

# Análise de Desempenho do Software Incompact3D em uma Arquitetura com Múltiplos Núcleos

Lucas R. de Araujo, Cristian M. Weber  
Fernando E. Puntel, Andrea S. Charão, João Vicente F. Lima

<sup>1</sup> Laboratório de Sistemas de Computação  
Universidade Federal de Santa Maria

*Resumo.* Este trabalho utiliza o software Incompact3D, situado na área de dinâmica de fluidos, para a realização de testes de desempenho em arquiteturas com múltiplos núcleos. Estas arquiteturas surgem como alternativa para a baixa disponibilidade de grandes clusters, que costumam suprir a demanda de processamento de softwares como o Incompact3D. Os resultados indicam que as arquiteturas utilizadas são capazes de executar diversos casos do Incompact3D.

## 1. Introdução

Os problemas presentes na área de dinâmica dos fluidos são indiscutivelmente de grande complexidade, sendo que a resolução dos mesmos é de extrema importância para a comunidade científica. O estudo de [Laizet et al. 2010] indica que avanços nesta área de pesquisa foram conquistados expressivamente nos últimos 40 anos, devido ao advento dos computadores e avanços nos métodos numéricos. Nesse contexto, o Incompact3D é um *software* que realiza simulações com o objetivo de resolver problemas na área. Devido à complexidade dos cálculos e ao alto nível de paralelização do Incompact3D, ele caracteriza-se como um *software* típico de computação científica de alto desempenho.

O código do Incompact3D foi feito inicialmente para processadores seriais, posteriormente convertido para processadores vetoriais até chegar no modelo atual, que foi feito para processadores em paralelo [Laizet et al. 2010]. Para tal modelagem ocorre a utilização do MPI (Message Passing Interface), que é a linguagem padrão para programação paralela em sistemas distribuídos e apresenta importantes vantagens de utilização como o desempenho atingido e a sua portabilidade [Jin et al. 2011].

A necessidade de grandes *clusters* para execução de um algoritmo como o Incompact3D esbarra no problema de disponibilidade dessas máquinas para fins de pesquisa. Os trâmites administrativos e as filas de espera para utilização de grandes *clusters* acabam resultando na utilização de *desktops* para a realização de estudos prévios por parte dos pesquisadores. Todavia, tal prática está longe do ideal para pesquisa, pois as execuções se tornam extremamente demoradas. Arquiteturas paralelas com múltiplos núcleos surgem como uma alternativa de meio-termo, tanto por fatores de desempenho quanto por fatores econômicos, para a resolução de problemas na área de dinâmica dos fluidos e na área de computação de alto desempenho.

Nesse contexto, este trabalho busca conhecer o comportamento do Incompact3D em arquiteturas paralelas de múltiplos núcleos. Esse comportamento foi investigado através da análise dos dados gerados pelo Incompact3D e por ferramentas auxiliares durante os testes realizados.

## 2. Incompact3D

O Incompact3D é um *software* que realiza simulações envolvendo turbulência e fluidos, dada uma malha no plano cartesiano 3D. Diversas variáveis definem a complexidade e o comportamento da simulação, entre elas: o tamanho da malha no plano cartesiano, a condição de contorno, o número de colunas e linhas em que ocorrerá a decomposição da malha, o número de iterações abrangido pela simulação, entre outras.

Como apresentação geral do código, [Laizet e Li 2011] definem que o Incompact3D resolve as equações incompressíveis de Navier Stokes usando *compact schemes* de sexta ordem para a discretização espacial. O *software* divide a malha cartesiana em um número de partes definido pelo usuário, tal que esse número seja igual ao número de processos utilizados durante a execução. A partir daí existem rotinas de inicialização, rotinas que fazem parte do ciclo de iterações e rotinas de finalização da aplicação.

É no ciclo de iterações que acontecem as simulações, em função do tempo decorrido. É possível identificar, através do código e de trabalho anteriores, uma sequência de estados pela qual a malha passa através das operações de transposição, onde esses estados representam a orientação em que a malha decomposta se encontra. Há também a realização de várias operações matemáticas durante esses diversos estados, como, por exemplo, a resolução da equação de Poisson para pressão. No decorrer da execução do Incompact3D, a saída é gerada em tempo real para o usuário e pode também ser transcrita para arquivos.

## 3. Materiais e Métodos

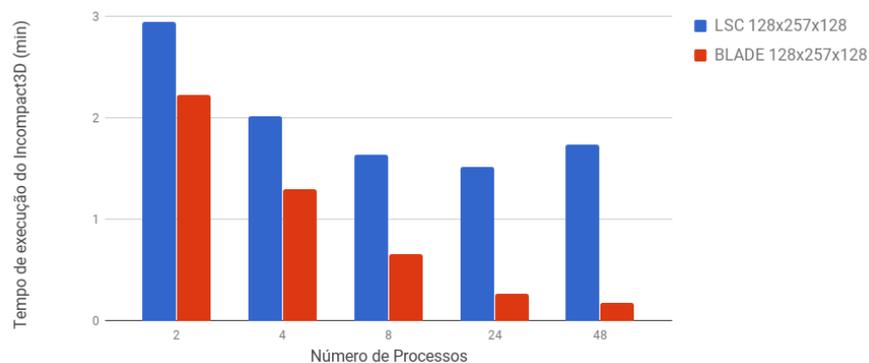
Os experimentos foram realizados em duas máquinas diferentes acessadas através de uma conexão SSH (Secure Shell), realizada em *desktops*. As máquinas utilizadas são uma SGI Altix XE 210, que apresenta arquitetura com 8 núcleos (processador Intel® Xeon® CPU E5620 @ 2.40GHz) e 16GB de memória RAM e uma NUMA SGI UV2000 de arquitetura com 48 núcleos (processador Intel® Xeon® CPU E5-4617 @ 2.90GHz) e 512GB de memória RAM. As duas máquinas são apresentadas neste trabalho, respectivamente, como LSC5 E BLADE01. Ambas as máquinas utilizam o sistema operacional GNU/Linux (distribuição Debian) e a implementação do MPI utilizada foi a Open MPI, na versão 2.0.2. A ferramenta EZTrace [Trahay et al. 2011], que é configurada para executar junto ao Incompact3D, foi utilizada para obter dados do uso de funções do MPI, referentes a trocas de mensagens entre os processos, durante a execução do Incompact3D.

Dados os ambientes de execução do Incompact3D, o *software* foi configurado de diversas formas. As principais mudanças ocorreram no tamanho da malha do problema, que é alterada a partir de mudanças em quaisquer das três malhas do plano cartesiano 3D (X, Y e Z). O número de processos utilizados durante as execuções também foi alterado, o que reflete na decomposição da malha. O número de iterações realizadas durante a execução também foi configurado de diferentes formas, o que reflete no tempo consumido pela simulação. O número de iterações utilizado na maioria dos casos foi o de 100 iterações, exceto em casos que buscavam explorar o comportamento dessa variável em diferentes situações.

## 4. Resultados e Discussão

A abordagem inicial realizada se baseou em testes feitos sem o auxílio de qualquer ferramenta, onde o Incompact3D foi configurado para execução e o próprio apresentou o

tempo total gasto nas iterações, ao final das execuções. A Figura 1 ilustra os resultados desses testes nas máquinas LSC5 e BLADE01 para uma configuração de malha com 4210688 (128x257x128) nós e uma variação entre 2 e 48 processos utilizados. O gráfico indica que para um mesmo número de processos, ainda assim há divergência no tempo de execução.



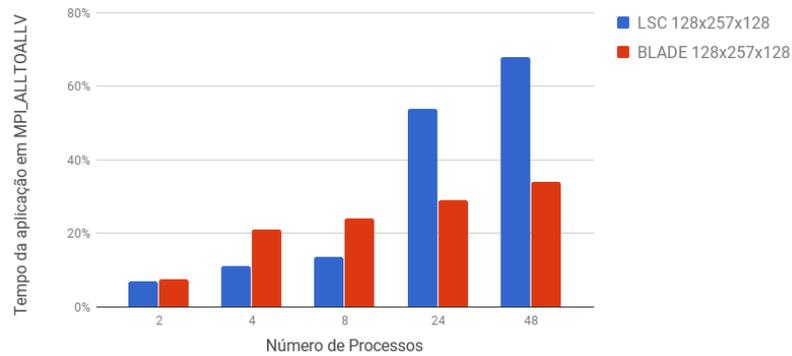
**Figura 1. Tempo total de cada execução em função do número de processos.**

Esta divergência é explicada pela diferença de arquitetura entre as máquinas LSC5 e BLADE01, principalmente devido a diferença de frequência entre os processadores de cada máquina. Nos casos de 2, 4 e 8 processos as duas máquinas operam com todos os processos simultaneamente, porém a máquina LSC5 executa o Incompact3D mais lentamente devido a menor frequência de seu processador. Já nos casos de 24 e 48 processos a diferença é ainda maior e a explicação é simples, dado que enquanto a BLADE01 consegue utilizar os processos criados simultaneamente, a máquina LSC5 não tem essa capacidade. Logo, o revezamento entre os processos ocasiona um tempo de espera para a execução de outros na máquina LSC5.

A outra abordagem de testes visou observar a interação entre os processos durante a execução do Incompact3D, que ocorre através de funções do MPI. A interação mais observada foi a MPI\_ALLTOALLV (onde todos os processos mandam dados para todos os outros e recebem dados de todos os outros), que é bastante utilizada durante as operações de transposição. Nessas operações, ocorre uma troca massiva de dados, a fim de alterar o estado atual de decomposição da malha do Incompact3D.

Através da ferramenta EZTrace foi possível monitorar a porcentagem de tempo ocupada por essas operações. Esses dados são apresentados na Figura 2 e mostram que, apesar de diminuir o tempo total de execução, o aumento no número de processos resulta em um aumento na proporção de tempo ocupado por operações de transposição. Isso se deve ao fato de que mais processos precisam se comunicar entre si, gerando um aumento na duração das operações de transposição. O alto crescimento nos casos de 24 e 48 processos na máquina LSC5 se devem ao tempo de espera pelo qual os processos passam enquanto não estão sendo executados.

Outra variável observada durante os testes foi a de número de iterações. Dessa forma, alguns testes foram conduzidos a fim de verificar qual a curva de crescimento de tempo em relação a curva de aumento do número de iterações. Para experimentos com 200 iterações, há um aumento de tempo em torno de 0.08 segundos por iteração, o que passa para mais de 0.1 segundo para o maior caso testado (600 iterações). A variação de tempo



**Figura 2. Porcentagem de tempo da comunicação entre os processos em relação ao tempo total.**

apresentada é em relação a uma curva de crescimento linear do tempo em relação a curva do número de iterações. Há então uma tendência de aumento para casos maiores, devido a variação observada nesse intervalo. O que deixa claro que não há um comportamento constante por parte das iterações, ainda que a complexidade do problema seja a mesma.

## 5. Considerações Finais

A partir dos resultados obtidos nos testes é possível afirmar que as máquinas LSC5 e BLADE01 foram capazes de executar com sucesso o Incompact3D nos casos propostos. Também observou-se que as operações de transposição não se beneficiam do aumento do número de processos, ocupando cerca de 34% da execução com 48 processos na BLADE01 contra cerca de 7% no caso de 2 processos. Por outro lado, devido a diminuição de tempo mostrada na figura 1 é possível afirmar que outras operações se beneficiam desse aumento. Em trabalhos futuros, espera-se investigar mais profundamente o comportamento de cada iteração do Incompact3D e maneiras de melhorar o desempenho das operações de transposição.

## 6. Agradecimentos

Os autores agradecem ao financiamento e recursos cedidos pela FAPERGS-CNPQ PRO-NEX, pois foram imprescindíveis para a realização da pesquisa e do presente trabalho.

## Referências

- Jin, H., Jespersen, D., Mehrotra, P., Biswas, R., Huang, L., e Chapman, B. (2011). High performance computing using mpi and openmp on multi-core parallel systems. *Parallel Computing*, 37(9):562 – 575.
- Laizet, S., Lamballais, E., e Vassilicos, J. (2010). A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution dns of fractal generated turbulence. *Computers & Fluids*, 39(3):471 – 484.
- Laizet, S. e Li, N. (2011). Incompact3d: A powerful tool to tackle turbulence problems with up to O(105) computational cores. *International Journal for Numerical Methods in Fluids*, 67:1735–1757.
- Trahaay, F., Rue, F., Faverge, M., Ishikawa, Y., Namyst, R., e Dongarra, J. (2011). EZTrace: a generic framework for performance analysis. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, Newport Beach, CA, United States.