

그리드 서비스 기반 PSE 포탈 구축

김형준⁰ 권용원 류소현 정창성 고려대학교 대학원 전자컴퓨터학과
박형우* 한국과학기술정보연구원 슈퍼컴퓨팅센터
(hjkim⁰, messias, luco)@snoopy.korea.ac.kr, csjeong@charlie.korea.ac.kr, hwpark@hpcnet.ne.kr

Developing A Grid Service based PSE Portal

Hyung-Jun Kim⁰ So-Hyun Ryu Yong-Won Kwon Chang-Sung Jeong
Dept. of Electronic and Computer, Korea University
Park Hyung Woo*, Korea Institute of Science and Technology Information

요약

분산되어 있는 고성능 컴퓨팅 자원과 대규모 데이터를 효과적으로 활용하기 위해서 최근에 그리드 환경에 대한 연구가 매우 활성화되었다. 본 논문에서는 그리드 환경을 쉽게 이용하고 복잡한 문제에 대해서 병렬/분산 처리할 수 있는 여러 도구들을 편리하게 이용할 수 있도록 그리드 포탈을 구축하고 활용하는 과정에 대해서 연구하여 소개한다. 그리드 PSE 포탈 구조와 현재 요구되는 사용자 중심의 서비스에 대해서 연구하고, 그리드 포탈을 구축하기 위해서 GPKD, Myproxy, Java CoG을 이용하였으며, 병렬/분산 처리를 위한 프로그램으로서 Cactus를 사용하였다.

1. 그리드 환경과 PSE

그리드 환경은 지리적으로 분산되어 있는 고성능 계산 컴퓨팅 자원이나 대용량 데이터를 과학이나 공학 기술자와 연구자가 활용할 수 있도록 만들어진 분산 환경으로서 최근 이에 대한 연구가 매우 활성화되어 있다. 그리드 환경의 기본 서비스로 Globus 프로젝트[1]에서 제공하는 Globus Toolkit이 있는데, PKI를 기반으로 한 보안 시스템인 GSI는 사용자와 컴퓨팅 자원간에 상호인증을 통해서 보안 기능을 제공하고, 원격지 호스트에 프로그램을 실행할 수 있는 GRAM은 로컬 자원 관리 기능을 제공한다. 그리고, 호스트간에 파일을 전송할 수 있는 GridFTP와 메모리, CPU, 저장 장치 등 자원들의 상태를 제공하는 정보 서비스로써 MDS가 있다.

그리드를 사용자가 폭넓게 효과적으로 활용하기 위해서는 다양한 분야에 적합한 소프트웨어와 사용 환경이 구축되어야 한다. 사용자 계정을 만들고 호스트에 프로그램을 실행하거나 자원에 대한 정보 서비스 등과 같은 일반적인 그리드 서비스를 위해서 여러 그리드 포탈이 구축되어 있는데, 천체물리학에 관한 대규모 시뮬레이션을 위한 ASC Grid Portal[2], 고성능 컴퓨팅 자원을 이용할 수 있는 Hotpage[3], NASA의 Information Power Grid를 위한 Launch Pad[4], 영국의 HPC 자원을 위한 HPCGrid[5] 등 여러 나라와 기관에서 소유한 자원들을 효과적으로 활용하기 위해서 그리드 포탈을 구축하고 있다. 그리드를 보다 성공적으로 여러 분야에 활용하기 위해서는 각 분야에 맞는 문제를 해결하기 위한 소프트웨어들이 구축되어 쉽게 사용할 수 있어야 하는데, 이러한 환경을 PSE (Problem Solving Environment)라고 한다. PSE로서 Cactus[6]는 고속 연산이 필요한 과학분야의 계산문제를 해결하기 위해 분산 응용프로그램을 개발할 수 있는 환경을 제공하고, UNICORE[7]는 이기종 분산 환경에 쉽게 적용할 수 있는 단일하고 편리한 그래픽 사용자 환경을 제공한다. DISCOVER[8]는 고성능 병렬/분산 응용 프로그램의 활용을 위한 협업환경을 제공하는 것이 목적이며, GrADS[9]는 분산 이기종 환경에서 응용프로그램을 쉽게 개발할 수 있는 환경 구축에 목적을 두고 있다.

본 논문에서는 그리드 포탈을 이용하여 PSE 환경을 구축하

* 본 연구는 2002년 한국과학기술정보연구원의 그리드 미들웨어 연구인 어플리케이션을 위한 PSE 기술 과제에서 지원을 받았습니다

기 위한 도구와 방법을 연구하여 소개한다. 2절에서는 기존의 그리드 포탈 구축 기술에 대해서 설명하고 3절에서는 전체적인 구조와 필요한 서비스에 대해서, 4절에서는 구현 과정에 대해서 기술하고 5절에서는 이후 과제, 6절에서는 결과를 정리한다.

2. PSE 포탈 구축을 위한 기술

2.2 Myproxy [10]

Globus 서비스는 기본적으로 GSI라는 보안체계 안에서 운영이 되는데, GSI(Globus Security Infrastructure)는 PKI를 기반으로 하여 공개키 암호화 기술과 X.609 인증서, 그리고 SSL 통신 프로토콜을 사용하고 있다. 기본적으로 Globus를 이용하기 위해서는 GSI를 통한 상호 인증을 받아야 하고, 인증을 받으면 프락시가 생성되어 이후 다른 여러 자원을 사용하는데 사용자가 직접 인증을 하지 않고 프락시가 인증을 맡게 된다. 이러한 보안 체계를 그리드 포탈에 적용하기 위해서는 사용자의 인증서를 온라인 상에 저장해 두어야 포탈 서버에 프락시를 생성하여 놓을 수 있다. 이것을 지원하는 프로그램이 Myproxy이다. 이것은 클라이언트와 서버로 구성되어 있고, myproxy-init 프로그램으로 서버에 사용자의 인증서를 저장할 수 있고, 포탈 서버에서는 myproxy-get-delegation 프로그램을 통해서 사용자의 프락시를 얻을 수 있다.

2.3 GPKD (Grid Portal Development Kit) [11]

GPKD는 자바로 제작된 그리드 포탈 구축 도구로써 JSP, Java Servlet, Java Beans, Java CoG로 제작되었다. 가장 핵심적인 역할을 하는 것은 Java CoG로서 그리드 서비스를 이용할 수 있도록 자바로 작성된 라이브러리이다. 시스템은 크게 2부분으로 나뉘어 지는데, Grid Service Beans 부분은 Java CoG[12], LDAP SDK, SSH등을 이용하여 그리드 환경에서 정보 서비스, 작업 제출(Job Submission), 파일 전송등의 저수준 서비스를 할 수 있도록 작성된 자바 빈즈로 구성되고 포탈을 구성하는 부분은 MVC 패턴을 적용하여 다시 3가지 요소인, Portal Engine, Logic, Presentation으로 나뉘어져 있다. Portal Engine은 서버로부터 클라이언트의 요청에 따라 필요한 Logic을 구동시키고, Logic 부분은 자바 클래스로 사용자 프로파일, 인증, 작업제출, 정보 서비스, 파일 전송 등의 그리드 서비스를 제공하며, Presentation은 JSP로 작성되어 클라이언트 인터페이

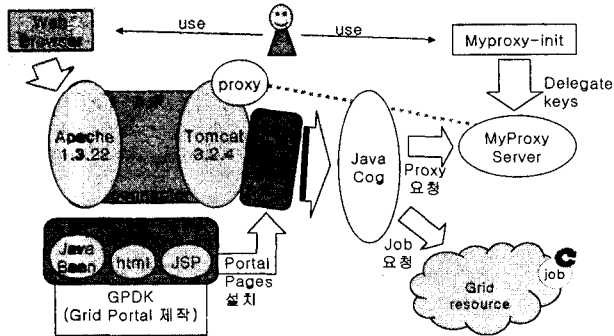


그림 1. 시스템 구조

스와 요청의 결과를 보여준다. 새로운 기능을 작성하기 위해서는 이 모델에 따라 Logic 부분과 Presentation 부분에 코드를 추가하면 된다.

GPDK에서 제공하는 기본 기능은 Myproxy를 이용한 인증 과정, 생성된 사용자의 프락시, 사용할 호스트 리스트, 작업 제출 이력 등의 사용자 환경 관리, 작업 제출과 그 결과 보기, 파일 전송, 정보 검색이 있다. 그러나 Globus 2.0의 GridFTP, MDS 2.1등은 지원하지 않고 있지 않다.

3. PSE 포탈 구조와 기능

3.1 시스템 구조

그리드 포탈은 웹 기술을 이용하여 그리드 자원을 이용하기 위한 인터페이스로서 기본적으로는 사용자 GUI인 웹 브라우저, 그리드 서비스 인터페이스, 그리드 자원 및 응용프로그램, 3계층으로 이루어진다. 그림 1의 예와 같이 그리드 서비스 인터페이스는 그리드 포탈 서버로 구현되는데, GPDK를 바탕으로 하여 아파치 웹서버와 서블릿을 위한 Tomcat, Globus 서비스를 이용하기 위한 Java Cog로 구성된다. 그리고, 그리드 자원 및 응용프로그램에는 Globus 서비스와 Cactus를 설치하여 병렬/분산 응용프로그램을 사용할 수 있도록 한다.

3.2 포탈 서비스

사용자를 위한 포탈 서비스로는 사용자 환경 관리와 파일 편집 기능, GridFTP를 이용한 파일 전송, MDS를 통한 정보 검색, GRAM을 이용한 프로그램 실행과 결과 보기가 있다. 사용자 환경 관리에서는 사용자를 인증하여 프락시를 생성하고 사용할 가능한 호스트 리스트를 저장하고 요청한 작업 이력을 저장하여 보여준다. 프로그램이나 스크립트 파일을 작성하기 위해서 원격지 호스트에 있는 파일 편집 기능이 필요하여 그리드 포탈 서버에 어떤 호스트의 파일을 복사하여 두고 편집한 다음 원본을 갱신할 수 있는 파일 편집 기능을 제공한다. 병렬/분산 프로그램을 실행하기 위해서는 실행 파일을 여러 호스트에 배치해야 하기 때문에 GridFTP를 이용하여 파일을 전송할 수 있는 기능을 제공하고, 이용 가능한 자원의 목록과 자원에 대한 사양이나 상태 정보를 얻기 위해서 MDS 2.1를 이용한 정보 검색 서비스를 지원한다. 프로그램 실행을 위해서 GRAM을 이용한 작업 제출 기능을 제공하며 그 결과를 보여준다.

3.3 Cactus

포탈을 통해 사용자는 웹 인터페이스를 이용하여 그리드 자원에 접근하는 것이 가능해진다. 여기에 주어질 문제를 풀기 위한 그리드 기반 환경인 Cactus를 연계함으로써 PSE 포탈의 기본구조가 완성된다. Cactus는 경계조건방정식동 격자점 단위의 수치해석에 특화된 문제 해결 환경이다. 다양한 수학적 틀

뿐만 아니라 그리드 자원상의 연산 병렬화를 자동 제어해 주는 기반구조를 제시하며 포트란 및 C, C++등의 다양한 언어를 이용 가능하다는 특징이 있다. 따라서 분산 병렬 응용프로그램의 개발에 필요한 부가적 지식 없이도 보다 쉽게 사용자가 자신에게 익숙한 언어로 그리드 기반 문제 해석도구를 개발하는 것이 가능하다. 이러한 장점을 획득하기 위해 Cactus는 특유의 thorn-flesh 구조를 통해 각각을 기능별 모듈화 하여 사용자에게 제공한다.

Cactus는 문제풀이에 사용되는 수학적 도구부터 그리드 기반의 분산환경을 제어하는 기능등 전체적인 기능을 thorn이라 불리는 객체지향성을 띄는 프로그램 모듈로 분리하여 제공한다. 사용자는 필요에 따라 여러 가지 기능을 갖는 thorn을 Cactus flesh라 불리는 코드구조를 중심으로 연결하여 하나의 분산 응용프로그램을 완성한다. 사용자는 자신의 목적에 맞는 프로그램 코드를 개발한 후 여기에 Cactus configuration language를 이용한 설정 파일을 덧붙임으로써 자신만의 thorn을 개발할 수 있으며 이런 사용자 thorn은 Cactus에서 제공하는 다양한 도구용 thorn들의 집합인 infrastructure thorn arrangement의 기능을 차용하여 이용하는 것이 가능하다.

사용자는 자신이 만든 문제해석 코드를 설정파일에 하나의 thorn으로 선언하고 코드의 초기 상태를 결정짓는 매개변수를 지정하고 flesh 코드가 제공하는 동작 상태에 대한 정보에 따라 자신의 thorn의 동작 순서를 스케줄링하면 간단히 자신만의 고유의 thorn을 완성할 수 있다. thorn은 자신이 연산시 이용할 메모리 영역으로 Cactus variable을 선언하는 것이 가능하며 이것은 실제 동작시 병렬화 하여 동작하는 각 프로세서간에 통신을 통해 경계영역이 동기화 될 수 있는 공유 메모리 영역으로 이용된다.

flesh 코드는 각각의 thorn을 연결하여 하나의 분산응용프로그램으로 만들어주는 핵심을 이룬다. flesh는 각 thorn의 설정 파일의 내용에 따라 구성 thorn들의 기능별 동작 순서를 스케줄링 할 뿐 아니라 고유의 variable을 통해 응용프로그램의 실행 환경에 대한 정보를 thorn들에게 제공하기도 한다.

사용자 thorn과 infrastructure thorn, flesh 코드가 완성된 하나의 실행파일로 컴파일되며 그리드 환경 상에서 Globus를 이용해 구동된다. Cactus는 할당된 자원과 초기값으로 제안된 계산영역의 범위를 통해 병렬화 되어 동작하는 각각의 프로세서에 업무를 분담하고 스케줄에 따른 연산과 variable의 경계영역 동기화를 진행하여 결과를 도출하게 된다.

4. 구현

4.1 기본적인 포탈 환경 구축

그리드 환경을 구축하기 위해서 기본적인 서비스로서 Globus Toolkit을 설치하는데 Myproxy나 Cactus 등을 이용하기 위해서는 소스 버전을 설치하고, 그 과정은 [1]나 [13]를 참고한다. Globus를 설치한 후에는 GSI에 의한 인증을 위해서 호스트와 사용자, MDS나 그리드 포탈 서버 등의 각 서비스에 해당하는 인증서와 키를 공인 CA로부터 받아야 하는데 실험 단계임으로 [14]에서 CA 프로그램을 받아서 설치하여 사용한다.

GPDK를 사용하기 위해서는 기본적으로 아파치 웹 서버, tomcat 3.2.4, 자바 SDK 1.3, Ant, Myproxy, OpenSSH를 설치해야 한다. 각각의 설치과정은 배포하는 설명서를 따르면 되고, 아파치 웹서버를 https로 접속하고 GPDK에서 Java CoG를 사용하기 위해서는 그리드 포탈 서버에 해당하는 인증서와 키를 만들어서 아파치는 httpd/conf 아래에 설치하고 GPDK는 그 위치를 gpdk.properties 파일에 설정해야 한다. 또한 CA 인증서의 위치를 설정하는 것도 잊지 말아야 한다.

Myproxy 서버는 Java CoG에서 제공하는 클라이언트로 접

속하는 경우 인증에서 에러가 발생하는 경우가 있다. 그래서 GPDK에서 만들어주는 데모 사이트에서 로그인 되지 않는 경우가 많은데, Myproxy 서버가 GRAM의 gatekeeper의 인증서를 이용하기 때문에 루트권한으로 실행하는지 확인하거나 다른 버전을 설치하여 본다. 본 논문의 시험 시스템에서는 Myproxy 0.4.3 버전으로 성공을 하였다.

OpenSSH[15]는 gpt 1.0을 사용하는 Globus와는 달리 gpt 2.0이상으로 설치해야 한다. 그런데, GPDK에서는 GSISsh를 사용하고 있고 Globus에서는 최근에 OpenSSH를 권장하고 있어서 GPDK 소스 수정이 필요하고 GSISshSubmissionBean의 runBlockingJob() 멤버 안에서 Globus 설치 디렉터리 위치를 환경변수로 넣어주어야 한다.

사용자가 원하는 프로그램의 실행은 GPDK에서 제공하는 작업 제출 인터페이스를 통해서 할 수 있었으며, GPDK에서 제공하는 소스를 바탕으로 필요한 부분만을 수정하고 추가하여 구현한 새로운 기능을 소개한다. Cactus 코드는 [6]로부터 얻어서 설치하고 작업 제출 기능으로 사용할 수 있다.

4.2 MDS 2.1 정보 서비스

MDS 2.1을 이용하여 여러 자원의 정보를 얻기 위해서는 Globus Toolkit을 설치한 후에 GIIS를 구축해야 한다. GIIS는 자원을 트리 형태로 묶어서 여러 자원에 대한 정보를 제공한다. GPDK에서는 Netscape LDAP SDK과 MDS 2.1 이전 버전을 사용하고 있어서 데모 사이트의 정보 검색 기능인 Resource Info와 Queue Info 부분은 동작하지 않는다. 따라서 MDS 2.1 스키마를 이용하도록 코드를 수정해야 한다. MDS 2.1을 이용하여 정보 검색을 하기 위해서 Java Cog의 MDS 클래스의 search 멤버를 이용하고 MDS 2.1 스키마로 질의를 작성하면 된다. GPDK 데모 사이트에 정보 검색 버튼을 추가하기 위해서는 gridservices.html 파일에 버튼과 버튼의 "action" 변수값을 추가하고 pages.config 파일에 버튼에 해당하는 action 값과 Logic에 해당하는 빈즈 이름과 JSP 파일 이름을 추가한다. 빈즈는 BasicPage를 사용하고 JSP 파일에는 java CoG의 MDS 클래스를 이용하여 정보 검색 기능을 작성하였다.

4.3 GridFTP를 이용한 파일 전송

Globus 2.0에서는 GridFTP를 제공하지만, GPDK에서는 GSIFTP를 이용하고 있어서 GridFTP를 이용한 파일 전송을 할 수 없다. 이를 지원하기 위해서 jftp[16]라는 자바로 작성된 GridFTP 클라이언트를 이용한다. jftp를 사용하기 위해서는 CoG 0.9.13이 필요한데, GPDK에서는 globus.jar 라는 파일로 0.9.12를 사용하고 있어서 0.9.13 버전을 컴파일하여 globus.jar 대신 이용하면 된다. 이때 버전의 차이로 GPDK에서 에러가 발생하는데, RslAttributes의 패키지 이름을 org.globus.common에서 org.globus.rsl로 바꾸면 된다. jftp의 GridFTPClient 클래스의 transfer() 멤버를 이용하여 원거리 호스트간의 파일 전송이 가능하지만 이 부분에서도 2 가지 수정 부분이 있는데, jftp의 소스 중에서 TestEnv라는 클래스 객체를 사용하는 부분을 지우는 것과 GPDK에 로그인한 사용자의 프락시를 기본 프락시로 하기 위해 GlobusProxy.setDefaultUserProxy()를 호출하는 것이다. 그렇지 않으면 그리드 포탈 프락시를 이용하여 원거리 호스트에 접속하기 때문에 인증 에러가 발생한다.

4.4 파일 편집 기능

원거리 호스트에 있는 파일의 편집 기능을 제공하기 위해서 복사본을 그리드 포탈 서버에 만드는 기능을 구현하였다. 복사본을 사용자가 다운로드 받아서 편집한 후에 다시 업로드하고 복사본을 통해서 원본 파일을 갱신할 수 있다. 또는 웹 브라우저에서 직접 편집하고 저장할 수 있다. 파일을 업로드에 Multipart request를 이용하기 위해서 MultiPartRequest 클래스를 사용하였다. 사용법은 [17]에서 쉽게 알아 볼 수 있으나, GPDK에서 적용하기 위해 소스 수정이 필요하다. GPDK에서는 "action"이라는 변수의 값을 통해서 요청한 기능을 선택하는데 MultiPart request로 "action"의 값을 읽기 위해서는 MultiPartRequest 클래스를 이용해야 하기 때문에 이 부분을 지원해야 한다. 그래서, GPDK의 Logic 부분에 해당하는 PageCommand와 Page 클래스의 execute() 멤버에 HttpServletRequest와 MultiPartRequest를 모두 인자로 주어서 요청의 형태에 따라서 변수값들을 읽을 수 있도록 변경하였다.

5. 결과 및 향후 과제

본 논문에서는 그리드 환경에서 응용프로그램을 이용하기 위한 그리드 PSE 포탈 환경을 구축하기 위해서 GPDK, Myproxy, Cactus등의 기술과 전체 시스템 구조와 필요한 기능에 대해서 연구하여 설명하였으며, 새로운 기능 구현을 위한 과정 및 발생한 에러에 대한 처리 방법을 설명하여 보다 쉽고 빠르게 그리드 포탈을 구축할 수 있도록 연구자들에게 도움을 줄 수 있을 것이다. 앞으로 보다 편리하고 강력한 PSE 포탈을 구축하기 위해서 보다 많은 PSE 도구들을 사용할 수 있도록 기능을 확장하고 사용자가 편리하고 쉽게 이용할 수 있도록 다양한 시각적인 도구들을 제작하여 지원할 것이다.

참고 문헌

- [1] Globus Project, <http://www.globus.org>
- [2] Astrophysics Simulation Collaboratory: ASC Grid Portal, <http://www.ascportal.org>
- [3] HotPage, <http://hotpage.npaci.edu>
- [4] LaunchPad, <http://www.ipg.nasa.gov/launchpad/servlet/launchpad>
- [5] UK HPCGrid, <http://esc.dl.ac.uk/HPCPortal>
- [6] Cactus Code, <http://www.cactuscode.org>
- [7] Romberg, M., "The UNICORE Architecture Seamless Access to Distributed Resources", High Performance Distributed Computing, pp.287-293, 1999,
- [8] DISCOVER, <http://www.caip.rutgers.edu/TASSL/Projects/DISCOVER/DISCOVER.html>
- [9] GrADS 프로젝트, <http://nhse2.cs.rice.edu/grads/index.html>
- [10] Myproxy, <http://dast.nlanr.net/Projects/MyProxy>
- [11] J. Novotny., "The Grid Portal Development Kit", Concurrency: Pract. Exper. Vol. 00, pp.1-7, 2000
- [12] Gregor von Laszewski, "A Java Commodity grid kit", Concurrency: Pract. Exper. Vol. 13, pp.645-6622001
- [13] KISTI Grid Testbed, <http://gridtest.hpcnet.ne.kr/>
- [14] CA 프로그램, <ftp.psy.uq.oz.au/pub/Crypto/SSL/>
- [15] OpenSSH, www.ncsa.uiuc.edu/Divisions/ACES/GSI/openssh/installgpt.html
- [16] jftp, <http://www-unix.globus.org/cog/jftp/index.html>
- [17] Multipartrequest 예제, <http://aboutjsp.com/lec/multipart.jsp>