Cluster Quick Start

Douglas Eadlinedeadline@plogic.com 1999년 2월 2일

양 유성yooseong@kldp.org

다음문서는 완전하지 않지만, 클러스터를 작동시키는데 일반적으로 알려 진 정보를 포함하고 있다.

목 차

제 1 절 서론 **2**

1.1 우선 읽어야할 것: . . . 2

제 2 절 개요: **3**

2.1 클러스터와 워크스테이션의 네트웍과는 무엇이 다른가? . . . 3

2.2 일반적인 셋업 . . . 3

2.3 요구사항 . . . 4

제 3 절 하드웨어 관련 **4**

3.1 네트웍 인터페이스 카드(Network Interface Cards): . . . 4

3.2 주소부여: . . . 4

3.3 게이트웨이 노드: 클러스터중 하나의 노드가 LAN에 연결된 게이트웨이 노드이기에 이 노드는 두개의 NIC를 포함해야한다. 하나의 NIC는 192.168.x.x와 같이 주소가 부여되고, 다른 하나는 내부 호스트 IP 주소를 가지고 있어야한다. 물론 라우트 방법을 이용 게이트웨이와 연결할 수 있다. (즉, 파일서버와 연결이 가능하다.) . . . 5

3.4 스위치 모형: . . . 5

3.5 직접적인 접근: . . . 5

제 4 절 보안관련: **5**

4.1 .rhosts VS hosts.equiv . . . 5

4.2 root rlogin 접근 . . . 6

4.3 root telnet 접근 . . . 6

4.4 root ftp 접근 . . . 7

제 5 절 Sample 호스트 파일 **7**

제 6 절 사용자 계정과 파일 시스템 **7**

6.1 각 노드에 /home 디렉토리 마운트 . . . 7

6.2 사용자 계정의 추가 . . . 8

제 7 절 관리: CMS **8**

제 8 절 컴파일러: 9

제 9 절 통신꾸러미들 9

9.1 PVM VS MPI 9

제 10 절 변환 소프트웨어: 10

제 11 절 Sample .cshrc 10

제 12 절 벤치마킹과 시스템 테스트 10

12.1 네트웍 퍼포먼스: netperf 10

12.2 네트웍 퍼포먼스: netpipe 10

12.3 병렬 퍼포먼스: NASA 병렬 벤치마크 10

제 13 절 이더넷 채널 본딩: 11

제 14 절 LAM입문: 11

14.1 LAM 데몬의 시작 11

14.2 데몬이 작동하는지 확인: 12

14.3 LAM 제거: 12

14.4 프로그램의 실행: 12

제 15 절 동일한 시간으로 맞춤 13

제 16 절 NIC 와 스위치 문제들 14

제 17 절 BEOWULF FAQ 14

제 1 절 서론

이 문서는 GPL(GNU PUBLIC LICENCE) Version 2(1991)에 의해 배포될 수 있다. 이 라이센스(license)에 관한 것은 <http://www.fsf.org/copyleft/gpl.html>에서 볼 수 있다.

Copyright (C) 1999 Paralogic, Inc., 115 Bethlehem PA, 18015 (<http://www.plogic.com>)

1.1 우선 읽어야할 것:

최근의 내용을 보시려면 다음 사이트를 확인해보시기 바랍니다. <http://www.xtreme-machines.com/x-cluster-qs.html>

이 문서는 모든 머신(machine)이 기본적으로 레드햇 리눅스 5.2이상으로 구동 된다고 가정한다. 또한 모든 시스템들은 자신의 하드 드라이브에서 구 동되며 클러스터 네트워크를 제어하는 스위치에 각각의 빠른 이더넷 네트워크로 연결되어있다. 머신들중 하나는 LAN(Local Area Network)에 게이트웨이 노드로서 연결되어 있다. 시스템을 구동시키는 방법에는 다른 방법들(diskless 리눅스 박스)이 있으나 이 문서는 이러한 방법들은 포함시키지 않는다.

클러스터를 형성하는 방법은 여러가지가 있으나 여기서는 한가지 방법 만을 기술할 것이다. 이문서는 완전치 않으며, 클러스터의 정상작동을 위해 많은 과정이 필요할 것이다. 실제 모든 클러스터가 다를 수 있기 때문에 클러스터를 만드는 방법을 차례대로 제시한다는 것은 매우 힘 든 일이다.

분명히 나는 이 문서에서 많은 것을 다루지 못했기 때문에 이 문서에 대해 의문이 생기면 *deadline@plogic.com* <mailto:deadline@plogic.com>으로 알려주기 바란다. 그러한 질문에 대한 해답을 엮어서 뒷부분의 FAQ에 명기할 것이다. 커널을 형성하고 컴파일하는 일, 적절한 하드웨어의 선택, 리눅스 Installation와 같은 일들은 이 문서에서는 다루지 않을 것이다.

물론 주석이나 제안은 언제나 환영이며 cluster quick start의 주제로 건설적인 주석을 보내주기바란다. i

원본 Extreme Linux CD가 오래되었기 때문에 CD에 있는 문서작업을 제외한 나머지는 사용되지 않았다.

다른 중요한 정보를 얻으려면 Beowulf-HOWTO등을 참조하기 바란다. 또한 Beowulf mailing list 모음 (<http://www.beowulf.org/listarchives/beowulf/>)에서 많은 내용을 확인할 수 있다.

마지막으로, 우리는 이 문서에서 언급된 모든 소프트웨어를 위한 RPM을 준비할 계획을 가지고 있다.

또한 Beowulf cluster를 구입하는 모든 사람들은 그냥 박스에서 머신들을 꺼내어 cluter를 구동시킨 후 <http://www.xtreme-machines.com>을 참조하기 바란다. (즉, 이 문서의 모든 사항들을 읽고 실행시켜 볼 필요는 없다는 이 야기다.)

제 2 절 개요:

2.1 클러스터와 워크스테이션의 네트웍과는 무엇이 다른가?

학문적인 관점을 떠나서, 주된 차이점은 보안, 응용소프트웨어, 관리, 부팅과정, 파일 시스템이다. 최근의 리눅스 배포본들은 높은 보안성을 유지하고 있다. 하지만 이러한 보안문제는 클러스터 컴퓨팅에 종종 장애가 되고 있다.

응용소프트웨어는 MPI(역자주: Message Passing Interface의 약자 로 클러스터 컴퓨팅에서 필요한 정보를 전달하고 받는데 사용되는 소프트웨어로 표준화 되어있음.)와 PVM(역자주: Parallel Virtual Machine의 약자로 MPI와 비슷한 일을 하나 아직 표준화가 되지 않음.)과 같은 Message Passing 시스템을 사용하고 있다.

만일 클러스터 환경에서 이러한 것이 수행되지 않으면 오직 하나의 CPU에서 작동할 것이며 특정 병렬 코드 변환이 없었다면 APACHE나 MySQL과 같은 것을 다중 CPU에서 구동시킬 수 없을 것이다.

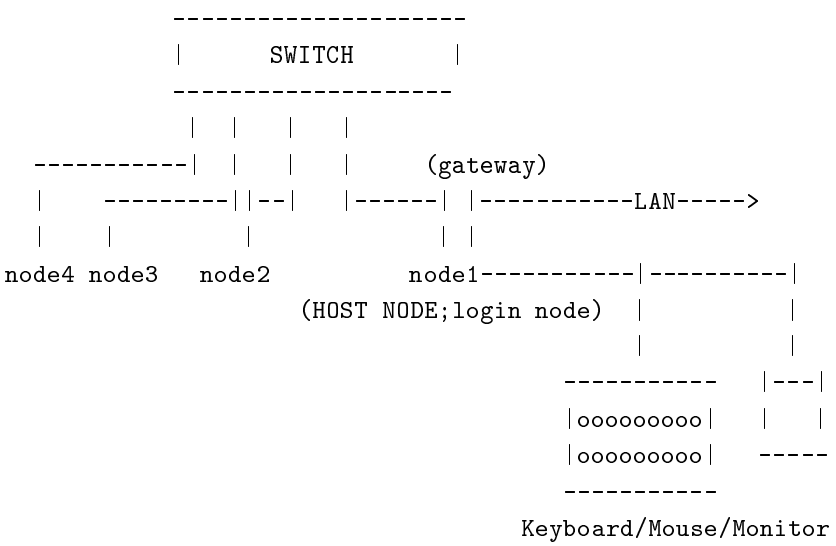
다행히 클러스터를 관리하는 도구가 있어서 예를 들어, 손으로 32 노드를 구동시키는 시간낭비를 막을 수 있다. 관리의 일부분중의 하나가 클러스터간의 보안을 완화시키는 것이다. root계정을 이용 다른 머신에 rlogin을 한다는 것은 클러스터상태에서는 아주 중요한 일이지만, 외부로 열려있는 네트웍의 경우는 아주 위험한 일이 된다.

diskless booting이 아주 매력적으로 보일지 모르지만, 각 호스트가 자신의 하드 드라이브를 부팅시키는 것 이외로 해야할 일이 많다. 여기서는 이러한 diskless booting은 생략하기로 한다.

마지막으로, 파일시스템이 정상작동을 해야하며 최소한 모든 노드간 /home 디렉토리는 공유해야한다. 다른 파일 시스템은 사용하는 유저의 요구에 따라서 공유될 수 있다.

2.2 일반적인 셋업

앞에서 언급했던 것과 같이, 이 문서는 매우 단순한 모형을 가정한다. 클러스터는 두개 혹은 그 이상의 리눅스 박스들로 구성되어 있고 이러한 박스들은 빠른 이더넷 스위치로 연결되어 있다.(허브를 이용할 수도 있지만 좀더 느리다.) 그중 하나의 노드는 LAN에 연결 되어있는 게이트웨이로 사용된다. 그러한 게이트웨이 노드는 종종 "로그인 노드"라고 불리우며 키보드와 모니터, 마우스를 갖고있다. 다른 노드들은 "머리없는(headless)" 즉, 키보드와 모니터, 마우스를 갖고 있지 않고 단지 "컴퓨터 노드"라고 불리운다. 다음의 다이어 그램은 일반적인 셋업형태를 볼 수 있다.



2.3 요구사항

클러스터를 만들고 모니터링하는데는 리눅스 관리 기술이 요구된다. 이러한 기술은 이 문서의 범위를 넘어서기때문데 다루지 않는다. 기본적으로 네트워킹에 관한 지식, 주소부여, 부팅, 커널형성, 이더넷 드라이버, NSF, 패키지 설치, 컴파일과 같은 것들이 성공적인 관리에 필요한 요소이다. 이러한 내용들에 관해서는 LDP (Linux Docu mentation Project <http://metalab.unc.edu/LDP>)에서 필요한 자료를 얻을 수 있을 것이다.(역자주: 물론 <http://kldp.org>에도 수많은 자료들을 얻을 수 있다.) 또한 Beowulf HOWTO에 관한 문서도 <http://metalab.unc.edu/LDP/HOWTO/Beowulf-HOWTO>에서 얻을 수 있다.

제 3 절 하드웨어 관련

일반적으로 이 문서는 클러스터를 형성하고 사용하는데 필요한 하드웨어 에 관한 내용만을 포함한다.

3.1 네트웍 인터페이스 카드(Network Interface Cards):

최상의 성능을 위해 Intel 또는 DEC tulip 기반(2114X)의 NIC를 사용해야 한다. 이러한 NIC는 "와이어 스피드"성능을 제공한다. 드라이버의 유무는 tulip NIC에서 특히 중요하다.

Linux-Tulip 정보에 관해서는 " <http://maximus.bmen.tulane.edu/~siekas/tulip.html> 을 살표보면 tulip 드라이버의 최신버젼을 설치하는 방법을 알 수 있다. Don Becker의 Linux와 DEC "Tulip Chip"를 보면 더욱 많은 정보를 얻을 수 있다.(<http://cesdis.gsfc.nasa.gov/linux/drivers/tulip.html>)

3.2 주소부여:

클러스터 노드간에는 정해진 IP 주소를 부여해야만 하는데 이 주소는 외부와 연결이 되어있지 말아야한다. 그리하여 192.168.x.x의 범위에서 주소가 부여되어야한다.

3.3 게이트웨이 노드: 클러스터중 하나의 노드가 LAN에 연결된 게이트웨이 노드이기에 이 노드는 두개의 NIC를 포함해야한다. 하나의 NIC는 192.168.x.x와 같이 주소가 부여되고, 다른 하나는 내부 호스트 IP 주소를 가지고 있어야한다. 물론 라우트 방법을 이용 게이트웨이와 연결할 수 있다. (즉, 파일서버와 연결이 가능하다.)

3.4 스위치 모형:

대부분의 스위치들은 IP주소가 부여되며 스위치에 telnet을 이용해서 연결 한다. 만일 노드들이 192.168.0.1, 192.168.0.2,...로 주소가 부여되었다면 스위치는 192.168.0.254와 같은 주소를 사용할 수도 있다. 스위치는 반드시 어떠한 확장된 나무형태를 가지고 있어서도 안되며 모든 포트는 100Tx-Full Duplex로 들어와야한다. 만일 이런 경우가 아니라면 NIC와 스위치가 서로 잘 작동하고 있는지 확인할 필요가 있다.

3.5 직접적인 접근:

때로는 여러분이 각 노드에 모니터와 키보드를 이용, 직접 접근할 필요가 있거나 그러고 싶은 경우 여분의 모니터와 키보드를 이용하여 접근하고자 하는 노드에 연결을 한다. 또 다른 방법으로는 KVM (Keyboard Video Mouse) 디바이스를 구입하여 한개의 키보드와 마우스, 모니터를 각 노드간 에 나누어 사용할 수 있다. push 버튼이나 hot-key를 이용 머신들간을 이동할 수 있으며 문제를 해결할 수 있다.

제 4 절 보안관련:

보안상의 변화는 오직 게이트웨이 노드에서만 이루어지는 것을 권고한다. 이렇게 함으로써 게이트웨이의 보안이 안정적이 된다.

4.1 .rhosts VS hosts.equiv

클러스터간에 패스워드를 삭제하는 방법은 두가지가 있다. /etc/hosts.equiv 파일에 입력을 하거나 home 디렉토리에의 각 계정에 .rhosts를 만드는 일이다.

.rhosts을 만드는 방법은 각 유저들의 계정에 하나씩 있기 때문에 선호 되는 방법이다. /etc/hosts.equiv는 클러스터의 각 노드마다 유지되어야 하며 이는 새로운 계정을 만들거나 없앨 때 관리자의 입장에서는 아주 복잡한 일이 된다.

.rhosts 파일의 형식은 다음과 같다:

```
#.rhost file for coyote cluster
# must be read/writable by user only!
coyote1
coyote2
coyote3
coyote4
```

hosts.equiv 파일의 형식은 다음과 같다.

```
#hosts.equiv file for coyote cluster
#node name       user name
coyote1          deadline
coyote2          deadline
coyote3          deadline
```

```
coyote4          deadline
coyote1          wgates
coyote2          wgates
coyote3          wgates
coyote4          wgates
coyote5          wgates
```

4.2 root rlogin 접근

root가 rlogin을 이용 클러스터의 각 노드에 접근하기 위해, .rhosts 파일을 각 노드의 root 디렉토리에 첨가해야한다. .rhosts 파일은 클러스터에 있는 모든 노드들을 명기해야한다. 중요한점: .rhosts 파일은 소유자만이 읽고 쓸 수 있어야 한다. ("chmod go-rwx .rhosts" : 역자주 group과 other가 .rhosts를 읽고 쓰고 실행하지 못하도록 한다.)이러한 것은 반드시 게이트웨이 노드에서는 이루어 지지 않아 야 한다. (역자주: 보안상의 문제 때문에)

덧붙여서, /etc/pam.d/rlogin 파일에 처음 두줄을 바꾸어 준다.:

```
#orginal /etc/pam.d/rlogin
auth     required       /lib/security/pam_securetty.so
auth     sufficient     /lib/security/pam_rhosts_auth.so
auth     required       /lib/security/pam_pwdb.so shadow nullock
auth     required       /lib/security/pam_nologin.so
account  required       /lib/security/pam_pwdb.so
password required       /lib/security/pam_cracklib.so
password required       /lib/security/pam_pwdb.so shadow nullock
                                                  use_authtok
session  required       /lib/security/pam_pwdb.so

#first two lines are swapped /etc/pam.d/rlogin
auth     sufficient     /lib/security/pam_rhosts_auth.so
auth     required       /lib/security/pam_securetty.so
auth     required       /lib/security/pam_pwdb.so shadow nullock
auth     required       /lib/security/pam_nologin.so
account  required       /lib/security/pam_pwdb.so
password required       /lib/security/pam_cracklib.so
password required       /lib/security/pam_pwdb.so shadow nullock
                                                  use_authtok
session  required       /lib/security/pam_pwdb.so
```

4.3 root telnet 접근

게이트웨이 노드를 제외한 모든 노드에 /etc/securetty 파일에 다음과 같은 내용을 첨가한다:

```
ttyp0
ttyp1
ttyp2
ttyp3
ttyp4
```

이러한 변화는 remote telnet을 이용 클러스터내의 어떠한 노드로 연결 이 가능케하는 것이다.

4.4 root ftp 접근

root의 ftp 접근이 필요한 시스템의 경우, /etc/ftpusers 파일에 다음과 같이 root 부분에 주석을 단다.

```
#Comment out root to allow other systems ftp access as root
#root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

제 5 절 Sample 호스트 파일

다음은 주소가 있는 스위치를 갖는 8노드 클러스터의 /etc/hosts 파일 에 관한 /etc/hosts 파일 sample이다. 각 노드는 정확한 IP 주소를 갖 고 있고 리눅스가 설치되어 있거나 네트웍을 통해 설정되어 있다고 가정한다.

```
#sample /etc/hosts
127.0.0.1       localhost       localhost.cluster
192.168.0.1     node1           node1.cluster
192.168.0.2     node2           node2.cluster
192.168.0.3     node3           node3.cluster
192.168.0.4     node4           node4.cluster
192.168.0.5     node5           node5.cluster
192.168.0.6     node6           node6.cluster
192.168.0.7     node7           node7.cluster
192.168.0.8     node8           node8.cluster
192.168.0.254   switch
```

제 6 절 사용자 계정과 파일 시스템

각 사용자는 모든 노드에 계정을 갖고 있어야한다. 효율적인 관리를 위해 호스트 노드로부터 /home이 모든 노드에 NSF를 이용 마운트 되어 있다.

6.1 각 노드에 /home 디렉토리 마운트

/home 디레토리는 각 노드에 마운트 되는 것이 좋다. 호스트 노드를 제외한 모든 노드들은 /home에 어떠한 것도 없어야 한다. (이는 노드에 사용자를 첨가해서는 안된다는 의미이다.)

/home을 첨가하기 위해서 모든 노드의 /etc/fstab에 다음과 같이 입력하여준다.(물론 호스트 파일은 제외된다.)

```
hostnode:/home          /home   nfs     bg,rw,intr      0 0
```

여기서 "hostnode:"는 여러분의 호스트 노드의 이름을 집어넣으면 된다. 만일 호스트 노드가 CDROM을 갖고 있는 경우 CDROM또한 NFS를 이용 CDROM에 접근할 수 있는데 다음과 같은 내용을 /etc/fstab에 집어 넣는다.(각 노드는 /mnt/cdrom을 갖고 있다고 본다.)

```
hostnode:/mnt/cdrom     /mnt/cdrom  nfs  noauto,ro,soft 0 0
```

다음은 호스트 노드에서 /etc/exports를 변경시켜야 한다.

```
#allow nodes to mount /home and read CDROM
/home   node1(rw) node2(rw), node3(re)
/mnt/cdrom node1(ro) node2(ro), node4(ro)
```

nfs를 다시 시작하고 마운트한다. (ps 명령을 이용하여 rpc.nfsd와 rpc.mountd의 pid를 알아낸 후 "kill -HUP pid"를 실행한다.)

모든 것이 잘 작동하는 경우, "mount /home"을 모든 노드에서 실행 해보면 /home이 마운트되어야 한다. 만일 그렇지 않으면 /var/log/ messages에서 에러를 찾아내고 마운트에 관한 man page를 이용하여 확인한다.

시스템이 시작되고 자동적으로 /home이 마운트 되지만, CDROM은 마운트 되지 않을 것이다. 하나의 노드에 CDROM을 마운트하고 싶으면, "mount /mnt/cdrom"을 실행하고나면 /mnt/cdrom에 CDROM의 내용이 보일 것이다.

만일 문제가 생기면, /var/log/messages를 확인하고 mount와 nfs 에 관한 매뉴얼을 확인한다.

6.2 사용자 계정의 추가

하나의 리눅스 워크스테이션에서 처럼 호스트 노드에 사용자 계정 을 추가한다. 각 노드에 사용자를 추가하는 가장 손쉬운 방법은 호스트 노드에 있는 /etc/passwd로 부터 사용자에 관한 내용을 복사하여 각 노드의 /etc/passwd에 복사해 넣는 것이다. 반드시 사용자와 그룹의 id가 클러스터를 통해서 모두 같은가를 확인한다.

사용자는 클러스터 전체에 걸쳐서 로그인 할 수 있다.

다른 방법은 NIS를 이용하는 것이다.(역자주: NIS에 관해서는 <http://kldp.org>에서 관계 내용을 참조하면 된다.)

제 7 절 관리: CMS

CMS (Cluster Management System)라고 불리는 꾸러미가 있다. 이는 <http://smile.cpe.ku.ac.th/software/scms/index.html>에서 정보를 얻을 수 있다. 새로운 버전에 관해서는 테스트 해볼 시간 이 없었다. 그 전에 나온 버전에 관해서는 실시간 원격 모니터링 을 제외하고 잘 작동하였다. 이는 시스템의 재시작과 중지에 관한 내용을 포함하고 있다.

제 8 절 컴파일러:

코드를 컴파일 하는 경우 g77을 포함하고 있는 egcs를 사용하기를 권장한다. Source: <http://egcs.cygnus.com/> Version: egcs-1.1.1 gzip 형태

egcs는 한번 컴파일하고 설치가 된후, /usr/local/에 존재한다. 이런식으로 사용자들은 적당한 버젼으로 자기자신의 경로를 지정 해 놓는다.(표준 gcc는 /usr/bin/에 존재한다.)

Note: 커널형성에는 gcc를 사용한다.

"gcc -V"와 "which gcc"는 여러분이 사용중인 버젼을 확인할 수 있다.

g77은 egcs의 FORTRAN 컴파일러이다.

제 9 절 통신꾸러미들

다음에 제시하는 목록보다 많은 꾸러미가 있으나 여기에서 언급 되는 꾸러미들은 가장 일반적인 것들이다. 클러스터는 지역 저장 머신(local memory machine)의 모임이다. 따라서 A 노드에서 B 노드로 통신을 하려면 네트웍을 통해서 이루어진다. 이러한 노드들간의 메세지 전달의 구조 위에 소프트웨어가 형성된다. 메세지 전달 코드는 개념적으로는 단순하지만, 그것의 작동과 디버깅은 매우 복잡할 수 있다.

여기서는 두가지의 메세지 전달 라이브러리를 소개한다:

9.1 PVM VS MPI

PVM과 MPI 모두 메세지 전달을 도와주는 적용이 간편한 소프트웨어이다. 역사적으로 볼 때 PVM이 먼저 개발 되었고 워크스테이션의 네트웍에 맞게 설계되었다.(Parallel Virtual Machine) 이는 분산 저장을 하거나 하지 않거나에 관계없이 많은 병렬 수퍼컴퓨터에 적용되어져 왔다. PVM의 여러제반 사항에 관해서는 그것을 만든이들이 주로 관여하고 있다.

MPI는 그와는 달리 많은 하드웨어 판매자에 의해 지원되고 있으며 PVM보다 더 많은 기능을 제공한다. 클러스터를 위한 버젼이 있다. MPI에 관계된 사항은 표준위원회에서 결정한다.

많은 경우 PVM과 MPI 둘 중 어느하나를 써야한다는 규칙은 없다. MPI의 경우 표준이 정해져 있기 때문에 많은 사람들이 MPI를 선호한다. 하지만 PVM도 사용되고 있다. 이 문서는 각 소스에 관한 정보를 포함하고 있다.

`MPI:`

자유롭게 사용할 수 있는 두가지의 MPI 버젼이 있다. (역자주: 그 이외에도 여러가지가 있으며 <http://kluster.kaist.ac.kr> 등에서 확인할 수 있다.)

`MPICH(역자주: MPI Chameleon의 약자):`

Source: <http://www-unix.mcs.anl.gov/mpi/mpich/> Version: mpich.tar.gz (역자주: 최근 1.2 버젼까지 나왔음.) Notes: 우리를 포함한 사람들이 리눅스 버젼에 관해 몇가지 문제점들을 제시하고 있음.

`LAM-MPI:`

Source: <http://www.mpi.nd.edu/lam/> Version: lam61.tar.gz (역자주: 최근 6.4버젼까지 나왔음.) Notes: 패치(lam61-patch.tar)을 설치한다. LAM의 경우 -c2c 모드를 사용할 경우 좋은 결과를 나타냄. (역자주: -c2c는 옵션임)

PVM:

Version: pvm3/pvm3.4.beta7.tgz Source: <http://www.epm.ornl.gov/pvm/> Notes: 많은 PVM 코드와 예제들이 나와있음.

제 10 절 변환 소프트웨어:

기존의 소프트웨어를 병렬처리에 알맞게 변환한다는 것은 시간이 오래걸리는 작업이다. 자동변환은 매우 힘들다. 자동변환은 FORTRAN 변환에만 적용되고 있다. C를 변환하는 것은 포인터 때문에 매우 힘듦.

FORTRAN 코드의 변환방법을 BERT라고 불리우며 리눅스 시스템 에서 작동한다. <http://www.plogic.com/bert.html>에서 자유롭게 얻을 수 있다.

제 11 절 Sample .cshrc

```
#Assume LAM-MPI, PVM and MPICH
setenv LAMHOME  /usr/local/lam61
setenv PVM_ROOT  /usr/local/pvm3
setenv PVM_ARCH LINUX
setenv MPIR_HOME  /usr/local/mpich

set path = {. $path}
# use egcs compilers first
set path = (/usr/local/bin $path)
set path = ($path /usr/local/pvm3/lib/LINUX)
set path = ($path /usr/local/lam61/bin)
set path = ($path /usr/local/mpich/lib/LINUX/ch_p4)
```

제 12 절 벤치마킹과 시스템 테스트

12.1 네트웍 퍼포먼스: netperf

Source: <http://www.netperf.org/netperf/NetperfPage.html>

```
Run Script:
./netperf -t UDP_STREAM -p 12865 -n 2 -1 60 -H NODE -- -s
          65535 -m 1472
./netperf -t TCP_STREAM -p 12865 -n 2 -1 60 -H NODE
```

NODE는 원격 노드 이름이다.

12.2 네트웍 퍼포먼스: netpipe

Source: <http://www.scl.ameslab.gov/Projects/Netpipe/>

12.3 병렬 퍼포먼스: NASA 병렬 벤치마크

Source: <http://www.nas.nasa.gov/NAS/NPB>

제 13 절 이더넷 채널 본딩:

채널 본딩에 관한 내용은 <http://www.beowulf.org/software/software.html>

요구사항: 시스템당 두개의 이더넷 NIC 각 채널당 두개의 허브 또는 각 채널당 두개의 스위치 또는 버추얼 LAN으로 분리될 수 있는 스위치

수행과정: (리눅스 커널 2.0.36)

1. ifenslave.c 프로그램을 다음 사이트에서 받는다. (<http://beowulf.gsfc.nasa.gov/software/bonding/html>) 35라인에 주석처리 "#include " 그리고 "gcc -Wall -Wstrict-prototypes -O ifenslave.c -o ifenslave" 를 실행한다.

2.커널패치를 한다.(<ftp://ftp.plogic.com>에서 얻은 linux-2.0.36-channel-bonding.path를 커널 패치한다.)그리고 xconfig를 실행시키고 Beowulf Channel Bonding을 가능케 한다.

3. 커널을 재형성하고 컴파일한다. 각 채널은 각기 다른 스위치 또는 허브(또는 분리된 스위치)에 있 어야 하며 두번째 네트웍 인터페이스는 IP 주소를 부여할 필요가 없다. 하지만 그 인터페이스를 분리된 네트웍으로 사용할 것이다. (채널 본딩없이) 이는 몇가지 응용에 이점이 있다.

채널 본딩을 위해 각 시스템이 root로 로그인하여 다음과 같은 명령을 실행한다.

./ifenslave -v eth0 eth1

이는 eth1과 eth0을 연결시켜 준다. 물론 eth0는 이미 시스템에서 받아들여져 있고 클러스터 네트웍으로 사용하고 있다. eth1은 단지 시스템 시작시 OS(Linux)에 의해서 감지된다.

여러분은 반드시 호스트 노드전에 모든 노드들을 슬레이브화함으로 써 이러한 작동을 시킬 수 있다. 각 노드는 다음 과정을 수행한다.

a. 창을 연다. b. 노드2에 로그인 한다. c. root계정으로 위 명령을 수행한다. d. 다른 창을 열어 노드1에 대해서 위의 명령을 수행한다.

그러면 여러분 클러스터는 채널본딩이 된 것이다. netperf나 비슷한 벤치마크를 해봄으로써 이러한 것을 시험해 볼 수 있다.

채널본딩의 멈춤은 쉬운 문제가 아니다. 우리는 이를 잘 살펴보아야 하며 채널본딩이 자동적으로 형성되고 멈추는 명령행을 입력해야한다. 하나의 채널 퍼포먼스를 저장하기 위한 가장 안전한 방법은 각 시스 템을 다시 시작하는 것이거나 네트웍 메니저(제어판의 일부)을 이용 각 인터페이스를 재시작하고 멈추게 할 수 있다.

기억할 점: 채널 본딩이 된 노드들과 그렇지 않은 노드간의 통신은 가능하지만 매우 느리다. 따라서 전체 클러스트가 채널 본딩을 해야 만한다.

제 14 절 LAM입문:

LAM은 워크스테이션의 네트웍에 알맞게 고안되었다. LAM을 실행하기 위해서 각 노드에 LAM 데몬이 시작되어야 한다. 데몬은 시험이나 디버깅 목적에 매우 적합하다. (LAM은 실시간 디버깅 정보를 dead lock 조건을 포함해서 제공할 수 있다.) 만든 코드가 작동하고 표준 소켓 인터페이스가 최대 의 속도로 실행될 수 있다. 데몬은 여전히 LAM의 시작과 멈춤에 사용된다.

14.1 LAM 데몬의 시작

여러분의 홈디렉토리에서

```
lamboot -v lamhosts
```

를 실행한다. 이는 "lamhosts"파일(여러분 머신의 목록이 적혀 있음)에 기초하여 데몬이 실행됨을 의미한다. 여기서 "-v" 옵션 은 디버그 정보를 보여주는 것이다. NOTE: LAM 데몬은 여러분이 로그아웃 상황에서도 계속 상주한다.

14.2 데몬이 작동하는지 확인:

"ps auxww|grep lam"

을 수행하면 lamd에 관계된 모든 것을 보여준다. *NOTE:* 각 사용자는 그들 자신의 LAM 데몬을 실행시킬 수 있다.

"mpitask"

의 명령 또한 데몬이 돌아가는지 확인할 수 있는 방법이다.

이 명령을 수행하면 다음과 같이 화면에 나타난다.

```
TASK (G/L)     FUNCTION    PEER|ROOT   TAG   COMM  COUNT
```

만일 수행되는 작업이 없는 경우는 위와같이 아무것도 나타나지 않는다. 만일 실행되는 작업이 있는 경우 여려분은 작업목록을 볼 수 있을 것이다.

14.3 LAM 제거:

만일 작업종료로 인해 LAM 데몬을 없애고 싶다면 "lamclean" 명령을 통해 LAM 데몬을 없앨 수 있다. 또는 "wipe lamhosts" 명령을 이용해서도 LAM 데몬을 없앨 수 있다.

14.4 프로그램의 실행:

프로그램의 실행은 MPICH와 같이 mpirun명령을 통해서 이루어 진다. 하지만 몇가지 다른 옵션이 있다.

```
mpirun -O -c 2 -s h -c2c program

-O = 일정한 환경을 가정한다. (특별한 코딩이 없다는 의미)

-c = 몇개의 실행파일이 돌아가는지를 의미하는 것으로 이 경
우는 2개임. NOTE: -c 옵션은 프로그램에 "round robin"을
"hostfile"에 특정화된 순서를 이용하여 할당.
만일 -c 옵션이 호스트 파일의 머신 개수보다 많을 경우 LAM은
호스트 파일에 지적되 있는 머신에 수행되는 작업을 과부하
시킴.

-s = 실행가능한 소스 (NFS가 필요한 것은 아님) "h"의 의미는
호스트로부터 수행가능한 소스를 얻는 다는 의미.

-c2c = "socket" 모드를 요청하는 클라이언트를 이용하라는 의미
이는 LAM을 더욱 빠르게 하지만, 데몬이 노드간의 통신에 사용
되지 않아서 디버깅이나 여러분이 수행하는 응용소프트웨어의
정보를 추적할 수 없게 된다.

<sect1>다른 정보:
<p>
```

```
여러분은 man을 이용 mpirun, lamclean, lamboot등의 정보를
얻을 수 있으며 또한
/usr/local/src/lam61/doc/mpi-quic-tut/lam_ezstart.tut
을 참조할 수 있고
<url url="http://www.mpi.nd.edu/lam/">
에서 더욱 많은 정보를 얻을 수 있다.
<p>
<sect>PVM 입문
<p>
<url url="http://netlib.org/pvm3/book/pvm-book.html">
을 확인하기 바람.
<p>
<sect>스위치 Configuration:
<verb>
BayStack 350T Main Menu

      IP Configuration...
      SNMP Configuration....
      System Characteristics...
      Switch Configuration...
      Console/Service Port Configuration...
      Spanning Tree Configuration...
      TELNET Configuration...
      Software Download...
      Display Event Log...
      Reset
      Reset to Default Settings
      Logout
```

화살표를 이용 원하는 옵션에 마킹을 하고 옵션을 선택한다.

제 15 절 동일한 시간으로 맞춤

2.0.X대의 SMP 커널에는 몇가지 문제점이 있는데 그 중하나가 시간문제이다. 이는 몇가지 방해문제들 때문에 일어나는데, 가장 좋은 해결책은 xntp를 이용해서 외부와 시간을 일치시키는 것이다. 어떠한 경우도, 여러분의 클러스터들은 시간이 모두 같아야 한다. 다음은 xntp의 사용법이다.

1. 모든 시스템의 시각을 현재시각으로 맞춘다. 2. 그 시간을 CMOS 실시간 시간으로 "clock -w"을 이용 변경한다. 3. 각 시스템에 cdrom을 마운트한다. 4. /mnt/cdrom/RedHat/RPMS로 이동한다. 5. root계정으로 "rpm -i xntp3-5.93-2.i386.rpm"을 실행한다. 6. /etc/ntp.conf를 편집한다. 모든 시스템에 대해서 다음과 같이 주석을 붙인다.

```
#multicastclient             #listen on default 224.0.1.1
#broadcastdelay 0.008
```

호스트 시스템을 제외한 나머지 시스템을 다음과 같이 편집한다.

```
server HOSTNAME # local clock
#fudge 127.127.1.0 startum 0
```

물론 여기서도 HOSTNAME은 호스트 노드의 이름이다. /etc/ntp.conf 파일을 닫고 나온다.

7. xntp를 모든 시스템에서 실행한다. "/sbin/xntpd"

여러분은 시스템을 시작할 때마다 이를 /etc/rc.d/rc.local 파일에 첨가함으로써 실행할 수 있다.

시간의 일치는 시간이 좀 걸릴 것이지만 여러분은 /var/log/ messages에서 xntp에서 나온 메세지를 확인할 수 있을 것이다.

방금 여러분이 한 것은 호스트 노드에게 xntp를 수행하고 시스템 시각을 표준으로 이용하겠다는 것이다. 결국 모든 노드의 시간은 호스트의 시간과 동일하게 될 것이다.

xntp는 시스템의 시간과 RTC(Real Time CMOS)의 시간을 일치시켜 준다. 하루에 한번씩 일치시켜주는 것이 좋을 것이다. 이는 root로서 /etc/cron.daily을 이용할 수도 있고 다음의 내용을 포함하는 "sync_clocks"라는 파일을 만들어서 수행할 수 있다.

```
#Assume ntp is running, so sync the CMOS RTC to OS system
#clock

/sbin/clock -w
```

이렇게 되면 클러스터의 모든 시각들은 일치하게 되고 호스트를 기준으로 삼을 수 있게 된다. 만일 외부시간을 기준으로 삼고 싶 다면 xntp 문서에서 방법을 찾을 수 있을 것이다.

제 16 절 NIC 와 스위치 문제들

제 17 절 BEOWULF FAQ

만일 외부시간을 기준으로 삼고 싶 다면 xntp 문서에서 방법을 찾을 수 있을 것이다.

1. extreme 리눅스 CD와 일반 RedHat 배포판의 차이는? 2. extreme 리눅스를 가지고 클러스터을 만들 수 있으며 Oracle 8을 실행할 수 있나?

이문서는 GPL Version 2에 의해 배포된다. 이 라이센스에 관해서는 <http://www.fsf.org/copyleft/gpl.html>

Copyright (C) 1999 Paralogic, Inc., 115 Bethlehem PA, 18015 (<http://www.plogic.com>)