Avaliação do uso eficiente de algoritmos paralelos de filogenia em supercomputadores

Kary Ocaña, Joice Alves, Micaella Coelho, Marcelo Galheigo, Carla Osthoff

Laboratório Nacional de Computação Científica (LNCC), Brasil

{karyann, joice, micaella, galheigo, osthoff}@lncc.br

Resumo. *Ambientes de processamento de alto desempenho (PAD) são requeridos pela bioinformática, em especial nas áreas de filogenia e evolução de organismos. O presente trabalho apresenta uma análise comportamental da ferramenta de filogenia RAxML em ambientes de PAD. O RAxML implementa paralelismo e distribuição de tarefas pelo uso de threads e bibliotecas MPI. As execuções foram realizadas nos clusters do supercomputador Santos Dumont (SDumont). Os resultados em termos de eficiência evidenciam que o impacto é gerado pela variação da configuração, acoplamento e uso do RAxML e do SDumont. Essa informação permitirá um melhor entendimento e uso eficiente desses ambientes especializados.*

Abstract. *Bioinformatics, especially the areas of phylogeny and evolution of organisms, requires high performance computing (HPC) environments. The present work presents a behavioral analysis of the RAxML phylogeny tool in HPC environments. RAxML implements parallelism and task distribution by using threads and MPI libraries. Executions were performed in the Santos Dumont (SDumont) supercomputer. Efficiency results show the impact is generated by the variation in configuration, coupling and use of RAxML and SDumont. This information will allow a better understanding and efficient use of these specialized environments.*

1. Introdução

A bioinformática é uma ciência multidisciplinar que apoia à análise e interpretação da grande quantidade de dados gerados pela biologia molecular, médica e da saúde, por meio do desenvolvimento de algoritmos e abordagens computacionais. A filogenia aplica algoritmos probabilísticos na construção de árvores filogenéticas, no apoio a análise sobre a vida evolutiva dos organismos, genes ou genomas. Alguns métodos de análise filogenética de estado de caractere, como a máxima verossimilhança (MV) e inferência Bayesiana (IB), implementam modelos probabilísticos complexos e estão atualmente dentre dos mais utilizados [Felsenstein 1996; Ronquist and Huelsenbeck 2003; Stamatakis 2014].

O método de MV busca a árvore mais verossímil dentre um conjunto de árvores geradas, tomando em conta informações como os dados genéticos, modelos evolutivos, taxas de mutação, entre outros. O programa de MV RAxML [Stamatakis 2014] foi escolhido por usufruir de características computacionais como paralelismo, distribuição e escalabilidade, o que o faz um candidato interessante para ser acoplado efetivamente em ambientes de processamento de alto desempenho (PAD) como *clusters*, grades e nuvens.

O RAxML, além da versão sequencial, apresenta as versões paralelas PThread, MPI e híbrida, o que possibilita a avaliação de diferentes algoritmos computacionais em ambientes de PAD [Stamatakis 2014]. A versão PThread emprega o paralelismo interno, *i.e.*, aquele que ocorre dentro de um nó do agregado, onde os processadores compartilham espaço de memória e se comunicam por instruções de leitura e escrita (arquitetura de memória compartilhada). Essa versão apresenta a paralelização em grãos finos, onde

pequenas quantidades de tarefas computacionais são realizadas entre os eventos de comunicação. A versão MPI emprega o paralelismo externo, *i.e.*, aquele que ocorre entre os nós agregados, com a comunicação por troca de mensagens através de uma rede que interconecta os nós (arquitetura de memória distribuída). Essa versão apresenta a paralelização em grãos grossos, onde grandes quantidades de tarefas computacionais são realizadas entre os eventos de comunicação e sincronização. Já a versão híbrida explora ambos os métodos PThread e MPI [Abramson *et al*. 2011; Aho *et al*. 2006; Pfeiffer e Stamatakis 2010].

O presente trabalho tem como objetivo avaliar o desempenho das versões paralelas do RAxML nos *clusters* do supercomputador Santos Dumont (SDumont). Os experimentos foram desenhados a fim de explorar algumas das configurações mais representativas, com variações nos dados de entrada (tamanho de dados biológicos) e da parametrização (número de replicações ou *bootstrap*) e das versões do RAxML. Os resultados fornecerão informações importantes de modo a dar suporte aos usuários externos e administradores do SDumont, para o uso eficiente de ferramentas em ambientes de supercomputadores.

Por exemplo, atualmente o LNCC hospeda portais científicos das mais diversas áreas da ciência, dentre eles o BioinfoPortal de bioinformática (http://bioinfo.lncc.br/) e o DockThor de ancoragem molecular (https://dockthor.lncc.br/). Esses portais estão alocados nos *clusters* do SDumont no SINAPAD e são usados intensivamente pela comunidade de pesquisadores das áreas respectivas, no Brasil e no exterior. Por tanto, é de interesse para a comunidade o uso otimizado dessas ferramentas em ambientes de PAD, devido a que pode diminuir o tempo de espera na execução de tarefas nesses portais.

O artigo está organizado em quatro seções, além dessa introdução. A Seção 2 apresenta a configuração do experimento, a Seção 3 discute sobre os resultados de desempenho e finalmente, a Seção 4 conclui este artigo.

2. Ambiente Computacional

O SDumont possui uma arquitetura de configuração híbrida com capacidade de processamento paralelo na ordem de 1,1 Petaflops. Ele é composto por 18.144 núcleos de CPU, distribuídos em 756 nós computacionais e 24 núcleos por nó. Também possui o nó especial MESCA2, de 240 núcleos e memória compartilhada de 6 Terabytes. Todos os nós do SDumont então interconectados por uma rede de alta velocidade, a Infiniband FDR, que integra o sistema de arquivos paralelos. O gerenciador de filas do SDumont é o Slurm.

2.1 Configuração do Ambiente com o RAxML

O RAxML é um programa de código aberto do GNU GPL, distribuído através do repositório GitHub em https://github.com/stamatak/standard-RAxML. A compilação do código é feita atendendo os requisitos dos recursos de CPU de maneira a melhor aproveitar suas características computacionais. Enquanto aos recursos do processador, o RAxML suporta o SSE3 (*Streaming SIMD Extensions 3*), AVX (*Advanced Vector Extensions*) e AVX2 (extensão mais rápida do SSE). Essas instruções são usadas pelo RAxML para acelerar substancialmente os cálculos de probabilidade e parcimônia que conduz. O RAxML pode ser executado com as versões sequencial (HPC) ou paralelas usando MPI, PThreads ou híbrida (MPI + *Threads*). O RAxML foi compilado no SDumont com o AVX.

2.2 Dados Experimentais

Os *datasets* usados são superalinhamentos de genes ortólogos universais de genomas de protozoários. A Tabela 1 mostra as características dos *datasets* usados como número de táxons, tamanho e número de caracteres do alinhamento. Para a execução do RAxML é necessário informar o arquivo no formato PHYLIP [Felsenstein 1985] (dado de entrada), a versão sequencial ou paralela do RAxML (tipo de processamento), o valor de *bootstrap* e o modelo de substituição. Para a análise de eficiência foram realizados dois experimentos: (i) variando o tamanho/caractere do dado de entrada (D0, D5, D3), fixando o *bootstrap* em 100 e (ii) variando os valores de *bootstrap* (100, 1.000, 2.000), fixando o dado de entrada em D5. Os experimentos foram executados três vezes e os resultados apresentados como a média dos valores obtidos.

As versões sequencial, PThread, MPI e híbrida do RAxML foram acopladas no SDumont. Sobre o uso dessas versões, a versão sequencial é geralmente selecionada para tratar conjuntos de dados pequenos a médios (filogenia de genes). As versões paralelas são indicadas para a execução de alinhamentos muito longos ou superalinhamentos (filogenia de genomas) e o seu desempenho depende do tipo *hardware* usado.

Tabela 1. Principais características dos *datasets* usados nos experimentos

Dataset	**Táxons**	**Tamanho**	**Caracteres**
D0	10	3,2 Kilobytes	233
D5	12	79 Kilobytes	4.941
D3	26	792 Kilobytes	22.906

3. Análise dos Resultados de Desempenho dos Experimentos

A Figura 1 apresenta a escalabilidade em termos de eficiência para o experimento com *datasets* de caracteres diferentes ("menor" D0 de 233, "médio" D5 de 4.941, "maior" D3 de 22.906), fixando o valor de *bootstrap* em 100 (considerado o mínimo ideal para a execução no RAxML) em 6 nós (144 *cores*). Pode ser observado que a melhor eficiência (0.9) foi atingida em 5 nós para ambos os *datasets* D5 e D3.

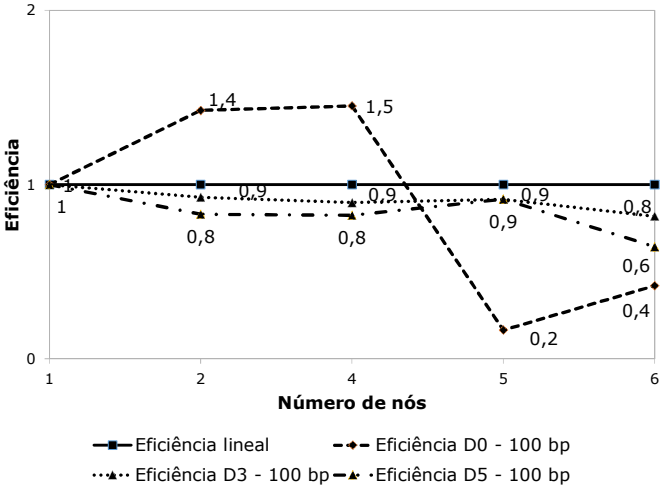

Figura 1. Eficiência do Experimento 1: variando o número de caracteres (D0, D5, D3) e fixando o *bootstrap* (100).

A Figura 2 apresenta a escalabilidade em termos de eficiência para o experimento com valores diferentes no *bootstrap* (100, 1.000, 2.000), fixando o *dataset* em D5 (4.941caracteres). Na Figura 2, o *bootstrap* 100 possui uma melhor eficiência para 5 nós, o *bootstrap* 1.000 apresenta melhor eficiência para 2 nós, e o *bootstrap* 2.000 para 6 nós.

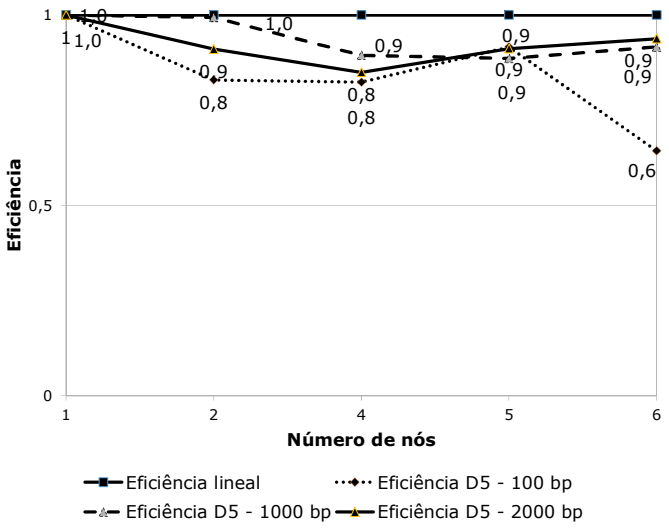

Figura 2. Eficiência do Experimento 2: variando o *bootstrap* (100, 1.000, 2.000), fixando os caracteres com D5.

4. Conclusão

O presente trabalho visa analisar o comportamento da ferramenta de bioinformática RAxML em ambientes de PAD. Os resultados das análises de filogenia indicam que o impacto na eficiência das execuções pode ser afetado por fatores relacionados às características biológicas (número de caracteres), parâmetros usados do RAxML (*bootstrap*) ou configuração dos *clusters* (bibliotecas de compilação ou natureza do processador). Desta maneira, é preciso uma análise exploratória das ferramentas, seus algoritmos e parâmetros, em especial quando experimentos em larga escala são alocados em ambientes especializados como o SDumont.

Como trabalhos futuros serão realizados estudos exaustivos das outras ferramentas acopladas no SDumont. Visa-se investir no uso de perfiladores que levem a desenvolver algoritmos que identifiquem gargalhos e erros de execução e ajudem na toma de decisões do melhor parâmetro, ferramenta ou ambiente de PAD. Finalmente, este trabalho retornará um *feedback* para os administradores do SINAPAD com o intuito de apoiar no uso eficiente de processadores.

Agradecimentos. A pesquisa foi desenvolvida no Laboratório de Bioinformática (LABINFO) do LNCC (https://www.labinfo.lncc.br/), as horas de execução no SDumont fornecidas pelo SINAPAD (http://sdumont.lncc.br/). Ela foi financiada pelo Programa Universal MCTI/CNPq 01/2016, Processo: 429328/2016-8, pelo CNPq, CAPES e FAPERJ

Referências Bibliográficas

Abramson, D., et al. (2011). Parameter Exploration in Science and Engineering Using Many-Task Computing. *IEEE Trans. Parallel Distrib. Syst.*, v. 22, n. 6, p. 960–973.

Aho, A. et al. (2006). *Compilers: Principles, Techniques, and Tools*. 2. ed. Addison Wesley.

Felsenstein, J. (1985). Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution*, v. 39, n. 4, p. 783–791.

Felsenstein, J. (1996). Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Methods in Enzymology*, v. 266, p. 418–427.

Pfeiffer, W. e Stamatakis, A. (2010). Hybrid MPI/Pthreads parallelization of the RAxML phylogenetics code. IEEE. http://ieeexplore.ieee.org/document/5470900/.

Ronquist, F. e Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics (Oxford, England)*, v. 19, n. 12, p. 1572–1574.

Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, v. 30, n. 9, p. 1312–1313.