

최종 연구보고서

고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축

(Implementation of a Computational Grid based
on Support from High-Performance Cluster
Resources)

2003. 12. 12.

연구기관 : 포항공과대학교

주 의

1. 이 보고서는 한국과학기술정보연구원에서 위탁연구과제로 시행한 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 한국과학기술정보연구원에서 시행한 위탁연구과제의 연구결과임을 밝혀야 합니다.

연구
과
제
명

2
0
0
3

한
국
과
학
기
술
정
보
연
구
원

최종 연구보고서

고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드
구축

(Implementation of a Computational Grid based
on Support from High-Performance Cluster
Resources)

2003. 12. 12.

연구기관 : 포항공과 대학교

한국과학기술정보연구원

제 출 문

한국과학기술정보연구원장 귀하

본 보고서를 “고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축”에 대한 연구의 최종 보고서로 제출합니다.

2003년 12월 12일

수탁기관 : 포항공과대학교

수탁기관장 : 박찬모 (인)

연구책임자 : 이승구 (전자전기)

참여연구원 : 김상철 (전자전기)

김용호 (컴퓨터)

이일평 (전자전기)

보고서 초록

연구과제코드	G-03-SD-01-24R-2			연구기간	2003.4.15 - 2003.12.15	
연구과제명	고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축 (Implementation of a Computational Grid based on Support from High-Performance Cluster Resources)					
연구책임자	이승구	참여연구원수	총 : 3 명 내부 : 3 명 외부 : 명	연구비	정부: 20,000천원 기업: 천원 계: 20,000천원	
연구기관명 및 소속부서명	포항공과대학교		참여기관명	전자전기공학과		
요약(연구결과를 중심으로 개조식 600자 이내)					보고서 면수	29
<p>본 연구실은 리눅스 운영체제가 설치되어 있는 고성능 PC 노드 40대를 보유하고 있다. 각 PC 노드들은 Pentium III, 미리넷 및 이더넷 네트워크로 연결되어 있기 때문에 그리드 자원으로 사용하기에 적합한 자원이라고 판단된다. 이 고성능 클러스터를 그리드 자원의 일부로 연결하기 위해서 KISTI 기관과 협력하여 관련 패키지 설치에 주력하였다.</p> <p>그리드 자원의 일부로 동작할 수 있는 고성능 클러스터를 구성하는데 성공하였다. 먼저 이 일을 해 주기 위해서 여러 가지 클러스터 및 그리드 관련 패키지를 설치하여야만 했다. 글로버스 2.2.4와 PBS, MPICH-G2를 성공적으로 설치하고 내부 클러스터 연동을 통해서 그리드의 일부 자원으로서 사용될 수 있는지에 대해서 확인하였다. 실제 KISTI에서 포항공대 클러스터와의 GRAM 및 기타 그리드 연동을 테스트 해 본 결과 연동이 성공이라는 것을 보고 하였다. 그 후 제공 자원의 모니터링을 위해 SCMSWEB 모니터링 툴을 설치하고 이 툴을 통해 자원이 어떻게 사용되고 있는지 보여주었다.</p> <p>이 보고서는 그 설치 과정을 자세하게 보여 주고 또한 문제점에 대해서 상세히 보고해주고 있다. 만일 누군가가 자신이 고성능 클러스터 자원을 보유하고 있고 그리드 자원의 일부로 참여하고 싶어 한다면 그 과정이 어떠한지 그리고 무엇을 해야 하는지에 대해서 상당히 알고 싶어 할 것이다. 이 보고서는 그러한 사람에게 매우 필요할 것으로 생각된다.</p> <p>마지막으로 독립적인 기관들은 글로버스 및 기타 다른 패키지를 설치할 때 그 경험을 될 수 있으면 많이 공유하는 것이 좋다. 이를 위해 웹 페이지를 통해서 빠르게 정보를 공유할 수 있는 서비스를 제공 하는 것이 최우선적으로 필요하다.</p>						
색인어 (각 5개 이상)	한 글	클러스터, 글로버스, 그리드, 자원, 정보				
	영 어	cluster, globus, grid, resource, information				

최종연구보고

1. 위탁과제명 : 고성능 클러스터 자원제공을 통한 컴퓨팅 그리드 구축

2. 수탁기관 : 포항공과대학교

3. 연구책임자 : 이승구

4. 일정계획 대 진도

정상 진행, 현재 시점까지 100% 완료

5. 주요 연구개발 추진 내용 및 최종결과

그리드를 구축할 수 있도록 클러스터 자원에 글로버스 및 관련 툴킷 설치 완료, KISTI에서의 연동 테스트 완료, SCMSWEB 모니터링 툴 설치 완료.

6. 추진상의 문제점 및 앞으로의 추진 대책

서로 다른 기관들 사이의 정보 공유 부족, 정보 공유를 위한 수단 (웹 페이지 등) 구축이 필요

7. 주요 연구 기자재 및 시설 사용 계획

없음

요 약 문

I. 제 목

고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축

II. 연구개발의 목적 및 중요성

본 연구는 고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축을 목적으로 한다. 본 연구 책임자의 연구실에는 56 CPU를 사용하는 고성능 클러스터 시스템이 있다. Intel Pentium III 933Mhz의 CPU를 두개씩 탑재한 컴퓨터가 16대가 있으며 Intel Pentium III 733Mhz 의 CPU를 탑재한 고성능 클러스터용 컴퓨터가 24대 있다. 전자는 Myricom사의 Myrinet 2000을 이용하여 노드당 초당 4기가비트(전송 2기가비트 + 수신 2기가비트로 합이 4기가 비트) 대역폭을 갖는 고성능 네트워크로 연결되어 있으며, 후자는 Myricom사의 Myrinet을 이용하여 노드당 초당 2.56기가비트(전송 1.28기가비트 + 수신 1.28기가 비트로 합이 2.56기가 비트)의 대역폭을 갖는 고성능 네트워크로 연결되어 있다. 이러한 고성능 컴퓨터 자원을 국내 과학자들과 효율적으로 공유하기 위해서 Globus Toolkit 2.2 및 기타 그리드 지원 소프트웨어를 개발 또는 지원할 것이다.

III. 연구개발의 내용 및 범위

본 연구에서 수행하고자 하는 연구는 국내외 슈퍼컴퓨터와 고성능 클러스터를 단일 자원처럼 연동할 수 있는 그리드 테스트베드를 구축하기 위한 자원 확보를 지원하기 위한다. 이를 위해서 본 연구 책임자의 연구실에서는 고성능 클러스터 시스템을 그리드 시스템에 연동시키기 위한 지원을 수행한다. 본 연구 책임자의 연구실에서 지원할 클러스터 시스템에는 Linux 2.4.2를 사용하고 있다. 이 시스템에 Globus 2.2를 설치하여 현재 그리드 연동 테스트를 수행중에 있다. 이 테스트를 국가적 그리드 시스템에 적용

시키기 위한 방법을 연구하도록 한다.

본 연구 책임자의 연구실에는 56 CPU의 클러스터 시스템을 보유하고 있다. 우선 16대의 Pentium III 933Mhz CPU를 두개씩 장착한 메모리 256메가 바이트의 고성능 클러스터 시스템이 있다. 이 시스템은 리눅스 커널 2.4.2를 사용하고 있으며 Globus Toolkit 2.2를 지원하고 있다. 이 시스템은 두개의 개별 네트워크를 가지고 있다. 우선 기가 대역의 초고속 네트워크로 Myricom사의 Myrinet 2000을 사용하고 있다. Myrinet 2000은 양방향 각각 초당 2기가비트의 대역폭을 제공하는 초고속 네트워크로 일반 PC의 주변장치 연결 버스인 PCI보다 2배정도 빠른 속도를 제공하고 있다. 이를 위해서 Myrinet 2000으로 연결된 시스템은 PCI 시스템도 보다 좋은 사양을 사용해야 한다. 또한 100메가비트의 고속 이더넷을 32포트의 스위칭 허브를 통해 연결되어 있다. 이 두개의 네트워크를 공통적으로 사용하여 보다 유연성있는 시스템의 구축이 가능하다. 그리고 24대의 Intel사의 Pentium III 733 Mhz CPU를 탑재한 메모리 128메가 바이트의 고성능 클러스터 시스템이 있다. 이 시스템에도 역시 리눅스 커널 2.4.2를 사용하고 있으며 Globus Toolkit 2.2가 설치되어 있다. 이 시스템도 역시 두개의 네트워크 시스템을 가지고 있다. 우선 Myricom사의 Myrinet을 이용한 네트워크이다. 이 네트워크는 양방향 각 초당 1.28기가비트의 대역폭을 지닌 초고속 네트워크이다. 또한 초당 100메가비트의 대역폭을 지닌 고속 이더넷을 32포트의 스위칭 허브를 통해 연결되어 있다. 두 시스템은 현재의 메모리 용량이 조금 부족하다고 할 수 있으나 조만간 이를 확충할 계획이다.

포항공과 대학교의 네트워크는 KREONET(연구전산망)에 연결되어 있고 HP V2500 슈퍼컴퓨터를 보유하고 있으며 2002년에 이 슈퍼컴퓨터가 국가 그리드 시스템에 연동되었다. 이 슈퍼컴퓨터와 더불어 본 연구를 통한 고성능 클러스터 시스템의 제공으로 국가 그리드 시스템의 구축에 적극 기여할 것으로 사료된다.

IV. 연구개발결과

현재 그리드 자원의 일부로 동작하기 위한 고성능 클러스터를 구축하였다. 각 클러스터 노드의 운영체제는 리눅스 7.3을 기반으로 하였고 그리드 미들웨어는 글로버스 2.2.4를 설치하였다. 모든 사람들에게 작업을 공정하

게 할당하도록 하기 위해서 PBS 일괄 처리 큐잉 시스템을 설치하였고 이것은 클러스터 노드가 사용가능할 때마다 작업이 할당되게 한다. 또한 구축된 클러스터 자원은 MPICH-G2 1.2.5가 설치되어 병렬 작업이 할당될 수 있도록 한다. 따라서 사용자는 순차적 작업 또는 병렬 작업을 실행시킬 수 있도록 하였다. 클러스터 자원은 TCP Wrapper를 사용하여 비 신뢰적인 외부의 접근에 대해서 차단하도록 하였기 때문에 보안면에 있어서도 매우 안전하다. KISTI 주관 연동 테스트를 통해서 이미 시스템이 매우 안정적으로 동작하고 정상 동작함을 확인하였다. 웹 상으로 클러스터의 상태를 모니터링 할 수 있는 SCMSWEB 서비스도 설치가 되어 클러스터의 상태를 확인가능하다.

V. 연구개발의 활용에 대한 건의

서로 다른 기관들 간에 독자적으로 그리드 구축을 위한 글로버스를 설치하다보니 상호간에 정보 교류가 쉽지가 않은 실정이다. 예를 들면 클러스터 같은 자원인 경우에는 그 설치 방법이 대부분 대동소이하다. 각 기관에서 글로버스를 독립적으로 설치하여 쉽게 성공할 수 있으면 별문제가 없지만 초보자에게는 매우 고역일 것이다. 이럴 경우 미리 설치 경험이 있는 전문가와 정보 교류가 잘 되어 있어서 프로그램 설치에 대한 정보를 공유할 수 있다면 일이 훨씬 빨리 진행 될 것이다.

그리고 전체 기관의 글로버스 연동이 성공하게 되면 그 과정이 어떠했는지 그리고 어디까지 진행되었는지에 대해서 연동을 진행한 기관에서만 알 것이 아니라 모든 기관들이 그 정보를 공유하는 것이 필요하다. 독립적인 기관에서도 전체적으로 일이 어떻게 진행되는지 알아야지만 자신의 현재 위치를 알고 향후 나갈 방향을 잡을 수 있기 때문이다.

앞으로 훨씬 더 좋은 성능의 클러스터 자원을 제공하기 위해서는 많은 패키지를 설치해야 할 것으로 예상된다. 미리 패키지에 대한 사전지식에 대해서 전해주고 그것의 중요성, 그리고 동작이 성공했을 때의 결과 등에 대해서 알려 줄 필요가 있다. 이렇게 되었을 때 패키지를 설치하는 각 기관에서는 당황하지 않고 일을 수행할 수 있게 된다.

VI. 기 대 효 과

본 연구를 통해서 국가 그리드 구축을 위한 테스트베드(Testbed)를 만들 수 있다. 이 테스트베드를 이용하여 추후 국가 그리드 시스템을 구축하기 위한 기반 기술을 익힐 수 있으며, 그 기술을 통해서 높은 그리드 시스템을 구축할 수 있다. 그리고 구축된 그리드 시스템은 다른 과학 기술 및 사회 과학 발전을 위한 계산 목적의 그리드 시스템으로 사용될 수 있을 것이며, 그것을 통해서 많은 학문적, 기술적 발전을 가져올 수 있을 것이다.

Summary

I . Title

Implementation of a Computational Grid based on Support from High-Performance Cluster Resources

II. Objective of the study and its importance

The aim of this research is to build a computing Grid by providing high-performance cluster resources. We have a high performance cluster system that has 56 CPU resources. Each of 16 nodes in that has 2 CPUs (which operates 933MHz) and each of 24 nodes has 1 CPU (which operates 733Mhz). All of these CPUs are Intel processors. The former are connected via high performance network called Myrinet which is from Myricom company), producing 4 Giga bits per second in bandwidth while the latter are connected via also Myrinet but producing 2.56 Giga bit per second. To share these high performance computing resources with domestic scientists, we'll develop or support Globus Toolkit 2.2 or other softwares.

III. Content and scope of the study

This research focuses on supporting resources to build Grid infrastructure consisting of several resources such as super computer or high performance clusters. To accomplish this, we'll support high performance cluster systems to interact with other grid resources. We're currently using Linux 2.4.2 Operating systems on cluster. This system is also under testing Grid operations by installing Globus 2.2 middleware. The research is totally focused on application of this cluster resources to

domestic Grid testbed system.

We have a system that consists of 56 CPUs. Each of 16 nodes in that system has Pentium III 933Mhz based CPU and 256 Mbytes memory. This system uses linux kernel 2.4.2 and supports Globus Toolkit 2.2.4. This system has two separate networks. First of all, it uses Myrinet 2000 system from Myricom company. Myrinet 2000 provides 2 Gbit/s in bandwidth, which is more faster than those of PCI components by a factor of two. Because of that, a system connected with Myrinet 2000 should use a better component. Also it is connected with 32 ports switching hub that operates 100Mbit/s. A flexible system can be build by using thsesse two networks in common. And we have a cluster system that have 24 nodes each of which has Intel based 733 MHz CPU and 128 Mbytes of memory. This system also has linux operating system, version kernel 2.4.2 and Globus toolkit 2.2. This system is connected with Myrinet 1280 which produces 1.28 Gbit/s. Two systems are currently deficient in memory but we'll compensate it soon.

Network in POSTECH was constructed in the early term of year 2003 attaching to KREONET and POSTECH has HP V2500 supercomputer which is used in the national Grid system. Our cluster system together with this supercomputer is expected to contribute to building the national Grid system.

IV. Result of the study

We build a high performance cluster system that can be a part of resources in the national wide Grid. Each operating system is based on Linux 7.3 and Globus 2.2.4 is installed as a Grid middleware.

To distributed a Grid job fairly to people who want to use resources, PBS batch queuing system is installed, which allocated jobs whenever an idle nodes are found. Also, MPICH-1.2.5 was installed to run the job in parallel. Therefore, users can execute the sequential or parallel job as they wish. Cluster resources are fundamentally blocked off from outside

attack by using TCP Wrapper, this will keep secure system. We confirmed that our system is highly reliable and operates in normal by testing KISTI co-operating tests. SCMSWEB service, the service that provides monitoring service of cluster via WWW has been installed and status of cluster can be monitored on the WWW.

V. Suggestion for utilization

It's not easy to communicate with information among the different institutions because the Globus toolkit is installed on them separately. For example, in case of cluster resources, the installation procedure is almost the same. If that installation is successful on each institution, that is not a problem but it's very burdensome for beginner to Globus toolkit. In this case, the fast sharing of information between expert who has experiences in installing will accelerate the work.

And after accomplishing the interacting test on Globus systems, all information from that should be shared among not only the institution that accomplish the work but also other independent institutions. We can get in our way well after each institution is able to be aware of the current situation.

In the future, we expect that more packages must be installed on cluster resources for high performance. We need to get information of those packages in advance. That is, we have to be notified of the importance of those packages and the results when the installation was successful. We can work well after this works well.

VI. Expected Effects

We can build the national Grid testbed by this research. If we use this testbed, we can learn the technologies based on building national wide Grid system afterwards, through which high performance Grid can

be constructed. And the system will be used in scientific and social department or other computing-purpose field. This will bring to develop the knowledge and technology.

Contents

I. Introduction.....	13
1. The purpose of research	
2. The importance of research	
3. Application	
II. Body.....	14
1. Preliminaries	
2. High performance PC cluster	
3. The procedure of toolkit for high performance Grid resources	
(A) Installation of PBS	
(B) Installation of Globus	
(C) Installation of MPICH-G2	
(D) Installation of SCMSWEB Monitoring Tool	
(E) Setting user environment	
III. Conclusion	29

목 차

I. 서론	13
1. 연구의 목적	
2. 연구의 중요성	
3. 활용 방안	
II. 본론.....	14
1. 개요	
2. 고성능 PC 클러스터	
3. 고성능 그리드 자원을 위한 툴킷 설치 과정	
가. PBS 설치	
나. 글로버스 툴킷 설치	
다. MPICH-G2 설치	
라. SCMSWEB 모니터링 툴 설치	
마. 사용자 환경 설정	
III. 결론.....	29

표차례

❖ 표 없음

그림차례

<그림 1> NASA의 OVERFLOW-D2 예제	4
<그림 2> 제공된 고성능 클러스터 시스템	16
<그림 3> greet.c 프로그램	22
<그림 4> POSTECH 클러스터가 그리드 웹 페이지에 등록된 모습	28
<그림 5> POSTECH 클러스터 정보 화면	28
<그림 6> POSTECH 클러스터 모니터링 페이지 화면	28
<그림 7> POSTECH 클러스터 내의 hush1노드에 대한 자원 활용 현황	28

I. 서론

1. 연구의 목적

본 연구는 고성능 클러스터 자원 제공을 통한 컴퓨팅 그리드 구축을 목적으로 한다. 본 연구 책임자의 연구실에는 56 CPU를 사용하는 고성능 클러스터 시스템이 있다. Intel Pentium III 933Mhz의 CPU를 두개씩 탑재한 컴퓨터가 16대가 있으며 Intel Pentium III 733Mhz 의 CPU를 탑재한 고성능 클러스터용 컴퓨터가 24대 있다. 전자는 Myricom사의 Myrinet 2000을 이용하여 노드당 초당 4기가비트(전송 2기가비트 + 수신 2기가비트로 합이 4기가 비트) 대역폭을 갖는 고성능 네트워크로 연결되어 있으며, 후자는 Myricom사의 Myrinet을 이용하여 노드당 초당 2.56기가비트(전송 1.28기가 비트 + 수신 1.28기가 비트로 합이 2.56기가 비트)의 대역폭을 갖는 고성능 네트워크로 연결되어 있다. 이러한 고성능 컴퓨터 자원을 국내 과학자들과 효율적으로 공유하기 위해서 Globus Toolkit 2.2 및 기타 그리드 지원 소프트웨어를 개발 또는 지원할 것이다

본 연구의 목표는 국가적 그리드 시스템 구축을 위한 자원 제공을 목표로 한다. 이를 위해서 고성능 클러스터를 그리드 시스템의 연동을 위한 자원으로 제공하고, 연동을 위한 OS, 네트워크, 보안등을 지원하고, 기타 제반 사항에 대한 협력을 목표로 한다. 이를 통해 국가 그리드 시스템 구축을 하고, 국가 그리드 시스템을 사용하여 타 기관등의 응용 프로그램 개발등에 협력할 수 있도록 한다.

2. 연구의 중요성

본 연구를 통해서 국가 그리드 구축을 위한 테스트베드(Testbed)를 만들 수 있다. 이 테스트베드를 이용하여 추후 국가 그리드 시스템을 구축하기 위한 기반 기술을 익힐 수 있으며, 그 기술을 통해서 높은 그리드 시스템을 구축할 수 있다. 그리고 구축된 그리드 시스템은 다른 과학 기술 및 사회 과학 발전을 위한 계산 목적의 그리드 시스템으로 사용될 수 있을 것이며, 그것을 통해서 많은 학문적, 기술적 발전을 가져올 수 있을 것이다.

3. 활용방안

과학 기술의 발전을 위해서는 높은 성능의 컴퓨터가 반드시 필요하다. 항공기 구조 설계, 핵 융합 반응의 전산 모사, 생명 공학 연구, 반도체 회로 설계 등 일반 개인용 컴퓨터로써는 상상도 할 수 없을 정도의 빠른 계산을 요구하는 응용프로그램들이 많다. 이러한 응용프로그램들은 현재까지 병렬 컴퓨터를 이용하여 계산되어

져 왔으나, 그 비용이 너무 높았다. 병렬 컴퓨터는 각각의 컴퓨터마다 서로 다른 부속품을 사용하기 때문에 고비용 뿐 만 아니라 그 성능 유지에도 많은 비용이 들어가고, 추후 환경 변화(새로운 병렬 컴퓨터를 구입하는 등의)가 발생할 때 사용하던 응용프로그램을 바로 수행시키는데 문제가 있는 등, 많은 문제점들을 야기하고 있다. 이에 요즘에는 분산 컴퓨터를 이용하여 위에서 열거한 여러 가지 문제들을 풀고자 하는 노력들이 최근 몇 년간 있어왔다. 그러한 노력들은 좋은 결과를 보여 왔으나, 아직까지도 분산 컴퓨터 시스템의 계산 용량은 많이 부족함이 사실이다.

국가적으로 초고속 네트워크로 연결되어 있는 모든 컴퓨터들을 그리드 시스템으로 묶어서 사용할 수 있다면 거대한 과학 기술 분야의 발전에 많은 도움을 줄 수 있을 것이다. 이는 빠른 계산 속도를 위한 것 뿐 만 아니라 저비용의 제품 생산이

라는 면에서도 매우 중요하다. 그리드 시스템은 기존의 병렬 컴퓨터처럼 고비용을 요구하지 않는 새로운 기술이다. 그러므로 이 기술을 사용하면 기존에 병렬 컴퓨터를 구입, 유지 비용들이 대폭적으로 감소하게 되어 생산성 향상에 큰 도움이 될 것이다. 위에서 언급한 Globus를 이용하여 이미 몇몇의 성공적 결과를 보이고 있다. 그림 1은 이러한 결과물의 예를 보여준다. 그림 1은 1500개의 프로세서를 이용하여 항공기의 대기 흐름을 전산 모사 한 결과를 보여주는 것이다. 이러한 응용들은 과학 기술 분야의 발전과 대한민국의 산업 발전에 큰 영향을 미칠 것이다. 성공적 개발 뒤에는 항공기 구조

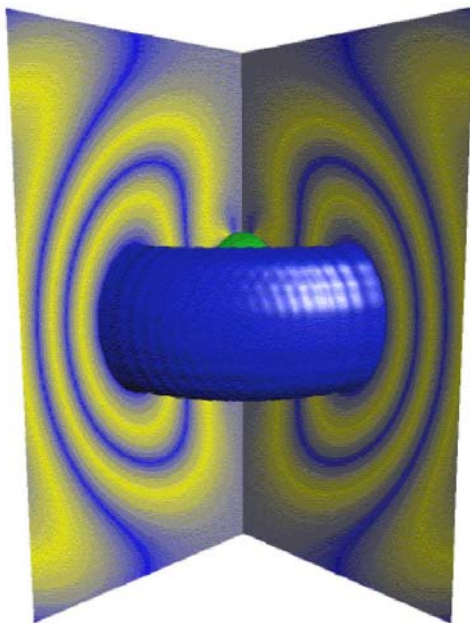


그림 1 NASA의 OVERFLOW-D2
예제

설계나, 반도체 회로 설계 분야 등에서 계산 그리드를 사용하여 저비용으로 고성능의 제품을 생산할 수 있게 될 것이다.

II. 본론

1. 개요

본 연구를 수행하기 위해서는 기본적으로 이미 구축되어진 클러스터 시스템을 요구로 한다. 본 연구 책임자의 연구실에서는 수년간 구축해온 고성능 클러스터 시

시스템을 보유하고 있다. 이 시스템을 그리드 시스템과 연동할 수 있도록 하고자 한다. 이를 위해서 우선 그리드 시스템 구축에 필요한 다양한 문제들을 해결해야 한다. 우선 그리드 시스템 구축을 위해서 Globus Toolkit을 설치하여야 한다. Globus Toolkit을 설치하고 그것을 이용한 간단한 연동 실험을 수행하도록 한다. 연동 실험이 성공하게 되면 다시 MPI나 GM과 같은 고성능 네트워크를 사용하여 병렬 프로그램을 작성할 수 있는 통신 프로토콜을 지원하는 툴킷의 설치가 필요하다. 이 toolkit을 설치하고 Globus 와의 연동 시험을 수행하여야 한다. Globus 와의 연동이 완료되면 기타 응용 프로그램 개발에 필요한 C, C++, Fortran 등의 언어에 대한 컴파일러 설치 역시 필요하다. 이러한 기반 시스템 구축을 한 후에는 다시 KREONET 등의 기존에 사용하지 못하던 고속 네트워크와의 연동이 필요하다. 이 연동이 끝난 후에는 클러스터 시스템을 사용하는 사용자들의 편의성을 향상시키기 위한 노력이 필요할 것이다.

본 연구에서는 위에서 언급한 내용들을 모두 지원할 것이다. 우선 현재 일부 시스템에만 설치되어 있는 Globus Toolkit 2.2를 모든 노드로 확장 설치를 수행할 것이다. 이를 통해서 본 연구 책임자의 고성능 클러스터 시스템을 외부의 그리드 시스템과 연동을 할 수 있도록 만들 것이다. 또한 이 연동을 할때는 interactive job 실행을 지원할 것이다. 이는 그리드 큐를 사용하여 지원 될 것이다. 뿐만 아니라 본 연구를 수행하기 위해서 root계정이 반드시 필요한 문제에 대해서는 root계정을 지원해 줄 수 있도록 한다. 이 지원은 본 연구에서 구축하고자 하는 그리드 시스템에서 요구하는 사항이므로 반드시 수행시킬 수 있도록 한다.

Globus Toolkit 2.2의 확장 설치를 수행함과 동시에 응용 프로그램들을 작성할 수 있도록 하기 위한 지원들을 수행할 것이다. 응용 프로그램을 작성할 수 있으려면 Globus Toolkit만 필요한 것이 아니라 MPI, GM등의 통신 라이브러리도 필요하며, C, C++, Fortran등의 응용 프로그램 컴파일러 역시 필요하다. 이러한 것들을 모두 지원할 수 있도록 설치하도록 한다. 또한 응용 프로그램 작성에 따라 별도의 필요한 소프트웨어가 있으면 저작권 문제를 침해하지 않는 한 설치를 지원해 주어서 응용프로그램 개발을 지원 하도록 한다.

그리드 시스템은 네트워크로 구성된 시스템이기 때문에 외부의 사용자에게 의한 보안 문제가 대두된다. 이를 해결하기 위해서 방화벽을 설치하고, 허가된 사용자와 허가된 포트 및 IP에 대해서만 그리드 시스템이 동작하도록 구성해야 한다. 또한 제안서에 명기된 시간만큼은 그리드 시스템에서 사용할 수 있도록 자원을 할당한다. 이를 통해서 응용 프로그램의 구축이 가능하도록 가능한 한 지원하도록 한다.

2. 고성능 PC 클러스터

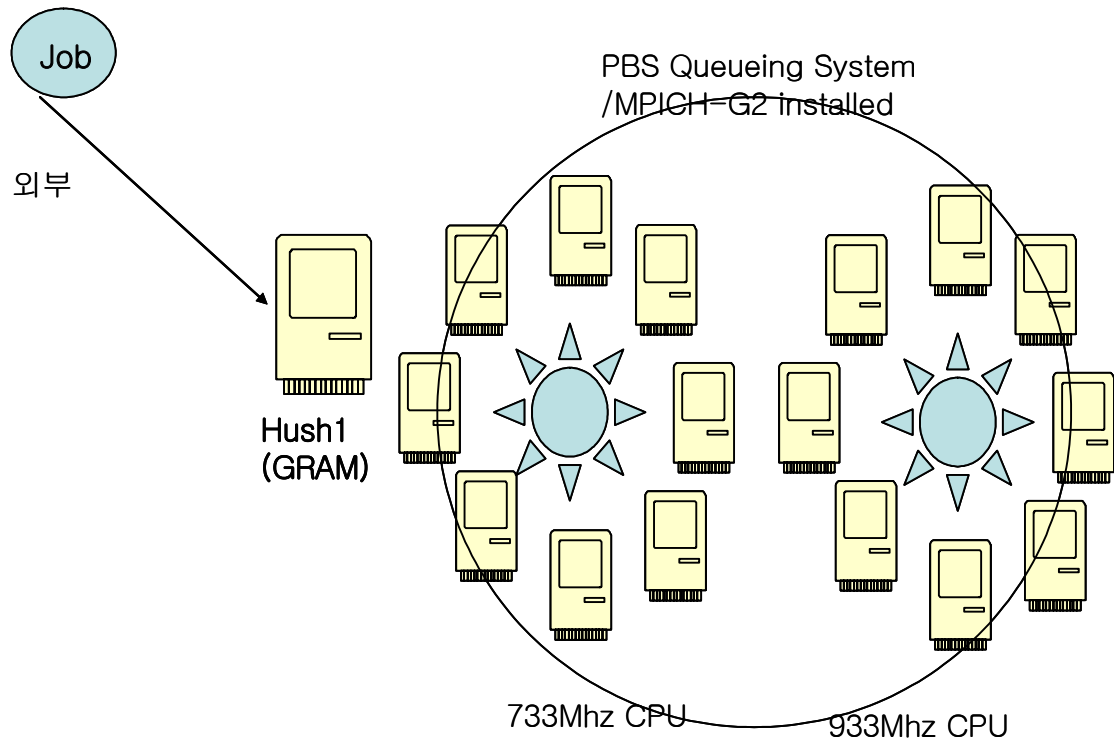


그림 2 제공된 고성능 클러스터 시스템

그림 2는 그동안 연구 과정에서 설치된 고성능 클러스터 환경을 그림으로 나타낸 것이다. 실제 노드 수는 40개이며 각각은 리눅스가 운영체제로 설치되어 있다. 고성능 클러스터 시스템은 글로벌스 작업에 대해서 아직은 미리넷을 이용하고 있지는 않다. 즉 100Mbps 이더넷을 사용하여 모든 실제적인 작업을 하고 있다.

구축된 클러스터 시스템이 외부에서는 단일 시스템 이미지로 보여야 한다. 실제 클러스터 노드들은 40개이지만 마치 하나의 시스템이 존재하는 것처럼 보여야 하기 때문에 전체 시스템에서 대표되는 노드가 하나 할당이 되어야 한다. 우리가 제공하는 클러스터에서는 hush1-hush40의 40개 노드 중에서 hush1이 대표 노드이다. 따라서 모든 외부 자원에서는 클러스터에 작업을 할당하기 위해서 hush1으로 작업을 할당해야 한다.

그것을 위해서 먼저 PBS Batch Queueing System을 설치하였다. 큐잉 시스템은 여러 사용자가 공정하게 자원을 사용할 수 있도록 도와 준다. 일괄 처리 시스템은 사용자가 작업을 제출해 놓고 나중에 결과를 기다리는 방식이다. hush1이 대표 노드이기 때문에 우리는 PBS 서버 노드를 hush1으로 정했다. 나머지 노드들 (hush2-hush40)은 PBS 클라이언트 노드이다. PBS 큐잉 시스템은 웹페이지 <http://www.openpbs.org> 에서 등록만 하면 무료로 다운 받을 수 있었다. PBS를 사

용하지 않고서도 그리드를 구축할 수 있으나 이것은 매우 자원을 효율적으로 사용하지 못하는 단점이 있다. LSF, Condor, NQE와 같은 다른 전용 큐잉 시스템도 있지만 아직은 테스트 단계이고 현재 국내에서 가장 많이 쓰이는 것을 다운 받아 설치하였다.

그 다음에는 글로버스를 설치하였다. 글로버스는 KISTI에서 요구한 버전 2.2.4이고 이것도 역시 마찬가지로 hush1을 중심으로 설치하였다. 큐잉 시스템의 서버와 글로버스의 GRAM(Globus Resource Allocation Manager)가 hush1으로 일치한다. 외부에서는 항상 GRAM으로 작업을 실행 시켜야 하고 GRAM은 hush2-hush40의 머신의 상태를 살펴보고 이용 가능한 노드가 있으면 즉시 작업을 실행시킨다. 이와 같은 기법으로 클러스터를 구성하였기 때문에 외부 사용자는 이 클러스터를 hush1이라는 인터페이스를 통해서 작업을 할당할 수 있다.

다음으로 MPICH-G2 버전 1.2.5를 설치하였다. 이 툴킷은 병렬 작업을 클러스터에서 실행시키기 위한 것으로 글로버스 버전과 같이 연동하도록 만들어진 것이다. 이 툴킷의 설치로 인해서 사용자는 클러스터에 단일 작업이나 병렬 작업을 선택하여 실행할 수 있게 된다.

마지막으로 SCMS라는 클러스터 자원을 실시간으로 모니터링을 할 수 있는 패키지를 설치하였다. SCMS가 있으면 그리드에서 자신의 응용 프로그램이 언제 어떤 자원을 얼마만큼 사용할 수 있는지 빠르게 판단할 수 있다. 이 SCMS를 설치하기 위해서는 Apache등 웹에 관련된 패키지를 먼저 설치해야 한다. 기본적으로 웹을 이용하여 자원의 상태를 모니터링을 하기 때문에 사용자를 그래픽 인터페이스를 사용함으로써 매우 쉽게 그 상황을 판단할 수 있다.

3. 고성능 그리드 자원을 위한 툴킷 설치 과정

가. PBS의 설치

PBS는 소스로 설치할 수도 있고 RPM으로 설치할 수도 있다. 제공된 클러스터 시스템은 레드햇 리눅스 운영체제이므로 간단히 RPM을 이용하여 설치하였다.

그 설치 과정은 다음과 같다.

1. www.openpbs.org에서 PBS 최신 버전을 다운로드 받는다.

2.1 마스터 노드 (hush1에 PBS 서버 설치)

```
# rpm -ivh --nodeps openpbs-2.3pl2-1.i386.rpm
```

```
Preparing ... ##### 100%
```

```
1: openpbs ##### 100%
```


2.2. 환경 변수의 추가

```
# vi ~/.bashrc
...
# PBS
export PATH=/usr/pbs/bin: /usr/pbs/sbin:$PATH
export PBS_SERVER_HOME=/usr/spool/PBS
export PBS_HOME=/usr/spool/PBS
....
:wq
#source ~/.bashrc
```

2.3. 서버설정

```
#cd /usr/spool/PBS
#vi server_priv/nodes
hush2
hush3
hush4
...
..
hush40
```

```
#vi server_name
hush1.postech.ac.kr
:wq
```

2.4. 서버 데몬 시작하기

```
#/usr/pbs/sbin/pbs_server
#/usr/pbs/sbin/pbs_sched
#ps -aux | grep pbs
```

3. 클라이언트 노드에 설치하기 (hush2-hush40에 PBS 설치)

3.1 RPM 패키지 설치

```
# rpm -ivh --nodeps openpbs-exehost-2.3pl2-1.i386.rpm
Preparing ... ##### 100%
```

```
1:openpbs-exehost ##### 100%
```

3.2 클라이언트 설정

```
# cd /usr/spool/PBS
# vi server_name
hush1.postech.ac.kr
:wq
#vi mom_priv/config
$clienthost hush1.postech.ac.kr
:wq
```

3.3 클라이언트 데몬 시작하기

```
#/usr/pbs/sbin/pbs_mom
#ps -aux | grep pbs
```

나. 글로버스의 설치

1. 글로버스를 <http://www.globus.org>에서 다운 받는다.

2. 글로버스의 계정(globus)을 만든다.

```
# /usr/sbin/adduser globus
# mkdir /usr/local/globus
# chown globus:globus /usr/local/globus
```

3. 글로버스의 계정으로 GPT를 설치하기 시작한다. GPT는 글로버스의 패키지 관리 소프트웨어이다.

```
#export GLOBUS_LOCATION=/usr/local/GLOBUS
```

GPT 2.2를 다운 받아서 설치한다.

```
#export GPT_LOCATION=/usr/local/globus
#tar xzvf gpt-2.2.2-src.tar.gz
#cd gpt-2.2.2
#./build_gpt
```

4. 글로버스 2.2.4의 각 번들을 차례대로 설치한다.

```
#/usr/local/globus/sbin/gpt-build globus-data-management-client-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-data-management-sdk2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-data-management-server-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-information-services-client-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-information-services--sdk-2.2.4-src_bundle. tar.gz gcc32dbg
```

```
#/usr/local/globus/sbin/gpt-build globus-information-services-server-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-resource-management-client-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-resource-management-sdk-2.2.4-src_bundle. tar.gz gcc32dbg
#/usr/local/globus/sbin/gpt-build globus-resource-management-server-2.2.4-src_bundle. tar.gz gcc32dbg
```

5. globus사용자에 대한 환경설정을 한다. (여기서는 globus 사용자가 .bashrc 환경을 사용한다고 가정)

```
#vi ~globus/.bashrc
...
export GLOBUS_LOCATION=/usr/local/GLOBUS
export GPT_LOCATION=/usr/local/GLOBUS
. $GLOBUS_LOCATION/etc/globus-user-env.sh
...
#source ~globus/.bashrc
```

6. globus-build한 뒤에 후처리 인스톨을 시작한다.

```
#/usr/local/globus/sbin/gpt-postinstall
```

7. 글로버스가 제대로 설치 되었는지 확인한다.

```
#/usr/local/globus/sbin/gpt-verify
```

8. 글로버스 인증서를 발급한다.

```
#/usr/local/globus/bin/grid-cert-request -cn ksc (사용자 인증서)
#/usr/local/globus/bin/grid-cert-request -service host -host hush1.postech.ac.kr
```

9. CA 인증서를 설치한다.

```
#/usr/local/globus/bin/openssl x509 -hash -in temp.pem -noout
82da68f0.0
#mv temp.pem 82da68f0.0
#vi /etc/grid-security/certificates/82da68f0.signinig_policy
# EACL entry #1
access_id_CA X509 '/C=KR/O=Globus/CN=KISTI Supercomputing Center CA2'
pos_rights globus CA:sign
cond_subjects globus `"/C=KR/O=Globus/*"/O=Grid/O=Globus/*"
```

10. Gatekeeper를 Linux 서비스로 등록을 한다.

```
#vi /etc/services
globus-gatekeeper 2119/tcp    #Globus Gatekeeper
:wq

#vi /etc/xinetd.d/globus-gatekeeper
services globus-gatekeeper
{
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /usr/local/GLOBUS/sbin/globus-gatekeeper
    server_args = -conf/usr/local/GLOBUS/etc/globus-gatekeeper.conf
    disable = no
}
:wq
```

11. xinetd를 재 구동한다.

```
# service xinetd restart
```

12. 사용자를 맵 파일에 추가하기

```
#vi /etc/grid-security/grid-mapfile
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=KSC" ksc
:wq
```

13. 글로버스가 정상 동작하는지 확인을 한다.

```
#grid-proxy-init
#globusrun -a -r hush1.postech.ac.kr
GRAM Authentication test successful이라는 메시지가 나오면 성공이다.
```

14. 추가적인 서비스들을 추가한다. (GRAM Report, Grid FTP, MDS 설정 등)

이들도 차례대로 설치해 주면 된다. 이것들은 부차적인 것들이므로 여기서 설명하지 않았다. 좀더 자세한 정보를 원하면 다음 웹페이지를 참조하면 된다.

다. MPICH-G2의 설치

1. MPICH-G2를 웹페이지 <http://www-unix.mcs.anl.gov/mpi/mpich>에 접속하여 다운로드 받는다.

2. 압축을 풀고 패치가 필요하면 패치를 수행한다.

```
#tar xzvf mpich-g2.tar.gz
```

```
#patch -p0 < patch.all
```

3. configure를 실행한다.

```
#./configure --with-device=globus2:-flavor=gcc32dbg
```

4. make와 make install을 실행한다.

```
#make
```

```
#make install
```

여기까지 수행했을 때 자체 클러스터에서 글로버스가 성공적으로 동작하는지 확인을 할 수 있었다. 테스트 방법은 다음과 같다.

greet.c 라는 예제 MPI 프로그램을 작성하여 글로버스 미들웨어와 연동하여 병렬 작업을 수행하는지 테스트 하였다.

병렬 작업은 먼저 hush1의 GRAM gatekeeper에게 가서 인증을 받고 GRAM에 의해서 pbs-jobmanager를 접촉한다. 그러면 pbs-jobmanager는 MPI 병렬 프로그램이 들어오면 그것을 병렬로 수행을 하게 된다.

그림 3은 greet.c라는 프로그램이다.

먼저 이 프로그램을 실행하기 전에 machines라는 파일이 로컬에 있어야 한다.

```
#vi machines
```

```
"hush1.postech.ac.kr" 10
```

```
"hush1.postech.ac.kr" 10
```

```
"hush1.postech.ac.kr" 10
```

```
"hush1.postech.ac.kr" 10
```

:wq

여기서 "hush1.postech.ac.kr" 40이라고 써 주지 않고 "hush1.postch.ac.kr" 10을 4번 써 주는 이유는 MPICH-G2의 버그 때문이다. MPICH-G2는 한번에 최대 12개까지 프로세스 생성을 허용한다. 그러나 위의처럼 써 주면 마치 40개의 프로세스를 허용하는 것처럼 되기 때문에 별 문제는 없다.

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char **argv) {
    int my_rank;
    int p;
    int source;
    int dest;
    int tag = 50;
    char message[100];
    MPI_Status status;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    MPI_Comm_size(MPI_COMM_WORLD, &p);

    if (my_rank != 0) {
        sprintf(message, "Greetings from process %d!",
            my_rank);
        dest = 0;
        MPI_Send(message, strlen(message)+1, MPI_CHAR, dest, tag,
            MPI_COMM_WORLD);
    }
    else {
        for (source = 1; source < p; source++) {
            MPI_Recv(message, 100, MPI_CHAR, source, tag,
                MPI_COMM_WORLD, &status);
            printf("%s\n", message);
        }
    }
    MPI_Finalize();
}
```

그림 3 greet.c 프로그램

이 프로그램을 다음과 같이 실행 하였다.

```
# mpicc greet.c -o greet
# mpirun -dumprsl -np 12 greet > greet.rsl
# vi greet.rsl
+
( &(resourceManagerContact="hush1.postech.ac.kr/jobmanager-pbs")
(count=12)
(label="subjob 0")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)
(LD_LIBRARY_PATH /usr/local/GLOBUS/lib/))
```

```
(directory="/home/users/ksc/greet")
(executable="/home/users/ksc/greet/greet")
)
```

이 스크립트가 자동적으로 생성이 된다.

이 스크립트에서 jobmanager-pbs라고 써 준 것은 PBS jobmanager로 사용하기 위해서이다. 만일 PBS를 이용하지 않으면 default로 jobmanager-fork를 사용하게 되며 이 때에는 40대의 노드를 이용하여 병렬로 작업을 수행하는 것이 아니라 한 노드, 즉 hush1에서 40개의 프로세스를 fork하여 프로그램을 수행하게 된다. 따라서 이런 경우를 방지하기 위해서 반드시 pbs를 default로 지정을 해 주거나 스크립트를 위와 같이 고쳐 주어야 한다.

다음과 같이 MPICH-G2를 실행한다.

```
#mpirun -globusrl greet.rsl
```

```
Greetings from process 1!
```

```
Greetings from process 2!
```

```
Greetings from process 3!
```

```
Greetings from process 4!
```

```
Greetings from process 5!
```

```
Greetings from process 6!
```

```
Greetings from process 7!
```

```
Greetings from process 8!
```

```
Greetings from process 9!
```

```
Greetings from process 10!
```

```
Greetings from process 11!
```

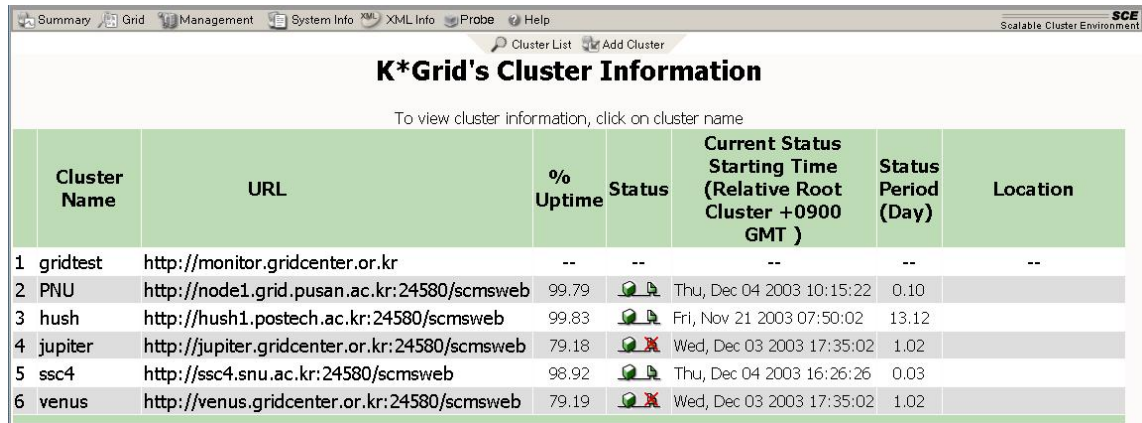
이로서 글로버스 설치와 MPICH-G2, PBS 까지 제대로 설치되었다는 것을 알 수 있다.

라. SCMSWEB의 설치

SCMSWEB은 자원의 활용도를 실시간으로 또는 통계적으로 보기위한 모니터링 툴로 SCMSWEB 2.3버전이 HUSH 클러스터에 설치되어 있다. 이전에 2.1버전을 설치했을 때 발생했던 에러는 2.3버전에서는 나타나지 않고 있다 설치하는 SCMSWEB의 홈페이지인

<http://hpcnc.cpe.ku.ac.th/moin/InstallingSCMSWeb>에 접속하여 모니터링 툴 및 설치 가이드를 다운받아 설치 할 수 있다.

SCMSWEB 2.3버전은 2.1 버전과 큰 차이는 없고 웹 인터페이스가 조금 틀리며 2.1버전에 비해 더 안정적이다.



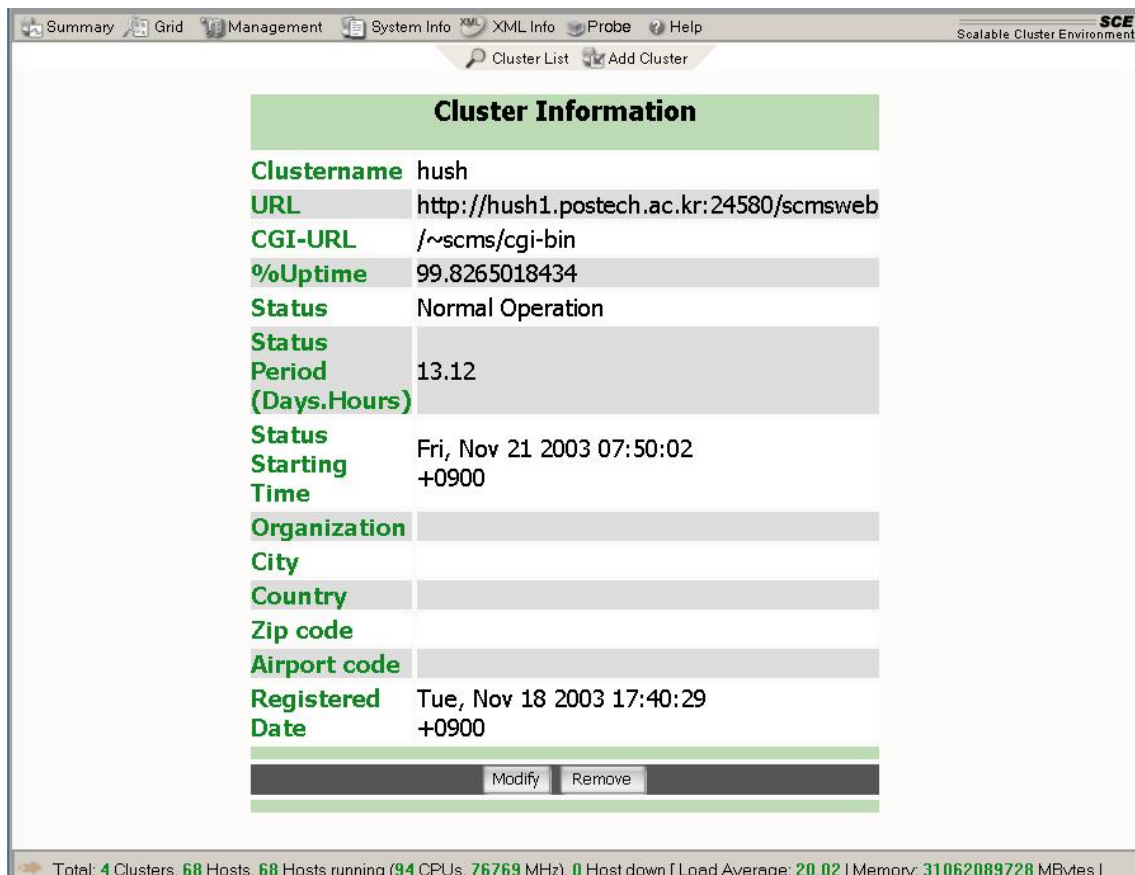
K*Grid's Cluster Information

To view cluster information, click on cluster name

	Cluster Name	URL	% Uptime	Status	Current Status Starting Time (Relative Root Cluster +0900 GMT)	Status Period (Day)	Location
1	gridtest	http://monitor.gridcenter.or.kr	--	--	--	--	--
2	PNU	http://node1.grid.pusan.ac.kr:24580/scmsweb	99.79		Thu, Dec 04 2003 10:15:22	0.10	
3	hush	http://hush1.postech.ac.kr:24580/scmsweb	99.83		Fri, Nov 21 2003 07:50:02	13.12	
4	jupiter	http://jupiter.gridcenter.or.kr:24580/scmsweb	79.18		Wed, Dec 03 2003 17:35:02	1.02	
5	ssc4	http://ssc4.snu.ac.kr:24580/scmsweb	98.92		Thu, Dec 04 2003 16:26:26	0.03	
6	venus	http://venus.gridcenter.or.kr:24580/scmsweb	79.19		Wed, Dec 03 2003 17:35:02	1.02	

그림 4. POSTECH 클러스터가 K*그리드 웹 페이지에 등록된 모습

그림 4. 는 K*그리드에 등록된 클러스터에 대한 정보를 보여주는 그림으로 3번 hush클러스터가 포항공대에서 제공하고 있는 클러스터이다. 2.1버전에서 나타난 문제점은 나타나지 않고 있다.



Cluster Information

Clustername hush

URL http://hush1.postech.ac.kr:24580/scmsweb

CGI-URL /~scms/cgi-bin

%Uptime 99.8265018434

Status Normal Operation

Status Period (Days.Hours) 13.12

Status Starting Time Fri, Nov 21 2003 07:50:02 +0900

Organization

City

Country

Zip code

Airport code

Registered Date Tue, Nov 18 2003 17:40:29 +0900

Total: 4 Clusters. 68 Hosts. 68 Hosts running (94 CPUs. 76769 MHz). 0 Host down (Load Average: 20.02 | Memory: 31062089728 MBytes |

그림 5. POSTECH 클러스터 정보 화면

SCMSWEB 모니터링 설치 후 K*그리드에 등록을 한 후에는 주기적으로 hush 클러스터의 자원 활용 상황이 K*그리드 모니터링 툴로 보고되어져 K*그리드 사이트에서 종합적으로 모니터링 할 수 있으며 hush 클러스터에 직접 접속하여 모니터링 할 수도 있다. 아래의 그림들은 모니터링 상황을 나타낸 그림들이다.

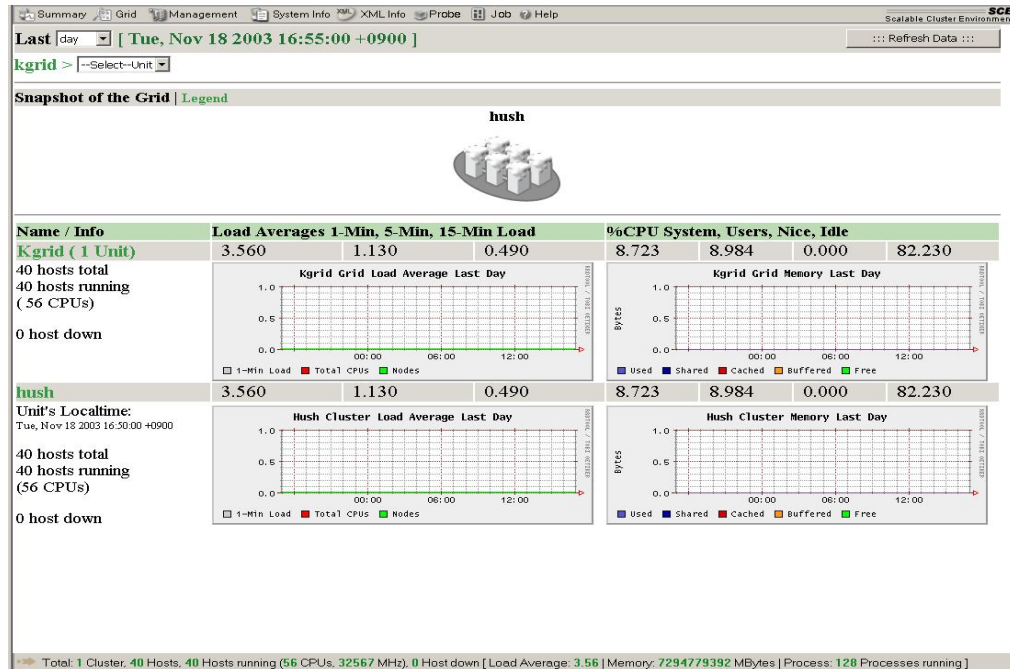


그림 6. POSTECH 클러스터 모니터링 페이지 화면

메뉴에서 Grid를 선택하면 클러스터의 자세한 정보가 표시된다. 클러스터의 각 노드별로 CPU 사용, MEMORY 사용, 부하, 실행되고 있는 프로세스등의 정보를 실시간으로 모니터링 할 수 있으며 지난 시간의 정보도 그래프를 통해 표시된다.

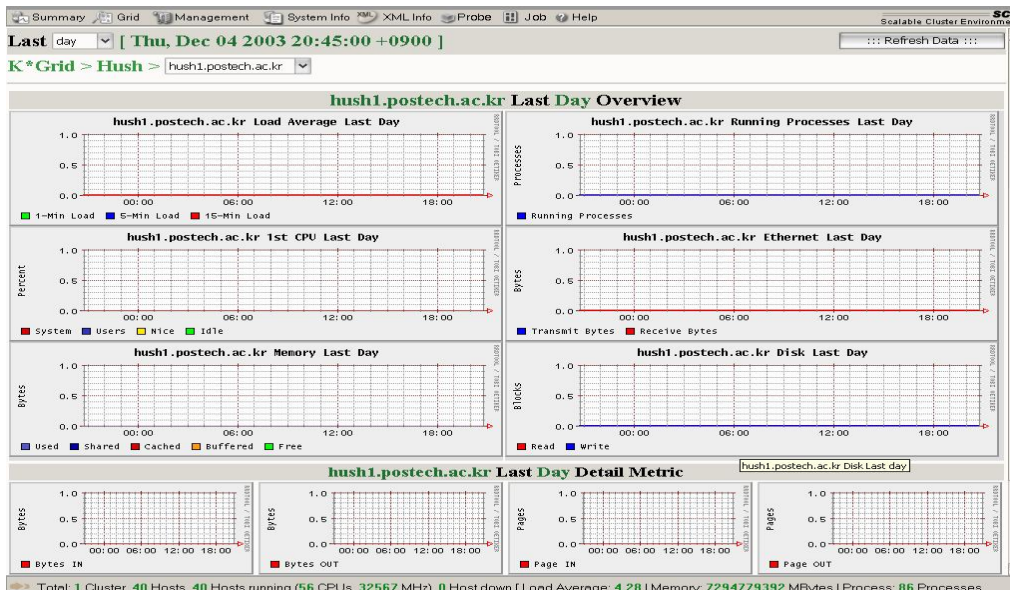


그림 7. POSTECH 클러스터 내의 hush1노드에 대한 자원 활용 현황

마. 사용자 환경 설정

다른 사람들이 그리드를 사용하기 위해서는 보안 및 계정에 관련해서 여러 가지 셋팅을 해 주어야 한다. 다음은 현재까지 등록된 그리드 맵 파일을 보여 준다.

```
#vi /etc/grid-security/grid-mapfile

"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hush1" globus
# For Kisti
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=kisti" kisti
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=globus" kisti
"/O=Grid/O=Globus/OU=kumho.co.kr/CN=Globus Manager at Kumho" kisti
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Gee-Bum Koo" globus
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Jae Hyuck Kwak" globus
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=giljael" globus
# For ANT Testbed
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hush1" globus
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=kisti" kisti
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=globus" kisti
"/C=KR/O=Globus/OU=Yonsei/CN=KyungLang Park" lanx
"/O=Grid/O=Globus/OU=ajou.ac.kr/CN=Park Sangmin" sm
"/C=KR/O=Globus/OU=Yonsei/CN=leehwangjik" bear22
"/C=KR/O=Globus/OU=KUT/CN=bear22" bear22
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=KSC" ksc
"/O=Grid/O=Globus/OU=chonbuk.ac.kr/CN=KimKyongSu" boarder
"/O=Grid/O=Globus/OU=nclab.yu.ac.kr/CN=srman" srman
"/O=Grid/O=Globus/OU=cherry.icu.ac.kr/CN=lake" lake
"/O=Grid/O=Globus/OU=me.pusan.ac.kr/CN=sdhong" hsd
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Gee-Bum Koo" voxel
# For researcher
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=Yoon Myung Lee" sori72
# For HPC member
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hpc1" tcruise
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hpc2" tcruise
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hpc3" tcruise
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=jackclee-globus" tcruise
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=hush1-tcruise" tcruise

#
# Testbed Administrators
#
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Jae Hyuck Kwak" jhkwak
"/O=Grid/O=Globus/OU=hpcnet.ne.kr/CN=Jae Hyuck Kwak" jhkwak
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Gee-Bum Koo" voxel
"/O=Grid/O=Globus/OU=hpcnet.ne.kr/CN=giljael" giljael
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=globus" giljael
"/C=KR/O=Grid/OU=hpcnet.ne.kr/OU=hpcnet.ne.kr/CN=Gee-Bum Koo" voxel
```

```

#
# KISTI users
#
"/O=Grid/O=Globus/OU=hpcnet.ne.kr/CN=HongSuk Yi at KISTI" kis001
"/O=Grid/O=Globus/OU=cnu.ac.kr/CN=globuslinux" kis002
"/O=Grid/O=Globus/OU=hpcnet.ne.kr/CN=KumWon Cho" kis003
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=Sangwan Kim" kis004
"/O=Grid/O=Globus/OU=hpcnet.ne.kr/CN=Sang Min Lee" kis005

#
# N*Grid users
#
"/O=Grid/O=Globus/OU=postech.ac.kr/CN=KSC" ngrid001
"/O=Grid/O=Globus/OU=snu.ac.kr/CN=floydfan" ngrid002
"/O=Grid/O=Globus/OU=kaist.ac.kr/CN=kaistCFD" ngrid003
"/O=Grid/O=Globus/OU=gridcenter.or.kr/CN=cfdkwon" ngrid003
"/O=Grid/O=Globus/OU=me.pusan.ac.kr/CN=sdhong" ngrid004
"/O=Grid/O=Globus/OU=sookmyung.ac.kr/CN=grid1" ngrid005
"/O=Grid/O=Globus/OU=grid.pusan.ac.kr/CN=gridnode1" ngrid006
"/O=Grid/O=Globus/OU=SNU/CN=dcslab" ngrid007
"/O=Grid/O=Globus/OU=kaist.ac.kr/CN=globus_at_wwf" ngrid008
"/O=Grid/OU=GlobusTest/OU=simpleCA-dss3.grid.or.kr/OU=grid.or.kr/CN=na
notech" ngrid009
"/O=Grid/O=Globus/OU=mju.ac.kr/CN=globus" ngrid010
"/O=Grid/O=Globus/OU=konkuk.ac.kr/CN=konkukimc" ngrid011

```

그리고 각 계정에 관계되는 사람들을 위해서 각 노드를 NFS로 hush1에서 hush40까지 /home/users 디렉토리를 생성시켰다. 제공하는 클러스터에는 자동적으로 모든 노드에 계정을 생성하는 스크립트를 가지고 있으므로 이런 작업들은 쉽게 해결할 수 있다.

또한 보안을 위해서 TCP Wrapper를 사용하고 있는데 /etc/hosts.deny 와 /etc/hosts.allow 파일을 이용하여 접속 가능한 머신 IP 리스트와 그렇지 않은 머신들을 구분하여 철저히 보안에 신경을 쓰고 있다.

```

#vi /etc/hosts.deny
all:all

#vi /etc/hosts.allow
all:141.223.165.131
all:141.223.165.132
all:141.223.165.133
all:141.223.165.134
.....
.....
all:164.125.131.76

```

III. 결론

우리는 컴퓨팅 그리드 자원에 일부로 사용될 고성능 클러스터 자원을 구축하였다. 글로버스 2.2.4와 PBS, MPICH-G2를 성공적으로 설치하여 실제로 GRAM 그리드 연동이 가능함을 이 보고서에서 보여 주었다. 실제로 이 시점에서 KISTI에서 포항공대의 자원에 대해서 그리드 연동이 성공하였다고 보고 하였다. 그러나 그 뒤에 KISTI측에서 요구한 SCMSWEB모니터링 툴이 제대로 설치가 되지않아 문제가 있었다. 11월 중순에는 이를 해결하여 현재는 설치를 완료하였다.

그리드 관련 글로버스 프로그램 및 모니터링 툴을 설치하면서 느낀 것은 글로버스 뿐만 아니라 그것과 관계되는 여러 가지 패키지의 상관관계에 대해서 매우 많은 전문지식이 필요하다는 것이다. 그것은 단순히 그 문서만 가지고 알 수 있는 사항은 아니다. 이것은 컴퓨터 전반에 걸쳐서 공부를 해야 알 수 있는 것인데, 이러한 것은 실제 경험자들에 의해서 많이 배울 수 있었다. 그리고 이 보고서에서 제안했듯이 패키지들을 설치하면서 좋은 정보들은 사람들끼리 많이 공유를 해야 한다고 생각한다. 그리드 관련 소프트웨어들은 그리드의 특성상 다양한 아키텍처의 시스템에 설치되어야 하는 것들이라 그 정보들의 공유는 그리드 사용을 더 가속시키는데 유용하게 이용될 것으로 보인다.

각 프로그램을 설치하고 모니터링을 하면서 실제로 클러스터가 그리드 상에서 사용되고 있는 것은 확인을 할 수 있었으나 어떤 프로그램이 실행되고 있는지는 알 수가 없었다. 이 점을 보완하여 현재 클러스터에서 어떤 프로그램이 어느 정도의 자원을 사용하고 있는가를 알 수 있는 시스템이 구축되면 더 효율적인 자원 관리를 할 수 있을 것이라고 본다. 이에 대한 연구를 진행하는 것도 테스트베드를 위해서 도움이 될 것이다.

이번 자원 제공 관련 프로젝트를 통해 그리드를 운영할 때 생기는 많은 실제적인 문제점 및 그 해결방법을 얻을 수 있었다. 이는 그리드를 운영할 때 도움을 줄 수 있는 자료가 될 것이다. 이러한 의미에서 자원 제공 관련 프로젝트는 그 의미가 작지 않은 것으로 본다.

올해의 경우 KISTI와 자원제공 기관간의 연락 및 업무가 체계적으로 잡히지 않아 서로 간에 업무진행이 원활하지 않았다. 아직 초기단계라 문제점도 많을 수밖에 없고 이러한 문제는 앞으로 시간이 갈수록 나아지리라 본다.

참고문헌

- [1] <http://www.globus.org>
- [2] <http://gridtest.gridcenter.or.kr>
- [3] Malcolm Atkinson, "Rationale for Developing with the Open Grid Services Architecture"
- [4] Jim Basney and Miron Livny, "Deploying a High Throughput Computing Cluster", High Performance Cluster Computing, Rajkumar Buyya, Editor, Vol. 1, Chapter 5, Prentice Hall PTR, May 1999.
- [5] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.
- [6] Abramson D., Sosic R., Giddy J. and Hall B., "Nimrod: A Tool for Performing Parametised Simulations using Distributed Workstations", The 4th IEEE Symposium on High Performance Distributed Computing, Virginia, August 1995
- [7] Nirav H. Kapadia, Jose' A. B. Fortes, Mark S. Lundstrom, and Dolors Royo., PUNCH: A Computing Portal for the Virtual University, International Journal of Engineering Education (IJEE). In special issue on Virtual Universities and Engineering Education. Vol. 17, No. 2.

[연구보고서 작성지침]

가. 보고서의 인쇄 규격

- 크기 : A4 (210x297)

나. 작성 소프트웨어

- MS-Word나 한글로 작성한다.

다. 인쇄 방법

- (1) 표지 : 바탕 백색, 활자 흑색
- (2) 내용 : 흑색 지정활자로 인쇄한다.
- (3) 제본 : 좌철