

Como Compilar o Modelo MPAS - Benchmark - Projeto RISC

Como baixar o modelo MPAS

Baixando o código-fonte do repositório Git do MPAS, utilizando *branch* relativo a versão 6.3 do MPAS:

```
$ git clone --depth 1 --branch v6.3 https://github.com/MPAS-Dev/MPAS-Model.git
MPAS-Model_v6.3
$ cd MPAS-Model_v6.3
$ git switch -c branch_v6.3
```

Pré requisitos

O modelo MPAS faz uso de diversas bibliotecas para seu correto funcionamento. Todas elas são gratuitas e podem ser obtidas diretamente pela internet. As bibliotecas necessárias **e suas dependências** são:

- NetCDF-C: <https://www.unidata.ucar.edu/software/netcdf>
- NetCDF-Fortran: <https://www.unidata.ucar.edu/software/netcdf>
- Parallel-NetCDF: <https://parallel-netcdf.github.io>
- HDF5: <https://github.com/HDFGroup/hdf5>
- Parallelio: <https://ncar.github.io/Parallelio>

A maioria dos sistemas de computação possuem a opção de se carregar as bibliotecas via o comando `module load`

O MPAS é executado com comandos de MPI. Logo o sistema necessita estar preparado para isso e preferencialmente a construção das bibliotecas deve usar o proprio MPI que preferencialmente de ve estar compilado com o mesmo compilador. O modelo já foi testado com MPI da INTEL, MPICH, OpenMPI.

Compilando o modelo MPAS

Criar script para instalação do MPAS

```
$ vi make_mpas.sh
```

Contendo o conteúdo a seguir, definindo as variáveis de ambiente `NETCDF`, `PNETCF` e `PIO`, e executando comando `make` com alguns parâmetros a serem seguidos na durante a compilação do código-fonte. O cabeçalho do script explica o significado de cada parâmetro. Neste exemplo o MPAS será compilado em precisão simples (single).

```
#!/bin/bash
```

```
#Usage: make target CORE=[core] [options]
```

```
#Example targets:
```

```
# ifort
# gfortran
# xlf
# pgi
```

```
#Available Cores:
```

```
# atmosphere
# init_atmosphere
# landice
# ocean
# seaice
# sw
# test
```

```
#Available Options:
```

```
# DEBUG=true - builds debug version. Default is optimized version.
# USE_PAPI=true - builds version using PAPI for timers. Default is off.
# TAU=true - builds version using TAU hooks for profiling. Default is off.
# AUTOCLEAN=true - forces a clean of infrastructure prior to build new core.
# GEN_F90=true - Generates intermediate .f90 files through CPP, and builds with them.
# TIMER_LIB=opt - Selects the timer library interface to be used for profiling the model. Options are:
#                 TIMER_LIB=native - Uses native built-in timers in MPAS
#                 TIMER_LIB=gptl - Uses gptl for the timer interface instead of the native interface
#                 TIMER_LIB=tau - Uses TAU for the timer interface instead of the native interface
# OPENMP=true - builds and links with OpenMP flags. Default is to not use OpenMP.
# OPENACC=true - builds and links with OpenACC flags. Default is to not use OpenACC.
# USE_PI02=true - links with the PIO 2 library. Default is to use the PIO 1.x library.
# PRECISION=single - builds with default single-precision real kind. Default is to use double-precision.
# SHAREDLIB=true - generate position-independent code suitable for use in a shared library. Default is false.
```

```
export PIO= < full path of ParallelIO install folder >
export NETCDF= < full path of NetCDF-C install folder >
export PNETCDF= < full path of Parallel-NetCDF install folder >

#make -j 8 [gfortran|ifort|pgi|xlf] CORE=atmosphere USE_PIO2=true
PRECISION=single 2>&1 | tee make.output
make -j 8 gfortran CORE=atmosphere USE_PIO2=true PRECISION=single 2>&1 | tee
make.output
```

Executa o script

```
$ source make_mpas.sh

....

*****
MPAS was built with default single-precision reals.
Debugging is off.
Parallel version is on.
Papi libraries are off.
TAU Hooks are off.
MPAS was built without OpenMP support.
MPAS was built with .F files.
The native timer interface is being used
Using the PIO 2 library.
*****
o
```

A mensagem final acima informa que a compilação foi bem-sucedida e alguns dos parâmetros de instalação que foram empregados. Os seguintes executáveis devem ter sido gerados: `atmosphere_model` e `build_tables`, além do arquivo `make.output`, contendo a saída em tela da compilação. **É fundamental que os compiladores e bibliotecas sejam compatíveis, preferencialmente compilados com o mesmo compilador** para que não haja erros na montagem do modelo.

É recomendável copiar os executáveis gerados em um diretório a parte, preferencialmente com nome que remeta a alguma propriedade da compilação efetuada.

```
$ mkdir bin_single
$ cp build_tables atmosphere_model make.output bin_single/
```

Compilando com suporte a OpenMP

Tentar repetir a instalação, mas agora com ativando a opção com OpenMP (`OPENMP=true`). Antes, deve-se limpar a instalação atual:

```
$ make clean CORE=atmosphere
$ make -j 8 gfortran CORE=atmosphere OPENMP=true USE_PI02=true PRECISION=single
2>&1 | tee make.output
```

Se tudo correr bem, ao final irá aparecer a mensagem abaixo, semelhante a anteriormente vista

```
....

*****
MPAS was built with default single-precision reals.
Debugging is off.
Parallel version is on.
Papi libraries are off.
TAU Hooks are off.
MPAS was built with OpenMP enabled.
MPAS was built with .F files.
The native timer interface is being used
Using the PIO 2 library.
*****
```

Copiar para um novo diretório os executáveis recém criados:

```
$ mkdir bin_single_openmp
$ cp build_tables atmosphere_model make.output bin_single_openmp/
```

Compilando com suporte a OpenACC (GPU)

O compilador PGI/NVIDIA é o indicado para compilar a versão do código que implementa diretivas OpenACC no MPAS, que está presente no pacote NVIDIA HPC SDK (nvhpc: <https://developer.nvidia.com/hpc-sdk>).

No momento, a versão estável do código em OpenACC do MPAS está disponível no branch `atmosphere/v6.x-openacc` :

```
$ git clone --depth 1 --branch atmosphere/v6.x-openacc https://github.com/MPAS-Dev/MPAS-Model.git MPAS-Model_OpenACC.git
$ cd MPAS-Model_v6.x_openacc
$ git switch -c branch_v6.x-openacc
```

Relembrando que é as bibliotecas sejam compatíveis, preferencialmente compilados com o mesmo compilador, aqui no caso o compilador PGI/NVHPC:

```
$ vi make_mpas_openacc.sh
```

```
#!/bin/bash
```

```
export PIO= < full path of ParallelIO (compiled with PGI) install folder >
export NETCDF= < full path of NetCDF-C (compiled with PGI) install folder >
export PNETCDF= < full path of Parallel-NetCDF (compiled with PGI) install folder >
```

```
make -j 8 pgi CORE=atmosphere OPENACC=true USE_PIO2=true PRECISION=single 2>&1
| tee make.output
```

Executa o script de instalação:

```
$ source make_mpas_openacc.sh
```

```
....
```

```
*****
MPAS was built with default single-precision reals.
Debugging is off.
Parallel version is on.
Papi libraries are off.
TAU Hooks are off.
MPAS was built without OpenMP support.
MPAS was built with OpenACC enabled.
MPAS was built with .F files.
The native timer interface is being used
Using the PIO 2 library.
*****
```

Copiar para um novo diretório os executáveis recém criados:

```
$ mkdir bin_single_openacc  
$ cp build_tables atmosphere_model make.output bin_single_openmp/
```

Não há garantia que este passo-a-passo vá funcionar em seu sistema. Contudo serve de auxílio na instalação.