

Grid 환경을 지원하기 위한 PSE 설계

안준언*, 박형우**, 이상산***

한국과학기술정보연구원 슈퍼컴퓨팅 센터

e-mail : {rocahn*, hwpark**, sslee***}@hpcnet.ne.kr

A Design of PSE for Supporting the Grid Environment

Joon-Eun Ahn*, Hyung-Woo Park**, Sang-San Lee***

Korea Institute of of Science and Technology Information Supercomputing Center

요 약

네트워크의 발달로 그리드 환경의 구현이 가능하게 되었다. 이기종간의 리소스들을 연결하여 초고속 고성능의 환경을 제공하는 그리드 환경을 사용하기 위해 보다 쉽고 투명성 있는 사용자 환경이 요구되고 있다. 이에 따라 PSE(Problem Solving Environment)가 그리드 사용자 환경으로 대두되고 있다. 이 논문에서는 PSE 의 연구 동향을 조사 분석하고, 국내 그리드 환경을 위한 PSE 를 정립하고자 한다. 접근방법은 PSE 가 제공해야 할 환경들을 정의하고 구현에 필요한 PSE 시스템을 설계 제시하고자 한다.

1. 서론

이전부터 메타 컴퓨팅이라는 용어로 슈퍼 컴퓨터 혹은 워크스테이션의 연결을 하여 단일 컴퓨터처럼 사용하고자 하는 연구가 계속되었다. 그러나 네트워크의 속도로 인해 LAN 환경 상에서의 구현이 고작이었으며 동기종간의 연동만이 실제 구현되었다. 그러나 인터넷의 보급과 네트워크의 발달로 인해 WAN 상의 컴퓨팅 리소스들의 연동 환경의 구현이 가능하게 되었으며, 그리드라는 용어로 대두되었다. 그리드란 지역적으로 분산된 고성능 고용량의 자원들을 초고속 네트워크로 연동하여 단일 시스템처럼 사용할 수 있도록 하는 정보 통신 인프라이다. 이러한 환경의 지원을 위해 현재 많은 연구들이 진행되고 있으며 세계 여러 국가에서 추진 지원하고 있다.[1,2,3,4] 그러나 현재 이러한 환경을 사용하기 위해서는 제약이나 어려움이 많다. 사용 환경의 설치 뿐만 아니라 실제 사용에 있어서도 미들웨어 기술에 대한 습득이 요구되고 있으며 이로 인해 그리드 환경의 필요성을 인지함에도 접근에 힘들어 하고 있다.

이러한 사용상의 어려움으로 인해 보다 쉽고 투명성 있는 사용자 환경이 요구되고 있다. 현재 진행되고 있는 그리드 프로젝트들도 사용자 환경의 개발도 병행하고 있다. 또한 분산 환경상에서의 사용자 개발 환

경을 지원하던 PSE 들이 그리드 환경을 지원함으로써 PSE 가 그리드 사용자 환경으로 대두되고 있다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 PSE 의 일반적인 개념과 그리드 환경을 지원하기 위해 제공되어야 할 그리드 서비스를 소개한다. 3 장에서는 기존의 그리드 환경을 지원하는 PSE 들의 특징을 분석하고 4 장에서는 그리드 환경에서의 PSE 의 구현을 위해 제공되어야 할 환경과 N*Grid 를 위한 PSE 구현 방향을 간략히 소개한다.

1.1 PSE(Problem Solving Environment) 소개

1.1.1 일반적인 개념

PSE(Problem Solving Environment)란 특정 분야에서 어떤 문제(Problem)를 풀기 위한(Solving) 제반의 환경(Environment)을 제공하는 컴퓨터 시스템이다. 여기에서 제반의 환경이란 문제를 풀기 위해 필요한 모든 컴퓨팅 리소스들을 말한다. 그것은 컴퓨터, 네트워크, 실험 장치 등의 물리적인 리소스뿐만 아니라 프로그램, 알고리즘, 최신의 정보기술 등의 논리적인 리소스도 포함한다. 간단하게 말해서 문제를 풀고자 하는 사용자에게 그 문제를 풀기 위해 필요한 기술이나 정보 등을 제공하여 보다 쉽고 편하게 풀 수 있도록 도와주는 환경을 말한다. 사용자는 별다른 교육이나 기술의 습득 없이도 PSE 가 제공하는 최신의 기술을 이용하여 자신의 문제를 풀 수 있다.

1.1.2 그리드 환경에서의 PSE

그리드는 기존의 동일 기종의 컴퓨팅 리소스들뿐만 아니라 이기종의 컴퓨팅 리소스들을 연결하는 것이다. 따라서 하나의 프로그램이 여러 OS 나 여러 기종에서 컴파일 되고 실행이 될 것이다. 기존에 병렬프로그램을 작성하여 여러 이기종 컴퓨터에서 수행하기 위해서는 수행하고자 하는 컴퓨터에서 소스를 각각 컴파일하고 필요한 파일들을 복사해 두고 실행을 시켜야 했다. 만약 소스를 수정하고 새로 수행하기 위해서는 다시 소스를 복사하고 컴파일하고 하는 번거로운 절차를 거쳐야 했다. 이와 같이 이기종 상에서의 사용자들의 사용환경을 제공하기 위해 기존 병렬분산 환경을 지원하는 PSE 를 이용하여 그리드 환경을 이용하는 것이다.

PSE 가 그리드 환경을 지원하기 위해서는 다음과 같은 서비스들을 지원해야 한다. 다음의 서비스를 반드시 PSE 자체가 제공할 필요는 없다. 앞서 나온 글로버스와 같은 그리드 미들웨어와 연동하여 다음의 서비스를 지원할 수도 있다.

1. 보안 서비스(Security Service) : GSI(Grid Security Infrastructure)
2. 정보 서비스(Information Service) : GIS(Grid Information Service)
3. 스케줄링 서비스(Scheduling Service)
4. 데이터 전송 서비스(Data Transfer Service)

다음은 추가적인 서비스로 사용자 개발 환경 지원을 위한 서비스이다.

5. 협업 개발 환경(Collaborative Development Environment)
6. 시각화(Visualization)
7. 원격 제어(Remote Steering)

협업 개발 환경은 개발자들 간의 협업을 지원해 주는 환경을 말한다. 프로젝트 관리 뿐 아니라 커뮤니티 환경까지 제공할 수도 있다. 시각화는 프로그램이나 실험의 결과를 화면이나 CAVE 와 같은 툴을 이용해 보여줄 수 있도록 해준다. 원격 제어는 프로그램 수행이나 실험 시 파라미터(parameter)값들을 동적으로 변경할 수 있도록 해준다.

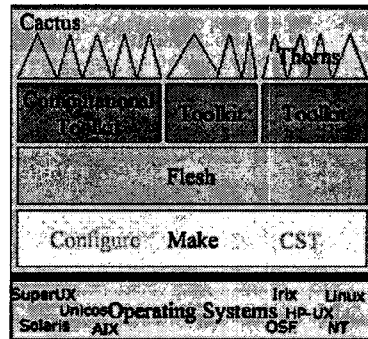
2. 그리드 환경을 지원하는 PSE 관련 연구

본절에서는 그리드 환경을 지원하는 PSE 연구를 조사 분석하여 국내연구 환경에 맞는 PSE 요구 도출에 확고하고자 한다.

2.1 Cactus[5]

Cactus 는 오픈 소스 PSE 로 AEI(Albert Einstein Institute)에서 Numerical Relativity 계산을 위해 구현된 것으로 1997 년부터 개발되어 현재 버전 4.0 의 베타 12 까지 나와 있다. Cactus 는 애플리케이션 모듈인 'thorn'과 확장 인터페이스를 통해 thorn 들을 연결해주는 코어 부분인 'Flesh'로 이루어져 있다. thorn 은 응용분야 사용자들이 자신의 문제 해결을 위해 개발하

는 어플리케이션 thorn 과 특정 IO 나 드라이버 등을 지원하는 infrastructure thorn 으로 나뉜다. 각각의 관련 있는 thorn 들의 집합을 'arrangement'이라고 하며, 이런 arrangement 들의 집합을 'toolkit'이라고 한다. 특히 여러 다양한 애플리케이션을 지원하기 위한 계산 인프라를 제공하는 arrangement 들의 집합을 'Cactus 계산 툴킷'이라고 한다[그림 1]. 현재 대부분의 아키텍처와 운영체제를 지원하며, F77, F90, C, C++를 지원하며 현재 Java 지원 thorn 이 개발 중이다. 개발자들은 자신의 기존 소스를 사용할 수 있다. 소스에 대하여 파라미터 환경(parameter configuration) 파일, 인터페이스 환경(Interface configuration) 파일과 스케줄링 환경(Scheduling Configuration) 파일을 작성하고 기존 소스 파일에 약간의 코드를 수정을 하여 그리드 환경이나 병렬분산 환경에서 수행 시킬 수 있다. 각각의 환경 파일은 CST(Cactus Specification Tool)을 통해 파싱되어 수행 시 적용되게 된다.



[그림 1] Cactus 아키텍처

그리드 서비스를 지원하기 위해 그리드 미들웨어인 글로버스[6]와 연동하여 지원한다. 그러나 실제로 글로버스의 서비스를 이용하여 그리드 서비스를 제공하는 것은 아니다. 글로버스를 통해 프로그램이 실행될 수 있도록 실행파일만을 제공할 뿐이다. 글로버스의 서비스를 이용하기 위해서는 사용자가 모든 서비스를 직접 이용하여 실행하여야 한다. 즉 보안 서비스, 정보 서비스, 스케줄링 서비스와 데이터 전송 서비스를 이용하기 위해 글로버스의 GSI, MDS(Monitoring & Discovery Service), GRAM(Globus Resource Allocation Manager)와 GridFTP 등을 이용하여 사용자가 직접 환경을 설정한 후 Cactus 소스를 글로버스 라이브러리를 이용하여 컴파일하여 실행파일을 globusrun 을 이용하여 실행하여야 한다. 협업 개발 환경은 CVS(Concurrent Versions System)를 이용하여 툴킷 개발 관리 등을 지원하고 있다. 시각화는 결과를 HDF5, XML, IEEEIO 와 같은 포맷을 지원하여 같은 포맷을 사용하는 시각화 툴에서 사용 가능하다. 또한 파라미터 환경파일을 이용하여 원격 제어(Steering)가능한 파라미터를 지정하여 원격제어가 가능하다.

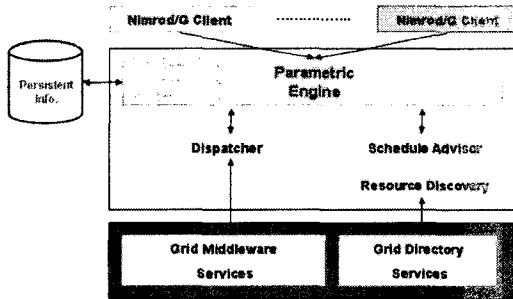
2.2 Nimrod/G[7]

Nimrod/G 는 분산환경에서의 파라미터 기반의 응용 프로그램을 수행 관리하는 Nimrod 의 그리드 환경을

지원하는 버전이다. Nimrod/G 는 다음의 기본 컴퍼런트들로 구성되어 있다.

- 클라이언트(Client or User Station)
- 파라메트릭 엔진(Parametric Engine)
- 스케줄러(Scheduler)
- 디스패처(Dispatcher)
- 작업 래퍼(Job-Wrapper)

클라이언트는 사용자 인터페이스를 제공한다. 이를 통해 파라미터를 설정하고 작업에 대한 모니터링과 제어를 할 수 있다. 파라메트릭 엔진은 작업의 관리 뿐 아니라 유지에 있어서 중심 컴퍼런트로 작업의 실제 수행과 관리를 수행한다. 스케줄러는 리소스의 발견(discovery), 리소스의 선택과 작업의 배치(assignment)를 수행한다. 디스패처는 선택된 리소스상에서의 작업의 수행을 담당한다. 실제로 작업 래퍼를 실행시킨다. 작업 래퍼는 작업과 데이터의 스테이징(staging), 작업의 실행 및 결과를 디스패처를 통해 파라메트릭 엔진으로 보내주는 역할을 한다. 각각의 컴퍼런트들의 상호연동을 그림 2 에 나타나 있다.



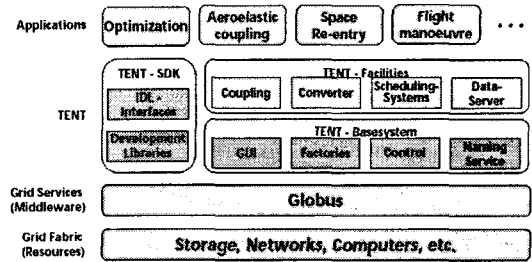
[그림 2] Nimrod/G 아키텍처

Nimrod/G 또한 글로버스의 서비스를 통해 그리드 서비스를 제공한다. 각각의 컴퍼런트들은 글로버스의 MDS, GRAM, GASS(Global Access to Secondary Storage) 와 GSI 를 이용하여 수행한다. 협업 서비스를 위한 모듈을 없으나 유저 인터페이스를 통한 작업의 실행과 모니터링을 할 수 있다.

2.3. TENT[8]

TENT 는 공학 분야의 응용프로그램의 수행을 위한 PSE 로 CORBA 기반의 컴퍼런트 시스템이다. 각각의 코드들은 래퍼(wrapper)를 통해 캡슐화(encapsulated)되어 있어 수정 없이 통합할 수 있다. 워크플로우(WorkFlow)방식의 사용자 인터페이스를 제공하며 분산환경의 수행도 지원한다. 온라인 방식의 시각화와 원격제어를 지원한다. TENT 의 구조는 그림 3 에 나타나 있다.

TENT 도 마찬가지로 글로버스를 통해 그리드 서비스를 제공한다. 글로버스의 GSI, MDS 와 GridFTP 를 통해 보안, 정보 및 데이터 전송 서비스를 제공한다



[그림 3] TENT 아키텍처

글로버스와의 연동을 위해 JAVA CoG 를 이용하였으며, 각각의 GUI 는 자바를 통해 구현되어 플러그인 방식으로 제공된다.

3. 국내 그리드 환경을 위한 PSE 의 설계

3.1 국내 그리드 환경

국내 그리드 환경은 현재 KISTI 를 중심으로 구축되어 지고 있다. 99 년부터 국내 슈퍼컴퓨터를 중심으로 연구를 계속해 오고 있으며 세계 테스트베드와의 연동을 준비 중이다. 현재 글로버스 기반의 80 노드로 구성된 PC 클러스터가 테스트 베드로 제공 중이며 올해 안으로 KISTI, 포항공대와 전북대학교의 슈퍼컴퓨터로 이루어진 테스트베드의 테스트가 이루어질 예정이다. 사용자 환경은 현재 명령어 기반으로 이루어지고 있으며 현재 N*Gate 포털이 구축되어 테스트 중이며 곧 서비스 될 예정이다.

3.2 PSE 의 구현을 위해 제공되어야 할 환경

앞에서 나온 PSE 들의 경우 처음부터 그리드 환경을 지원하기 위해 설계 구현된 PSE 는 아니다. 각각 분산환경을 지원하기 위해 설계 구현되었으나 글로버스를 통해 그리드 환경을 지원한다. Nimrod/G 와 TENT 는 그리드 환경을 지원하기 위해 글로버스의 서비스를 직접 이용하지만 Cactus 는 반대로 글로버스가 Cactus 를 수행하는 구조로 되어 있다. 또한 각각의 PSE 들이 지원하는 응용 분야도 차이가 있다. 이와 같이 각각의 PSE 들 마다 지원하는 기능이 상이하며 제공하는 환경도 다르다. 일반적인 응용 분야의 프로그램을 지원하기 위한 그리드 환경에서의 PSE 가 제공하여야 할 환경에 대해 정의를 했다.

- 사용자 인터페이스 환경 : 사용자가 쉽게 그리드 환경을 이용할 수 있는 사용자 인터페이스가 제공되어야 한다. 그리드 환경을 이용하고자 하는 사용자들에게 투명성을 제공하여야 한다. 현재 사용자 인터페이스 구현에 자바가 많이 사용되고 있으며 웹 인터페이스를 통해 구현되고 있다.
- 작업 정의 및 관리 환경 : 일반적인 PSE 의 기능으로 수행하고자 하는 작업의 입력 및 출력 지정, 수행 리소스 설정 및 수행 시간 등의 정의 및 관리 등의 환경을 제공하여야 한다.
- 작업 수행 환경 : 작업을 수행하기 위해 필요한 리소스 정보 및 작업 수행에 필요한 정보들이 제공되

어야 한다. 현재 사용자가 사용할 수 있는 리소스의 종류와 각 리소스의 사용 허용 시간, 리소스의 비용 등의 정보가 제공되어야 한다.

- 통합 환경 : 응용 분야의 사용자들은 여러 종류의 개발 언어를 사용하고 있다. 또한 기존 개발된 프로그램의 수행을 지원하는 환경의 지원도 중요한 요소이다. 그러나 실제 기존 개발된 프로그램을 지원하는 환경의 설계 및 구현상 많은 어려움이 존재한다.

- 협업 환경 : 그리드 환경의 특징 중의 하나로 협업 환경의 지원을 들 수 있다. 실제 수행되고 있는 많은 프로젝트들은 많은 사람들이 공동으로 수행하는 경우가 많다. 따라서 모든 사람들이 프로젝트 수행 참가, 정보 교환 등의 협업 환경을 제공하여야 한다. 실제로 앞에서 나온 PSE 들 중 협업 환경의 제공은 제대로 되지 않고 있다.

- 그리드 서비스 제공 환경 : 일반적인 그리드 서비스를 지원하여야 한다. 그리드 서비스를 제공하기 위한 방법은 그리드 미들웨어와의 연동하는 방법이 가장 효과적이며 실용적이다. 그리드 미들웨어와 연동하는 방법은 미들웨어에 종속적이다. 현재 앞에서 나온 PSE 들은 모두 글로벌스와 연동하며 그 연동 방법은 다양하다. 현재 대부분 CoG(Commodity of Globus)를 통한 연동 방법이 가장 많이 사용되고 있다.

- 모니터링 환경 : 작업이 실행 중 일 때 작업의 전반적인 상태를 모니터링 할 수 있는 환경을 제공하여야 한다. 뿐만 아니라 상태에 따른 전반적인 리포터, 작업의 재실행 및 작업의 종료 등의 기능도 제공되어야 한다.

- 가시화 환경 : 선택적인 환경으로 작업의 수행 결과물에 대한 가시화 환경을 제공되어야 한다. 기존 상업적으로 사용되고 있는 프로그램에 호환되는 형식의 출력물을 산출하거나 연동되도록 제공 되어 질 수 있다. 또는 PSE 에서 직접 출력물에 대한 가시화 모듈을 제공할 수 있다. 전자의 경우 사용자가 가시화 프로그램을 보유하고 있을 경우에 효율적이며, 후자의 경우 사용자의 부가적인 소프트웨어의 구입 비용을 줄일 수는 있지만 호환성의 문제가 생길 수 있다.

- 원격 제어 환경 : 선택적인 환경으로 수행하고자 하는 작업에 따라 원격 제어 환경이 제공되어야 한다. 실제 PSE 의 설계 목적에 따라 이 환경의 개발에 많은 비중을 차지할 수 있다.

위의 정의된 환경들의 관계는 그림 4 와 같다.

3.3 N*Grid 를 위한 PSE 구현 방향

앞에서 나온 환경을 기준으로 N*Grid 를 위한 PSE 를 구현할 예정이다. 각각의 환경들은 컴퍼넌트 기반으로 구현될 것이다. 사용자 인터페이스로는 자바 기반으로 구현될 것이다. 각각의 환경의 구현 방향은 다음과 같다.

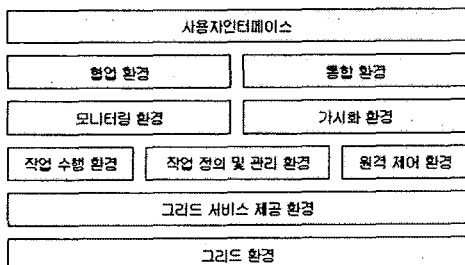
- 사용자 인터페이스 : 자바 기반의 애플리케이션 혹은 애플릿(그리드 포탈과 연동 가능)
- 작업 정의 및 관리 환경 : 응용분야에 따른 분석을 통한 인터페이스 정의와 CoG 혹은 GAT(Grid Application Toolkit)을 이용한 인터페이스를 설계
- 작업 수행 환경 : MDS 를 통한 리소스 정보 제공
- 통합 환경 : 응용 분야에 따른 분석 후 설계
- 협업 환경 :
- 그리드 서비스 제공 환경 : 그리드 미들웨어인 글로벌스를 기반으로 서비스를 제공하도록 설계
- 모니터링 환경 : 로컬 스케줄링 소프트웨어를 통한 API 사용
- 가시화 환경 : 기존 소프트웨어와 호환되는 파일 양식 제공 혹은 응용 분야에 따른 사용자 요구사항에 따라 설계
- 원격 제어 환경 : 현재는 구현 하지 않을 예정

응용 분야의 요구 분석 및 문제 분석을 통해 전반적인 구조가 결정될 예정이며 현재 개발 중인 그리드 포탈인 N*Gate 와 연동할 예정이다.

4. 결론 및 향후계획

PSE 정의를 정리하고 그리드 환경에서의 PSE 가 제공하여야 할 서비스를 정의 하였다. 또한 기존 그리드 환경을 지원하는 PSE 인 Cactus, Nimrod/G 와 TENT 를 살펴보고 이를 통해 그리드 환경을 지원하기 위해 PSE 에서 제공되어야 할 환경을 정의 하고 이에 따른 PSE 구현 방향을 제시하였다.

향후 응용 분야의 요구 분석을 통해 앞에서 정의된 환경에 따라 구조를 정의하고 구현을 할 예정이다.



[그림 4] 그리드 환경 지원 PSE 제공 환경

참고문헌

- [1] IPG Homepage : www.ipg.nasa.org
- [2] www.teragrid.org
- [3] www.eurogrid.org
- [4] <http://www.e-science.clrc.ac.uk>
- [5] CactusCode Project Homepage : WWW.cactuscode.org
- [6] Globus Project Homepage : www.globus.org
- [7] Nimrod/G Project Homepage : www.csse.monash.edu.au/~davida/nimrod/
- [8] TENT Project Homepage : <http://www.sistec.dlr.de/tent/index.shtml>