DOI: 10.5902/2179460X11600
Revista do Centro de Ciências Naturais e Exatas - UFSM
Revista Ciência e Natura, Santa Maria
EISSN: 2179-460X, Edição Esp. Dez. 2013, p. 216 - 218

TURBULENT PARAMETERIZATION FOR CCATT-BRAMS BY GPU

Renata S. R. Ruiz[1], Haroldo F. de Campos Velho[1], Leandro S. Lessa[1], Andrea S. Charão[2]

[1]Instituto Nacional de Pesquisas Espaciais (INPE), Brazil.
[2]Universidade Federal de Santa Maria (UFSM), Brazil

renata@la.inpe.br, Haroldo@lac.inpe.br, Leandro_santos_lessa@yahoo.com.br, andrea@inf.ufsm.br

ABSTRACT
A strategy to increase the computer code performance is to employ of hybrid computing, combining CPU with GPU or/and FPGA. Two turbulent parameterizations, Smagorinsky and Mellor-Yamada, were codified on GPU for the CCATT-BRAMS code, using the CUDA standard. The results show a good speed-up for the model.

RESUMO
Uma estratégia para aumentar o desempenho de códigos computacionais é o uso de computação híbrida, que combina CPU com GPU e/ou FPGA. Duas parametrizações de turbulência, Smagorinsky e Mellor-Yamada, foram codificadas em GPU para o código CATT-BRAMS:. O padrão CUDA foi usado na implementação. Os resultados mostram um aumento significativo no speed-up do modelo.

1. INTRODUCTION
The environmental prediction model CCATT-BRAMS (Freitas et al., 2009; 2013) is a complex computer code, and it is operationally employed by the CPTEC-INPE. This model allows atmospheric simulation with emission, transport, pollutant dispersion, and chemical reactions. The CCATT-BRAMS is already prepared to run on parallel machines. But, depending on the model resolution, this is a very expensive code. This motivates the research to improve its performance. Two parameterizations in the CCATT-BRMAS are codified on GPU: Smagorinsky (1963), and Mellor-Yamada (1982).

1.1 Smagorinsky's parameterization
The turbulent fluxes are parameterized using the theory of gradient-flux. The Reynolds tensors are given as following:

$$\overline{u'_i u'_j} = -(K_m)_{ij}(D)_j \quad (1a)$$

$$(D)_j = \partial u_i/\partial x_j + \partial u_j/\partial x_i \quad (1b)$$

$$K_m = (cs_z \Delta z)^2 \left[|D_v| + H(N)\right] f(R_i) \quad (1c)$$

where cs_z is a fitting coefficient, Δz is the discretization for vertical grid, and $|D_v|$ is the magnitude of the deformation tensor for vertical direction. $H(N)$ and $f(R_i)$ are given by

$$H(N) = \sqrt{\max(0, -N^2)} \quad (2a)$$

$$f(R_i) = \max\{0\ ;\ 1-(K_h/K_m)R_i\} \quad (2b)$$

DOI: 10.5902/2179460X11600
Revista do Centro de Ciências Naturais e Exatas - UFSM
Revista Ciência e Natura, Santa Maria
EISSN: 2179-460X, Edição Esp. Dez. 2013, p. 216 - 218

with N being the Brunt-Väisälä frequency, and (K_h/K_m) is the ratio between the heat and momentum eddy diffusivities.

1.2 Mellor-Yamada's parameterization

This is a 2.5 closure scheme for turbulence, where the third order tensors are parameterized, with a prognostic equation for the turbulent kinetic energy (e):

$$\frac{\partial \overline{\psi'_i \phi'_j}}{\partial t} = f\left(\overline{\psi'_i \phi'_j \zeta'_k},\ \overline{\psi'_i \phi'_j},\ \psi_m, t\right)$$

$$\overline{\psi'_i \phi'_j \zeta'_k} \approx K_p \nabla \overline{\psi'_i \phi'_j} \tag{3a}$$

$$\psi, \phi, \zeta = u, v, w, q, \theta \tag{3b}$$

where $K_p = S_p l\sqrt{2e}$, $l = \dfrac{\kappa(z+z_0)}{1+\kappa(z+z_0)/l_\infty}$, $l_\infty = 0.1\dfrac{\int_0^d z\sqrt{2e}\,dz}{\int_0^d \sqrt{2e}\,dz}$ (wind components, moisture, and temperature). Key parameters are a length scale l, and TKE (e):

(4)

where S_P are constants suggests by Mellor-Yamada (1982).

2. CUDA PROGRAMMING ON GPU

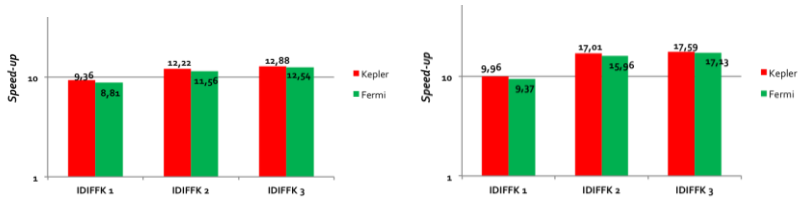
CUDA (Compute Unified Device Architecture) (Nvidia: CUDA, 2012) was developed for the parallel processing for the GPUs produced Nvidia Corporation. The CUDA programming could be understood as an extension of languages like C, C++, and Fortran, adding qualifiers to functions and data, producing kernels for execution. The job on a kernel is to be divided among thousands of threads, organized into blocks, and grids, with dimensions *blockDim* and *gridDim*. The kernel uses indexes *blockIdx* and *threadIdx* for defining the job to a thread.

The turbulence routines were codified on Nvidia GPUs: Fermi GTX-580 (512 cores), and Kepler GTX-680 (1536 cores).

3. RESULTS AND FINAL COMMENTS

Two horizontal resolutions were evaluated: 40 km (grid points number: 95×48×38), and 20 km (194×100×40). Figure 1 shows the speed-up results to Smagorinsky's approach for both resolutions, and for Fermi and Kepler architectures.

DOI: 10.5902/2179460X11600
Revista do Centro de Ciências Naturais e Exatas - UFSM
Revista Ciência e Natura, Santa Maria
EISSN: 2179-460X, Edição Esp. Dez. 2013, p. 216 - 218



(a) (b)

Figure 1: GPU execution for Smagorinsky's approach under two resolutions: (a) 40 km, (b) 20 km.

Similarly, the speed-up for the Mellor-Yamada scheme is presented in Figure 2, also considering two resolutions, and Fermi and Kepler architectures performance.

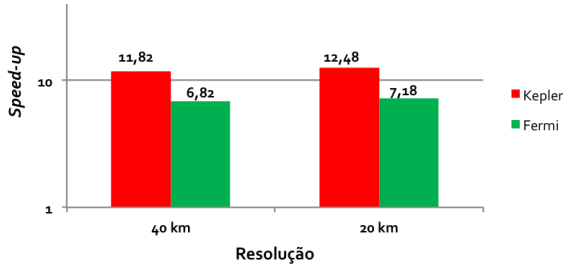


Figure 2: GPU execution for Mellor-Yamada's approach under two resolutions: 40 km, and 20 km.

For the lower resolution worked here, similar speed-up results were obtained for both GPU architectures – the gain for the Mellor-Yamada approach has superior speed-up than the Smagorinsky's parameterization. The results show a tendency to improve the speed-up for a finer resolution.

5. REFERENCES

FREITAS, S. R. Et al. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system (CATT-BRAMS). part 1: Model description and evaluation. **Atmos. Chem. Phys. Discuss.**, v. 9, n. 8**,** p. 2843–2861, 2009.

LONGO, K. et al. The chemistry CATT-BRAMS model (CCATT-BRAMS 4.5): a regional atmospheric model system for integrated air quality and weather forecasting and research. **Geosci. Model Dev**., v. 6, 1173–1222, 2013.

SMAGORINSKY, J. General circulation experiments with the primitive equations. I. The Basic Experiment. **Mon. Weath. Rev.**, v. 91, 99–164, 1963.

MELLOR, G. L., YAMADA, T. Development of a turbulence closure model for geophysical fluid problems. **Reviews of Geophysics and Space Physics**, v.20, n. 4, 851–875, 1982.

NVIDIA Corporation: CUDA Toolkit Documentation, 2012, available at: http://docs.nvidia.com/cuda/. Accessed: July-2013.