

# 원격 SGE(Sun Grid Engine) 연동을 위한 Ganga 플러그인 설계 및 구현

김한기<sup>o</sup>, 황순욱

한국과학기술정보연구원 기반기술개발실

[hgkim@kisti.re.kr](mailto:hgkim@kisti.re.kr), [hwang@kisti.re.kr](mailto:hwang@kisti.re.kr)

## Design and Implementation of Ganga Plugins for Remote SGE

Hangi Kim<sup>o</sup>, Soonwook Hwang

KISTI, Dept. of Infrastructure Technology Development

### 요 약

고에너지 물리나 바이오 인포매틱스 연구 분야는 대량의 컴퓨팅 자원을 필요로 한다. 이를 위해서 클러스터 컴퓨팅, 그리드 컴퓨팅, 슈퍼 컴퓨팅, 클라우드 컴퓨팅 자원 등의 여러 컴퓨팅 자원이 사용된다. 사용자는 각 컴퓨팅 자원을 이용하기 위해서 각각의 사용 방법을 배우고, 저마다의 방법으로 각 컴퓨팅 자원을 사용한다. 사용자가 각 컴퓨터 자원의 사용 방법을 습득하기 위해서는 많은 비용과 시간이 필요하다. 사용자가 보다 쉽고 빠르게 컴퓨팅 자원을 활용하기 위하여 여러 컴퓨팅 자원을 사용하기 위한 동일 인터페이스 개발이 필요하다. 본 논문에서는 한국과학기술정보연구원의 타키온 슈퍼컴퓨터를 이용하는 사용자에게 동일한 사용자 인터페이스를 제공하기 위한 Ganga의 원격 SGE 백엔드(Backend) 플러그인을 설계하고 구현하였다.

### 1. 서 론

고에너지 물리나 바이오 인포매틱스 연구 분야는 대량의 컴퓨팅 자원을 필요로 한다. 이를 위해서 클러스터 컴퓨팅, 그리드 컴퓨팅, 슈퍼 컴퓨팅, 클라우드 컴퓨팅 자원 등의 여러 컴퓨팅 자원이 사용된다. 사용자는 각 컴퓨팅 자원을 이용하기 위해서 각각의 서로 다른 사용 방법을 배우고, 저마다의 방법으로 각 컴퓨팅 자원을 사용한다. 사용자가 각 컴퓨팅 자원의 사용 방법을 습득하기 위해서는 많은 비용과 시간이 필요하다. 사용자가 보다 쉽고 빠르게 컴퓨팅 자원을 활용하기 위하여 여러 컴퓨팅 자원을 사용하기 위한 동일 인터페이스 개발이 필요하다.

CERN[1]에서 개발된 작업 정의 및 관리도구인 Ganga[2]를 사용하면 클러스터 컴퓨팅, 그리드 컴퓨팅 등 여러 컴퓨팅 자원을 동일한 인터페이스로 사용할 수 있다. Ganga는 Sun에서 개발된 배치(Batch) 작업 스케줄러인 Sun Grid Engine(이하 SGE)도 현재 지원하고 있다. 하지만 Ganga가 설치되어 있는 로컬(Local) 서버에 SGE가 설치되어 있는 경우에만 사용가능하기 때문에 SGE가 원격에 설치되어 있는 경우에는 Ganga를 이용할 수 없다. 본 논문에서는 한국과학기술정보연구원의 타키온(Tachyon) 슈퍼컴퓨터의 경우와 같이 원격지 서버에 SGE가 설치되어 있는 경우에도 Ganga에서 이를 이용할 수 있도록 원격 SGE 백엔드(Backend)를 설계하고 개발하였다.

본 논문의 구성은 다음과 같다. 2장에서는 CERN에서 개발된 Ganga에 대해서 알아보고, 3장에서는 타키온 슈퍼컴퓨터와 SGE에 대해서 알아본다. 4장에서는 원격 SGE 백엔드의 구현에 대해서 설명하고, 마지막으로 5장

에서 결론을 맺도록 하겠다.

### 2. Ganga

Ganga는 파이썬으로 구현된 작업 정의 및 관리 도구로 여러 컴퓨팅 자원을 백엔드로 사용할 수 있다. Ganga에서 하나의 작업은 그림 1과 같이 여러 개의 빌딩 블록으로 이루어져 있다.[3]

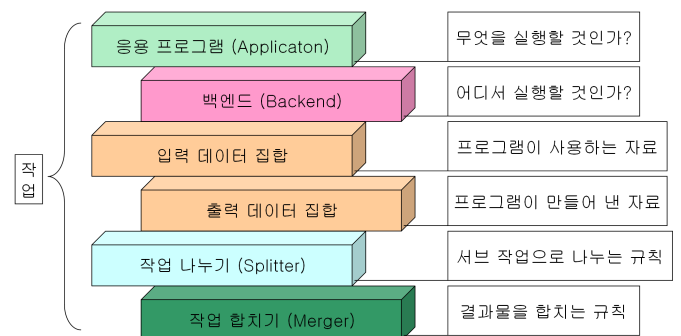


그림 1. Ganga의 작업 구성요소

모든 작업에는 실행 하고자 하는 응용 프로그램과 작업을 수행하고자 하는 백엔드(컴퓨팅 자원)가 정확히 명시되어야 한다. 여기에서 백엔드는 사용자의 응용 프로그램이 실제로 실행되는 시스템을 의미한다. Ganga에서는 응용 프로그램이 실행되는 백엔드를 로컬호스트, 배치 시스템, 그리드(Grid) 등으로 간편하게 변경해서 실행할 수 있다. Ganga는 “Configure once, run anywhere”, 즉 프로그램 실행에 필요한 설정사항을 한번

만 설정해 주면 여러 시스템 환경에서 프로그램을 동일하게 실행할 수 있는 환경을 제공해 주고 있다.

### 3. 타키온 슈퍼컴퓨터 & SGE

한국과학기술정보연구원 슈퍼컴퓨터 4호기 초병렬 컴퓨팅 시스템의 1차 시스템인 타키온(TACHYON)은 SUN Blade 6048 시스템을 기반으로 구성되었으며, 이론최고성능(Rpeak) 24TFlops를 보이고 있다. OS로 CentOS 4.6버전을 사용하고 있으며, 클러스터의 배치(Batch) 작업 스케줄러로 SGE 6.1 버전을 사용하고 있다.[4]

SGE는 Sun Microsystems에서 개발한 오픈 소스 기반의 배치-큐잉(batch-queuing) 시스템이다. SGE는 일반적으로 컴퓨터 팜(farm)이나 HPC(High-performance computing) 클러스터에 사용되며 사용자 작업을 받고, 스케줄링하고, 실행하고, 관리하는 작업을 처리한다.[5]

### 4. 원격 SGE 백엔드 구현

한국과학기술정보연구원 슈퍼컴퓨터 타키온 시스템을 이용하는 일반적인 방법은 그림 2와 같다.

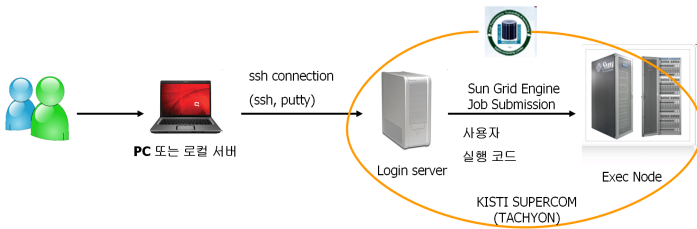


그림 2. 일반적인 타키온 시스템 사용방법

사용자는 먼저 자신의 PC나 로컬 서버에서 ssh 연결을 지원하는 터미널 프로그램(예:ssh, putty)을 사용해서 타키온의 로그인 노드(tachyon.ksc.re.kr)로 접속한다. 그리고 실행하고자 하는 사용자 실행 코드가 포함된 SGE 스크립트를 작성한다. 그리고 qsub 명령을 사용해서 계산 노드에 작업을 제출한다.

그림 3에서는 Ganga에서 원격 SGE 백엔드 플러그인을 사용해서 타키온 시스템을 사용하는 방법을 보여주고 있다.

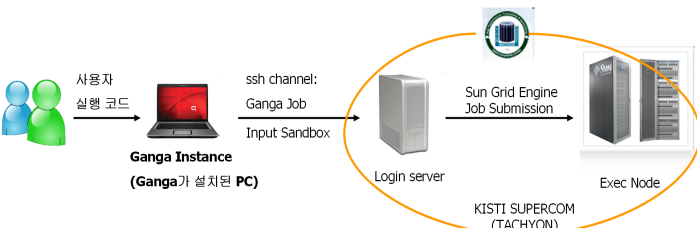


그림 3. 원격 SGE 백엔드 플러그인 방식의 타키온 시스템 사용방법

사용자는 Ganga 스크립트에서 백엔드를 SGE 백엔드로 지정해 주고, 로그인 노드 접속에 필요한 파라미터만 지정해 주면 나머지는 원격 SGE 백엔드 플러그인이 알아서 처리해 준다.

아서 처리해 준다.

원격 SGE 백엔드는 paramiko(SSH2 protocol for python) 라이브러리[6]를 사용해서 사용자 PC와 로그인 노드 사이에 ssh 채널을 만든다. 그리고 SGE 스크립트를 만들어서 로그인 노드로 업로드(upload)한 후, SGE 스크립트를 qsub 명령을 사용해서 계산 노드에 제출한다. Ganga에서는 주기적으로 로그인 노드를 통해서 SGE의 qstat 명령으로 작업 상태를 확인하며, 제출된 작업이 완료되면 작업 결과를 사용자 PC로 가져오고, ssh 채널을 닫는다.

#### 4.1 KSC 백엔드 스키마

원격 SGE 지원을 위한 KISTI Super Computing(이하 KSC) 백엔드는 host, ssh\_key, key\_type, username, ganga\_dir, queue\_name, password, subprocess\_dir, pre\_script, remote\_job\_id, actualCE라는 스키마로 구성되어 있다.

host 스키마에는 사용자가 접속하고자하는 SGE가 설치되어 있는 원격 호스트의 이름이나 주소를 설정해 준다.

ssh 접속은 사용자가 직접 패스워드를 입력하거나 key 파일을 만들어서 패스워드 입력 없이 접속할 수 있다. ssh\_key 스키마에는 ssh 접속에 필요한 key 파일의 위치를 지정해 준다. 이때 지정된 key 파일에 저장된 key 타입은 key\_type 스키마의 정보와 일치하여야 한다.

key\_type 스키마에는 ssh\_key에서 사용할 key 타입을 지정한다. 'RSA' 나 'DSS' 값을 사용할 수 있다.

username 스키마에는 host 스키마에서 설정된 호스트에 접속할 사용자 아이디를 지정해준다.

ganga\_dir 스키마에는 원격 호스트에서 사용될 여러 스크립트 파일(SGE 작업 제출 스크립트, SGE 작업 상태 확인 스크립트)이 저장될 원격 호스트의 디렉터리 경로를 지정해 준다. 지정된 ganga\_dir 디렉터리가 없는 경우 자동으로 생성된다.

SGE와 같은 배치 작업 스케줄러는 여러 개의 큐(queue) 구성을 가질 수 있는데 queue\_name 스키마에는 사용자가 작업을 제출하고자하는 큐의 이름을 지정해 준다.

password 스키마에는 username 스키마에 지정된 사용자의 암호를 설정한다. 만약 암호가 설정되어 있지 않으면 작업 제출 시에 암호 입력을 요구한다.

Ganga는 프로그램을 실행하기 위해 subprocess 라이브러리를 사용하는데 파이썬 2.4 버전에 새로 추가된 기능[7]이라 이전 버전에서는 따로 subprocess 라이브러리를 설치해 주어야 한다. subprocess\_dir 스키마에는 원격 호스트에서 subprocess 라이브러리가 설치되어 있는 디렉터리 경로를 지정해준다.

pre\_script 스키마에는 SGE 작업 제출 스크립트가 실행되기 전에 실행되어야 하는 명령어를 적어준다. 예를 들어 한국과학기술정보연구원 슈퍼컴퓨터 타키온의 경우, SGE에 작업을 제출하기 전에 스크래치 디렉터리로 이동해야 하므로 해당 명령을 pre\_script 스키마에 적어주면 된다. pre\_script 스키마는 문자열 배열로 실행하고자 하는 명령어들을 넘겨주면 하나의 셸(shell) 스크립

트로 만들어서 실행시켜준다.

remote\_job\_id 스키마에는 SGE에서 작업 관리 목적으로 사용하는 job-ID가 저장된다. 저장된 remote\_job\_id 정보는 Ganga에서 해당 Job 객체의 name 필드에 함께 저장되어 jobs 명령을 실행하면 화면에 표시된다.

actualCE 스키마에는 ssh 채널이 만들어진 호스트의 이름과 작업 큐 정보가 호스트 이름/작업 큐 이름의 형태로 저장된다. 예) tachyon.ksc.re.kr/long

#### 4.2 KSC 백엔드 필수 함수

Ganga에서 새로운 백엔드를 만들기 위해서는 Ganga에서 제공하는 IBackend라는 클래스를 상속해서 구현한다. 이 클래스를 상속받아 필수적으로 구현해야 하는 함수로는 submit(), resubmit(), kill(), updateMonitoringInformation() 등이 있다.

submit() 함수는 크게 두 개의 부분으로 구성된다. 첫째 부분에서는 host 스키마와 username 스키마 정보를 사용해서 원격 호스트와 사용자 PC 사이에 ssh 채널을 만든다. 만약 ssh\_key 스키마에 값이 지정되어 있으면 password를 묻지 않고 ssh 채널을 만든다. 하나의 작업(Ganga의 Job 객체)에 대해서 하나의 ssh 채널만이 만들어지며 원격 호스트에 제출된 작업이 완료되어 실행결과를 가져올 때까지 채널은 유지된다. 작업이 제출된 후, 원격 호스트의 SGE 큐의 상태 정보를 확인할 때에도 이 채널을 통해서 상태 확인 스크립트로 보내고 결과를 가져온다. 각각의 ssh 채널은 Job객체의 fqid 필드 값을 사용해서 구별된다.

두 번째 부분에서는 사용자 코드 실행에 필요한 모든 파일을 샌드박스(sandbox)라는 이름의 두 개의 압축파일로 만들고, 압축파일을 풀고 사용자 코드를 실행하는 파이썬 스크립트와 함께 원격 호스트에 전송한다. 사용자 코드 실행에 필요한 샌드박스과 파이썬 스크립트가 원격 호스트로 전송되고 나면, 다음으로 SGE 스크립트 템플릿과 SGE의 qsub 명령을 사용해서 SGE에 작업을 제출하기 위한 쉘 스크립트를 원격 호스트로 전송한다.

#### SGE 스크립트 템플릿

```
#!/bin/bash
#$ -q all.q
#$ -l h_rt=01:00:00
#$ -cwd
#$ -S /bin/sh
#$ -M hgkim@kisti.re.kr
#$ -m e
python /home/eairs/remote_jobs/__jobscript__HXIAOW
```

실행에 필요한 모든 파일이 원격 호스트로 전송되고 나면 SGE에 작업을 제출하기 위한 쉘 스크립트를 원격에서 실행한다. 쉘 스크립트가 정상적으로 실행되고 나면 실행결과를 파싱(parsing)하여 SGE에 제출된 작업을 추적하기 위해 필요한 job-ID 값을 알아낸 후, 이 값을 remote\_job\_id 스키마에 저장한다.

resubmit() 함수는 작업을 제출하는 과정에서 문제가 발생하여 Ganga 작업 상태가 new로 되어 있을 때, 다시 작업 제출을 요청하면 실행되는 함수이다.

resubmit() 함수가 호출되면 먼저 만들어진 ssh 채널이 있는지 확인하고, 없으면 새로운 ssh 채널을 만든다. 다음으로 압축파일을 풀고 사용자 코드를 실행하는 파이썬 스크립트를 원격 호스트에 전송한다. 그리고 난 후 SGE에 작업을 제출하기 위한 SGE 스크립트 템플릿을 원격 호스트로 전송한다. 다음 과정으로 qsub 명령으로 SGE 스크립트 템플릿을 SGE에 제출하기 위한 쉘 스크립트를 원격 호스트로 전송한 후, 이를 실행한다. 이후 과정은 submit() 함수와 마찬가지로 쉘 스크립트가 정상적으로 실행되고 나면 실행결과를 파싱하여 job-ID 값을 알아낸 후, 이 값을 remote\_job\_id 스키마에 저장한다.

kill() 함수는 Ganga에서 작업을 제거하는 명령을 실행하였을 때 호출되는 함수이다. Ganga에서 작업을 제거하기 위해서는 remove() 명령어를 사용한다. remove() 명령어를 통해 kill()함수가 호출되면 SGE 큐에서 작업을 제거하기 위한 qdel 명령을 호출하면 된다. qdel 명령의 인자로 필요한 SGE의 job-ID값은 KSC 백엔드의 remote\_job\_id 스키마에 저장되어 있기 때문에 qdel 명령의 인자로 이 값을 넘겨주면 된다.

kill() 함수가 호출되면 먼저 remote\_job\_id 스키마에 job-ID값이 설정되어 있는지를 확인한다. job-ID값이 존재하면 qdel 명령을 실행하기 위한 쉘 스크립트 파일을 만들어서 원격 호스트로 전송하고, 이를 실행한다. 그리고 난 후 qdel 명령이 성공적으로 끝났는지, 아니면 오류가 나서 제대로 끝나지 못했는지에 대한 상태 정보를 리턴하게 된다.

updateMonitoringInformation() 함수는 작업의 상태가 어떻게 바뀌어 가는 지 체크하면서 상태가 바뀔 때마다 상태 정보를 갱신해주는 일을 담당하는 함수이다. 이 함수는 Ganga에서 주기적으로 실행되면서 상태 정보를 갱신하도록 되어 있다. SGE에서는 작업들의 상태 정보를 알아보기 위해 qstat이라는 커맨드 라인 명령어를 제공한다. 이 명령어를 실행하면 SGE로 제출된 모든 작업들의 상태 정보를 확인할 수 있다. 이 결과를 파싱하여 Ganga의 job id와 맞는 작업의 실제 실행 호스트와 상태 정보를 갱신한다. 그리고 작업이 완료 되었으면 사용자가 지정한 아웃풋 파일들을 Ganga의 워킹 디렉터리의 output 디렉터리로 옮겨오는 작업을 수행한다. updateMonitorInformation은 인스턴스를 만들지 않아도 실행 가능한 static 함수로 선언해야 하므로 KSC 백엔드 클래스에 updateMonitoringInformation = staticmethod(updateMonitoringInformation) 이라는 문구를 추가해 주어야 한다. 다음 글상자 1에서는 KSC 백엔드에서 상태 정보를 갱신하기 위해서 사용된 qstat 명령의 출력 예제를 보여주고 있다.

글상자 1. SGE qstat 명령의 출력예제

job-ID	prior	name	user	state
submit/start at	queue	slots	ja-task-ID	
3667	0.55500	__jobscrip	eairs r	04/15/2010
15:43:24	all.q@compute-0-3.local	1		

리눅스의 echo 명령을 사용해서 화면에 "Hello World" 를 찍어주는 간단한 Ganga 스크립트를 가지고 일반적인 원격 SGE용 Ganga 스크립트와 슈퍼컴퓨터 타키온용 Ganga 스크립트의 차이점을 비교해 보면 다음과 같다.

슈퍼컴퓨터 타키온의 경우 SGE에 작업을 제출하기 전에 현재 디렉토리를 스크래치 디렉터리로 설정해 주어야 하므로 pre\_script 스키마에 필요한 명령어를 적어 주어야 한다. 그리고 타키온에는 파이썬 2.3이 설치되어 있으므로 subprocess 라이브러리를 사용하기 위해서 subprocess\_dir 스키마에 subprocess 라이브러리가 설치된 위치를 설정해 주면 된다.

글상자 2. 일반적인 원격 SGE용 Ganga 스크립트

```
j=Job()
j.application=Executable(exe='/bin/echo',args=['Hello World'])
j.backend=KSC()
j.backend.host="remote SGE hostname"
j.backend.username="userid"
j.backend.password="*****"
j.backend.ganga_dir="/home/userid/remote_jobs"
j.backend.queue_name="all.q"
```

글상자 3. 슈퍼컴퓨터 타키온용 Ganga 스크립트

```
j=Job()
j.application=Executable(exe='/bin/echo',args=['Hello World'])
j.backend=KSC()
j.backend.host="tachyon.ksc.re.kr"
j.backend.username="userid"
j.backend.password="*****"
j.backend.ganga_dir="/work01/userid/remote_jobs"
j.backend.queue_name="long"
j.backend.pre_script=['cd /work01/userid/']
j.backend.subprocess_dir=
"/work01/userid/Ganga/external/subprocess/2.4.2/noarch/lib/python2.2/site-packages"
j.submit()
```

## 5. 결론

이상과 같이 한국과학기술정보연구원(KISTI)의 타키온 슈퍼컴퓨터를 이용하는 사용자에게 동일한 사용자 인터페이스를 제공하기 위한 원격 SGE 백엔드(Backend)의 설계 및 구현에 대해서 알아보았다. 개발된 SGE 백엔드 플러그인은 새로운 Ganga 배포판에 추가될 예정이다.

## 참고문헌

- [1] The European Organization for Nuclear Research, [www.cern.ch](http://www.cern.ch)
- [2] Ganga, <http://ganga.web.cern.ch/ganga/>
- [3] J.T.Mościcki, F.Brochu, J.Ebke, U.Egede, J.Elmsheuser, K.Harrison, R.W.L.Jones, H.C.Lee, D.Liko, A.Maier, A.Muraru, G.N.Patrick, K.Pajchel, W.Reece, B.H.Samset, M.W.Slater, A.Soroko, C.L.Tan, D.C.Vanderster, M.Williams, "Ganga: a tool for computational-task management and easy access to Grid resources", Computer Physics Communications, Volume 180, Issue 11, (2009)
- [4] KISTI 슈퍼컴퓨팅 센터, <http://www.ksc.re.kr>
- [5] [http://en.wikipedia.org/wiki/Sun\\_Grid\\_Engine](http://en.wikipedia.org/wiki/Sun_Grid_Engine)
- [6] paramiko, <http://www.lag.net/paramiko/>
- [7] Python v2.6.5 documentation, <http://docs.python.org/library/subprocess.html>