Vine Linux による PC Cluster の構築

幸谷智紀

平成 15 年 3 月 20 日

1　初めに

本レポートは，著者の職場で PC Cluster(cs-pccluster) を構築した際に作成したメモに基づくものである。ネットワークは Ethernet(100BASE-TX)，PC は数年前の Pentium III 1GHz と Celeron 1GHz のもの，OS は Vine Linux[12] をインストールし，rsh 上で mpich[4] を使用した。このような環境であるため，パフォーマンスの点でも拡張性の点でもあまり自慢できるものではない。が，一般的かつ安価なもののみ用いているため，UNIX の設定運用の基礎知識と，複数台の PC があれば，誰にでも構築できる内容になっている。分散処理の入門学習としてあれこれ弄ってみると，いろいろ見えてくるものがあるだろう[1]。Vine Linux(2.6r1) に依存した設定手順も多いが，NIS/NFS, rsh, mpich のためのものであることがきちんと理解できれば，本レポートで示したものは他の UNIX or UNIX compatible OS にも応用可能である。… と偉そうに語っているが，設定の大部分は超並列研究会 [11] で公開されている文書群を参考にさせて頂いた。ここで感謝の意を表したい。

なお，当然のことであるが，本レポートの手順に従った結果被った損害等について著者は一切関知しない。全ては自己責任 (at your own risk) で行って頂きたい。

2　セットアップ手順

構築した PC クラスタ (以下，cs-pccluster(図 1) と称する) は，以下の手順を経て完成した。この文書で述べるまでもない一般的な手続きについては説明を略し，肝要な部分のみを抜粋して述べることにする。

cs-pccluster は NIS/NFS Server となる親 (cs-southpole) マシンと，NIS/NFS Client となる子 (cs-room443-*) マシンから成る。マシン構成の詳細については後述する。

1. [2.1 節] ハードウェアの準備と TCP/IP の設定
2. [2.2 節] 親 (NFS/NIS Server) に以下のソフトウェアをインストール
   (a) 最新の gcc/g++/g77
   (b) BLAS(+ATLAS)
   (c) LAPACK
3. [2.3 節] 親及び子 (NFS/NIS Clients) の rsh を有効にする

[1] ちなみに，著者も分散処理については素人の域を出ない。

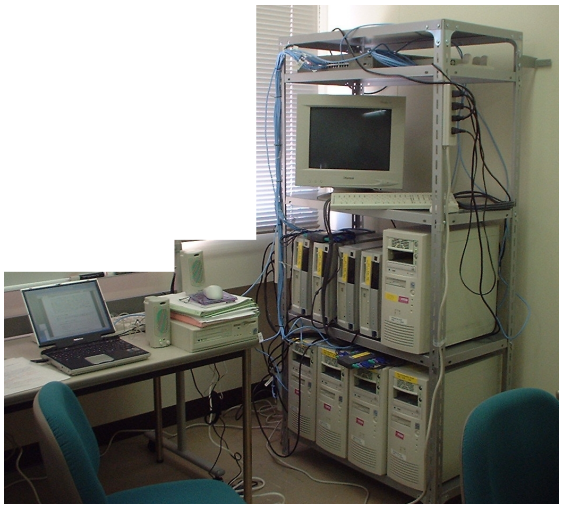

図 1: CS-PCCLUSTER 完成写真

4. [2.4 節] 子の/home, /usr を親のものに NFS mount する
5. [2.5 節] 親は NIS Server/NIS Client としての，子は NIS Client としての設定をする
6. [2.6 節] 親および子において rsh の設定とテスト
7. [2.7 節] 全ての子マシンを NFS/NIS Client にする
8. [2.8 節] 親に以下のソフトウェアをインストール
   (a) mpich(設定と mpirun のテスト)
   (b) MPIBLACS(MPI 用の BLACS)
   (c) ScaLAPACK

本稿の最後に，ScaLAPACK のテストプログラム実行結果と，姫野ベンチマークの結果を掲載しておく。

2.1 ハードウェアの準備と TCP/IP の設定

まず，cs-pccluster を構成する各クラスタマシンのスペックを表 1 に示す。これらのマシン全てに，Vine Linux 2.6r1(kernel Version: 2.4.19-0vl11) をフルインストールしてある。

ハードウェアの構成を図 2 に，TCP/IP の構成を 3 に示す。これ以外にも，ラックには納めないが，cs-room443-01〜05 までの 5 台のマシンも必要に応じて cs-pccluster に組み込めるようにしておく。これらは cs-southpole/cs-room443-b0x と同じスペックである。従って，普段はラックに収まった 9 台のマシンを使い，更にマシンパワーが必要になれば，5 台追加して計 14 台が使用可能となる。

cs-southpole の`/etc/hosts` は以下の通り。

表 1: PC のスペック表

	親: cs-southpole 子 1: cs-room443-b01〜b04	子 2: cs-room443-s01〜s04
CPU	Pentium III 1GHz Coppermine	Celeron 1GHz
L1 Cache (KB)	16	16/16(L1 D)
L2 Cache (KB)	256	256
RAM(MB)	512	128
IDE HDD(GB)	40	40
100BASE-TX NIC	3COM 3c905	RealTek RTL-8139B

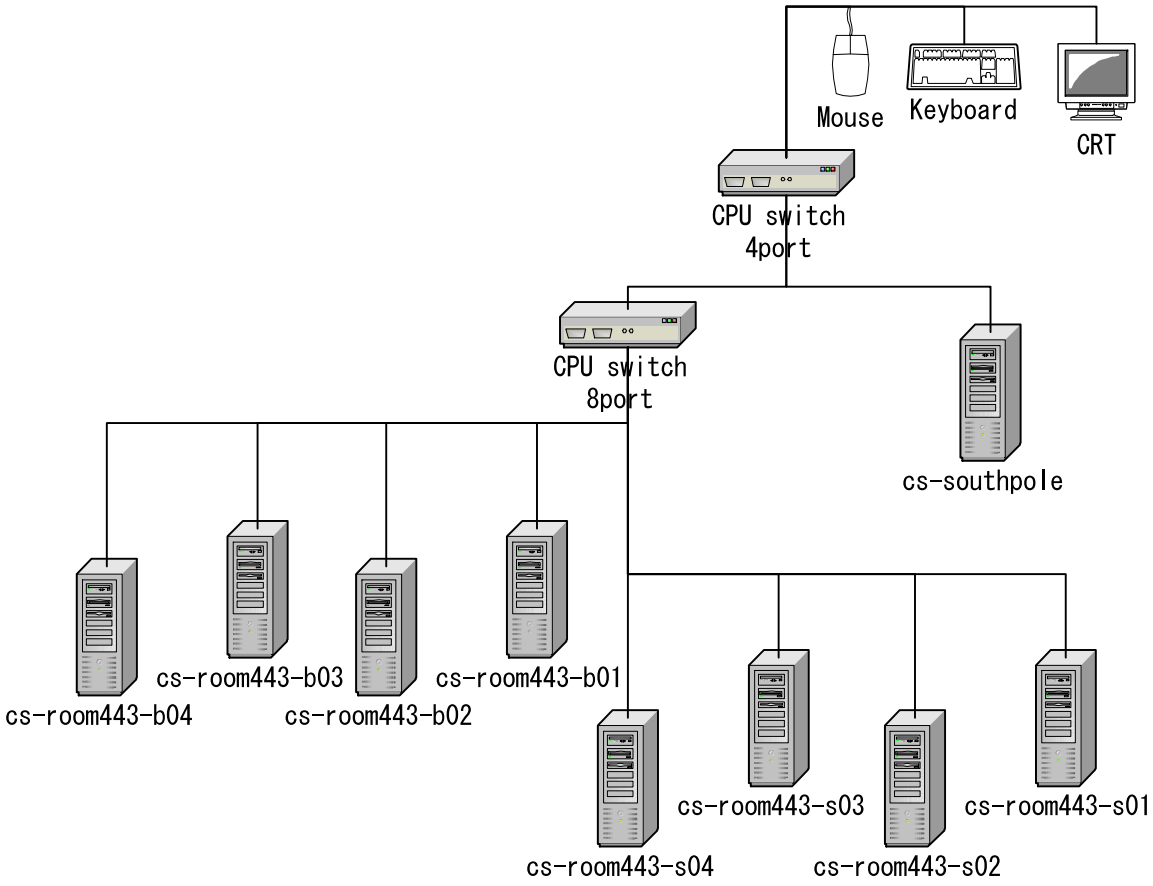

図 2: CPU Switch の配置

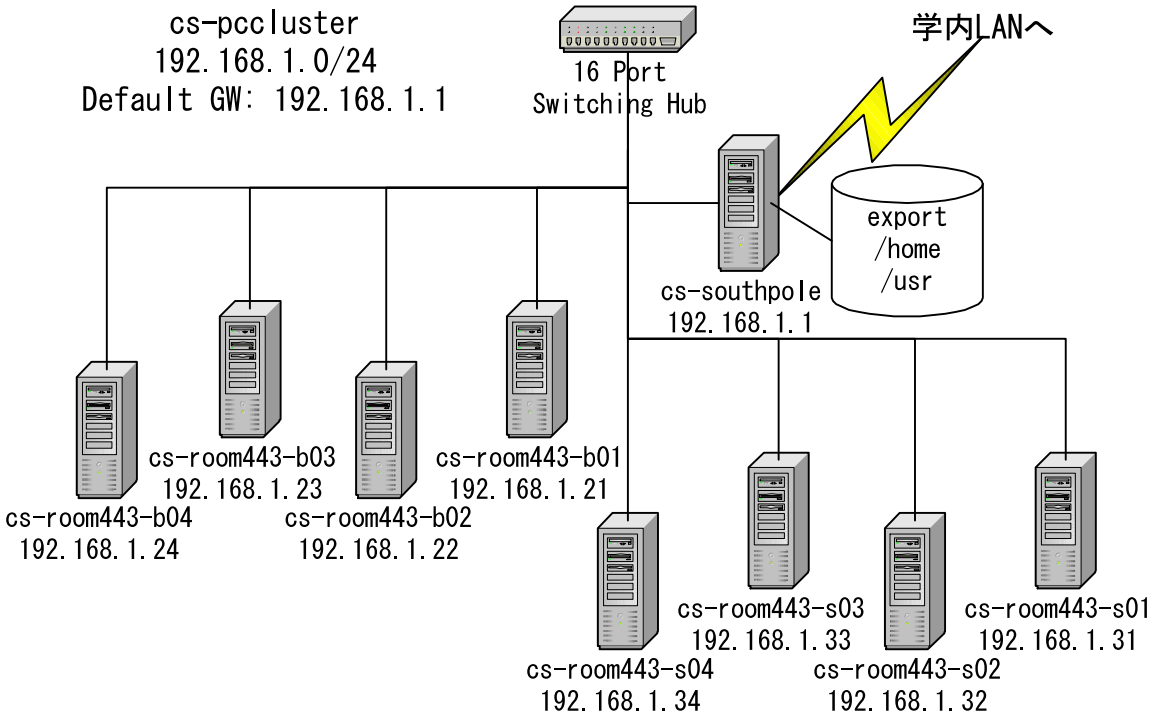

図 3: Ethernet と IP ネットワーク

```
#
127.0.0.1       localhost
#
192.168.1.21    cs-room443-s01
192.168.1.22    cs-room443-s02
192.168.1.23    cs-room443-s03
192.168.1.24    cs-room443-s04
#
192.168.1.31    cs-room443-b01
192.168.1.32    cs-room443-b02
192.168.1.33    cs-room443-b03
192.168.1.34    cs-room443-b04
# xxx.xx.xxx.xx  out-cs-southpole
192.168.1.1     cs-southpole
192.168.1.11    cs-room443-01
192.168.1.12    cs-room443-02
192.168.1.13    cs-room443-03
192.168.1.14    cs-room443-04
192.168.1.15    cs-room443-05
```

この時点で，全てのマシンに Vine がインストールされ，この図 2, 3 通りに配線され，TCP/IP

環境が構築されているとする。インストール時には，各マシンにおけるローカルな root のパスワードと，ローカルユーザ (仮に “user01”としておく) を登録しておくこと。

最後に，全てのマシンに telnet 出来ることを確認する。

```
[user01@cs-southpole user01]$ telnet cs-room443-s01
Trying 192.168.1.21...
Connected to cs-room443-s01.
Escape character is '^]'.

Vine Linux 2.6 (La Fleur de Bouard)
Kernel 2.4.19-0vl11 on an i686
login: user01
Password:
Last login: Sat Feb 22 14:33:43 from 192.168.1.1
[user01@cs-room443-s01 user01]$ su
Password:
[root@cs-room443-s01 user01]# exit
exit
[user01@cs-room443-s01 user01]$ logout
[user01@cs-southpole user01]
```

全てのクライアントマシンの/etc/hosts には，NIS/NFS サーバとなる cs-southpole のエントリをあらかじめ追加しておく。

```
192.168.1.1 cs-southpole
```

2.2 ソフトウェアのインストール 1

Netlib[5] から LAPACK(lapack.tgz) をダウンロードし解凍。

```
[user01@cs-www pool]$ tar zxvf lapack.tgz
LAPACK/
LAPACK/SRC/

LAPACK/Makefile
LAPACK/README
LAPACK/latape
LAPACK/make.inc
LAPACK/BLAS/
LAPACK/BLAS/SRC/

LAPACK/BLAS/TESTING/

LAPACK/INSTALL/
LAPACK/INSTALL/Makefile
```

```
LAPACK/INSTALL/make.inc.ALPHA
LAPACK/INSTALL/make.inc.HPPA
LAPACK/INSTALL/make.inc.IRIX64
LAPACK/INSTALL/make.inc.LINUX
LAPACK/INSTALL/make.inc.O2K
LAPACK/INSTALL/make.inc.RS6K
LAPACK/INSTALL/make.inc.SGI5
LAPACK/INSTALL/make.inc.SUN4
LAPACK/INSTALL/make.inc.SUN4SOL2
LAPACK/INSTALL/make.inc.pghpf

LAPACK/TESTING/
LAPACK/TESTING/EIG/
LAPACK/TESTING/LIN/
LAPACK/TESTING/MATGEN/
LAPACK/TIMING/
LAPACK/TIMING/EIG/

[user01@cs-www pool]$ cd LAPACK
[user01@cs-www LAPACK]$ ls
BLAS  INSTALL  Makefile  README  SRC  TESTING  TIMING  latape  make.inc
[user01@cs-www LAPACK]$ ls INSTALL
Makefile       lawn81.tex       make.inc.RS6K      second.f.RS6K
dlamch.f       lsame.f          make.inc.SGI5      secondtst.f
dlamchtst.f    lsametst.f       make.inc.SUN4      slamch.f
dsecnd.f       make.inc.ALPHA   make.inc.SUN4SOL2  slamchtst.f
dsecnd.f.RS6K  make.inc.HPPA    make.inc.pghpf     tstiee.f
dsecndtst.f    make.inc.IRIX64  org2.ps
lawn81.pdf     make.inc.LINUX   psfig.tex
lawn81.ps      make.inc.O2K     second.f
[user01@cs-www LAPACK]$ cp INSTALL/make.inc.LINUX ./make.inc
```

まず，BLAS のコンパイル。

```
[user01@cs-www LAPACK]$ cd BLAS/SRC
[user01@cs-www SRC]$ ls
Makefile  chpr.f    daxpy.f   dsymv.f   izamax.f  sspr.f    zaxpy.f   zhpr.f
caxpy.f   chpr2.f   dcabs1.f  dsyr.f    lsame.f   sspr2.f   zcopy.f   zhpr2.f
ccopy.f   crotg.f   dcopy.f   dsyr2.f   sasum.f   sswap.f   zdotc.f   zrotg.f
cdotc.f   cscal.f   ddot.f    dsyr2k.f  saxpy.f   ssymm.f   zdotu.f   zscal.f
cdotu.f   csscal.f  dgbmv.f   dsyrk.f   scasum.f  ssymv.f   zdscal.f  zswap.f
cgbmv.f   cswap.f   dgemm.f   dtbmv.f   scnrm2.f  ssyr.f    zgbmv.f   zsymm.f
cgemm.f   csymm.f   dgemv.f   dtbsv.f   scopy.f   ssyr2.f   zgemm.f   zsyr2k.f
cgemv.f   csyr2k.f  dger.f    dtpmv.f   sdot.f    ssyr2k.f  zgemv.f   zsyrk.f
```

```
cgerc.f   csyrk.f   dnrm2.f   dtpsv.f   sgbmv.f   ssyrk.f   zgerc.f   ztbmv.f
cgeru.f   ctbmv.f   drot.f    dtrmm.f   sgemm.f   stbmv.f   zgeru.f   ztbsv.f
chbmv.f   ctbsv.f   drotg.f   dtrmv.f   sgemv.f   stbsv.f   zhbmv.f   ztpmv.f
chemm.f   ctpmv.f   dsbmv.f   dtrsm.f   sger.f    stpmv.f   zhemm.f   ztpsv.f
chemv.f   ctpsv.f   dscal.f   dtrsv.f   snrm2.f   stpsv.f   zhemv.f   ztrmm.f
cher.f    ctrmm.f   dspmv.f   dzasum.f  srot.f    strmm.f   zher.f    ztrmv.f
cher2.f   ctrmv.f   dspr.f    dznrm2.f  srotg.f   strmv.f   zher2.f   ztrsm.f
cher2k.f  ctrsm.f   dspr2.f   icamax.f  ssbmv.f   strsm.f   zher2k.f  ztrsv.f
cherk.f   ctrsv.f   dswap.f   idamax.f  sscal.f   strsv.f   zherk.f
chpmv.f   dasum.f   dsymm.f   isamax.f  sspmv.f   xerbla.f  zhpmv.f
[user01@cs-www SRC]$ make
g77  -funroll-all-loops -fno-f2c -O3 -c isamax.f
(略)
g77  -funroll-all-loops -fno-f2c -O3 -c cher2k.f
ar cr ../../blas_LINUX.a scasum.o scnrm2.o icamax.o caxpy.o ccopy.o cdotc.o cdot
u.o csscal.o crotg.o cscal.o cswap.o isamax.o sasum.o saxpy.o scopy.o snrm2.o ss
cal.o \
lsame.o xerbla.o cgemv.o cgbmv.o chemv.o chbmv.o chpmv.o ctrmv.o ctbmv.o ctpmv.o
 ctrsv.o ctbsv.o ctpsv.o cgerc.o cgeru.o cher.o chpr.o cher2.o chpr2.o cgemm.o c
symm.o csyrk.o csyr2k.o ctrmm.o ctrsm.o chemm.o cherk.o cher2k.o
ranlib ../../blas_LINUX.a
g77  -funroll-all-loops -fno-f2c -O3 -c dcabs1.f
(略)
g77  -funroll-all-loops -fno-f2c -O3 -c zher2k.f
ar cr ../../blas_LINUX.a dcabs1.o dzasum.o dznrm2.o izamax.o zaxpy.o zcopy.o zdo
tc.o zdotu.o zdscal.o zrotg.o zscal.o zswap.o idamax.o dasum.o daxpy.o dcopy.o d
nrm2.o dscal.o \
lsame.o xerbla.o zgemv.o zgbmv.o zhemv.o zhbmv.o zhpmv.o ztrmv.o ztbmv.o ztpmv.o
 ztrsv.o ztbsv.o ztpsv.o zgerc.o zgeru.o zher.o zhpr.o zher2.o zhpr2.o zgemm.o z
symm.o zsyrk.o zsyr2k.o ztrmm.o ztrsm.o zhemm.o zherk.o zher2k.o
ranlib ../../blas_LINUX.a
[user01@cs-www SRC]$ cd ../../
```

次に LAPACK 本体の構築。

```
[user01@cs-www LAPACK]$ make
( cd INSTALL; make; ./testlsame; ./testslamch; \
  ./testdlamch; ./testsecond; ./testdsecnd; \
  cp lsame.f ../BLAS/SRC/; cp lsame.f ../SRC; \
  cp slamch.f ../SRC/; cp dlamch.f ../SRC/; \
  cp second.f ../SRC/; cp dsecnd.f ../SRC/ )
make[1]: 入ります ディレクトリ ‘/home/user01/pool/LAPACK/INSTALL’
g77  -funroll-all-loops -fno-f2c -O3 -c lsame.f
(略)
```

```
g77  -o testieee tstiee.o
make[1]: 出ます ディレクトリ ‘/home/user01/pool/LAPACK/INSTALL’
  ASCII character set
  Tests completed
  Epsilon                      =   5.96046448E-08
  Safe minimum                 =   1.17549435E-38
  Base                         =   2.
  Precision                    =   1.1920929E-07
  Number of digits in mantissa =   24.
  Rounding mode                =   1.
  Minimum exponent             =  -125.
  Underflow threshold          =   1.17549435E-38
  Largest exponent             =   128.
  Overflow threshold           =   3.40282347E+38
  Reciprocal of safe minimum   =   8.50705917E+37
  Epsilon                      =   1.11022302E-16
  Safe minimum                 =   2.22507386E-308
  Base                         =   2.
  Precision                    =   2.22044605E-16
  Number of digits in mantissa =   53.
  Rounding mode                =   1.
  Minimum exponent             =  -1021.
  Underflow threshold          =   2.22507386E-308
  Largest exponent             =   1024.
  Overflow threshold           =   1.79769313E+308
  Reciprocal of safe minimum   =   4.49423284E+307
 Time for 1,000,000 SAXPY ops  =   0.00     seconds
 *** Error:  Time for operations was zero
 Including SECOND, time        =  0.100E-01 seconds
 Average time for SECOND       =  0.200E-02 milliseconds
 Time for 1,000,000 DAXPY ops  =  0.100E-01 seconds
 DAXPY performance rate        =   100.     mflops
 Including DSECND, time        =  0.100E-01 seconds
 Average time for DSECND       =   0.00     milliseconds
 Equivalent floating point ops =   0.00     ops
( cd SRC; make )
make[1]: 入ります ディレクトリ ‘/home/user01/pool/LAPACK/SRC’
g77  -funroll-all-loops -fno-f2c -O3 -c sgbbrd.f
(略)
[user01@cs-www LAPACK]$
```

blas_LINUX.a, lapack_LINUX.a, tmglib_LINUX.a ができあがっていれば完成。

```
[user01@cs-www LAPACK]$ ls
```

```
BLAS     Makefile  SRC      TIMING        lapack_LINUX.a  make.inc
INSTALL  README    TESTING  blas_LINUX.a  latape          tmglib_LINUX.a
```

2.3 rsh の有効化

/etc/inetd.conf を編集し，rsh を有効にしておく。

```
shell   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd
```

inetd の再起動

```
[root@cs-room443-03 /etc]# /sbin/service inet restart
Stopping INET services:                                    [  OK  ]
Starting INET services:                                    [  OK  ]
[root@cs-room443-03 /etc]#
```

2.4 NFS の設定

親の/etc/exports を次のように変更する。

```
/home   192.168.1.*(rw)
/usr    192.168.1.*(rw)
```

変更したら，/usr/sbin/exportfs しておく。

```
[root@cs-southpole user01]# /usr/sbin/exportfs
/home           192.168.1.*
/usr            192.168.1.*
```

子の nfs サービスも on にしておく

```
[root@cs-room443-01 user01]# /sbin/chkconfig --list
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
nfslock         0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@cs-room443-01 user01]# /sbin/chkconfig nfslock on
[root@cs-room443-01 user01]# /sbin/chkconfig nfs on
[root@cs-room443-01 user01]$ /sbin/chkconfig --list
nfs             0:off   1:off   2:off   3:on    4:on    5:on    6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

子の/etc/fstab に以下の行を追加。

```
cs-southpole:/home      /home                   nfs     defaults        0 0
cs-southpole:/usr       /usr                    nfs     defaults        0 0
```

が，これではマウントできなかった時が悲惨である。現在は，JF 文書を参考にして以下のように変更した。

```
[user01@cs-room443-s01 user01]$ cat /etc/fstab
# append for cs-pccluster
cs-southpole:/home      /home                   nfs     rw,hard,intr    0 0
cs-southpole:/usr       /usr                    nfs     rw,hard,intr    0 0
```

2.5 NIS の設定

portmap, ypserv, ypbind, yppasswdd daemon を有効にする。

```
[user01@cs-southpole user01]# /sbin/chkconfig --list
ypbind          0:off   1:off   2:off   3:off   4:off   5:off   6:off
yppasswdd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypserv          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@cs-southpole user01]# /sbin/chkconfig ypserv on
[root@cs-southpole user01]# /sbin/chkconfig ypbind on
[root@cs-southpole user01]# /sbin/chkconfig yppasswdd on
[root@cs-southpole user01]# /sbin/chkconfig portmap on
[root@cs-southpole user01]# /sbin/chkconfig --list
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
ypbind          0:off   1:off   2:off   3:on    4:on    5:on    6:off
yppasswdd       0:off   1:off   2:off   3:on    4:on    5:on    6:off
ypserv          0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

/var/yp/securenets の設定を行い，Private Address(192.168.1.0/24) の subnet 内のマシンにのみ，NIS Client 権限を持たせる。

```
# Always allow access for localhost
255.0.0.0       127.0.0.0

# This line gives access to everybody. PLEASE ADJUST!
#0.0.0.0                0.0.0.0
255.255.255.0           192.168.1.0
```

/etc/yp.conf に以下の行を追加。

```
domain cs-pccluster server cs-southpole
```

/etc/sysconfig/network に以下の行を追加。NIS ドメインは “cs-pccluster” とする。

```
NISDOMAIN="cs-pccluster"
```

NIS ドメインの確認手順は以下の通り。

```
[root@cs-southpole yp]# domainname
(none)
[root@cs-southpole yp]# domainname cs-pccluster
[root@cs-southpole yp]# domainname
cs-pccluster
```

以上の手順が完了したら，/var/yp に移動して make する。

```
[root@cs-southpole yp]# make
gmake[1]: 入ります ディレクトリ ‘/var/yp/cs-pccluster’
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating mail.aliases...
gmake[1]: 出ます ディレクトリ ‘/var/yp/cs-pccluster’
[root@cs-southpole yp]#
```

その後，ypserv(NIS Server) と ypbind(NIS Client) を再起動。cs-southpole は NIS Server でもあり，NIS Client でもある。

```
[root@cs-southpole yp]# /sbin/service ypserv restart
Stopping YP server services:                               [  OK  ]
Starting YP server services:                               [  OK  ]
[root@cs-southpole yp]# /sbin/service ypbind restart
Shutting down NIS services:                                [失敗]
Binding to the NIS domain...                               [  OK  ]
Listening for an NIS domain server: cs-southpole
```

ypcat コマンドで，NIS の動作状況を確認する。

```
[root@cs-southpole yp]# ypcat passwd
user01:$1$h797QvRf$glek2ChLDqWXmr6G8v/A31:500:500:Tomonori Kouya:/home/user01:/bin/bash
[user01@cs-southpole user01]$ ypcat hosts
133.88.121.78   out-cs-southpole
127.0.0.1       localhost
192.168.1.1     cs-southpole
192.168.1.15    cs-room443-05
192.168.1.14    cs-room443-04
192.168.1.13    cs-room443-03
192.168.1.12    cs-room443-02
192.168.1.11    cs-room443-01
```

これらが全て reboot 後も有効になっているかどうかを確認する。

```
[user01@cs-southpole user01]$ domainname
cs-pccluster
[user01@cs-southpole user01]$ ypcat passwd
user01:$1$h797QvRf$glek2ChLDqWXmr6G8v/A31:500:500:Tomonori Kouya:/home/user01:/bin/bash
[user01@cs-southpole user01]$ ypcat hosts
133.88.121.78   out-cs-southpole
127.0.0.1       localhost
192.168.1.1     cs-southpole
192.168.1.15    cs-room443-05
192.168.1.14    cs-room443-04
192.168.1.13    cs-room443-03
192.168.1.12    cs-room443-02
192.168.1.11    cs-room443-01
```

子マシンについては portmapper と ypbind と yppasswdd のみを有効化する。この手順は略す。

2.6 rsh の有効化

親および子全てに/etc/hosts.equiv を作成して

```
cs-southpole    user01
cs-room443-01   user01
cs-room443-02   user01
cs-room443-03   user01
cs-room443-04   user01
cs-room443-05   user01
```

とするか/home/user01/.rhosts を作成して上記の内容にするか。/home を NFS マウントする必要があるので，後者の方が better。新たにユーザを追加する際には，このファイルを新規ユーザ作成時にホームディレクトリにコピーしておけばよい。

2.7 残りの子の設定

以上が，親と子との間で確認できれば，後は他の子を設定すればよい。順番は以下の通り。

1. 子マシンに telnet して，以下のサービスを on にする
   (a) portmap
   (b) nfs
   (c) nfslock(不要かな？)
   (d) ypbind
   (e) yppasswd(不要かな？)

2. /etc/inetd.conf を編集し，rsh を有効にする
3. /etc/yp.conf を設定

```
domain cs-pccluster server cs-southpole
```

4. /etc/sysconfig/network に次の行を追加

```
NISDOMAIN=cs-pccluster
```

5. /etc/fstab に次の行を追加

```
cs-southpole:/home      /home                   nfs     defaults        0 0
cs-southpole:/usr       /usr                    nfs     defaults        0 0
```

6. reboot して再立ち上げ
7. もう一度ログインして以下の確認作業を行う

```
[user01@cs-room443-02 user01]$ domainname
cs-pccluster
[user01@cs-room443-02 user01]$ ypcat hosts
133.88.121.78   out-cs-southpole
127.0.0.1       localhost
192.168.1.1     cs-southpole
192.168.1.15    cs-room443-05
192.168.1.14    cs-room443-04
192.168.1.13    cs-room443-03
192.168.1.12    cs-room443-02
192.168.1.11    cs-room443-01
[user01@cs-room443-02 user01]$ ypcat passwd
user01:$1$h797QvRf$glek2ChLDqWXmr6G8v/A31:500:500:Tomonori Kouya:/home/user01:/bin/bash
[user01@cs-room443-02 user01]$ rsh cs-southpole ls
Xrootenv.0
pool
rhosts
rpm
[user01@cs-room443-02 user01]$ rsh cs-southpole cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=cs-southpole
GATEWAY=133.88.120.11
NISDOMAIN=cs-pccluster
[user01@cs-room443-02 user01]$
```

2.8 MPICH, ScaLAPACK のインストール

まず，MPICH[4] をダウンロードし，/usr/local にインストールする。

```
[user01@cs-southpole pool]$ tar zxvf mpich-1.2.5-1a.tar.gz
mpich-1.2.5/
mpich-1.2.5/bin/
mpich-1.2.5/bin/tarch
mpich-1.2.5/bin/tdevice
mpich-1.2.5/www/
mpich-1.2.5/www/www1/
mpich-1.2.5/www/www1/mpirun.html
(略)
mpich-1.2.5/www.index
[user01@cs-southpole pool]$ cd mpich-1.2.5
[user01@cs-southpole pool]$ ./configure --prefix=/usr/local
[user01@cs-southpole pool]$ make
[user01@cs-southpole pool]$ su
[user01@cs-southpole pool]$ make install
$
```

この時点で rsh が有効になっていれば，ch_p4 アーキテクチャが使用可能となっている。

最後に，cs-pccluster で使用するマシンリストを/usr/local/share/machines.LINUX に追加する。SMP マシンが混じっているようなら CPU 数も記入する。ここでは全て 1CPU マシンなので略してある。

```
# Change this file to contain the machines that you want to use
# to run MPI jobs on.  The format is one host name per line, with either
#    hostname
# or
#    hostname:n
# where n is the number of processors in an SMP.  The hostname should
# be the same as the result from the command "hostname"
cs-southpole
cs-room443-b01
cs-room443-b02
cs-room443-b03
cs-room443-b04
cs-room443-s01
cs-room443-s02
cs-room443-s03
cs-room443-s04
cs-room443-01
cs-room443-02
cs-room443-03
```

```
cs-room443-04
cs-room443-05
```

インストールが終わったら，テストプログラムをコンパイルしてチェックしてみる。

```
[user01@cs-southpole mpich-1.2.5]$ cd examples
[user01@cs-southpole examples]$ ls
Makefile  Makefile.in  README  basic/  io@  nt/  perftest/  test/
[user01@cs-southpole examples]$ cd basic
[user01@cs-southpole basic]$ ls
Makefile     README  cpilog.c  hello++.cc*  pcp.c       prm.c     systest.c
Makefile.in  cpi.c   fpi.f     iotest.c     pi3f90.f90  srtest.c  unsafe.c
[user01@cs-southpole basic]$ make
/home/user01/pool/mpich-1.2.5/bin/mpicc   -c cpi.c
(略)
[user01@cs-southpole basic]$
[user01@cs-southpole examples]$
```

数値積分のサンプルプログラムを並列実行してみる。

```
[user01@cs-southpole basic]$ mpirun -np 1 ./cpi
Process 0 of 1 on cs-southpole
pi is approximately 3.1415926544231341, Error is 0.0000000008333410
wall clock time = 0.000976
[user01@cs-southpole basic]$ mpirun -np 2 ./cpi
Process 0 of 2 on cs-southpole
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.003419
Process 1 of 2 on cs-room443-b01
[user01@cs-southpole basic]$ mpirun -np 4 ./cpi
Process 0 of 4 on cs-southpole
pi is approximately 3.1415926544231239, Error is 0.0000000008333307
wall clock time = 0.075359
Process 1 of 4 on cs-room443-b01
Process 2 of 4 on cs-room443-b02
Process 3 of 4 on cs-room443-b03
[user01@cs-southpole basic]$ mpirun -np 8 ./cpi
Process 0 of 8 on cs-southpole
pi is approximately 3.1415926544231247, Error is 0.0000000008333316
wall clock time = 0.316260
Process 4 of 8 on cs-room443-b04
Process 2 of 8 on cs-room443-b02
Process 6 of 8 on cs-room443-s02
Process 1 of 8 on cs-room443-b01
Process 3 of 8 on cs-room443-b03
```

```
Process 5 of 8 on cs-room443-s01
Process 7 of 8 on cs-room443-s03
[user01@cs-southpole basic]$ mpirun -np 9 ./cpi
Process 0 of 9 on cs-southpole
pi is approximately 3.1415926544231256, Error is 0.0000000008333325
wall clock time = 0.237792
Process 4 of 9 on cs-room443-b04
Process 2 of 9 on cs-room443-b02
Process 1 of 9 on cs-room443-b01
Process 5 of 9 on cs-room443-s01
Process 3 of 9 on cs-room443-b03
Process 6 of 9 on cs-room443-s02
Process 8 of 9 on cs-room443-s04
Process 7 of 9 on cs-room443-s03
[user01@cs-southpole basic]$
```

ここでちょっと不具合が出た。Ethernet が 2 枚刺さっている cs-southople マシンで実行すると

```
[user01@cs-southpole basic]$ mpirun -np 6 ./cpi
Process 0 of 6 on out-cs-southpole
pi is approximately 3.1415926544231239, Error is 0.0000000008333307
wall clock time = 0.001517
Process 1 of 6 on out-cs-southpole
Process 3 of 6 on cs-room443-02
Process 5 of 6 on cs-room443-04
Process 2 of 6 on cs-room443-01
Process 4 of 6 on cs-room443-03
[user01@cs-southpole basic]$
```

というように，1 マシンで 2 process 走ってしまう。1 枚刺しのマシンだと

```
[user01@cs-room443-05 basic]$ mpirun -np 6 ./cpi
Process 0 of 6 on cs-room443-05
pi is approximately 3.1415926544231239, Error is 0.0000000008333307
wall clock time = 0.001874
Process 1 of 6 on out-cs-southpole
Process 2 of 6 on cs-room443-01
Process 3 of 6 on cs-room443-02
Process 5 of 6 on cs-room443-04
Process 4 of 6 on cs-room443-03
[user01@cs-room443-05 basic]$
```

となって，正常に動作する。どうやら、NIS で共有しているホストファイル `hosts` にリストアップされているとダメらしい。よって，cs-southople の/etc/hosts から out-cs-southpole のエントリ (学内 LAN への接続口) を削除すると

```
[user01@cs-southpole mpipc]$ mpirun -np 14 ./a.out
Process 0 of 14 on cs-southpole
pi is approximately 3.1415926544231270, Error is 0.0000000008333338
wall clock time = 0.274901
Process 4 of 14 on cs-room443-b04
Process 1 of 14 on cs-room443-b01
Process 5 of 14 on cs-room443-s01
Process 3 of 14 on cs-room443-b03
Process 2 of 14 on cs-room443-b02
Process 6 of 14 on cs-room443-s02
Process 8 of 14 on cs-room443-s04
Process 7 of 14 on cs-room443-s03
Process 9 of 14 on cs-room443-01
Process 10 of 14 on cs-room443-02
Process 11 of 14 on cs-room443-03
Process 13 of 14 on cs-room443-05
Process 12 of 14 on cs-room443-04
[user01@cs-southpole mpipc]$
```

となって，cs-southpole でも 1 プロセスのみ動作するようになった。

この後，MPIBLACS と ScaLAPACK のインストールを行う。大体は BLAS と LAPACK のインストール手順と同じである。

1. mpiblacs.tgz を解凍し，BLACS/ディレクトリを生成する
2. パッチがあれば必要に応じてあてる
3. BLACS/BMAKES/Bmake.MPI-LINUX を BLACS/Bmake.inc に上書きコピー
4. make する
5. make 後，BLACS/LIB/ディレクトリが

   ```
   [user01@cs-southpole BLACS]$ ls LIB
   LIB.log                   blacsF77init_MPI-LINUX-0.a
   blacsCinit_MPI-LINUX-0.a  blacs_MPI-LINUX-0.a
   [user01@cs-southpole BLACS]$
   ```

   となっていれば O.K.
6. BLACS/LIB/のライブラリを/usr/local/lib へコピー
7. scalapack.tgz を解凍し，SCALAPACK/ディレクトリを生成する
8. SCALAPACK/INSTALL/SLmake.LINUX を適宜編集して，SCALAPACK/SLmake.inc へ上書き
9. make する

10. libscalapack.a が生成されていれば O.K.

参考までに，使用した SLmake.inc を以下に示す。

```
[user01@cs-southpole SCALAPACK]$ cat SLmake.inc
###########################################################################
#
#  Program:         ScaLAPACK
#
#  Module:          SLmake.inc
#
#  Purpose:         Top-level Definitions
#
#  Creation date:   February 15, 2000
#
#  Modified:
#
#  Send bug reports, comments or suggestions to scalapack@cs.utk.edu
#
###########################################################################
#
SHELL         = /bin/sh
#
#  The complete path to the top level of ScaLAPACK directory, usually
#  $(HOME)/SCALAPACK
#
#home          = $(HOME)/SCALAPACK
home          = $(HOME)/pool/SCALAPACK
#
#  The platform identifier to suffix to the end of library names
#
PLAT          = LINUX
#
#  BLACS setup.  All version need the debug level (0 or 1),
#  and the directory where the BLACS libraries are
#
BLACSDBGLVL   = 0
#BLACSdir      = $(HOME)/BLACS/LIB
BLACSdir      = /usr/local/lib
#
#  MPI setup; tailor to your system if using MPIBLACS
#  Will need to comment out these 6 lines if using PVM
#
USEMPI        = -DUsingMpiBlacs
```

```
#SMPLIB        = /usr/lib/mpi/build/LINUX/ch_p4/lib/libmpich.a
SMPLIB        = /usr/local/lib/libmpich.a
BLACSFINIT    = $(BLACSdir)/blacsF77init_MPI-$(PLAT)-$(BLACSDBGLVL).a
BLACSCINIT    = $(BLACSdir)/blacsCinit_MPI-$(PLAT)-$(BLACSDBGLVL).a
BLACSLIB      = $(BLACSdir)/blacs_MPI-$(PLAT)-$(BLACSDBGLVL).a
TESTINGdir    = $(home)/TESTING

#
#  PVMBLACS setup, uncomment next 6 lines if using PVM
#
#USEMPI        =
#SMPLIB        = $(PVM_ROOT)/lib/$(PLAT)/libpvm3.a
#BLACSFINIT    =
#BLACSCINIT    =
#BLACSLIB      = $(BLACSdir)/blacs_PVM-$(PLAT)-$(BLACSDBGLVL).a
#TESTINGdir    = $(HOME)/pvm3/bin/$(PLAT)

CBLACSLIB     = $(BLACSCINIT) $(BLACSLIB) $(BLACSCINIT)
FBLACSLIB     = $(BLACSFINIT) $(BLACSLIB) $(BLACSFINIT)

#
#  The directories to find the various pieces of ScaLapack
#
PBLASdir      = $(home)/PBLAS
SRCdir        = $(home)/SRC
TESTdir       = $(home)/TESTING
PBLASTSTdir   = $(TESTINGdir)
TOOLSdir      = $(home)/TOOLS
REDISTdir     = $(home)/REDIST
REDISTTSTdir  = $(TESTINGdir)
#
#  The fortran and C compilers, loaders, and their flags
#
F77           = mpif77
CC            = mpicc
NOOPT         =
F77FLAGS      =  -funroll-all-loops -O3 $(NOOPT)
DRVOPTS       = $(F77FLAGS)
CCFLAGS       = -O4
SRCFLAG       =
F77LOADER     = $(F77)
CCLOADER      = $(CC)
F77LOADFLAGS  =
```

```
CCLOADFLAGS   =
#
#  C preprocessor defs for compilation
#  (-DNoChange, -DAdd_, -DUpCase, or -Df77IsF2C)
#
CDEFS         = -Df77IsF2C -DNO_IEEE $(USEMPI)
#
#  The archiver and the flag(s) to use when building archive (library)
#  Also the ranlib routine.  If your system has no ranlib, set RANLIB = echo
#
ARCH          = ar
ARCHFLAGS     = cr
RANLIB        = ranlib
#
#  The name of the libraries to be created/linked to
#
SCALAPACKLIB  = $(home)/libscalapack.a
#BLASLIB       = /usr/lib/libblas.a
BLASLIB       = /usr/local/lib/blas_LINUX.a
#
PBLIBS        = $(SCALAPACKLIB) $(FBLACSLIB) $(BLASLIB) $(SMPLIB)
PRLIBS        = $(SCALAPACKLIB) $(CBLACSLIB) $(SMPLIB)
RLIBS         = $(SCALAPACKLIB) $(FBLACSLIB) $(CBLACSLIB) $(BLASLIB) $(SMPLIB)
LIBS          = $(PBLIBS)
```

3 ScaLAPACK のベンチマークテスト

以上の環境で，ScaLAPACK のテストプログラムが正常に動作することを確認する。

```
[user01@cs-room443-05 TESTING]$ mpirun -np 6 xclu
ScaLAPACK Ax=b by LU factorization.
'MPI Machine'

Tests of the parallel complex single precision LU factorization and solve.
The following scaled residual checks will be computed:
 Solve residual         = ||Ax - b|| / (||x|| * ||A|| * eps * N)
 Factorization residual = ||A - LU|| / (||A|| * eps * N)
The matrix A is randomly generated for each test.

An explanation of the input/output parameters follows:
TIME    : Indicates whether WALL or CPU time was used.
M       : The number of rows in the matrix A.
N       : The number of columns in the matrix A.
```

```
NB      : The size of the square blocks the matrix A is split into.
NRHS    : The total number of RHS to solve for.
NBRHS   : The number of RHS to be put on a column of processes before going
          on to the next column of processes.
P       : The number of process rows.
Q       : The number of process columns.
THRESH  : If a residual value is less than THRESH, CHECK is flagged as PASSED
LU time : Time in seconds to factor the matrix
Sol Time: Time in seconds to solve the system.
MFLOPS  : Rate of execution for factor and solve.

The following parameter values will be used:
  M       :             4    10    17    13
  N       :             4    12    13    13
  NB      :             2     3     4
  NRHS    :             1     3     9
  NBRHS   :             1     3     5
  P       :             1     2     1     4
  Q       :             1     2     4     1

Relative machine precision (eps) is taken to be       0.596046E-07
Routines pass computational tests if scaled residual is less than   1.0000

TIME     M     N  NB NRHS NBRHS    P    Q  LU Time Sol Time  MFLOPS  CHECK
---- ----- ----- --- ---- ----- ---- ---- -------- -------- -------- ------

WALL     4     4   2     1    1    1    1     0.00     0.00     1.17 PASSED
(略)
WALL    13    13   4     9    5    4    1     0.01     0.00     1.81 PASSED

Finished    240 tests, with the following results:
  240 tests completed and passed residual checks.
    0 tests completed and failed residual checks.
    0 tests skipped because of illegal input values.


END OF TESTS.
```

4 姫野 Bench による測定結果

Serial 1 CPU(Pentium III Coppermine 1GHz) の場合

```
[user01@cs-room443-01 himenoBMTxp_s]$ g77 himenoBMTxp_s.f
```

```
[user01@cs-room443-01 himenoBMTxp_s]$ ./a.out
  mimax= 129 mjmax= 65 mkmax= 65
  imax= 128 jmax= 64 kmax= 64
  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  63.3372955  time(s):  0.779999971  0.00328862783
 Now, start the actual measurement process.
 The loop will be excuted in 230 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  230 times
  Gosa :  0.0015757794
  MFLOPS:  63.1156464  time(s):  60.0100021
  Score based on Pentium III 600MHz :  0.76189822
```

Serial 1 CPU(Pentium 4 1.7GHz CacheSize 256KB) の場合

```
[user01@cs-www himenoBMTxp_s]$ ./a.out
  mimax= 129 mjmax= 65 mkmax= 65
  imax= 128 jmax= 64 kmax= 64
  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  107.39801  time(s):  0.460000038  0.00328862783
 Now, start the actual measurement process.
 The loop will be excuted in 391 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  391 times
  Gosa :  0.00116495986
  MFLOPS:  106.727486  time(s):  60.329998
  Score based on Pentium III 600MHz :  1.2883569
PAUSE  statement executed
To resume execution, type go.  Other input will terminate the job.
```

Serial 1 CPU(Pentium 4 2.0GHz, CacheSize 512KB) の場合

```
[user01@cs-minerva-new himenoBMTxp_s]$ ./a.out
  mimax= 129 mjmax= 65 mkmax= 65
  imax= 128 jmax= 64 kmax= 64
  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  131.059021  time(s):  0.376952976  0.00328862783
 Now, start the actual measurement process.
 The loop will be excuted in 477 times.
 This will take about one minute.
```

```
 Wait for a while.
  Loop executed for  477 times
  Gosa :  0.00100518297
  MFLOPS:  130.616302  time(s):  60.1386719
  Score based on Pentium III 600MHz :  1.57672989
```

MPICH 使用の場合

```
[user01@cs-room443-01 f77_xp_mpi]$ mpirun -np 1 ./a.out
 Sequential version array size
  mimax= 129 mjmax= 65 mkmax= 65
 Parallel version  array size
  mimax= 129 mjmax= 65 mkmax= 65
  imax= 128 jmax= 64 kmax= 64
  I-decomp=  1 J-decomp=  1 K-decomp=  1

  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  63.0183226  time(s):  0.783948  0.00328862783
 Now, start the actual measurement process.
 The loop will be excuted in 229 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  229 times
  Gosa :  0.00157886976
  MFLOPS:  63.0317608  time(s):  59.828606
  Score based on Pentium III 600MHz :  0.760885596

[user01@cs-room443-01 f77_xp_mpi]$ mpirun -np 2 ./a.out
 Sequential version array size
  mimax= 129 mjmax= 65 mkmax= 65
 Parallel version  array size
  mimax= 67 mjmax= 65 mkmax= 65
  imax= 65 jmax= 64 kmax= 64
  I-decomp=  2 J-decomp=  1 K-decomp=  1

  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  115.033222  time(s):  0.429468  0.00330138975
 Now, start the actual measurement process.
 The loop will be excuted in 419 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  419 times
```

```
 Gosa :  0.00110981509
 MFLOPS:  115.085942  time(s):  59.954887
 Score based on Pentium III 600MHz :  1.38925576

[user01@cs-room443-01 f77_xp_mpi]$ mpirun -np 4 ./a.out
 Sequential version array size
  mimax= 129 mjmax= 65 mkmax= 65
 Parallel version  array size
  mimax= 67 mjmax= 35 mkmax= 65
  imax= 65 jmax= 33 kmax= 64
  I-decomp=  2 J-decomp=  2 K-decomp=  1

  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  227.728016  time(s):  0.216939  0.00329221319
 Now, start the actual measurement process.
 The loop will be excuted in 829 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  829 times
  Gosa :  0.000569808995
  MFLOPS:  228.889065  time(s):  59.643391
  Score based on Pentium III 600MHz :  2.763026

[user01@cs-room443-01 f77_xp_mpi]$ mpirun -np 6 ./a.out
 Sequential version array size
  mimax= 129 mjmax= 65 mkmax= 65
 Parallel version  array size
  mimax= 46 mjmax= 35 mkmax= 65
  imax= 44 jmax= 33 kmax= 64
  I-decomp=  3 J-decomp=  2 K-decomp=  1

  Start rehearsal measurement process.
  Measure the performance in 3 times.
   MFLOPS:  340.602898  time(s):  0.145046  0.00329310773
 Now, start the actual measurement process.
 The loop will be excuted in 1240 times.
 This will take about one minute.
 Wait for a while.
  Loop executed for  1240 times
  Gosa :  0.000301655731
  MFLOPS:  369.818199  time(s):  55.216166
  Score based on Pentium III 600MHz :  4.46424675
```

謝辞

本稿を作成するにあたり使用したソフトウェアの開発者の方々全てに対して，感謝の意を表します。

参考文献

[1] BLACS, http://www.netlib.org/blacs/

[2] 姫野ベンチマーク, http://w3cic.riken.go.jp/HPC/HimenoBMT/

[3] LAPACK, http://www.netlib.org/lapack/

[4] MPICH, http://www-unix.mcs.anl.gov/mpi/mpich/

[5] Netlib, http://www.netlib.org/

[6] Linux NFS-HOWTO, http://www.linux.or.jp/JF/JFdocs/NFS-HOWTO/

[7] Linux NIS-HOWTO, http://www.linux.or.jp/JF/JFdocs/NIS-HOWTO/

[8] PC Cluster Consortium, http://www.pccluster.org/

[9] PHASE Project, http://www.hpcc.jp/

[10] ScaLAPACK, http://www.netlib.org/scalapack/

[11] 超並列計算研究会, http://www.is.doshisha.ac.jp/SMPP/

[12] Vine Linux, http://www.vinelinux.org/