

멀티미디어 통신 그리드의 설계 및 구현

김일민*

요약

그리드(Grid)는 WWW 컴퓨팅 모델을 계승할 차세대 분산 처리 모델이며, 자바 언어는 그리드 시스템을 설계하고 구현하는 훌륭한 개발 언어이다. 자바언어는 분산 처리를 위해 개발되었으며, 현재 WWW 응용 개발에 널리 사용되고 있다. 본 논문에서는 그리드의 기본 개념을 소개하고, 자바 언어를 사용한 컴퓨팅 그리드를 설계하고 구현하였다. 구현된 컴퓨팅 그리드에 알맞은 분산 응용 프로그램의 실행 결과를 분석하였다. 본 논문에서 구현된 그리드에 사용된 분산 멀티미디어 응용의 실행 결과는 기존의 시스템에 비하여 매우 향상된 성능을 보여주고 있다.

Design & Implementation of a Multimedia communication Grid

ilmin-Kim*

Abstract

Grid is a distributed computing model for next to the WWW computing model. Java language is developed for distributed computing and is being commonly used for WWW applications. Java can also be used for designing and implementing Grid systems. We introduced the basic concepts of Grid and implemented a new computing grid system. We test proper distributed applications on the multimedia grid system and analyzed the execution results. The execution results showed that the grid performance was much better than the legacy systems.

Keyword : Communication Grid, Grid System

1. 서론

개인용 컴퓨터의 계산 능력과 저장 용량의 확대로 인하여 분산 컴퓨팅 시스템이 보다 널리 쓰이게 되었다. WWW(world wide web)은 가장 널리 사용되고 있는 분산 컴퓨팅 시스템의 한 모델이다. 보다 효율적이며, 사용이 용이한 분산 컴퓨팅 시스템을 설계하고 구현하는 것은 현대 컴퓨터 관련 연구의 주요 주제 중에 하나이다. 인터넷 상에서 많은 자원을 공유하기 위해서 여러 가지 기술이 시도되었다. 소리바다와 같은 P2P(Peer-to-Peer)기술, 슈퍼컴퓨터들을 분산 환경에서 공유하기 위한 메타 컴퓨팅, 캠퍼스 네트

워크를 이용한 고성능 리눅스 클러스터 등이다.

많은 컴퓨터 연구자들은 웹 서비스의 다음의 분산 컴퓨팅의 가장 발전된 모델로서, 그리드 시스템을 구현하려고 노력하고 있다. 그리드 시스템에서는 네트워크에 연결된 다양한 자원을 통합하여 하나의 가상 컴퓨터처럼 보이도록 하는 것이다. 컴퓨팅 그리드에서 응용 프로그램은 사용한 가능한 여러 사이트의 컴퓨팅 파워를 연동하여 실행할 수 있어야 한다.

컴퓨팅 그리드 설계의 주요 요점은 유용한 여러 컴퓨팅 파워를 동적으로 연동시키는 것이다. 또한 컴퓨팅 그리드는 컴퓨터 사이트의 변화와 다양한 네트워크 프로토콜을 처리할 수 있을 용

1.1 그리드의 기본 서비스

그리드 컴퓨팅을 설계하기 위해서는 다음과 같은 주요 서비스를 포함하고 있어야 한다[3].

① 네이밍 서비스. 그리드에서도 다양한 객체를 등록하거나 검색할 수 있어야 한다. 예를 들어 다양한 컴퓨터 하드웨어, 서비스, 데이터 검

※ 제일저자(First Author) : 김일민
접수일자:2007년01월15일, 심사완료:2007년02월24일
* 한성대학교 컴퓨터 공학과
ikim@hansung.ac.kr
▣ 본 연구는 한성대학교 2006년도 교내 연구비 지원을 받아 수행되었음.

색이 네이밍 서비스에 포함된다. X.500의 이름 체계 또는 DNS에 따라 그리드 컴퓨팅 시스템의 네이밍 서비스를 구현할 수 있다.

② 계층적인 관리체계. 그리드 컴퓨팅 환경은 다양한 분산 컴퓨팅 컴포넌트로 구성되어 있으며, 언제나 확장이 가능한 구조이다. 이를 적절히 관리하기 위한 계층적인 관리 모듈을 구성할 필요가 있다.

③ 통신 서비스. 그리드 컴퓨팅에서 실행되는 응용 프로그램의 통신은 안정적인 1:1 통신에서부터 불안정적인 멀티캐스트 통신까지 매우 다양하다. 그리드 컴퓨팅의 통신 서비스는 분산객체에서 사용될 수 있는 대용량 데이터, 스트리밍 데이터, 그룹 통신 등을 지원한다.

④ 보안과 승인. 그리드 컴퓨팅은 다양한 분산 자원이 통합된 새로운 가상 자원을 제공한다. 이러한 가상 자원이 모든 사용자에게 제공될 수는 없으며, 적절한 권한을 가진 사용자나 프로세스에게 허용된 자원에 승인된 액세스만을 제공해야 한다. 이는 현재 웹의 보안에 비하여 훨씬 복잡하고, 어려운 문제이다.

1.2 그리드 프로젝트

전 세계적으로 많은 그리드 프로젝트가 수행되고 있으며, 이를 분류하면 다음과 같다[2]. 통합된 그리드 시스템, 시스템 미들웨어, 사용자 레벨 미들웨어, 적용 분야별 응용 그리드로 분류될 수 있다. 다음 표에서는 자바 언어와 .Net으로 구현된 시스템을 중심으로 나열하였다.

1.2.1 통합된 그리드 시스템

① javelin. UCSB에서 수행되고 있으며, 자바 기반 프로그래밍 실행시스템이다.

② Unicore. 독일에서 진행되고 있는 그리드 프로젝트이며, 원격 슈퍼컴퓨터 접근을 위한 자바 기반 시스템이다.

1.2.2 시스템 미들웨어

① GridSim. Monash 대학에서 진행 중이며, 그리드 시뮬레이션을 위한 toolkit이다.

② JXTA. 자바언어를 만든 Sun 마이크로 시스템에서 P2P 컴퓨팅을 위한 자바 기반 인프라를 제공하기 위한 것이다.

③ P2P Accelerator. Intel사에서 제작한 .Net을 사용한 P2P 응용을 위한 인프라이다.

1.2.3 사용자 레벨 미들웨어

① MIPICH-G. Northern Illinois University에서 Globus 툴킷을 사용하여 구현한 MPI이다.

② Cactus. MPICH-G와 Globus 사용하여 구현하였으며, 병행 처리 프로그램을 위한 기반 툴이다.

1.2.4 적용 분야별 그리드

① European DataGrid. CERN에서 주도하는 응용 그리드 프로젝트이며, 적용 분야는 에너지 물리학, 생물학, 지구과학 등이다.

② Virtual Laboratory. 제약 설계를 위한 분자 모델링이 목적이며, Monash 대학과 WEHI에서 공동 수행 중인 그리드 프로젝트이다.

2. 컴퓨팅 그리드 (Computing Grid)

컴퓨팅 그리드는 지역적으로 분산되어 있는 컴퓨팅 파워를 공유하여 마치 한 대의 고성능 컴퓨터처럼 사용할 수 있게 해 주는 것을 말한다. 그리드의 원형으로 자주 언급되는 SETI@Home 프로젝트가 대표적인 초기 사례이다. 현실에서 외계 생명체를 실제로 찾고 있는 것이 SETI (Search for Extraterrestrial Intelligence) 프로젝트이다. 전파 망원경을 이용하여 지구 밖의 신호를 수집하여 분석하는 작업을 진행한다. 그런데 방대한 외계의 신호 데이터를 분석하기 위해서는 고성능 슈퍼컴퓨터들이 필요하다. 그래서 고안된 것이 가정에서 전기만 허버하고 있는 전세계 수천만 대의 PC를 이용하는 것이다. SETI@Home 홈페이지(setiathome.ssl.Berkeley.edu)를 통해 자원 봉사자들을 모집하였다. 자원 봉사자들의 컴퓨터가 쉬는 동안에(화면 보호기가 작동하는 동안) 각종 관측 자료를 분석하는 프로그램을 수백만 대의 PC를 이용하여 수행하였다.

SETI@Home은 1997년 5월부터 2년 동안 미국 캘리포니아의 버클리 대학이 중심이 되어 세계 최대의 전파천체망원경 Arecibo의 데이터를 해석하여 외계의 지적 생명체를 탐구하는 프로

젝트였다. 이 프로젝트는 기존의 클라이언트·서버모델상에 유휴상태인 클라이언트 PC 파워를 사용하여 데이터를 해석하였다.

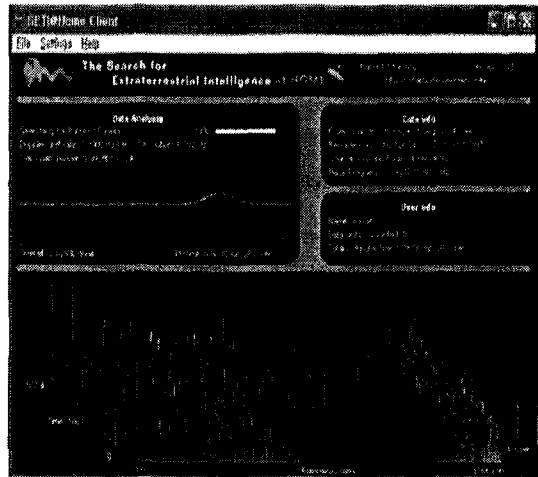
이와 같은 데이터의 해석작업 (Collect Data → Find Candidate Signals → Check Data Integrity → Remove Radio Interference → Identify Final Candidates)에는 고성능의 슈퍼컴퓨터가 이용되지만 이 프로젝트는 세상의 유휴자원을 이용하여 저가로 고속처리를 수행하는 것을 목표로 하였다. SETI@Home은 신기술을 사용한 것은 아니지만 큰 일거리를 나누어 처리하는 PC 그리드의 한 응용으로 인정되고 있다. 이 소프트웨어를 자신의 PC상에 설치하여 분할된 전파망원경의 데이터를 해석한 지원자는 2001년 5월에 300만명을 돌파하였다[4].

현재 가장 빠른 슈퍼컴퓨터는 IBM이 만든 ASCI White인데, 이것은 지금 미국 정부에서 사용되고 있다. 이 컴퓨터의 가격은 11억 달러이며 무게는 106톤이고 최고 성능은 12.3 Tera-Flops이다. 반면에, SETI@Home은 아스키 화이트 가격의 1% 미만인 경비로, 평상시에 SETI@Home 클라이언트는 약 70만 개의 작업 단위를 처리하는데 이것은 20Tera-Flops 이상이다. 그래서 당초 2년간의 프로젝트였지만 그 반응과 실적이 우수하여 연구기간을 연장하여 새로운 프로젝트의 수행을 계속하고 있다.

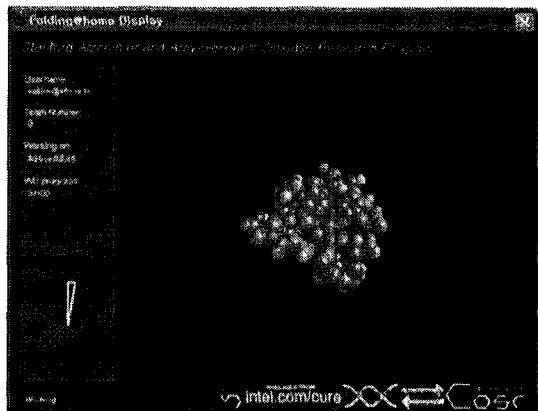
SETI@Home의 설립자 중 한 사람인 David P. Anderson 박사는 동일한 원리를 적용하여 United Devices사를 설립하였고 미국 암학회, 미국 국립 암 연구 재단 등과 단백질 구조 해석을 위한 프로젝트를 수행하고 있다. 이 프로젝트는 참여자의 PC가 처리한 데이터 양에 따라 포인트 점수를 부여하고, 그 포인트 점수에 따른 보상을 통하여 적극적인 참가를 유도하고 있다.

한편, 인텔은 P2P기술을 이용한 가상 슈퍼컴퓨터 기법을 사용한 암 연구프로그램으로 Intel* philanthropic peer-to-peer program을 수행하고 있다. (그림 1)은 SETI@Home v3.03 을 <http://www.setiathome.ssl.berkeley.edu/download.html>에서, (그림 2)는 인텔의 Folding @home을 <http://www.intel.com/cure/>에서 2002년 1월에 다운로드한 프로그램을 설치하여 수행한 화면이다. 현재 SETI@Home에는 390만 명 이상의 회원이 가입하고 있다[5].

현재 전세계적으로 많은 그리드 프로젝트가 진행 중이며, 그리드 개발을 위한 많은 기술적 인 표준과 개발도구가 개발 중이다[6]. 국내에서도 2001년 5월 정보통신부가 주축이 되어 국가 그리드 기본 계획이 수립되고, 이의 보급과 확대를 위한 Grid Forum Korea[7]가 구성되었다.



(그림 1) seti@home의 동작 화면



(그림 2) 인텔의 Folding@home의 동작화면

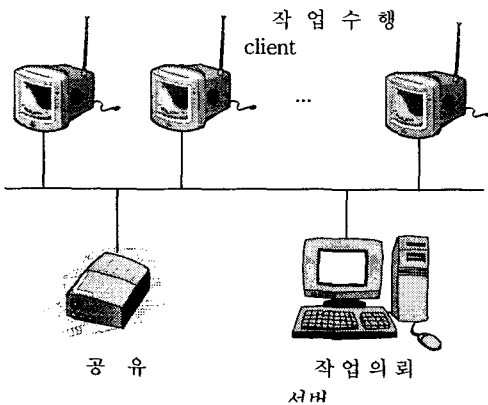
3. 자바 그리드의 설계

3.1 그리드 시스템의 구성

그리드 시스템은 참여한 분산 자원의 활용도를 감시하고, 사용되지 않고 있는 자원을 감지하여 활용할 수 있는 능력이 있어야한다. 컴퓨팅

그리드에서 실행될 분산 프로그램은 참여하는 시스템의 수, 시스템 하드웨어 및 운영체제 플랫폼 및 네트워크 상태에 대한 정보를 사전에 알 수 없으므로, 시스템 설계시에 시스템 융통성을 고려하는 것이 매우 중요하다. 인터넷으로 연결된 컴퓨터 시스템의 분산 투명도를 제공하는 자바 가상 머신(JVM)을 채택하여, 컴퓨팅 그리드의 설계와 구현이 보다 용이하도록 하였다. 다만 컴퓨팅 그리드를 사용하기 위해서는 프로그래머가 응용 프로그램에서 병행 실행이 가능한 모듈과 적절한 인자를 설정하여야 한다.

그리드를 구성하는 컴퓨터 시스템은 하나의 서버 노드와 다수의 클라이언트 노드로 나누어진다. 서버 노드는 클라이언트의 관리, 작업 분배, 실행 결과를 저장하는 기능을 수행하므로, 작업 의뢰 시스템이라고 한다. 클라이언트는 현재의 상태를 체크하여 작업 수행 가능여부를 판단하고, 이를 서버에 보고한다. 서버의 작업의뢰를 받아 작업을 수행하고 그 결과를 서버에 보고하므로, 작업 수행 시스템이라고 한다.



(그림 3) 컴퓨팅 그리드의 구성

3.2 그리드의 동작과정

그리드를 구성하는 모든 컴퓨터는 JVM(Java Virtual Machine)이 설치되어 있다고 가정한다. 컴퓨팅 그리드에 참가하기 위해서는 서버 홈페이지에 접속하여 클라이언트 자바 프로그램을 다운받아 실행하여야 한다.

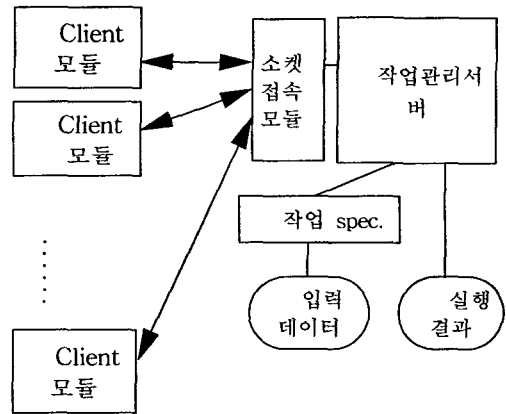
3.2.1 클라이언트 모듈의 역할

① 클라이언트 모듈은 서버 홈페이지에 접속

해서 누구나 다운받아 설치가 가능해야 한다. 자바언어를 기반으로 컴퓨팅 그리드를 설계하였으므로, 클라이언트 모듈은 자바 바이트 코드로 구성되며, JVM이 설치된 어떠한 컴퓨터 시스템에서도 실행시킬 수 있다.

② client 모듈이 설치된 컴퓨터의 CPU 사용 상태를 주기적으로 체크한다. 만일 CPU 사용률이 5%이하를 15분 동안 유지한다면, 로컬 컴퓨터가 비사용 상태라고 판단할 수 있다. 물론 로컬 컴퓨터를 사용하는 사용자가 이러한 옵션을 지정할 수 있어야 한다.

③ 로컬 컴퓨터가 비사용 상태이면, 서버 모듈에 접속하고 실행할 데이터를 받아와 서버가 요구한 부분의 컴퓨팅을 수행한다. 수행이 완료되면, 수행결과를 서버로 전송한다. 수행 중 로컬 컴퓨터가 다시 사용상태가 되면, 실행을 중단하고, 서버 모듈에 cancel 메시지를 전송한다.



(그림 4) 컴퓨팅 모듈의 구성

3.2.2 서버 모듈의 역할

① 서버 모듈은 두 개의 서브 모듈로 나누어진다. 첫째 실행 요구를 접수하는 모듈과 실행 결과를 받아들이는 모듈이다. 실행 요구를 받아들이는 서버 모듈은 서버 소켓을 사용하여 클라이언트의 소켓 접속 요구를 대기한다. 클라이언트의 접속요구가 있으면, 새로운 스레드를 생성하여 실행 결과를 받아들이는 모듈을 실행시킨다. 클라이언트에게 실행 요구를 보낸 입력 데이터는 적절한 마크를 하여 다른 클라이언트에게 중복 실행 요구를 하지 않도록 한다.

② 실행 결과를 받아들이는 모듈은 정상적인

실행결과와 비정상 종료 두 가지로 나누어진다. 정상 실행결과는 주어진 시간내에 결과 값이 클라이언트로부터 서버로 반환된 경우를 의미하며, 결과 파일에 반영되어야 한다. 비정상 종료는 클라이언트가 비정상 종료 메시지를 보낸 경우와 일정시간이 지나도 아무런 응답이 없는 경우이다. 이러한 경우는 입력 데이터를 실행 요구전 상태로 복구하여, 새로운 클라이언트에게 전송할 수 있도록 준비한다.

```
int[] bool=new int[Size];
while(true) {
    find available client c & empty slot of bool
b:
    create a thread
    { mark b on bool;
      send msg to c with b;
      get result from c;
    }
}
```

(그림 5) 서버의 프로그램

```
while(true) {
    check current CPU status;
    if (CPU is available) {
        send msg. to the Server;
        get job from the Server;
        compute;
        return the results;
    }
    sleep(some_constant);
}
```

(그림 6) 클라이언트의 실행 프로그램

본 논문에서 구현한 컴퓨팅 그리드는 기존의 PC 그리드를 문제 spec 모듈과 서버 컨트롤 모듈로 구분하여, 새로운 컴퓨팅 문제를 보다 손쉽게 그리드로 처리할 수 있도록 하였다.

4. 그리드의 성능평가

4.1 구현 환경 및 평가 프로그램

시험 시스템 구축을 위하여 네트워크에 연결된 연구실의 컴퓨터 시스템을 사용하여 그리드를 구성하였다. 각 시스템들은 10Mbps 이더넷 카드를 장착하였으며, MS 윈도우즈 운영체제를

설치하였다. 시험 시스템은 9대의 컴퓨터로 구성되었으며, 8대의 PC는 그리드의 컴퓨팅 서버로 사용되며, 1대는 그리드 작업의뢰 시스템으로 사용되었다. 성능 평가 실험은 복합 행렬 곱셈(Complex Matrix Multiplication)과 소수 계산(Prime Number Detection)을 사용하여 실행하였다. 이중 복합 행렬 곱셈 벤치마크는 행렬의 크기에 따라서 구분하였으며, MatMult_A(100x100), MatMult_B(400x400), MatMult_C(800x800)으로 나누어 수행하였다.

복합 행렬 곱셈은 3차원 그래픽 응용 프로그램에서 많이 사용되는 것으로 입면체를 이동하거나 회전시키기 위해 입면체를 다각형 단위로 나누어 다각형을 이루는 점변환 연산에 사용된다. 이는 입면체를 구성하는 많은 수의 점들에 대한 동일한 연산을 수행하는 것으로 연산할 데이터를 입출력하는데 많은 네트워크가 필요하지만, 컴퓨팅 그리드에 적용할 수 있는 데이터의 병렬 수행이 가능한 특징을 가진다.

소수를 찾는 계산은 주어진 정수가 소수인지 아닌지를 판단하는 연산으로, 큰 소수는 암호화 및 복호화 키의 중요 요소로 사용된다. 소수 계산은 특정 정수 범위를 여러 조각으로 나누어 독립적으로 수행하는 것이 가능하므로, 그리드 컴퓨팅에 적절한 응용이다. 또한 소수 계산은 전체 수행에서 통신비용의 비율이 낮아서 복합 행렬 곱셈과는 대조되므로, 벤치마킹에 적절하다고 사료된다.

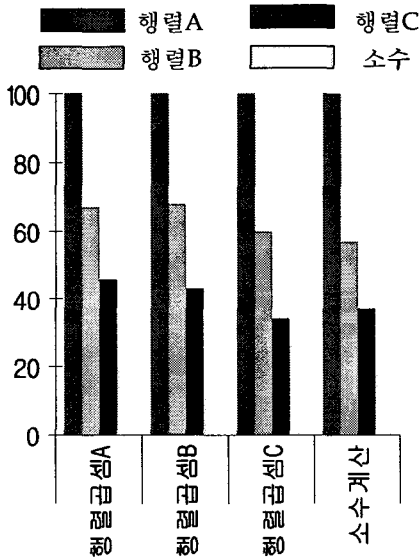
4.2 실험 결과 및 분석

실행한 여러 벤치마크 결과를 표 1과 같이 나타내었다. <표 1>을 살펴보면, 그리드를 구성한 컴퓨터의 수와 이에 따른 성능향상의 비율을 알 수 있다. 구성 컴퓨터의 수가 1인 경우는 기존의 호스트 기반 실행과 동일한 환경이다.

일반적으로 컴퓨팅 그리드를 구성하는 컴퓨터의 수가 증가함에 따라서, 병렬로 수행되는 코드 부분은 동일한 비율로 증가하므로, 일반적인 병렬 컴퓨터의 성능 향상비와 비슷함을 알 수 있다. 특히 그리드에 참여하는 컴퓨터의 수가 증가함에 따라서, 소수를 계산하는 응용의 성능이 많이 향상되는 것은 전체 수행시간 중에서 통신비용이 차지하는 비율이 낮기 때문이라고 판단된다.

<표 1> 컴퓨팅 그리드 수행 시간(초)

컴퓨터수	1	2	4	8
행렬곱셈 A (100)	5.16	3.46	2.36	1.87
행렬곱셈 B (400)	243.9	142.1	73.3	54.4
행렬곱셈 C (800)	2127.2	1271.1	715.5	561.1
소수 계산	1120.4	638.4	414.6	209.5



(그림 7) 실행시간 비교

각 응용 프로그램에 따른 성능향상을 보이기 위해서 그리드 구성 노드 수에 따른 실행 시간의 감소 비율을 (그림 7)로 나타내었다. 왼쪽의 적색바는 단독으로 처리하였을 때 요구되는 컴퓨팅 시간을 100으로 하였다. 그리드의 구성 컴퓨터의 수에 따라 요구되는 수행 시간의 비율을 나타낸 것이다. 즉 두 번째 막대는 구성 컴퓨터의 수가 2대인 경우, 세 번째 막대는 4대인 경우, 4번째 막대는 8대인 경우의 실행 시간을 나타낸 것이다. <표 2>에서 알 수 있는 것은 전체 수행 시간에서 병행 처리 시간의 비율이 높을수

록 성능향상이 더 많이 됨을 보여 주고 있다.

5.결론

본 논문에서 구현한 컴퓨팅 그리드는 네트워크로 연결되어 있는 PC, 프린터, 파일서버 등의 자원을 하나로 통합하여 활용함으로써 고성능을 필요로 하는 어플리케이션의 수행을 위한 플랫폼으로 사용될 수 있음을 보여 주었다. 또한 컴퓨팅 그리드는 고성능 컴퓨팅 파워를 제공하는 저비용 시스템의 한 예이며, 이는 데이터 그리드 및 가상 컴퓨터로 발전할 초기의 시범 시스템이라고 할 수 있다. 제한된 실험실 환경에서 테스트 결과는 기존의 수퍼 컴퓨터의 성능과 비교될 만큼 우수하다고 사료된다. 특히 여러 모듈이 독립적으로 병행 실행될 수 있는 경우에 그리드의 컴퓨팅 능력을 최대로 활용할 수 있다. 이 논문에서도 그리드의 컴퓨팅 능력을 최대로 활용할 수 있는 행렬의 곱셈 및 소수의 계산과 같은 예제를 사용하였으므로, 실제의 많은 응용 프로그램의 성능향상은 보다 제한적일 것이다. 그리드의 컴퓨팅 네트워크로 연결된 실제 오피스 환경에서 사용될 경우의 많은 오류와 보안 문제는 앞으로 연구해야할 과제라고 사료된다.

참고문헌

- [1] Ken Arnold and James Gosling. The Java Programming Language 4rd Edition, Sun Microsystems, Inc. 2005.
- [2] Meyer, T. Downing, Java Virtual Machine. O Reilly, 1997.
- [3] I. Foster, C. Kesselman, S.Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organization. International J. Supercomputer Applications, 15(3), 2001.
- [4] Mark Baker, Rajkumar Buyya, and Domenico Laforenza. Grids and Grid technologies for wide-area distributed computing, Software-Practice and Experience. 2002.
- [5] I. Foster, C. Kesselman. Blueprint for a New Computing Infrastructure, 2nd edition Morgan-Kaufmann, 2004.
- [6] Fran Berman, G. Fox, A. Hey, Grid Computing Maki

ng the Global Infrastructure a Reality, Wiley, 2003.

[7] Grid Forum Korea web-page <http://www.gridforumkorea.org/>

김 일 민



1984년 : 경북대학교 전산과 졸업

1996년 : 아리조나주립대학교 박사

1985년 - 1987년 : 한국 전자 통신

연구소(ETRI) 연구원

1996년 - 1997년 : 삼성 SDS 멀티

미디어 교육센터 책임

1997 - 현 재: 한성대학교 컴퓨터공학과 교수

관심분야 : 멀티미디어 통신, 운영체제