

# Avaliação de Desempenho modelo ETA

03 de setembro de 2007

**Daniel Merli Lamosa**

## 1. Criação do Ambiente de Teste

Iniciou-se a criação do ambiente de Teste para o modelo **Eta Workstation** através do arquivo *set\_parmeta* contendo as configurações para a grade 199x299x38 centrada na posição (-23°, -45°) e com resolução de 5Km. O modelo foi configurado para duas rodadas na mesma configuração, mas com tempos de integração diferentes. Na primeira rodada foi usado um tempo de integração de 6h e na segunda rodada o tempo considerado foi de 96h. Em ambos os casos o *time step* foi de 10 segundos.

## 2. Geração de Scripts para a Tomada de Tempo

Para a tomada de tempo foi utilizado o comando *date* (usando o tempo total de processamento), pois deseja-se determinar uma primeira medida de tempo para identificar prováveis “gargalos” no processamento do modelo.

Foram editados os *scripts*, localizados no diretório *caminho/worketa/eta/scripts*, ***new\_prep.sh\_Templ***, ***run.com\_real\_mpi*** e ***outjob\_special.ksh*** para a criação de um arquivo chamado ***relatorio.txt***, localizado no diretório de trabalho (*caminho/worketa/etc/.199X249X38\_Xproc*, X = número de processos). No arquivo *new\_prep.sh\_Templ* foram considerados como tempo de pré-processamento os arquivos *select.x*, *etatopo.exe*, *vegmsk.x* e *initbc.exe*. A soma dos tempos de cada arquivo resulta no tempo total de pré-processamento. A medida de tempo do modelo se dá pelo arquivo *etafcst\_all.x* localizado no script *run.com\_real\_mpi* e o tempo de pós-processamento foi obtido da soma dos tempos de *outjob\_special* e *Finaliza.ksh*.

Os tempos obtidos são calculados em segundos e o formato do arquivo *relatorio.txt* é mostrado a seguir.

```

Modelo ETA: Relatorio de tempos #####

Tempos de pre-processamento -----
Tempo de select.x = xxxxs
Tempo de etatopo.x = xxxxs
Tempo de vegmsk.x = xxxxs
Tempo de initbc.x = xxxxs

Tempo total de pre-processamento = xxxxs
# -----

# Tempo de processamento do modelo -----
Modelo etafcst_all.x = xxxxs
# -----

# Tempo de pos-processamento do modelo -----
Tempo de outjob_special = xxxxs
Tempo de Finaliza = xxxxs

Tempo Total de pos-processamento = xxxxs
# -----

# #####

```

### 3. Resultados Computacionais Encontrados

Para os dois tempos de integração foram usados os mesmos números de processadores visando uma comparação do comportamento do modelo. Foram usados nas rodadas os seguintes números de processadores: 6 (5+1), 11 (10+1), 21 (20+1), 41 (40+1), 61 (60+1), 81 (80+1), 101 (100+1), 201 (200+1), 301 (300+1), 401 (400+1), 501 (500+1), 601 (600+1), 701 (700+1), 801 (800+1), 901 (900+1) e 1001 (1000+1). Note-se que o número de processadores é formado pela soma do número de processadores voltado a integração do modelo e do número de processadores responsável por I/O (leitura e escrita em arquivos), que no caso é sempre 1. Deve-se lembrar que este esquema é o atualmente adotado para o uso do modelo.

A seguir é mostrado os tempos de processamento para a integração de 6h.

## Tempos de Processamento: Eta 6h

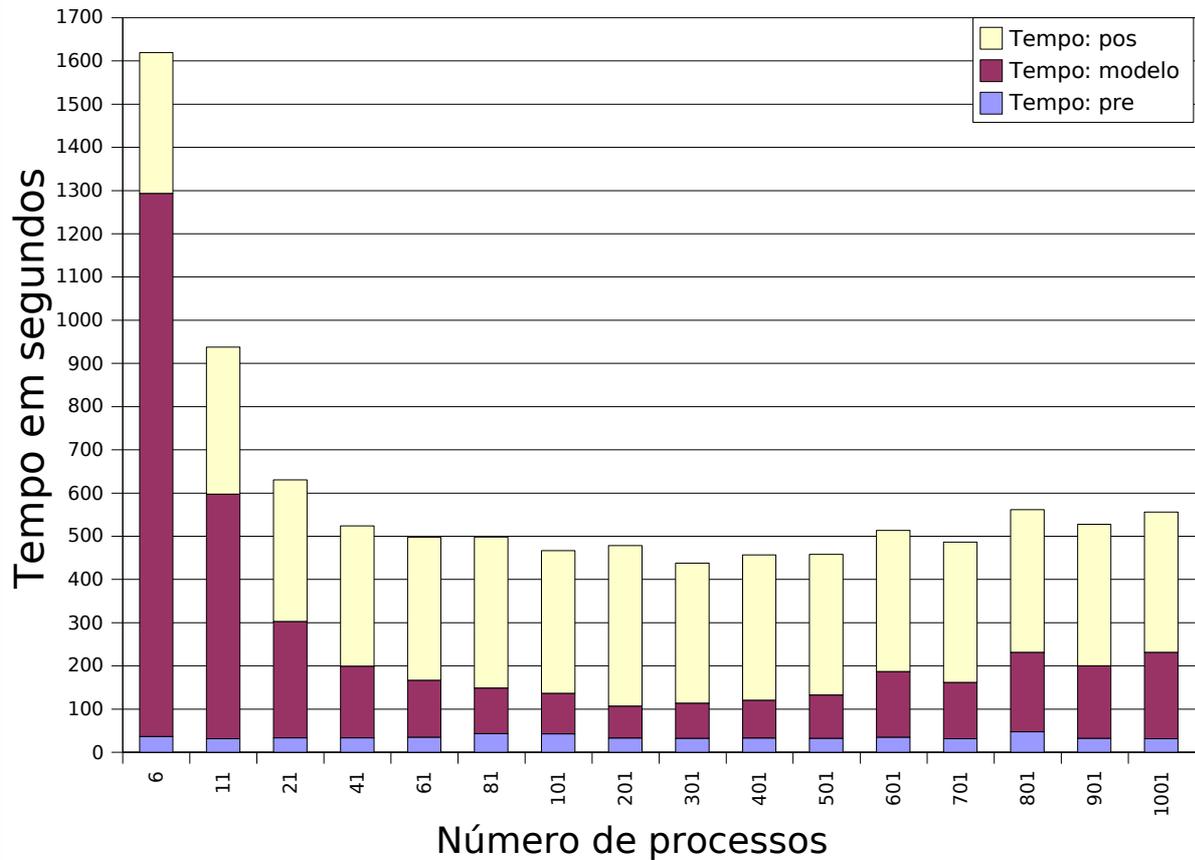


Figura 1: Tempos de processamento eta 6h

Percebe-se, na Figura 1, que o tempo de integração do modelo cai de uma forma acentuada com poucos processadores e se estabiliza entre 81 e 501 processadores. Também é observado um leve aumento do tempo de integração a partir de 601 processadores. Os tempos de pré-processamento e pós-processamento permanecem praticamente constantes devido ao ambos serem executados em modo serial.

Na Figura 2 é apresentado os tempos de processamento do modelo com tempo de integração de 96h.

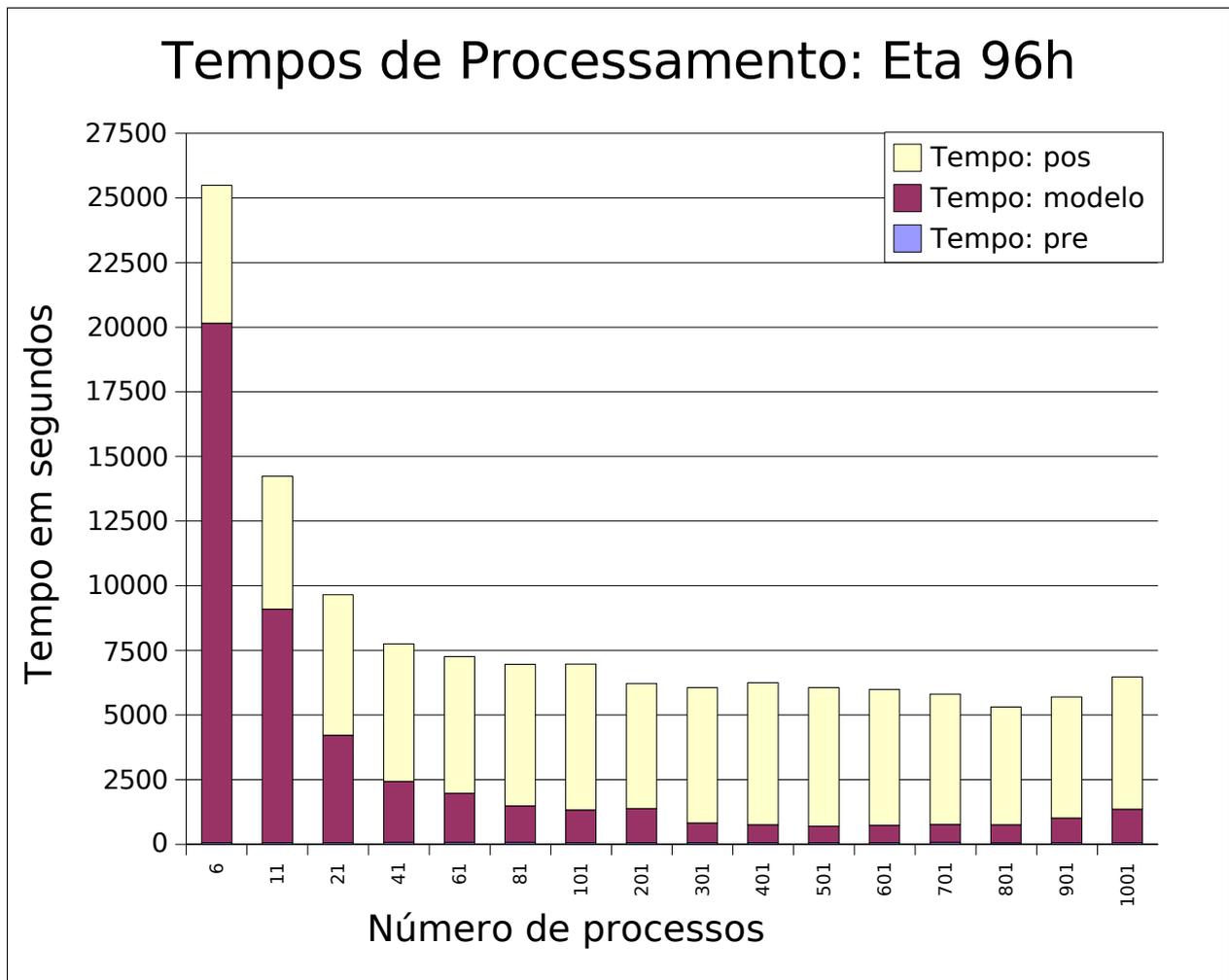


Figura 2: Tempos de processamento eta 96h

Percebe-se, na Figura 2, que o mesmo comportamento encontrado com um menor tempo de integração é obtido para a execução do modelo. A partir de 101 até 801 processadores ocorre uma estabilização do tempo computacional. Também é possível notar que o tempo de pré-processamento praticamente não influencia no tempo total. Para o caso do pós-processamento, o tempo computacional passar a ser decisivo no computo geral.

Na Figura 3, é apresentado um gráfico com o *speedup* para a integração de 6h, uma medida de desempenho largamente usada como um meio de comparação entre diferentes rodadas. O *speedup* mede quantas vezes um programa implementado com ferramentas que possibilitam o paralelismo é mais rápido que uma rodada de referência. O padrão é rodar em modo serial e comparar com as diversas rodadas em paralelo

executadas variando-se o número de processadores. Neste caso, considerou-se que a rodada padrão de teste continha 5 processadores. Para o caso ideal, se utilizássemos 10 processadores o *speedup* deveria ser o dobro (10 / 5). Também deve-se notar que a comparação de tempos de processamento é apenas da integração do modelo, ou seja, os tempos de pré e pós-processamento não são considerados. Além disso, o processador responsável pela parte de I/O também não é considerado.

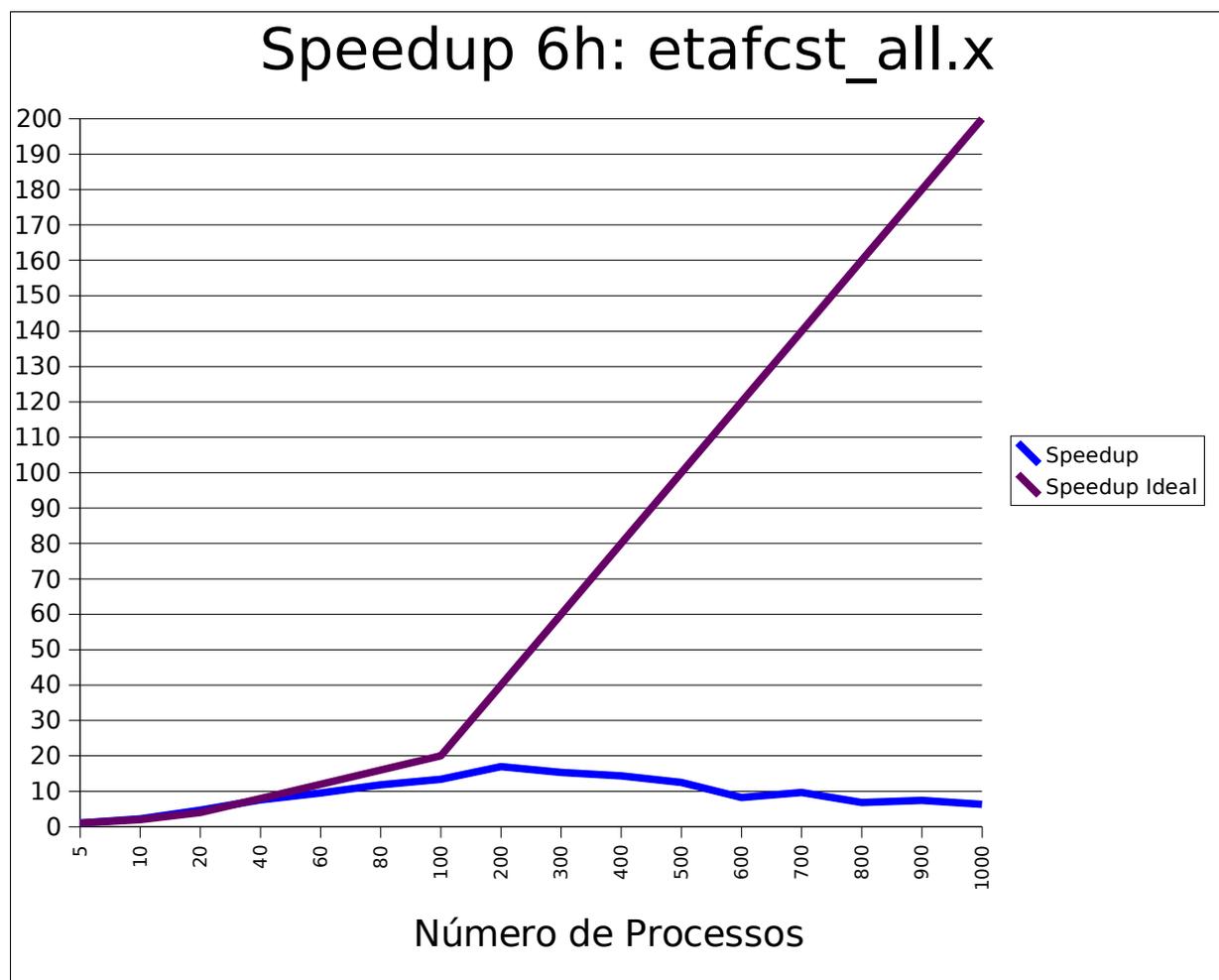


Figura 3: *Speedup* eta 6h

Na Figura 3, é possível observar que o speedup para 20 processadores ultrapassa o valor ideal. É provável que esse comportamento ocorra devido a função de medição de tempo adotada (date) não isolar apenas o *CPU time* (tempo efetivo de processamento), pois este leva em consideração o tempo total de processamento, incluindo o tempo que o sistema atende outro processo da máquina, tempo de gerenciamento do sistema, tempo de processamento do modelo, etc. Provavelmente o tempo com 10 processos tenha sofrido um acréscimo de tempo pelos fatores listados anteriormente. Embora o desempenho inicial seja próximo do ideal, a medida que o número de processos cresce o desempenho cai se estabilizando em 200. A partir desse ponto, o desempenho passa a ser extremamente ruim.

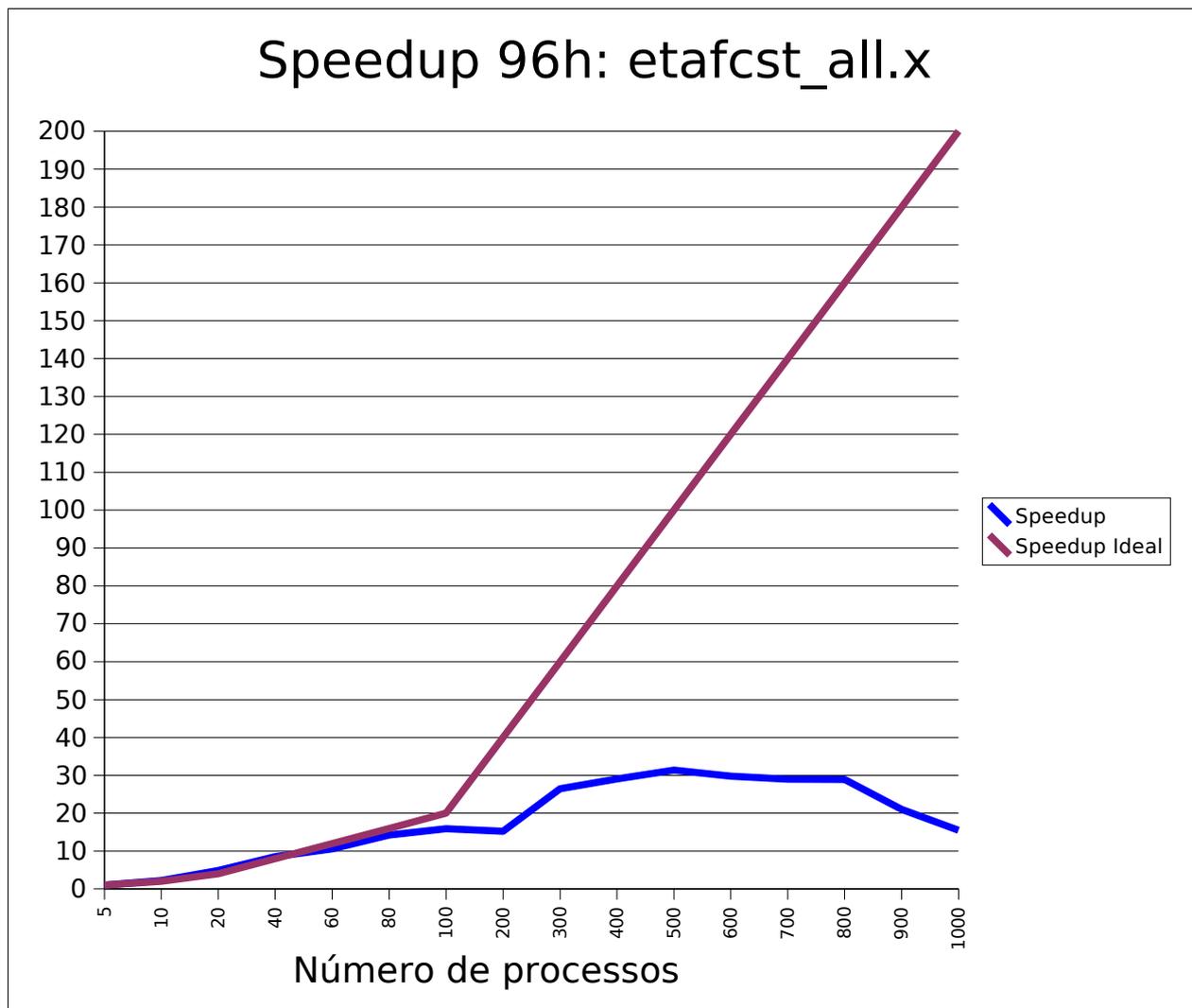


Figura 4: Speedup eta 96h

Na Figura 4, percebe-se que o mesmo comportamento inicial comparada a figura anterior. Devido a quantidade de informações processadas os valores encontrados crescem comparado a Figura 3, mas a curva de desempenho é próxima. Também é nítido um pequeno aumento no desempenho a partir do uso de 200 processadores. Porém, a tendência de queda se mostra com mais processadores.

#### 4. Conclusão

Dois pontos chaves foram considerados promissores para uma melhoria no desempenho do modelo: O alto tempo de pós-processamento comparado ao tempo de processamento; e a queda no desempenho com grande número de processadores.

Resolver o problema de pós-processamento parece promissor, pois, para um tempo maior de integração, esse tempo é decisivo. Contudo, o tempo de processamento do modelo não será alterado e o baixo desempenho não seria melhorado.

Serão necessários mais testes de desempenho com maior precisão e para grades bem maiores para identificar se o desempenho do modelo será realmente decisivo.

Maiores informações podem ser obtidas no endereço: [www.cptec.inpe.br/pad](http://www.cptec.inpe.br/pad)