인터넷 수퍼컴퓨팅 기술의 구현

Realization of Internet Supercomputing Technology

김승조[1)]

In this work, Internet Supercomputing methodology is introduced and the concept is materialized for large-scale finite element analysis. The primary resources of Internet Supercomputing are numerous idling PCs connected by Internet with no regards to their locations. Therefore, it becomes one of the most affordable ways to achieve supercomputing power unlimitedly if the appropriate parallel algorithm and the operating program are developed for this slow network environment. Under the above concept, virtual supercomputing system InterSup I is constructed and tested. To establish the InterSup I system, 64 CPU nodes, which are located in several places and connected by Internet, are conscripted, and parallel finite element software is developed for linear static analysis of structures based on the parallel multi-frontal algorithm. By the established InterSup I system, analysis of finite element structural model having around five million DOFs are solved to check the affordability and effectiveness of Internet Supercomputing.

1. 서 론

수퍼컴퓨터는 당대에 사용할 수 있는 컴퓨터중에서 제일 빠르고 용량이 큰 컴퓨터를 일컫는다. 최근까지는 CRAY XMP, YMP, CRAY-2 C90 등이 수퍼컴퓨터 기능을 수행했는데, 벡터 프로세싱 기법을 사용하여 고속으로 과학 계산에 필요한 계산을 수행할 수 있게 한 것이다. 따라서 완전하게 '벡터화'가 이루어진 프로그램들에 대해서는 수퍼컴퓨터라는 이름에 걸맞는 성능을 낼 수 있었다. 그러나 벡터형은 개발원가가 높아 사용자의 비용부담이 크고, 관리비용도 큰 시스템의 경우 연간 수억원대에 달하기 때문에 새로운 성능향상을 이룬 벡터컴퓨터의 출현이 점점 힘들어지고 있다. 그래서 최근에는 벡터형이 안고 있는 비용부담의 대안으로 90년대 초반 최소 4분의1 이상 저렴한 MPP(Massively Parallel Processing) 시스템이, 96년경부터 10분의 1 이상 저렴한 SMP(Symmetric Multi-Processors) 시스템이 수퍼컴퓨터 대체재로 부각 받기 시작했다. 또한 SMP 구조를 기반으로 하면서 MPP 방식을 응용한 NUMA(Non-Uniform Memory Access)형도 부각되기 시작했다. MPP, 그리고 특히 SMP와 NUMA의 등장으로 수퍼컴퓨터의 혜택은 선택된 몇몇 연구원만의 것이 아니라 보편적인 사용자들에게 돌아가게 되었으며 수퍼컴퓨터보다는 'HPC(High Performance Computing) 서버' 라는 용어가 훨씬 많이 쓰이는 것도 이것 때문이다. 그리고 또 한가지 중요한 사실은 SMP 형 컴퓨터와 MPP 형 컴퓨터 등의 등장과 함께, 변화된 컴퓨터 하드웨어에 적합한 프로그램 개발방식, 즉 "분산병렬처리"기법의 중요성이 부각되기 시작한 것이다. 지금까지 대부분의 프로그래머들이 사용하던 순차적 방식의 프로그램으로는 변화된 SMP 형과 MPP 형 컴퓨터 등이 대다수를 차지하게 된 수퍼컴퓨팅 환경에서는 뛰어난 성능을 낸다는 것 자체가 불가능해진 것이다. 그런데, 이러한 새로운 방식의 HPC 서버들이 과학기술 분야의 연구원들에게 상당한 수준의 계산을 예전에 비해서 손쉽게 처리할 수 있게 해주기는 했지만, 우주공간에서의 항공우주비행체의 가상 시뮬레이션이나, 수백만 자유도 이상의 항공기 동체 전체 구조모델에 대한 고정밀 안정성 해석과 같은 계산을 수행하기 위한 비용은 여전히 엄청나게 높은 것이 현실이다. 그래서 최근에 새롭게 등장한 수퍼컴퓨팅파워 구현기술이 "초고속 네트워크를 이용한

1) 서울대학교 항공우주공학과

수퍼컴퓨팅" 기술이다. 이 기술을 적용하여 추진했던 대표적인 예를 살펴보면, 미국 에너지성의 Accelerated Strategic Computing Initiative(ASCI) 프로그램을 들 수 있다. ASCI Red 수퍼컴퓨터는 모두 9632개의 Pentium II Xeon 프로세서를 초고속 네트워크을 통해서 서로 연결시켜 만든, 초당 3.6 Teraflops의 연산속도를 갖는 수퍼컴퓨팅 시스템이다. 미국은 ASCI Red 수퍼컴퓨터를 사용하여 여러 가지의 시뮬레이션 기법을 개발하였다. 그러나, ASCI Red 수퍼컴퓨터 역시 여전히 엄청난 개발 및 제작비용을 요구하는 전용시스템이기 때문에 우리 나라와 같은 환경에서 그대로 수용하기에는 적합하지 않을 뿐만 아니라, 초고속 네트워크를 이용한 수퍼컴퓨터에서 원하는 성능을 얻기 위해서는 효율적인 분산병렬처리 알고리듬의 개발이 선행되지 않고서는 불가능하다. 이러한 연구동향 및 수퍼컴퓨팅 하드웨어 변화 양상을 고려할 때 하나의 고성능, 대용량의 수퍼컴퓨터를 이용하기보다는 분산병렬처리기법을 활용하여 여러 대의 컴퓨터들을 동시에 활용하여 문제를 해결하려는 시도는 이미 활발하게 전개되고 있다는 것을 알 수 있다.

본 연구에서는 새로운 개념의 수퍼컴퓨팅 파워 구현방법으로서 "인터넷 수퍼컴퓨팅 기술"을 제안하고자 한다. 최근에 개인용 컴퓨터(PC, Personal Computer) 성능의 급속한 발전과 인터넷의 대중화가 급속도로 진전되는 환경의 변화를 고려하여, 인터넷에 연결되어 있는 기존의 개인용 PC들을 활용한 새로운 형태의 네트워크 수퍼컴퓨팅기술을 구현하였다. 즉, 인터넷에 연결되어 있는 PC들을 소프트웨어적으로 연결하여 전용 시스템이 아닌 가상의 수퍼컴퓨터 시스템을 직접 구현하여 특징 및 장단점을 살펴보고, 그 활용방안을 검토해 보았다.

2. 본 론

2.1 인터넷 수퍼컴퓨팅의 개념

인터넷 수퍼컴퓨팅 기술이라는 것은 기존의 초고속 네트워크를 이용한 분산병렬처리 기술과는 다르게, 인터넷으로 연결되어 있는 개인용 컴퓨터(PC, Personal Computer)들을 소프트웨어적으로 서로 연결해서, 인터넷상에 존재하는 가상적인 수퍼컴퓨터를 구현하여, 이를 통해서 수퍼컴퓨팅을 수행하는 기술을 말한다. 기존의 네트워크를 이용한 수퍼컴퓨팅 기술과 비교하면 하드웨어적으로 우선 가장 큰 차이점은 인터넷을 사용한다는 점과 PC를 사용한다는 점이다. 그래서 앞에서 언급한 ASCI Red 수퍼컴퓨터처럼 각각의 컴퓨터들을 연결하는 별도의 초고속 네트워크가 필요 없게 되고, 기존의 개인용 PC를 사용함으로써 획기적으로 계산비용을 낮출 수 있게 된 것이다. 그리고 더욱 좋은 점은 기존의 수퍼컴퓨터는 초기 구입가격뿐만이 아니라, 시스템 유지보수비도 필요하지만, 인터넷 수퍼컴퓨팅에 동원되는 PC들은 별도의 시스템 유지보수비가 필요 없다는 점이다. 이미 각각의 유저가 자기의 업무를 수행하기 위해서라도 유지보수를 하기 때문이다.

그리고 최근에 많이 소개되고 있는 PC 클러스터링 시스템과 비교한다면, PC를 이용한다는 점은 같지만, 시스템의 사용과 운영상에 근본적인 차이가 있다. 클러스터링 시스템의 경우, 과학기술계산용이나, 웹서비스 등과 같이 특정목적을 위해서 설계되고 개발되어진다. 그래서 시스템을 구축하기 위해서는 별도의 PC를 구입해야만 한다. 그리고 지속적으로 클러스터링 시스템을 이용할 연구계획이나 시스템의 운영계획이 없으면, 상당한 시간 동안을 그냥 전원만 켜 놓은 채, 아무 일도 수행하지 않고 있게 된다. 또한 클러스터링 시스템을 구성하는 개별 컴퓨터 각각은 독자적인 아무런 의미도 가지고 있지 않다. 단지 전체 클러스터링 시스템의 한 부분으로만 존재할 뿐이다. 하지만, 인터넷 수퍼컴퓨팅의 경우는 이미 사무용이나, 웹서버 등의 특정목적을 위해 존재하고 있는 PC들을 활용하여, 필요할 경우에 일시적으로 필요한 수만큼의 PC를 하나의 가상 수퍼컴퓨터로 구현하기 때문에, 계산을 수행하지 않을 때에는 각각의 PC들은 원래의 목적으로 활용될 수 있다. 그리고 클

러스터링의 경우에 아무리 분산병렬 환경으로 시스템을 구현한다고 해도, 노드 하나를 교체하거나, 노드를 증설해야 할 경우에는 상당한 어려움이 있다. 컴퓨터를 더 구입해야 하고, 필요에 따라서는 전체 시스템의 설정을 다시 해야할 경우도 많다. 하지만 인터넷 수퍼컴퓨팅에서는 이러한 확장성이나 노드 교체가 상당히 간단하다. 실제로 시스템을 구현하는 것이 아니라, 소프트웨어적으로 가상적으로 구현하기 때문에 언제든지 특정 컴퓨터를 다른 컴퓨터로 교체할 수 있고, 전체 컴퓨터 수도 언제든지 손쉽게 변경할 수 있다. 그리고 역시 클러스터링 시스템에서도 별도의 고속 네트워크를 필요로 한다. 그리고 일반적으로 클러스터링의 경우, 각각의 노드들의 하드웨어 사양이 모두 같은 것이 일반적이지만, 인터넷 수퍼컴퓨팅의 경우에는 각각의 컴퓨터들의 하드웨어 사양은 일정하지 않다.

지금까지 설명한 것처럼, 인터넷에 연결된 기존의 PC들을 활용하기 때문에 인터넷 수퍼컴퓨팅기술이 기존의 방법들에 비해서 많은 장점을 가지게 되지만, 반드시 장점만이 있는 것은 아니다. 우선, 가장 큰 단점은 역시 별도의 고속 네트워크를 구축하지 않고, 인터넷을 통해서 컴퓨터들간의 통신이 이루어지므로 전송속도가 기존의 수퍼컴퓨팅 전용 HPC 서버나 클러스터링 시스템에 비해서 상당히 느리다는 것이다. 그래서 인터넷 수퍼컴퓨팅 기술을 활용하여 분산병렬처리 프로그램을 개발하려고 할 경우에는 컴퓨터들간의 통신을 최대한으로 줄이는 방향으로 프로그램을 작성해야만 할 것이다. 이러한 이유로 인해서 실제 풀고자 하는 문제를 가장 잘 파악하고 있는 것이 중요하다. 바꾸어서 이야기하면 인터넷 수퍼컴퓨팅의 경우, 상대적으로 느린 시스템의 통신속도에서의 발생하는 단점을 효율적인 분산병렬 알고리듬의 개발로 극복해야 하는 어려움이 있는 것이다. 그리고 앞에서 언급한 것처럼, 가상 수퍼컴퓨터를 구성하는 컴퓨터들의 하드웨어 사양이 일정하지 않기 때문에, 분산병렬처리에서 가장 중요한 작업량 분배(Load Balancing)을 할 때 이러한 하드웨어 다양성을 고려해야 한다.

여기서 잠깐 많은 사람들이 인터넷 수퍼컴퓨팅 기술과 비슷하다고 오해하고 있는 SETI(Search for Extraterrestrial Intelligence) 프로그램[1]과의 차이점을 알아보자. SETI Program은 광할한 우주로부터 쏟아져 오는 여러 가지 정보들 중에서 외계의 지적 생명체로부터 보내진 의미 있는 신호를 찾아내는 프로젝트이다. 최근에는 인터넷을 통해 자원자들을 모집해서, 각자의 컴퓨터에 프로그램을 다운로드 받아서 설치하면 사용자가 컴퓨터를 사용하지 않을 때 그 프로그램이 실행되어 계산을 수행하게 방식을 활용하고 있는데, SETI Program과 인터넷 수퍼컴퓨팅 기술은 근본적으로 다르다. 즉, SETI 프로그램은 특정패턴의 자료를 서로 영역을 나누어서 찾는 일이므로 병렬작업을 수행할 때 컴퓨터들끼리의 통신이 거의 필요없는 반면, 작업에 참여하는 모든 컴퓨터들이 정해진 패턴에 의해 필요한 시점에서 서로들간의 데이타를 주고받으면서 동시에 마치 하나의 컴퓨터처럼 작업을 수행해야 하는 인터넷 수퍼컴퓨팅과는 근본적으로 다른 것이다.

2.2 가상 수퍼컴퓨팅 시스템의 작동원리

인터넷 수퍼컴퓨팅의 구체적인 작동방식과 구현형태에 대해서 살펴보도록 하자. 제일 먼저 수행해야 할 일은 가상 수퍼컴퓨터를 구성하게 될 네트워크에 연결되어 있는 컴퓨터들을 검색하여 각 컴퓨터들의 정보를 검색하는 일이다. 각각의 컴퓨터들의 정보란, CPU, 메인 메모리와 하드디스크의 크기 및 현재 수행되어 지고 있는 프로세서들을 포함한다. 현재, 본 연구를 통해서는 웹을 통해서 컴퓨터에 접속해서 원하는 정보를 획득할 수 있도록 하였다. 그리고 나면 취합된 정보를 바탕으로 가상 수퍼컴퓨터를 구성할 컴퓨터들의 리스트들을 작성할 수 있게 된다. 그런 다음에는 이들 컴퓨터들을 마치 하나의 컴퓨터처럼 움직일 수 있도록 관리하고 제어하고, 컴퓨터들간의 데이타 교환(Message Passing)을 가능하게 해주는 프로그램을 각각의 컴퓨터에 원격으로 접속해서 실행시키면, 일단 인터넷 상의 가상 수퍼컴퓨터 시스템은 구현된 것이다. 물론 이 때 각각의 머신에 일반 유저로서 접속할 수 있는 권한을 가지고 있어야 하지만, 유닉스 시스템의 루트(root)와 같은

특별한 권한을 필요로 해서는 안 된다. 현재, 개발된 시스템은 리눅스를 기반으로 개발되었으며, 첫 번째 단계에서 작성한 컴퓨터 리스트를 활용하여 각각의 컴퓨터들에 접속해서 데몬 형태의 프로세서를 Remote Shell을 통해서 실행시킬 수 있도록 되어 있다. 그 다음 단계는 MPI(Message Passing Interface)를 이용해서 작성한 ipSAP(Internet Parallel Structural Analysis Program)과 같은 응용문제 해석 프로그램을 실행시키는 단계이다.

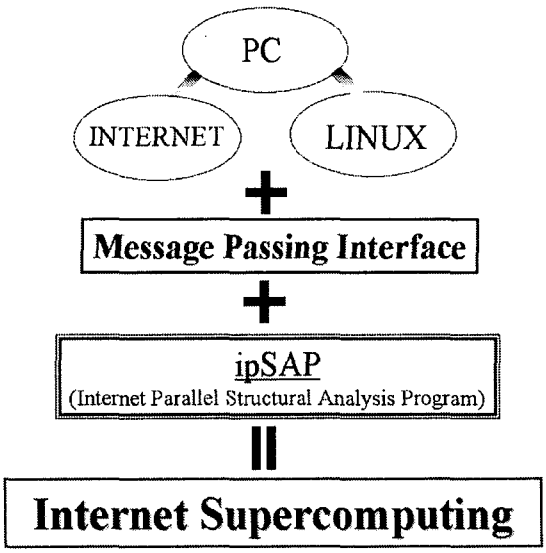


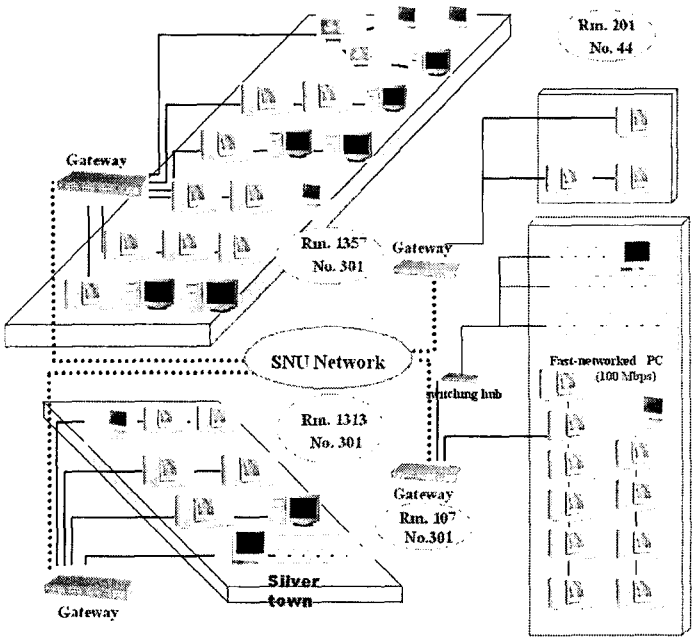
Fig. 1 인터넷 수퍼컴퓨팅의 작동원리

이상과 같은 인터넷 수퍼컴퓨팅의 작동원리 및 단계를 Fig. 1에 도식화하여 나타내었다. 주의할 점은 실행화일은 각각의 시스템에 적합하게 해당 시스템에서 컴파일되어 있어야 하며, 응용프로그램을 실행시키기 이전에 전송해 주어야 한다는 점이다. 이러한 점도 클러스터 시스템과 다른 점이기도 하다. 이상과 같은 크게 3가지 단계를 거쳐서 가상 수퍼컴퓨터 시스템이 구성되고, 구성된 가상 수퍼컴퓨터 시스템을 이용하여 인터넷 수퍼컴퓨팅 기술은 구현되는 것이다.

2.3 InterSup I의 구현

본 연구에서는 2개의 건물에 서로 산재해 있고 인터넷으로 연결되어 있는 PC들을 이용하여 인터넷 수퍼컴퓨팅을 수행할 수 있는 InterSup I을 구현하였다. 우선 보다 안정한 시스템을 구성하기 위한 각종 실험 및 테스트를 수행하였고, 인터넷 수퍼컴퓨팅에 가장 적합하다고 판단되는 OS(Operating System)인 리눅스(Linux) 시스템에 대한 조사 및 기초연구도 함께 수행되었다.

먼저 본 연구에서는 여러 가지 다양한 OS에 대해서 시스템의 성능검사 수행을 수행한 결과, 뛰어난 다중 처리 기능과 네트워크 기능 및 안정성을 지닌 리눅스 시스템을 인터넷 수퍼컴퓨팅을 위한 가장 최적의 OS로 채택하였다. 그리고 병렬처리 시스템 구현에 있어서 필수적인 소프트웨어인 MPI(Message Passing Interface) 구현 소프트웨어들에 대한 기초적인 성능 테스트 및 안정성에 대한 기초 조사를 수행하였다. 본 연구에서는 유닉스 시스템을 기반으로 개발되었고, 리눅스 시스템도 지원하는 MPICH[2]와 LAM(Local Area Multi-computer)[3] 사용했으며, 안정적으로 작동하는 것을 확인했다. 여기서 MPI는 표준 인터페이스라고 한 마디로 정의할 수 있다. 1992년도에 Oak Ridge 국립 연구소와 Rice 대학에서 처음 구성된 MPI Forum에서 기존에 널리 사용되고 있던 PVM, Express, P4,



Alpha PC
2 node

CPU : Alpah 633/533
Main Memory : 1GB/512MB
HDD Memory : 18GB(SCSI)/14GB(IDE)
OS : Alpha Linux

Pentium PC
18 node

CPU : Pentium III 650e - 2
Pentium II 500e - 1
Pentium II 450 - 1
Pentium II 350 - 14
Main Memory : 768MB~384 MB
HDD Memory : 20GB~8 GB
OS : Linux

Dual CPU
27 node

CPU : Pentium III 650e dual -3
Pentium III 550 dual - 22
Pentium 466A dual - 1
Pentium II 450 dual -1
Main Memory : 512MB ~ 1GB
HDD Memory : 40GB(IDE)
OS : Linux

Athlon PC
16 node

CPU : Athlon 550
Main memory : 768MB~384MB
HDD Memory : 20 GB
OS : Linux

Fig. 2 InterSup I의 시스템 및 네트워크 구성도

Zipcode, Parmacs 등의 소프트웨어에서 사용하고 있던 Interface들을 근간으로 하여 표준 Interface를 정의하게 된 것이다[4]. 그 이후에 MPI를 기준으로 하여, 여러 가지 다양한 소프트웨어들이 개발이 되었는데, 이러한 Software들을 MPI Implementation이라고 부르기도 하고, 약칭하여 MPI라고도 많이 부른다. Table 1에는 이러한 MPI Implementation들에 기본적인 기능들을 정리하였다.

Table 1 MPI에 정의된 주요기능들

· Task Management Routines
· Blocking/Non-blocking Point-to-Point Communication Routines
· Collective communication routines
· Communicator Management Routines
· Group Management Routines

여러 가지 다양한 MPI Implementation들 중에서 리눅스 기반으로 개발되었고, 가장 널리 사용되어 있는 MPICH와 LAM가 있다. 앞에서 언급한 것처럼, 본 연구에서도 MPICH와 LAM을 사용했다. MPICH와 LAM의 중요특징을 살펴보면, 공개 소프트웨어이므로 프로그램 및 상세한 문서자료들을 인터넷을 통할 수 있다는 장점이 있다. 그리고 C/C++뿐만 이 아니라 Fortran도 지원하며 인터넷의 기본 프로토콜인 TCP/IP를 기반으로 개발되었다

는 특징을 들 수 있다. 또한 다양한 하드웨어 플랫폼을 지원한다. 그 중에 MPICH는 지원하는 하드웨어 플랫폼이 LAM에 비해 상대적으로 다양한 것이 장점이다. LAM은 MPI에 추가적으로 Heterogeneous 환경에서 송수신을 고려한 IMPI(Interoperable MPI)[5]를 기준으로 개발되어, 인터넷 수퍼컴퓨팅에는 MPICH보다 더 적합한 특성을 보이고 있다. 실제로, 64 bit 시스템과 32 bit 시스템 사이의 데이터 송수신에서 MPICH는 불안정한 동작을 한 반면, LAM은 안정적으로 작동한 것을 수치실험을 통해 확인하였다. 그리고 작동방식에 있어서, LAM은 Remote shell보다 더 보안적인 면에서 뛰어난 SSH(Security Shell)과 함께 리눅스의 데몬(Daemon) 형태도 작동하며, 일반유저 권한으로 원격으로 쉽게 LAM 데몬을 작동시킬 수 있어서 인터넷 수퍼컴퓨팅에 더욱 적합하다고 할 수 있다. Table 2에 MPICH와 LAM의 기본적인 특징을 정리하였다.

Table 2 MPICH와 LAM의 비교

	MPICH	LAM
현재 버전	1.2	6.3.1
표준인터페이스 (MPI 1.2) 함수 지원여부	All	All(except MPI_Cancel)
소스 공개여부	Yes	Yes
지원하는 하드웨어 플랫폼	Solaris, HP UX, SPP UX, AIX FOR RS6000, Silicon Graphics IRIX, DEC UNIX, LINUX, SUN OS 4.x, CRAY XMP, YMP, C90, T3E, INTEL Paragon	Silicon Graphics IRIX, HP UX, Solaris, AIX FOR RS6000, LINUX

본 연구를 통해 구현된 InterSup I에 대한 시스템 구성을 살펴보자. 위의 Fig. 2에서 보는 바와 같이, 크게 4개의 서로 다른 소규모의 연구실에 흩어져 있는 63대의 컴퓨터를 이용하여 InterSup I을 구성하였다. 4개의 연구실 중 3개는 서울대학교 신공학관에 위치하고 있지만, 나머지 하나의 연구실은 완전히 건물이 다른, 실험동인 44동에 위치하고 있다. 한편, 전체 컴퓨터 수는 63대이지만, Dual CPU 시스템이 27대가 있으므로, 최대 활용가능한 CPU 노드 수는 90이 된다. 그러나 실제로 계산을 수행할 때는 이 중에서 계산에 동원될 수 없는 시스템도 있을 수 있다. 다른 계산을 수행하고 있다거나, 이용 가능한 하드웨어 리소스가 거의 없는 시스템도 있기 마련이다. 그래서 실제로 계산을 수행할 때는 이들 컴퓨터들 중에서 64 CPU 노드 시스템을 구성해서 InterSup I을 구성하게 된다. 시스템 구성을 살펴보면, Alpha 시스템, AMD K7(Athlon) 시스템과 인텔의 Pentium 시스템으로 구성되어 있다. 각 시스템에는 리눅스 시스템이 모두 설치되어 있으며, 컴파일러는 GCC(egcs-2.91.66)으로 설치하였다.

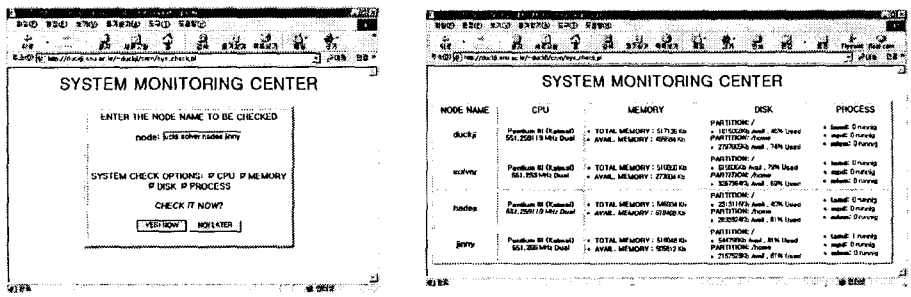


Fig. 3 노드 모니터링 시스템의 작동

그리고 네트워크 구성을 보면, Fig. 2에 나타난 바와 같이 각각의 연구실은 모두 서로 다른 게이트웨이를 가지고 있으며, 대부분 10Mbps 이더넷(Ethernet)을 이용하여 인터넷에 접속되어 있다. 컴퓨터간의 일대일 통신의 속도를 측정해 보면, 대략 850 kbytes/sec (6.8 Mbps)에서부터 11,000 kbytes/sec (88 Mbps)까지 분포되는 것을 알 수 있었다. 그리고 모든 시스템은 독자적인 IP 주소를 가지고 있으며, 특정 유저 아이디로 원격으로 SSH를 이용해서 접속할 수 있도록 설정되어 있다. 한편, 인터넷 상의 가상의 수퍼컴퓨터 시스템 즉, InterSup I을 구성하기 위해서는, 앞에서 언급한 것처럼 사용가능한 시스템들의 리

스트를 작성해야 한다. 본 연구에서는 웹 기반으로 SNMP(Simple Network Management Protocol)과 Perl(Practical Extract and Report Language)을 사용하여, 노드 모니터링 시스템을 자체 개발하였다. Fig. 3에 노드 모리터링 시스템의 작동모습을 나타내었다. Fig. 3에 보는 바와 같이, 정보를 검색하고 싶은 컴퓨터의 이름을 입력하면, CPU 종류와 주기억장치 용량 및 하드디스크 용량과 함께, 현재 이 시스템이 인터넷 수퍼컴퓨팅에 동원되고 있는지의 여부도 알 수 있다.

2.4 인터넷 수퍼컴퓨팅 예제

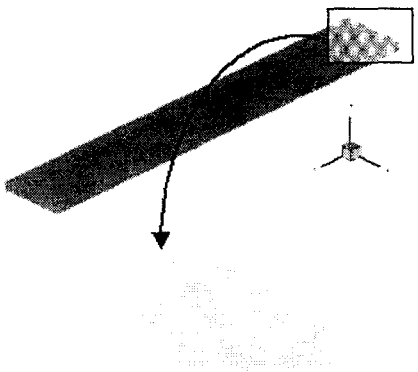


Fig. 4 가상 수치실험의 유한요소 모델링

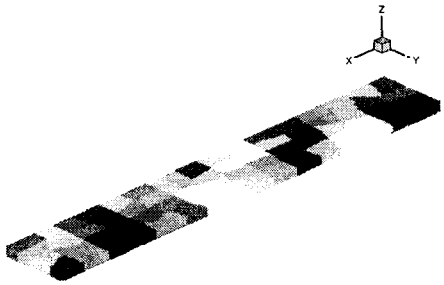
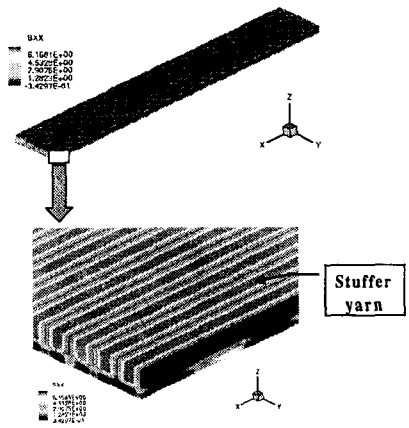
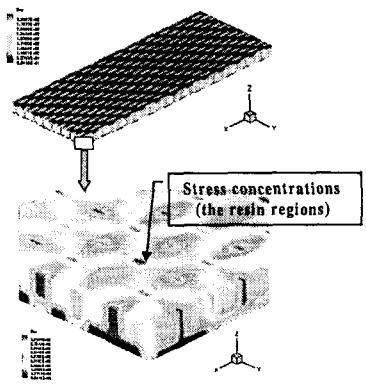


Fig. 5 유한요소 Mesh 분할결과 (64개 영역)



σ_{11} 분포

Fig. 6 가상 수치실험 결과 (x 방향 인장시험)



σ_{12} 분포

Fig. 7 가상 수치실험 결과 (xy 방향 전단시험)

본 연구에서는 가상 수퍼컴퓨터인 InterSup I을 활용하여, 기존의 기술로는 국내에 있는 수퍼컴퓨터로는 해석이 불가능한 500만 자유도 이상의 초대형 구조해석을 성공적으로 수행함으로써 인터넷 수퍼컴퓨팅의 가능성 및 타당성을 검증했다. 인터넷 수퍼컴퓨팅으로 해석한 예제는 3차원 직교직물 복합재료 구조물(3D orthogonal woven composite)의 상세 응력해석 및 가상실험을 통한 물성치 획득에 관한 문제이다. 이러한 초대형 구조해석을 수행하기 위해 병렬 Multifrontal Solver를 이용하였다[4]. 가상 수치실험을 위한 3차원 직교 직물복합재료 구조물의 유한요소 모델링은 [6]의 연구를 참조하여 구성하였으며, 전체

절점의 수는 1,714,960가 되고 8절점 고체요소의 수는 1,580,760 개나 된다. 전체 자유도 수는 무려 5,141,663이 된다. 해석모델은 실제로는 길이 26cm, 폭 2cm에 높이 2.79mm가 되는 시편에 해당된다. 그리고 병렬처리를 위한 영역분할은 Multifrontal Solver의 프론트 분할기법과 WEM(Weight Edge Mapping)을 이용한 다단계 그래프 분할알고리듬을 사용하였으며, 영역분할의 결과는 Fig. 5에 나타내었다[7]. 한번의 가상 수치실험을 InterSup I에서 64 CPU 노드를 사용하여 해석하기 위해서는 23 GB의 주기억장치와 101 GB의 보조기억장치가 필요했고, 대략 23 시간(83,041 초)의 시간이 소요되었다.

3. 결 론

본 연구를 통해서 경제적으로 훨씬 저렴하고, 확장성이 뛰어난 새로운 개념의 수퍼컴퓨팅 파워 구현의 방법으로서 인터넷 수퍼컴퓨팅 기술을 제안하였다. 기존에 이미 고유의 목적(워드 프로세싱, CAD 등)으로 설치되어 활용되어 지고 있고 서로 인터넷으로 연결되어 있는 PC들을 활용하여, 64 CPU 노드 시스템은 InterSup I을 구성하여 500만 미지수 수준의 초대형 고정밀 구조해석을 수행함으로써 그 타당성을 검증하였다. 그리고 인터넷 수퍼컴퓨팅 구현에 있어서 가장 중요한 것은 효율적인 병렬처리 컴퓨팅 알고리듬의 개발이며, 실제 계산 수행에 있어서는 서로 다른 기종간의 원활한 통신문제, 성능의 차이를 고려한 효율적인 작업분할이 중요하다는 것을 알 수 있었다.

그리고 인터넷 수퍼컴퓨팅 기술을 통해 상대적으로 열악한 국내 수퍼컴퓨팅 환경에서, 수퍼 컴퓨팅의 저변을 확대할 수 있을 것이다. 아직까지는 국내의 수퍼컴퓨터를 기업이나 학교의 연구원들이 손쉽게 접근해서 자기 연구에 활용하기에는 여러 가지 어려움이 많은 것이 현실이다. 이것은 경제적인 비용뿐만이 아니라, 최근 등장하는 대부분의 수퍼컴퓨터의 기본형태인 MPP형 시스템에서 효율적으로 작동하는 병렬처리 알고리듬의 개발이 미흡할 뿐만 아니라, 연구원들의 병렬처리에 대한 인식이 부족하기 때문이기도 하다. 이런 현실에서 인터넷 수퍼컴퓨팅 기술이 널리 보급된다면, 수퍼컴퓨터나 고가의 전용 계산 시스템을 구축하지 않고서도 병렬처리 알고리듬을 개발하여 그 효율을 검증할 수 있어서 병렬처리에 대한 저변확대에 기여할 수 있을 뿐만 아니라, 인터넷 수퍼컴퓨팅 환경에서 개발된 병렬처리 알고리듬은 MPP형 수퍼컴퓨터에서도 아주 효율적으로 작동하므로 수퍼컴퓨터 시스템을 활용한 연구도 활성화될 수 있을 것이다.

참고문헌

[1] Werthimer, Bowyer, Ng, Donnelly, Cobb, Lampton and Airieau (1997): The Berkeley SETI Program: SERENDIP IV Instrumentation; in the book "Astronomical and Biochemical
Origins and the Search for Life in the Universe", eds Cosmovici, Bowyer and Werthimer ,http://setiathome.ssl.berkeley.edu/
[2] W. Gropp and E. Lusk, "User's Guide for Mpich, A Portable Implementation of MPI version 1.2.0", Technical Report ANL/MCS-TM-ANL-96/6, Argonne National Laboratory, 1999, pp. 8-18
[3] Jeffrey M. Squyres, Kinis L. Meyer and Andrew Lumsdaine, Release notes, University of Notre Dame, 1999
[4] J. Malard, "MPI: A Message-Passing Interface Standard. History, Overview and Current Status," Technology Watch Report, Edinburgh Parallel Computing Center, 1996
[5] IMPI Steering Committee, IMPI-interoperable message-passing interface, http://impi.nist.gov/IMPI, 1998
[6] Tan P, Tong L, Steven GP. Behavior of 3D orthogonal woven CFRP comoposites. Part II. FEA and analytical modeling approaches. Composites, Part A: Applied Science and Manufacturing 2000;31:273-281[9]
[7] 김정호, "분산병렬처리 기법에 의한 초대형 구조물의 유한요소 해석", 서울대학교 공학박사 논문, 1998년 11월, pp. 20-24