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_p4mpd device for Workstation Networks and Clusters

Contents

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

```
% make |& tee make.log
```

4. (Optional) If you wish to install MPICH in a public place so that others may use it, use

```
% make install
```

to install MPICHin . to the directory specified b827(y)-407(the)]TJ/F7 10.91 Tf 170.1TD[(in)--prefixin
]TJ/F41 10.91 Tf 27.918 TD[(in)',' and in

3.1 Compiling, linking, and running programs

The MPICH implementation provides four commands for compiling and linking C (mpicc), C++ (mpicxx), Fortran 77 (mpif77), and Fortran 90 (mpif90) programs.

Use these commands just like the usual C, Fortran 77, C++, or Fortran compilers. For example,

```
mpicc -c foo.c
mpif77 -c foo.f
mpicxx -c foo.cxx
mpif90 -c foo.f90
```

and

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

Commands for the linker may include additional libraries. For example, to use routines from the C math library library, use

```
mpicc -o foo foo.o -lm
```

Combining compilation and linking in a single command, as shown here,

```
mpicc -o foo foo.c
mpif77 -o foo foo.f
mpicxx -o foo foo.cxx
mpif90 -o foo foo.f
```

may also be used (on most systems).

Note that while the suffixes .c for C programs and .f for Fortran-77 programs are standard, there is no consensus for the suffixes for C++ and Fortran-90 programs. The ones shown here are accepted by many but not all systems. MPICH tries to determine the accepted suffixes, but may not always be able to.

Earlier versions of

6(thC333(lograry)mo(or,332(comaing)-34634-370.)27(u10.9pdumm7(ya□ng)-f(deTD[(.)TJ-16.77-(ust)-FD[(Y)raris)-344(example5)

info andto have MPICH compilation scripts
linking libraries that are required for correct operation of
Solaris systems, the _shmem

You can override the choice of compiler by specifying the environment variable MPICH_-CC, MPICH_F77, MPICH_CCC, or MPICH_F90. However, be warned that this will work only if the alternate compiler is compatible with the default one (by compatible, we mean that is uses the same sizes for all datatypes and layouts, and generates object code that can be used with the MPICH libraries). If you wish to override the linker, use the environment variables MPICH_CLINKER, MPICH_F77LINKER, MPICH_CCLINKER, or MPICH_F90LINKER.

If you want to see the commands that would be used without actually running them, add the command line argument -show.

In addition, the following special options are supported for accessing some of the features of the MPE environment for monitoring MPI calls from within an application:

-mpilog Build version that generates MPE log files.

-mpitrace Build version that generates traces.

-mpianim Build version that generates real-time animation.

These are described in more detail in Section 3.5.

3.1.1 Compiling and Linking without the Scripts

In some cases, it is not possible to use the scripts supplied by MPICH for compiling and linking programs. For example, another tool may have its own compilation scripts. In this case, you can use -compile_info and -link_

```
    mpirun -np 4 a.out
```

to run the program ‘a.out’ on four processors. The command `mpirun -help` gives you a complete list of options, which may also be found in Appendix

`-whole` The collection and merging of `stdout` is by default done in a way that maximizes efficiency. The disadvantage is that sometimes lines of output from differenencyt processes

3.3 MPMD Programs

3.4.3 Starting jobs with a debugger

The -dbg=<name of debugger> option to mpirun

Debugging with TotalView. You can set breakpoints by clicking in the left margin on a line number. Most of the TotalView GUI is self-explanatory. You select things with the left mouse button, bring up an action menu with the middle button, and “dive” into functions, variables, structures, processes, etc., with the right button. Pressing `cntl-?`

```
(mpigdb) n
0-4: 52                 x = h * ((double)i - 0.5);
(mpigdb) p x                            # bcast print command
0: $1 = 0.0050000000000000001           # 0’s value of x
1: $1 = 0.014999999999999999            # 1’s value of x
2: $1 = 0.025000000000000001            # 2’s value of x
3: $1 = 0.035000000000000003            # 3’s value of x
4: $1 = 0.044999999999999998            # 4’s value of x
```

There are several available log formats and `logviewer` selects the version of jumpshot appropriate for a particular logfile. See the MPE manual, distributed along with this manual, for more details.

3.6 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
```

3.7 Performance measurements

The ‘

Figure 1:

4.1.3 Signals

In general, users should avoid using signals with MPI programs. The manual page for `MPI_Init` describes the signals that are used by the MPI implementation; these should not be changed by the user.

Because Unix does not chain signals, there is the possibility that several packages will attempt to use the same signal, causing the program to fail. For example, by default, the `ch_p4` device uses `SIGUSR1`; some thread packages also use `SIGUSR1`.

For example, if MPICH was built with cc as the compiler but a user wanted to use gcc instead, either the commands

compilers), or use mixed case (an extension of Fortran, which is only monocase). Each of these choices requires a *separate* MPICH configure and build step. MPICH has been tested in the mode where monocase names are generated; this case is supported because only this case supports common (and necessary for MPICH) extensions such as getarg and iargc. By default, MPICH forces the Absoft compiler to use lowercase; this matches most Unix Fortran compilers. MPICH will find the appropriate versions of getarg and iargc for this case. Because the examples and the test suite assume that the Fortran compiler is case-insensitive; the Fortran library produced by MPICH

4.4 C++

The C++ support in MPICH has been provided by Indiana University (this group was

4.6 File System Issues

Most users do not need to worry about file systems. However, there are two issues: using NFS (the Network File System) with MPI-IO and using NFS with some automounters. These issues are covered in the next two sections.

4.6.1 NFS and MPI-IO

To use MPI-IO multihost on NFS file systems, NFS should be version 3, and the shared NFS directory must be mounted with the “no attribute caching” (noac) option set (the directory cannot be automounted). If NFS is not mounted in this manner, the following error could occur:

```
MPI_Barrier: Internal MPI error: No such file or directory
File locking messages
```

I order to reconfigure NFS to handle MPI-IO properly, the following sequence of steps are
(win4d)-(r333(o)63(nod)-(prop)-2m333(ersiod)-r04(25(6(t:ons.)]TJE31.122.963.813.541temse)-6(reconrmsiod)-

4.6.2 Dealing with automounters

Automounters are programs that dynamically make file systems available when needed. While this is very convenient, man(con)0(automoun)26(ters)-nare unable to recognize the file system

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 is anything said there about your problem. If not, follow the directions in Section 6.2 for submitting a bug report. To simplify checking for problems, it is a good idea to use

```
    make |& tee make.log
```

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

‘/usr/local/mpich-1.2.7/solaris/ch_p4’ and
‘/usr/local/mpich-1.2.7/solaris/ch_shmem’

If you intend to run the MPD daemons as root, then you must configure with --enable-root as well. Then it will be possible for multiple users to use the same set of MPD daemons to start jobs.

After configuration, the usual

```
make
make install
```

will install MPICH2 and thein the

```
% rlogin terra
Password: xxxxxxxxxxxxx
% /usr/local/mpich/bin/mpd -h shakey -p 39182 &
% logout
```

If you do have a working remote shell program, you can use a shell loop to start the processes on the remote machines. For example, if you have a list of nodes in the file ‘machines `mpdlistjobs` lists active jobs managed by `mpds` in ring.

`mpdkilljob job_id` aborts the specified job.

Several options control the behavior of the daemons, allowing them to be runopti3(of2pto)-434(b)27(y)]91

4.11 Internationalization

MPICH has support for providing error messages in different languages. This makes use of the X/Open message catalog services, which are a standard way of providing multi-language support. This multi-language support is often called NLS, for National Language Support. MPICH comes with error messages in US English; additional languages will be (viding)-2ed as

- As a Technical report [6].
- As Postscript and HTML at `www.mpi-forum.org`, for both MPI-1 and MPI-2.
- As a journal article in the Fall 1994 issue of the Journal of Supercomputing Applications [17] for MPI-1 and as a journal article in the International Journal

6.1 Things to try first

If sometgs goes wrong, the first tgs to do is to ch4ck the output of `configure` or `make` for

Connection Refused. This problem may be caused by Internet security settings on your system that restrict the number and frequency of interprocess connection operations. Check with your systems administrator. Linux users (depending on the Linux distri-

6.5 Other Problems

The following items describe miscellaneous problems that we have encountered. These may help solve less common problems.

6.6 Problems configuring

6.6.1 General

1. Q:

```
overtake.o(.text+0x59): undefined reference to ‘MPI_COMM_WORLD’
overtake.o(.text+0x81): undefined reference to ‘MPI_COMM_WORLD’
...
```

A:

6.7.5 Linux

1. Q: The link test failed on Linux with

   ```
   ...
   cc  -o overtake overtake.o test.o -L/usr/local/mpich/LINUX/ch_p4/lib
   -lmpi
   overtake.o(.text+0x71): undefined reference to ‘MPI_COMM_WORLD’
   overtake.o(.text+0x82): undefined reference to ‘MPIR_I_DOUBLE’
   overtake.o(.text+0xe1): undefined reference to ‘MPI_COMM_WORLD’
   ...
   ```

   A: We have been informed that there is a error in the `f77` script in some versions of Linux which causes this problem. Try either getting a patch for the `f77` script or reconfiguring with `-nof77`.

2. Q: The build fails for the `ch_p4`

6.8.1 General

1. Q: The test ‘`pt2pt/structf`

```
getdomainname
/home/mpich/lib/solaris/ch_shmem/libmpi .a(shmempriv.o)
```

The easiest but ugliest possibility is use `f2c` to convert Fortran to C, then use the C compiler to compile everything. If you take this route, remember that *every* Fortran routine has to be compiled using `f2c` and the C compiler.

Alternatively, you can use various options (check the man pages for your compilers) to see what libraries that add when they link. Add those libraries to the link line for the *other* either I get an error message or the program hangs.

A: On some systems such as IBM SPs, there are many mutually exclusive ways to run parallel programs; each site can pick the approach(es) that it allows. The script mpirun tries one of the more common methods, but may make the wrong choice. Use the -v or -t option to mpirun to see how it is trying to run the program, and then compare this with the site-specific instructions for using your system. You may need simplest fix may be to have your system administrators place the necessary shared libraries into one of the directories that is searched by default. If this is not possible, then you will need to help the compiler and linker out.

Many linkers provide a way to specify the search path for shared libraries. The

6. Q: Programs seem to take forever to start.

   A:

```
    if ($?USER == 0 || $?prompt == 0) exit
```

near the top of your ‘.cshrc’ fi9e (but *after* any code that sets up the runtime environment, such as library paths (e.g.,

6. Q: When using mpirun I get strange output like

   ```
   arch: No such file or directory
   ```

   A: This is usually a problem in your '.cshrc' file. Try the shell command

   ```
       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 program either did not create the expected input

   dop4mmandrysing og run a programbuilat

   the

You probably have '`/usr/lib`' in your path ahead of '`/usr/ucb`' or '`/usr/bin`'. This picks the 'restricted' shell instead of the 'remote' shell. The eas7est fix is to just remove '`/usr/lib`' from your path (few people need it); alternately, you can move it to after the directory that contains the 'remote' shell rsh.

then your system has a faulty installation of rsh. Some FreeBSD systems have been observed with this problem. Have your system administrator correct the problem (often one of an inconsistent set of rsh/rshd programs).

12. Q: My programs seem to hang in MPI_Init.

    A: There are a number of ways that this can happen:

    (a) One of the workstations you selected to run on is dead (try 'tstmachines' if you are using the ch_p4 device)

    (b) You linked with the FSU pthreads package; this has been reported to cause problems, particularly with the system select call that is part of Unix and is used by MPICH.

    Another is if you use the library '-ldxml' (extended math library) on Compaq Alpha systems. This has been observed to cause MPI_Init to hang. No

6.11.3 IBM RS6000

1. Q: When trying to run on an IBM RS6000 with the ch_p4 device, I got

   ```
   % mpirun -np 2 cpi
   Could not load program /home/me/mpich/examples/basic/cpi
   Could not load library libC.a[shr.o]
   Error was: No such file or directory
   ```

   A:

```
ERROR: 0031-124  Less than 2 nodes available from pool 0
```

A: This means that the IBM POE/MPL system could not allocate the requested nodes when you tried to run your program; most likely, someone else was using the

might compute

$$(((((((a + b) + c) + d) + e) + f) + g) + h)$$

```
class Z {
public:
  Z()  { cerr << "*Z" << endl; }
  ~Z() { cerr << "+Z" << endl; }
};

Z z;

int main(int argc, char **argv) {
  MPI_Init(&argc, &argv);
  MPI_Finalize();
}
```

6.13.4 Workstation Networks

1.

These arguments are provided to the program, not to mpirun. For example,

```
   mpirun -np 2 a.out -mpiversion
```

Acknowledgments

The work described in this report has benefited from conversations with and use by a large number of people. We also thank those that have helped in the implementation of MPICH, particularly Patrick Bridges and Edward Karrels. Particular thanks goes to Nathan Doss and Anthony Skjellum for valuable help in the implementation and development of MPICH. Debbie Swider, who worked with the MPICH group for several years, helped support and

- Compiler Switches
- C++ Builds Fail
- Fortran programs give errors about mismatched types
- Missing Symbols When Linking
- Warning messages while building MPICH
- MPMD (Multiple Program Multiple Data) Programs
- Reporting problems and support
- Algorithms used in MPICH
- Jumpshot and X11

A.1 Introduction

MPICH is a freely available, portable implementation of MPI, the Standard for message-passing libraries [illegible]

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 traffic 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

```
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]
            [--enable-traceback] [--disable-traceback]
            [--enable-long-long] [--disable-long-long]
            [--enable-long-double] [--disable-long-double]
            [-prefix=INSTALL_DIR]

            [-c++[=C++_COMPILER] ] [noc++]
            [-opt=OPTFLAGS]
            [-cc=C_COMPILER] [-fc=FORTRAN_COMPILER]
            [-clinker=C_LINKER] [-flinker=FORTRAN_LINKER]
            [-c++linker=CC_LINKER]
            [-cflags=CFLAGS] [-fflags=FFLAGS] [-c++flags=CCFLAGS]
            [-optcc=C_OPTFLAGS] [-optf77=F77_OPTFLAGS]
            [-f90=F90_COMPILER] [-f90flags=F90_FLAGS]
            [-f90inc=INCLUDE_DIRECTORY_SPEC_FORMAT_FOR_F90]
            [-f90linker=F90_LINKER]
            [-f90libpath=LIBRARY_PATH_SPEC_FORMAT_FOR_F90]
            [-lib=LIBRARY] [-mpilibname=MPINAME]
            [-mpe_opts=MPE_OPTS]
            [-make=MAKEPGM ]
            [-memdebug] [-ptrdebug] [-tracing] [-dlast]
            [-listener_sig=SIGNAL_NAME]
            [-cross]
            [-adi_collective]
            [-automountfix=AUTOMOUNTFIX]
            [-noranlib] [-ar_nolocal]
            [-rsh=RSHCOMMAND] [-rshnol]
            [-noromio] [-file_system=FILE_SYSTEM]
            [-p4_opts=P4_OPTS]
where
   ARCH_TYPE       = the type of machine that MPI is to be configured for
   COMM_TYPE       = communications layer or option to be used
   DEVICE          = communications device to be used
   INSTALL_DIR     = directory where MPI will be installed (optional)
   MPE_OPTS        = options to pass to the mpe configure
   P4_OPTS         = options to pass to the P4 configure (device=ch_p4)
   C++_COMPILER    = default is to use xlC, g++, or CC (optional)
   OPTFLAGS        = optimization flags to give the compilers (e.g. -g)
   CFLAGS          = flags to give C compiler
   FFLAGS          = flags to give Fortran compiler
   MAKEPGM         = version of make to use
   LENGTH          = Length of message at which ADI switches from short
                     to long message protocol
   AUTOMOUNTFIX    = Command to fix automounters
   RSHCOMMAND      = Command to use for remote shell
   MPILIBNAME      = nd ObYto 9(=)-525(Command)Fe whiin525(give)-525(tn=)-525(ndCommand)ve treshort
```

```
                     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,is muP_[(pi-TJ0▭11hne4ppear.specifi
```

```
piofi (5veral.95▭D[(5iofs)-524((IBM),)-525(hfs)-525((HP),)-525(sfs)-525((NEC),)Intel),
./configure: ./mpid/ch_p4/setup_ch_p4: Permission denied

Options for device ch_p4mpd:
--with-device=ch_p4mpd[:-listener_sig=SIGNALNAME][-dlast][-socksize=BYTES]
```

The option ’-listener_sig’ applies to the ch_p4mpd device, and changes the

```
The following are intended for MPI implementors and debugging of configure
--enable-strict       - Try and build MPICH using strict options in Gnu gcc
```

The option ’-noranlib’ causes the ’ranlib’ step (needed on some systems to build an object library) to be skipped. This is particularly useful 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
```

- The mpd's fork that number of `manager` processes (the executable is called `mpdman` and is located in the '`mpich/mpid/mpd`' directory). The managers are forked consecutively by the mpd's around the ring, wrapping around if necessary.

- The managers form themselves into a ring, and fork the application processes, called *clients*.

- The console disconnects from the mpd and reconnects to the first manager. `stdin` from `mpirun` is delivered to the client of manager 0.

- The managers interceptinman.341□0.nmI-404(the)0.nof is

```
cd tcl7.6
./configure -prefix=/usr/local/tcl73tk36
```

5. Build and install Tcl. Before you execute the `make install` step, make sure that the directory specified in the `-prefix` argument to `configure` exists.

```
mkdir /usr/local/tcl73tk36
make
make install
```

2. Q: I get errors compiling or running Fortran programs.

   A:

References

[1] James Boyle, Ralph Butler, Terrence Disz, Barnett Glickfeld, Ewing Lusk, Ross Overbeek, James Patterson, and Rick Stevens. *Portable Programs for Parallel Processors*. Holt, Rinehart, and Winston, New York, NY, 1987.

[2] R. Butler, W. Gropp, and E. Lusk. A scalable process-management environment for

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