

Aumentando a Escalabilidade de um Cluster com a versão paralela do *Ocean-Land-Atmosphere Model*

André Luiz Lima da Costa¹, Josemar Rodrigues de Souza^{1,2}

¹Centro Integrado de Manufatura e Tecnologia (SENAI-CIMATEC) – Salvador, BA – Brasil

²Universidade do Estado da Bahia (ACSO/UNEB) – Salvador, BA – Brasil
{andreluiz.bsi, josemarsbr}@gmail.com

Resumo. *A escalabilidade em clusters utilizando Message Passing Interface (MPI) vem sendo uma das maiores dificuldades para se obter melhores resultados de processamento massivo na Computação de Alto Desempenho. Com a versão paralela do Ocean-Land-Atmosphere Model (OLAM) não é diferente, pois sua escalabilidade é limitada a uma determinada quantidade de processadores. Diante deste problema o autor explana neste artigo como pretende, ao final de sua dissertação de mestrado, aumentar a escalabilidade do OLAM.*

1. Introdução

A Computação de Alto Desempenho vem sendo largamente utilizada com a finalidade de resolver problemas que exigem um alto poder de processamento. Com a limitação de *hardware* para atender demandas cada vez maiores que chegam à ordem de um trilhão de operações por segundo, tem se adotado a Computação Paralela como uma possível solução para suprir esta necessidade. Na Computação Paralela é utilizado o conceito de “Dividir para Conquistar”, no qual um problema principal é dividido em problemas menores e distribuído em diferentes máquinas de um *cluster* para serem processados paralelamente.

Porém, mesmo com a possibilidade de se utilizar diversas máquinas trabalhando juntas para resolverem um mesmo problema, a Computação Paralela tem uma limitação de processamento decorrente de sua deficiência em relação à escalabilidade. Já que a partir de determinada quantidade de máquinas o *cluster* não apresenta um bom *SpeedUp* devido ao alto processamento necessário para gerenciar a comunicação entre os seus nós. Um problema que exemplifica a utilização da Computação Paralela, mas que tem sua escalabilidade limitada são os modelos de simulação atmosférica utilizados na previsão meteorológica, pois necessitam de um alto poder de processamento para realizar cálculos massivos para prever fenômenos climáticos em tempo real.

Diante deste problema, será abordado nesse artigo como o autor pretende, em sua dissertação de mestrado em modelagem computacional, aumentar a escalabilidade da versão paralela do Ocean-Land-Atmosphere Model (OLAM), com o intuito de obter da comunidade científica o seu parecer sobre a pesquisa.

2. Ocean-Land-Atmosphere Model - OLAM

O *Ocean-Land-Atmosphere Model* (OLAM) é um modelo computacional criado para realizar previsões atmosféricas através da simulação numérica desenvolvida pela Universidade Americana Duke [Silva 2009]. Antes do OLAM, existiam dois tipos de modelos de simulação atmosférica: o global e o regional, onde (SUBSTITUIR O “ONDE” POR “NO QUAL” o modelo global tem como escopo realizar a simulação em toda a área da terra e não pode ser utilizado para uma representação eficaz de fenômenos climáticos em escala regional, já o modelo regional tem uma alta resolução espacial, porém é limitado a restritas áreas de atuação como cidades e estados [Osthoff et al 2011]. O OLAM foi criado para suprir as necessidades dos modelos global e regional, unindo em um só modelo a atuação tanto em escala global (países, continentes e etc.) quanto em escala regional (estados, cidades e etc).

3. Computação Paralela

A Computação Paralela estabelece um novo conceito de desenvolvimento de aplicações de alto desempenho, totalmente contrário ao antigo paradigma de programação sequencial, que é o paradigma de programação paralela. Neste paradigma um programa é executado simultaneamente por diversos processos [Santana 2007]. Para que haja a comunicação entre eles é utilizada uma biblioteca para a troca de mensagens entre esses processos, como a biblioteca MPI. O objetivo de paralelizar uma aplicação é aumentar o seu *SpeedUp*, tornando o tempo de processamento da aplicação mais eficiente e mais rápido do que a sua versão serial.

3.1 Message Passing Interface - MPI

O *Message Passing Interface* (MPI) é uma biblioteca que implementa um padrão para a troca de mensagens entre os nós de um *cluster* em uma arquitetura de memória distribuída, e representa o paradigma de programação paralela. O MPI está disponível para as linguagens C, C++, FORTRAN 77 e FORTRAN 90 [Ribeiro 2011].

4. Objetivo

O objetivo da pesquisa é estudar métodos e tecnologias que permitam aumentar a escalabilidade em um *cluster* MPI executando o OLAM, ou seja, possibilitar a inclusão de novos nós resultando no aumento do seu *SpeedUp*. Como consequência do aumento na escalabilidade da versão paralela do OLAM, os institutos de pesquisas de clima e tempo poderão realizar previsões meteorológicas de forma mais rápida e eficiente utilizando esta versão modelo atmosférico.

5. Justificativa

O mundo tem presenciado nas últimas décadas significativas mudanças climáticas, este fato é evidenciado pelo aumento da temperatura global, diminuição na camada de gelo na zona polar Ártica, dentre outros fenômenos [Silva 2009]. A computação de alto desempenho vem sendo utilizada por institutos de pesquisas meteorológicas para prever estes fenômenos climáticos e assim tentar minimizar as consequências causadas por estes fenômenos. Mesmo com a utilização *clusters* para executar os modelos de simulações atmosféricas de forma mais eficiente, ainda há uma carência no poder de processamento para realizar previsões em uma menor escala de tempo. A exploração da

escalabilidade nestes modelos torna-se uma forma viável para conseguir atingir este objetivo.

6. Metodologia Prorosta

Inicialmente será instalada a versão paralela o OLAM 3.x no *cluster* do SENAI CIMATEC. Após o ambiente devidamente instalado e configurado serão realizados testes massivos com diferentes quantidades de nós no *cluster* a fim de avaliar o seu *SpeedUp* e identificar o grau de escalabilidade do OLAM naquele *cluster*.

Com base nos resultados dos testes o autor pretende realizar customizações na paralelização do OLAM, onde em um primeiro momento testará o modelo em diferentes tipos de aplicação paralela, como o *Data-parallel Model*, o modelo *Master/Worker*, *Pipeline*, o modelo híbrido, dentre outros. Em um segundo momento será testado a paralelização em multi-core, onde processos serão divididos e enviados para processadores diferentes em cada nó do *cluster* utilizando o *Open Multi-Processing* (OpenMP). E por fim, serão realizados testes com um modelo híbrido de paralelização utilizando o MPI e o OpenMP.

7. Considerações Finais

Ao final da pesquisa o autor espera propor uma customização do OLAM que apresente uma maior escalabilidade quando comparado com sua versão padrão disponibilizada pela *Duke University*, e como consequência melhorar a precisão e eficiência das previsões meteorológicas que utilizam o OLAM.

Como as pesquisas encontram-se em seus estados embrionários, o retorno da comunidade científica em relação ao escopo e metodologia proposta torna-se importante para que o autor possa realizar uma análise crítica que reflita em melhores resultados ao final de sua dissertação de mestrado.

Referencias

- Santana, F. J. S. B. Um modelo de balanceamento de carga para inclusão dinâmica de *workers* em HNOW. Fundação Visconde de Cairu, Salvador, 2007.
- Ribeiro, N. S. Explorando programação Híbrida no contexto de *Clusters* de Máquina NUMA. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2011.
- Silva, R. R. Modelo OLAM (*ocean-land-atmosphere-model*): descrição, aplicações, e perspectivas. Rev. bras. meteorol., São Paulo, v. 24, n. 2, jun. 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-77862009000200004&lng=pt&nrm=iso>. Acesso em: 19 julho 2011.
- Osthoff, C.; Grunmann, P. J.; Boito, F.; Kassick, R.; Pilla, L.; Navaux, P. O.; Schepke, C.; Panetta, J.; Maillard, N. B.; Dias, P. L. S.; Walko, R. *Improving performance on atmospheric models through a hybrid OpenMP/MPI implementation*; In: The 9th IEEE International Symposium on Parallel and Distributed Processing with Applications, 2011.