

Análise de Desempenho do BEAST 1.10 em ambientes de HPC: Explorando CPU, GPU, Multi-GPU

Guilherme Freire^{1,2}, Micaella Coelho¹, Carla Osthoff¹, Kary Ocaña¹

¹Laboratório Nacional de Computação Científica (LNCC), RJ - Brasil.

²Faculdade de Ensino Tecnológico do estado do Rio de Janeiro (FAETERJ), RJ - Brasil.

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

Abstract. *This work presents a performance study of the Bayesian inference application BEAST 1.10, coupled with the high-performance library BEAGLE 3, in executions performed on the nodes of the Santos Dumont supercomputer. In the phylogeny experiments, we used genomic data of the Dengue virus, serotype DENV-1, in XML format. We analyzed the variability of genome size, chainLength and evolutionary models of BEAST 1.10, the number of threads and the computational environment (CPU and GPU) of SDumont. The results of the study of the performance of BEAST in the BioInfo-Portal allow a more efficient use of the computational resources of SDumont, according to the parameters allocated in the submission of the jobs.*

Resumo. *Este trabalho apresenta um estudo de desempenho da aplicação de inferência bayesiana BEAST 1.10, acoplada à biblioteca de alto desempenho BEAGLE 3, em execuções realizadas nos nós do supercomputador Santos Dumont. Nos experimentos de filogenia, utilizamos dados genômicos do vírus da Dengue, sorotipo DENV-1, em formato XML. Analisamos a variabilidade do tamanho dos genomas, o chainLength e modelos evolutivos do BEAST 1.10, o número de threads e o ambiente computacional (CPU e GPU) do SDumont. Os resultados do estudo do desempenho do BEAST no BioInfo-Portal, possibilitam uma utilização mais eficiente dos recursos computacionais do SDumont, segundo os parâmetros alocados na submissão dos jobs.*

1. Introdução

A filogenia bayesiana é uma técnica estatística crucial para inferir relações evolutivas entre espécies, especialmente relevante para estudos de vírus, utilizando dados moleculares como sequências de DNA ou RNA. Combinando probabilidades observadas com distribuições a priori dos parâmetros evolutivos, e utilizando algoritmos como o *Markov Chain Monte Carlo* (MCMC), ela estima árvores filogenéticas e fornece estatísticas de credibilidade para as inferências [PENNY et al. 2003]. Na bioinformática, a filogenia de vírus é fundamental para entender a evolução, transmissão e resistência a medicamentos, ajudando a traçar estratégias de controle e tratamento mais eficazes [AYRES, 2019].

Atualmente, existe uma crescente demanda de dados da comunidade científica de bioinformática, com isso as aplicações precisam usufruir de muito poder computacional, para o processamento desses dados biológicos. Necessitam de ambientes computacionais mais aprimorados [COELHO, 2022]. O uso da arquitetura paralela dos núcleos da GPU, proporciona uma redução considerável no tempo de execução de tarefas, com grande demanda de processamento, um dos campos de interesse desse modelo de estudo é a de computação de alto desempenho (CAD).

Este estudo analisou o desempenho da aplicação BEAST 1.10 [SUCHARD et al. 2018] (<https://beast.community/>), integrada à biblioteca BEAGLE 3, em ambientes de CAD. Os dados genômicos foram processados nos nós *multithreads*, GPU e Multi-GPU do supercomputador Santos Dumont, SDumont (<https://sdumont.lncc.br/>). O objetivo foi avaliar o desempenho e a escalabilidade do BEAST 1.10 no ambiente computacional do SDumont, visando otimizar a alocação de recursos e sua utilização eficiente. Exploramos o impacto do tamanho dos dados genômicos, a influência dos modelos evolutivos e o parâmetro *chainLength* do BEAST 1.10 em diferentes configurações, filas, *threads* e nós do SDumont.

O Bioinfo-Portal (<https://bioinfo.lncc.br/>), foi desenvolvido sob a arquitetura do *middleware* CSGrid, está hospedado no Sistema Nacional de Computação de Alto Desempenho (SINAPAD) e utiliza os recursos computacionais do SDumont do Laboratório Nacional de Computação Científica (LNCC). O portal suporta execuções de *software*, *workflows* e bibliotecas de bioinformática que foram instaladas e alocadas no ambiente computacional do SDumont [COELHO et al. 2022]. Por esse motivo, o desempenho que o *gateway* oferece aos seus usuários está diretamente ligado ao desempenho da execução das tarefas no supercomputador, incluindo a escolha de parâmetros como o número de nós e os *threads* a serem utilizados.

Em seguida, abordaremos a seção *Background*, onde falaremos sobre a filogenia e o BEAST. Logo depois, a seção destinada a trabalhos relacionados. Uma seção sobre a metodologia utilizada e a configuração do ambiente. Após, os resultados, com as análises dos experimentos. Ao final, a seção apresentando a conclusão.

2. Background em Filogenia

A análise filogenética esclarece as relações de parentesco entre espécies. A filogenia é essencial para sistematizar a diversidade biológica, integrando dados evolutivos sobre características morfológicas, comportamentais e moleculares. Diversos métodos filogenéticos estão disponíveis, incluindo métodos de distância, máxima parcimônia, máxima verossimilhança e bayesianos, cada um escolhido conforme os dados e objetivos da análise [PENNY et al. 2003].

O BEAST [SUCHARD et al. 2018] é um *software* avançado para inferências filogenéticas e filodinâmicas bayesianas. Ele unifica a reconstrução filogenética, a modelagem da evolução de características discretas e contínuas, a datação por divergência e os modelos demográficos coalescentes, empregando métodos MCMC para uma inferência estatística robusta e eficiente.

O pacote BEAST inclui um programa de interface gráfica de usuário (GUI) aprimorado, *Bayesian Evolutionary Analysis Utility* (BEAUti) que permite acesso a modelos avançados em sequência molecular e evolução de características fenotípicas. Fornece ferramentas para visualizar e resumir análises coalescentes e filogeográficas multiespécies [DRUMMOND et al. 2012]. O BEAUti (<https://beast.community/beauti>) é usado para importar dados, projetar a análise e gerar o arquivo de controle BEAST.

3. Trabalhos relacionados

FREIRE, 2020-A apresenta experimentos com BEAST 2, no ambiente CPU (*Ivy Bridge*) (24 *cores*), RunA (20 KB) XML. Os resultados do menor tempo total de

execução (TTE) foi em 12 *threads*, para uma tarefa no SDumont. Em FREIRE, 2020-B usando o BEAST 1.8, em CPU, GPU e Multi-CPU/GPU. Com a BEAGLE 3 e Vírus da Dengue (XML). Resultados mostram que apesar da configuração Multi-CPU/GPU apresentar menor TTE, o ganho de desempenho CPU/GPU é muito pequeno ao da CPU, o mais indicado em termos de eficiência, é utilizar a CPU.

AYRES, 2019 apresenta a biblioteca BEAGLE 3, acoplada a *softwares* de filogenia, com novas implementações paralelas, maior desempenho para conjuntos de dados, melhor escalabilidade e usabilidade. O trabalho atual utiliza BEAST 1.10, seguindo como base os trabalhos relacionados, acoplado a BEAGLE 3 e ambientes CPU e GPU do SDumont, explorando um maior número de genomas do DENV-1, diversificando o tamanho do genoma, modelos evolutivos, parâmetro *chainLength* e nós CPU, GPU, na busca de identificar quais seriam as configurações para obter uma melhor eficiência no consumo computacional durante as execuções.

4. Metodologia

A combinação do BEAST 1.10 com a BEAGLE 3 [AYRES et al. 2019] permite paralelizar múltiplas partições de dados em um único dispositivo de alto desempenho, utilizando toda a capacidade dos dispositivos, reduzindo as sobrecargas computacionais. Abordamos neste estudo 3 experimentos, variando as arquiteturas CPU (*Ivy Bridge*) versus CPU (*Cascade Lake*) e GPU K40 versus GPU V100 com modelo HKY; CPU (*Cascade Lake*) e GPU V100 com variação de modelos evolutivos; GPU V100 e Multi-GPU com modelo HKY.

Os dados biológicos dos genomas do vírus do Dengue, do tipo DENV-1, com o genoma humano como hospedeiro, obtidos das bases de dados do Sistema *Bacterial and Viral Bioinformatics Resource Center* (BV-BRC) e do *National Center for Biotechnology Information* (NCBI). Os genomas alinhados com o *software* MAFFT. Os alinhamentos em formato FASTA foram convertidos para XML com a BEAUti.

Os dados genômicos utilizados na entrada do BEAST 1.10, foram 3 conjuntos de genomas em XML: *World* “W” (27 MB), *SouthAmerica* “SA” (0,99 MB) e *Brazil* “BR” (458 KB). Os experimentos executados no ambiente do SDumont, em 1 nó computacional, variando o parâmetro *chainLength* 100000 e 20,000000, configurações CPU, GPU e modelos evolutivos JC, GTR, HKY e TN93, sendo o JC e HKY modelos que efetuam análises mais simples, o GTR e TN93 mais complexos, os quais necessitam de maior tempo computacional. O parâmetro *chainLength* especifica o número de etapas que a cadeia MCMC executará, quanto maior seu valor, maior a intensidade computacional, impactando diretamente no tempo de execução.

4.1 Configuração do ambiente computacional

CPU (*Ivy Bridge*): Em 1 nó computacional (24 *cores*) do SDumont com GPUs K40, por 2 CPUs *Ivy Bridge Intel Xeon E5 2695v2 12cores@2.4GHz* e 64 GB de memória RAM. Os *threads* 1, 4, 8, 12 e 24. **CPU (*Cascade Lake*):** Nó Sequana, CPU 2x *Intel Xeon Cascade Lake Gold 6252*, com 48 núcleos (24 *cores* por CPU), 384 GB de memória RAM. Os *threads* 1, 4, 8, 12 e 24. **GPU (V100):** Nó Sequana GPU, 384 GB de memória

RAM, 4x *NVIDIA* Volta V100. As arquiteturas GPU (*NVIDIA* V100) e CPU (*Cascade Lake*) são mais modernas.

4.2 Comparação entre GPU versus Multi-GPU

Os dados foram particionados no BEAUti para execuções com Multi-GPU em um nó com 4 GPUs *NVIDIA* Volta V100. Foram realizadas execuções nas configurações de GPU única e Multi-GPU, utilizando dados com duas e três partições.

5. Resultados

O objetivo foi diversificar o tamanho do genoma, modelos evolutivos, parâmetros do BEAST/BEAGLE como *chainLength* e nós CPU/GPU. Dessa forma, buscando para cada conjunto de genoma, as configurações mais adequadas e obtendo assim uma melhor eficiência no consumo computacional das execuções no ambiente do SDumont.

5.1 Resultado do experimento com variação de arquiteturas e modelo HKY

A Figura 1 apresenta o TTE com o *chainLength* 100000, em CPU, com 1, 4, 8, 12 e 24 *threads*. No ambiente CPU (*Cascade Lake*), o menor TTE é com 8 *threads*, o genoma “W” o TTE de 15,49 minutos, “SA” o TTE de 0,20 e “BR” o TTE de 0,26. O ambiente CPU (*Ivy Bridge*), o menor TTE é em 4 *threads*, o genoma “W”, o TTE de 17,31 minutos, o “SA” o TTE de 0,28 minutos e “BR” o TTE de 0,40 minutos. Nos ambientes computacionais em ambos os genomas o menor TTE é na CPU (*Cascade Lake*), devido à maior eficiência de paralelização e otimização de *threads* da CPU (*Cascade Lake*) (arquitetura mais recente) em comparação com a CPU (*Ivy Bridge*).

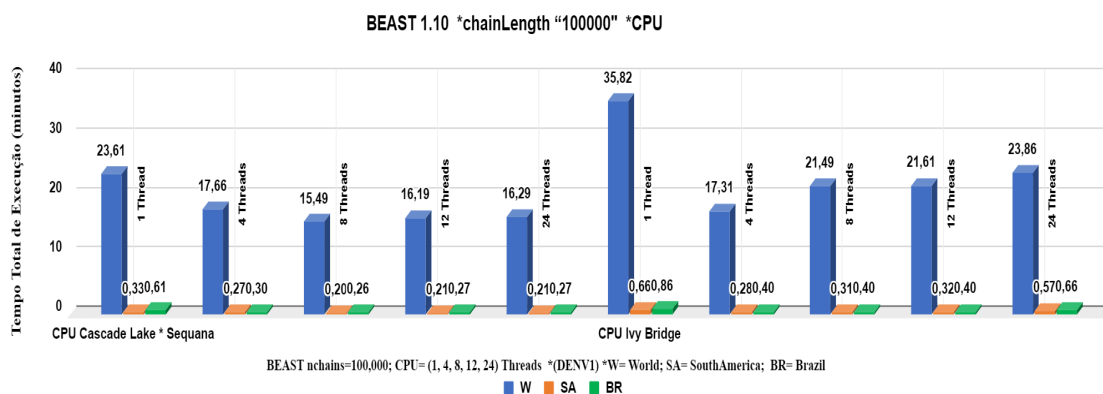


Figura 1. TTE em CPU com 1, 4, 8, 12 e 24 threads e chainlength 100000

A Figura 2 apresenta o TTE com o *chainLength* 20,000000, em CPU, com 1, 4, 8, 12 e 24 *threads*. No ambiente CPU (*Cascade Lake*) Sequana, o menor TTE é com 8 *threads*, o conjunto “W” o TTE de 1440,96 minutos, “SA” o TTE de 35,39 minutos e “BR” o TTE de 49,30 minutos. No ambiente CPU (*Ivy Bridge*), o menor TTE é com 4 *threads*, o conjunto de genoma “W” o TTE de 1798,56 minutos, o “SA” o TTE de 48,29 minutos e “BR” o TTE de 72,22 minutos. Em ambos os genomas, o menor TTE é na CPU (*Cascade Lake*) Sequana.

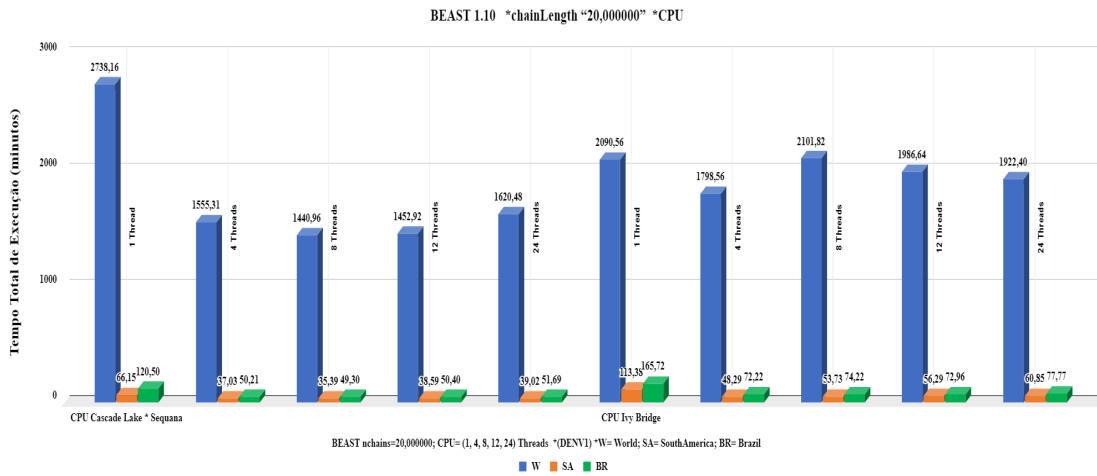


Figura 2. TTE em CPU com 1, 4, 8, 12 e 24 threads e chainlength 20,000000

Os resultados com BEAST 1.10/BEAGLE 3 em 1 nó computacional, com ambientes GPU (NVIDIA K40) versus GPU (NVIDIA V100) Sequana. Na Figura 3-A mostra o TTE, com chainLength 100000, em GPU. O ambiente GPU (NVIDIA V100) Sequana, o conjunto “W” TTE 4,01 minutos, “SA” o TTE 0,18 e “BR” TTE 0,22. O ambiente com GPU (NVIDIA K40), o conjunto “W” TTE 6,38 minutos, “SA” o TTE 0,61 e “BR” TTE 0,52. Nos ambientes computacionais, os conjuntos de genomas têm o menor TTE na GPU (NVIDIA V100) Sequana.

A Figura 3-B mostra o TTE, com chainLength 20,000000, em GPU. No ambiente GPU (NVIDIA V100) Sequana, o conjunto “W” TTE de 660,95 minutos, “SA” o TTE 32,80 minutos e “BR” 38,93 minutos. Nas execuções com GPU (NVIDIA K40), o conjunto “W” o TTE 989,50 minutos, “SA” o TTE 72,63 minutos e “BR” TTE 70,53 minutos. Os ambientes computacionais, em ambos os genomas, o menor TTE é na GPU (NVIDIA V100) Sequana.

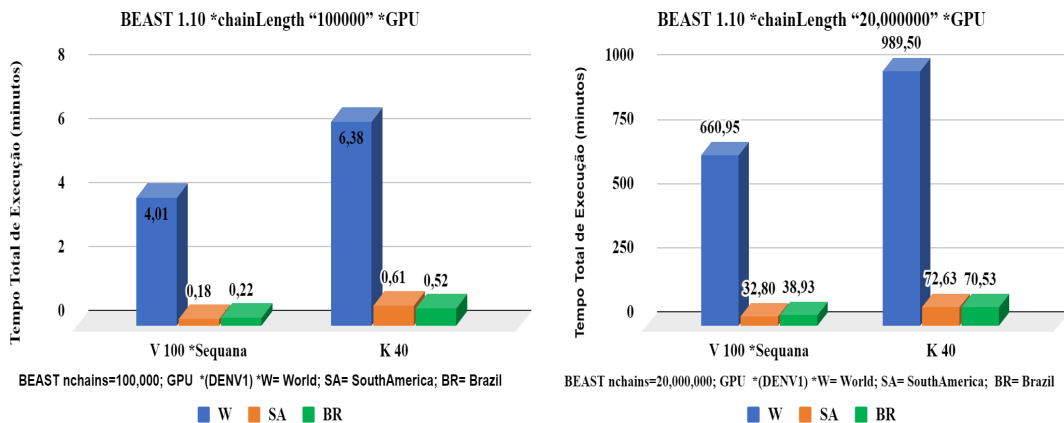


Figura 3. TTE GPU chainlength 100000 e 20,000000

Nas análises de desempenho do ambiente GPU (NVIDIA K40), o menor TTE ocorre para arquivos maiores que (0,99 MB). Ao executar o BEAST 1.10 no ambiente

CPU (*Ivy Bridge*), o menor TTE ocorre em 4 *threads* e o nó computacional pode vir a ser compartilhado com outra aplicação.

Os experimentos sugerem o uso da GPU (*NVIDIA V100*) apresentando menor TTE. Podemos concluir que a configuração mais eficiente, no ambiente CPU é (*Cascade Lake*) com 8 *threads* nos conjuntos de genoma menores que (0,99 MB), no nó Sequana do SDumont.

5.2 Resultado com variações nos modelos evolutivos

Os resultados do BEAST 1.10/BEAGLE 3, com os conjuntos de dados DENV-1 variando os modelos evolutivos GTR, HKY, TN-93 e JC, em 1 nó computacional: CPU (*Cascade Lake*) Sequana e GPU V100. A Figura 4 apresenta os resultados de TTE com o *chainLength 100000*, em CPU com 8 *threads*. O conjunto “W” o menor TTE foi 9,70 minutos no modelo (JC). O conjunto “SA” TTE 0,17 no modelo (TN93) e “BR” TTE de 0,26 no modelo (HKY). Os resultados mostram que os modelos evolutivos, interferem diretamente no TTE.

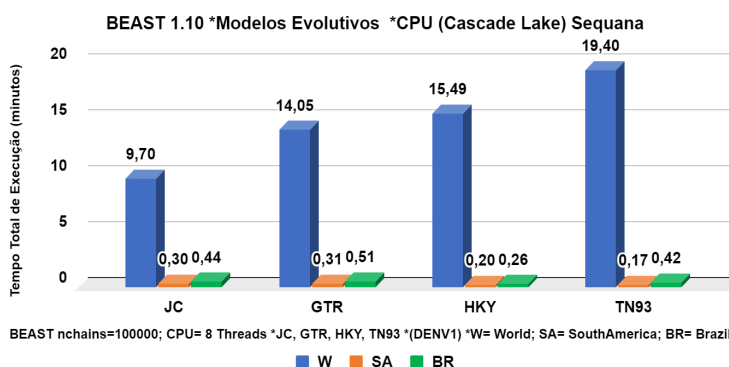


Figura 4. TTE CPU *chainlength 100000* com modelos JC, GTR, HKY e TN93

A Figura 5 apresenta os resultados de TTE com o *chainLength 20,000000*, em CPU, com 8 *threads*. O conjunto “W” o menor TTE foi 1196,58 minutos no modelo (JC) e o “SA” o TTE 31,82 minutos. O conjunto “BR” TTE 49,30 minutos com modelo (HKY). Em comparação aos modelos evolutivos utilizados, os genomas “W” e “SA” têm menor TTE no modelo (JC), esse modelo é computacionalmente menos intensivo.

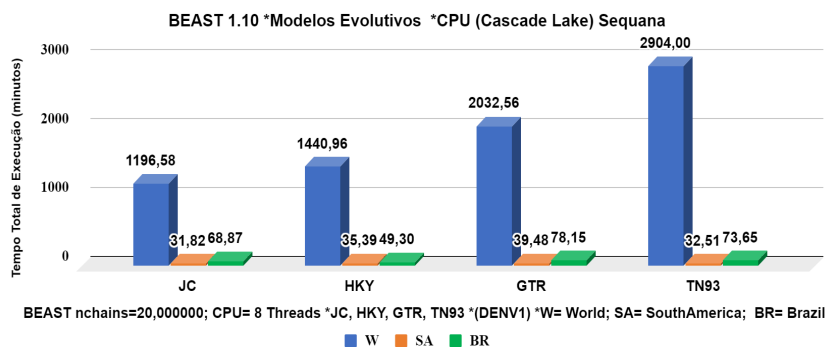


Figura 5. TTE CPU *chainlength 20,000000* com modelos JC, HKY, GTR e TN93

A Figura 6-A mostra o TTE, com o *chainLength 100000*, em GPU. O menor TTE foi no modelo (JC), o conjunto “W”, o TTE de 2,86 minutos, o “SA” o TTE de 0,15 e “BR” o TTE de 0,18. A Figura 6-B apresenta o TTE, com o *chainLength 20,000000*, em GPU. O menor TTE foi no modelo (JC), o conjunto “W”, o TTE de 445,68 minutos, o “SA” o TTE de 28,57 minutos e “BR” o TTE de 31,88 minutos. Nas execuções o conjunto “W” que é o maior dado, apresenta uma redução no TTE em todos os modelos evolutivos.

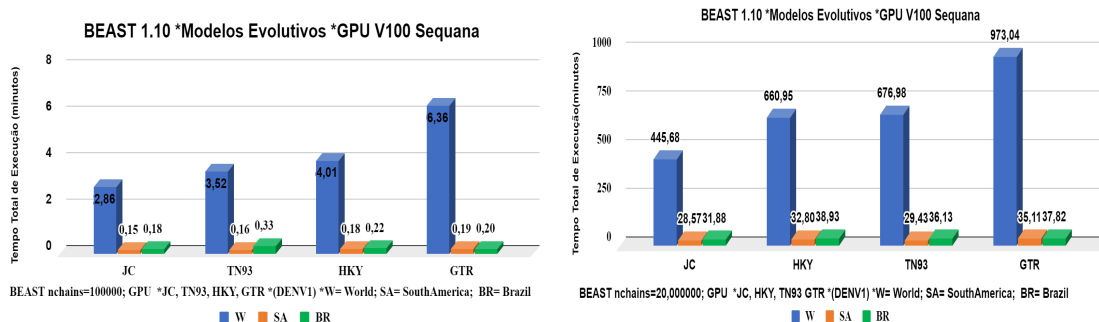


Figura 6. TTE GPU *chainlength 100000 e 20,000000* modelos JC, TN93, HKY e GTR

As execuções sugerem o uso da configuração GPU (*NVIDIA V100*), com o menor TTE para arquivos grandes no ambiente do nó Sequana. O modelo evolutivo (JC) necessita menor tempo de computação.

5.3 Resultado dos testes em ambientes GPU e Multi-GPU com modelo HKY

A Figura 7 apresenta o TTE, com *chainLength 100000*, em GPU e Multi-GPU. O menor TTE foi com 1 GPU, o conjunto “W” o TTE de 4,01 minutos, no “SA” o TTE de 0,18 e “BR” o TTE de 0,22. O ambiente computacional de 1 GPU obteve o menor TTE com todos os conjuntos em comparação ao Multi-GPU com dados particionados.

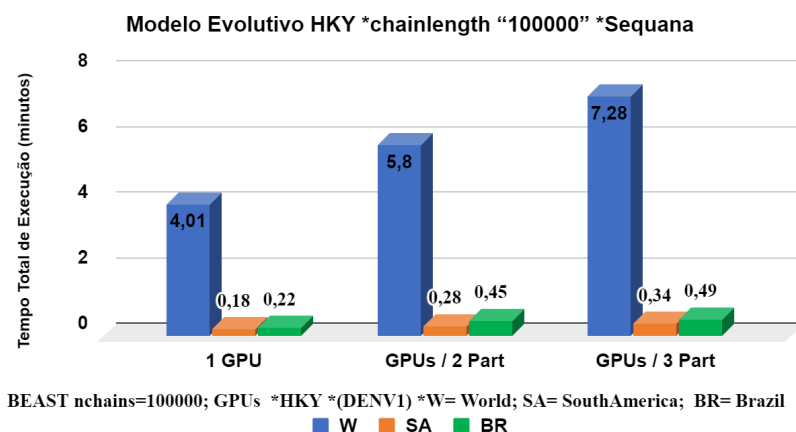


Figura 7. TTE GPU/Multi-GPU, *chainlength 100000* com modelo HKY

6. Conclusão

O estudo de desempenho com o BEAST 1.10/BEAGLE 3, no ambiente do SDumont, teve como objetivo apresentar a configuração de *hardware* mais eficiente segundo os parâmetros, tamanho dos arquivos, *chainLength*, modelos evolutivos, número de *threads*

e ambiente computacional CPU e GPU, tomando como estudo de caso os dados biológicos dos genomas do vírus do Dengue, do tipo DENV-1. A variabilidade no *chainLength*, destacou que quanto maior o valor de *chainLength*, maior a necessidade computacional da aplicação.

Podemos observar que o modelo evolutivo JC é um modelo menos complexo que os demais e computacionalmente menos intensivo. Observamos que para todos os modelos evolutivos e para arquivos maiores que (0,99 MB), a configuração mais eficiente para execução do BEAST 1.10 no SDumont é o nó computacional GPU (NVIDIA V100) Sequana. Para a execução do modelo HKY no nó computacional CPU (Cascade Lake) Sequana, com os conjuntos de genomas menores que (0,99 MB) é recomendado 8 *threads* e no nó computacional CPU (Ivy Bridge) é recomendada a configuração de 4 *threads*. Esses resultados fornecem um guia valioso para otimizar o uso de recursos computacionais em análises filogenéticas complexas. Como trabalhos futuros, pretendemos utilizar um perfilador durante as execuções, para analisar o comportamento dos ambientes durante as análises, principalmente com Multi-GPU.

References

- Penny, D., *Inferring Phylogenies*.— Felsenstein, J., 2003. Sinauer Associates, Sunderland, Massachusetts., *Systematic Biology*, Volume 53, Issue 4, August 2004, Pages 669–670, <https://doi.org/10.1080/10635150490468530>
- Ayres, D. L., Cummings, M. P., Baele, G., Darling, A. E., Lewis, P. O., Swofford, D. L., Huelsenbeck, J. P., Lemey, P., Rambaut, A., & Suchard, M. A. (2019). BEAGLE 3: Improved Performance, Scaling, and Usability for a High-Performance Computing Library for Statistical Phylogenetics. *Systematic biology*, 68(6), 1052–1061. <https://doi.org/10.1093/sysbio/syz020>
- Suchard, M. A., Lemey, P., Baele, G., Ayres, D. L., Drummond, A. J., & Rambaut, A. (2018). Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus evolution*, 4(1), vey016. <https://doi.org/10.1093/ve/vey016>
- Drummond, A. J., Suchard, M. A., Xie, D., Rambaut, A., Bayesian Phylogenetics with BEAUti and the BEAST 1.7, *Molecular Biology and Evolution*, Volume 29, Edição 8, agosto de 2012, páginas 1969–1973, <https://doi.org/10.1093/molbev/mss075>
- Coelho, M. 2022. Desenvolvimento de um Framework de Aprendizado de Máquina no Apoio a Gateways Científicos Verdes, Inteligentes e Eficientes: BioinfoPortal como Caso de Estudo Brasileiro. In: (SSCAD), 23., 2022, Florianópolis/SC. Anais [...]. Porto Alegre: SBC, p. 205-216. DOI:<https://doi.org/10.5753/wscad.2022.226377>.
- Freire, G., Ocaña K., Coelho M., and Osthoff, C. 2020. Exploração de Módulos Paralelo Híbrido de Bioinformática para Ambientes GPU de Supercomputação. In: *Anais da VI ERAD-RJ*, novembro 30, 2020, Evento Online, Brasil. SBC, Porto Alegre, Brasil, 57-59. DOI:<https://doi.org/10.5753/eradrj.2020.14521>.
- Freire, G., Ocaña K., Coelho M., and Osthoff, C. 2023. Avaliação da aplicação RAXML no Ambiente do Supercomputador SDumont. In: *Anais da ERAD-RJ*, 8., 2023, Rio de Janeiro. Porto Alegre: SBC, 2023. p.23-25. DOI:<https://doi.org/10.5753/eradrj.2023.231711>.