

Auditoria de código e planejamento de otimização em ambiente HPC para modelos de Previsão Numérica de Tempo

Isabel de Freitas Barboza^{1,2}, Eduardo Lucio Mendes Garcia², Roberto Pinto Souto²

¹Faculdade de Educação Tecnológica do Estado do Rio de Janeiro (CPTI/FAETERJ)

²Laboratório Nacional de Computação Científica (LNCC)

{ifreitas, bidu, rpsouto}@lncc.br, ibarboza@faeterj-petropolis.edu.br

Abstract. In general, Numerical Weather Prediction (NWP) models require intensive computations during their execution. These applications depend on High Performance Computing (HPC) resources, which highlights the need for auditing, performance analysis, and optimization to make their codes more efficient, enabling more detailed and timely forecasts. The objective of this work is to present the performance analyses and computational efficiency of the Model for Prediction Across Scales (MPAS), which serves as the foundation and preparation for experiments with the Model for Ocean-LaNd-Atmosphere prediction (MONAN), currently under development by INPE in partnership with MCTI, with LNCC contributing to the optimization of its codes.

Resumo. De modo geral, os modelos de Previsão Numérica de Tempo (PNT), demandam muitos cálculos durante a sua execução. Essas aplicações dependem de recursos de High Performance Computing (HPC), o que reforça a necessidade de auditoria, análise de desempenho e otimização, de modo a tornar seus códigos mais eficientes permitindo previsões detalhadas e antecipadas. O objetivo deste trabalho é apresentar as análises de desempenho e a eficiência computacional do Model for Prediction Across Scales (MPAS), que serve de base e preparação para os experimentos com o Model for Ocean-LaNd-Atmosphere predictioN (MONAN), atualmente em desenvolvimento pelo INPE, em parceria com o MCTI, com participação do LNCC na otimização de seus códigos.

1. Introdução

Fazer previsões de tempo mais detalhadas intensifica o uso de recursos de HPC, capazes de realizar simulações complexas em tempo adequado. Entretanto, esse uso também traz desafios, como a necessidade de otimização dos códigos e análise de desempenho para garantir o uso eficiente dos recursos computacionais.

Nesse contexto, destaca-se o MONAN, modelo atmosférico em desenvolvimento pelo Ministério da Ciência, Tecnologia e Inovação (MCTI) e pelo Instituto Nacional de Pesquisas Espaciais (INPE), com foco em fornecer previsões de alta qualidade para a região tropical da América do Sul. O MONAN utiliza como base o núcleo dinâmico do modelo MPAS [Skamarock et al. 2012], desenvolvido originalmente pelo *National Center for Atmospheric Research* (NCAR)¹ e Los Alamos National Laboratory (LANL)².

¹<https://ncar.ucar.edu/>

²<https://www.lanl.gov/>

Assim, todos os experimentos realizados neste trabalho com o MPAS têm caráter preparatório, servindo como etapa de aprendizado e de validação das metodologias de análise de desempenho e otimização que futuramente serão aplicadas ao MONAN. O objetivo principal é compreender o comportamento computacional do MPAS no ambiente de HPC, identificando gargalos, avaliando estratégias de paralelização e propondo melhorias que contribuam para o desenvolvimento e a eficiência do MONAN.

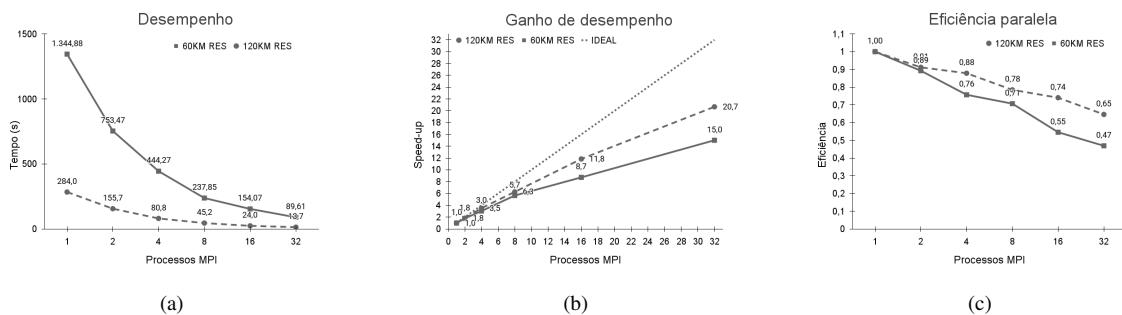
2. Metodologia e Ferramentas

Inicialmente, o modelo foi executado sem o uso de ferramentas específicas, como mostra a Figura 1. Esses resultados fornecem uma visão preliminar do comportamento do modelo. A fim de se obter uma análise mais detalhada, foram empregadas as ferramentas **Scalasca** [Adhianto et al. 2010] e **HPCToolkit** [Geimer et al. 2010] que analisam custos de execução, ineficiências e escalabilidade de programas paralelos, utilizando as ferramentas de visualização CubeGUI³ e HPCviewer⁴.

Por meio desta visualização, puderam ser obtidas as métricas de desempenho do *Performance Optimisation and Productivity Centre of Excellence in HPC*⁵ (POP-COE), importantes para quantificar o impacto dos fatores de paralelização, permitindo comparar o desempenho paralelo e identificar ineficiências nos códigos.

3. Resultados Parciais e Discussão

Os experimentos foram executados no supercomputador Santos Dumont, em nós da partição Sequana X1000 usando somente CPU⁶. São mostrados nesta seção os resultados das execuções do modelo MPAS com resoluções da malha em 120 km e 60 km, considerando simulações de 4 horas e 2 horas, rodando com 1, 2, 4, 8, 16 e 32 núcleos de processamento.



que a execução de 60 km corresponde a uma simulação de 2h. Essa normalização permite ter uma ideia mais clara da diferença de carga de trabalho entre as duas configurações.

Pode-se observar que ambas as resoluções, o tempo de execução diminui com o aumento do número de núcleos. Entretanto, observa-se queda gradual da eficiência à medida que mais processos são utilizados, sendo essa perda mais acentuada no caso de 60 km, devido ao maior uso de recursos computacionais, desbalanceamento de carga e intensificação da comunicação entre processos.

Com as ferramentas Scalasca e HPCToolkit, juntamente com as métricas do POP-COE, temos análises mais detalhadas do desempenho. Na Figura 2, o HPCviewer mostra a linha do tempo da execução do MPAS. Cada faixa horizontal representa um processo e cada bloco colorido corresponde ao tempo gasto em uma função específica durante a execução. Observa-se que determinados blocos apresentam durações distintas entre os processos, indicando desequilíbrios de carga e diferenças no tempo de CPU entre eles. Esses desbalanceamentos contribuem para a redução da eficiência paralela e ajudam a explicar os resultados da Figura 1 (c).

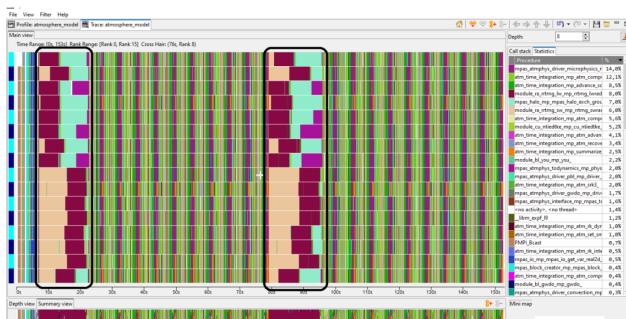


Figura 2. HPCViewer: rodada MPAS 60km com 16 processos MPI

Esses desequilíbrios decorrem principalmente do ciclo diurno: como a simulação é global, diferentes regiões do planeta estão sujeitas a condições de dia ou noite, o que gera variação no tempo de execução das rotinas relativas à radiação de onda curta e longa, além de sobrecarga de comunicação entre processos.

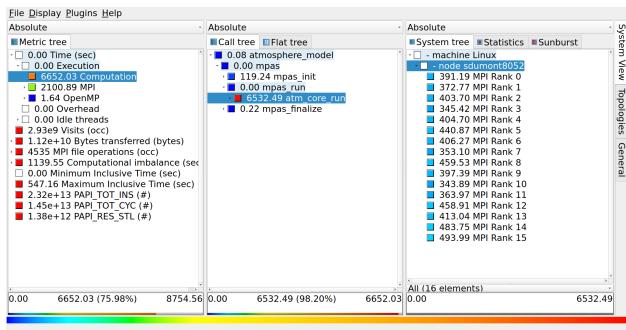


Figura 3. CubeGUI: rodada MPAS 120km com 16 processos MPI

A Figura 3 mostra a análise do MPAS no CubeGUI, organizada em três partes: **Metric tree**, que mostra o tempo integrado das rotinas da aplicação (*Computation*), das rotinas relativa às bibliotecas *MPI* e *OpenMP*; **Call tree**, que detalha a hierarquia de chamadas de rotina; e **System Tree**, que mostra o mapeamento dos processos MPI nos nós

de execução. Observa-se que parte significativa do tempo total é dedicada à comunicação (*MPI*) e sincronização entre processos, o que confirma a presença de gargalos e desbalanceamento na carga computacional. As diferenças entre os tempos de execução dos ranks MPI com variação de aproximadamente 370s a 500s podem indicar que alguns processos esperam outros finalizarem suas tarefas, reduzindo eficiência paralela.

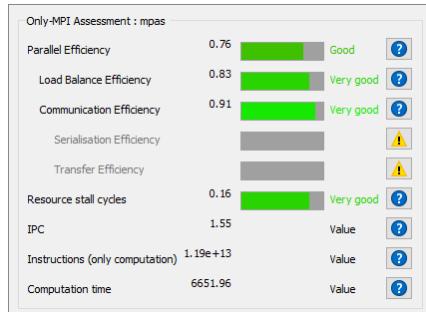


Figura 4. CubeGui: métricas rodada MPAS 120km com 16 processos MPI

A Figura 4 apresenta a avaliação do experimento anterior segundo as métricas POP-COE. O modelo obteve **Eficiência Paralela de 0,76**, considerada boa, refletindo desempenho razoável em múltiplos processos. A **Eficiência de Balanceamento de Carga (0,83)** indica distribuição adequada do trabalho, enquanto a **Eficiência de Comunicação (0,91)** mostra boa capacidade de troca de dados entre processos, apesar de sobrecargas pontuais. O valor de **Resource Stall Cycles (0,16)** sugere que o processador permaneceu ocioso por curtos períodos, o que reforça o impacto moderado das esperas MPI.

4. Conclusão

Este trabalho destacou a importância da HPC para a PNT, apresentando análises de desempenho do modelo MPAS. Observou-se boa escalabilidade inicial, porém queda de eficiência com o aumento do número de processos, indicando limitações de escalabilidade. As análises com Scalasca, HPCToolkit e POP-COE revelaram gargalos de carga e sobrecarga de comunicação MPI. Como melhorias futuras, destacam-se ajustes no balanceamento de carga, otimizações de comunicação e exploração mais eficiente do paralelismo. Esses resultados contribuem para direcionar futuras otimizações no modelo MONAN, visando maior eficiência computacional e melhor desempenho nas simulações atmosféricas.

Referências

- Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., and Tallent, N. R. (2010). Hpctoolkit: tools for performance analysis of optimized parallel programs <http://hpctoolkit.org>. *Concurr. Comput.: Pract. Exper.*, 22(6):685–701.
- Geimer, M., Wolf, F., Wylie, B. J. N., Ábrahám, E., Becker, D., and Mohr, B. (2010). The scalasca performance toolset architecture. *Concurr. Comput.: Pract. Exper.*, 22(6):702–719.
- Skamarock, W. C., Klemp, J. B., Duda, M. G., Fowler, L. D., Park, S.-H., and Ringler, T. D. (2012). A multiscale nonhydrostatic atmospheric model using centroidal voronoi tessellations and c-grid staggering. *Monthly Weather Review*, 140(9):3090 – 3105.