Como rodar o benchmark do MPAS

Obtenção do benchmark

O benchmark pode ser obtido a partir deste site da UCAR
https://www2.mmm.ucar.edu/projects/mpas/benchmark/

Há dados para as versões v5.2, v6.x e v7.0. Neste documento os exemplos foram preparados para trabalhar com dados da **versão v6.x**. Baixar os dados com resolução de 10 km.

```
$ wget https://www2.mmm.ucar.edu/projects/mpas/benchmark/v6.x/MPAS-
A_benchmark_10km_L56.tar.gz
$ tar zxvf MPAS-A_benchmark_10km_L56.tar.gz
$ cd MPAS-A_benchmark_10km_L56
$ ls -A1
CAM_ABS_DATA.DBL
CAM_AEROPT_DATA.DBL
GENPARM.TBL
LANDUSE.TBL
OZONE_DAT.TBL
OZONE_LAT.TBL
OZONE_PLEV.TBL
RRTMG_LW_DATA
RRTMG_LW_DATA.DBL
RRTMG_SW_DATA
RRTMG_SW_DATA.DBL
SOILPARM.TBL
VEGPARM.TBL
namelist.atmosphere
stream_list.atmosphere.diagnostics
stream_list.atmosphere.output
stream_list.atmosphere.surface
streams.atmosphere
x1.5898242.graph.info
x1.5898242.graph.info.part.1024
x1.5898242.graph.info.part.1536
x1.5898242.graph.info.part.16384
x1.5898242.graph.info.part.2048
x1.5898242.graph.info.part.3600
x1.5898242.graph.info.part.4096
x1.5898242.graph.info.part.512
x1.5898242.graph.info.part.6144
x1.5898242.graph.info.part.768
x1.5898242.graph.info.part.8192
x1.5898242.init.nc
```

Preparação para rodar o benchmark

Para executar o benchmark em paralelo, deve-se realizar uma preparação prévia. O arquivo `x1.5898242.init.nc` é o arquivo de malha, onde o número 5898242 corresponde ao número de células da malha, relativo a resolução de 10 km.

O arquivo `x1.5898242.graph.info` refere-se ao grafo associado a malha. Neste caso, o grafo possui 17694720 vértices e 11796480 arestas. Por exemplo, o arquivo `x1.26214422.graph.info.part.512` contém informação do particionamento do grafo, dividido em 512 sub-regiões. Ao ser executado o MPAS em paralelo com "p" processos MPI (mpi ranks), deve existir no mesmo diretório de submissão o arquivo `x1.5898242.graph.info.part.<p>`. Portanto, o benchmark de 10 km já vem habilitado para ser executado em paralelo com número de mpi ranks entre 36 e 16384 mpi ranks. Para rodar com outro número de processos MPI, pode-se gerar o arquivo particionado com a bliblioteca METIS, sobretudo quando não houver o arquivo correspondente ao número de processos MPI que deseja ser utilizado.

Alguns ambientes computacionais já possuem o pacote/módulo da biblioteca METIS instalados para gerar o arquivo e particionamento em quantas partições forem necessárias. Por exemplo, para criar o arquivo de particionamento para executar o MPAS com 256 processos MPI:

```
$ module load metis
$ gpmetis -minconn -contig -niter=200 x1.5898242.graph.info 256
```

Execução do benchmark do MPAS

O arquivo `namelist.atmosphere` configura os parâmetros de execução. O parâmetro que define o tempo de integração é dado por `config_run_duration = '5_00:00:00'`, ou seja, 5 dias.

namelist.atmosphere

```
&nhyd_model
    config_time_integration_order = 2
    config_dt = 60.0
    config_start_time = '2010-10-23_00:00:00'
    config_run_duration = '5_00:00:00'
    config_split_dynamics_transport = true
    config_number_of_sub_steps = 2
    config_dynamics_split_steps = 3
    config_h_mom_eddy_visc2 = 0.0
    config_h_mom_eddy_visc4 = 0.0
    config_v_mom_eddy_visc2 = 0.0
    config_h_theta_eddy_visc2 = 0.0
    config_h_theta_eddy_visc4 = 0.0
    config_v_theta_eddy_visc2 = 0.0
    config_horiz_mixing = '2d_smagorinsky'
    config_len_disp = 10000.0
    config_visc4_2dsmag = 0.05
    config_w_adv_order = 3
    config_theta_adv_order = 3
    config_scalar_adv_order = 3
    config_u_vadv_order = 3
    config_w_vadv_order = 3
    config_theta_vadv_order = 3
    config_scalar_vadv_order = 3
    config_scalar_advection = true
    config_positive_definite = false
    config_monotonic = true
    config_coef_3rd_order = 0.25
    config_epssm = 0.1
    config_smdiv = 0.1
/
&damping
    config_zd = 22000.0
    config_xnutr = 0.2
/
&io
    config_pio_num_iotasks = 0
    config_pio_stride = 1
/
&decomposition
    config_block_decomp_file_prefix = 'x1.5898242.graph.info.part.'
/
&restart
    config_do_restart = false
/
```

```
&printout
    config_print_global_minmax_vel = true
    config_print_detailed_minmax_vel = false
/
&IAU
    config_IAU_option = 'off'
    config_IAU_window_length_s = 21600.
/
&physics
    config_sst_update = false
    config_sstdiurn_update = false
    config_deepsoiltemp_update = false
    config_radtlw_interval = '00:10:00'
    config_radtsw_interval = '00:10:00'
    config_bucket_update = 'none'
    config_physics_suite = 'mesoscale_reference'
/
&soundings
    config_sounding_interval = 'none'
/
```

O arquivo `streams.atmosphere` configura quais campos (variáveis) serão gravadas em disco durante a execução do modelo. Por exemplo, exemplo abaixo, somente as variáveis diagnósticas serão gravadas `<stream name="diagnostics"`, e em um intervalo 3 horas entre as saídas `output_interval="03:00:00"`.

streams.atmosphere

```
<streams>
<immutable_stream name="input"
                  type="input"
                  io_type="pnetcdf,cdf5"
                  precision="single"
                  filename_template="x1.5898242.init.nc"
                  input_interval="initial_only" />

<immutable_stream name="restart"
                  type="input;output"
                  filename_template="restart.$Y-$M-$D_$h.$m.$s.nc"
                  input_interval="initial_only"
                  output_interval="none" />

<stream name="output"
        type="output"
        filename_template="history.$Y-$M-$D_$h.$m.$s.nc"
        output_interval="none" >

        <file name="stream_list.atmosphere.output"/>
</stream>

<stream name="diagnostics"
        type="output"
        filename_template="diag.$Y-$M-$D_$h.$m.$s.nc"
        output_interval="03:00:00" >

        <file name="stream_list.atmosphere.diagnostics"/>
</stream>

<stream name="surface"
        type="input"
        filename_template="x1.5898242.sfc_update.nc"
        filename_interval="none"
        input_interval="none" >

        <var name="sst" />
        <var name="xice" />
</stream>

<immutable_stream name="iau"
                  type="input"
                  filename_template="x1.5898242.AmB.$Y-$M-$D_$h.$m.$s.nc"
                  filename_interval="none"
                  packages="iau"
```

```
                  input_interval="initial_only" />

</streams>
```

Para fazer uma execução do benchmark do MPAS, por exemplo, com 256 processos MPI:

```
mpirun -np 256 ./atmosphere_model
```

Se a execução tiver sido bem-sucedida, no final haverá o arquivo `log.atmosphere.0000.out` mais os arquivos de saída `diag.$Y-$M-$D_$h.$m.$s.nc`:

```
$ ls -A1 diag.2010-10-2*
diag.2010-10-23_00.00.00.nc
diag.2010-10-23_03.00.00.nc
diag.2010-10-23_06.00.00.nc
diag.2010-10-23_09.00.00.nc
diag.2010-10-23_12.00.00.nc
diag.2010-10-23_15.00.00.nc
diag.2010-10-23_18.00.00.nc
diag.2010-10-23_21.00.00.nc
diag.2010-10-24_00.00.00.nc

...

diag.2010-11-02_00.00.00.nc
```

No final do arquivo `log.atmosphere.0000.out`, com conteúdo similar abaixo, onde constam diversas informações sobre o tempo de execução do modelo. Eestes tempos mostrados abaixo **NÃO SÃO** referência para este *benchmark*. Servem unicamente para ilustrar como os tempos devem aparecer no arquivo `log.atmosphere.0000.out`.

```
 *******************************************************
    Finished running the atmosphere core
 *******************************************************


  Timer information:
     Globals are computed across all threads and processors

  Columns:
     total time: Global max of accumulated time spent in timer
     calls: Total number of times this timer was started / stopped.
     min: Global min of time spent in a single start / stop
     max: Global max of time spent in a single start / stop
     avg: Global max of average time spent in a single start / stop
     pct_tot: Percent of the timer at level 1
     pct_par: Percent of the parent timer (one level up)
     par_eff: Parallel efficiency, global average total time / global max total
time


    timer_name                                          total       calls
min            max            avg      pct_tot   pct_par     par_eff
  1 total time                                     3774.92407         1
3774.91846     3774.92407     3774.92139   100.00       0.00       1.00
  2  initialize                                     150.40375         1
150.31792      150.40375      150.39342     3.98       3.98       1.00
  2  time integration                               3618.48730      1440
1.97318       33.40007       2.51205    95.86      95.86       1.00
  3   physics driver                                1794.53015      1440
0.47506       30.66323       1.01738    47.54      49.59       0.82
  4    calc_cldfraction                                37.72757       144
0.10280       0.30112        0.19196     1.00       2.10       0.73
  4    driver_sfclayer                                 38.91610      1440
0.00829       0.10979        0.01556     1.03       2.17       0.58
  5     Monin-Obukhov                                  37.74515      1440
0.00763       0.10173        0.01476     1.00      96.99       0.56
  4    Noah                                            398.69632      1440
0.00670       0.31040        0.10756    10.56      22.22       0.39
  4    PBL Scheme                                      28.18152      1440
0.01831       0.18569        0.01912     0.75       1.57       0.98
  5     YSU                                            18.03782      1440
0.01189       0.03441        0.01238     0.48      64.01       0.99
  4    GWDO_YSU                                         5.10650      1440
0.00311       0.01023        0.00333     0.14       0.28       0.94
  4    New_Tiedtke                                     69.10799      1440
```

```
0.03966        0.15385        0.04497     1.83       3.85       0.94
  3   atm_rk_integration_setup                              2.44509      1440
0.00148        0.26099        0.00164     0.06       0.07       0.96
  3   atm_compute_moist_coefficients                       5.66098      1440
0.00371        0.03462        0.00382     0.15       0.16       0.97
  3   physics_get_tend                                    19.40721      1440
0.01049        0.28351        0.01262     0.51       0.54       0.94
  3   atm_compute_vert_imp_coefs                           7.26115      4320
0.00157        0.06722        0.00164     0.19       0.20       0.98
  3   atm_compute_dyn_tend                               226.43816     12960
0.01394        0.25307        0.01695     6.00       6.26       0.97
  3   small_step_prep                                     14.93314     12960
0.00108        0.25423        0.00112     0.40       0.41       0.97
  3   atm_advance_acoustic_step                          160.02155     17280
0.00855        0.02139        0.00900     4.24       4.42       0.97
  3   atm_divergence_damping_3d                           11.98194     17280
0.00066        0.00524        0.00067     0.32       0.33       0.97
  3   atm_recover_large_step_variables                    43.91561     12960
0.00320        0.01631        0.00329     1.16       1.21       0.97
  3   atm_compute_solve_diagnostics                       72.96220     12960
0.00497        0.07662        0.00547     1.93       2.02       0.97
  3   atm_rk_dynamics_substep_finish                      25.99814      4320
0.00438        0.03459        0.00581     0.69       0.72       0.97
  3   atm_advance_scalars                                 56.72322      2880
0.01886        0.03387        0.01907     1.50       1.57       0.97
  3   atm_advance_scalars_mono                           105.81931      1440
0.06356        0.50133        0.07237     2.80       2.92       0.98
  3   atm_rk_reconstruct                                   3.24145      1440
0.00214        0.02676        0.00218     0.09       0.09       0.97
  3   microphysics                                       108.32672      1440
0.03956        0.09928        0.07134     2.87       2.99       0.95
  4    WSM6                                               88.15473      1440
0.02281        0.07295        0.05739     2.34      81.38       0.94
  3   atm_rk_summary                                     111.37351      1440
0.04168        0.29677        0.06442     2.95       3.08       0.83
  3   mpas update GPU data on host                        46.42823      1440
0.02295        0.07186        0.02970     1.23       1.28       0.92

 ----------------------------------------
 Total log messages printed:
    Output messages =                18973
    Warning messages =                   3
    Error messages =                     0
    Critical error messages =            0
 ----------------------------------------
```

As instruções para a versão v7.0 e superior seguem o mesmo padrão.

Execução do benchmark do MPAS com OpenACC (GPU)

Referência: GPU-enabled MPAS-Atmosphere (site) https://mpas-dev.github.io/atmosphere/OpenACC/running.html

Suportado com MPAS compilado com o compilador PGI/NVIDIA.

Configuração do ambiente

O número de processos MPI neste caso deverá ser atribuído para execuçao híbrida em GPU e em CPU, é determinado em tempo de execução por variáveis de ambiente. Antes de rodar o executável **atmosphere_model**, as duas variáveis de ambiente a seguir devem ser definidas previamente:

MPAS_DYNAMICS_RANKS_PER_NODE - o número de processos MPI por nó que executará o modelo em GPUs.

MPAS_RADIATION_RANKS_PER_NODE – o número de processos MPI por nó que executará os esquemas de radiação nas CPUs.

No momento, essas duas variáveis de ambiente devem ser definidas para números inteiros pares, pois o código pressupõe que cada nó tenha ao menos dois *sockets*, nos quais as tarefas de CPU e GPU devem ser distribuídas uniformemente. A soma de MPAS_DYNAMICS_RANKS_PER_NODE e MPAS_RADIATION_RANKS_PER_NODE deve ser igual ao número total de processos MPI em execução em cada nó.

Por exempo:

```
$ export MPAS_DYNAMICS_RANKS_PER_NODE = 4    # processamento em GPU
$ export MPAS_RADIATION_RANKS_PER_NODE = 12  # processamento em CPU
$ mpirun -np 16 ./atmosphere_model
```

E devem ter sido previamente gerados também os respectivos arquivos de partição `x1.5898242.graph.info.part.4` (para porcessamento em GPU) e `x1.5898242.graph.info.part.12` (para porcessamento em CPU).

Em geral, se o modelo for executado em ***n*** nós, dois arquivos de partição de malha serão necessários pelo modelo: um arquivo com partições (*n* × MPAS_DYNAMICS_RANKS_PER_NODE) e outro arquivo com partições (*n* × MPAS_RADIATION_RANKS_PER_NODE).

Por exemplo:

```
$ export MPAS_DYNAMICS_RANKS_PER_NODE = 4    # processamento em GPU
$ export MPAS_RADIATION_RANKS_PER_NODE = 12  # processamento em CPU
$ mpirun -np 64 ./atmosphere_model
```

A execução paralela sendo realizada em 4 nós, devem ter sido previamente gerados também os respectivos arquivos de partição `x1.5898242.graph.info.part.16` (para porcessamento em GPU) e `x1.5898242.graph.info.part.48` (para processaento em CPU).

Arquivos de saída da execução

Na versão padrão do MPAS, todos os *ranks* MPI participam de todos os cálculos de física e dinâmica, e apenas o *rank* 0 grava um arquivo de log por padrão. Quando diferentes partições do comunicador global MPI executam funções diferentes – seja computação de radiação em CPUs ou física e dinâmica de não radiação em GPUs – cada uma das duas funções criará um arquivo de *log*.

O arquivo `log.atmosphere.role01.0000.out` contém mensagens de log do *rank* 0 do intracomunicador MPI rodando em GPUs, enquanto o arquivo `log.atmosphere.role02.0000.out` contém mensagens de log do *rank* 0 do intracommunicator MPI rodando em CPUs .