

# Proposta e Avaliação de um *Cluster* de Banana Pi Single Boards com NAS Parallel Benchmarks\*

Marcos Ani Cury Vinagre Silva<sup>†</sup>, Henrique Cota de Freitas

Departamento de Ciência da Computação

Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

30.535-901 – Belo Horizonte – MG – Brasil

macvsilva@sga.pucminas.br, cota@pucminas.br

**Abstract.** *Clusters for high-performance parallel programming can pose a significant challenge for low-budget research projects. With this in mind, this paper proposes a Banana Pi single-board cluster as a financially viable solution for use in research and education. The paper also presents a comparison with another Raspberry Pi-based cluster developed in the past by the same group. During testing, it was possible to evaluate results that assess the performance and scalability of NAS benchmark applications. A maximum speedup of 13.18 was achieved with 165% efficiency using 8 cores for the class-B CG workload.*

**Resumo.** *Clusters para programação paralela de alto desempenho podem ser um grande problema para projetos de pesquisas com baixo orçamento. Pensando nisso, esse artigo propõe um cluster de Banana Pi single boards como solução viável financeiramente para uso em pesquisa e ensino. O artigo apresenta também uma comparação com outro cluster baseado em Raspberry Pi desenvolvido no passado pelo mesmo grupo. Durante os testes, foi possível avaliar resultados que verificam o desempenho e escalabilidade de aplicações do NAS benchmarks. Um speedup máximo de 13.18 foi alcançado com uma eficiência de 165% com 8 núcleos para a carga CG de classe B.*

## 1. Introdução

A computação paralela (Huang, 2022) é uma forma de diminuir o tempo de processamento de um algoritmo, executando diversos trechos de um código simultaneamente. Além disso, a mesma tem potencial de utilizar melhor a capacidade computacional de uma determinada arquitetura de *hardware*. Entretanto, as arquiteturas de computadores para alto desempenho acabam tendo um alto custo de aquisição para executar códigos com um alto nível de paralelismo. Uma alternativa mais acessível são os *clusters* de computadores.

Um *cluster* (Yeo et al., 2006) é um conjunto de computadores independentes que são interconectados e trabalham em conjunto para executar processos. Assim, é habitual utilizar *clusters* compostos por computadores pessoais ou servidores de maior desempenho. Então, a ideia central é utilizar desses computadores para que, interconectados a uma rede de alto desempenho, aumentem o poder de processamento, a capacidade de armazenamento e a eficiência do sistema.

---

\*Os autores agradecem ao CNPq, à FAPEMIG e ao FIP PUC Minas pelo suporte na execução dessa pesquisa.

<sup>†</sup>Marcos Silva é aluno de iniciação científica.

No entanto, um problema recorrente é a falta de recursos para a aquisição de *clusters* de alto desempenho. Nesse artigo, busca-se uma solução mais viável de *cluster* para pesquisas científicas, com viés educacional, que exploram o paralelismo como estratégia para aumento do desempenho. Assim, para atingir esse objetivo esse trabalho aborda o desenvolvimento de um *cluster* de Banana Pi *single boards*, que possui um baixo custo de aquisição e manutenção. Para realizar os testes, cargas de trabalho do *NAS Parallel Benchmarks*<sup>1</sup> são utilizadas.

O presente trabalho está dividido na seguinte forma: a Seção 2 apresenta alguns trabalhos relacionados; a Seção 3 apresenta conceitos fundamentais abordados nesse artigo; a Seção 4 apresenta a descrição dos materiais e método utilizados para o experimento; a Seção 5 apresenta e discute os resultados obtidos pelos experimentos; e, por fim, a Seção 6 apresenta a conclusão.

## 2. Trabalhos Correlatos

Esta seção apresenta os principais trabalhos que abordam o uso de *clusters* baseados em dispositivos *single boards*.

Garcia e Freitas (2015), abordaram a construção de um *cluster* composto por 12 Raspberry Pi quad-core. Com o foco de atingir uma maior eficiência energética com um bom *speedup*, o *cluster* obteve um *speedup* máximo de 23,39 vezes, apresentando uma eficiência de 48,7%.

No trabalho de Saffran et al. (2017), é abordado a utilização do mesmo *cluster* composto por 12 Raspberry Pi para utilização em algoritmos de mineração de dados. Visando consumir menos energia, os algoritmos *Apriori* e *Kmeans* alcançaram 88,35% e 85,17% menos energia consumida se comparado a um computador de alto desempenho convencional.

Os autores Adnan et al. (2020) apresentam a elaboração de um conjunto de testes em *single node* (uma placa) e *multi nodes (single-board cluster)* todos executando o *framework Hadoop*. O foco dos testes é testar a diferença em utilizar armazenamentos microSD e *Network Attached Storage (NAS)* entendendo qual impacto isso teria no desempenho final.

No trabalho dos autores Matos et al. (2022), é abordado a utilização de um *cluster* de Raspberry Pi focado em aprendizado de máquina para o problema de câncer de pele. O objetivo era determinar se um *cluster* com 4 Raspberry Pi seria suficiente para treinar a rede e alcançar uma acurácia de 89%. Constatou-se que o *cluster* não era capaz de atingir esse nível de precisão; no entanto, foi possível alcançar uma acurácia de 80% com os recursos disponíveis.

Donca et al. (2024) apresentam um *cluster* de Raspberry Pi dentro de um ambiente kubernetes com um sensor de ambiente monitorando o *cluster*. Essa pesquisa aborda lacunas críticas de segurança em sistemas de monitoramento ambiental contemporâneos, e apresenta uma estrutura que integra ferramentas de segurança avançadas.

Os autores Di Pierro e Hank (2024) apresentam um *cluster* composto por CPU (Raspberry Pi) e GPU (Nvidia Jetson) para implementação de aplicações de fluidodinâmica computacional. A conclusão é de que um *cluster* composto por *single boards* é mais eficiente em energia.

Ignácio e Dias (2023) apresentam um estudo de algoritmos paralelos em Raspberry Pi. Para a entrada de 5 milhões de números, os *speedups* obtidos para o algoritmo *OpenMP* foi de 3x e 1.7x, comparado com as implementações sequencial e MPI, respectivamente.

---

<sup>1</sup>URL: <https://www.nas.nasa.gov/software/npb.html>

Todos os trabalhos correlatos apresentados abordam *clusters* feitos com *single boards* e aplicações paralelizadas. A diferença apresentada neste artigo refere-se à execução de três cargas do NAS Parallel *benchmarks* em comparação com outro *cluster* baseado em Raspberry Pi.

### 3. Conceitos Principais

Esta seção apresenta alguns conceitos utilizados no trabalho.

#### 3.1. Benchmarking

*Benchmarking* (Poenaru et al., 2021) é um processo de comparação de desempenho de um *software* ou *hardware* com uma referência conhecida, ou outro *software* semelhante.

#### 3.2. Speedup e Eficiência

*Speedup* (Schryen, 2024) é uma métrica extraída a partir da comparação do tempo de execução com mais de um núcleo com o tempo de execução de um núcleo. O *speedup* pode ser definido por:

$$S = \frac{T_1}{T_p}$$

Onde:

- $T_1$  é o tempo de execução do programa usando um único núcleo (tempo sequencial).
- $T_p$  é o tempo de execução do programa usando  $p$  núcleos.

Essa métrica tem como finalidade, entender o ganho de desempenho obtido a partir de uma paralelização se comparado ao tempo sequencial. Já a eficiência visa entender, em relação ao sequencial, o quanto o paralelo utiliza de forma eficiente todos os núcleos. Eficiência pode ser definida por:

$$E = \frac{S}{p} = \frac{T_1}{p \times T_p}$$

Onde:

- $S$  é o *speedup*.
- $p$  é o número de núcleos (ou processadores) utilizados.

#### 3.3. Message Passing Interface

*Message Passing Interface* (MPI) é uma especificação e uma biblioteca amplamente utilizada para programação paralela em sistemas distribuídos e *clusters* de computadores (Kumari e Singh, 2020). Além disso, ele oferece um conjunto de funções e de rotinas que permitem a comunicação e a coordenação eficiente entre processos ou threads em um ambiente paralelo (Huang, 2022). O MPI é adotado amplamente em computação de alto desempenho (Qian, 2016) e supercomputação (Oyanagi, 2022), onde os programas podem ser executados em várias máquinas interconectadas. Logo, ele permite a passagem de mensagens entre processos, sincronização, distribuição de tarefas e comunicação colaborativa, tornando-o uma ferramenta essencial para resolver problemas complexos que exigem o processamento paralelo e a escalabilidade em recursos computacionais distribuídos.

### 3.4. Cluster

Um *cluster* é um conjunto de computadores interconectados que trabalham unidos como uma única entidade para resolver problemas computacionais complexos. Além disso, cada computador no *cluster*, chamado de “nó”, contribui com seu poder de processamento e recursos de armazenamento, permitindo a divisão de tarefas e a execução paralela de processos. Ou seja, *clusters* são frequentemente usados em aplicações que requerem alto desempenho, escalabilidade e confiabilidade (Yeo et al., 2006). Por exemplo, computação de alto desempenho, análise de dados, processamento distribuído e servidores de aplicativos (Karim et al., 2020). Sendo assim, eles oferecem uma solução eficaz para lidar com cargas de trabalho intensivas, distribuindo o processamento e melhorando a redundância. Logo, tornando-os essenciais em muitos cenários de computação de alto desempenho e empresas que buscam atender às crescentes demandas de processamento e armazenamento de dados.

## 4. Metodologia

Esta seção apresenta os materiais e método adotados visando obter os resultados da pesquisa.

Como sistema operacional, será utilizado *armbian* em todos os nós do *cluster*, além disso, será utilizado MPI para troca de mensagens entre os componentes do *cluster*. Para os códigos a serem executados em paralelo, será utilizado C como linguagens de programação dos algoritmos. Enfim, como *hardware* serão utilizados dez (10) Bananas Pi como componentes do *cluster* (*Allwinner A20 Dual-core 1.0GHz CPU, 1 GB DDR3 memory*), *switch* de 32 portas para conectar os Bananas Pi e alguns componentes adicionais para montagem do *cluster* de forma que tenha ventilação, energia e segurança adequada.

Para medir o desempenho, foram selecionados dois *kernels* e uma pseudo-aplicação do *NAS Parallel Benchmarks* (Bailey et al., 1991), o que possibilita diferenças computacionais em termos de poder de processamento, modelo de comunicação e arquitetura. O NPB foi desenvolvido pela Divisão de Supercomputação Avançada da NASA (*National Aeronautics and Space Administration*). Esse conjunto de *benchmarks* é utilizado para avaliar o desempenho de supercomputadores altamente paralelos. As seguintes cargas de trabalho do *NAS Parallel Benchmarks* foram selecionadas:

- *Embarrassingly Parallel* (EP) é um *kernel*, que gera pares de desvios aleatórios gaussianos de acordo com um esquema específico e tábua o número de pares quadrados em sucessivos círculos. A única necessidade de comunicação é para combinação da computação de vários processos no final.
- *Conjugate Gradient* (CG) é um *kernel*, que resolve sistemas esparsos de equações lineares, que são comumente encontrados em aplicações científicas e de engenharia. O benchmark mede o desempenho de algoritmos iterativos para a solução desses sistemas, com foco na eficiência da comunicação e sincronização entre processos paralelos.
- *Block Tri-Diagonal Solver* (BT) é uma pseudo-aplicação, que é parte essencial da dinâmica dos fluidos computacional. O BT resolve um sistema sintético de equações diferenciais parciais não lineares, o que envolve dependências de dados globais.

Para o projeto os nós foram configurados individualmente com o sistema Operacional *Armbian 23.02.2 Bullseye*. Além disso, foi configurado o MPI em todos os nós do *cluster* (Banana PI) para interagirem entre si por meio de um nó principal (*front end*). Após finalizar as configurações utilizou-se a técnica de *benchmarking*. Cada carga foi executada 5 vezes e foi calculada a média. Assim, foram utilizados tempo decorrido em segundos, intervalo de confiança  $C_i$ , *speedup* e eficiência.

Dessa forma, nos testes de *benchmarking* foram utilizadas 3 cargas (BT, CG, e EP) e elas possuem um parâmetro de classe que define o tamanho do problema (W, S, A, B, C, D, E em ordem crescente de tamanho). Sendo assim, foram executados o BT com a classe A, CG com a classe B e o EP com a classe C.

Entretanto, algumas classes possuem limitações na quantidade de processos simultâneos. Assim, a classe BT só executa com  $n$  (número de processos) sendo raízes perfeitas. Já o CG só executa com  $n$  sendo potências de 2 e o EP não possui limitações.

## 5. Resultados e Discussão

Conforme resultados (Figuras 1, 2 e 3), é possível verificar que o *cluster* em todas as classes desempenhou melhor que um único Banana pi como esperado. Entretanto, é possível notar que ao atingir um determinado ponto em todas as classes ele parou de escalar, o que é chamado de ponto de saturação.

Conforme Figura 1 (BT-A), o tempo até o ponto de saturação foi melhor em todos os casos em relação ao *cluster* de Raspberry Pi (Garcia e Freitas, 2015). É possível observar, que o problema deixa bem claro a importância da comunicação entre os nós do *cluster* e quanto isso impacta no desempenho. Há uma inclinação na reta maior no *cluster* de Banana, pois a velocidade de comunicação pode chegar até 1Gbps, ao contrário do de Raspberry Pi com uma velocidade máxima de 100Mbps. No entanto, o *cluster* de Raspberry Pi escalou até 36 *threads* ao contrário de 16 *threads* para o *cluster* de Banana Pi. Isso pode ser explicado pela maior quantidade (dobro) de núcleos por placa. O *cluster* de Banana Pi, para 4 núcleos, atingiu um tempo de execução de 291,94s. Já a execução sequencial alcançou um tempo de 1083,45s. Dessa forma, foi obtido uma eficiência de 93% e um *speedup* de 3,71.

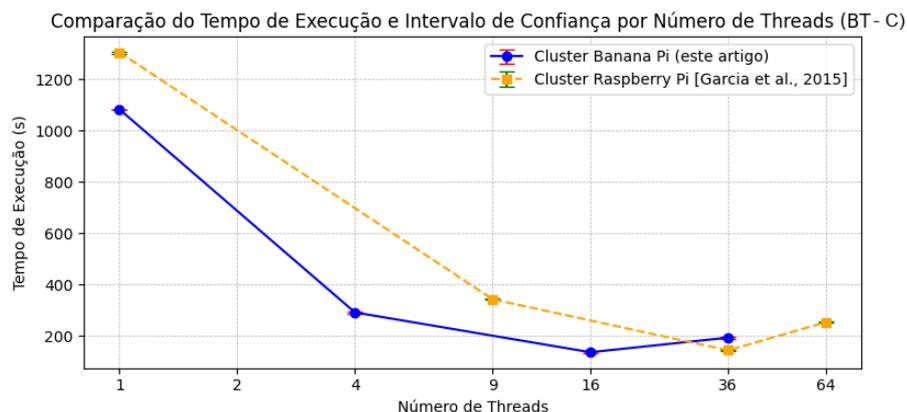
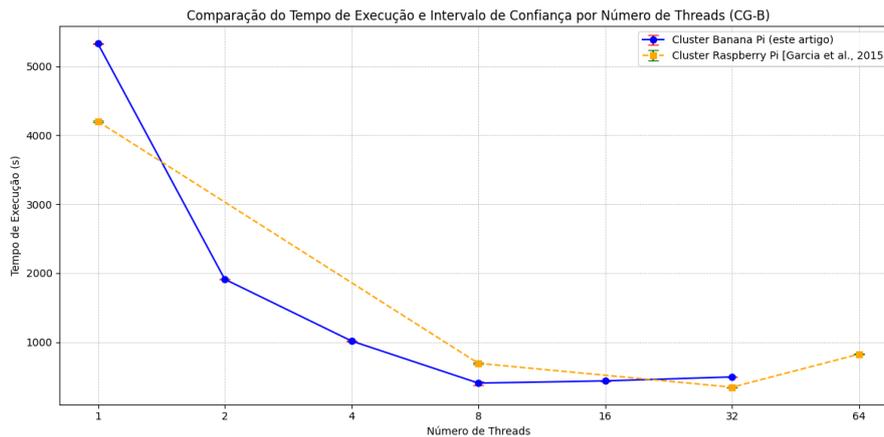


Figura 1. Gráfico de execução BT-A

Ao analisar os resultados do CG-B, é visto na Figura 2, que o ponto de saturação do *cluster* de Banana Pi foi anterior ao *cluster* de Raspberry Pi. Analisando melhor, é possível verificar que com 1 núcleo o *cluster* de Banana Pi teve um pior desempenho comparado ao de Raspberry. Porém, com 8 núcleos houve uma maior eficiência, assim como após o ponto de saturação uma tendência de aumento menor que a do *cluster* de Raspberry Pi. Dessa forma, foi possível perceber que a maior largura de banda de comunicação do *cluster* de Banana Pi possibilita um melhor desempenho. Durante os testes, foi possível chegar ao resultado de 165% de eficiência e um *speedup* de 13,18 com 8 núcleos. Para estes resultados, o tempo de execução da versão paralela foi 404,36s e o tempo de execução da versão sequencial foi de 5328,77s.

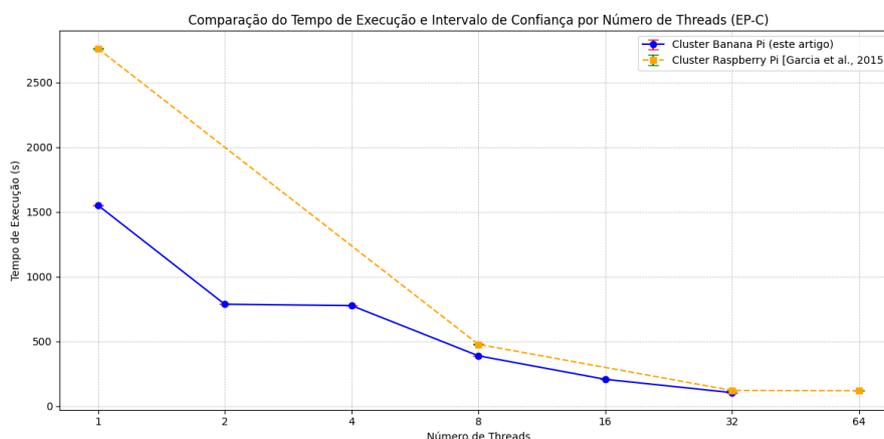


**Figura 2. Gr fico de execu o CG-B**

Em rela o ao resultado do EP-C presente na Figura 3,   poss vel constatar o mesmo comportamento, aonde o ponto de satura o do *cluster* de Banana Pi ocorre antes do ponto de satura o do *cluster* de Raspberry Pi. Para esta carga, o *cluster* de Banana Pi obteve um melhor desempenho e escalabilidade at  32 *threads*. Foi poss vel alcan ar com 6 n cleos, um tempo de execu o foi 258,82s. J  o sequencial fez um tempo de 1589,89s. Para estas execu es, a efici ncia foi de 102% para um *speedup* de 6,14.

Ao analisar os resultados obtidos do *cluster* e compar -los com os resultados do trabalho de Garcia e Freitas (2015), foi poss vel perceber que mesmo com 10 Bananas Pi e 20 n cleos no total, o desempenho do *cluster* de Banana Pi foi melhor que o de Raspberry Pi em muitas varia es de *threads*. H  uma quantidade menor de n cleos por placa, por m, o *cluster* de Banana Pi tira vantagem da largura de banda de 1Gbps. Em rela o ao ganho superlinear, uma hip tese   uma melhor utiliza o de mem ria em compara o a apenas 1 n cleo, uma vez que com mais *threads* a carga est  distribu da entre mais n cleos e n s (placas) do *cluster*. No entanto, esta hip tese ser  verificada em trabalhos futuros.

O *cluster* de Banana Pi   uma op o vi vel para universidades e institui es que n o possuem investimento suficiente para servidores e *clusters* de alto desempenho.   poss vel comprar um placa, em valores atuais, por aproximadamente R\$160,00.



**Figura 3. Gr fico de execu o EP-C**

## 6. Conclusão

A proposta desse artigo apresentar uma avaliação de desempenho e escalabilidade de uma solução viável de um *cluster* para programação paralela com um baixo orçamento e baixo custo energético para que estudantes universitários tenham mais opção no desenvolvimento de suas pesquisas científicas. Para cumprir esse objetivo, foi implementado um *cluster* com 10 Bananas Pi, utilizando o NAS como *benchmark* para avaliação dos resultados de desempenho e escalabilidade. Foram utilizados dois *kernels* (EP e CG) e uma pseudo-aplicação (BT) para os testes. Alguns resultados chegaram a ultrapassar mais de 100% de eficiência.

Esse projeto deixou como possibilidade o desenvolvimento de diversos trabalhos futuros, tais como: avaliação do consumo energético; testes de escalabilidade com foco em outras aplicações e classes do NAS e outros *benchmarks* disponíveis, e.g., CAP Bench (Souza et al., 2017), buscando uma comparação entre diversos dispositivos e arquiteturas, o que envolve *clusters* de alto desempenho ou de eficiência energética. Entretanto, é importante salientar que esse trabalho possui algumas limitações, uma vez que os Bananas Pi e Raspberries Pi utilizados são versões mais antigas, objetivando entender o impacto da variação em largura de banda de rede e quantidade de núcleos. Logo, versões mais atuais destes hardwares possibilitam expandir experimentos e avaliações. No entanto, ainda com foco nos dois *cluster* disponíveis, é possível executar diversos outros tipos de aplicações paralelas, voltadas para algum contexto específico (Castro et al., 2019; Rodrigues et al., 2012), para o entender melhor o desempenho e escalabilidade.

## Referências

- Adnan, Z. Tahir, C. Yohannes, e Ariel. Performance evaluation of mini single board computer in hadoop big data cluster. *IOP Conference Series: Materials Science and Engineering*, 875 (1):012037, jun 2020. doi: 10.1088/1757-899X/875/1/012037. URL <https://dx.doi.org/10.1088/1757-899X/875/1/012037>.
- D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, e S. Weeratunga. The nas parallel benchmarks. *The International Journal of Supercomputing Applications*, 5(3):63–73, 1991. doi: 10.1177/109434209100500306. URL <https://doi.org/10.1177/109434209100500306>.
- G. T. Castro, L. E. Zárate, C. N. Nobre, e H. C. Freitas. A fast parallel k-modes algorithm for clustering nucleotide sequences to predict translation initiation sites. *Journal of Computational Biology*, 26(5):442–456, 2019. doi: 10.1089/cmb.2018.0245. URL <https://doi.org/10.1089/cmb.2018.0245>. PMID: 30785342.
- B. Di Pierro e S. Hank. Cpu and gpu parallel efficiency of arm based single board computing cluster for cfd applications. *Computers Fluids*, 272:106187, 2024. ISSN 0045-7930. doi: <https://doi.org/10.1016/j.compfluid.2024.106187>. URL <https://www.sciencedirect.com/science/article/pii/S0045793024000197>.
- I.-C. Donca, O. P. Stan, M. Misaros, A. Stan, e L. Miclea. Comprehensive security for iot devices with kubernetes and raspberry pi cluster. *Electronics*, 13(9), 2024. ISSN 2079-9292. doi: 10.3390/electronics13091613. URL <https://www.mdpi.com/2079-9292/13/9/1613>.
- G. A. Garcia e H. C. Freitas. Avaliação de desempenho de um cluster raspberry pi com nas parallel benchmarks. In *Workshop de Iniciação Científica do Simpósio em Sistemas Computacionais de Alto Desempenho*, 2015.
- Y. Huang. Parallel computing and its applications. In *2022 IEEE International Conference*

- on *Artificial Intelligence and Computer Applications (ICAICA)*, pages 715–718, 2022. doi: 10.1109/ICAICA54878.2022.9844487.
- A. Ignácio e W. Dias. Análise do desempenho computacional de algoritmos paralelizados com openmp e mpi executados em raspberry pi. In *Anais Estendidos do XXIV Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 41–48, Porto Alegre, RS, Brasil, 2023. SBC. doi: 10.5753/wscad\_estendido.2023.235967. URL [https://sol.sbc.org.br/index.php/sscad\\_estendido/article/view/26447](https://sol.sbc.org.br/index.php/sscad_estendido/article/view/26447).
- M. R. Karim, O. Beyan, A. Zappa, I. G. Costa, D. Rebholz-Schuhmann, M. Cochez, e S. Deccker. Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, Volume 22, Issue 1, January 2021, Pages 393–415, 2020.
- A. Kumari e A. Singh. Message passing interface: An overview. *The IUP Journal of Computer Sciences*, Vol. XIV, No. 2, April 2020, pp. 33-42, 2020.
- E. Matos, E. Moreno, e K. Bispo. Machine learning and raspberry pi cluster for training and detecting skin cancer. In *Proceedings of the 18th International Conference on Web Information Systems and Technologies - Volume 1: WEBIST*,, pages 75–82. INSTICC, SciTePress, 2022. ISBN 978-989-758-613-2. doi: 10.5220/0011575500003318.
- Y. Oyanagi. Future of supercomputing. *Journal of Computational and Applied Mathematics Volume 149, Issue 1, 1 December 2002, Pages 147-153*, 2022.
- A. Poenaru, W.-C. Lin, e S. McIntosh-Smith. A performance analysis of modern parallel programming models using a compute-bound application. In B. L. Chamberlain, A.-L. Varbanescu, H. Ltaief, e P. Luszczek, editors, *High Performance Computing*, pages 332–350, Cham, 2021. Springer International Publishing. ISBN 978-3-030-78713-4.
- D. Qian. High performance computing: a brief review and prospects. *National Science Review*, Volume 3, Issue 1, March 2016, Page 16,, 2016.
- L. M. Rodrigues, L. E. Zárate, C. N. Nobre, e H. C. Freitas. Parallel and distributed kmeans to identify the translation initiation site of proteins. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1639–1645, 2012. doi: 10.1109/ICSMC.2012.6377972. URL <https://doi.org/10.1109/ICSMC.2012.6377972>.
- J. Saffran, G. Garcia, M. A. Souza, P. H. Penna, M. Castro, L. F. W. Góes, e H. C. Freitas. A low-cost energy-efficient raspberry pi cluster for data mining algorithms. In *Euro-Par 2016: Parallel Processing Workshops*, pages 788–799, Cham, 2017. Springer International Publishing. ISBN 978-3-319-58943-5. doi: 10.1007/978-3-319-58943-5\_63. URL [https://doi.org/10.1007/978-3-319-58943-5\\_63](https://doi.org/10.1007/978-3-319-58943-5_63).
- G. Schryen. Speedup and efficiency of computational parallelization: A unifying approach and asymptotic analysis. *Journal of Parallel and Distributed Computing*, 187:104835, 2024. ISSN 0743-7315. doi: <https://doi.org/10.1016/j.jpdc.2023.104835>. URL <https://www.sciencedirect.com/science/article/pii/S0743731523002058>.
- M. A. Souza, P. H. Penna, M. M. Queiroz, A. D. Pereira, L. F. W. Góes, H. C. Freitas, M. Castro, P. O. Navaux, e J.-F. Méhaut. Cap bench: a benchmark suite for performance and energy evaluation of low-power many-core processors. *Concurrency and Computation: Practice and Experience*, 29(4):e3892, 2017. doi: <https://doi.org/10.1002/cpe.3892>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3892>. e3892 cpe.3892.
- C. S. Yeo, R. Buyya, H. Pourreza, e R. Eskicioglu. *Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers*. Handbook of Nature-Inspired and Innovative Computing (pp.521-551), 2006.