Journal of the Korean
Data & Information Science Society
1999, Vol. 10, No. 2, pp. 473 ~ 483

PC-Clustering과 병렬가상장치에 의한 수치계산용 슈퍼컴퓨팅 PC 시스템 구축과 성능 테스트[1]

홍우표[2] · 김종제[3] · 오광식[4]

요 약

Linux 운영체계를 기반으로 한 PC 시스템을 고가의 상용 워크스테이션에 필적하도록 성능을 극대화하고, 각 단위 Linux PC 시스템을 네트워크를 통해 CPU와 memory를 공유하게 하는 병렬가상장치(PVM: Parallel Virtual Machine) 방식의 소프트웨어를 사용하여 군집(clustering)함으로써 슈퍼컴퓨터급 기능을 발휘하는 분산형 PC 시스템을 시험 구축하였다. 구축된 시스템의 성능을 PVM 방식의 병렬프로그램을 사용하여 벤치마킹해 본 결과, 병렬효율 (parallel efficiency)이 90%급에 접근함을 확인하였다.

주제어: PC clustering, 병렬가상장치, 병렬효율

1. 서 론

오늘날 micro-processor의 성능이 급속히 향상되어 초고속 연산이 가능해 짐에 따라 학문적인 연구에 있어서 컴퓨터의 활용은 연구활동의 보조 수단으로서 뿐만 아니라, 이전과는 다른 전혀 새로운 분야의 학문들을 개척해 가는 탐색도구로서도 이용되는 질적인 변화를 초래하고 있다(Balluder 1995). 이에 따라, 국내외를 막론하고 많은 연구기관에서는 경쟁적으로 연산속도가 보다 빠른 고성능의 슈퍼컴퓨터를 확보하기 위해 많은 예산을 책정하고 도입·운용을 서두르고 있다. 그러나, 초당 수백억번 이상의 연산이 가능하며 안정된 네크워크 서버(network server) 역할을 수행할 수 있는 슈퍼컴퓨터를 확보하는데 있어서는 막대한 구입비용과 높은 유지비용 등 많은 경제적인 제약을 받고 있는 실정이다.

한편, 공개형 Unix인 Linux 운영체계의 발전에 힘입어 Linux 운영체계를 탑재한 저가 개인용 컴퓨터(PC) 시스템은 고가의 상용 워크스테이션 (workstation) 보다 가격 대 성능비

[1]이 연구는 대구효성가톨릭대학교 1997년도 연구소지원 정책연구비에 의한 결과임.
[2](712-702) 경북 경산시 하양읍 금락 1리 330, 대구효성가톨릭대학교 물리학과 조교수
[3](712-702) 경북 경산시 하양읍 금락 1리 330, 대구효성가톨릭대학교 물리학과 교수
[4](712-702) 경북 경산시 하양읍 금락 1리 330, 대구효성가톨릭대학교 정보통계학과 교수

면에서 탁월한 우수성을 나타냄이 속속 입증되고 있다 (Linux 관련 웹사이트 1999). 아울러, 최근에 시판되는 Pentium III나 Alpha CPU 등이 장착된 PC를 사용하면 개인 차원에서도 일반 상용 워크스테이션에 필적될 정도로 고성능인 저가형 네트워크 서버를 충분히 구축할 수 있게 되었다.

본 논문에서는 다수의 저가 PC를 사용하여 각 단위별로는 상용 워크스테이션 수준의 연산 능력을 유지하면서 LAN(Local Area Network)으로 군집하여 사용할 때 슈퍼컴퓨터급의 연산성능을 발휘할 수 있는 PC 시스템의 구축과 활용 가능성을 소개한다. 특히, 국내외에서 현재 활발하게 시도되고 있는 병렬가상장치((PVM: Parallel Virtual Machine, 일명 a.k.a.) 방식의 분산형 PC 시스템을 연구실 단위에서 실제로 시험 구축하여 성능 테스트를 해 본 결과를 제시한다.

본 논문은 다음과 같이 구성된다. 제 2장에서 분산형 슈퍼컴퓨팅 시스템을 구성하기 위해 필요한 PVM의 개요와 소프트웨어의 설치 과정을 간략히 소개한다. 제 3장에서는 실제 데이터의 병렬 처리를 위해 작성한 간단한 행렬계산 프로그램에 대한 설명과 구축된 PC 시스템의 벤치마킹 결과를 제시하고, 제 4장에서는 본 연구 결과의 요약과 네트워크형 슈퍼컴퓨팅 PC 시스템의 전망을 제시한다.

2. 분산형 슈퍼컴퓨팅 PC 시스템 구축

2.1. Parallel Virtual Machine (PVM)의 소개

컴퓨터의 병렬처리(parallel processing) 즉, 큰 계산을 작은 단위의 작업들로 나누어서 동시에 수행하는 연산처리 기법은 현대 컴퓨터 연산 알고리즘에 있어서 가장 중요한 방법으로 부상했다. 병렬처리 기법은 (i) 수백개 이상의 CPU들로 이루어진 거대 다중프로세서(MPP: massively parallel processors) 방식과 (ii) 네트워크에 분산된 컴퓨터의 CPU 및 메모리를 공유하는 분산형 컴퓨팅(distributed computing) 방식으로 크게 두 종류로 구분되어 진다. 분산형과 MPP 방식의 공통점은 메시지 전달(message passing)을 사용한다는 점이다. 모든 병렬처리방식에 있어서 데이터는 주어진 계산 상의 작업들(tasks) 사이에 원활하게 교환된다. 이러한 데이터의 교환을 극대화하기 위한 방법으로 메모리 공유하기(shared memory), 병렬화된 컴파일러(parallel compilers) 사용, 그리고 메시지 전달 등이 고려되었다. 메시지전달 방식은 많은 프로세서들에 의해 지원되며 응용성과 편리성 때문에 널리 사용되고 있다.

메시지전달을 사용하는 분산형 컴퓨팅의 대표적인 예로는 PVM 방식과 MPI(message passing interface) 방식이 있다 (Graham, Fagg, and Dongarra 1996): MPI는 여러 MPP 시스템에서 다양한 목적을 가진 사용자가 효과적인 병렬 프로그래밍을 편리하게 작성할 수 있도록 도와주는 라이브러리이다. MPI 라이브러리는 Fortran 77, C, C++ 등에서 사용할

수 있고, 같은 source code를 여러 다른 병렬처리 컴퓨터 환경에서도 그대로 사용할 수 있다. MPI를 이용한 프로그래밍은 비교적 간단한 편이어서 사용자는 한 개의 프로그램만을 작성하면 된다. 이 프로그램을 각 프로세서들이 동시에 각각 독립적으로 수행하게 되는데, 이때 각 프로세서에 할당되는 고유번호를 이용하여 자료를 주고받거나 각각의 프로세서에 다른 일을 할 수 있게 한다.

PVM은 여러 종류의 컴퓨터들로 구성된 시스템에서 일반적인 목적의 병렬처리 컴퓨팅을 지원하는 소프트웨어 도구와 라이브러리 모음이다. 최초의 PVM package인 PVM/1.0은 1989년 미국 Los Alamos 소재 Oak Ridge National Laboratory의 한 연구실에서 개발되어 그 연구실의 수치계산 도구로 사용되었다. 그 후, 1991년 PVM/2.0 version은 University of Tennesse, Knoxville에서 독자적으로 개량되어 전세계의 과학발전을 위해 무료 배포되고 있다. 현재 개량된 PVM/3.4가 발표되어 전세계적으로 많은 사용자들로부터 각광을 받고 있다 (PVM 웹사이트 1999). PVM 라이브러리는 C, C^{++}, Fortran 77 등을 모두 지원한다. PVM 라이브러리는 본격적인 병렬처리 프로그램을 작성하는데 이용할 수 있는데, MPI에서 이용할 수 있는 모든 일 이외에도 각 프로세서별로 완전히 다른 프로그램을 실행하여 각 프로그램 간의 자료를 전송을 하거나 필요한 프로세서에 다른 프로그램을 수행시키며 또한 중단시킬 수 있는 등 다양한 작업을 가능하게 한다 (Sunderam 1990).

PVM 소프트웨어의 가장 큰 특징으로는 PVM 관리자인 PVMD(PVM daemon)가 있다는 점이다. PVMD는 군집된 모든 컴퓨터들이 개별적인 작업을 수행하게 하고 작업의 동시성(synchronization)을 자동 관리하는 역할을 수행한다. 그러므로 네트워크상의 어떤 컴퓨터를 PVM에 참가시키기 위해서는 PVMD라는 프로세서를 작동하면 된다. PVMD는 각 컴퓨터에 1개만 존재하며, N 개의 컴퓨터들로 구성되는 네트워크 상의 PVM에는 1개의 '주인(master)' PVMD와 $N-1$ 개의 '노예(slave)' PVMD들로 구분되어진다. PVM 상에서 주인 PVMD는 사용자가 로그인 하는 컴퓨터에서 일반적으로 시작되며 전체 PVMD들을 가동하게 하는 역할을 담당한다. 만약, 한명 이상의 사용자가 네트워크 상의 모든 시스템을 사용하여 새로운 PVM을 구성할 경우는 기존의 사용자와 차별되지 않는다. 즉, CPU와 메모리가 허용되는 한, 다수의 사용자가 각자의 PVM을 구성할 수 있다는 것이다. 마치 우편배달을 위한 서비스 시스템을 PVM이라 한다면 각 컴퓨터의 PVMD는 지역우체국과 같은 역할을 담당한다고 비유해 볼 수 있다. 보다 구체적으로 설명하면, 어떤 컴퓨터에서 수행 중인 여러 작업들은 그 곳의 PVMD에 의하여 각각의 작업들에 대한 정보(information)를 정리·포장하여 그 작업의 일련번호에 해당하는 작업번호(task identification)를 부여한 다음, 네트워크 상의 각 컴퓨터들의 PVMD로 배달된다. 배달된 작업들은 PVMD에 의하여 정보가 풀어져 필요한 작업들에 배당되어 수행되게 된다 (Geist and Suderam 1992). 여기서 주목할 점은 하나의 컴퓨터에서 다수의 작업들이 수행될 수 있으며 PVMD에 의하여 정보가 제어된다는 점이다. 따라서, 다중의 CPU를 가진 시스템에 있어서도 PVM의 사용은 가능하다 (Geist 1994).

그림1은 이러한 PVM의 역할을 개략적으로 도식한 것으로, 네트워크에 연결된 각 컴퓨

터에서 수행하는 작업들(T1, T2, T3)이 어떻게 전달되는가를 보여주고 있다. PVM은 메시지를 전달하고 받고 (communication), 작업들을 관장하며 (process control), 사용자들이 프로그램을 작성할 수 있는 라이브러리 등으로 구성되어 지는데, 이러한 각 기능은 PVM 소프트웨어가 제공하는 PVMD와 PVM 라이브러리에 의해 수행되게 된다. PVMD는 자기 자신이 수행하는 다수의 작업들을 관리하고 다른 PVMD로 자신이 수행하는 작업에 관한 정보를 전달하거나 받게 된다.

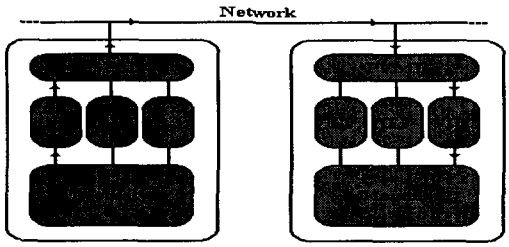

Figure 1. PVM과 작업을 관장하는 PVMD와의 관계.

2.2. PVM의 설치 및 구성

PVM의 최신 버전은 PVM 사이트 (XPVM 웹사이트 1999)에서 소스코드와 문서를 무료로 다운로드하여 설치한다. PVM은 사용자가 자기의 디렉토리에 설치하거나 아니면 시스템 관리자(root)가 지정한 특정 디렉토리에 설치한다. 시스템 관리자가 PVM/3.4를 설치하는 과정을 간략하게 기술하면 다음과 같다:

- PVM ftp 사이트(ftp://www.netlib.org/pvm)에서 내려받은 pvm3.4.1.tar.gz 파일을 적당한 공간 (/usr/local)에서 gunzip과 tar을 사용하여 풀면 /usr/local/pvm3 디렉토리가 만들어진다.
- PVM root 디렉토리를 설정하기 위하여 /etc/profile에 파일에 다음을 추가한다:

```
PVM_ROOT=/usr/local/pvm3
PVM_DPATH=$PVM_ROOT/lib/pvmd
export PVM_ROOT PVM_DPATH.
```

- /usr/local/pvm3에서 makefile을 실행하면 /usr/local/pvm3/bin/LINUX에 pvm 실행파일들이 생성된다.
- /root에서 .rhosts 파일에 사용을 원하는 네트워크상의 모든 hosts의 이름을 기록한다.

- PVM 소프트웨어를 설치한 후 PVM을 구동하기 위하여 /usr/local/pvm3/bin/LINUX에 있는 실행파일 pvm을 실행하여 PVMD를 시작하고 원하는 네트워크 상의 컴퓨터를 PVM에 등록한다.

PVM을 효율적으로 사용하기 위한 도구로는 XPVM (XPVM 웹사이트 1999)이 있다. XPVM은 X-Window상에서 PVM 프로그램을 실행하고 그 진행과정을 모니터할 수 있는 소프트웨어다. XPVM을 이용하면 PVM상의 병렬 작업들의 통신 상태를 시각적으로 확인하고 작업량을 조절하며 프로그램의 디버깅(debugging)을 원활하게 할 수 있다. 그림 2는 XPVM 1.2.1의 실제 화면을 나타낸 것이다. XPVM은 TCL/TK 무료 소프트웨어와 같이 사용하여 설치할 수 있으며 Linux 운영체계를 각 컴퓨터에 설치할 때 포함된다. 시스템 관리자로서의 XPVM 설치 과정은 다음과 같다:

- PVM이 설치된 디렉토리에 tar을 사용하여 PVM web site에서 가져온 xpvm1.1.tar 를 풀어 /usr/local/pvm3/xpvm 디렉토리를 만든다.
- /usr/local/pvm3/xpvm의 makefile을 열어서 TCL/TK 라이브러리 위치를 설정한 다음 컴파일하면 /usr/local/pvm3/bin/Linux에 XPVM 관리자를 구동하는 실행파일 xpvm 이 생성된다.
- /etc/profile 에 다음을 추가한다:

```
XPVM_ROOT=/usr/local/pvm3/xpvm
export XPVM_ROOT
```

- 사용자의 디렉토리에 .hosts의 내용과 동일한 .xpmv_hosts 파일을 만든다.

3. PVM PC 시스템의 벤치마킹

이상에서 소개한 PVM PC 시스템은, LAN이 설치된 대학이나 연구소의 연구실 단위에서 구성할 경우 부가적인 하드웨어 설치가 필요없이 간단하게 구축될 수 있다. 필자들은 1997년 8월 부터 대구효성가톨릭대학교 물리학과 연구실에 Linux 운영체계의 Pentium Pro 200 MHz PC 5대를 10 MBPS의 Ethernet으로 연결하여 직접 시험용 PVM PC 시스템을 구축하고, 구축된 PVM 시스템에 대해 실제 연산성능을 벤치마킹해 보았다. 아래에 Fortran으로 작성한 한 테스트용 PVM 프로그램과 벤치마킹의 결과를 소개한다.

PVM을 사용하여 병렬프로그램을 작성할 때 가장 편리한 것은 '대중(crowd)'이라는 알고리즘이다. 이는 네트워크 상에서 비슷한 용량의 CPU에 동일한 프로그램의 일부분을 수행하면서 중간계산 과정을 PVMD를 통하여 교환하고 최종 결과를 특정한 프로세서가 관리하는 체제로서 두 가지 방식으로 처리되어 진다 (Geist 1994):

- 주인-노예*(master-slave)* 방식 - 전체 PVM에 속하는 모든 프로세서들을 관리하는 독자적인 '지휘(control)' 프로그램, 즉 주인 프로그램을 작성하여 각 프로세서를 작동시키고 초기화하며 프로그램 수행결과를 종합하여 결과를 나타나게 한다. 노예 프로그램들은 실제 계산들을 수행하게 된다. 이때 각 노예 프로그램에 할당되는 계산량은 주인 프로그램에 의하여 고정적(static) 혹은 동적(dynamic)으로 지정할 수 있다.
- 노예 만*(slave-only)*의 방식 - 한 개의 프로그램을 여러번 반복하여 실행하는 경우로서 노예 프로세서들 중의 하나가 비계산적인 임무를 수행하는 동시에 그 자신 또한 계산에 참여하는 방법을 말한다.

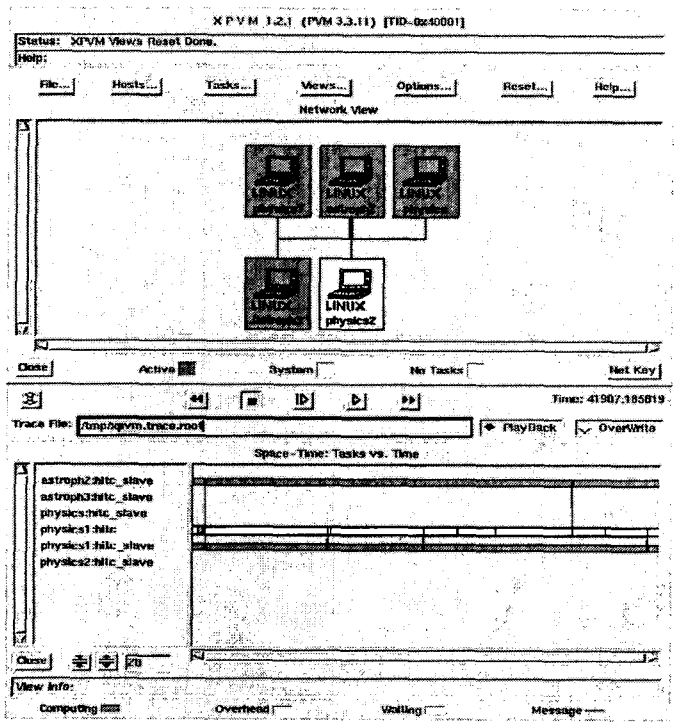

Figure 2. PVM 작업을 실시간적으로 확인·통제·관리하는 XPVM 1.2.1 화면.

대중 알고리즘은 주어진 작업을 세 단계에 걸쳐 수행하게 된다: (i) 제 1단계 - 프로세서 그룹을 초기화하여 각 프로세서에 작업을 할당한다. (ii) 제 2단계 - 할당된 계산을 각 프로세서에서 실행한다. (iii) 제 3단계 - 결과를 종합하고 최종 출력을 보여준 후, 각 프로세서 그룹의 임무를 중단시킨다. 이러한 작업은 XPVM 1.2.1 화면에서 확인할 수 있는데, 그림 2에서는 PVM을 구성하고 있는 5 대의 Linux PC cluster 시스템에 주인-노예방식의 수치계산 프로그램이 진행 중인 상태를 보여주고 있다.

이제, 행렬의 곱 $\mathbf{C}(n,n) = \mathbf{A}(n,n) \times \mathbf{B}(n,n)$을 계산하는 직렬 프로그램(smatrix.f)의 병렬화를 고려하자. (이하, 상세한 프로그램 및 그 사용법은 http://physics.cataegu.ac.kr/~wphong/linutk/index.html을 참조.) 가능한 병렬 알고리즘 중에서 계산할 작업량을 계산 이전에 결정하는 경우, 가장 편리한 PVM 프로그램 알고리즘은 주인-노예 방식이다. PVM을 이용한 smatrix.f의 병렬프로그램화는 주인 PVM 프로그램인 matrix-master.f와 노예 PVM 프로그램인 matrix-slave.f로 구성된다.

먼저, 주인이 존재하는 PVM에서 호스트를 결정하고 필요한 숫자의 노예 프로세서들을 생성하여 계산할 수 있는 작업량을 분배한다. 주인 프로세서는 행렬 $\mathbf{A}$를 모든 가능한 PVM 상의 프로세서로 보내고 행렬 $\mathbf{A}$에 곱할 행렬 $\mathbf{B}$의 column들을 수행 가능한 노예 프로세서의 숫자를 고려하여 분배한다. 각 노예 프로세서는 $\mathbf{A} \times \mathbf{B}$를 계산하여 주인 프로세서가 있는 컴퓨터로 되돌려 보낸다. 주인 프로세서는 각 노드에서 전송된 값을 정리하여 $\mathbf{C}$를 계산한다.

그림 3은 PVM PC 시스템을 사용하여 $\mathbf{C} = \mathbf{A} \times \mathbf{B}$를 실제 계산하는데 소요된 CPU 시간과 사용된 컴퓨터 댓수와의 상관 관계를 나타낸 것이다. 행렬의 크기가 $n = 500, 750, 1000$인 경우, 1대의 Pentium Pro 200 MHz Linux PC에서 smatrix.f를 개별 수행하면 각각 $t_1(n) =$ 78초, 288초, 758초의 CPU 시간이 요구되었다. 그런데, PVM PC 시스템에 대해 병렬계산을 수행하면 $\mathbf{C}$의 계산에 소요된 총 CPU 시간 $T(n, N_c)$는 PVM에 참가한 컴퓨터의 댓수 N_c 및 1대의 컴퓨터에서 직렬방식의 계산에 소요된 시간 $t_1(n)$과의 관계는 근사적으로 다음과 같이 주어진다.

$$T(n, N_c) = t_1(n)\ N_c^{-\eta} \tag{1}$$

여기서, η는 계산프로그램의 병렬효율로서 n =500, 750, 1000에 대해 각각 $\eta \simeq 0.9, 0.87, 0.85$로 추정되었다. (이상적인 병렬효율은 $\eta = 1$인 경우이다.) 병렬 효율은 PVM에 참가한 컴퓨터 댓수가 증가할수록 감소하며, 메세지의 전달 크기 즉, 행렬의 크기가 증가할수록 감소하는 경향을 보였다.

행렬 크기에 따른 병렬효율의 감소 경향은 주로 각 프로세서들 간의 정보전달 시 발생하는 네트워크 상에 걸린 부하에서 기인하는 것으로 예측된다. 본 연구에서 사용한 10 MBPS Ethernet 카드는 적절한 크기의 병렬계산 작업을 충분히 수행할 수 있을 정도로 빠른 장치인데, 실제 병렬프로그램을 수행했을 경우에도 심각한 정보전달 상의 지연 문제는 발생되지 않았다. 그러나, 교환되는 정보량이 급격히 증가하는 대규모 행렬을 계산할 경우에는 보다 빠른 고속 네트워크의 사용이 PVM의 병렬효율을 향상시킬 가장 중요한 요소가 될 것이다. 실제로 ATM(Asynchronous Transfer Mode) 혹은 FDDI(Fiber Distributed Data Interface)방식의 초고속 전산망을 이용하여 PVM을 구성하는 경우, 계산의 병렬효율이 20% 이상 증가된다 사실이보고되고 있다 (Flower, Kolawa, and Bharadwaj 1991).

이러한 문제에 대해 저비용으로 PVM의 성능을 향상시킬 수 있는 가장 경제적인 방안의

하나로는 100 MBPS fast Ethernet을 사용하여 군집하는 것이다. 이 경우 이론적으로는 약 200대 정도까지 Linux PC의 군집이 가능하며 거의 선형적인 병렬효율 향상을 기대할 수 있다. 특히, Linux 운영체계에서는 100 MBPS 네트워크 카드에 대한 디바이스가 제공되고 TCP/IP 프로토콜을 사용하기 때문에 가장 현실적인 방법으로 간주될 수 있다. 이 방식을 택하여 네트워크를 구성하면 각 노드간의 통신전달시간은 자체 loopback 시간에 거의 접근하게 되므로 10 MBPS Ethernet를 사용하는 경우보다 약 1/2 이하의 전달시간 감축이 기대된다. 빠른 네트워크를 사용하여 메세지 전달의 시간을 줄이는 방법 이외에도 대규모 메모리 공유(memory channel) 방법을 사용하거나 버퍼에 전달되는 정보의 양을 줄이는 방법인 MPI 방식의 병렬프로그램을 사용하면 병렬 효율성을 거의 90% 이상 수준에 까지 끌어올릴 수 있다 (Sterling, Savarese, Becker, Fryxell, and Olson 1995).

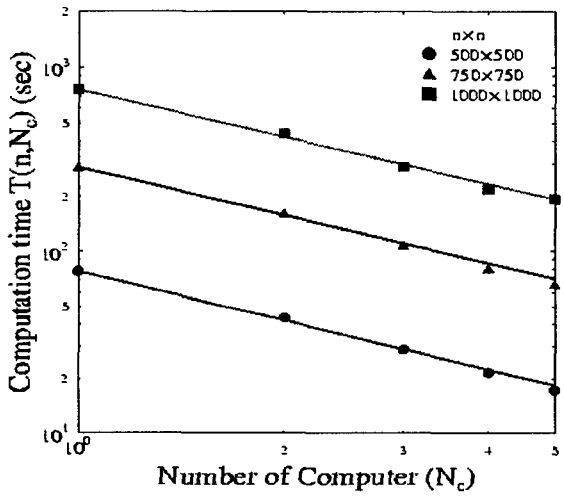

Figure 3. 가상병렬장치의 병렬효율 (η). 병렬효율 η는 소요된 CPU 시간 (T)과 사용된 컴퓨터 댓수(N_c)의 상관 직선의 기울기 값으로써 추정된다.

4. 요약 및 전망

본 논문에서는 이미 국내외에서 많은 관심을 불러일으키고 있는 분산형 병렬컴퓨팅 시스템인 PVM에 대한 개념 소개와 설치 방법, 그리고 간단한 벤치마킹을 통하여 PVM PC 시스템의 실제 성능 테스트를 하고 그 활용 가능성을 점검하여 보았다. 5대의 Pentium Pro 200 MHz Linux PC를 10 MBPS Ethernet 으로 군집한 시험용 PVM PC 시스템에서 병렬효율은 약 90%에 접근하고 있다.

Main Frame 슈퍼컴퓨터를 확보하기 곤란한 우리의 대학이나 연구소의 재정적 현실을 고려할 때, 고성능 테크니컬 컴퓨팅 및 데이터 분산처리 환경 구축이 절실한 계산과학 분야

에서는 이상에서 논의한 저가의 PVM PC 시스템을 구성하여 사용하는 것도 연구환경 개선을 위한 효율적인 대안으로 고려될 수 있을 것으로 생각된다. 하드웨어의 부품별 성능을 극대화하고 Linux 운영체계를 설치한 경우, 웹서버, Data Base 서버, 메일서버, FTP 서버 등 전산업무 환경에 있어서 Linux PC 서버는 연산속도, 안정성, 보안성 등 그 총체적 성능이 상용 Unix 서버나 NT 서버와 비교해 보더라도 가격 대 성능비는 대단히 우수함을 인정받고 있다. 또한, 연산 속도를 최우선으로 하는 테크니컬 컴퓨팅 환경에서도 이미 600 MHz 이상의 CPU clock을 지닌 Alpha 프로세서 및 Pentium III 프로세서의 탁월한 부동소수점 연산기능으로 인해 PVM PC 시스템은 새롭게 관심을 끌고 있다. 특히, 병렬효율을 향상시킬 100 MBPS 이상의 고속 네트워크 장치의 가격 하락과 함께 네트워크의 작동성능을 극대화시키는 드라이버를 Linux에서 제공하는 데 힘입어 네트워크 분산형 컴퓨팅에 대한 관심은 더욱 높아질 것으로 기대된다.

이러한 추세에 따라, 이미 국외(CESDIS 웹사이트 1999)뿐만 아니라 국내(한국 Beowulf 웹사이트 1999)에서도 네트워크 병렬 컴퓨팅 그룹 사이트를 개설하고 있고 다수의 대학과 연구소에서는 PVM 및 MPI 방식의 프로그램을 작성하여 실제 작업을 진행하고 있다. 그 예로서, 현재 전 세계에서 운용중인 약 500대의 슈퍼컴퓨터 중에서 113위에 등록된 Avalon 네트워크 슈퍼컴퓨터(Avalon cluster 웹사이트 1999)는 140개 알파 Linux node로 구축되어 대규모 수치계산을 성공적으로 수행한 바 있으며, 영화 '타이타닉'에서는 128개의 Alpha 프로세서를 군집한 Linux PVM 시스템이 시뮬레이션에 효율적으로 이용된 바 있다 (타이타닉 웹사이트 1999).

참 고 문 헌

1. Balluder, K. B. (1993). Selecting an operating system: Linux, *Computers in Physics* 10, p. 17-24.

2. Geist, G. A., Sunderam, V. S. (1992). Network-based concurrent computing on the PVM system, *Concurrency: Practice and Experience*, 4 (4): 293-311.

3. Geist, G. A. (1994). *PVM: Parallel Virtual Machine; A users' guide and tutorial for networked parallel computing*, MIT Press.

4. Graham, E., Fagg, E., and Dongarra, J. J. (1996). PVMPI: An integration of the PVM and MPI systems, *Calculateurs Parallels*, 2-10.

5. Sterling, T., Savarese, D., Becker, D., Fryxell, B., and Olson, K. (1995). Communucation Overhead for Space Science Applications on the Beowulf Parallel Worstation, *Proceedings of the fourth IEEE Symposium on High Performance Distributed Computing (HPDC)*, 23-30.

6. Sunderam, V (1990). PVM: A Framework for Parallel Distributed Computing, *Concurrency: Practice and Experience*, 315-339.

7. Avalon Cluster 웹사이트 (1999). http://cnls.lanl.gov/avalon.

8. CEDIS 웹사이트 (1999). http://loki-www.lanl.gov.

9. Linux 관련 웹사이트 (1999), http://www.linux.org.

10. KAIST Beowulf 웹사이트 (1999). http://nurapt.kaist.ac.kr/beowulf.

11. PVM 웹사이트 (1999). http://www.epm.ornl.gov/pvm.

12. XPVM 웹사이트 (1999). http://www.epm.ornl.gov/xpvm.

13. Flower, J., Kolawa, A., and Bharadwaj, S. (1991). The Express way to distributed processing. *Supercomputer Review*, p. 54-55.

14. 타이타닉 웹사이트 (1999). http://www.ssc.com/lj/issue46/2494.html.

Construction and Performance Test of a Supercomputing PC System using PC-clustering and Parallel Virtual Machine

Woo-Pyo Hong[5] · Jong-Jae Kim[6] · Kwang-Sik Oh[7]

Abstract

We introduce a way to construct a supercomputing capable system with some networked PCs, running the Linux operating system and computing power comparable with expensive commercial workstations, and with the Parallel Virtual Machine (PVM) software which enables one to control the total CPUs and memories of the networked PCs. By benchmarking the system using a PVM parallel program, we find that the system's parallel efficiency is close to 90 %.

Key Words and Phrases: PC Clustering, Parallel Virtual Machine, Parallel efficiency

[5]Assistant Professor, (712-702) Department of Physics, Catholic University of Taegu-Hyosung
[6]Professor, (712-702) Department of Physics, Catholic University of Taegu-Hyosung
[7]Professor, (712-702) Department of Information and Statistics, Catholic University of Taegu-Hyosung