

# Medições de Consumo de Energia em Arquiteturas HPC

Laura R. Soares<sup>1</sup>, Otho José Sirtoli Marcondes<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

lrsoares@inf.ufrgs.br, otho.marcondes@inf.ufrgs.br

## 1. Introdução

A atividade humana contribui fortemente para o aumento da temperatura global por meio da emissão de Gases de Efeito Estufa (GHG, do inglês *Greenhouse Gases*), dos quais o dióxido de carbono (CO<sub>2</sub>) é o mais comum. A principal fonte de geração de CO<sub>2</sub> é a queima de combustíveis fósseis (carvão, petróleo e gás natural). Em 2023, a concentração de CO<sub>2</sub> na atmosfera aumentou cerca de 50% em relação aos níveis pré-industriais [Friedlingstein et al. 2023]. Emissões net-zero (equilibrar a emissão de CO<sub>2</sub> removendo a mesma quantidade da atmosfera), apesar de serem uma abordagem popular para combater as mudanças climáticas, provavelmente vão levar o clima a um estado diferente do que seria se evitadas completamente as emissões vindas de combustíveis fósseis [Zickfeld et al. 2023]. Portanto, é crucial que todas as áreas da atividade humana reavaliem suas necessidades energéticas e incentivem o uso de fontes de energia limpas e livres de CO<sub>2</sub> em sua cadeia produtiva.

Apesar do cenário crítico na esfera ambiental, a demanda por energia está maior do que nunca. Estima-se que *data centers* vão consumir 1,5% de toda energia produzida no mundo em 2023, estimulados pelos recentes desdobramentos na área de Inteligência Artificial (AI) especificamente em Grandes Modelos de Linguagem (LLM, do inglês *Large Language Models*) [Chen 2025]. Em prognósticos de curto prazo, até 2026 a estimativa é que o gasto energético atinga o dobro do valor de 2022–ano em que o consumo do setor foi de 460 terrawatts-hora (TWh) [Vasconcelos 2025]. Apenas nos Estados Unidos, é esperado que consumo total de energia elétrica por *data centers* alcance até 12% do consumo total do país até 2028, totalizando 580 TWh no ano [Shehabi et al. 2024]. É crucial que *data centers* e demais infraestruturas apresentem ferramentas robustas de monitoramento energético para garantir que suas demandas por eletricidade sejam condizentes com sua carga de trabalho e não resultem em desperdício.

Dentro do ecossistema de parques computacionais complexos, qualquer otimização de performance energética depende de ferramentas de monitoramento capazes de informar o custo associado às tarefas—seja em nível de aplicação, do nó, do cluster, e até mesmo do data center como um todo. O MareNostrum5 por exemplo, do Centro de Supercomputação de Barcelona (BSC) [Banchelli et al. 2025], usa o framework EAR (Energy Aware Runtime) para fazer o monitoramento preciso do consumo de componentes e tarefas. O EAR, framework de código aberto<sup>1</sup>, utiliza uma combinação de componentes de hardware fornecidos por cada fabricante—como por exemplo, contadores RAPL, IPMI, Perf, monitores da nvideia, entre outros [Corbalan et al. 2020]—para monitoramento e também para possibilitar otimizações energéticas. O framework é particularmente relevante na hora de estimar o impacto da tecnologia de resfriamento líquido usada no MareNostrum, necessária devido aos nós computacionais densamente compactados no ambiente.

---

<sup>1</sup><https://github.com/eas4dc/EAR>

Esse trabalho busca trazer a discussão sobre monitoramento e eficiência energética para o Parque Computacional de Alto Desempenho (PCAD) do Grupo de Processamento Paralelo e Distribuído (GPPD) do Instituto de Informática da UFRGS. A proposta investiga quais métricas estão disponíveis para monitoramento e investiga os detalhes de sua viabilização. Através de réguas programáveis de distribuição de energia (PDUs, do inglês power distribution units) e protocolos de gerenciamento de redes como SNMP foram feitas algumas medições de consumo de energia de experimentos de HPC. Essas medições serviram como prova de conceito para fundamentar a necessidade de uma estrutura de observabilidade dentro do PCAD, e também para definir os próximos passos para sua implementação. Essa estrutura pretende possibilitar trabalhos futuros sobre análises de medições experimentais de energia, o que, por sua vez, permite o desenvolvimento de sistemas mais robustos que não tenham necessidade de provisionamento excessivo—contribuindo assim para aplicações mais sustentáveis.

Esse artigo está dividido da seguinte forma: a Seção 2 traz informações sobre o estado-da-arte em ferramentas de monitoramento de energia em outros parques computacionais e supercomputadores. A Seção 3 detalha os protocolos e métricas usados nas medições, além da aplicação escolhida para as execuções. A Seção 4 explica nossa abordagem e como foram conduzidos os experimentos deste trabalho. As medições feitas no ambiente do PCAD estão na Seção 5 e, por fim, a Seção 6 conclui o trabalho.

## 2. Trabalhos Relacionados

W.I.P.

## 3. Background

As PDUs programáveis abastecendo os *racks* no PCAD fornecem métricas de energia como corrente (V), voltagem (A), frequência (Hz), energia acumulada (kWh), e potência ativa (W), além de sensores de umidade, temperatura, entre outros. Essas métricas podem ser acessadas através de *requests* SNMP de dentro da rede interna do PCAD. Essa seção aborda o básico sobre o protocolo SNMP, a fatoração LU (aplicação escolhida para as execuções) e as métricas de energia usadas nas medições.

### 3.1. SNMP: Simple Network Management Protocol

O SNMP (em português, Protocolo Simples de Gerência de Redes) é um dos protocolos mais usados para monitoramento e gerenciamento de equipamentos como roteadores, *switches*, servidores, entre outros. O protocolo possui três versões em utilização (SNMPv1, SNMPv2c e SNMPv3). O padrão definido pela IETF [Fedor et al. 1990] inclui também as estruturas de dados usadas pelo protocolo.

Tipicamente, o ambiente de gerenciamento é composto por um ou mais dispositivos sendo monitorados (chamados de agentes) e o dispositivo gerente fazendo o monitoramento. As *requests* SNMP, por exemplo GET e SET, são feitas do gerente para o agente. Esses *requests* se baseiam em estruturas em forma de árvore chamadas de MIBs (do inglês *management information base*, ou base de informações de gerência) em que cada folha armazena um OID (identificadores de objeto). Cada OID corresponde a uma informação de gerenciamento que pode ser lida ou configurada através de *requests* SNMP.

As MIBs contendo as informações de quais dados podem ser fornecidos via SNMP para o gerente podem vir de diversas fontes, como entidades padronizadoras e fabricantes de equipamentos. A sintaxe utilizada está definida no RFC 2578 [Schönwälter et al. 1999]. Geralmente,

o equipamento a ser monitorado é embarcado pelo fabricante com as MIBs necessárias já instaladas. As mesmas informações precisam estar presentes no gerente. Por serem estruturas complexas, diversos gerentes SNMP integram "MIB browsers" para facilitar a visualização das informações.

### 3.2. Fatoração LU

Neste trabalho, utilizamos a implementação da fatoração LU do Chameleon [Agullo et al. 2010]. Como mostrado na Figura 1, a fatoração LU de uma dada matriz  $A$  é definida como  $A = LU$ , onde  $L$  é uma matriz triangular inferior e  $U$  é uma matriz triangular superior. O algoritmo de LU se baseia em três diferentes kernels do LAPACK: DGTRF–NOP IV, DTRSM e DGEMM. Essa aplicação tende a ser dominada por kernels DGEMM quando  $N$  é grande, o que torna obrigatório que as submatrizes estejam bem distribuídas entre os nós.

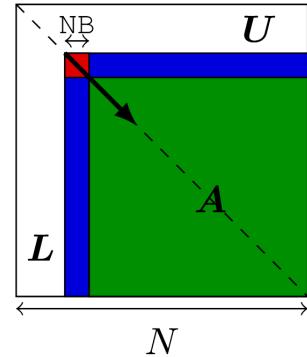
---

```

for (k = 0; k < N; k++)
    DGTRF-NOP IV (RW, A[k][k]);
    for (m = k+1; m < N; m++)
        DTRSM (RW, A[m][k], R, A[k][k]);
        DTRSM (RW, A[k][m], R, A[k][k]);
    for (n = k+1; n < N; n++) // Update
        for (m = k+1; m < N; m++)
            DGEMM (RW, A[m][n], R, A[m][k],
                    R, A[k][n]);

```

---



**Figura 1. Algoritmo LU (esquerda) sem pivoteamento, e regiões de  $A$  que são atualizada na iteração  $k$  (direita). [Nesi et al. 2020]**

### 3.3. Métricas de Consumo de Energia

W.I.P.

## 4. Metodologia

Nosso estudo tinha como foco a observação do padrão energético de dois clusters do PCAD, sendo eles: Poti e Tupi. Entretanto, devido à problemas nas máquinas da partição Tupi e alta demanda da mesma por outros usuários foi feita a escolha de observar apenas a partição Poti. Os experimentos foram realizados utilizando o pacote Stress variando os parâmetros de: CPU, IO e memória. Além disso, foi realizado outro experimento utilizando a implementação da fatoração LU do pacote Chameleon [Agullo et al. 2010] e StarPU-MPI[Augonnet et al. 2012]. A fatoração LU é uma aplicação de álgebra linear comumente utilizada como carga de trabalho para clusters em diversos campos de pesquisa [Xia et al. 2023], sendo assim uma escolha ideal para nossa investigação.

As medições de consumo energético realizadas no PCAD foram feitas usando um *script bash* executando comandos SNMP GET. Essas medições foram guardadas em arquivos do tipo CSV, sendo feito o tratamento dos mesmos após as execuções para manter apenas os tempos referentes aos nossos experimentos.

## 4.1. Configuração de Hardware e Software

As especificações da partição Poti, na qual os experimentos foram conduzidos, pode ser observado na Tabela 1. A partição Poti contém 5 nós, utilizando uma rede 1-Gigabit para comunicação. No quesito software, foi utilizado o gerenciador de pacote Guix para manter as versões dos pacotes estáveis. Para a execução das aplicações, foram utilizados os pacotes Chameleon 1.3.0, StarPU 1.4.9, OpenMPI 4 e Stress 1.0.7. Além disso, a frequência dos núcleos do processador foi delimitada em 3.4GHz e utilizando apenas os P-cores no caso da fatoração LU por intermédio de variáveis de ambiente do StarPU.

**Tabela 1. Especificação do hardware da partição Poti.**

| Nome            | CPU  | Acelerador              |
|-----------------|--|-------------------------|
| poti[1,2,3,4,5] | Intel(R) Core(TM) i7-14700KF, 3.40 GHz, 28 threads, 20 cores | NVIDIA GeForce RTX 4070 |

## 4.2. Design dos Experimentos

A criação dos projetos dos experimentos foi feita pelo pacote da linguagem R `DoE.base`, sendo o resultado desse projeto salvo em seu respectivo arquivo CSV. Posteriormente, esse arquivo gerado é lido pelo *script* dos experimentos para a realização do mesmo.

### 4.2.1. Stress

Para os experimentos utilizando o pacote Stress, foram feitas execuções variando o número de nós e os seguintes parâmetros de carga da aplicação: CPU, IO e memória. O tempo para cada uma dessas execuções foi fixado em cinco minutos. Para cada configuração foram realizadas três repetições devido à alta demanda da partição. Uma amostra de algumas das parametrizações dos experimentos pode ser observada na Tabela 2.

**Tabela 2. Amostra de design dos experimentos de Stress.**

| Tempo | CPU | IO | Memória | Nós |
|-------|-----|----|---------|-----|
| 5min  | 24  | 0  | 0       | 1   |
| 5min  | 8   | 8  | 8       | 2   |
| 5min  | 0   | 12 | 12      | 5   |

### 4.2.2. Fatoração LU

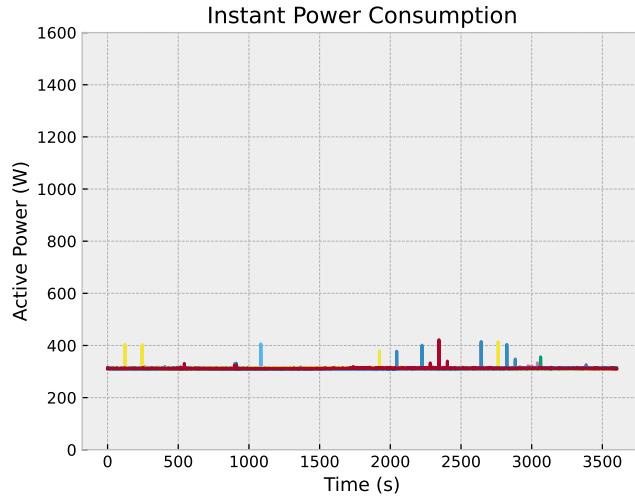
Para os experimentos utilizando a fatoração LU, foi utilizada uma matriz de tamanho  $60000 \times 60000$ , tamanho de bloco 100, variando apenas o número de nós. Para cada configuração foram realizadas três repetições devido à alta demanda da partição.

## 5. Resultados

Nós detalhamos os resultados acerca das medições de consumo de energia na partição Poti em seu estado ocioso, executando a aplicação Stress e por fim executando a fatoração LU. Para as aplicações, foram analisados os impactos do número de nós e diferentes configurações na consumo energético da partição.

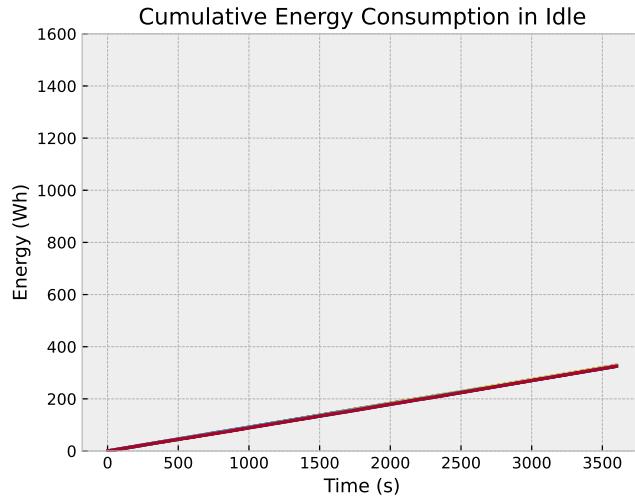
### 5.1. Partição ociosa

A Figura 2 apresenta a potência ativa em função do tempo para partição  $Pot_i$  enquanto está ociosa. É possível observar que apesar de haver alguns picos que se destacam, majoritariamente a potência ativa fica em torno de  $320W$ , o que demonstra pouca variabilidade enquanto as máquinas estão ociosas.



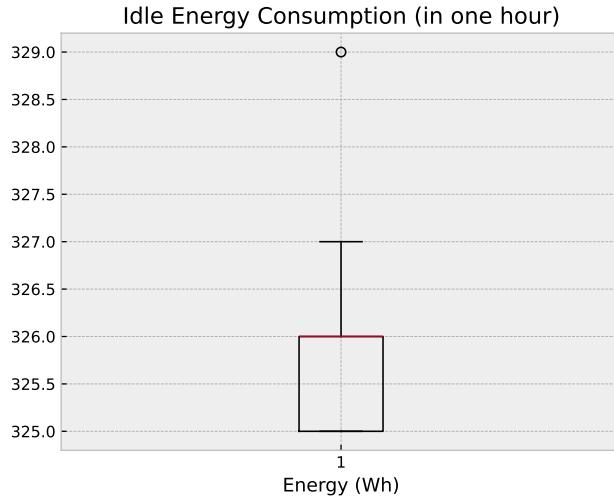
**Figura 2. Potência ativa da partição  $Pot_i$  em estado ocioso.**

A Figura 3 demonstra a energia cumulativa em função do tempo para a partição  $Pot_i$  em estado ocioso, em intervalos de uma hora. Como pode ser visualizado, o consumo de energia se mantém linear com o passar, estando de acordo com o gráfico da Figura 2 que não demonstrou grandes variações na potência ativa ao longo do tempo.



**Figura 3. Energia cumulativa da partição  $Pot_i$  ociosa.**

A Figura 4 mostra um *boxplot* da energia da partição ociosa dividida em intervalos de tempo de uma hora. É possível observar que há pouca variação entre os intervalos de tempo, tendo a energia cumulativa concentrada entre  $325Wh$  e  $326Wh$ .



**Figura 4.** Valores do consumo de energia em idle em intervalos de uma hora.

## 5.2. Stress

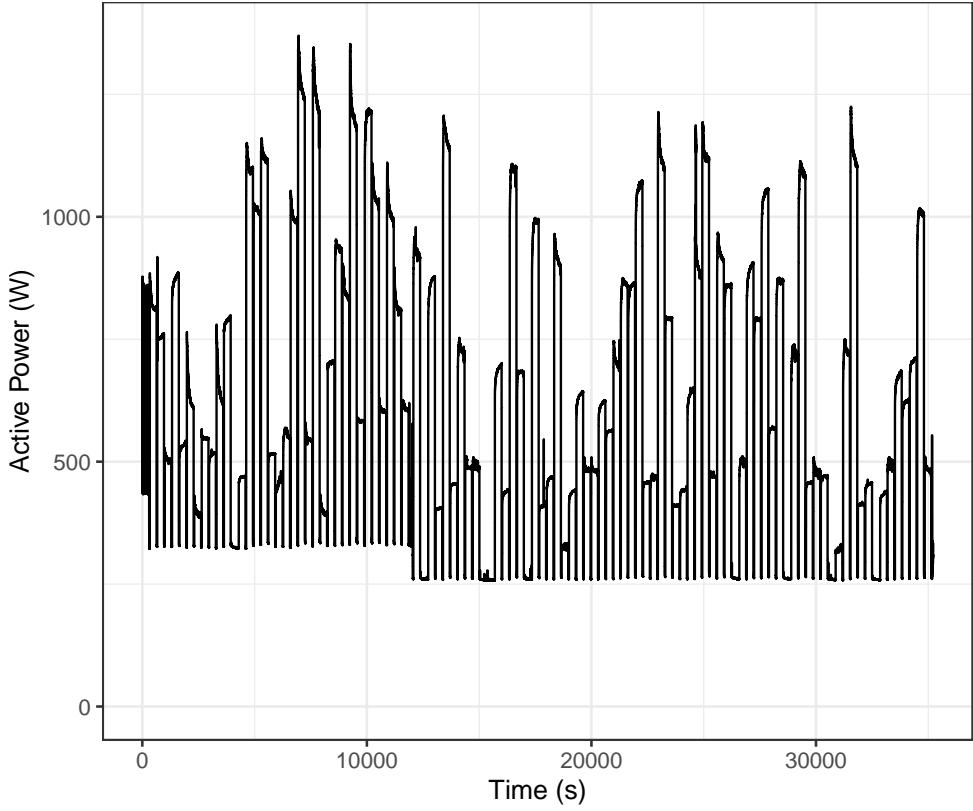
O experimento de Stress foi feito com 3 replicações, uma delas no dia 25 de setembro de 2025 e outra no dia 28 de setembro de 2025. A Figura 5 apresenta a potência ativa em função do tempo para as execuções utilizando a aplicação de Stress. Apesar desse experimento ter sido realizado em duas partes (datas diferentes), para essa visualização o tempo final da última execução do primeiro experimento foi concatenado com o tempo inicial da primeira execução do segundo experimento. Como pode ser observado, o consumo mínimo dos experimentos teve uma diferença significativa para execuções em dias diferentes. Tal observação pode ser explicada por causa de uma ventoinha de CPU da máquina `poti2` que apresentou defeito até o dia 27 de setembro de 2025. Após sanada a irregularidade, é possível perceber que mesmo sob as mesmas cargas de trabalho, a potência ativa mínima apresentou uma redução de  $\approx 20\%$ .

A Figura 6 demonstra a energia em função do número de nós para as diferentes configurações de Stress: CPU, IO e memória. Como pode ser visualizado, a carga de IO (entrada/saída) apresenta o menor consumo de energia em relação às demais aplicações. Tal comportamento pode ser explicado devido à CPU nesse tipo de carga ficar em estado de espera por boa parte da execução, o que diminui o consumo energético.

A Figura 7 apresenta a energia em função dos nós novamente, entretanto separadas com base nas diferentes cargas de Stress e nas execuções realizadas antes e após o conserto da ventoinha da CPU da `poti2`. Assim, pode ser observado o impacto do defeito da ventoinha nas execuções, nas qual a execução antes do reparo apresenta um consumo energético maior em relação as outras duas após o reparo.

## 5.3. Fatoração LU

A Figura 8 mostra os tempos de execução da aplicação de uma matriz quadrada de 60000 em função do número de nós. Pode ser observado que há uma queda de  $\approx 800s$  entre a execução de um único nó em comparação aos demais números de nós. Essa diferença de um nó para dois nós pode ser explicada pela presença do paralelismo, entretanto, as execuções com números maiores de nós apresentaram resultados aquém do esperado. Uma possível explicação pode ser o aumento da comunicação entre os nós que sobrecregou a rede, configuração de distribuição de dados inefficiente ou tamanho pequeno do problema para um número maior de nós.



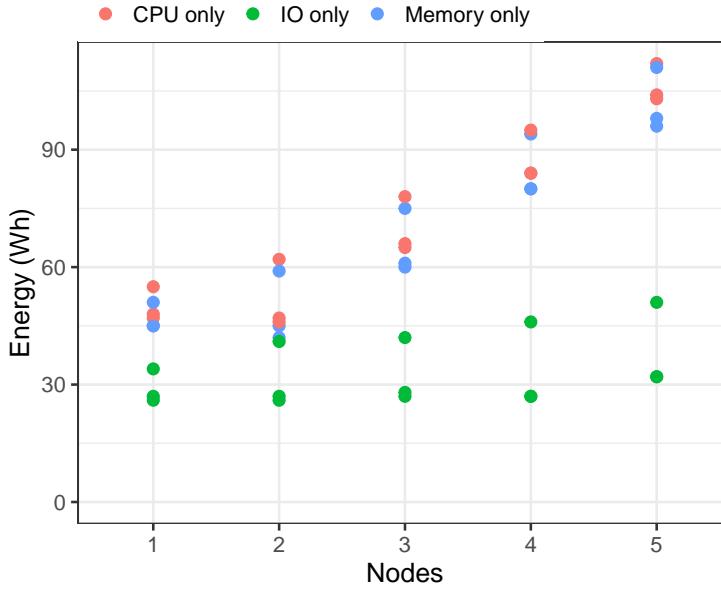
**Figura 5. Potência ativa conforme o tempo dos experimentos da aplicação Stress**

A Figura 9 apresenta a potência ativa em função do tempo para todas as execuções da aplicação. Como pode ser visualizado, há um pico da potência ativa durante as execuções. Acreditamos que seja possivelmente um erro de medição da PDU ou algum comportamento inesperado da partição durante essa execução. Como não é possível ser feita a medição de energia por nó e não foram coletados rastros durante as execuções, não podemos afirmar com certeza sobre o motivo de tal comportamento.

A Figura 10 apresenta a potência ativa em função do tempo de execução das execuções, organizadas em facetas relativas ao número de nós utilizados. Como pode ser observado, o outlier com o pico de potência ativa foi observado em uma execução com três nós, destoando do comportamento das demais execuções com a mesma configuração.

A Figura 11 apresenta o mesmo formato da Figura 10, entretanto com a retirada do outlier para melhor visualização do comportamento das execuções restantes. É possível visualizar que apesar das execuções com apenas um único nó apresentarem uma potência ativa mais constante e baixa em relação às execuções com três nós ou mais, elas possuem um tempo de execução mais extenso. Assim, não se pode confirmar que elas possuem um consumo de energia mais baixo em relação às outras com base nesse gráfico.

Os exemplos de medições dessa seção demonstram que é possível fazer a captura do perfil de consumo energético durante experimentos de HPC, mas também mostram que mudanças são necessárias na topologia elétrica do PCAD para facilitar medições de energia no futuro. Por exemplo, cada partição possui uma PDU própria ou utilizar um Wattmeter que possibilite a medição de energia por tomada, alocar os *switches* em uma régua própria, assim como fazer a separação dos demais periféricos (como monitores, *kvm switches*, entre outros) também pode reduzir a geração de interferência durante as medições de energia, o que facilitaria a



**Figura 6. Consumo de energia da aplicação Stress em relação ao número de nós.**

correlação das características do perfil de consumo energético com as tarefas do experimento.

## 6. Considerações Finais

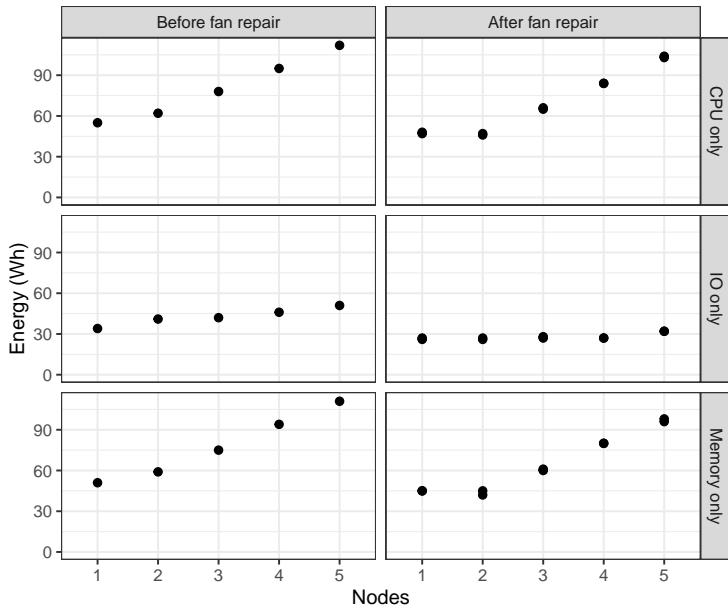
O aumento da demanda global por energia invariavelmente leva ao aumento das emissões de CO<sub>2</sub> causadas pela queima de combustíveis fósseis, dado que apenas uma pequena parcela da energia produzida no mundo vem de fontes renováveis. Projeções mostram que o consumo de energia por data centers tende a crescer e ocupar uma parcela cada vez maior da matriz energética dos países. Portanto, é vital que data centers e demais infraestruturas de TI de larga escala tenham meios de gerenciar e otimizar suas demandas energéticas.

Esse trabalho utilizou PDUs gerenciáveis para investigar a possibilidade de realizar medições de energia no PCAD do Instituto de Informática da UFRGS. Os resultados mostram que os experimentos geram perfis de consumo energético distintos que podem ser correlacionados com tarefas, e que uma ferramenta de monitoramento tem potencial de apontar otimizações no consumo de energia do parque. Como pode ser visto em nossos resultados, a falha na ventoinha da CPU de uma das máquinas produziu um aumento no consumo energético da partição Poti, o que poderia ter sido utilizado para identificação do problema de forma mais ágil. Esses resultados também argumentam fortemente em favor do desenvolvimento de *frameworks* de observabilidade utilizando, por exemplo, ferramentas de estado-da-arte como Grafana. Esses *frameworks* podem incorporar não só medições de energia como demais monitores de uso de CPU, memória, rede, entre outros.

O desenvolvimento de ferramentas de observabilidade é por si só uma contribuição científica relevante. Além disso, o uso contínuo dessas ferramentas uma vez disponíveis durante a realização de experimentos no PCAD iria potencializar a relevância dos trabalhos futuros produzidos pelo grupo, principalmente no que diz respeito ao consumo de energia por arquiteturas de HPC.

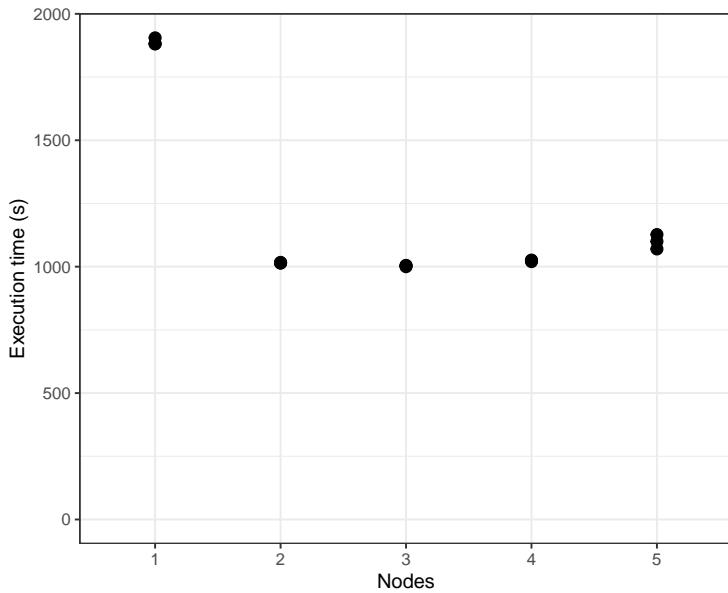
## Referências

Agullo, E., Augonnet, C., Dongarra, J., Ltaief, H., Namyst, R., Thibault, S., and Tomov, S. (2010). Faster, Cheaper, Better – a Hybridization Methodology to Develop Linear Algebra



**Figura 7. Comparação do consumo de energia da aplicação Stress antes e após o reparo.**

- Software for GPUs. In *GPU Computing Gems*, volume 2. Morgan Kaufmann.
- Augonnet, C., Aumage, O., Furmento, N., Namyst, R., and Thibault, S. (2012). Starpu-mpi: Task programming over clusters of machines enhanced with accelerators. In *European MPI Users' Group Meeting*, pages 298–299. Springer.
- Banchelli, F., Garcia-Gasulla, M., Mantovani, F., Vinyals, J., Pocurull, J., Vicente, D., Eguzkitza, B., Galeazzo, F. C. C., Acosta, M. C., and Girona, S. (2025). Introducing MareNostrum5: A European pre-exascale energy-efficient system designed to serve a broad spectrum of scientific workloads.
- Chen, S. (2025). Data centres will use twice as much energy by 2030 — driven by AI. <https://www.nature.com/articles/d41586-025-01113-z>. Access on July 17th, 2025.
- Corbalan, J., Alonso, L., Aneas, J., and Brochard, L. (2020). Energy optimization and analysis with ear. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 464–472.
- Fedor, M., Schoffstall, M. L., Davin, J. R., and Case, D. J. D. (1990). Simple Network Management Protocol (SNMP). RFC 1157.
- Nesi, L. L., Schnorr, L. M., and Legrand, A. (2020). Communication-aware load balancing of the lu factorization over heterogeneous clusters. In *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 54–63. IEEE.
- Friedlingstein et al. (2023). Global carbon budget 2023. *Earth System Science Data*, 15:5301–5369. <https://doi.org/10.5194/essd-15-5301-2023>.
- Schönwälder, J., Perkins, D. T., and McCloghrie, K. (1999). Structure of Management Information Version 2 (SMIV2). RFC 2578.
- Shehabi, A., Hubbard, A., Newkirk, A., Lei, N., Siddik, M. A. B., Holecek, B., Koomey, J., Masanet, E., Sartor, D., et al. (2024). 2024 United States Data Center Energy Usage Report.

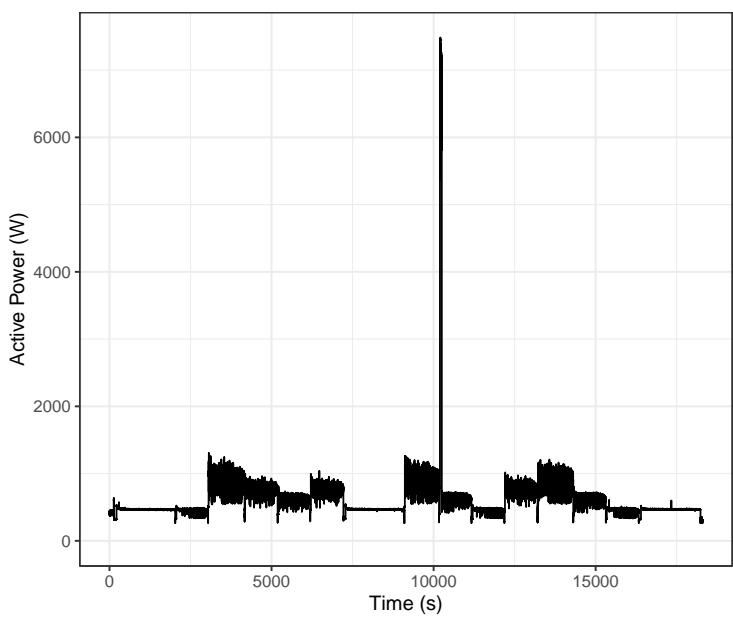


**Figura 8. Comparação do tempo de execução da Fatoração LU.**

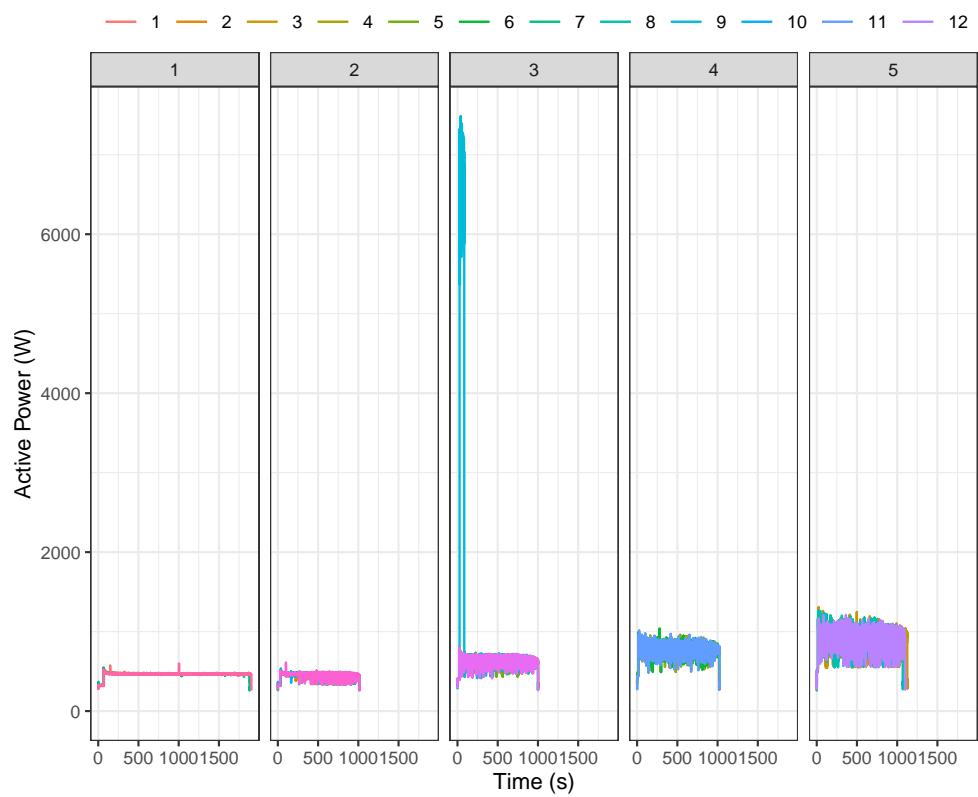
Vasconcelos, Y. (2025). Os impactos ambientais da computação. <https://revistapesquisa.fapesp.br/os-impactos-ambientais-da-computacao/>. Access on July 17th, 2025.

Xia, Y., Jiang, P., Agrawal, G., and Ramnath, R. (2023). End-to-end lu factorization of large matrices on gpus. In *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, PPoPP '23, page 288–300, New York, NY, USA. Association for Computing Machinery.

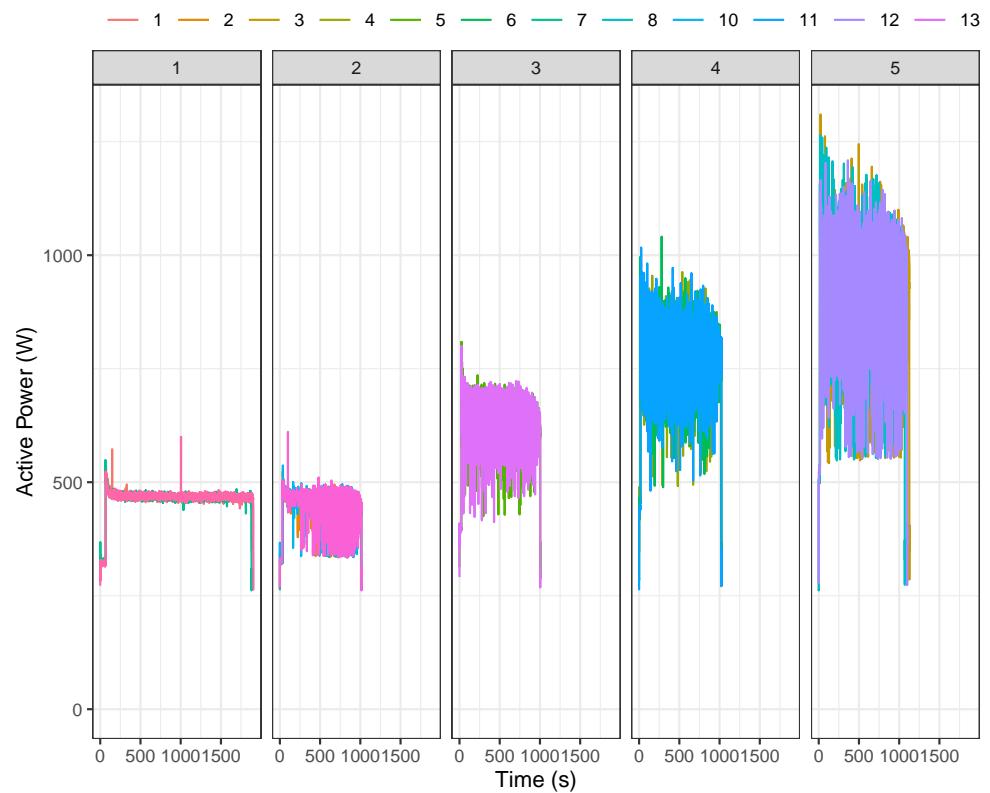
Zickfeld, K., MacIsaac, A. J., Canadell, J. G., Fuss, S., Jackson, R. B., Jones, C. D., Lohila, A., Matthews, H. D., Peters, G. P., Rogelj, J., et al. (2023). Net-zero approaches must consider earth system impacts to achieve climate goals. *Nature Climate Change*, 13(12):1298–1305.



**Figura 9.** Potência ativa dos experimentos de fatoração LU.



**Figura 10.** Potência ativa dos experimentos de fatoração LU por número de nós.



**Figura 11. Potência ativa dos experimentos de fatoração LU por número de nós (sem outlier).**