Adlib version 1.0
Installation Notes

D.B. Carpenter

March 1, 1996

1 Prerequisites for installation

Adlib is a C++ class library for data-parallel programming. The current implmentation is layered on MPI.

Prerequisites for installation are:

- a C++ compiler, and
- an MPI installation.

These two systems form the *software platform* for the Adlib installation. Section 5 records the platforms on which we have successfully installed Adlib so far. If you have a significantly different platform, please contact `dbc@npac.syr.edu`. We will make our best efforts to help with the installation (and incorporate the knowledge gained into subsequent releases).

Public domain releases of C++ and MPI are available by anonymous ftp as follows:

- The GNU C++ compiler is available from various sites, e.g. Imperial College at `ftp://src.doc.ac.uk/pub/GNU/gcc-2.6.3.tar.gz`. You will also need `ftp://src.doc.ac.uk/pub/GNU/libg++-2.6.2.tar.gz`.
- The Argonne National Laboratory MPI implementation, MPICH, is available from `ftp://info.mcs.anl.gov/pub/mpi`. For further information, see the WWW page at `http://www.mcs.anl.gov/mpi/index.html`.

2 How to obtain Adlib

The Adlib release package is available by anonymous ftp from *[. . .]* where it is contained in the archive file `Adlib.1.0.tar.Z`.

Working in a suitable root directory, extract the archive file. For example:

```
zcat mpich.tar.Z | tar xvf -
```

This creates a subdirectory called `Adlib` in the current directory. `Adlib` is now the home directory for the installation.

3 Building Adlib

1. Ensure that the C++ compiler is on your path.
2. Change directory to the Adlib home directory, e.g.:

   ```
   cd Adlib
   ```

3. Configure the release with the `configure` script located in this directory. This script tailors various scripts and `make` files to the local software platform. Eventually we hope to provide a more intelligent GNU-style configuration script. For now, you must provide the required information about the platform as arguments to `configure`. The usage of `configure` is:

   ```
   configure  -c++com=C++_COM
             [-mpilib=MPI_LIB] [-mpiinc=MPI_INC_DIR] [-cmpih]
   ```

   where:

   `C++_COM` is the command to execute the C++ compiler. The `-c++com` argument is compulsory.

   `MPI_LIB` is the library (or list of libraries) containing MPI. The library may be specified by an absolute file name (usually with a `.a` or `.so` extension) or in the conventional `-l`*name* form. This option can be omitted if the library concerned is imported by the C++ command.

   `MPI_INC_DIR` is the directory where MPI include files are located. The `-mpiinc` argument may be omitted if the include directory is specified in the C++ command.

   Section 6 provides a complete guide to the `configure` arguments. This release includes various wrapper scripts called `configure.`*host* which illustrate the argument values we have used for to configure the software for various platforms. You may be able to choose the wrapper script for the software platform most similar to your own, and edit it. See section 5 for further details.

4. If `configure` completed successfully, build everything by typing:

   ```
   make
   ```

4 Using ad++

Add the `bin` directory in the Adlib home directory to your path, e.g.:

```
set path = ($path ~/Adlib/bin)
```

and the `man` directory to your `man`-path, e.g.:

```
setenv MANPATH ${MANPATH}:~/Adlib/man
```

The command

```
man ad++
```

tells you how to use the `ad++` command to compile an Adlib program.

The output of the compilation is an executable containing calls to the MPI library. The procedure for executing this program depends on your MPI platform.

For example, suppose that Adlib has been configured to use a P4-based installation of MPICH. If `test.C` is an Adlib program requiring 3 processors for its execution, it may be compiled and run by the commands:

```
ad++ test.C
mpirun -mr_pg -machinefile ~/machines -mr_np 3 a.out
```

where `machines` is a file in your current working directory or personal home directory, containing a list of host names upon which MPI processes can be spawned. For instance, if you want to allow a maximum of 4 MPI processes to be run on just one workstation called `weed`, the `machines` file might contain the lines:

```
weed
weed
weed
weed
```

5 Platforms on which Adlib has been installed

Table 1 lists the platforms on which Adlib has been installed so far. They are labelled by host name or site.

The release directory contains a set of wrapper scripts called `configure.`*host* which illustrate the options we used for these platforms. If your software platform is the same as any of those in Table 1, it should be possible to construct an appropriate set of `configure` options with minimal changes to the corresponding wrapper.

C++	MPI	architecture	host
GNU g++	MPICH/P4	Sun 4	bacchus
SunOS CC	”	”	nag
Solaris CC	”	”	uxa
GNU g++	CHIMP MPI	”	epcc
GNU g++	MPICH/P4	IBM RS6000	ben
AIX xlC	”	”	”
AIX xlC	MPICH/EUI	IBM SP2	sp2
”	Native MPI	”	sp2xx
DEC cxx	MPICH/P4	DEC Alpha	asterix
IRIX CC	MPICH/P4	SGI	pawnee-atm

Table 1: Some installation platforms

6 More on configuring

The information in this section is provided for completeness—it may or may not be useful to typical installers. In later releases we intend to provide a more intelligent configuration script, but first we have to gain experience manually configuring the system on as many platforms as possible.

What the `configure` script does

It creates the following files in the Adlib home directory:

```
src/adlib/Makefile
bin/ad++
```

They are constructed from templates, by filling in details about the local software platform. It also creates an empty directory, `lib`.

Once this has been done, the installation can be built by simply typing `make` in the Adlib home directory, which invokes the above `Makefile` to compile the library.

Arguments to `configure`

The usage of the `configure` script was given in section 3. This section gives more detail on the choice of the various fields in the arguments for `configure`.

`-c++com=C++_COM`

`C++_COM` is the command to execute the C++ compiler. It is invoked to compile the HPF run-time library when the installation is being built. The command will be applied to one C++ file at a time, with filename suffix `.C`.

It is also invoked by the `ad++` script to compile the code the users program, and link it with the Adlib library.

Often the `C++_COM` string will include `-L` options to tell the linker where to search for MPI libraries.

`-mpilib=MPI_LIB`

`MPI_LIB` is the file or list of files containing the MPI library. It is passed to the linker by the `ad++` script. The `-l`*name* form will also work, provided the compiler command knows where to find the named library—see the description of `C++_COM`.

This argument can be omitted if the MPI library is automatically imported by the compiler command.

`-mpiinc=MPI_INC_DIR`

`MPI_INC_DIR` is the directory where MPI include files are located. This argument affects compilation of the Adlib library, which occurs when the installation is being built.

This argument can be omitted if the MPI include directory is automatically searched by the C++ command.

`-cmpih`

This flag affects the compilation of the Adlib library. It forces `#include mpi.h` directives to be wrapped in an `extern "C"` linkage specifier. Normally this is unnecessary (it is omitted by default). Unfortunately some combinations of C++ compiler and MPI implementation require this wrapper, and (worse) some combinations refuse to accept it.