ANL/MCS-TM-ANL-01/x Rev x

Installation and User's Guide to MPICH, a Portable Implementation of MPI Version 1.2.7 The ch_p4 device for Workstation Networks

by

William Gropp and Ewing Lusk

MATHEMATICS AND
COMPUTER SCIENCE
DIVISION

Contents

Abstract 1

1 Introduction 1

1.1 Features of recent releases 2

4.1.3 Signals 24
4.2 Workstation Networks with the ch_p4 device 24
4.3 Special issues for heterogeneous networks and the ch_p4 device 25
4.4 Setting up rsh 25
4.5 Configuring with ssh 26
4.5.1 Original SSH (version 1) 26

6.7.1 General 50

A.12 C++ Builds Fail 83
A.13 Fortran programs give errors about mismatched types 83
A.14 Missing Symbols When Linking 83
A.15 Warning messages while building MPICH 84
A.16 MPMD (Multiple Program Multiple Data) Programs 84
A.17 Re8g0100uting problems and sup8g0100ut 84
A.18 Algorithms used in MPICH 84
A.19 Jumpshot and X11 85

B History of MPICH 85

C File Manifest 85

D Configure Usage 86

E Mpirun Usage 93

F Deprecated Features 96
F.1 Getting Tcl, Tk, and wish 96
F.2 Obsolete Systems 98
F.3 More detailed control over compiling and linking 99

References 100

Abstract

MPI (Message-Passing Interface) is a standard specification for message-passing libraries. MPICH is a portable implementation of the full MPI-1 specification for a wide variety of parallel and distributed computing environments. MPICH contains, along with the MPI library itself, a programming environment for working with MPI programs. The programming environment includes a portable startup mechanism, several profiling libraries for studying the performance of MPI programs, and an X interface to all of

  - Miscellaneous new `MPI_Info` and `MPI_Datatype` routines.

- `MPICH`

we developed for the NEC SX-4 [9]. The ch_shmem

instead.)

This will configure MPICH for the default device; this is usually the appropriate choice. Section 4.1 discusses the options that can be given to configure to customize MPICH.

The output of configure is piped to tee; this program both writes the output to

2.4 Sample MPI programs

The MPICH distribution contains a variety of sample programs, which are located in the MPICH source tree. Most of these will work with any MPI implementation, not just MPICH.

examples/baaic contains a few short programs in Fortran, C, and C++ for testing the simplest features of MPI.

examples/test contains multiple test directories for the various parts of MPI. Enter "make testing" in this directory to run our suite of function tests.

examples/perftest Performance benchmarking programs. See the script runmpptest for information on how to run the benchmarks. These are relatively sophisticated.

```
mpicc -o foo foo.o
mpif77 -o foo foo.o
mpicxx -o foo foo.o
mpif90 -o foo foo.o
```

-mpianim Build version that generates real-time animation.

These are described in more detail in Section

individually, though programs to help start these processes exist (see Section 3.3.4 below). Because workstation clusters are not already organized as an MPP, additional information is required to make use of them. MPICH should be installed wh.21d128(e)-450(list)-128(of)-451(participatir

3. Can user programs be run on remote systems? This checks that shared libraries and other components have been properly installed on all machines.

3.3.2 Changing the Remote Shell Program

You can change the remote shell command that the `ch_p4` device uses to start the remote processes with the environment variable `P4_RSHCOMMAND`. For example, if the default remote shell program is `rsh` but you wish to use the secure shell `ssh`, you can do

```
setenv P4_RSHCOMMAND ssh
mpirun -np 4 a.out
```

This only works for different remote shell commands that accept the same command line

may contain 3 Sun SPARC (sun4) workstations and 3 SGI IRIX workstations, all of which communicate via the TCP/IP protocol. The `mpirun` command may be told to use all of these by using multiple `-arch` and `-np` arguments. For example, to run a program on 3 sun4s and 2 SGI IRIX workstations, use

```
mpirun -arch sun4 -np 3 -arch IRIX -np 2 program.%a
```

The special program name `program.%a` allows you to specify the different executables for the program, since a Sun executable won't run on an SGI workstation and vice versa. The `%a` is replaced with the architecture name; in this example, `program.sun4` runs on the

```
sun1   0  /users/jones/myprog
sun2   1  /users/jones/myprog
sun3   1  /users/jones/myprog
hp1    1  /home/mbj/myprog    mbj
```

The above file specifies four processes, one on each of three suns and one on another workstation where the user's account name is different. Note the 0 in the first line. It is there to indicate that no *other* processes are to be started on host sun1 than the one started y the user, 2826(y)-832(his)-333(command)]TJ 16.94-20.74 TD[(Y)83(ou)-379(migh)27(t)-379(w)27(an)27(t)-379(

d use

```
/jbg/main
n/graphics
```

ovide different command line argument to different MPI processes.

3.3.8 Using special or multiple interconnects

In some installations, certain hosts can be connected in multiple ways. For example, the "normal" Ethernet may be supplemented by a high-speed FDDI ring. Usually ... file. In this case, it is important to use ... -2826(y)-404(a)-406(high-sp)-28(eed)-406(FDDI)-406(ring.)-662(Usu

ocal 0 but to use the name of the local host. For example, if hosts host1

host1-atm host2-atm

3.3.9 Using Shared Libraries with the ch_p4 device

As described at the end of Section 4.10, it is sometime necessary to ensure that environment variables have been communicated to the remote machines before the program that makes use of shared libraries starts. The various remote shell commands (e.g., rsh and ssh) do not do this. Fortunately, the secure server (Section 3.3.4) does communicate the environment variables. This server is built and installed as part of the ch_p4 device, and can be installed

on the first machine and

```
% gdb cpi
GNU gdb 5.0
Copyright 2000 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
```

3.7 Execution tracing

Execution tracing is easily accomplished using the `-mpitrace` command line argument while linking:

```
mpicc -c cpi.c
mpicc -o cpi -mpitrace cpi.o
```

3.8 Performance measurements

The ‘`mpich/examples/perftest`


30.5

Use `--with-arch=IRIX` to force 32 bit pointers and

```
localhost
your_machine_name
```

where your_machine_name is the name that you've given your machine in '/etc/hosts'.

(Cray UNICOS is handled separately). If you are building shared libraries, you will also

4.11.1 NFS and MPI-IO

4.11.2 Dealing with automounters

This will clean all the directories of previous object files (if any), compile both profiling and non-profiling versions of the source code, including Romio and the C++ interface, build all necessary libraries, and link both a sample Fortran program and a sample C program as a test that everything is working. If anything goes wrong, check Section 6 to see if there

```
make install PREFIX=/usr/local/mpich-1.2.7
```

However, so9e features, particularly the ability of Totalview to show MPICH message queues,

```
rpcinfo -p mysun
```

2. Start the secure server. The script ‘sbin/chp4_servs’

```
sbin/chp4_servs -port=n -arch=$ARCH
```

can be used to start the secure servers. This makes use of the remote shell command (rsh, remsh, or ssh) to start the servers; if you cannot use the remote shell command, you will need to log into each system on which you want to start the secure server and start the server manually. The command to start an individual server using port 2345 is

```
serv_p4 -o -p 2345 &
```

serv_p4

4.15 Thorough Testing

The 'examples/test' directory contains subdirectories of small programs that systematically test a large subset of the MPI functions. The command

```
make testing
```

in the MPICH directory will cause these programs to be compiled, linked, executed, and their output to be compared with the expected output. Linking all these test programs takes up considerable space, so you might want to do

```
make clean
```

in the test directory afterwards. The individual parts of MPI (point-to-point, collective, topology, etc.) can be tested separately by

```
make testing
```

in the separate subdirectories for examples/test.

If you have a problem, first check the troubleshooting guides and the lists of known problems. If you still need help, send detailed information to mpi-bugs@mcs.anl.gov.

4.16 Tu3334(74(P4gh)-37P)30(erinfornceng)]TJ/F41 0.91 Tf0 -2 7.79 TD[(Tr8(e)-31ar8(e)-3

TCP Tuning. The command line option `-p4sctrl`

For emacs users, check the Emacs info under “European Display”. The commands

```
M-x standard-display-european
M-x iso-accents-mode
```

can be used to input most European languages. You can also load ‘ `iso-transl`’ and use `C-x 8` to compose characters (this sets the high bit in the character). `MPICH` does not support languages that require multi-byte character sets (such as Japanese). However, the only changes needed are in the file ‘`src/env/errmsg.c`’; if you are interested in developing a multi-byte character set version, please let us know.

By default, `MPICH` uses the value of ‘`NLSPATH`’ to find the message catalogs. If this fails, it tries ‘`MPICHNLSPATH`’, and if that fails, it uses English language versions that are coded into the library.

The catalogs are not, however, installed into these directories. Instead, you will find them in the library directory for a particular architecture; for example, ‘`mpich/rs6000/lib`’.

5 Documentation

This distribution of `MPICH` comes with complete `man` pages for the MPI routines, commands moypageyfor MPI and ext liEmpidocnsl

– *Using MPI-2: Advanced Features of the Message-Passing Interface*, by Gropp, Lusk, and Thakur [14].
– *MPI—The Complete Reference: Volume 1, The MPI Core*, by Snir, et al. [19].
–

6.2 Submitting bug reports

Send any problem that you can not solve by checking this section to mpi-bugs@mcs.anl.gov.

Please include:

- The version of MPICH (e.g., 1.2.7)
-

```
iptables --list
ipchains --list
```

that `mpirun`

6.2.1i31(uxal)]TJETBT/F4 f17.793 1.12169.34.6 D[(1.)]TJ/F6 10.91 f13.940 D[(Q:)]TJ/F4 10.91

6.6 Problems configuring

6.6.1 General

Q: Configure reports that floating point is not commutative! How do I fix it?

A: Check your compiler's documentation. On RS/6000's, the `-qnomaf` (no multiply-add floating point) option. On some other systems, intermediate results may be stored in 80-bit registers (Intel CPUs do this);indoCPUsdooi4ion.32pEdati.ed

A:

Q: 2. Q: When running make on MPICH, I get errors when executing ranlib.

A: Many systems implement ranlib with the ar command, and they use the '/tmp' directory by default because it "seems" obvious that using '/tmp' would be faster ('/tmp' is often a local disk). Unfortunately, some systems have ridiculously small '/tmp' partitions, making any use of '/tmp' very risky. In some cases, the ar commands used by MPICH will succeed because they use the l option—this forces ar to use the local directory instead of '/tmp'. The ranlib command, on the other hand, may use '/tmp' and cannot be fixed.

In some cases, you will find that the ranlib command is unnecessary. In these cases, you can reconfigure with -noranlib. If you must use ranlib, either reduce the space used ' /tmp' increase the size of the ' /tmp' (your administrator will need to do this). There should be at least 20–30 MBytes free in '/tmp'.

3. Q: When doing the link test, the link fails and does not seem to find any of the MPI routines:

```
  /homes/me/mpich/IRIX32/ch_p4/bin/mpicc -o overtake overtake.o test.o
ld: WARNING 126: The archive /homes/me/mpich/IRIX32/ch_p4/lib/libmpi.a
  defines no global symbols. Ignoring.
ld: WARNING 84: /usr/lib/libsun.a is not used for resolving any symbol.
ld: ERROR 33: Unresolved data symbol "MPI_COMM_WORLD" -- 1st referenced
  by overtake.o.
ld: ERROR 33: Unresolved text symbol "MPI_Send" -- 1st referenced by
  overtake.o.
...
```

A: Check that the ar and ranlib programs are compatible. One site installed the Gnu ranlib in such a way that it would be used with the vendors ar program, with which it was incompatible. Use the -noranlib option to configure if this is the case.

2. Q: When building the ch_p4 device, I get errors of the form

```
making p4 in directory lib
        make  libp4.a
        cc -I../include -I../../../../include   -c p4_globals.c
        cc -I../include -I../../../../include   -c p4_MD.c
        cc -I../include -I../../../../include   -c p4_error.c
cc-142 cc: WARNING File = p4_error.c, Line = 152
  The number of old style and prototype parameters does not agree.
cc-142 cc: WARNING File = p4_error.c, Line = 162
  The number of old style and prototype parameters does not agree.
cc-142 cc: WARNING File = p4_error.c, Line = 169
  The number of old style and prototype parameters does not agree.
cc-142 cc: WARNING File = p4_error.c, Line = 174
  The number of old style and prototype parameters does not agree.
```

A: These have to do with declarations for a signal handler, and can be ignored. Specifically, P4 is using the `SIG_IGN` (ignore signal) and `SIG_DFL` (default behavior) which are defined in '`/usr/include/signal.h`', and these definitions are not correct.

6.7.3 Cray T3D

1. Q: When linking I get

```
mppldr-133 cf77: CAUTION
    Unsatisfied external references have been encountered.

Unsatisfied external references
Entry name      Modules referencing entry

GETARG (equivalenced to $USX1)
                MPIR_GETARG
```

A: You may have specified the Fortran compiler with the `F77` environment variable or the `-fc` argument to configure. The Fortran implemenation of MPI uses a common extension, `GETARG`, to get command line arguments. Most Fortran

```
Signal: SIGSEGV in Back End Driver phase.
> ### Error:
> ### Signal SIGSEGV in phase Back End Driver -- processing aborted
> f77 ERROR:  /usr/lib64/cmplrs/be died due to signal 4
```

6.7.6 Compaq ULTRIX and Tru64

1. Q: When trying to build, the make aborts early during the cleaning phase:

   ```
   amon:MPICH/mpich>make clean
           /bin/rm -f *.o *~ nupshot
   *** Error code 1
   ```

   A: This is a bug in the shell support on some Compaq ULTRIX systems. You may be able to work around this with

   ```
   setenv PROG_ENV SYSTEM_FIVE
   ```

   Configuring with `-make=s5make` may also work.

6.8 Problems in testing

The MPICH test suite, in 'examples/test', performs a fairly complete test of an MPI imple-

6.9 Problems compiling or linking Fortran programs

6.9.1 General

1. Q: When linking the test program, the following message is generated:

```
f77 -g -o secondf secondf.o -L/usr/local/mpich/lib -lmpich
invalid option -L/usr/local/mpich/lib
ld: -lmpich: No such file or directory
```

A: This f77 program 9J/Fm 9J/(es/Fm)-2not/Fmfol0.91 Tf24.510 138[(program)-L10.91 Tf16.690 4[(Th

A: You need to add the link option `-lV3`. The `ch_p4` device uses the System V signals on the HP; these are provided in the 'V3' library.

6.10.3 LINUX

1. Q: When linking a Fortran program, I get

```
Linking:
foo.o(.data+0x0): undefined reference to 'pmpi_wtime_'
```

A:

6.11.2 Workstation Networks

1. Q: When I use `mpirun`, I get the message `Permission denied`.

   A: See Section 6.3

4. Q:

```
which hostname
```

If you see the same strange output, then your problem is in your '`.cshrc`' file. You may have some code in your '`.cshrc`' file that assumes that your shell is connected to a terminal.

7. Q: When I try to run my program, I get

```
p0_4652:  p4_error: open error on procgroup file (procgroup): 0
```

A: This indicates that the `mpirun`

A: This means that you are trying to run MPICH in one mode when it was configured for another. In particular, you are specifying in your p4 procgroup file that several processes are to shared memory on a particular machine by either putting a number

A similar problem can happen on IBM SPs using the `ch_mpl` device; the cause is the same but it originates within the IBM MPL library.

15. Q: Sometimes, I get the error

```
Exec format error. Wrong Architecture.
```

A: You are probably using NFS (Network File System). NFS can fail to keep files updated in a timely way; this problem can be caused by creating an executable on one machine and then attempting to use it from another. Usually, NFS catches up with the existence of the new file within a few minutes. You can also try using the sync command. `mpirun` in fact tries to run the sync

6.12 Programs fail at startup

2. Q: My Fortran program fails with a BUS error.
   A: The C compiler that

A:

uses fork

results in program tracing information at a level of 20 being written to `stdout` during execution. For more information about what is printed at what levels, see the p4 Users' Guide [2].

Acknowledgments

- C++ Builds Fail
- Fortran programs give errors about mismatched types
-

A.5 Notes on getting MPICH running Under Linux

Introduction The purpose of this document is to describe the steps necessary to allow MPICH processes to be started and to communicate with one another. The installation

At this point you should receive a "Permission denied." if you attempt a command such as "rsh localhost hostname" as a non-root user (or as root for that matter).

To allow users to rsh without passwords you need to edit '`/etc/hosts.equiv`', the

At this point ssh on the localhost should work, although a password will still be required. However, our firewall rules will be preventing connections from other machines.

We again modify '`/etc/sysconfig/ipchains`', this time to allow ssh tra□c in and out.

```
# -A input -p tcp -s 0/0 -d 0/0 7100 -y -j REJECT
#
# End of removed rules
#
```

This modification, in conjunction with one to allow process startup, should prepare your system for MPICH jobs.

A.6 poll: protocol failure during circuit creation

A.8 Mac OS X and hostname

Under Mac OS X, the hostname handling is unusual. The hostname that the `hostname` command or the `gethostname` function return is determined by the name set in the "Sharing preference" pane. If this name contains a space, you may get only the leading part

A.12 C++ Builds Fail

If the C++ build fails with messages about ambiquities in the definitions, try reconfiguring

```
make clean
make profile
ar ../../../lib/libpmpich.a *.o
ranlib ../../../lib/libpmpich.a
```

case for the systems that we have been developing on. Thus, we will always need the

README

```
[--with-mpe] [--without-mpe]
[--disable-f77] [--disable-f90] [--with-f90nag] [--with-f95nag]
[--disable-f90modules]
[--disable-gencat] [--disable-doc]
[--enable-cxx ] [--disable-cxx]
[--enable-mpedbg] [--disable-mpedbg]
[--enable-devdebug] [--disable-devdebug]
[--enable-debug] [--disable-debug]
```

```
                    or libmpich.  This can be used on systems with
                    several different MPI implementations.
   FILE_SYSTEM     = name of the file system ROMIO is to use.  Currently
                    supported values are nfs, ufs, pfs (Intel),
                    piofs (IBM), hfs (HP), sfs (NEC), and xfs (SGI).
   SIGNAL_NAME     = name of the signal for the P4 (device=ch_p4) device to
                    use to indicate that a new connection is needed.  By
                    default, it is SIGUSR1.

All arguments are optional, but if 'arch', 'comm', or 'prefix' arguments
are provided, there must be only one.  'arch' must be specified before
'comm' if they both appear.

Packages that may be included with MPICH

--with-device=name     - Use the named device for communication.  Known
                         names include ch_p4, ch_mpl, ch_shmem, and globus2.
                         If not specified, a default is chosen.  Special
                         options for the device are specified after the
                         device name, separated by a colon.  E.g.,
                         --with-device=globus2:-flavor=mpi,nothreads
--with-romio[=OPTIONS] - Use ROMIO to provide MPI-I/O from MPI-2 (default).
                         The options include -file_system=FSTYPE, where
                         fstype can be any combination of nfs, ufs,
                         pfs (intel), piofs (IBM), hfs (HP), sfs (NEC), and
                         xfs (SGI), combined with '+'.  If romio is not
                         included, the Fortran 90 modules cannot be built.
--with-mpe             - Build the MPE environment (default)
--with-f90nag          - Choose the NAG f90 compiler for Fortran
                         (preliminary version intended for use *instead*
                         of a Fortran 77 compiler)
--with-f95nag          - Choose the NAG f95 compiler for Fortran
--with-cross=file      - Use the file for cross compilation. The file
                         should contain assignments of the form
                            CROSS_SIZEOF_INT=4
                         for each cross compilation variable.  The command
                            egrep 'CROSS_[A-Z_]*=' configure | sed 's/=.*//g'
                         will list each variable.

You can use --without-<featurename> to turn off a feature (except for
device).
Options for device ch_lfshmem:
--with-device=ch_lfshmem[:-usesysv]
The option '-usesysv' applies to the ch_shmem device, and causes the device
to attempt and use System V shared memory and semaphore routines, rather
than what would be chosen by default (often mmap or a system[(to)-525ture
-ptions for device ch_lmeiko
```

The option '-noranlib' causes the 'ranlib' step (needed on some systems to build an object library) to be skipped. This is particularly useful on systems where 'ranlib' is optional (allowed but not needed; because it

and with the installation directory equal to the current directory:

```
                Conforms IEEE 754 standard.
C:       sizeof (int)     = 4; sizeof (float) = 4
```

```
-v      Verbose - throw in some comments
-dbg    The option '-dbg' may be used to select a debugger.  For example,
        -dbg=gdb invokes the mpirun_dbg.gdb script located in the
```

Available only on IBM SPs.

Special Options for IBM SP2:

A: You are probably building MPICH on an old 386 running System V release 2. This version of Unix has very severe limitations on the length of filenames (more severe than we are willing to cater to). The specific problem here is that the name of the file 'mpich/src/context/keyval_create.c' is too long for this system, and was not

References

[14] William Gropp, Ewing Lusk, and Rajeev Thakur. *Using MPI-2: Advanced Features of*