

NT 컴퓨팅 그리드 PSE 포탈 서버 구축

허정민, 김윤희

숙명여자대학교 정보과학부
{jjung1022, yulan}@sookmyung.ac.kr

Developing a NT Computing Grid PSE Portal Server

Jung Min Huh, Yoonhee Kim
Dept. of Computer Science, Sookmyung Women's University

요 약

그리드(Grid)[1] 환경은 분산되어 있는 고성능 컴퓨터 자원과 방대한 데이터를 네트워크로 상호 연동하여 지역적으로 분산되어 있다는 제약을 받지 않고 효과적으로 활용할 수 있는 환경으로 최근에 많은 연구가 이루어지고 있다. 이러한 그리드 기술은 고성능 및 대용량의 계산 능력을 필요로 하는 과학 및 공학 응용 연구와, 상호 협력 작업을 가능하게 하는 협업 환경을 가능하게 해준다. 따라서 그리드 기술을 기반으로 분산된 자원을 사용자가 쉽게 사용할 수 있도록 해주는 문제해결환경(PSE : Problem Solving Environment)에 대한 개발이 요구되고 있다. 이 논문은 현존하는 PSE의 기능을 분석하고, NT응용에서 사용할 수 있는 포탈을 구축하고 활용하는 과정에서 실제 그리드 환경을 제공하는 서버측의 내용을 기술하였다.

1. 서 론

그리드 환경은 네트워크로 연결되어 있는 많은 분산 컴퓨팅 자원을 효율적으로 활용하고 다양한 학문 분야의 복잡한 문제들을 풀기 위해서 필요한 고성능 계산 환경을 제공하는 기술로 지금 세계적으로 연구가 매우 활발하게 이루어지고 있다. 그리드 컴퓨팅에서 사용자가 이를 폭넓고 효과적으로 이용하기 위해서는 글로벌스와 같은 그리드 환경을 구성할 수 있는 미들웨어를 필요로 한다. 그리고 복잡하고 많은 양의 계산이 필요한 문제에 대해서 분산, 병렬 처리하여 작업을 쉽게 실행 할 수 있는 PSE환경이 요구된다.

그리드 테크놀로지의 응용 분야 중 하나인 나노 테크놀로지(Nano Technology; NT)는 계산 시에 많은 자원과 방대한 데이터의 계산을 필요로 하는데 본 논문에서는 나노 기술 응용 중 큰 분야를 차지하고 있는 VASP (Vienna Ab-initio Simulation Package)의 워크플로우(Workflow) 지원을 기반으로 하는 그리드 서비스 PSE인 CEGA(A Computational Environment for Grid Application)를 개발하였다. 우리는 CEGA를 NT응용을 하는 사용자들에게 제공하여 그들이 대량연산의 계산문제를 신속하게 처리할 수 있는 환경을 제공하는데 목적을 두고 있다.

2장에서는 그리드 포탈을 이용하여 PSE 환경을 구축한 대표적인 PSE에 대한 사례연구, 3장은 NT PSE 포탈에 대한 설명, 4장에서는 실제 구축된 그리드 환경에 대해 기술, 5장에서는 결과 및 향후과제를 정리한다.

2. 현존하는 PSE 사례

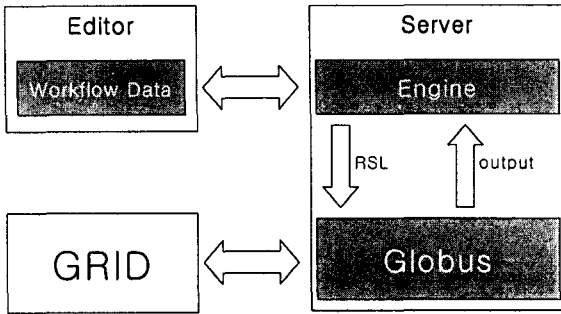
PSE는 네트워크로 연결된 고성능 컴퓨터와 장비 등의 자원을 활용하여 어플리케이션의 개발 및 실행을 효율적으로 수행할 수 있도록 한다. 최근에는 분산 그리드 시스템에 대한 웹 인터페이스를 제공하는 그리드 포탈에 대한 연구가 진행되고 있으며, 다양한 어플리케이션을 지원할 수 있는 포탈 구축 도구가 개발되고 있다. 대표적인 것으로 UNICORE와 Triana가 있다.

UNICORE는 서로 다른 환경 속에서 사용자 원하는 목적을 얻을 수 있도록 이기종 환경을 극복한 사용자 인터페이스를 제공하고 응용 개발에서 실행까지 5가지 단계로 구성되어있다.

Triana는 사용자에게 그래픽 서비스를 통한 개발 환경을 컴포넌트 기반을 이용해서 제공하며 그리드의 방대한 자원을 활용하여 동적으로 결과물을 수행하는 기능을 제공하는 분산 처리가 가능한 PSE다.[5]

3. 시스템 구조 및 제공되는 서비스

CEGA는 GUI를 통하여 사용자가 직접 원하는 작업을 워크플로우로 작성한 후 실행하게 되는 에디터 부분과, 그리드 환경에서 작업할 수 있도록 해주는 그리드 미들웨어인 글로벌스, 에디터와 글로벌스간의 통신을 해서 직접적으로 그리드 작업을 담당해주는 서버부분인 엔진으로 구성된다. 전체적인 시스템 구성은 [그림 1]과 같다.



[그림 1] 전체 시스템의 기본 구성

3.1 구축 환경

그리드 환경을 구축하기 위해서 기본적인 서비스로 Globus Toolkit을 설치[3]하였고 자체적인 CA를 설치하여 사용자와 호스트를 인증하도록 하였다. 이 인증 후에는 GSI에 의한 인증을 하게 되는 것이다. 이밖에도 기본적인 사용을 위하여 아파치 웹 서버, 톰캣, 자바 J2DK 1.4, Ant, Myproxy, MPICH를 설치하였다. 그리고 엔진 내부는 Java CoG Kit을 이용하였다.

3.2 에디팅 서비스

그리드 환경에서 분산된 자원을 텍스트 환경에서 사용할 경우 사용자는 복잡한 설계와 많은 시행착오를 겪어야 할 것이다. 그러나 CEQA의 경우 GUI를 제공하여 사용자가 쉽게 사용할 수 있도록 하였다. 에디터 상단에 자주 사용되는 아이콘을 배치하여 마우스 클릭 한 번만으로 필요한 기능을 사용하도록 했고 직접 그래픽컬한 워크플로우를 설계하여 작업을 실행할 수 있도록 하였다. 마우스를 이용하여 워크플로우를 구성하는 잡(job)들 간의 링크와 몇 가지 설정만으로 나타내는 패턴 표현법은 그래픽컬한 워크플로우의 장점을 살려 간단한 작업은 물론, 복잡한 작업(예, VASP실행에서 High Throughput 과정)도 사용자가 쉽게 작업을 할 수 있는 환경을 제공하여 그리드 환경에서 작업에 필요한 전문지식 없이도 쉽게 워크플로우를 작성하고 실질적인 잡을 수행할 수 있게 했다. 하나의 워크플로우 디자인을 통한 작업의 일괄 처리와 워크플로우 저장 기능은 반복적인 작업을 줄이게 되어 작업의 능률을 효과적으로 증가시킨다.

워크플로우 에디터는 사용자가 디자인한 워크플로우를 엔진으로 전송하여 글로버스를 통해 그리드의 분산된 자원을 활용하도록 하며 수행되는 작업의 모니터링 정보와 작업 로그 정보를 제공하여 사용자가 그 진행 상황을 일괄적으로 파악할 수 있게 하였다.

3.3 어플리케이션 익스큐팅 서비스

사용자가 에디팅 서비스를 통해 하나의 작업을 만들 때 서버에서는 임의의 클러스터를 사용하여 사용자의 잡을 돌리고, 모니터링 정보와 결과를 전송한다. 이 때 현재 사용가능한 자원들 중에서 제일 적은 수의 일을 돌리고 있는 것을 선택하여 작업의 효율성을 높일 수 있다.

실행하고자 하는 작업이 해당 클러스터에 전달되고 실행

되면, 하나의 워크플로우에 해당하는 작업이 모두 수행될 때까지 엔진은 Gram에서 제공하는 GramJobListener를 이용하여 해당 잡의 상태를 기록하고 그 모니터링에 관한 정보를 에디터 쪽으로 전달해주게 된다. 에디터는 실제 작업을 실행시킨 사용자에게 정보를 전달해 줘야 하기 때문에 사용자가 엔진 쪽에서 전달해준 정보를 쉽게 이해하기 위해서 각 작업은 READY, ACTIVE, DONE, PENDING, SUSPENDED, FAILED의 상태로 표현할 수 있고 상태 별로 그래픽하게 처리해 인식을 쉽게 하였다. 이런 모니터링 부분을 통하여 그리드 환경에서 수행되는 각 작업의 상태를 확인하고 그 과정 및 결과를 얻어볼 수 있다.

작업이 끝나거나, 에러가 발생하여 제대로 실행되지 못했을 경우 해당정보를 사용자에게 알려주는 서비스도 제공하게 된다.

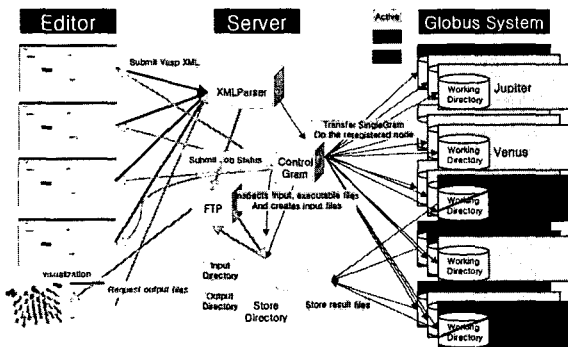
4. 엔진의 주요 기능

4.1 에디터와의 인터페이스

CEQA에서 에디터와 엔진간의 정보 교환은 XML형식으로 이루어지고 사용자가 에디터에서 작업을 수행하기 위해 디자인한 워크플로우에 관한 정보도 XML 형식으로 저장되어진다. 사용자가 에디터를 통해서 작업한 일에 대한 정보는 XML형식으로 엔진에 전달되고, 엔진에서는 XMLParser를 통해서 요청된 작업의 종류(작업실행, 파일전송, MDS-노드에 대한 정보, 인증서 생성, 잡로그(job log) 요청 등)를 파악한다. 엔진은 에디터에서 전송되어진 워크플로우 데이터를 분석하여 글로버스(globus)에서 수행될 수 있도록 데이터를 RSL(Resource Specification Language)형식으로 변화시키고 사용자가 에디터를 통하여 형성한 워크플로우에 따라서 각 작업을 실행한다. [그림 2]는 엔진과 에디터간의 전체적인 작업의 실행흐름을 보여주는 것으로, 해당 작업의 세부적인 작업들이 글로버스의 노드들에 맵핑 되어 돌아가는 것을 볼 수 있다. 엔진은 이런 전체적인 작업을 실행하고 작업의 결과 파일들을 자체적으로 저장한 후 해당 에디터로 필요한 정보를 뽑아서 보내준다. 에디터는 이 정보를 바탕으로 원하는 결과를 가시화해서 사용자에게 전달한다.

4.2 정보 매핑 및 관리 서비스

글로버스 시스템을 사용하기 위해서는 무엇보다 중요한 것이 인증서이다. 이를 위해서 필요한 것이 인증서를 관리하는 서비스를 지원하는 것이다. 현재 이 시스템을 이용하기 위해서 필요한 것은 먼저 에디터 시스템을 이용하기 전에 자신의 인증서가 있는 시스템에서 인증서를 생성하고 이 시스템을 쓰면 된다. 이 때 사용자의 환경을 설정하는 방법은 환경설정 파일인 user.xml에 사용자의 해당 호스트에서의 이름과 사용자명을 표시하고 해당 인증서의 플래시를 기입해 주면된다.



[그림 2] 서버와 에디터간의 실행 환경

4.3 응용 모니터링 서비스

CEGA 서버는 사용자가 실행한 작업의 상황을 모두 제어하며 그 정보를 관리하고 클라이언트에게 작업의 진행 상태를 전달해 주기 위해 모니터링 서비스를 하게 된다. Java Cog Kit을 이용하여 작업을 글로브스를 통해 수행하면 실시간으로 모니터링 정보를 얻게 된다. 엔진은 이 정보를 가지고 XML형태로 변환하여 에디터로 전송하고 작업에 대한 모니터링 정보를 나타내기 위하여 READY, ACTIVE, DONE, FAILED, SUSPENDED, PENDING 6가지의 상태로 체크되어 해당 잡의 모니터링 결과가 정상적으로 나올 경우 결과 파일을 가져온다.

모니터링은 각 작업에 해당하는 Gram에서 책임을 지게 되고 각 단계의 Gram은 모니터링 정보가 갱신될 때마다 자신의 상위 Gram에게 모니터링에 관한 정보를 전달하고 최상단의 Gram은 전달받은 모든 정보를 보고 작업의 상태를 바꾸게 된다. 엔진은 서버 측의 상태를 바꿈과 동시에 클라이언트 쪽으로 작업 상태의 변경을 전달하여 사용자가 작업의 진행도를 체크할 수 있게 해준다.

서버 내에서의 메시지 흐름은 정리하면 다음과 같다.

- ① 에디터에서 서버로 XML파일 전송-서버시스템에 작업 요청한다.
- ② 서버의 XMLParser부에서 요청된 작업을 적당한 사용자(서버 시스템에 등록되고, 인증서가 유효한 사용자)인지 확인하고 해당 모듈에 작업 내용을 보낸다.
- ③-1 MDS 모듈은 글로브스 환경에서 직접적으로 작업을 요청하고, 해당 정보를 얻어온다.
- ③-2 FTP 모듈은 사용자 인증서가 사용할 수 있는 호스트들 사이에서 파일의 교환을 지원한다.
- ③-3 GRAM 모듈은 사용자가 작업한 작업을 실행하기 위해서 몇 가지 패턴(SingleGram, SplitGram, MultiGram)을 지원한다.
- ④ 해당 작업에 대한 결과에 대한 내용을 보내기 위해 그에 해당하는 xml을 생성하여 에디터에 보낸다.

4.4 로깅 서비스

작업이 진행되는 클러스터에서 현재 상태를 모니터링하는 것도 중요한 것이지만, 작업이 끝나거나, 에러가 발생하여 제대로 실행되지 못했을 경우에는 그에 해당하는 정보를 가지고 있어야 한다. 이렇게 저장되어있는 정보는 작업을 관리하는 서버 측에서는 작업의 진행 상태나 시스템의 유추정보를 알 수 있고 사용자별로 작업 관리

를 할 수 있게 해준다.

엔진은 작업 수행 현황을 로그파일로 제작하여 엔진 내부에 저장하는데 현재 서버에서 적용되는 Logging 서비스는 에디터에서 받는 하나의 일에 대해서 “사용자명 : 작업명 : 작업 시작 시간 : 작업 종료 시간”을 표현하는 간단한 형태로 구성되어 있다. 이런 정보는 작업의 결과 파일을 저장하는 공간의 최상위 디렉토리에 'cega.log'라는 명칭으로 저장하게끔 하였다. 파일을 보았을 때 작업 종료 시간이 'null'로 표기되는 작업은 작업 실행 중에 에러가 발생하여 작업이 종료된 경우이다. 또한 이 파일을 이용해서 같은 사용자가 같은 이름의 작업 명을 사용하면 서버에서 그 작업을 거부하게 할 수 있다. 사용자명과 작업명이 같으면, 후에 결과를 찾아볼 때 찾을 수 없기 때문이다. 이 파일은 사용자가 수행한 작업의 정보를 가지고 있기 때문에 사용자가 요구할 경우에 에디터를 통하여 제공하게 해준다.

5. 결론 및 향후 과제

본 논문에서는 나노 테크놀로지에서 강력한 계산능력을 필요로 하는 VASP 프로그램을 국가 그리드 환경에서 실행할 수 있도록 하였고 다양한 실행환경 구성이 용이하도록 하여 실행 모니터링이 가능하도록 비주요한 모니터링을 실시간으로 제공하여 작업 효율성을 높였다. 또한 XML 형태의 데이터 정의를 바탕으로 정보의 표준화를 지원하며 차후 제고되는 기능과의 연동이 용이한 PSE를 설계하였다.

앞으로 VASP뿐만이 아니라 다양한 나노 응용 어플리케이션을 지원하는 PSE 포털을 구축하고, NWChem 같은 기존 시스템과의 상호 협력이 가능한 기능들이 추가되도록 하여 보다 효율적인 환경에서 사용자가 편리하고 쉽게 이용할 수 있도록 응용 어플리케이션 개발자에게 제공하는 것을 지원할 것이다.

참고문헌

- [1] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International J. Supercomputer Applications, 15(3), 2001.
- [2] Romberg, M., "The UNICORE Architecture Seamless Access to Distributed Resources", High Performance Distributed Computing, pp.287-293, 1999.
- [3] Globus Project, <http://www.globus.org>
- [4] Ian Taylor; Matthew Shields; Ian Wang., "RESOURCE MANAGEMENT OF TRIANA P2P SERVICES", Department of Physics and Astronomy Cardiff University
- [5] Triana, <http://www.triana.co.uk>
- [6] Globus Project, <http://www.globus.org>
- [7] KISTI Grid Testbed, <http://gridtest.hpcnet.ne.kr/>
- [8] Myproxy, <http://dast.nlanr.net/Projects/MyProxy>
- [9] G. Laszewski, I. Foster, J. Gawor "CoG Kits: A Bridge between Commodity Distributed computing and High-Performance Grid," ACM 2000 Java Grande Conference, San Francisco, June 2000.
- [10] N. Karonis, B. Toonen, and I. Foster. "MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface." Journal of Parallel and Distributed Computing, 2003.