

웹기반 그리드컴퓨팅 프레임워크의 설계 및 구현

강 경 우^o, 강 윤 희, 김 도 현, 조 광 문, 궁 상 환
천안대학교 정보통신학부

Design and Implementation of Web-based Grid-computing Framework

Kang kyung-woo, Kang yun-hee, Kim do-hyun, Cho kwang-moon, Kung sang-whan
Cheonan University
E-mail : kwkang@cheonan.ac.kr

요 약

고성능 컴퓨팅 자원에서의 그리드 컴퓨팅은 다양한 대규모 계산 문제를 해결하기 위해 적용되고 있다. 응용 프로그램 개발자들에게 분산된 컴퓨팅 자원을 제공하는 다양한 소프트웨어들이 개발되었다. 그러나, 이들 시스템들은 웹기반이 아니거나 협력작업을 위한 기능을 제공하지 않거나 실시간으로 결과를 볼 수 있는 도구를 제공하지 않는다. 네트워크 어플리케이션 개발 측면에서 보면 웹기술은 일반적으로 사용하는 기술이 되고 있다. 특히, 인터페이스가 플랫폼과 독립적이라는 특징 때문에 더욱 그렇다. 본 연구에서는 MPI를 이용해 작성된 병렬 SPMD 어플리케이션의 수행을 위한 웹기반 프레임워크를 제안한다. 또한, 실시간 가시화 기술을 이용해서 협력연구를 위한 환경을 개발하였다.

Abstract

Grid-computing on networked computers is increasingly applied to the variety of large-scale computation problem. Several software systems are developed for providing the application programmers with computers available. However, these systems are not web-based systems, lacks a collaborative environment or do not supply the real-time visualization facility. Web technology is becoming the general technology on the development of network application, in particular, because the interface can be made platform independent. In this paper, we propose the web-based framework for executing the parallel SPMD application written MPI. Also, a web-based collaborative environment is developed with a real-time visualization technology.

I. 서론

고성능 컴퓨팅 자원에서의 병렬분산 컴퓨팅은 다양한 대규모 계산 문제를 해결하기 위해 적용되고 있다. 응용 프로그램 개발자들에게 분산된 컴퓨팅 자원을 제공하는 다양한 소프트웨어들이 개발되었다.

지금까지 개발된 시스템들을 살펴보면 두 가지 형태의 작업 실행 환경이 있다. 첫 번째 부류는 PVM, pPVM, P4와 같은 다양성을 강조한 것이 있다.[7] 그러나, 이들은 웹기반이 아니다. 따라서 협력 연구를 위해서도 부적당하다. 단지, SPMD형태의 프로그램 수행에 적합할 뿐이다. 두 번째 부류는 FIDO[9], MIDAS[6] 와 같은 대규모 분산 어플리케이션

선 수행 환경이다. 그러나, 이들은 특정한 문제에 너무 치우쳐 있고 제한적이라는 단점이 있다. 또한, 협력 연구를 위한 지원이 부족하다는 단점을 가지고 있다. 웹기반 컴퓨팅 환경으로 WebFlow가 개발되었다.[8] 이것은 자바로 작성된 분산 어플리케이션을 수행하기에 좋은 구조를 가지고 있다. 그러나, 이 시스템은 어플리케이션 개발을 자바로 개발해야 하는 문제가 있다. 본 연구에서는 어플리케이션 프로그램 개발자들이 주로 사용하는 언어인 포트란과 C언어를 지원하고 분산환경에서 협력작업까지 지원하는 환경을 개발하였다.

본 연구에서는 두개의 시스템 그리드 포털과 실시간 가시화기를 개발하였다. 그리드 포털은 그리드 환경에서 SPMD 어플리케이션을 실행시키기 위한 웹기반 작업환경이고 실시간 가시화기는 실행중인 어플리케이션의 결과를 바탕으로 사용자에게 실시간으로 가시화 하는 도구로써 자바를 이용하여 개발하였다. SPMD 어플리케이션들은 일반적으로 전체문제를 해결하기 위해 서로 관련이 있는 여러 상이한 모듈로 이루어진다. 이들 모듈들은 서로 다른 분야에서 개발되고 독립적으로 최적화된다. 이들을 통합하고 최적화하는 전통적인 방법들은 길고 지루한 과정이다. 이 과정이 느린 것은 서로 같이 테스트하고 모니터링 할 협력의 장이 부족함에 기인한다. 이 문제를 해결하기 위해 우리는 실시간 가시화기를 개발하였다. 그리드 포털과 같이 실시간 가시화기도 이식성과 사용의 편리성을 위해 웹기술을 이용하였다.

II. 그리드 포털

본 연구에서는 SPMD형태의 사용자 어플리케이션을 쉽게 그리드 환경에서 수행하기 위해 웹기반 그리드 포털을 개발하였다. 그리드 포털의 기능으로는 웹 인터페이스를 통한 인증, 컴퓨팅 자원들을 결정하기 위한 자원의 상태 정보 제공, 웹 인터페이스를 통한 어플리케이션 분배 및 컴파일, 어플리케이션 프로그램의 실행 등이다. 이들 과정에 대한 시스템 구조는 다음 그림과 같다.

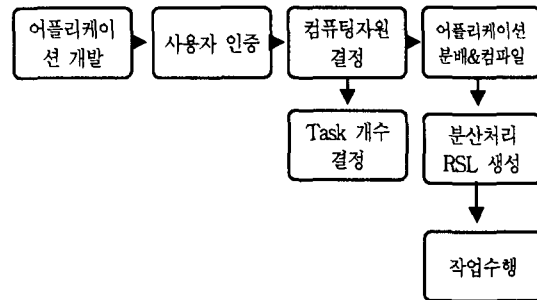


그림 1. 웹기반 프레임워크의 구조

1. 사용자 인증

그리드 보안은 분산 자원들을 공유함에 따라 발생하는 자연스러운 문제이다. 사용자의 입장에서는 안전하면서도 사용의 편리성을 요구할 것이고, 각 자원을 소유하고 관리하는 관리자의 입장에서는 자원이 그리드 환경에 노출되는 것이기 때문에 사용의 편리성보다는 더 안전한 보안을 원할 것이다.[1,3,4] 이를 위해 위의 사용자 인증부분은 single-sign-on을 제공하고 각 자원에 대해 인증을 위해 proxy를 이용한다. 사용자는 그리드환경에 한번의 인증과정을 거침으로 사용이 허용된 자원들을 사용할 수 있다. 이를 위해 본 연구에서는 SWING을 이용하였다.

2. 컴퓨팅 자원 결정

2.1 자원정보 가시화

사용자 어플리케이션 수행을 위한 컴퓨팅 자원 결정을 위해 컴퓨팅 자원의 상태정보를 제공한다. 각 자원의 현재 사용량을 그래프나 테이블을 통하여 실시간으로 사용자에게 제공함으로써 자원의 사용량이 많은 시간과 그렇지 않을 시간을 사용자가 손쉽게 할 수 있도록 하였다. 자원의 정보 제공은 LDAP을 이용하여 각 자원의 디렉토리를 구성하는 각 요소들을 웹 페이지를 통하여 제공하도록 하였다.[2] 예를 들어, 사용자가 Lsf를 이용하여 관리하는 클러스터의 상태정보를 보기 원한다면 아래 그림과 나타난다. 클러스터 내에 존재하는 노드들의 이름과 그 아래는 클러스터를 관리하는 로컬 스케줄러인 Lsf의 상태를 보여주고 있다. 각 노드들은 선택 가능하고 선택하면 선택된 노드의 최근 부하정보를 그래프 형태로 보여준다. Lsf의 상태를 나타내는 테이블은 사용자가 관심 있는 큐에 대해 현재 대기하고 있는 작업 수, 수행되는 작업 수, 큐의 우선순위, CPU할당 시간 등을 나타낸다. 이를 바탕으로 사용자는 자신의 작업이 큐에서 얼마나 대기해야 하는지를 가늠할 수 있게 된다.

Available Resources List

```

c : Joshua1.cheonan.ac.kr
c : Joshua2.cheonan.ac.kr
c : Joshua3.cheonan.ac.kr
c : Joshua1.cheonan.ac.kr
    
```

| QUEUE_NAME | FILE | SID | JS | MAX | LRU | LRP | LRP | LRP | LRP |
|------------|------|-------------|----|-----|-----|-----|-----|-----|-----|
| empty | UU | type:Active | 4 | 1 | - | - | - | - | U |
| pbfa | UU | type:Active | 2 | 8 | - | - | - | - | U |
| normal | UU | type:Active | 4 | 2 | - | - | - | - | U |

그림 2. 클러스터의 상태정보 가시화

2.2 Task 개수 결정

사용자는 사용가능 자원들의 상태를 파악한 후 어플리케이션의 Task 개수를 결정하게 된다. 본 연구에서는 SPMD 어플리케이션을 대상으로 하기 때문에 사용자는 컴퓨팅 자원간에 Task 개수를 어떻게 분배할 것인지도 결정해야 한다. 본 연구에서는 KISTI에 있는 Compaq GS320과 linux-cluster, 전북대에 있는 IBM SP2, 천안대에 있는 linux-cluster를 활용하였다. 컴퓨팅 자원의 선택은 호스트 이름 옆에 위치한 콤보박스를 눌러서 0이 아닌 숫자를 선택함에 따라 선택이 된다. 선택된 숫자가 해당 호스트에서 사용할 Task 개수이다. 작업은 interactive와 background로 나뉘어지는데 콤보박스 옆의 체크버튼을 선택함에 따라 background가 된다. background를 선택하게 되면 작업은 로컬 배치작업 스케줄러에게 전달된다. 아래의 예는 전북대 슈퍼컴과 KISTI의 linux-cluster 상에 각각 4개의 task를 PBS와 Loadleveler를 이용하여 수행시키기 위한 선택이다.

| Select the number of CPUs & Job Manager | | |
|--|--------------------------------|--------------------------|
| GS320 (whiparam.hpcnet.ne.kr) | <input type="text" value="0"/> | <input type="checkbox"/> |
| IBM SP2 (supercom.chonbuk.ac.kr) | <input type="text" value="4"/> | <input type="checkbox"/> |
| PC Cluster I (cluster.hpcnet.ne.kr) | <input type="text" value="4"/> | <input type="checkbox"/> |
| PC Cluster II (joshua.cheonan.ac.kr) | <input type="text" value="0"/> | <input type="checkbox"/> |

그림 3. Task 개수 결정

3. 어플리케이션의 분배 및 컴파일

웹 인터페이스를 통해 사용자는 어플리케이션을 업로드 할 수 있고 각 컴퓨팅 자원에 분배할 수 있다. 또한 각 컴퓨팅 자원에 맞게 컴파일을 수행한다. 이를 위한 웹페이지는 형태는 다음과 같다.

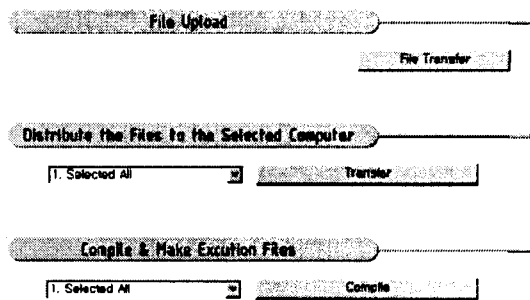


그림 4. 사용자 어플리케이션 분배

소스를 사용할 컴퓨팅 자원에 업로드 할지 여부를 선택하는 부분과 업로드 할 파일 등에 대한 선택에 관한 부분이 있다. 그리고 그 아래로 각 호스트에서 컴파일 하기 위한 버튼이 있다. 이를 통하여 각 컴퓨팅 자원에 맞는 실행파일을 생성할 수 있다.

III. 실시간 가시화

그리드환경에서 자신의 작업을 수행하는 어플리케이션 프로그램은 슈퍼컴퓨터 사용에 익숙하다. 슈퍼컴퓨터에서 작업을 하는 방법은 어플리케이션을 개발하고 데이터파일을 작성한 다음 배치작업 스케줄러를 통해 작업을 제출하고 작업이 끝날 때까지 기다리는 것이다. 어플리케이션이 컴퓨팅 자원에서 수행되는 시간은 짧아도 수 시간이고 길면 몇 일이 될 수도 있다. 이와 같은 방법으로 그리드 환경에서 작업을 수행하는데는 몇 가지 문제가 있다. 첫 번째는 프로그램 상에 오류가 있을 때이다. 결과가 의도한 것과 다르게 나오에도 불구하고 사용자는 작업이 마칠 때까지는 모르기 때문에 기다리는 수밖에 없다. 이 문제는 컴퓨팅 자원의 낭비를 초래한다. 사용자가 계산상의 중간 결과를 확인만 할 수 있어도 수행 도중에 작업을 정지시킬 수 있을 것이다. 두 번째 문제는 사용자는 반드시 자신의 자리로 돌아와야 작업결과를 확인할 수 있다는 것이다. 일반적으로 수행 후 나온 결과는 수 메가바이트 이상 되는 숫자 데이터 파일이다. 이것을 사용자들은 가시화 도구를 통해 형상화 한 후 결과를 확인한다. 따라서, 가시화 도구가 있는 곳이 아니면 결과를 확인할 수 없게 된다. 이것은 연구자의 연구의 지속성을 방해한다. 세 번째 문제점은 다른 지역에 있는 같은 분야 전문가와 자신의 결과를 놓고 토의하기가 어렵다는 점이다. 양쪽에서 같은 화면을 공유하기가 쉽지 않기 때문이다. 이것은 협력연구를 방해하는 요소가 된다.

본 연구에서는 위의 세 가지 문제를 해결하기 위해 웹기반 사용자 작업결과 실시간 가시화 시스템을 개발하였다.

본 연구에서 개발한 시스템의 구조는 아래 그림과 같다. 사용자 작업인 프로그램이 8개의 노드를 필요로 하고 두 호스트에서 4개씩 수행한다고 가정할 때 이들은 각각 로컬 디스크에 자신의 중간 결과들을 출력할 것이다. 출력된 결과들은 결과 취합 에이전트를 통해 모으게 된다. 각 태스크들은 계속해서 데이터파일을 만드는데 파일들을 서로 다른 파일에 저장한다면 파일이 너무 많아서 파일 시스템이 넘칠 것이다. 때문에 태스크들은 결과를 버퍼에 출력한다. 버퍼가 가득 차면 결과 취합 에이전트가 가져가고 태스크들은 다시 버퍼에 결과를 출력한다.

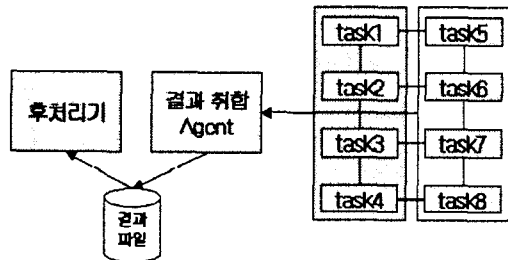


그림 5. 실시간 가시화기 구조

취합된 결과 파일은 가시화를 위한 후처리기가 존재하는 서버로 전송이 되고 후처리기를 통해 형상화한 이미지 파일이 생성된다. 본 연구에서는 결과를 형상화하기 위한 도구로써 gnuplot을 이용하였다. 다음은 gnuplot을 구동하기 위한 스크립트이다.

```

set pm3d at sb
set terminal png medium
set size 1.4
set output "vir0.png"
set view 180,0,1.5
set nokey
set nosurface
set contour
set cntrparam levels 15
set contour base
splot [-40:60][-100:400][0.45:0.95] "/tmp/vir.gpt" with lines
    
```

그림 6. gnuplot을 위한 스크립트

이 스크립트를 통해 취합된 데이터 파일 "/tmp/vir.gpt"은 이미지 파일 "vir0.png"로 변경된다. png파일은 jpg나 gif에 비해 선명하기 때문에 png를 택하였다. 그러나, 웹브라우저에 따라 png파일을 지원하지 않기 때문에 본 연구에서는 jpg파일도 생성하게 하고 있다.

생성되는 이미지 파일은 자바애플릿을 통해 웹브라우저에 출력된다. 자바애플릿은 이미지가 생성될 때마다 반복적으로 출력한다. 따라서, 후처리기가 결과 취합 에이전트가 가

져온 데이터를 기반으로 이미지만 만들면 자바애플릿이 출력하기 때문에 애니메이션 효과를 나타낼 수 있는 것이다.

이와 같은 웹기반의 실시간 가시화 시스템은 위에서 나열한 3가지 문제점을 해결해 주고 있다. 먼저, 사용자는 컴퓨팅 자원에서 자신의 작업이 다 끝날 때까지 기다릴 필요가 없게 된다. 중간결과를 기반으로 확인하기 쉬운 형상이 계속해서 제공되기 때문에 사용자 작업의 오류 또는 문제점을 쉽게 파악할 수 있게 된다. 이에 따라 작업을 중도에 멈출 수도 있다. 이것은 연구자의 연구기간을 단축 시켜줄 뿐만 아니라 컴퓨팅 자원의 낭비 또한 막을 수 있다. 두 번째 문제인 사용자가 특정한 곳에서만 결과를 확인해야 하는 문제를 해결하였다. 웹 기반으로 이미지를 보여주기 때문에 웹이 사용 가능한 어떤 환경 어떤 장소라도 결과를 확인할 수 있게 된다. 세 번째 문제인 같은 분야 연구자와의 협력 연구의 어려움도 해결할 수 있다. 아무리 원거리에 있다 할 지라도 같은 웹페이지를 공유할 수 있기 때문에 같은 결과를 보면 시뮬레이션 결과를 토의할 수 있게 된다.

IV. 결론

본 연구에서 개발한 웹기반 그리드 사용자 프레임워크는 사용자에게 웹 인터페이스를 제공함으로써 그리드 자원을 사용자가 편리하게 사용할 수 있도록 한다. 이로 말미암아 그리드 컴퓨팅의 고성능의 자원을 요구하는 과학 기술 분야의 연구자들은 자원을 사용하는데 있어 편리한 인터페이스를 제공받게 되고 이는 과학 기술 분야의 연구를 보다 편리하게 하는데 도움을 주고 메타컴퓨팅 자원의 수요를 증가시키게 된다. 또한 자원 모니터링을 실시간 그래프를 통하여 제공함으로써 자원의 수요가 많은 시간과 그렇지 않은 시간을 가시적으로 확인할 수 있으므로 그리드 컴퓨팅을 구성하는 자원을 보다 효율적으로 관리할 수 있고 사용할 수 있게 된다.

참고 문헌

- [1] I. Foster, C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit Intl. J. Supercomputer Applications, 11(2):115-128, 1997
- [2] "Creating New Information Providers," MDS 2.1 GRIS Specification Document, USC/ISI, May, 2002
- [3] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit" Intl. J. Supercomputer Applications, 11(2):115-128, 1997

- [4] I. Foster and C. Kesselman (eds.) "The Grid: Blueprint for a new Computing Infrastructure" Morgan Kaufmann Publishers, 1998
- [5] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Services for Distributed Resource Sharing." Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001
- [6] John C. Peterson. "Multidisciplinary Integrated Design Assistant For Spacecraft (MIDAS)," http://mishkin.jpl.nasa.gov/Midas_Page.
- [7] V. Sunderam. "PVM: A Framework for Parallel Distributed Computing," *Concurrency: Practice and Experience*, Vol 2 No 4, December 1990
- [8] Bhatia, D., Burzevski, etc. "WebFlow-a visual programming paradigm for Web/Java based coarse grain distributed computing," *Concurrency: Practice and Experience*, March 1997. Java Special Issue.
- [9] R. P. Weston, etc. "A Distributed Computing Environment for Multidisciplinary Design," 5th SMAO, Panama City, FL, Sep. 1994.