Beowulf mailing list FAQ

Kragen Sitaker kragen@pobox.com Version 2 1999년 5월 13일

양 유성, yooseong@kldp.org 2000년 6월 1일

이 FAQ는 Beowulf mailing list에 자주 반복되는 질문들에 대한 대답을 미리 언급해놓은 것이다. 수정은 언제나 환영한다. 잘못된 것이 있으면 고치겠다. 1999년 5월 13일은 내용이 최근에 편집된 날이며 여기 나온 내용이 최근에 갱신된 날짜가 아님을 밝힌다.

목 차

제 1 절 Acnowledgement **2**

제 2 절 짧은 답변들 (메일링 리스트에 질문을 올리기전에 먼저 읽어주기 바람!) **2**

제 3 절 긴 답변들 **4**

제 4 절 보충 정보와 자료 **5**

4.1 Beowulf에 유용한 소프트웨어들: 5

4.2 컴파일러: 5

5

5

4.5 BVIEW: (Beowulf의 상태를 감시하는 소프트웨어) [1999-05-13] 5

4.6 bWatch: (Beowulf의 상태를 감시하는 많은 소프트웨어 제공) [1999-05-13] 5

4.7 BPROC: (노드간의 프로세스를 시각화시켜 보여줌, 각각의 노드에 fork()를 허락하고 프로세스 이동을 가능케 하며, 모든 노드간에 kill()을 허락한다. – 현재는 pre-알파머신에 이용) [1999-05-13] 6

4.8 procps를 위한 클러스터 패치: (ps와 같은 /proc기반 프로그램들을 컴파일하게 하여 모든 클러스트에게 그 내용을 전달하는 역할을 한다.) [1999-05-13] 6

4.9 SMILE 클러스터 관리 시스템: (모든 노드에 명령을 실행하고 각 노드나 노드의 집합을 멈추게 하고 노드의 상태를 모니터할 수 있음. 클러스터를 좀더 관리하기 쉽게 만들어줌.) [1999-05-13] 6

4.10 병렬 가상 파일시스템: (LD_PRELOAD기반의 파일시스템의 변형은 많은 디스크에 걸쳐 커다란 파일을 분산 시킬 수 있음. 커다란 데이타 집합을 고성능으로 접근하게 해줌) [1999-05-13] 6

4.11 빠른 수학 라이브러리와 자유롭게 얻을 수 있는 빠른 수학라이브러리: (표준 수학 함수들을 빠르게 만들어줌) [1999-05-13] 6

4.12 '복제' 노드를 설정하는 스크립트: (노드를 새롭게 Beowulf에 첨가할 때 편하게 할 수 있음) [1999-05-13] 6

4.13 클러스터에서 여러가지를 수행하게 해주는 스크립트 – 백업, 셧다운, 시스템 재시작, 각 노드에서의 명령 [1999-05-13] 6

4.14 BERT 77: ("포트란 코드를 자동적으로 병렬화 시켜주는 프로그램") [1999-05-13] 6

4.15 펜티엄 컴파일러 그룹에서 나온 Pentium GCC, aka PGCC: (보통의 gcc보다 5%-30%의 속도 향상을 보여주는 펜티엄에 최적화된 컴파일러) [1999-05-13] 7

제 5 절 문서들: **7**

제 6 절 책들: **7**

제 1 절 Acnowledgement

Rober G. Brown, Greg Lindahl, Forrest Hoffman과 Putchong Uthayopas가 이 FAQ에 중요한 정보를 제공하였다.

Kragen Sitaker kragen@pobox.com는 이문서를 편집하고 몇몇 질문들에 대한 대답을 썼다. 잘못된 문서내용과 오래된 내용은 그의 잘못이다.

제 2 절 짧은 답변들 (메일링 리스트에 질문을 올리기전에 먼저 읽어주기 바람!)

만일 좀더 깊은 답변을 원한다면, "긴 답변들"부분을 참조하기 바란다.

- 무엇이 Beowulf인가? [1999-05-13] Beowulf라하는 것은 고성능의 많은 병렬컴퓨터를 상용화된 하드웨어와 리눅스와 FreeBSD와 같은 free software를 이용하여 빠른 사설 네트워크를 이용해서 연결해 놓은 것이다. 이는 PC나 워크스테이션들의 묶음으로 되어 있는데 주로 고성능 계산에 주로 이용된다. 클러스터에 있는 노드들은 사람들이 주로 사용하는 책상위에 있지 않다. 외부세계와의 연결은 서버인 노드만이 할 수 있도록 되어있다. 몇몇 리눅스 클러스터들은 속도보다는 안정성에 기반을 두고 만들어졌다. 이것들은 Beowulf라고 할 수 없다.

- Beowulf 소프트웨어는 어디서 구할 수 있나요? [1999-05-13] Beowulf라고 불리는 소프트웨어 패키지는 없다. 단지 Beowulf를 제작하기 위해 많은 사람들이 만들어 놓은 몇몇가지 소프트웨어가 있다. 그 소프트웨어는 MPICH (역자주: MPI(Message Passing Interface)의 일종으로 MPI Chameleon의 약자)와 LAM(역자주: MPICH와 같이 MPI의 일종으로 Local Area Multicomputer의 약자), PVM(역자주: Private Virtual Machine으로 MPI와는 달리 주로 이기종간의 컴퓨터 연결에 이용됨), 리눅스 커널, 리눅스 커널의 채널본딩 패치(이는 여러분이 빠른 "버츄얼" 이더넷 인터페이스에 다중의 이더넷 인터페이스를 연결시켜준다.)과 리눅스 커널에 global pid space 패치(이는 Beowulf에서 돌아가는 모든 프로세스를 ps나 kill등의 명령으로 제어할 수 있게 해줌), DIPC(이는 sysv 공유 메모리, 세마포어와 메세지큐를 투명하게 사용할 수 있게 해줌)

- 소프트웨어를 가지고 Beowulf에서 작동시킬 수 있고 빠르게 할 수 있나요? [1999-05-13] 될 것입니다. MPI나 PVM, 네트워크 소켓 또는 Sysc IPC를 이용해서 통신할 수 있는 병렬작업으로 본래의 작업을 분리하는 일이 필요할 것입니다. 그런후 다시 컴파일을 해야합니다.

  또는 Greg Lindahl이 지적했듯이 만일 여러분이 다른 입력 파일을 가지고 수천번의 동일한 프로그램을 실행하고 싶다면 쉘스크립트를 이용하면 충분할 것입니다.

  Christopher Bohn이 지적한 것처럼 multi-thread 소프트웨어도 자동적으로 속도향상을 꾀할 수 없다. multi-threaded 소프트웨어는 공유메모리를 가정한다. 몇몇 배포된 공유메모리 패키지가 개발중이다. (DIPC, Mosix, ...), 하지만 SMP 머신을 위해 만들어진 소프트웨어 내부에 있는 메모리 접근 형태는 궁극적으로 DSM 머신의 성능저하를 가져온다.

- PVM? MPI? Huh? [1999-05-13] PVM과 MPI는 클러스터에서 포트란이나 C를 이용해서 작동하도록 만든 message-passing 병렬 프로그램이다. PVM은 MPI가 나오기 전까지 사실상의 표준이었다. 하지만

아직도 많이 이용되며 성능면에서 좋다. MPI (Message Passing Interface)는 portable message-passing 병렬 프로그램에 사용되는 표준이다. MPI는 MPI Forum을 통해서 표준화가 되었고 모든 대용량의 수퍼 컴퓨터들에 사용되고 있다. (역자주: 보통 PVM은 이기종간의 병렬처리에 이용되고 MPI는 동종간의 처리에 많이 이용된다.) 더 많은 정보를 얻으려면 PVM과 MPI FAQ에서 내용을 참조할 수 있다.

- 내가 만든 코드를 병렬처리에 알맞게 자동적으로 변경시켜주는 컴파일러가 있습니까? SGI 컴파일러처럼? [1999-05-13] 없습니다. plogic.com에서 나온 BERT라는 프로그램이 있는데 이는 여러분의 포트란 코드를 여러분이 수동으로 변경해야합니다. NAG와 Portland 그룹의 포트란 컴파일러는 여러분의 포트란 코드를 병렬처리에 알맞게 변경해줄 수는 있는데 여러분으로부터 힌트가 주어져야합니다. (HPF와 OpenMP 디렉티브의 형태로) 이러한 버젼들은 비병렬버젼들보다 속도가 전혀 빠르지 않습니다.

- 왜 사람들은 Beowulf를 사용하나요? [1999-05-13] Beowulf가 좋다고 생각하거나 전형적인 수퍼컴퓨터 가격에 1/3에서 1/10 가격으로 거의 동일한 성능을 보이기 때문이라 생각해서 입니다.

- 노드가 하나인 머신에서 보다 Beowulf에서 더욱 빠르다는 데이타 베이스를 누가 가지고 있나요? [1999-05-13] 없습니다. Oracle과 Infomix가 이러한 기능을 갖게 될지 모르지만 리눅스에서는 아직 그런 기능은 하지 않습니다.

- 사람들이 키보드-비디오-마우스 스위치를 사용하나요? [1999-05-13] 대부분의 사람들은 그렇지 않습니다. 왜냐하면 별로 필요하지 않기 때문입니다. 그들이 리눅스를 구동시킨다면 대부분이 telnet을 이용하여 접속할 것입니다. 많은 Beowulf들이 모든 노드에 비디오 카드를 갖고 있지 않습니다. 콘솔 접속은 일반적으로 부팅할 수 없을 때 필요로 합니다.

  몇몇 사람들은 이대신 시리얼 포트를 사용합니다.

- 내가 귀를 기울여야 하는 사람은 누구이고 bozo는 누구입니까? [1999-05-13] 누가 bozo인지 본저자는 모르겠습니다. 아마 본저자일지도 모르겠습니다. Don Becker, Walter B. Ligon, Putchong Uthayopas, Chrisopher Bohn, Greg Lindahl, Doug Eadline, Eugene Leitl, Gerry Creager, William Rankin이 일반적으로 그런 사람으로 알려져 있으며 도움을 기꺼이 주는 분들이다. 이 카테고리에 다른 사람도 들어있다.

  Robert G. Brown은 bozo라고 선언했지만 본저자는 그렇게 믿지 않으며 심지어 그가 본저자에게 광대얼굴을 보여줬음에도 불구하고 그렇게 생각하지 않는다. Rob Nelson 또는 bozo라고 자신을 여기지만 그 또한 실수를 하고 있는 듯하다.

- KNI, 3DNow!와 MMX 지시사항에 최적화될 수 있는 코드를 인식하는 하는 리눅스 컴파일러를 갖고 있는 사람이 있나요? [1999-05-13] 없습니다. PentiumGCC가 이에대한 지원을 하고 있습니다.

- 100대의 386을 이용해서 리눅스 클러스터를 만들 수 있나요? [1999-05-13] 한개의 Celeron 333 머신보다 느릴 것이지만 좋은 배움의 기회가 될 것입니다.

- 레드햇을 이용해야합니까? [1999-05-13] 그렇지 않습니다. 원래 Beowulf는 슬랙웨어에서 구동되어 있습니다.

  저는 지금 Extreme 리눅스 시디를 이용하고 있는데... [1999-05-13] 사용하지 마세요. 오래된 것입니다.

- Beowulf는 glibc가 필요한가요? [1999-05-13] 필요없습니다. 만일 libc5기반의 Beowulf에 특별한 응용프로그램을 깔고자 한다면 libc5가 있어야 합니다. 마찬가지로 glibc기반의 Beowulf에서 특별한 응용프로그램을 깔려면 glibc로 컴파일이 잘되고 실행이 잘되는지 확인하시기 바랍니다.

  소프트웨어를 설정할 때 각노드들에 대하여 설정을 달리 하지 않을 것을 권합니다. 그렇게 되는 경우는 머리아픈 문제를 만들어 낼 수 있습니다.

- 어떠한 컴파일러가 있나요? [1999-05-13] gcc부류, Portland Group, KAI, Fujitsu, Absoft, PentiumGCC, NAG 등이 있습니다. 컴팩은 우수하다고 알려진 베타 알파리눅스 컴파일러를 사용할 것이며 몇몇 사람들은 Digital Unix에서 그들의 응용프로그램들을 이미 컴파일 했으며 알파리눅스에서 작동시킨다.

- 무엇이 가장 중요한가요: CPU 속도, 메모리 속도, 메모리 크기, 캐쉬크기, 디스크 속도, 디스크 크기, 네트워크 대역폭? 제가 dual-CPU 머신을 사용해야합니까? 알파, 파워피시, ARM, x86중 어느것을 사용 하는게 좋습니까? Xeon을 쓰는게 좋습니까? Fast Ethernet, Gigabit Ethernet, Myrinet, SCI, FDDI중에 어느 것을 사용해야합니까? 이더넷 스위치나 허브중에서는 어느것을 사용해야 하나요? [1999-05-13] 여러분이 응용을 어떻게 하느냐에 달려 있습니다. 벤치마킹 해보고 기록하고 병목현상이 있는지 찾아내고 고치고 반복하면 된다. 몇몇 사람들은 dual-CPU 머신들이 single-CPU보다 성능면에서 앞선다고 하는데 이유는 여러분의 계산이 다른 CPU가 모든 네트워크 인터럽트를 다루는 동안 인터럽트 될 수 있기 때문이다.
- 다른 종류의 머신들 – single 프로세서, dual 프로세서, 200MHz, 400MHz 등등 – 을 가지고 Beowulf 를 만들 수 있나요? [1999-05-13]] 물론입니다. 여러분의 응용프로그램을 잘 분리하면 훨씬 수월해질 것입니다.
- 더 많은 정보를 얻으려면 어떻게 해야죠? [1999-05-13] 이 FAQ의 "긴 답변들"을 참고 하면 됩니다. <http://beowulf.org> <http://beowulf-underground.org/> <http://beowulf.gsfc.nasa.gov> <http://www.extremelinux.org> <http://www.xtrem-machines.com/x-links.html> beowulf@beowulf.gsfc.nasa.gov 메일링 리스트를 이용하면 된다. 또한 FAQ의 "보충 정보와 자료 (Supplementary information and resources)" 섹션을 참조하면 된다.
- Beowulf를 제작하는데 필요한 방법이 잘나와 있는게 있나요? [1999-05-13] <http://www.xtreme-machines.com/x-cluster-qs.html>를 살펴보면 방법을 숙지 할 수 있습니다. 또한 이곳 FAQ의 "보충 정보와 자료 (Supplementary information and resources)"의 내용중 "Docs"부분에서 docs를 살펴보면 알 수 있습니다.

제 3 절 긴 답변들

SGI의 컴파일러 처럼 Beowulf를 위해 제가 만든 코드를 자동적으로 병렬화 시켜주는 컴파일러는 없나요? [1999-05-13]

Rober G. Brown이 씀:

병렬화를 어디에서 하고 어떻게 하는지를 알려주는 BERT와 같은 툴이나 plug-in 병렬버젼과 같은 몇몇 경우를 제외하고 코드를 병렬화 하기는 쉬운 일이 아닙니다. 이는 여러분이 만든 코드가 의존성이나 시간순서를 정확히 고려하지 않았기 때문에 이러한 문제들을 해결하는 믿을만한 툴을 만들기란 너무 어려운 일이다. C와 같이 포인터기반의 언어는 거의 불가능하다고 할 수 있다.

(여러분의 코드를 안전하게 병렬화 시키는 것 말고) 두번째 문제는 제대로 병렬화 할 수 있는 부분 을 결정하는 것이다. 병렬구조에서 효과적으로 실행될 수 있는 코드는 다른 구조에서 싱글스레드 코드보다 느릴 수 있습니다.

세번째문제는 여러분의 코드를 여러분이 갖고 있는 구조 (beowulf, cluster 등등)에서 가장 효과적 으로 실행될 수 있게 코드의 정리문제입니다. 때때로 몇몇 코드들은 재 정렬을 한 경우 효율적으로 실행되는 경우가 있습니다. 하지만 이러한 배열은 시리얼 폰 노이만 코드를 작성하거나 병렬화하고자 사람에게는 대개는 쉬운 일이아니고 병렬화 하고자 하는 원본 시리얼 코드와 같이 아무것도 아닌 것이 될 것이다.

이 질문에 적절한 대답은 결국: "아니다"라는 것이다. PVM이나 MPI를 사용하는 것이 필수는 아니다. 여러분 스스로가 원본 소켓을 이용할 수 있거나 "내가 하는 방법을 모두 알거나 배우려는 관심이 열려있고 파일을 작성할 수 있는 관심"위에서 NFS를 이용할 수 있다. 하지만 이런 경우에도 대답은 역시 "아니다"이다. 아주 똑똑한 컴파일러를 가지고 재컴파일을 해도 확실히 불충분하다. 문제는 너무 복잡해서 자동화 할 수 없고 기본이 되는 시리얼 코드가 플러그인 루틴이 아닌 완전한 재정렬이 필요할 것이다. <http://noel.feld.cvut.cz/magi/soft.html>에서 많은 정보를 얻을 수 있다.

제 4 절 보충 정보와 자료

4.1 Beowulf에 유용한 소프트웨어들:

- 몇몇 소프트웨어들: [1999-05-13] <http://www.beowulf.org/software/software.html>
- PVM (Parallel Virtual Machine): [1999-05-13] <http://www.epm.ornl.gov/pvm>
- MPI (Message Passing Interface): [1999-05-13] MPICH (Argonne National Laboratory's implementation of MPI): <http://www-unix.mcs.anl.gov/mpi/mpich/index.html>

  LAM/MPI (Local Area Mutlticomputer MPI, 오하이오 수퍼컴퓨터 센터에서 개발) <http://www.mpi.nd.edu/lam/>
- Globus (Metacomputing Environment): [1999-05-13] <http://www.globus.org>

4.2 컴파일러:

- Absoft Corp. <http://www.absoft.com/> - 이 사이트는 Extreme 리눅스의 권리를 주장하고 있음! (독점) [1999-05-13] FORTRAN 77 (f77)과 FORTRAN 90 (f90)
- Portland 그룹 <www.pgroup.com/> (독점) [1999-05-13] High Performance FORTRAN (pghpf) FORTRAN 77 (pgf77) C 와 C++ (pgcc)
- Numerical Algorithm Group <http://www.nag.com/> (독점) [1999-05-13] FORTRAN 90 (f90) FORTRAN 95 (f95)
- GNU CC/egcs <http://egcs.cygnus.com/> (free Fortran-77, C, Pascal과 C++ 컴파일러) [1999-05-13]

4.3 DQS Distributed Queueing (자유로운 배치 queueing 시스템) [1999-05-13]

<http://www.scri.fsu.edu/pasco/dqs.html>

4.4 ASCI Option Red 소프트웨어: (BLAS, fast-fourier transform (역자주: FFT(Fast Fourier Transform)은 과학 계산등에 많이 쓰이는 Fourier Transform (푸리에 변환)을 알고리즘을 번형하여 좀더 효율적으로 만든 프로그램 , 하드웨어 성능감시 유틸리티, 정확성이 증가된 수학에 쓰이는 여러가지 프로그램 – 모든 프로그램들은 제한적인 라이센스에 의해 무료로 사용할 수 있다.) [1999-05-13]

<http://cs.utk.edu/~ghenry/distrib/archive.htm>

4.5 BVIEW: (Beowulf의 상태를 감시하는 소프트웨어) [1999-05-13]

<http://w272..gsfc.nasa.gov/~udaya/Public/software/bview/bview.html>

4.6 bWatch: (Beowulf의 상태를 감시하는 많은 소프트웨어 제공) [1999-05-13]

<http://www.sci.usq.edu.au/staff/jacek/bWatch/>

4.7 BPROC: (노드간의 프로세스를 시각화시켜 보여줌, 각각의 노드에 fork()를 허락하고 프로세스 이동을 가능케 하며, 모든 노드간에 kill()을 허락한다. – 현재는 pre-알파머신에 이용) [1999-05-13]

<http://www.beowulf.org/software/bproc.html>

4.8 procps를 위한 클러스터 패치: (ps와 같은 /proc기반 프로그램들을 컴파일하게 하여 모든 클러스트에게 그 내용을 전달하는 역할을 한다.) [1999-05-13]

<http://www.sc.cs.tu-bs.de/pare/results/procps.html>

4.9 SMILE 클러스터 관리 시스템: (모든 노드에 명령을 실행하고 각 노드나 노드의 집합을 멈추게 하고 노드의 상태를 모니터할 수 있음. 클러스터를 좀더 관리하기 쉽게 만들어줌.) [1999-05-13]

<http://smile.cpe.ku.ac.th/software/scms/index.html>

4.10 병렬 가상 파일시스템: (LD_PRELOAD기반의 파일시스템의 변형은 많은 디스크에 걸쳐 커다란 파일을 분산 시킬 수 있음. 커다란 데이타 집합을 고성능으로 접근하게 해줌) [1999-05-13]

<http://ece.clemson.edu/parl/pvfs/>

4.11 빠른 수학 라이브러리와 자유롭게 얻을 수 있는 빠른 수학라이브러리: (표준 수학 함수들을 빠르게 만들어줌) [1999-05-13]

<http://people.frankfurt.netsurf.de/Joachim.Wesner/>

<http://www.lsc-group.phys.uwm.edu/~www/docs/beowulf/os_updates/fastMath.html>

4.12 '복제' 노드를 설정하는 스크립트: (노드를 새롭게 Beowulf에 첨가할 때 편하게 할 수 있음) [1999-05-13]

<ftp://ftp.sci.usq.edu.au/pub/jacek/beowulf-utils/disk-less/>

4.13 클러스터에서 여러가지를 수행하게 해주는 스크립트 – 백업, 셧다운, 시스템 재시작, 각 노드에서의 명령 [1999-05-13]

<ftp://ftp.sci.usq.edu.au/pub/jacek/beowulf-utils/misc_scripts/>

4.14 BERT 77: ("포트란 코드를 자동적으로 병렬화 시켜주는 프로그램") [1999-05-13]

<http://www.plogic.com/bert.html>

4.15 펜티엄 컴파일러 그룹에서 나온 Pentium GCC, aka PGCC: (보통의 gcc보다 5%-30%의 속도 향상을 보여주는 펜티엄에 최적화된 컴파일러) [1999-05-13]

<http://goof.com/pcg/>

제 5 절 문서들:

- boilerplate 소프트웨어 설치: [1999-05-13] <http://www.phy.duke.edu/brahma/##boilerplate>
- Beowulf HOWTO: [1999-05-13] <http://www.sci.usq.edu.au/staff/jacek/beowulf/BDP/HOWTO/>
- boilerplate 소프트웨어에 관한 더많은 소프트웨어: [1999-05-13] <http://www.lsc-group.phys.uwm.edu/~www/docs/beowulf/Proto-slave/autoinstall.html> <http://www.lsc-group.phys.uwm.edu/~www/docs/beowulf/Slave/Slave_build.html>
- "Beowulf 제작방법"에 관한 정보: [1999-05-13] <http://beowulf.gsfc.nasa.gov/howto/howto.html>
- 다른 Beowulf 설치 HOWTO: [1999-05-13] <http://lcdx00.wm.lc.ehu.es/~svet/beowulf/howto.html>
- Beowulf 시스템 제작: [1999-05-13] <http://www.cacr.caltech.edu/research/beowulf/tutorial/beosoft/>
- Beowulf-Class 클러스터 제작방법 (슬라이드): [1999-05-13] <http://smile.cpe.ku.ac.th/smile/beotalk/index.htm> (부분적으로 타이어이다.)
- Beowulf 만드는 법: An Electronic Book (슬라이드): [1999-05-13] <http://smile.cpe.ku.ac.th/beowulf/index.html>
- Beowulf Installation and Administration HOWTO: [1999-05-13] <http://www.sci.usq.edu.au/staff/jacek/beowulf/BDP/BIAA-HOWTO/>

제 6 절 책들:

MIT 출판사에서 나온 "How to build a Beowulf" [1999-05-13]