

Predição de Consumo Energético de Aplicações OpenMP em Máquinas Multi-core com Aprendizado de Máquina

Fellipe Queiroz¹, Erick Damasceno¹, Luan Siqueira¹, Marcos Amaris¹,
Thiago Rodrigues¹

¹ Universidade Federal do Pará

Programa de Pós-Graduação em Computação Aplicada (PPCA)

{fellipse.queiroz, erick.silva, luan.siqueira,
thiago.rodrigues}@tucurui.ufpa.br, amaris@ufpa.br

Abstract. *The work analyzes the energy consumption of OpenMP applications on multi-core machines using polynomial regression, highlighting that this technique is more effective than linear regression for predicting energy consumption. The CPU consumes between 82% and 88%, while DRAM accounts for 11% to 17%. With MAPE values close to 1%, third-degree polynomial regression proved efficient. There was no direct correlation between execution time and energy consumption, suggesting that task distribution optimizations can reduce energy consumption without affecting efficiency.*

Keywords: *Prediction, Performance Analysis, OpenMP.*

Resumo. *O trabalho analisa o consumo energético de aplicações OpenMP em máquinas multi-core usando regressão polinomial, destacando que essa técnica é mais eficaz que a linear para prever o consumo de energia. A CPU consome entre 82% e 88%, e a DRAM entre 11% e 17%. Com valores de MAPE próximos de 1%, a regressão polinomial de grau 3 se mostrou eficiente. Não houve correlação direta entre tempo de execução e consumo, sugerindo que otimizações na distribuição de tarefas podem reduzir o consumo energético sem afetar a eficiência.*

Palavras-Chave: *Predição, Análise de Desempenho, OpenMP.*

1. Introdução

Com o crescimento exponencial do mundo digital, os *data centers* tornaram-se essenciais para a eficiência da rede global. *Data centers* consumiram entre 240 a 340 TWh em 2022, com projeções de 752 TWh em 2030, o equivalente a 1,3% do consumo global de eletricidade e 0,3% das emissões globais de CO₂ [IEA 2023]. Esse aumento no consumo de energia levanta questões sobre a sustentabilidade dessas infraestruturas. Como resposta, a prática de *Green Computing* ganha destaque. Trata-se de uma área de pesquisa que busca uma computação mais sustentável e ecológica [Cordeiro et al. 2023].

Neste contexto, compreender o funcionamento das tecnologias atuais e desenvolver estratégias para otimizá-las é essencial. Prever o desempenho de uma aplicação ou seu consumo de energia é complexo devido à natureza frequentemente incompleta e não determinística das aplicações. No entanto, o Aprendizado de Máquina, com seu potencial para identificar relações complexas e não lineares nos dados, oferece uma solução promissora.

Este estudo visa obter informações de máquinas multi-core ao executar aplicações OpenMP, advindas do PolyBench/ACC. Os objetivos principais incluem a coleta de dados

de consumo energético e desempenho, a análise dos tempos de execução, consumos de energia e similaridades entre as aplicações testadas e o uso de técnicas de Aprendizado de Máquina, com ênfase em modelos de Regressão na tentativa de prever consumos das aplicações. O foco é não apenas entender as similaridades e disparidades das aplicações com uso do OpenMP e multithread das aplicações, mas também prever sua eficiência energética. Com isso, pretende-se criar um método de obtenção de informações para uma melhor tomada de decisão na gerência de tarefas computacionais, oferecendo uma compreensão das dinâmicas e características das aplicações analisadas.

O documento está organizado da seguinte forma: Seção 2 aborda o referencial teórico; Seção 3 apresenta os trabalhos relacionados; Seção 4 descreve a metodologia de coleta de dados e desenvolvimento dos modelos; Seção 5 discute os resultados; e Seção 6 conclui o estudo com contribuições e sugestões para trabalhos futuros.

2. Referencial Teórico

2.1. Green Computing

O aumento da demanda por poder computacional, elevou o consumo de energia e a emissão de CO₂. A Computação Verde, visa o uso eficiente de recursos computacionais, reduzindo o consumo de energia e as emissões de gases. Suas práticas incluem eficiência energética, design sustentável, gestão de resíduos, virtualização, centros de dados verdes e *Cloud Computing*.

Os problemas ambientais gerados pelo aumento da emissão de CO₂ e o custo financeiro ocasionado pelo consumo de energia têm impulsionado estudos que visam o desenvolvimento de mecanismos e tecnologias que façam uso eficiente da energia. Esses mecanismos e tecnologias são denominados computação verde ou *Green Computing* [Williams and Curtis 2008].

2.2. OpenMP

O OpenMP é uma API para programação multi-thread em sistemas Unix e Windows NT, introduzida em 1997, facilitando a paralelização de aplicações através de diretivas de compilação simples [Penha et al. 2002]. Utiliza o modelo fork-join, onde threads mestre se dividem para executar tarefas simultaneamente e depois se sincronizam. Apesar de sua padronização e portabilidade entre plataformas, enfrenta desafios em escalabilidade para muitos núcleos e complexidade em sistemas de grande escala.

2.3. Regressão Linear e Polinomial

O modelo de Regressão Linear simples é um método preditivo elementar que relaciona uma variável dependente Y com uma variável independente X, por meio da representação de uma reta [Chein 2019]. Embora possa indicar uma relação de causa e efeito, é importante notar que a Regressão Linear simples por si só não prova a causalidade.

A Regressão Polinomial é uma forma de Regressão Linear onde a relação entre a variável independente X e a variável dependente Y é modelada como um polinômio de grau n. Diferente do modelo linear simples, que só pode representar relações lineares, a Regressão Polinomial é capaz de capturar relações mais complexas e curvilíneas entre as variáveis.

2.4. Erro Percentual Médio Absoluto (MAPE)

O MAPE, é uma métrica que avalia a precisão de previsões estatísticas e modelagem preditiva. Geralmente usado para calcular a média dos erros percentuais absolutos entre valores previstos e reais, expressando essa precisão em termos percentuais. A interpretação dos valores de MAPE geralmente segue uma escala em que números menores indicam previsões mais precisas [Lewis 1982].

Tabela 1. Interpretação dos valores de MAPE.

MAPE	Interpretação
-10%	Previsão altamente precisa
10 a 20%	Previsão boa
20 a 50%	Previsão razoável
50% +	Previsão imprecisa

3. Trabalhos Relacionados

Evaluating execution time predictions on GPU kernels using an analytical model and machine learning techniques [Amaris et al. 2023], discutem várias abordagens para predição de tempo de consumo de funções CUDA executadas em placas gráficas de propósito geral (GPU). Explicam que em modelo analítico, há aplicações regulares onde modelos analíticos podem ser usados para prever desempenho, e outras mais irregulares onde técnicas de aprendizado de máquina obtém melhores previsões.

Plataforma para medir el consumo energético de algoritmos [Uribe 2022], a plataforma *PowerTester* foi desenvolvida para medida e análise de consumo energético e o desempenho de programas escritos em C++ por meio da tecnologia Intel RAPL e da ferramenta de análise *Perf*. Os dados coletados dos diferentes medidores são enviados a um servidor, organizados e disponibilizados para os usuários por meio de gráficos e arquivos CSV para análise online. A validação da plataforma foi realizada por meio de experimentos com voluntários, e a usabilidade foi avaliada através da escala SUS.

4. Metodologia

4.1. Seleção de Componentes de *Hardware* e *Software*

Como ferramentas de *hardware*, utilizou-se um processador *Intel Core i5-7200U*, com 2 núcleos físicos e 4 lógicos, operando a 2,50 GHz, e 12 GB de memória DDR4 a 2133 MHz. O sistema operacional foi o *Linux Ubuntu 24.04* com o compilador *GCC 13.2.0*. A escolha do processador se deve à sua disponibilidade e adequação a cenários de baixo consumo energético, comuns em máquinas de uso geral.

As aplicações utilizadas são provenientes do PolyBench/ACC, que deriva do conjunto de *benchmarks* PolyBench/C¹ e fornece implementações OpenMP, OpenACC, CUDA, OpenCL e HMPP. Os *benchmarks* selecionados foram:

Tabela 2. Benchmarks selecionados

2mm	3mm	Trmm
Symm	Gemm	Syr2k
Syrk	Gramschmidt	Correlation
Covariance	LU	Cholesky

¹Um conjunto de *benchmarks* projetado para avaliar o desempenho de hardwares.

Os *benchmarks* 2mm e 3mm realizam multiplicações de matrizes, Trmm testa a multiplicação de matrizes triangulares, Symm e Gemm avaliam multiplicações de matrizes simétricas e gerais. Syr2k e SyrK fazem atualizações simétricas, Gramschmidt implementa ortogonalização, Correlation e Covariance calculam correlação e covariância, LU e Cholesky realizam decomposições de matrizes.

4.2. Scripts de Coleta de Dados

A metodologia utilizou um *script* para automatizar a execução de *benchmarks*, cada um executado 101 vezes com entradas variando de 800 a 4000, incrementadas de 32, garantindo repetibilidade e análise de diferentes tamanhos de problemas.

O *script* compilou e executou os *benchmarks*, ajustando gradualmente os valores de entrada e, em paralelo, usou a ferramenta JouleIt² para monitorar e coletar dados de consumo em arquivos CSV e categorizados com base em atributos como CORE(Núcleo de processamento), CPU (Unidade Central de Processamento), DRAM (Memória Dinâmica de Acesso Aleatório), DURATION (Tempo de execução), UNCORE (Parte do processador que não é o núcleo de processamento) e INPUT (Valo fornecidos a um sistema para processamento).

As métricas de energia foram registradas em microjoules e a duração em microssegundos para capturar variações finas. A avaliação estática de código forneceu dados como INPUTS (quantidade de entradas), 1D (uso de arrays unidimensionais), ARRAYS (número de arrays), LOOPS (laços de repetição) e operações matemáticas (MULT, SOM, SUB, DIV), que ajudam a entender a complexidade e eficiência energética das aplicações.

4.3. Pré-processamento dos Dados

O pré-processamento dos dados foi conduzido usando Python no ambiente interativo Jupyter Notebook. Isso incluiu limpeza, verificação de erros e padronização das métricas para joules e segundos. Métricas irrelevantes como EXIT_CODE (Código de erros de execução), CORE foram removidas, enquanto uma nova métrica ENERGY foi criada para representar o consumo total de energia das aplicações.

4.4. Mineração de dados

As ações executadas para a mineração de dados, com foco no comportamento energético de aplicações, foram: análise de *outliers*, comparação entre consumo e entradas de dados, consumo por componentes de hardware, consumo total por *benchmark*, tempo de execução e agrupamento hierárquico para correlação entre *benchmarks* baseada no consumo energético.

4.5. Aprendizado de Máquina para Predição de Consumo energético

Foram aplicados modelos de Regressão Linear e Polinomial em duas abordagens para prever os consumos, com objetivo de compreender o comportamento atual e prever tendências futuras, garantindo um método preciso do desempenho dos modelos.

Abordagem 1: Utilização de dados específicos do *benchmark* para treinamento e previsão, variando o grau do polinômio de 1 a 3 e a carga de treinamento de 60% a 80%. Avaliação usando MAPE para comparar previsões com valores reais.

²Ferramenta utilizada para obter valores de consumo de energia e tempo de execução, faz uso da Intel Running Average Power Limit para medir em tempo real o consumo de energia dos componentes.

Abordagem 2: Combinação de múltiplos *datasets de benchmarks* para treinamento, variando o grau do polinômio de 1 a 3. Avaliação também feita com MAPE para medir a precisão das previsões.

A escolha dessas abordagens se dá pela necessidade de balancear simplicidade e eficácia na previsão de consumo energético. Optou-se por Regressão Linear e Polinomial até o grau 3 para evitar o *overfitting* e garantir interpretabilidade dos modelos, mantendo a precisão nas previsões sem a complexidade excessiva de modelos mais avançados. A variação no grau do polinômio e nas cargas de treinamento ajuda a explorar diferentes níveis de ajuste, assegurando que os modelos não apenas refletissem o comportamento atual, mas oferecessem previsões confiáveis para tendências futuras.

5. Resultados

5.1. Mineração de dados

A mineração de dados dos *benchmarks* revelou alguns pontos sobre o consumo energético e o desempenho computacional. A verificação de *outliers* mostrou que os valores de consumo energético não apresentaram anomalias, portanto, não foi necessário um tratamento.

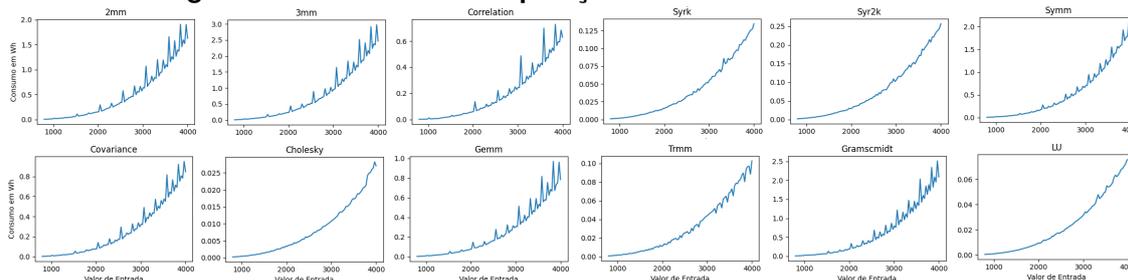
Verificou-se que a CPU é o componente de mais consome energético, representando entre 82% e 88% do consumo total, enquanto a DRAM contribui com 11% a 17%.

Tabela 3. Consumo de CPU e DRAM dos benchmarks

	2mm	Covariance	Syrk	Tmm	3mm	Cholesky	Syrk2k	Gramschmidt	Correlation	Gemm	Symm	LU
CPU	86.11%	84.72%	88.44%	88.92%	86.45%	86.28%	87.12%	84.30%	86.45%	85.80%	82.30%	83.42%
DRAM	13.88%	15.27%	11.54%	11.06%	13.54%	13.64%	12.86%	15.69%	13.54%	14.19%	17.70%	16.54%

Observou-se uma relação exponencial entre o consumo energético e a complexidade dos dados processados pelos *benchmarks* como ilustrado na Figura 1. Entre os *benchmarks* analisados, os com entradas divisíveis por 128, mostraram picos de consumo devido a possíveis ineficiências no gerenciamento de memória RAM, incluindo alinhamento ineficiente de memória, *cache misses* e *thrashing* de memória.

Figura 1. Consumo em Comparação com os Valores de Entradas.



Fonte: De autoria própria.

Ao analisar o consumo energético dos *benchmarks*, *Syrk2k* e *Syrk*, que realizam Álgebra Linear, foram os maiores consumidores. *Trmm*, *Symm*, *3mm*, *2mm* e *Gemm*, que realizam multiplicações de matrizes com algumas diferenças nas operações, apresentaram consumos semelhantes, refletindo variações nas operações específicas. *Cholesky* e *LU* mostraram consumos comparáveis por realizarem decomposição de matrizes, enquanto *Covariance*, *Gramschmidt* e *Correlation* tiveram os menores consumos, sem uma relação direta entre eles.

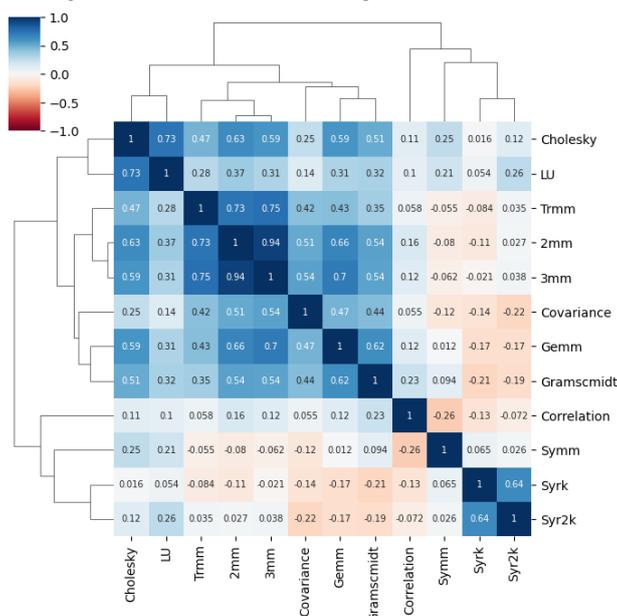
A análise do tempo de execução revelou que o consumo de energia não está diretamente relacionado à duração da execução, como mostrado na Tabela 4. *Syrk* e *Syr2k* apresentaram altos consumos com tempos curtos de execução, enquanto *Gramschmidt* teve menor consumo, mas uma execução mais longa. Isso sugere que otimizações na distribuição de tarefas em relação ao tempo, podem reduzir o consumo energético sem afetar a eficiência.

Tabela 4. Duração das aplicações em horas, minutos e segundos

	3mm	GramSchmidt	Symm	2mm	Covariance	Gemm	Correlation	Syr2k	Syrk	Tmm	LU	Cholesky
Duração	5:25:08	4:51:12	3:38:18	3:36:41	1:55:22	1:48:15	1:31:21	0:27:38	0:14:22	0:12:25	0:10:06	0:03:20
Consumo (watts)	14.50	13.00	15.25	14.00	12.30	13.50	12.85	17.40	17.00	15.55	14.25	15.25

Por fim, a análise de correlação com *cluster* hierárquico destacou padrões entre as aplicações, conforme observado na Figura 2. Relações de consumo energético foram observadas entre *Covariance*, *Gemm* e *Gramschmidt*, assim como entre *Correlation* e *Symm*. Isso indica que ajustes específicos podem ser implementados para melhorar a eficiência em aplicações similares, enquanto os isolados como *Cholesky* e *LU* podem exigir abordagens de otimização distintas.

Figura 2. Mapas de calor de correlação com *cluster* hierárquico.



Fonte: De autoria própria.

5.2. Resultado das Predições de Consumo energético

5.2.1. Abordagem 1: Predição por Datasets Próprios

Tabela 5. Valores de MAPE (%) - Regressão Linear

	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramschmidt	LU
50	67.9%	67.61%	66.53%	67.39%	61.06%	67.82%	61.15%	60.9%	69.25%	60.95%	69.67%	60.08%
60	60.2%	59.92%	58.26%	59.55%	54.04%	59.96%	53.45%	53.31%	61.47%	53.4%	61.43%	52.71%
70	52.0%	52.15%	50.6%	51.73%	46.9%	51.94%	45.74%	44.65%	53.37%	43.98%	53.63%	44.92%
80	41.35%	41.31%	39.32%	41.31%	38.98%	41.28%	36.47%	36.06%	43.31%	35.49%	42.8%	35.91%

A Regressão Linear apresentou uma taxa de acerto moderada, com valores de MAPE entre 20% e 50%, considerados razoáveis. Há uma distância significativa entre

os valores preditos e reais. A utilização de 80% dos dados para treinamento mostrou melhores resultados, conforme observado na Tabela 5, mas os valores de MAPE ainda foram altos. Portanto, o modelo não é adequado para as previsões desejadas.

Tabela 6. Valores de MAPE (%) - Regressão Polinomial Grau 2

	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramschmidt	LU
50	32.13%	31.9%	26.49%	29.23%	22.89%	33.79%	23.47%	20.69%	31.9%	22.85%	32.27%	16.05%
60	23.79%	23.92%	18.56%	21.74%	19.09%	23.59%	16.53%	15.98%	23.86%	17.55%	22.08%	14.92%
70	18.64%	19.89%	17.09%	18.61%	16.6%	18.8%	13.13%	9.56%	19.69%	8.28%	19.85%	11.88%
80	9.29%	9.0%	6.41%	9.31%	12.6%	9.42%	6.57%	7.65%	9.69%	6.85%	8.27%	6.48%

A Regressão Polinomial demonstrou maior eficiência, especialmente o modelo de segundo grau, com valores de MAPE próximos ou inferiores a 10, indicando alta precisão. A Tabela 6 expõe que os valores previstos pelo polinômio de grau 2 são próximos dos valores reais, destacando a capacidade do modelo de capturar as tendências dos dados sem *overfitting* ou *underfitting*.

Tabela 7. Valores de MAPE (%) - Regressão Polinomial Grau 3

	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramschmidt	LU
50	14.88%	14.15%	36.06%	8.35%	17.57%	25.95%	11.34%	1.89%	23.02%	8.94%	8.1%	7.48%
60	8.32%	7.69%	5.3%	7.56%	8.54%	10.55%	1.44%	1.38%	4.41%	8.19%	13.89%	7.68%
70	7.79%	8.58%	9.84%	7.68%	9.6%	8.53%	1.96%	5.83%	6.74%	11.53%	8.01%	4.44%
80	12.28%	12.86%	9.63%	9.67%	6.06%	13.89%	5.16%	1.55%	5.5%	4.05%	10.77%	3.82%

O modelo de polinômio de grau 3 foi o mais eficiente, com valores de MAPE consistentemente baixos, em alguns casos próximos a 1%. *benchmarks* como 3mm, Correlation, Covariance, Syrk e Syr2k tiveram melhor desempenho com 60% dos dados. Por outro lado, 2mm, Gemm e Gram-Schmidt precisaram de 70% dos dados, enquanto Cholesky, Trmm e LU com 80%. Isso demonstra a importância de ajustar a quantidade de dados conforme a especificidade de cada *benchmark* ao usar modelos de Polinomial.

5.2.2. Abordagem 2: União de Múltiplos Datasets

Tabela 8. Valores de MAPE para diferentes modelos de regressão

Benchmark	Regressão Linear	Polinomial Grau 2	Polinomial Grau 3
2mm	3.04%	1.86%	7.55%
3mm	6.37%	5.71%	7.06%
Correlation	8.96%	46.01%	33.42%
Covariance	62.15%	15.41%	15.90%
Cholesky	20.13%	30.26%	14.92%
Gemm	8.22%	2.81%	8.39%
Syrk	20.30%	7.61%	8.33%
Syr2k	16.23%	6.00%	4.12%
Symm	46.82%	39.33%	30.75%
Trmm	4.12%	5.87%	7.26%
Gramschmidt	41.96%	29.41%	59.52%
LU	8.11%	9.09%	4.26%

Os resultados revelam variações significativas nos valores de MAPE entre diferentes modelos de regressão. *benchmarks* como 3mm e Syrk tiveram aumento de MAPE com Regressão Polinomial de grau 3, indicando menor precisão. A Regressão Linear foi eficaz para o *benchmark* 2mm, mostrando uma relação linear entre dados de entrada e consumo energético.

A análise revelou que a complexidade dos dados exige a escolha de modelos adequados para cada *benchmark*. O modelo de grau 2 foi o mais eficiente, seguido pelo grau 3 e, por último, a Regressão Linear. Embora a abordagem unificada de *benchmarks* tenha apresentado MAPE baixos, a modelagem específica para cada benchmark foi mais precisa na maioria dos casos. A combinação de *benchmarks* pode reduzir a precisão das previsões ao diluir características únicas, e incluir mais dados de *benchmarks* pode melhorar a precisão dos modelos, indicando que mais dados são necessários para uma abordagem unificada ser eficaz.

6. Conclusão

Neste trabalho, analisou-se e previu-se o consumo energético de *benchmarks* em máquinas multi-core, focando nas relações entre consumo de energia, dados de entrada e componentes de hardware. Foram identificados padrões de consumo exponenciais, sugerindo a eficiência da Regressão Polinomial, e constatou-se que a CPU é o maior consumidor de energia, destacando possíveis otimizações. Não houve correlação direta entre tempo de execução e consumo energético. Agrupamentos hierárquicos revelaram padrões de consumo semelhantes entre *benchmarks*. A Regressão Polinomial de grau 3 mostrou-se eficaz individualmente, mas a abordagem unificada não apresentou os mesmos resultados.

Recomenda-se expandir o estudo para incluir mais *benchmarks* do PolyBench, explorar modelos avançados como Redes Neurais, analisar a correlação entre *benchmarks*, estudar o impacto de diferentes componentes de hardware e investigar a influência de fatores adicionais como configurações de sistema operacional e práticas de programação.

Referências

- Amaris, M., Camargo, R., Cordeiro, D., Goldman, A., and Trystram, D. (2023). Evaluating execution time predictions on gpu kernels using an analytical model and machine learning techniques. *JPDC*, 171:66–78.
- Chein, F. (2019). *Introdução aos modelos de regressão linear: um passo inicial para compreensão da econometria como uma ferramenta de avaliação de políticas públicas*. ENAP, Brasília.
- Cordeiro, D., Franceschini, E., Amaris, M., Castro, M., Baldassin, A., and Lima, J. (2023). Green cloud computing: Challenges and opportunities. In *Anais do XIX SBSI*, pages 129–131, Maceió/AL. SBC.
- IEA (2023). Tracking clean energy progress. <https://www.iea.org/reports/tracking-clean-energy-progress-2023>. Licença: CC BY 4.0.
- Lewis, C. D. (1982). *Industrial and Business Forecasting Methods*. Butterworths.
- Penha, D., Corrêa, J., and Martins, C. (2002). Análise comparativa do uso de multi-thread e openmp aplicados a operações de convolução de imagem. In *Anais do III Workshop em Sistemas Computacionais de Alto Desempenho*, pages 118–125, Porto Alegre, RS, Brasil. SBC.
- Uribe, D. (2022). Plataforma para medir el consumo energético de algoritmos. Universidad de Concepción.
- Williams, J. and Curtis, L. (2008). Green: The new computing coat of arms? *IT Professional Magazine*, 10(1):12.