Instructions to Build MPAS

MPAS Development Team

May 2, 2012

Contents

Chapter 1

Summary

This document is meant to provide instructions on how to build MPAS. It includes required libraries, some example commands on how to build both them and MPAS.

Chapter 2

Requirements

This section lists requirements before building MPAS. It is meant to be a list of general requirements, i.e. not core specific.

The required libraries currently are:

- NETCDF
- PNETCDF
- PIO (parallel io)

The optional libraries are:

- PAPI
- Zoltan

Chapter 3

Library compilation

This section will detail where to find, and how to build the libraries required for MPAS.

3.1 NETCDF

3.1.1 Download

NETCDF can be downloaded using the following link:
`http://www.unidata.ucar.edu/downloads/netcdf/index.jsp`

Any version should be able to be used but tested versions include 3.6.2, 3.6.3, and 4.1.3.

3.1.2 Compile

In order to build NETCDF no environment variables need to be set up. **(On conejo with PGI modules, the make can fail unless you set environment variables:** `CC,CXX,F77,FC,F90.`**)** One has to navigate into the directory that NETCDF was extracted into, and the following steps can be followed.

```
./configure --prefix=/path/to/install/into
make
make check
make install
```

In the configure line, the prefix variable can be set to define the path you want the libraries located after make install is completed. The default path is /usr/local.

3.2 PNETCDF

3.2.1 Download

PNETCDF (Parallel NETCDF) can be downloaded using the following link:
`http://trac.mcs.anl.gov/projects/parallel-netcdf/wiki/Download`

Versions that have been tested are 1.2.0 and newer.

3.2.2 Compile

To make compiling PNETCDF easier, several environment variable can be defined. A list of these can be found below:

Variable	Value name
CC	C Compiler
CXX	C++ Compiler
FC	Fortran Compiler
MPICC	MPI C Compiler
MPIFC	MPI Fortran Compiler
MPIF77	MPI Fortran 77 Compiler
MPIF90	MPI Fortran 90 Compiler

After these variables are defined, on simply has to navigate into the directory that PNETCDF was extracted into, and use the following steps.

```
./configure --prefix=/path/to/install/into
make
make install
```

Again in the configure line, the prefix variable can be set to define the path you want the libraries located after make install is completed. The default path is /usr/local.

3.3 PIO

3.3.1 Download

PIO (parallel io) can be downloaded using the following link:
`http://code.google.com/p/parallelio/downloads/list`

Versions 1.2.6 and newer have been tested.

3.3.2 Compile

In order to compile PIO, several environment variables need to be defined. A list of these can be found below.

Variable	Value name
CC	C Compiler
CXX	C++ Compiler
FC	Fortran Compiler
MPICC	MPI C Compiler
MPIFC	MPI Fortran Compiler
MPIF77	MPI Fortran 77 Compiler
MPIF90	MPI Fortran 90 Compiler
NETCDF_PATH	Path to where NETCDF was installed
PNETCDF_PATH	Path to where PNETCDF was installed

Assuming PIO was extracted into a pio_root directory, one simply has to navigate into pio_root/pio after the variables are defined. Once inside this directory, the following steps can be used:

```
./configure
make
```

PIO doesn't support the prefix, or make install options. So after make is executed, the PIO libraries and include files are located in pio_root/pio.

One Linux desktop with PIO1.2.6, gfortan and openmpi was unable to handle lines longer than 132 characters. Set environmental variable `FFLAGS=-ffree-line-length-0`.

Chapter 4

MPAS Compilation

After all the libraries are compiled, MPAS can be compiled. Any variables defined on the make line for MPAS can also be defined as environment variables. It's up to the user what their preference is. An example of how to compile MPAS in general can be seen below.

```
make target CORE=core NETCDF=netcdf_path PNETCDF=pnetcdf_path PIO=pio_path
```

Where in this example, target defines a compiler set within the MPAS makefile, core defines the target core to compile, netcdf_path should be equivalent to the prefix used during netcdf compilation, pnetcdf_path should be equivalent to the prefix used during pnetcdf compilation, and pio_path should be the path to pio_root/pio from the pio compilation section.

Example targets include, but can change at any time:

- gfortran
- intel
- pgi
- g95
- xlf
- ftn

To get a full list of targets (although this might include other potential non-targets) the user can execute the following command from the root MPAS directory.

```
grep ":$" Makefile
```

Example cores include, but can change at any time:

- ocean
- sw (shallow-water core)
- hyd_atmos
- nhyd_atmos

To get a full list of available cores, the use can execute the following command from the root MPAS directory.

```
ls src/ | grep ‘‘core_’’
```

PGI compilers do not support `isNaN`. Comment it out in:

```
./src/core_ocean/mpas_ocn_time_integration.F
```

Assuming make succeeded, a new executable named core_model.exe should exist in the directory, which can be used to run MPAS.