Avaliação da aplicação RAxML no Ambiente do Supercomputador SDumont

Guilherme Freire[1,2], Kary Ocaña[2], Micaella Coelho[2], Carla Osthoff[2]

[1]Faculdade de Educação Tecnológica do Estado do Rio de Janeiro (FAETERJ)
Petrópolis – RJ – Brasil

[2]Laboratório Nacional de Computação Científica (LNCC)
Petrópolis – RJ – Brasil

{gfreire, karyann, micaella, osthoff}@lncc.br

Abstract. *This work presents a performance study developed for the RAxML software. The study evaluates the RAxML execution time according to the allocated computational resources and the application input parameters in the Santos Dumont (SDumont) Supercomputer environment.*

Resumo. *Este trabalho apresenta um estudo de desempenho desenvolvido para o software RAxML. O estudo avalia o tempo de execução do RAxML segundo os recursos computacionais alocados e os parâmetros de entrada da aplicação no ambiente do Supercomputador Santos Dumont (SDumont).*

1. Introdução

O Portal de Bioinformática, Bioinfo-Portal (https://bioinfo.lncc.br/), é um ambiente computacional que gerencia a execução de aplicações e dados científicos de bioinformática em larga escala, e serve de apoio às pesquisas da comunidade científica de informática através de uma interface *Web* amigável e interativa. É gerenciado pelo Laboratório Nacional de Computação Científica, LNCC (https://lncc.br) e usa recursos e tecnologias de computação de alto desempenho, como o SDumont (https://sdumont.lncc.br).

O objetivo do projeto de pesquisa de Iniciação Científica é desenvolver estudos de desempenho das aplicações do Portal de Bioinformática, de forma a obter a melhor alocação de recursos computacionais, segundo os parâmetros de entrada dos usuários e gerar *scripts* de submissão de *jobs* otimizados para o Portal. Este trabalho apresenta o estudo desenvolvido para o *software* RAxML, que é largamente utilizado pelos pesquisadores da área de filogenia.

2. O RAxML no SDumont

O RAxML é um *software* para inferência, baseada em Máxima Verossimilhança de árvores filogenéticas. Pode ser usado para pós-análises de conjuntos de árvores filogenéticas, análises de alinhamentos e colocação evolutiva de ”leituras curtas”. O RAxML possui diversas versões, além da versão sequencial, possui uma para ser executada de forma paralela em ambientes *multicores*, chamada de versão *multithreads*; uma para ambientes distribuídos, chamada de versão *MPI* e uma para ambientes *multicores* distribuídos, chamada de versão Híbrida. Efetuamos testes com o RAxML, utilizando dados de Aminoácido em formato (*phylip*), que contém sequências biológicas de genes

ortólogos de 12 genomas de protozoários, para fazer um alinhamento simples, selecionado com base na variabilidade de suas características, por ser mais próximo de um caso de estudo real. O *Bootstrapping* é um procedimento estatístico que reamostra um único conjunto de dados para criar muitas amostras simuladas. O valor de *bootstrap* para um dado é a proporção das árvores replicadas que recuperaram aquele dado em particular. Neste estudo, avaliamos o desempenho da versão de ambientes *multicores* distribuídos, compatível com a arquitetura dos nós computacionais do SDumont, utilizados pelo Portal de Bioinformática.

3. Resultados

Esta seção apresenta detalhes sobre o experimento e análises de desempenho do RAxML. Cada teste passou por cinco repetições junto ao ambiente do SDumont. As execuções com o RAxML foram utilizando os dados de Aminoácido em formato *(phylip)*, de 3,2 KB de tamanho, 10 sequências e 230 aminoácidos, variando os valores do número de *threads*, de *bootstrap* e do número de nós computacionais. Foram alocados quatro nós computacionais compostos por duas CPUs *Ivy Bridge Intel Xeon E5-2695v2 (12cores @2.4GHz) e 64 GB de memória RAM*. Os 4 nós computacionais alocados para o experimento correspondem ao número máximo de nós que atualmente podem ser alocados pelo Portal de Bioinformática.

3.1. Exploração do RAxML com 1 nó computacional

Conforme a (Figura 1), no primeiro experimento avaliamos o tempo total de execução (TTE) em função do número de *threads*, em 1 nó computacional, para diversos valores de *bootstrap*, alternando entre 100 (linha azul), 500 (linha vermelha), 1000 (linha verde), 1500 (linha roxa) e 2000 (linha azul claro). Os parâmetros de *threads* selecionados foram 4 *threads*, 8 *threads*, 12 *threads*, 24 *threads*. Podemos observar que o menor tempo total de execução (TTE) ocorre para 24 *threads*, para todos os valores de *bootstrap* avaliados exceto para o valor *bootstrap* 100, que teve ganho desprezível. Podemos observar que para este valor não é eficiente executar o RAxML com múltiplos *threads*.

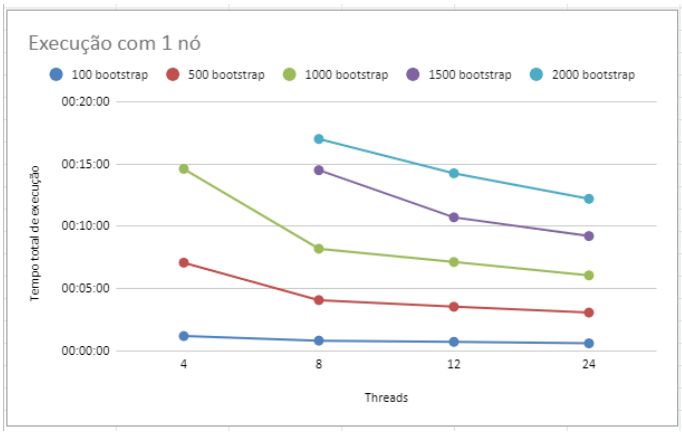

Figura 1. Tempo de execução do RAxML com 1 nó computacional

3.2. Exploração do RAxML em até 4 nós computacionais

O segundo experimento é um estudo de múltiplos nós computacionais, o gráfico na (Figura 2), apresenta o tempo de execução do RAxML, na versão Híbrida, de 1 a 4

nós computacionais, mantendo uma alocação fixa de 24 *threads*, que corresponde ao menor tempo de execução avaliado pelo experimento anterior. Assim como no experimento anterior, utilizamos os valores do *bootstrap* variando de 100 (linha azul), 500 (linha vermelha), 1000 (linha verde), 1500 (linha roxa) e 2000 (linha azul claro). Podemos observar que o tempo de execução para todos os valores de *bootstrap* ocorreram na a alocação de 4 nós computacionais e que para o valor 100, não é eficiente executar o RAxML com mais de um nó computacional.

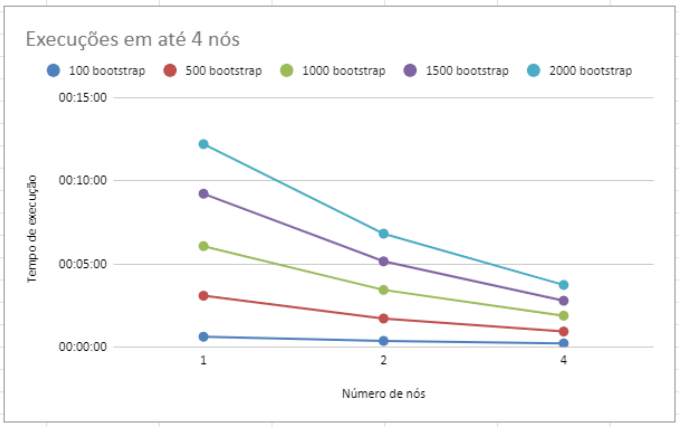

Figura 2. Tempo de execução do RAxML em até 4 nós computacionais

4. Conclusão

Os testes do RAxML na versão Híbrida apresentados na (Figura 1), permitem inferir que o menor tempo total de execução (TTE) para os valores de *bootstrap* testados em 1 nó computacional ocorre na alocação de 24 *threads*. Os testes com múltiplos nós computacionais, conforme a (Figura 2), permitem concluir que o menor tempo de execução, para todos os parâmetros de *bootstrap* testados, ocorre com 4 nós computacionais alocando (24 *threads*) cada. Podemos concluir também que não é indicado que o RAxML seja executado em ambiente paralelo quando executado com o valor de *bootstrap* 100.

Como trabalhos futuros iremos implementar testes com a aplicação RAxML, para um número maior de nós computacionais e de *bootstrap*. Os dados que estão sendo coletados serão aplicados numa ferramenta de inteligência artificial (IA), que está sendo desenvolvida para automatizar a escolha dos recursos do Bioinfo-Portal, segundo os parâmetros de entrada dos usuários.

5. Referências

COELHO, M; OSTHOFF, C; OCAÑA, K. Avaliação do RAxML no Supercomputador Santos Dumont. In: ARTIGOS CURTOS - SIMPÓSIO BRASILEIRO DE BIOINFORMÁTICA (BSB) , 2018, Niterói. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018. p. 37-42.

M. Coelho, G. Freire, K. Ocaña, C. Osthoff, M. Galheigo, A. R. Carneiro, F. Boito, P. Navaux, D. O. Cardoso. Desenvolvimento de um Framework de Aprendizado de Máquina no Apoio a Gateways Científicos Verdes, Inteligentes e Eficientes: BioinfoPortal como Caso de Estudo Brasileiro . XXIII Simpósio em Sistemas Computacionais de Alto Desempenho WSCAD 2022.