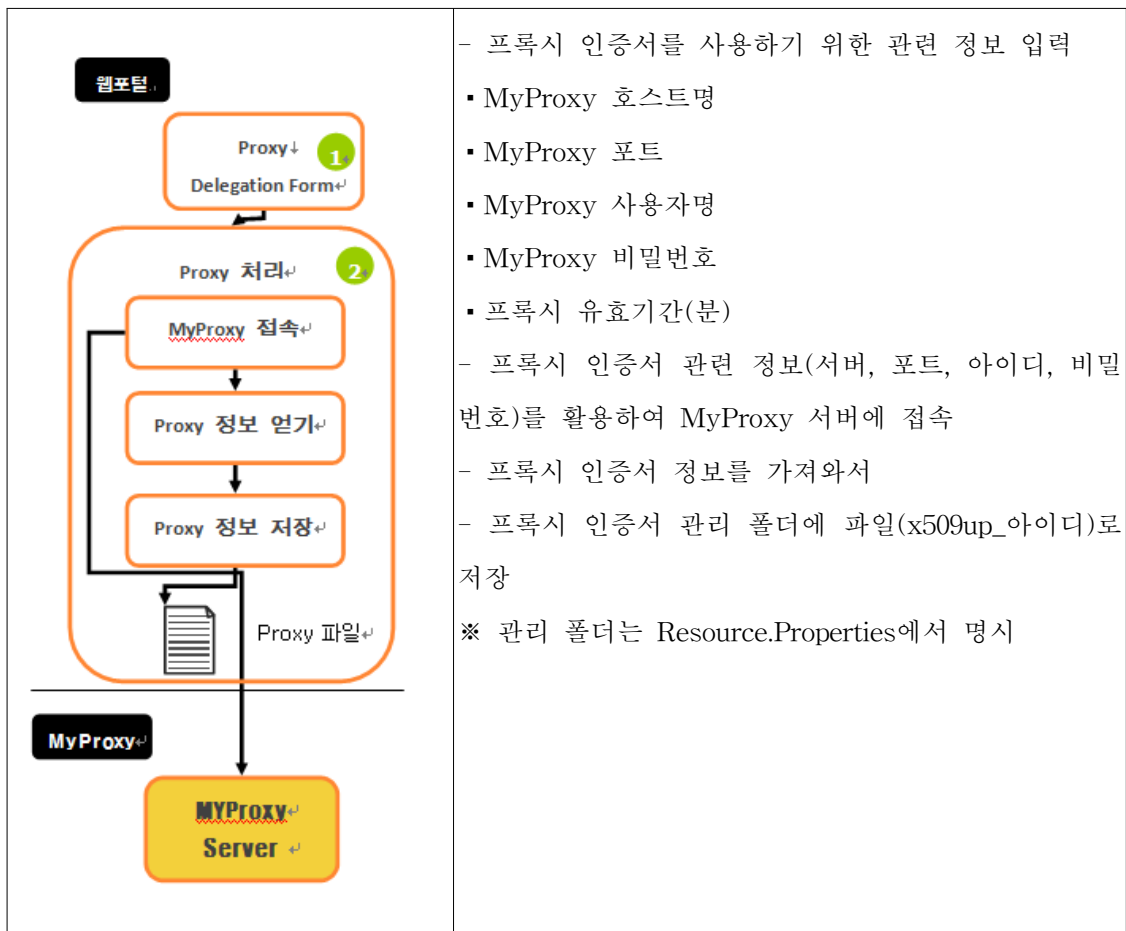
그림 2 통합망 소프트웨어 아키텍처

각 클러스터는 외부 환경에 연동된 공용 네트워크와 10Gbps로 연결된 별도의 사설 네트워크에 연결되어 있다. 만약, 사용자가 통합망 각 계산 자원 간의 대용량 데이터 전송을 주어진 시간 내에 완료해야 할 경우 해당 구간의 네트워크 자원의 대역폭을 미리 예약함으로써 신뢰성 있는 전송을 보장받을 수 있게 된다.

## 제 3 장 GRS 서비스 구현 프로세스 소개

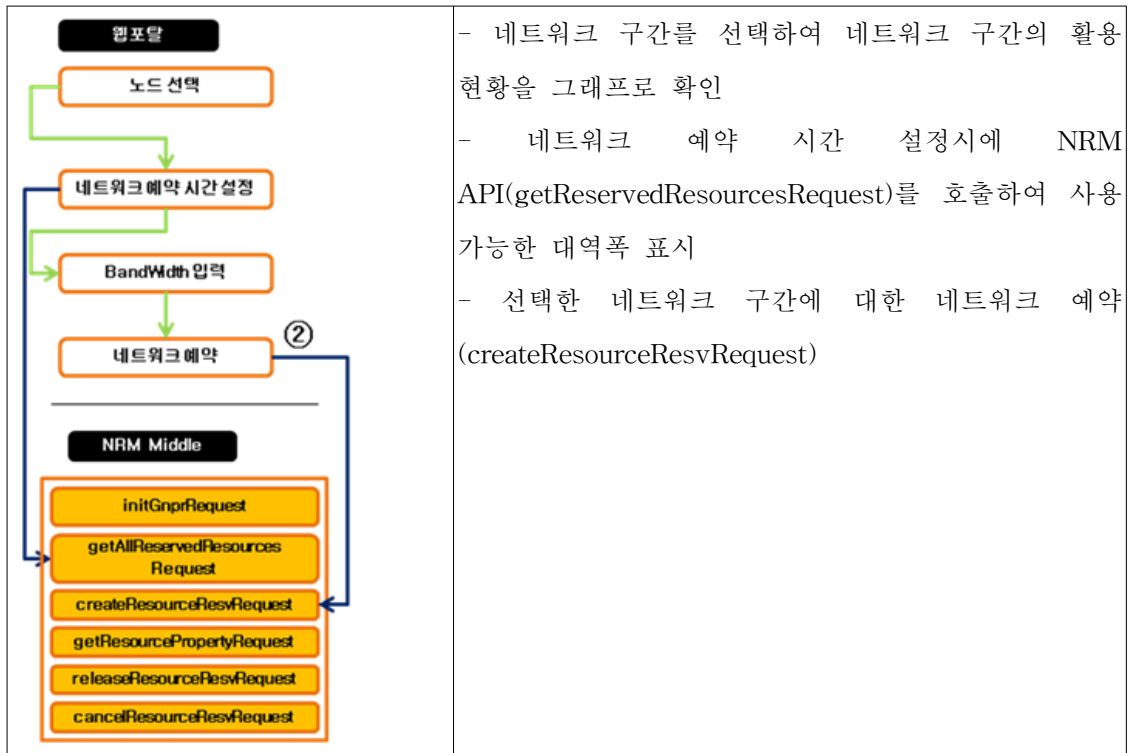
### 제 1 절 보안

GRS 서비스는 기본적으로 프록시 인증서에 기반하여 사용자 인증을 수행하게 된다. 먼저 사용자는 MyProxy서버를 이용해서 사용자의 X.509 인증서를 사용하여 프록시 인증서를 생성 및 관리하게 된다. 사용자는 GRS 포털 시스템에 이용시에 프록시 인증서를 생성한 후에 이를 GRS 서비스에 전달하여 서비스를 이용할 수 있는 사용자인지를 확인하게 된다.



### 제 2 절 네트워크 자원 예약

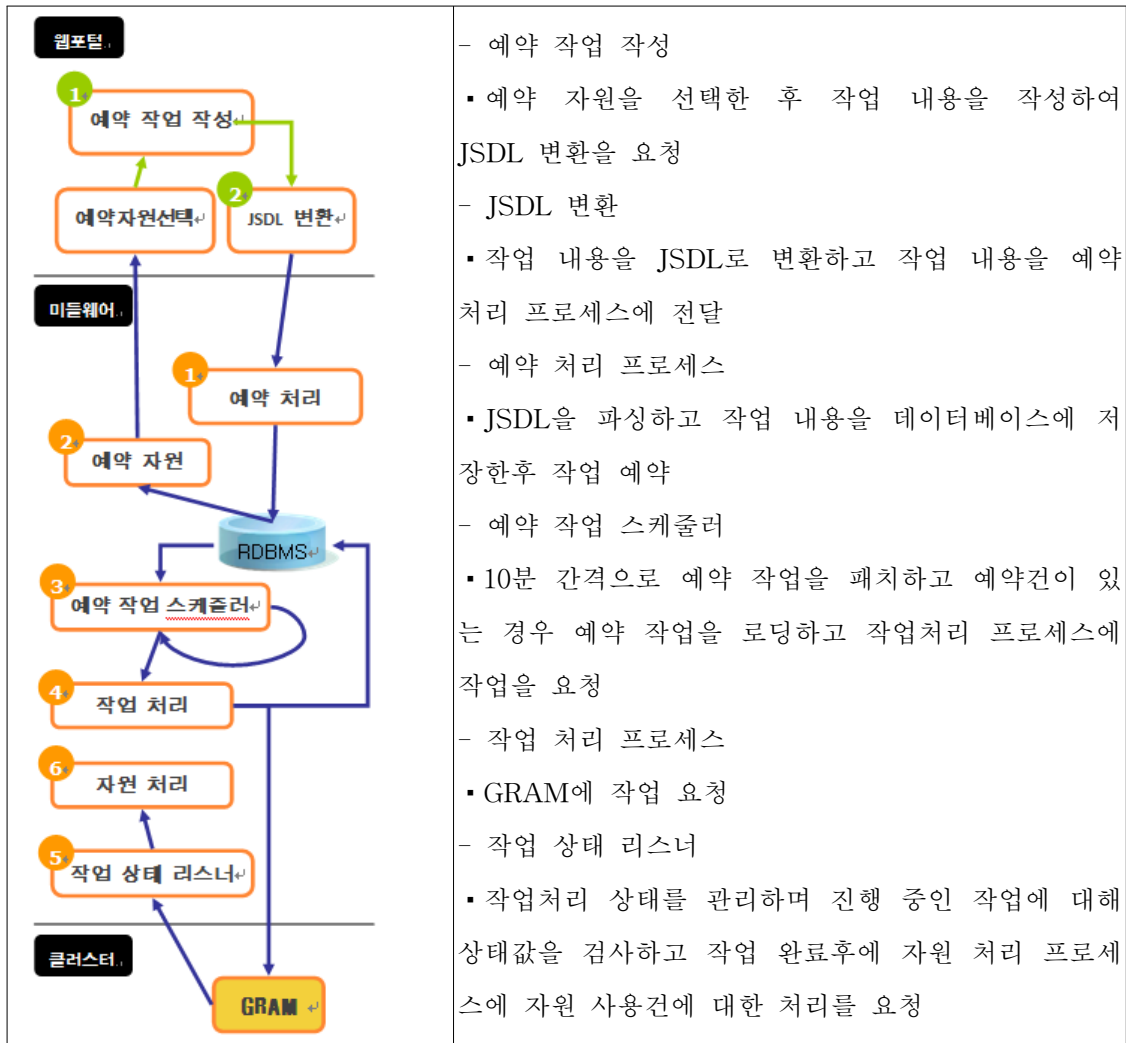
GRS 서비스가 네트워크 자원 예약을 수행하는 경우에는 네트워크 자원 관리 시스템 (NRM)이 제공하는 NRM API를 호출하여 수행한다. 이때 NRM API에 전달하는 중요한 전달 인자로서는 대역폭과 예약 시작 시간, 예약 종료 시간 등이 있다.



### 제 3 절 컴퓨팅 자원 예약

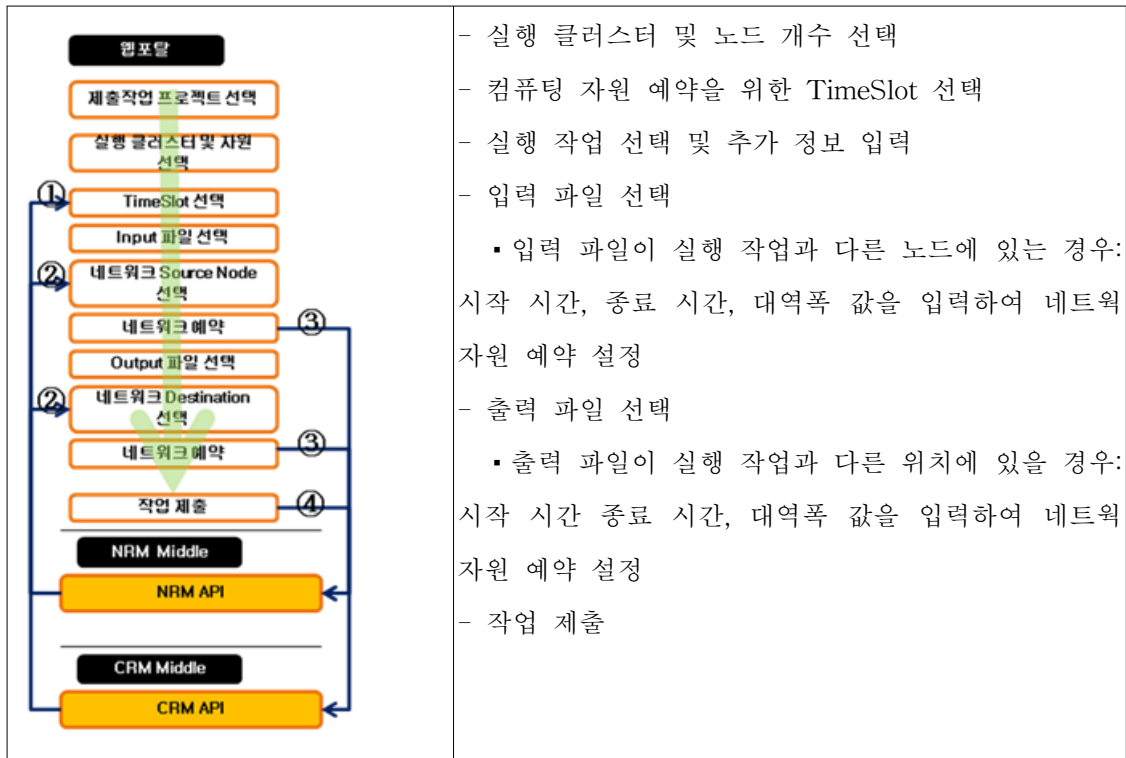
GRS가 컴퓨팅 자원 예약을 수행하는 경우에는 컴퓨팅 자원 관리 시스템(CRM)을 통해서 이루어지며, 타임테이블 기반의 컴퓨팅 자원 예약을 수행한다. 해당 예약 작업은 예약큐에 저장되고 예약 작업 스케줄러에 의해서 실행된다.





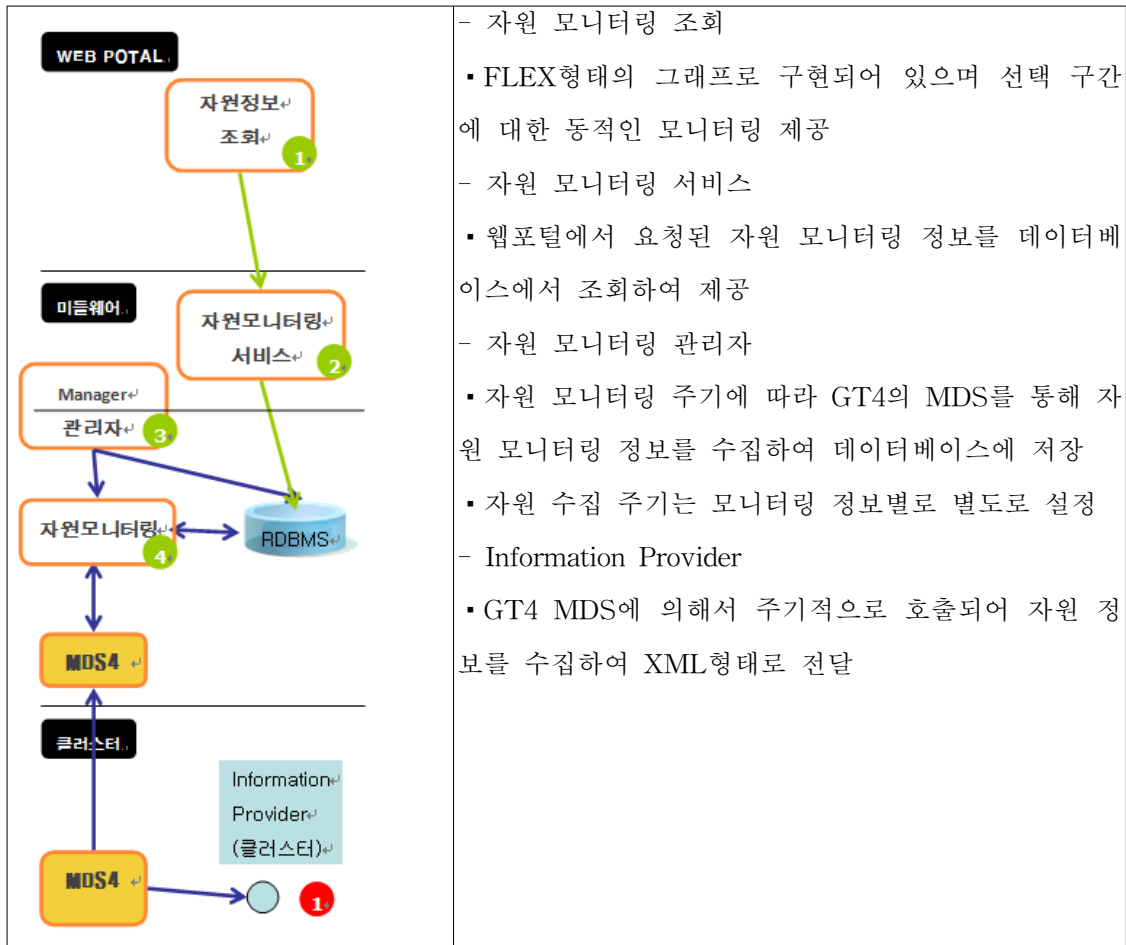
#### 제 4 절 네트워크 및 컴퓨팅 자원 동시 예약

GRS가 네트워크 자원과 컴퓨팅 자원을 동시에 예약하는 사용자 작업을 입력 받으면 컴퓨팅 작업을 담당하는 CRM에 요청하여 해당 작업에 대한 예약을 처리하고 네트워크 자원 예약이 있는 경우 NRM에 요청하여 입력 파일을 스테이지인(stage-in)하고, 출력 파일의 스테이지아웃(stage-out)하는 네트워크 구간에 대한 네트워크 자원을 예약한다. 이때 컴퓨팅 자원 예약과 관련하여 클러스터명, CPU갯수, 컴퓨팅 자원 예약 시작 및 종료 시간을 필요로 하며, 네트워크 자원 예약과 관련하여 네트워크 구간, 네트워크 자원 예약 시작 및 종료 시간, 대역폭을 필요로 한다.



## 제 5 절 컴퓨팅 자원 모니터링

컴퓨팅 자원 모니터링은 GT4가 제공하는 MDS에 기반하여 이루어진다. MDS는 계층적으로 구성되며 클러스터의 MDS는 클러스터에서 주기적으로 실행되는 Information Provider를 통해서 자원 정보를 수집하며 수집된 자원 정보는 CRM의 MDS로 전달되어 통합된다. 현재 제공되는 컴퓨팅 자원 모니터링 요소는 호스트명, CPU, 메모리, 디스크 등의 기본적인 시스템 모니터링 정보 이외에 배치 큐잉 스케줄러 정보, 소프트웨어 정보, MPI라이브러리 정보 등이 추가적으로 수집된다.



- 자원 모니터링 조회
  - FLEX형태의 그래프로 구현되어 있으며 선택 구간에 대한 동적인 모니터링 제공
- 자원 모니터링 서비스
  - 웹포털에서 요청된 자원 모니터링 정보를 데이터베이스에서 조회하여 제공
- 자원 모니터링 관리자
  - 자원 모니터링 주기에 따라 GT4의 MDS를 통해 자원 모니터링 정보를 수집하여 데이터베이스에 저장
  - 자원 수집 주기는 모니터링 정보별로 별도로 설정
- Information Provider
  - GT4 MDS에 의해서 주기적으로 호출되어 자원 정보를 수집하여 XML형태로 전달

## 제 4 장 GRS 서비스 자원별 설치 및 설정

본 장에서는 GRS 서비스 자원인 웹포털, CRM, NRM, 클러스터별로 설치 필수 사항과 설정에 대해서 알아볼 것이다.

### 제 1 절 웹포털

GRS 웹포털에서는 웹서버와 서블릿 엔진으로서 Tomcat 5 이상을 필요로 한다. 웹서버는 /software/KISTL\_Tomcat 5.0\_en에서 구동하고 있으며 다음과 같이 시작 및 종료할 수 있다.

```
bin/startup.sh (시작시)
```

```
bin/shutdown.sh (종료시)
```

GRS 웹포털은 다음의 주소에서 접속할 수 있다. 다만, 현재는 시스템 방화벽에 의해서 허가된 IP주소에서만 접근 가능하다.

```
http://gridportal.gridcenter.or.kr:8089
```

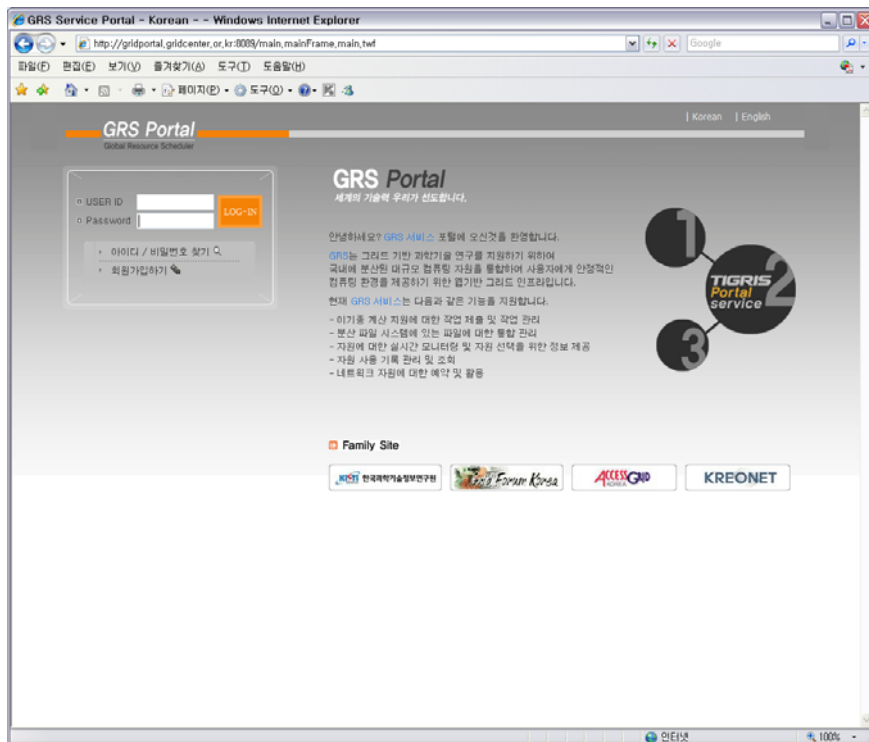


그림 3 GRS 웹포털

GRS 웹포털과 관련한 중요한 경로는 다음과 같다.

- Web Application Context Root Path: /home/thinktek/2010GRS\_WebPortal/WebRoot
- Class Path: /home/thinktek/2010GRS\_WebPortal/WebRoot/WEB-INF/classes  
/home/thinktek/2010GRS\_WebPortal/WebRoot/WEB-INF/lib
- Properties Path: /home/thinktek/2010GRS\_WebPortal/WebRoot/WEB-INF/classes

GRS에서는 서비스 관련 모든 정보를 저장하기 위해서 Oracle 10g 데이터베이스를 사용한다. 데이터베이스를 시작하고 종료하기 위해서는 다음과 같이 실행한다.

```
sqlplus sys/manager0 as sysdba
sql>startup (시작시)
sql>shutdown abort (종료시)

lsnrctl start (시작시)
lsnrctl stop (종료시)
```

웹포털은 호스트 이름 기반으로 GRS가 제공하는 웹서비스에 접근하므로 /etc/hosts에 모든 시스템의 호스트 이름을 기술해주는 것이 좋다.

```
vi /etc/hosts
150.183.24.2    nova01.gridcenter.or.kr nova01
150.183.24.3    nova02.gridcenter.or.kr nova02
150.183.24.4    nova03.gridcenter.or.kr nova03

203.230.116.199 nova07.gridcenter.or.kr nova07
203.230.116.200 nova08.gridcenter.or.kr nova08
203.230.116.201 nova09.gridcenter.or.kr nova09

210.119.43.4    nova04.gridcenter.or.kr nova04

192.168.125.7    nova01.gridcenter.or.kr nova01
192.168.150.7    nova07.gridcenter.or.kr nova07
192.168.175.7    nova04.gridcenter.or.kr nova04

150.183.250.80  mars.gridcenter.or.kr
```

```
150.183.249.95  pse.gridcenter.or.kr
150.183.250.4  gridportal.gridcenter.or.kr
```

## 제 2 절 CRM

CRM은 Globus Toolkit 4 기반의 WSRF 서비스로서 구현되어 있어서 먼저 Globus Toolkit 4가 설치되어 있어야 한다.

### 1. Globus Toolkit 4 미들웨어 설치

GT4를 설치하기 위해서는 JDK 1.4.2 이상, Jakarta Ant 1.6.5 이상이 미리 설치되어 있어야 한다. JDK는 <http://java.sun.com/j2se>에서 다운로드 받을 수 있으며 다음과 같이 설치될 수 있다.

```
tar xvzf j2sdk-1_4_2_13-linux-i586.tar.gz -C /installed
```

Jakarta Ant는 <http://ant.apache.org>에서 다운로드 받을 수 있으며 다음과 같이 설치될 수 있다.

```
tar xvzf apache-ant-1.7.0-bin.tar.gz -C /installed
```

JDK와 Ant에 대한 경로는 쉘 환경 변수에 미리 지정해 놓는 것이 좋다.

```
vi .bashrc
export JAVA_HOME=/installed/j2sdk-1_4_2_13
export ANT_HOME=/installed/apache-ant-1.7.0

export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:/installed/torque-2.1.6/bin:$PATH

source .bashrc
```

RFT(Reliable File Transfer)와 WS GRAM audit logging에 의해서 사용될 데이터베이스로 postgresql을 필요로 한다. 레드햇 기반 배포판의 경우 postgresql이 rpm으로 배포되므로 설치되어 있지 않다면 rpm으로 설치해 준다.

```
rpm -qa | grep postgresql
postgresql-pl-7.4.17-1.RHEL4.1.i386
```

```
postgresql-python-7.4.17-1.RHEL4.1.i386
postgresql-7.4.17-1.RHEL4.1.i386
postgresql-devel-7.4.17-1.RHEL4.1.i386
postgresql-tcl-7.4.17-1.RHEL4.1.i386
postgresql-server-7.4.17-1.RHEL4.1.i386
postgresql-docs-7.4.17-1.RHEL4.1.i386
postgresql-libs-7.4.17-1.RHEL4.1.i386
freeradius-postgresql-1.0.1-3.RHEL4.5.i386
postgresql-contrib-7.4.17-1.RHEL4.1.i386
postgresql-jdbc-7.4.17-1.RHEL4.1.i386
postgresql-odbc-7.3-8.RHEL4.1.i386
postgresql-test-7.4.17-1.RHEL4.1.i386
```

postgresql의 서버 데몬인 postmaster가 TCP/IP를 통한 연결을 받아들이도록 하기 위해서 다음과 같이 수정한다.

```
vi /var/lib/pgsql/data/postgresql.conf
#tcpip_socket = false
tcpip_socket = true
```

모든 호스트가 postgresql에 접속할 수 있게 하기 위해서 다음과 같이 수정한다. 만약, GT4가 설치되는 호스트가 방화벽 내에 있다면, 방화벽에서 postgresql과 관련된 포트 (5432/tcp)를 열어준다.

```
vi /var/lib/pgsql/data/pg_hba.conf
local    all         all         ident      sameuser
host     all         all         0.0.0.0    0.0.0.0    trust
```

변경 사항을 적용하기 위해서 postgresql을 다시 시작한다.

```
/etc/rc.d/init.d/postgresql restart
```

GT4를 설치하기 위한 의존성 소프트웨어에 대한 설치는 모두 끝났으며 이제 GT4를 설치할 차례이다. GT4는 <http://www.globus.org>에서 다운로드 받은 후 globus 사용자 권한으로 설치한다. GT4 소스 설치 파일을 다운로드 받은 후, 다음의 과정을 통해서 설치한다.

```
tar xvzf gt4.0.6-all-source-installer.tar.gz
cd gt4.0.6-all-source-installer
./configure --prefix=/installed/globus-4.0.6
make 2>&1 | tee make.log
make install
```

GT4에 대한 환경 변수와 GT4가 설치된 디렉토리에 대한 경로를 다음과 같이 수정한다.

```
vi .bashrc
export GLOBUS_LOCATION=/installed/globus-4.0.6
export GPT_LOCATION=/installed/globus-4.0.6
. $GLOBUS_LOCATION/etc/globus-user-env.sh

source .bashrc
```

GT4는 GSI(Grid Security Infrastructure) 기반으로 동작하므로 그리드 CA 루트 인증서가 설치되어 있어야 하며, 사용자 인증서와 호스트 인증서가 발급되어야 한다. 그리드 CA 루트 인증서는 KISTI Grid CA 사이트(<http://ca.gridcenter.or.kr/certs/>)에서 다운로드 받을 수 있으며, /etc/grid-security/certificates에 설치되어야 한다. 위의 디렉토리를 확인하면, 722e5071.0과 722e5071.signing\_policy를 찾을 수 있다.

```
cd /etc/grid-security/certificates/
ls
43966fa9.0          globus-host-ssl.conf.43966fa9
43966fa9.signing_policy  globus-user-ssl.conf.43966fa9
722e5071.0        grid-security.conf.43966fa9
722e5071.signing_policy
```

GT4에서 사용할 사용자 인증서와 호스트 인증서는 KISTI Grid CA 사이트에서 요구하는 절차를 따르면 발급이 가능하다. 사용자 인증서는 일반적으로 ~/.globus 디렉토리에 usercert.pem, userkey.pem으로 설치되며 호스트 인증서는 /etc/grid-security/에 hostcert.pem, hostkey.pem으로 설치된다.

이 외에 GT4 서비스 컨테이너로서 /etc/grid-security/의 containercert.pem와 containerkey.pem을 호스트 인증서로 사용하기 때문에 GT4 서비스 컨테이너가 사용할 호스



트 인증서를 설치하기 위해서 다음과 같이 실행한다.

```
cd /etc/grid-security
cp hostcert.pem containercert.pem
cp hostkey.pem containerkey.pem
chown globus.globus container*.pem
```

GT4의 GridFTP 서비스를 xinetd를 이용하여 실행하기 위해서 /etc/xinetd.d에 gsiftp를 다음과 같이 생성한다.

```
vi /etc/xinetd.d/gsiftp
service gsiftp
{
disable = no
instances          = 100
socket_type        = stream
wait               = no
user               = root
env                += GLOBUS_LOCATION=/installed/globus-4.0.6
env                += LD_LIBRARY_PATH=/installed/globus-4.0.6/lib
server             = /installed/globus-4.0.6/sbin/globus-gridftp-server
server_args        = -i
log_on_success     += DURATION
nice               = 10
}
```

Xinetd를 다시 시작하기 위해서 다음과 같이 실행한다.

```
/etc/rc.d/init.d/xinetd restart
```

GT4의 RFT(Reliable File Transfer) 서비스를 설정하기 위해서는 RFT 데이터베이스 관리에 대한 권한을 globus 사용자에게 넘겨줄 필요가 있다. RFT가 사용할 데이터베이스에 대한 유저를 생성하기 위해서 postgres의 권한으로 다음과 같이 실행한다.

```
createuser globus -P
Enter password for new user: changeit
```

```
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) y
CREATE USER
```

RFT가 사용할 데이터베이스와 테이블 스키마를 생성하기 위해서 다음과 같이 실행한다.

```
createdb rftDatabase
psql -d rftDatabase -f /installed/globus-4.0.6/share/globus_wsrft/rft_schema.sql
```

postgresql을 다시 시작하기 위해서 루트의 권한으로 다음과 같이 실행한다.

```
/etc/rc.d/init.d/postgresql restart
```

JNDI를 통한 rftDatabase에 대한 접속을 허용하기 위해서jndi-config.xml을 다음과 같이 수정한다. 아래 파일에서 changeit을 다른 값으로 변경한다.

```
vi /installed/globus-4.0.6/etc/globus_wsrft/jndi-config.xml
<resource name="dbConfiguration"
  type="org.globus.transfer.reliable.service.database.RFTDatabaseOptions">
<resourceParams>
  ...
<parameter>
  <name>
    connectionString
  </name>
  <value>
    jdbc:postgresql://nova01.gridcenter.or.kr/rftDatabase
  </value>
</parameter>
<parameter>
  <name>
    userName
  </name>
  <value>
    globus
```

```
        </value>
    </parameter>
    <parameter>
        <name>
            password
        </name>
        <value>
            changeit
        </value>
    </parameter>
    ...
</resourceParams>
</resource>
```

WS GRAM audit logging을 활성화하기 위해서 GT4 서비스 컨테이너의 log4j 설정을 다음과 같이 추가한다.

```
vi /installed/globus-4.0.6/container-log4j.properties
# GRAM AUDIT
log4j.category.org.globus.exec.service.exec.StateMachine.audit=DEBUG, AUDIT
log4j.appender.AUDIT=org.globus.exec.utils.audit.AuditDatabaseAppender
log4j.appender.AUDIT.layout=org.apache.log4j.PatternLayout
log4j.additivity.org.globus.exec.service.exec.StateMachine.audit=false
```

audit logging이 사용할 데이터베이스와 테이블 스키마를 생성하기 위해서 다음과 같이 실행한다.

```
createdb auditDatabase
psql -d auditDatabase -f
/installed/globus-4.0.6/share/gram-service/gram_audit_schema_postgres-8.0.sql
```

postgresql을 다시 시작하기 위해서 루트의 권한으로 다음과 같이 실행한다.

```
/etc/rc.d/init.d/postgresql restart
```

GT4 서비스 컨테이너가 audit logging 데이터베이스에 접근하도록 하기 위해서 pg\_hba.conf를 다음과 같이 수정한다.

```
vi /var/lib/pgsql/data/pg_hba.conf
hostssl auditDatabase globus 150.183.225.74 255.255.255.255 md5
host auditDatabase globus 150.183.225.74 255.255.255.255 md5
```

JNDI를 통한 auditDatabase에 대한 접속을 허용하기 위해서 jndi-config.xml을 다음과 같이 수정한다. 아래 파일에서 changeit을 다른 값으로 변경한다.

```
vi /installed/globus-4.0.6/etc/gram-service/jndi-config.xml
<resource name="auditDatabaseConfiguration"
  type="org.globus.exec.service.utils.AuditDatabaseConfiguration">
  <resourceParams>
<parameter>
  <name>url</name>
  <value>jdbc:postgresql://nova01.gridcenter.or.kr/auditDatabase</value>
</parameter>
<parameter>
  <name>user</name>
  <value>globus</value>
</parameter>
<parameter>
  <name>password</name>
  <value>changeit</value>
</parameter>
</resourceParams>
</resource>
```

GT4 기반 클러스터 자원에 작업을 제출할 수 있는 권한을 설정하기 위해서 grid-mapfile과 sudoers를 수정할 필요가 있다. grid-mapfile은 /etc/grid-security/ 안에 생성해야 하며 사용자 인증서와 계정을 맵핑해주는 역할을 담당한다.

```
vi /etc/grid-security/grid-mapfile
"/C=KR/O=KISTI/O=GRID/O=KISTI/CN=79024450 JaeKyoung Song" globus
```

GT4 서비스 컨테이너는 일반적으로 globus에 의해서 실행되는데, 만약, 작업을 제출하는 유저가 globus가 아닌 경우에는 globus가 해당 유저의 권한으로 작업을 실행할 수 있도록 /etc/sudoers를 수정한다.

```
# Globus GRAM entries
globus ALL=(jhwak,hychul) \
NOPASSWD: /installed/globus-4.0.6/libexec/globus-gridmap-and-execute \
-g /etc/grid-security/grid-mapfile \
/installed/globus-4.0.6/libexec/globus-job-manager-script.pl *
globus ALL=(jhwak,hychul) \
NOPASSWD: /installed/globus-4.0.6/libexec/globus-gridmap-and-execute \
-g /etc/grid-security/grid-mapfile \
/installed/globus-4.0.6/libexec/globus-gram-local-proxy-tool *
```

GT4 서비스 컨테이너를 시작하고 종료하기 위해서는 다음과 같이 실행한다.

```
globus-start-container
globus-stop-container -soft
```

GT4가 제대로 설치되었는지 확인하기 위해서 다음과 같이 예제 작업을 제출해 본다.

```
grid-proxy-init
globusrun-ws -s -submit -c /bin/hostname
Delegating user credentials...Done.
Submitting job...Done.
Job ID: uuid:9be5324c-fbda-11dc-9008-0010dcff7dc4
Termination time: 03/28/2008 08:48 GMT
Current job state: Active
Current job state: CleanUp-Hold
nova01.gridcenter.or.kr
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
Cleaning up any delegated credentials...Done.
```

## 2. GRS 서비스 디플로이

먼저 웹포털이 사용할 WSRF 웹서비스 제공을 위해서 필요한 라이브러리를 GT4 설치 디렉토리의 lib/디렉토리에 복사해주어야 한다. 다음은 관련 라이브러리의 파일 목록이다.

- middleBfc.jar
- middle\_cluster\_manager.jar
- middle\_job\_manager.jar
- middle\_monitoring\_manager.jar
- middle\_project\_manager.jar
- tigris-parser.jar
- easyFW1.03\_20070822.jar

GRS 서비스로서 작업 관리 서비스와 자원 모니터링 서비스를 설치하기 위해서 다음과 같이 실행한다.

```
globus-deploy-gar com_kisti_middlews_jobmanager.gar
globus-deploy-gar com_kisti_middlews_monitoring.gar
```

## 제 3 절 NRM

NRM은 네트워크 자원 관리 시스템으로 GMPLS (Generalized Multi-Protocol Label Switching) 기반으로 네트워크 자원 사전 예약 기능을 제공한다. 현재 GRS에서는 NRM으로서 KISTI에서 개발한 dynamicKL을 연동하고 있으며, dynamicKL은 Apache Muse 프로젝트를 이용하여 WSRF를 통한 API 통신을 지원한다. 현재 dynamicKL은 다음의 웹서비스 주소에서 이용 가능하다.

```
http://203.230.116.198:8080/GNRM_SERVER/services/anuGNRM
```

NRM 서버와 통신하기 위해서는 다음과 같은 필수 라이브러리들이 GRS 구성 요소에 설치되어 있어야 한다.

- muse-wsrf-impl-2.2.0.jar
- muse-core-2.2.0.jar
- muse-util-2.2.0.jar

- muse-util-qname-2.2.0.jar
- muse-wsa-soap-2.2.0.jar
- muse-util-xml-2.2.0.jar
- muse-wsrf-api-2.2.0.jar
- xercesImpl-2.5.1.jar
- gnrm.jar (NRM API용 소스 코드)

## 제 4 절 클러스터

각 클러스터는 사용자 작업이 실제로 수행되는 곳으로 Globus Toolkit 4가 설치되는 것을 요구한다. Globus Toolkit 4의 설치에 앞서 언급하였으므로 여기서는 생략한다.

클러스터에 설치되는 GRS 서비스는 파일 관리 서비스와 자원 모니터링 서비스이다. 먼저 파일 관리 서비스를 설치하기 위해서 다음과 같이 실행한다.

```
globus-deploy-gar com_kisti_clusterws_filemanager.gar
```

자원 모니터링 서비스는 바이너리 파일을 클러스터에 복사하여 압축을 푸는데, 여기서는 /home/globus/monitoring에 설치하는 것으로 가정한다. 융합망의 클러스터 자원은 리눅스 기반으로 torque를 배치 스케줄러를 사용하므로 monitoring\_pbs.jar와 Linux\_Host\_info.sh를 필요로 한다. 먼저 resource\_monitor.sh를 수정하여 프로그램이 설치된 디렉토리에 맞게 수정한다. resource\_monitor.sh 파일을 GT4가 설치된 디렉토리의 /libexec/aggrexec에 복사한다.

```
vi resource_monitor.sh
#!/bin/sh
/home/globus/monitoring/program/monitoringStart.sh
```

monitoringStart.sh를 클러스터 자원에 맞게 수정한다.

```
vi monitoringStart.sh
#!/bin/sh

export CLASSPATH=%JAVA_HOME%/lib/tools.jar:%JAVA_HOME%/jre/lib/rt.jar:
/home/globus/monitoring/monitoring_pbs.jar:

java -DPROPERTY_PATH="/home/globus/monitoring/monitoring.properties"
```

```
com.kisti.monitoring.application.Monitoring
```

monitoring.properties 설정 파일을 클러스터 환경에 맞게 수정한다.

```
vi monitoring.properties
# 호스트네임 명령어 위치
hostname.info=/bin/hostname
# gridmap 파일 위치
gridmap.info=/etc/grid-security/grid-mapfile
# sw_info.sh 파일 위치
software.info=/home/globus/project/monitoring/program/sw_info.conf
# 자원정보 수집 스크립트 위치 및 스크립트 이름
host.info=/usr/bin/ssh {hostname}
/home/globus/project/monitoring/program/Linux_Host_info.sh
```

GT4가 설치된 디렉토리의 etc/globus\_wsrf\_mds\_index/jndi-config.xml을 편집하여 자원 정보 모니터링을 등록한다.

```
vi jndi-config.xml
<resource name="configuration" type="org.globus... ">
<resourceParams>
...
<parameter>
<name>executableMappings</name>
<value>resMonitorProvider=resource_monitor.sh</value>
</parameter>
</resourceParams>
```

클러스터의 PBS 배치 큐잉 스케줄러의 자원 정보는 GT4 설치 디렉토리의 libexec/globus-scheduler-provider-pbs를 통해서 수집된다. 클러스터의 Torque가 설치된 위치에 맞게 위의 파일을 수정한다.

```
vi globus-scheduler-provider-pbs
#!/usr/bin/env perl
```



```

# Scheduler commands used in the program
my $qstat_cmd = "/software/torque-2.3.3/bin/qstat -Qf";
my $qstat_queue_cmd = "/software/torque-2.3.3/bin/qstat -q";
my $qstat_version_cmd = "/software/torque-2.3.3/bin/qstat --version 2>&1 | awk
'(NR==1) {print $
2}' | grep -v invalid";
my $pbsnodes_cmd = "/software/torque-2.3.3/bin/pbsnodes -a";
my ($version, $hostname) = "";
...

```

참고로 globus-scheduler-provider-pbs에 의해서 생성되는 파일은 GT4에서 인식하는 scheduler schema 형태로 표현되며 MDS에 의해서 GLUE schema 형태로 변환된다.

```

./globus-scheduler-provider-pbs
<scheduler xmlns="http://mds.globus.org/batchproviders/2004/09"
  xmlns:ce="http://mds.globus.org/glue/ce/1.1"
  xmlns:cfg="http://mds.globus.org/2005/09/cluster-config" >
  <Info ce:LRMSType="PBS" ce:LRMSVersion="2.3.3" ce:GRAMVersion="4.0.8"
ce:HostName="nova01.gridcenter.or.kr" ce:TotalCPUs="16" />
  <Queue name="batch">
    <totalnodes>16</totalnodes>
    <freenodes>16</freenodes>
    <maxCount>16</maxCount>
    <maxTotalMemory>unlimited</maxTotalMemory>
    <maxSingleMemory>unlimited</maxSingleMemory>
    <maxCPUtime>unlimited</maxCPUtime>
    <maxtime>6000</maxtime>
    <maxReqNodes>16</maxReqNodes>
    <runningJobs>0</runningJobs>
    <maxRunningJobs>unlimited</maxRunningJobs>
    <maxJobsInQueue>unlimited</maxJobsInQueue>
    <whenActive>unknown</whenActive>
    <dispatchType>batch</dispatchType>
    <totalJobs>0</totalJobs>
  </Queue>
</scheduler>

```

```
<status>enabled</status>
</Queue>
</scheduler>
```

클러스터에서 수집된 정보를 CRM의 MDS에 등록하기 위해서 GT4 설치 디렉토리의 `etc/globus_wsrif_mds_index/hierarchy.xml`을 다음과 같이 수정한다.

```
vi hierarchy.xml
<config>
<upstream>https://150.183.250.80:8443/wsrif/services/DefaultIndexService</upstream>
</config>
```

각 클러스터에는 GT4 서비스 컨테이너 시작과 클러스터 자원정보를 클러스터의 MDS에 등록하기 위한 스크립트인 `globus-start.sh`가 `/home/globus/monitoring/env/`에 설치되어 있으며 `mds-servicegroup-add`에서 클러스터의 MDS 정보가 제대로 되어 있는지 확인한다.

```
vi globus-start.sh
#!/bin/sh
grid-proxy-destroy >/dev/null 2>&1
sh container-proxy.sh

globus-start-container > globus-start-container.log 2> globus-start-container.err &
sleep 10
mds-servicegroup-add -s https://150.183.24.2:8443/wsrif/services/DefaultIndexService
resMonitorProvider-registration.xml > mds-servicegroup-add.log 2>
mds-servicegroup-add.err &
echo $! > mds-servicegroup-add.pid
```

## 제 5 장 결론

지금까지 과학기술자원융합망 테스트베드 구축 사례를 소개하고 이를 구축하기 위한 기술적 방법에 대해서 살펴보았다. 과학기술자원융합망은 그리드 컴퓨팅 기술을 기반으로 고성능의 컴퓨팅 자원과 고 대역폭의 네트워크 자원을 연계하여 서비스하기 위한 차세대 연구 환경으로서 특히, 대규모 대용량의 데이터 분석 처리를 요구하는 과학 응용 연구 분야에서 핵심 역할을 담당할 수 있을 것으로 기대된다.

과학기술자원융합망 테스트베드를 구축하는 과정에서 몇가지 문제점도 도출되었다. 특히, 네트워크 자원의 사전 예약을 위한 GMPLS 기능에서 아직까지 안정적이지 못한 부분이 존재하는 것으로 파악되며 컴퓨팅 자원과 네트워크 자원을 동시에 다루는 과정에서 문제 발생 시에 원인 파악이 어려운 경우도 다수 발견되었다.

과학기술자원융합망은 연구자에게 보다 나은 연구 환경을 제공하기 위한 새로운 시도인 만큼 앞으로 더 많은 연구 개발이 필요할 것으로 생각된다.