

# A++ and P++

## A++ & P++

- [A++ & P++](#)
  - [Prerequisites](#)
  - [Build Instructions](#)
  - [Developer website:](#)
  - [Troubleshooting](#)


## Prerequisites

P++ requires MPI. We have had success using [OpenMPI-1.4.4](#) [MPICH-1.2.7](#).

## Build Instructions

1. Download tarball of source code: [AppPpp-08-28-2007.tar.gz](#) and unpack to `$INSTALL_DIR/src/a++p++`

```
cd $INSTALL_DIR/src/
wget http://download.hao.ucar.edu/pub/schmitt/cism/LTR-para/a++p++-08-28-2007.tar.gz
tar -zxvf a++p++-08-28-2007.tar.gz
```

- a.  Developers who want the latest & greatest (and quite possibly the most broken version of code) may download from the CISM repository  
Note: This requires an Subversion account

```
cd $INSTALL_DIR/src
svn --username [yourname] co https://proxy.subversion.ucar.edu/cism_CSE/trunk/a++p++
```

2. Prepare the A++ & P++ build using a configure script. (**Note:** you will run `configure` three separate times during this process)

```
cd $INSTALL_DIR/src/a++p++
./configure --prefix=$INSTALL_DIR --enable-PXX --no-recursion
```

3. Configure, build and install A++

```
cd $INSTALL_DIR/src/a++p++/A++
./configure --prefix=$INSTALL_DIR --with-CC=$CC --with-CXX=$CXX
make
make install
```

4. Configure, build and install P++

- a.

```
cd $INSTALL_DIR/src/a++p++/P++
./configure --prefix=$INSTALL_DIR --enable-PXX --with-CC=$MPICC --with-CXX=$MPICXX --disable-
mpirun-check --with-mpi-libs=" "
make
make install
```

5. Ensure that the build for both A++ and P++ went okay:

```
cd $INSTALL_DIR/src/a++p++
make check
```

This will traverse source directories, verifying the build went okay. All tests must pass in order to build LFM-para.

6. Set a few soft links for the installed files:

```
ln -s $INSTALL_DIR/A++/install/include $INSTALL_DIR/A++/include
ln -s $INSTALL_DIR/A++/install/lib $INSTALL_DIR/A++/lib
ln -s $INSTALL_DIR/P++/install/include $INSTALL_DIR/P++/include
ln -s $INSTALL_DIR/P++/install/lib $INSTALL_DIR/P++/lib
```

- 7. You are now ready to install the next prerequisite, [PVM](#).

## Developer website:

<http://www.llnl.gov/CASC/Overture>

## Troubleshooting

- On an IBM AIX system, you must pass the following flag to the A+P+ configure scripts:

```
--host=powerpc-ibm-aix5
```

- If you are getting weird build errors, try using `gmake` instead of `make`.
- On an AIX system like BlueVista, you may get an error, such as `$INSTALL_DIR/src/a+p+/config/mklib.aix: 0403-006 Execute permission denied..` This probably happened because you forgot to add the flag `--host=powerpc-ibm-aix5` to configure. Otherwise, you can try to manually set the permissions on this file to executable.

```
chmod gu+x $INSTALL_DIR/src/a++p++/config/mklib.aix
```

- When installing, you may get the error `cannot stat libApp.so': A file or directory in the path name does not exist`. You must build A+P+ with shared libraries turned OFF.
- When building P++, you may get an error like:

```
catastrophic error:
#error directive: "SEEK_SET is #defined but must not be for the C\+\+ binding of MPI"
error "SEEK_SET is #defined but must not be for the C\+\+ binding of MPI"
```

Solution: add `-DMPICH_IGNORE_CXX_SEEK` and `-DMPICH_SKIP_MPICXX` to the `CPPFLAGS` environment variable. If you're using `BASH`:

```
export CPPFLAGS="-DMPICH_IGNORE_CXX_SEEK -DMPICH_SKIP_MPICXX"
```

If you're using `csh`:

```
setenv CPPFLAGS "-DMPICH_IGNORE_CXX_SEEK -DMPICH_SKIP_MPICXX"
```

Once this variable is set clean up and re-configure your build

```
gmake clean
./configure [configure-options from above|configure-options from above]
gmake
```

- When building A++ with the Portland Group compilers, you may get an error like:

```
pgcc-Error-Unknown switch: -Wstrict-prototypes
```

Solution: remove any references to `-Wstrict-prototypes` from the Makefile.

- When building P++ with the Portland Group Compilers, you may get the linker errors:

```
int_MDI_5.c:-( .text+0x5c1): undefined reference to }}__mth_i_dpowdx'
int_MDI_5.c:-( .text+0x909): undefined reference to {{__mth_i_dpowdx'
int_MDI_5.c:-( .text+0x10f3): undefined reference to }}__mth_i_dpowdx'
int_MDI_5.c:-( .text+0x1969): undefined reference to {{__mth_i_dpowdx'
int_MDI_5.c:-( .text+0x267c): undefined reference to }}__mth_i_dpowdx'
```

The problem is that your missing the PGI SSE math libraries `-lpgsse1 -lpgsse2` from your link line. While you can add these with to various compilers (i.e. `gcc`), make sure that you are using the Portland compilers to build the code. You might want to set the `LDFLAGS` environment variable to point to these and re-configure the library from scratch:

```
export LDFLAGS="-lpgsse1 -lpgsse2"
```

- When building P++, you get the error:

```
CC          -o test2000_01 test2000_01.o -L../src -lPpp -lPpp_static -lPpp      -lc -lm
/usr/lib64/libpthread.a(lowlevellock.o): In function `__l1ll_lock_wait_private':
/usr/src/packages/BUILD/glibc-2.9/nptl/./nptl/sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:75:
multiple definition of `__l1ll_lock_wait_private'
/usr/lib64/libc.a(libc-lowlevellock.o):/usr/src/packages/BUILD/glibc-2.9/nptl/./nptl/sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:75: first defined here
/usr/lib64/libpthread.a(lowlevellock.o): In function `__l1ll_unlock_wake_private':
/usr/src/packages/BUILD/glibc-2.9/nptl/./nptl/sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:267:
multiple definition of `__l1ll_unlock_wake_private'
/usr/lib64/libc.a(libc-lowlevellock.o):/usr/src/packages/BUILD/glibc-2.9/nptl/./nptl/sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:267: first defined here
^Ccleaning up after signal(2)...
gmake[2]: *** Deleting file `test2000_01'
gmake[2]: *** [test2000_01] Error 127
gmake[1]: *** [all-recursive] Interrupt
gmake: *** [all] Interrupt
```

Solution: Recompile without linking libc (ie. remove `"-lc"` from build command)