

Avaliação do RAxML no Supercomputador Santos Dumont

Micaella Coelho, Carla Osthoff, Kary Ocaña

Laboratório Nacional de Computação Científica (LNCC), Brasil

{micaella, osthoff, karyann}@lncc.br

Resumo. *Análises filogenéticas apoiam estudos sobre a vida evolutiva dos organismos. Ferramentas otimizadas como RAxML, baseadas em algoritmos de máxima verossimilhança, geram alto custo computacional pelos inúmeros cálculos para processar grandes quantidades de dados. O uso eficiente dessas ferramentas em ambientes paralelos é requerido. O presente trabalho visa explorar o desempenho do RAxML em supercomputadores explorando características de configuração do ambiente, programa e dados.*

Abstract. *Phylogenetic analyzes support studies about the evolutionary life of organisms. Optimized tools such as RAxML, based on maximum likelihood algorithms, generate high computational costs to process large amounts of data. Efficient use of these tools in parallel environments is required. The present work aims to explore the performance of RAxML in supercomputers exploring characteristics of environment, program and data.*

1. Introdução

A filogenia faz uso de algoritmos computacionais para representar a história da vida evolutiva de organismos. O algoritmo de máxima verossimilhança (MV) implementa modelos probabilísticos complexos e eficazes, mas que geram alto custo computacional [Stamatakis 2014]. Dentre os programas de MV mais usados estão RAxML/ExaML, PhyML e IQ-TREE.

RAxML [Stamatakis 2014] Sequencial destina-se a conjuntos de dados pequenos a médios (filogenia) e as versões paralelas PThread, MPI e Híbrido são usadas em maior escala (filogenômica). PThread emprega paralelismo interno em arquiteturas de memória compartilhada; MPI paralelismo externo em arquitetura de memória distribuída e Híbrido explora métodos [Abramson *et al.* 2011; Pfeiffer e Stamatakis 2010].

O objetivo deste trabalho visa avaliar o desempenho do RAxML¹ em ambientes de supercomputador através de uma análise comparativa sobre o impacto produzido pela variabilidade nas configurações de ambiente, programas e características dos dados genômicos. O artigo está organizado em 5 seções, além dessa introdução. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta o experimento e a Seção 4 apresenta os resultados. Finalmente, a Seção 5 conclui este artigo.

2. Trabalhos Relacionados

[Pfeiffer e Stamatakis 2010] apresentam uma análise de desempenho das versões paralelas implementadas no RAxML, sustentando a versão Híbrida como a mais eficiente. [Zhou *et al.* 2018] apresentam uma análise comparativa entre os programas

¹ <https://github.com/stamatak/standard-RAxML>

PhyML, IQ-TREE e RAxML/ExaML, concluindo que RAxML, além de apresentar maior escalabilidade, gera topologias de árvores com melhor qualidade.

O presente artigo explorar o desempenho do RAxML no supercomputador Santos Dumont² (SDumont), explorando configurações de ambiente, versões do RAxML, valores de *bootstrap* [Felsenstein 1985] e características dos dados (tamanho).

3. Configuração do Experimento

A Figura 1 apresenta a metodologia dividida em três atividades: (i) comparação das versões do RAxML; (ii) execução do RAxML Híbrido variando o tamanho de arquivo e (iii) execução do RAxML híbrido variando o *bootstrap*.

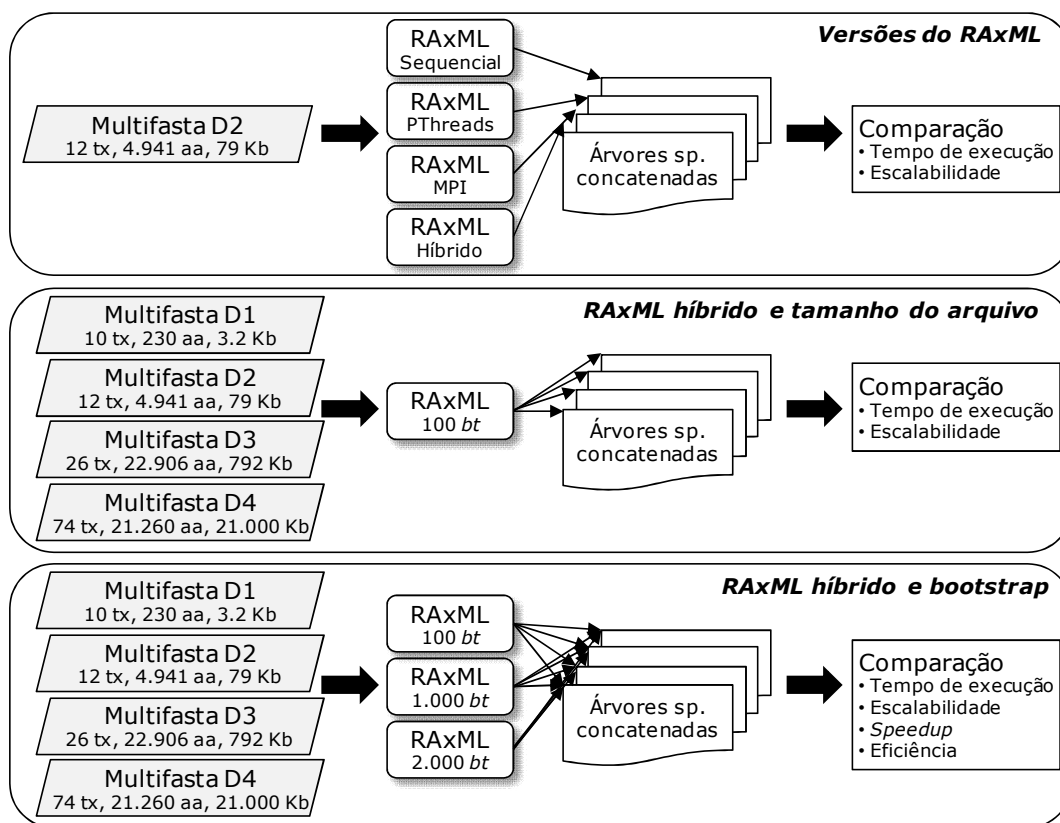


Figura 1. Metodologia do Experimento: Avaliação do RAxML no SDumont

As entradas são quatro arquivos PHYLIP de superalinhamentos de genomas de protozoários [Dávila e Ocaña 2011] com diferente número de táxons, caracteres e tamanho. As versões Sequencial, Pthreads, MPI e Híbrido do RAxML 8.2.12 foram executadas com *bootstraps* 100, 1.000, e 2.000 e os resultados analisados pelo desempenho (tempo, escalabilidade, *speedup* e eficiência).

Os *clusters* Altix³ e SDumont são gerenciados pelo SINAPAD/LNCC⁴. Altix é composto por 30 nós, cada um possui 2 processadores *quad core*, com 8 núcleos por nó, totalizando 240 núcleos de CPU. SDumont possui uma arquitetura de processamento

² <http://sdumont.lncc.br/>

³ <http://www.lncc.br/altix-xe/>

⁴ Sistema Nacional de Processamento de Alto Desempenho, <https://www.lncc.br/sinapad/>

paralelo de configuração híbrida de 1,1 Petaflops com 18.144 núcleos de CPU, 756 nós computacionais (24 núcleos por nó) interconectados pela rede Infiniband FDR. Os gerenciadores de filas utilizados são o SGE (Altix) e Slurm (SDumont).

4. Análise do Desempenho do Experimento

4.1. Versões do RAxML. TTE e Escalabilidade

A versões do RAxML Sequencial, PThreads, MPI e Híbrido foram executadas no Altix e SDumont, usando o arquivo D2, modelo evolutivo JTT (GAMMA) e *bootstrap* 100.

A Figura 2(A) apresenta o Tempo Total de Execução (TTE) em minutos: RAxML MPI apresentou melhor desempenho no Altix com 98,3% de melhoria (285,5 min. em 1 *core* e 4,5 min. em 120 *cores*) e RAxML Híbrido no SDumont com 98,8% de melhoria (161,2 min. em 1 *core* e 1,8 min. em 120 *cores*).

A Figura 2(B) apresenta a escalabilidade em minutos dessas melhores versões até 120 *cores*. O RAxML Híbrido no SDumont apresentou melhor desempenho para todos os *cores*. Até 24 *cores* o TTE teve uma caída drástica e até 120 *cores* a diminuição é menos gradativa, algumas das causas podem ser alocação de memória, afinidade de processo ou escrita em disco.

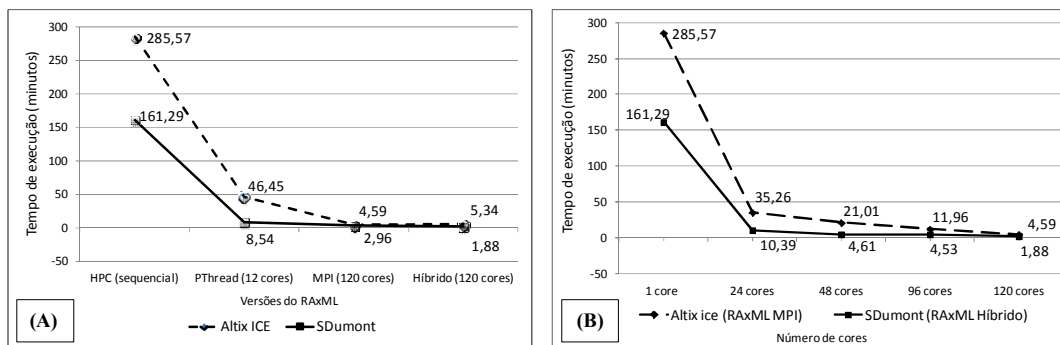


Figura 2. (A) TTE do RAxML no Altix e SDumont. (B) Escalabilidade do melhor desempenho do RAxML MPI no Altix e Híbrido no SDumont

4.2. RAxML híbrido e tamanho do arquivo. TTE e Escalabilidade

A Figura 3 apresenta o desempenho em minutos do RAxML Híbrido no SDumont com variação no tamanho de arquivos e *bootstrap* fixo em 100. Foi observada uma relação entre o incremento de tamanho dos arquivos e a melhoria de desempenho de aprox. 90% entre 1 até 10 nós. D1: 3.2 Kb e 81.3%, D2 79 Kb e 88.6%, D3 792 Kb e 88,4% e D4 21.000 Kb e 89%. Esses valores indicam que a paralelização se beneficia de arquivos maiores e que possuem mais táxons e/ou maior comprimento de sequências.

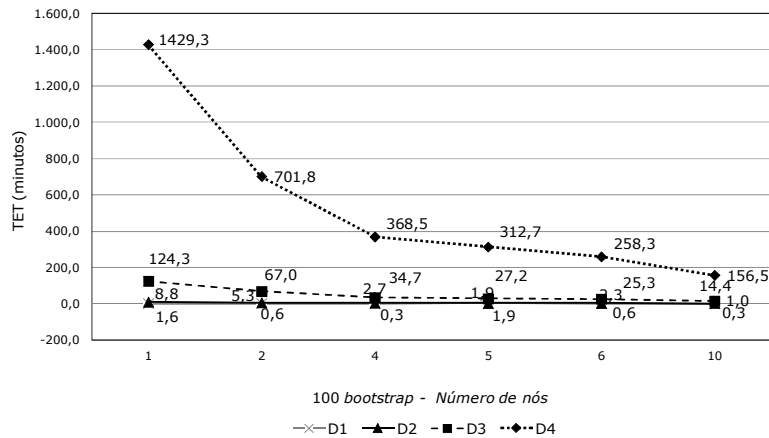


Figura 3. Desempenho do RAxML Híbrido no SDumont variando o tamanho dos arquivos

4.3. RAxML Híbrido e *bootstrap*. TTE, Escalabilidade, *Speedup* e Eficiência

A Figura 4 apresenta o desempenho em minutos do RAxML Híbrido no SDumont executado com os arquivos D1, D2, D3 e D4 e com *bootstraps* 100, 1.000 e 2.000. Em todos os casos, a diminuição do TTE é acentuada até 4 nós e a diminuição entre 5 e 10 nós foi em escala menor. Os resultados de eficiência (Figura 6) corroboram esses resultados. Por exemplo na Figura 6(D4), o arquivo maior D4 (21.000 Kb) apresenta os melhores resultados para todos os *bootstraps*. Já o arquivo menor D1 (3.2 Kb) é mais eficiente com 2 nós e todos os demais arquivos tiveram eficiência melhor com 4 nós. Pode se concluir que RAxML escala melhor com arquivos de tamanho e valor de *bootstrap* maiores. Extrapolando esses resultados a um uso real no SDumont, este alocaria arquivos e configurações desses tipos diretamente para até 4 nós.

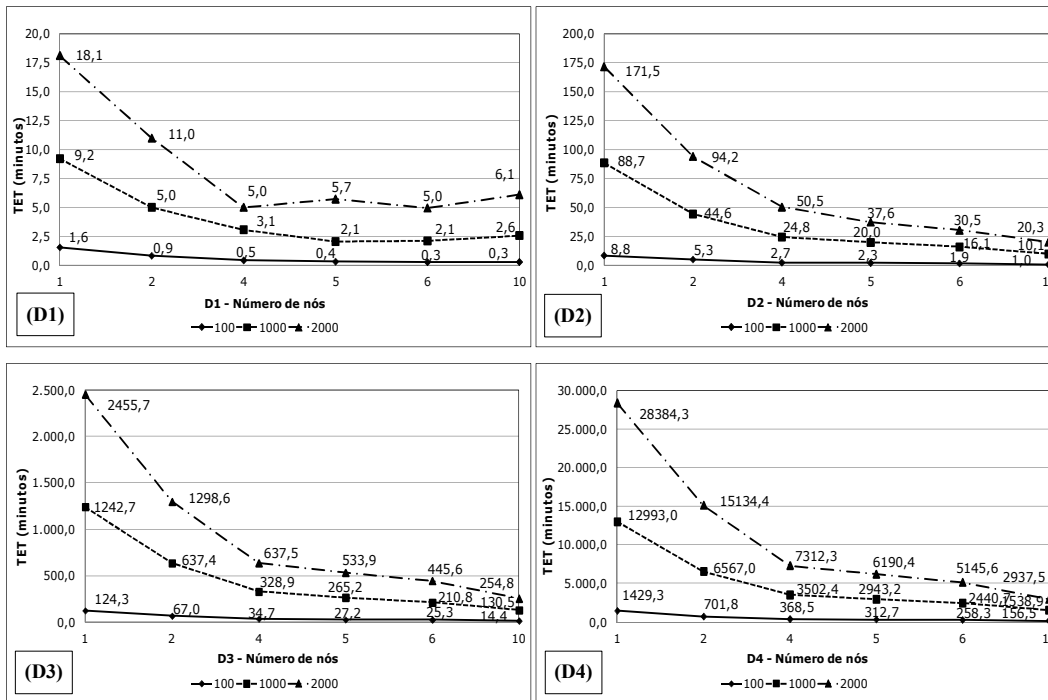


Figura 4. Desempenho do RAxML Híbrido no SDumont variando o valor de *bootstrap* para D1, D2, D3 e D4

A Figura 5 apresenta o *speedup* quase linear até 5 nós para D2, D3 e D4 (o maior dos arquivos), depois disso o *speedup* começa a degradar. O tamanho menor afeta o *speedup* que não permanece constante em comparação aos demais arquivos, em um experimento em que cada execução foi repetida três vezes. O *speedup* é o tempo de execução sequencial dividido pelo tempo de execução paralelo por nós.

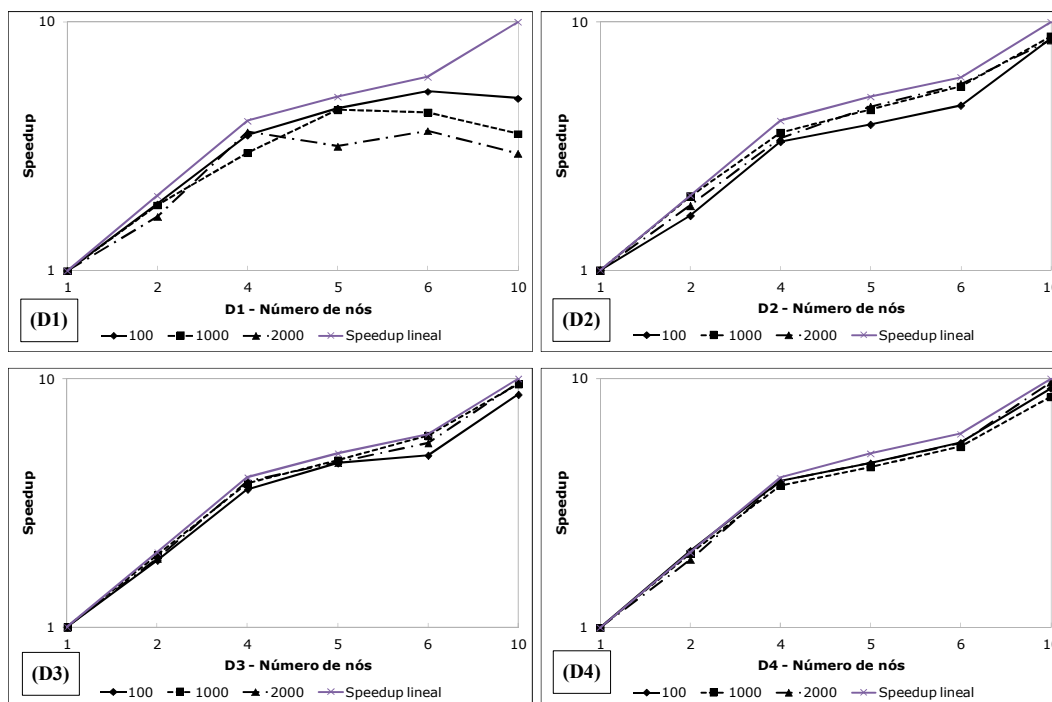
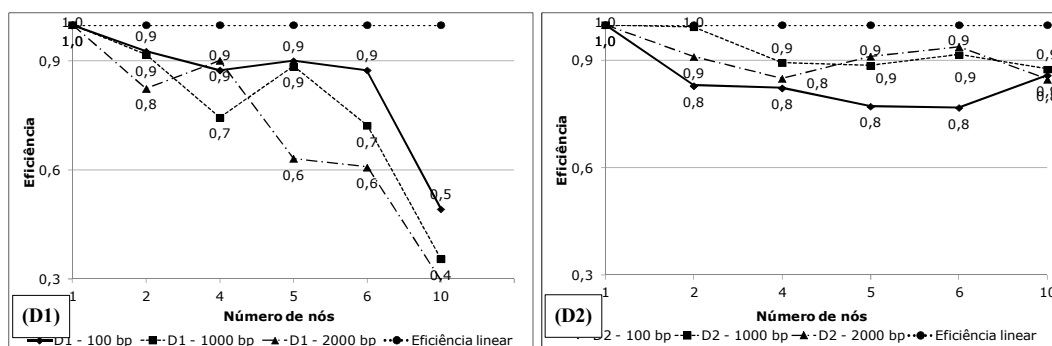


Figura 5. *Speedup* do RAXML Híbrido no SDumont variando os *bootstraps* para D1, D2, D3 e D4

A eficiência ideal (linear) é obtida ao dividir o *speedup* pelo número de nós usado, a curva de eficiência ideal mostra uma forma linear (Figura 6). A eficiência para os arquivos D2, D3 e D4 oscilou entre 0.8 e 1.0 exceto para D1 com valores menores. O arquivo maior D4 é o mais escalável e se beneficia melhor do ambiente e configurações paralelas. Segundo os testes *benchmark* [Pfeiffer e Stamatakis 2010] se espera que na medida em que o *bootstrap* aumente com arquivos de supermatrizes, as execuções se tornem mais escaláveis. Pode se concluir através da Figura 6, que D1 torna-se mais eficiente com 2 nós e todos os demais arquivos com 4 nós.



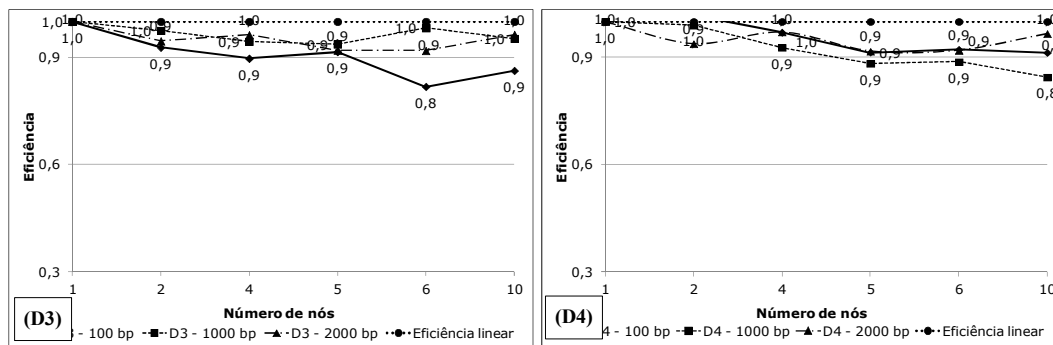


Figura 6. Eficiência do RAxML Híbrido no SDumont variando os *bootstraps* para D1, D2, D3 e D4

5. Conclusão

O supercomputador SDumont aloca portais *Web*⁵ e *software* de diversos projetos científicos⁶ no Brasil. Por tanto, é de interesse de administradores e usuários ter ciência sobre características de alocação e desempenho para garantir o seu uso efetivo.

Nesse trabalho avaliamos o desempenho do RAxML no SDumont. A eficiência demonstra que características como tamanho de dados e *bootstrap* influem no desempenho. O arquivo maior (21.000 Kb) com o maior valor de *bootstrap* (2.000) foi o mais escalável. Já o arquivo menor (3.2 Kb) teve melhor eficiência com 2 nós e todos os arquivos maiores com 4 nós. Fatores que impactam no desempenho são alocação de memória, afinidade de processo, escrita em disco e uso de gerenciadores de filas e compiladores. Perfiladores para a identificação de gargalhos estão sendo testados.

Agradecimentos. Este trabalho foi financiado pelos projetos Universal MCTI/CNPq n° 01/2016, processo 429328/2016-8 e JCNE FAPERJ n° 03/2017, processo 232985. As execuções foram realizadas no SDumont no SINAPAD/LNCC.

Referências Bibliográficas

Abramson, D., Bethwaite, B., Enticott, C., Garic, S. e Peachey, T. (2011). Parameter Exploration in Science and Engineering Using Many-Task Computing. *IEEE Trans. Parallel Distrib. Syst.*, v. 22, n. 6, p. 960–973.

Dávila, A. e Ocaña, K. (2011). Phylogenomics-Based Reconstruction of Protozoan Species Tree. *Evolutionary Bioinformatics*, p. 107.

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

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

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.

Zhou, X., Shen, X.-X., Hittinger, C. T. e Rokas, A. (2018). Evaluating Fast Maximum Likelihood-Based Phylogenetic Programs Using Empirical Phylogenomic Data Sets. *Molecular Biology and Evolution*, v. 35, n. 2, p. 486–503.

⁵ BioinfoPortal, <http://bioinfo.lncc.br/>

⁶ http://sdumont.lncc.br/projects_view.php?pg=projects&status=ongoing