

Evolução histórica do desempenho energético de tarefas cotidianas em uma distribuição Linux

Artur Egidio Launikas e Cupelli, Fábio Nakano, Flávio Luiz Coutinho, Daniel Cordeiro

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{artur.cupelli, fabionakano, flcoutinho, daniel.cordeiro}@usp.br

Abstract. *As software becomes more prevalent in daily life, the associated energy consumption and carbon emissions also increase. In addition to the longer time users spend interacting with applications, the complex functionalities of modern software further raise their energy demands. Using the Running Average Power Limit (RAPL), the research assesses the energy consumption of software across different versions of the Ubuntu operating system from 2008 to 2018. The experiments simulate the routine of an average user to provide a comprehensive view of energy performance over time.*

Resumo. *À medida que os softwares se tornam mais presentes no cotidiano, o consumo de energia e as emissões de carbono também aumentam. Além do maior tempo de interação dos usuários com aplicações, é possível que as funcionalidades complexas dos softwares modernos elevem ainda mais sua demanda energética. Utilizando o Running Average Power Limit (RAPL), a pesquisa avalia o consumo energético de softwares em diferentes versões da distribuição Linux Ubuntu, de 2008 a 2018. Os experimentos simulam a rotina de um usuário doméstico para obter uma visão abrangente do desempenho energético ao longo do tempo.*

1. Introdução

Nota-se cada vez mais a presença generalizada de softwares em todos os âmbitos cotidianos. Embora a presença de programas auxilie e desonere atividades humanas, há uma responsabilidade necessária com o impacto ambiental gerado por tais aplicações. A evolução da eletrônica elevou a capacidade de processamento nos aparelhos modernos, propiciando a implementação de funcionalidades computacionalmente mais complexas, tais como manipulação de textos, modelos de aprendizado de máquina, visão computacional, algoritmos de recomendação, entre diversas outras.

Porém, para tais sistemas continuarem em constante evolução, há uma degradação ambiental, principalmente relacionada às emissões de carbono da energia produzida para execução do programa, que nem sempre é considerada como métrica de desempenho. Segundo o relatório da International Energy Agency (IEA) de 2024, o consumo de eletricidade por centros de dados, inteligência artificial (IA) e o setor de cripto moedas pode dobrar até 2026 [IEA 2024]. Em 2022, os centros de dados do planeta consumiram cerca de 460 TW h (terawatts-hora) de eletricidade, e esse consumo total pode ultrapassar 1000 TW h em 2026, equivalente ao consumo elétrico total do Japão [IEA 2024]. Esse

aumento na demanda destaca a importância de atualizar as regulamentações e melhorar a tecnologia para controlar o crescimento do consumo de energia desses centros.

Uma das maneiras de mensurar a pegada de carbono de um software é através da medição do consumo energético durante sua operação. Tal medida pode ser observada com o uso de aparelhos externos ou por técnicas mais sofisticadas embutidas no hardware do computador, como a *Running Average Power Limit* (RAPL) [David et al. 2010]. RAPL é uma interface disponibilizada inicialmente por processadores Intel que provê acesso a contadores de hardware com a medição do consumo energético da CPU.

Neste projeto de Iniciação Científica, foram criados experimentos para medir o desempenho energético dos softwares com base no consumo de energia. A análise concentrou-se em como esse consumo varia ao longo do tempo para uma rotina de atividades padrão, simulando o comportamento de um usuário típico. O objetivo é caracterizar o desempenho energético de diferentes versões de um mesmo software durante atividades corriqueiras, permitindo compreender a evolução das necessidades energéticas ao longo do tempo.

2. Trabalhos Relacionados

A energia elétrica é um dos ingredientes vitais para o funcionamento do cotidiano civil, econômico, industrial e científico. Contudo, seu processo de geração é um ator importante na emissão de gases contribuintes ao efeito estufa, e conseqüentemente ao aumento da temperatura do planeta [Mathew 2022]. O fenômeno do aquecimento global é um problema alarmante e de responsabilidade multidisciplinar, inclusive da área computacional.

A execução de qualquer software impõe uma carga de trabalho na plataforma onde é executado. Aplicativos *mobile* irão consumir a bateria do aparelho celular, assim como sistemas hospedados em *data centers* utilizam recursos computacionais que precisam ser mantidos ligados. Com o aumento da presença e demanda por esses softwares no cotidiano, é automático pensar também no crescimento da demanda por energia. Portanto, torna-se essencial o desenvolvimento de sistemas cientes de seu impacto ambiental e que adotem estratégias para reduzir seu consumo.

[Pang et al. 2016] apontam em sua pesquisa que programadores possuem conhecimento limitado sobre boas práticas na redução da energia demandada pelo software que produzem. Mais de 80% dos entrevistados não consideram o gasto energético como um requisito quando desenvolvem algoritmos, e apenas 10% indicaram que medem o consumo de seus programas. O artigo também aponta a falta de compreensão dos profissionais sobre quais componentes do computador gastam mais energia. A maioria indicou que o monitor era o maior consumidor, sendo que o processador seria a resposta correta.

O primeiro passo para implementar algoritmos cientes de energia é entender as ferramentas que podem ser utilizadas para metrificá-los. [Sun et al. 2016] exploram e comparam medidores externos de energia em um contexto geral, amplamente implementados para medição periódica do consumo de residências ou aparelhos específicos. Os autores ressaltam que, uma vez conectados à Internet, os medidores inteligentes (*IoT*) fornecem acesso as medições em plataformas remotas, facilitando a análise dos resultados. Uma das etapas desta pesquisa foi estender o uso de um medidor inteligente conectando-o a um computador e medindo sua demanda elétrica durante a execução de um programa.

Contudo, existem técnicas mais adequadas e precisas para medição deste consumo por softwares. [Raffin and Trystram 2024] exploram a tecnologia *Running Average Power Limit* (RAPL), que se utiliza de sensores embutidos no hardware para medir o consumo elétrico do processador entre outros componentes. Os resultados são armazenados em registradores que serão acessados e disponibilizados para consulta pelo sistema operacional. Softwares de alto nível auxiliam no resgate desses resultados como o *PowerApi*, proposto por [Bourdon et al. 2013]. [Raffin and Trystram 2024] indicam que a vantagem desse medidor é sua precisão comparados a estimadores estatísticos, por seu acesso privilegiado aos componentes internos da máquina, além de estar menos suscetível a interferência, como os medidores externos.

Validando métodos eficientes para medição de energia, pode-se fazer estudos mais específicos sobre quais implementações dos desenvolvedores geram maior impacto no consumo. Por exemplo, [Ournani et al. 2021] investigam o impacto das refatorações de código no consumo de energia de softwares. Ao analisar a evolução do consumo energético de sete softwares open-source ao longo de cinco anos, os autores concluíram que as refatorações estruturais não impactam significativamente o consumo de energia. Este resultado é crucial para a medição do desempenho energético, pois indica que melhorias na estrutura do código podem ser implementadas sem aumentar o consumo energético, permitindo que desenvolvedores foquem em otimizações funcionais e computacionais para reduzir o consumo de energia.

3. Metodologia

A partir das técnicas levantadas pelos trabalhos relacionados, a estratégia desta pesquisa para avaliação energética dos softwares foi utilizar a leitura dos contadores embarcados no processador. Para definir situações para comparação, o método escolhido foi simular o mesmo comportamento de um usuário doméstico e mensurar o desempenho energético ao longo dos anos, com diferentes versões dos mesmos softwares.

3.1. Medidor

O medidor utilizado nesta pesquisa é o RAPL, tecnologia embarcada em processadores Intel modernos. Para consultar o seu valor registrado, foi utilizado o software *Scaphandre*, um facilitador que abstrai a necessidade da leitura em baixo nível¹. Tal ferramenta consegue distinguir a potência gasta por uma tarefa específica ao combinar a contagem de *jiffies* usados pela tarefa com as leituras de potência do sensor RAPL.

O *Scaphandre* lê os valores dos contadores de energia do processador, que registra o consumo total de energia da CPU e, em alguns casos, do controlador DRAM. A cada solicitação de medição, o sensor armazena esses valores junto com as estatísticas de uso da CPU e de cada processo em execução. Com esses dados, é possível calcular a proporção do tempo de CPU gasto por um determinado PID em relação ao tempo total de CPU ativo.

Essa proporção permite determinar a parcela do consumo de energia relacionada a um PID específico em um intervalo de tempo. Para serviços e programas que não operam em um único PID, o *Scaphandre* agrega o consumo de todos os PIDs relacionados,

¹*Scaphandre*. Disponível em: <https://hubblo-org.github.io/scaphandre-documentation/index.html>. Acesso em 24 jul. 2024.

Especificação	Capacidade
Memória	2 GB
Núcleos de processamento	2
Armazenamento	16 GB

Tabela 1. Especificações das máquinas virtuais

proporcionando uma visão precisa do consumo de energia por serviço. Nos experimentos, utilizamos apenas o consumo de potência utilizado pela máquina virtual.

3.2. Plataforma de experimentação

Os experimentos foram realizados em um computador com processador Intel Core i7-4790 de 3,6 GHz, TDP (*Thermal Design Power*) de 80 W e 16 GB de RAM (2 módulos de 8 GB DDR4, 2666 MT/s, *Part Number* BL8G26C16U4W.8FD). O sistema está equipado com um SSD NVMe de 1 TB e executa o sistema operacional Linux (distribuição Ubuntu 24.04 LTS), versão 6.8.0-38-generic.

3.3. Cenários avaliados

Neste experimento, foram conduzidas avaliações energéticas em diversas versões da distribuição Linux Ubuntu, abrangendo todos os lançamentos do tipo LTS (*Long Term Support*) desde 2008 até 2018. As versões selecionadas incluem Ubuntu 8.04 (2008), Ubuntu 10.04 (2010), Ubuntu 12.04 (2012), Ubuntu 14.04 (2014), Ubuntu 16.04 (2016) e Ubuntu 18.04 (2018). A escolha dessas versões deve-se ao fato de serem lançamentos do tipo LTS, conhecidos por oferecerem suporte estendido e maior estabilidade, características que garantem sua ampla utilização em ambientes de produção.

Para a implementação do experimento, as diferentes versões do Ubuntu foram instanciadas em máquinas virtuais configuradas com especificações de hardware idênticas, conforme detalhado na Tabela 1. Para a virtualização, foi utilizada a ferramenta QEMU–KVM² (*Quick Emulator — Kernel-based Virtual Machine*), versão 8.2.2.

3.4. Roteiro dos Experimentos

Os experimentos foram desenhados para avaliar o consumo de energia em situações cotidianas, refletindo o uso típico de um usuário da distribuição Ubuntu. Cada versão da distribuição foi utilizada com seu software e configurações padrões de fábrica. Os cenários avaliados envolveram a execução de atividades comuns, como o uso do navegador padrão *Mozilla Firefox*³, o envio de e-mails com o cliente de e-mails padrão *Mozilla Thunderbird*⁴ e a ilustração de imagens com o software GIMP⁵ (*GNU Image Manipulation Program*).

A primeira etapa do experimento envolveu a execução do navegador *Mozilla Firefox*, instalado por padrão em distribuições Ubuntu, e na realização de pesquisas por notícias. Esta tarefa simula uma atividade cotidiana comum para muitos usuários, onde

²*Quick Emulator — Kernel-based Virtual Machine*. Disponível em: <https://www.linux-kvm.org>.

³*Mozilla Firefox*. Disponível em: <https://www.mozilla.org/firefox/>.

⁴*Mozilla Thunderbird*. Disponível em: <https://www.thunderbird.net/>.

⁵*GNU Image Manipulation Program*. Disponível em: <https://www.gimp.org/>.

o navegador carrega páginas web com conteúdo textual e multimídia. Portanto, tal processo tem como objetivo avaliar o consumo energético associado a componentes como: a execução do navegador, a utilização da pilha de rede para fazer requisições HTTP e receber as respostas, e o consumo energético relacionado à renderização do conteúdo recebido da internet pelo navegador.

Na segunda etapa, ainda utilizando o navegador, realizou-se a pesquisa por imagens. Esta ação foi escolhida para aumentar a transferência de dados e exigir mais processamento gráfico e de rede. Ao carregar várias imagens, o sistema é submetido a uma carga maior, o que permite medir o consumo de energia durante operações que demandam maior uso de recursos de rede e gráficos.

A terceira etapa consistiu no envio de e-mails utilizando o cliente padrão da distribuição Ubuntu, o *Thunderbird*. Esta atividade reflete um cenário de uso comum, que envolve a interação com o software local e a comunicação com servidores externos, permitindo a medição do consumo energético associado a essas operações.

Por fim, a quarta etapa do experimento envolveu a abertura do software *GIMP* e a realização de uma ilustração. O *GIMP* é um editor de imagens de código livre (*open source*), sendo que a execução de tarefas gráficas nele requer recursos significativos de CPU e memória. Esta etapa foi projetada para avaliar o consumo energético durante atividades de manipulação gráfica, que são mais intensivas em termos de processamento.

Portanto, cada versão foi avaliada em relação a seguinte sequência de atividades:

1. Abrir o navegador padrão (*Firefox*), pesquisar pelo termo “notícias” no buscador padrão (*Google*) e abrir os dois primeiros resultados;
2. Ainda no navegador, pesquisar por imagens de “cachorros” e carregar todas as imagens das 10 primeiras páginas de resultados;
3. Enviar 5 e-mails pelo software padrão (*Thunderbird*) do Ubuntu;
4. Abrir o *GIMP*, fazer uma ilustração (um “cachorro”) e exportar a figura em extensão PNG.

Cada uma dessas etapas foi repetida 20 vezes para cada versão da distribuição escolhida. As atividades do experimento foram planejadas para cobrir uma variedade de cenários de uso cotidianos de um usuário doméstico, fornecendo uma visão mais abrangente do desempenho energético de cada versão do Ubuntu LTS para a realização de tarefas cotidianas.

4. Resultados

Os resultados apresentados nas figuras 1 e 2 ilustram a potência média demandada pela máquina virtual durante os 20 experimentos em diferentes versões do Ubuntu, abrangendo lançamentos de 2008 a 2018. Observa-se que, ao longo dos anos, há um aumento gradual na potência média consumida pelas versões mais recentes do sistema operacional.

Percebe-se no gráfico da Figura 2 que a versão 08.04 LTS (2008) apresenta a menor demanda energética, com uma potência média de aproximadamente 2,4 W. Com o lançamento das versões subsequentes, há um aumento notável na potência média consumida. A versão 10.04 LTS (2010) consome cerca de 2,8 W, seguida pela versão 12.04 LTS (2012), que apresenta uma leve redução para 2,7 W.

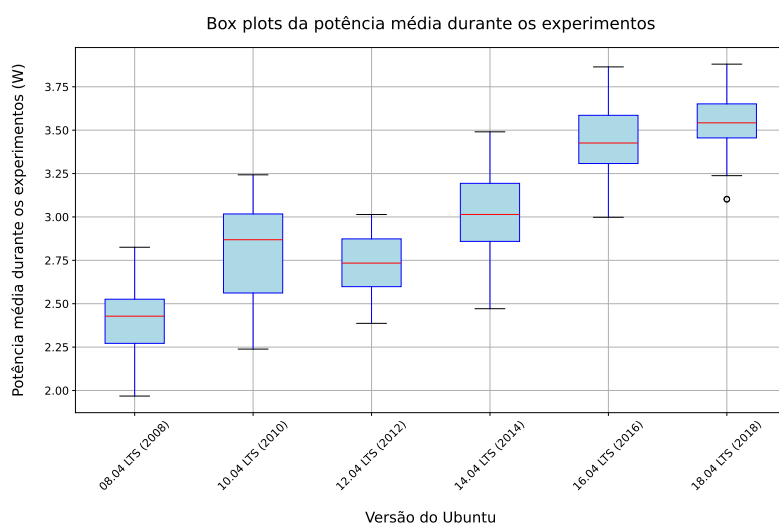


Figura 1. Distribuição da potência média demandada pela máquina virtual durante as 20 repetições do experimento para cada versão do Ubuntu.

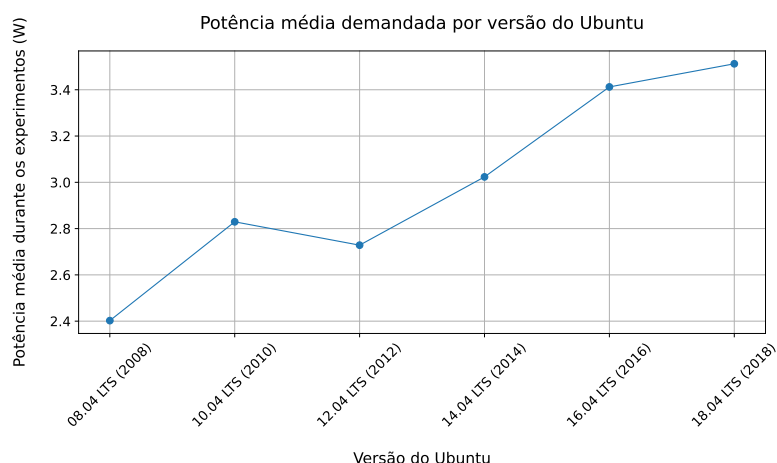


Figura 2. Potência média demandada pela máquina virtual durante as 20 repetições do experimento para cada versão do Ubuntu.

No entanto, a partir da versão 14.04 LTS (2014), observa-se um aumento constante na potência média demandada, atingindo cerca de 3,0 W. Este crescimento continua nas versões 16.04 LTS (2016) e 18.08 LTS (2018), com consumos médios de aproximadamente 3,3 W e 3,5 W, respectivamente.

Esses resultados sugerem que, apesar das melhorias e das novas funcionalidades introduzidas em cada versão do Ubuntu e dos softwares padrões (*Firefox*, *Thunderbird* e *GIMP*), há um aumento correspondente na demanda energética. Este fenômeno pode ser atribuído à maior complexidade, ao aumento do tamanho e recursos computacionais exigidos pelas versões mais recentes do sistema operacional. Na Figura 3 podemos analisar que as três ferramentas testadas apresentam um crescimento do espaço requerido para instalação ao longo do tempo, mesmo que com um aumento menos acentuado no caso do *GIMP*. Portanto, comparando as duas análises percebemos uma tendência de crescimento do consumo energético ligado ao aumento do tamanho do código binário e arquivos

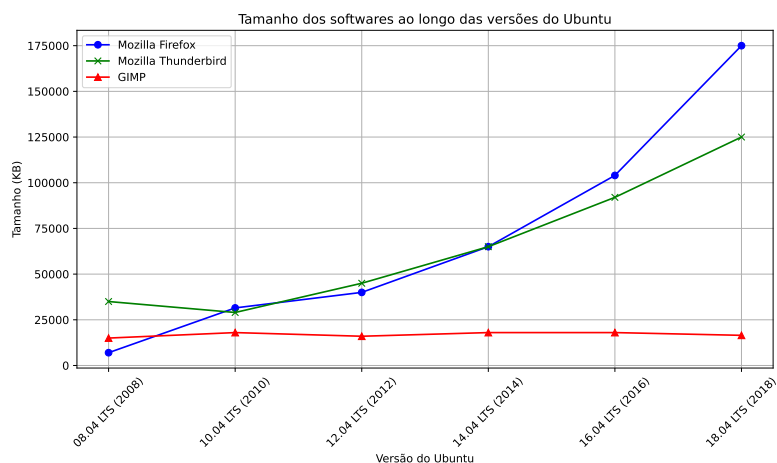


Figura 3. Tamanho dos softwares ao longo das versões do Ubuntu.

acessórios dos programas.

5. Limitações

A pesquisa realizada apresentou algumas limitações que devem ser discutidas. Primeiramente, a utilização de máquinas virtuais para simular diferentes versões do Ubuntu pode não refletir fielmente o consumo energético em hardware físico. A virtualização introduz uma camada adicional de abstração que pode influenciar a medição da energia consumida.

Outro ponto a ser considerado é a especificidade do hardware utilizado. O estudo foi realizado em um único modelo de processador Intel, o que pode limitar a generalização dos resultados para outros tipos de processadores ou arquiteturas, como AMD ou ARM. Diferentes modelos de CPU podem apresentar variações significativas no consumo energético, o que sugere a necessidade de experimentos adicionais em uma variedade mais ampla de hardware.

Adicionalmente, a rotina de uso simulada, embora planejada para representar um comportamento típico de usuário, pode não capturar todas as variações e complexidades do uso real. As atividades realizadas nos experimentos foram limitadas a tarefas específicas e podem não refletir a diversidade de operações executadas por diferentes tipos de usuários.

A análise temporal do consumo energético considerou um período de dez anos, mas o rápido avanço da tecnologia sugere que mais pesquisas são necessárias para entender plenamente as tendências e os fatores que influenciam o consumo energético dos softwares modernos. Estudos futuros devem considerar uma gama mais ampla de versões de software e hardware, bem como desenvolver metodologias que capturem um espectro mais amplo de comportamentos do usuário.

6. Conclusão

A avaliação do consumo energético de diferentes versões do Ubuntu revelou uma tendência de aumento gradual na potência média demandada ao longo dos anos. Com base nos resultados obtidos, é possível notar que, apesar das melhorias em termos de

desempenho e recursos, há uma necessidade crescente de otimizações voltadas para a eficiência energética.

A pesquisa também destaca a importância de considerar o consumo energético como uma métrica de desempenho ao desenvolver e escolher softwares. O aumento contínuo na demanda por energia reflete a necessidade de soluções mais eficientes e sustentáveis para minimizar o impacto ambiental associado ao uso dos softwares.

Além disso, as limitações do estudo indicam que mais pesquisas são necessárias para generalizar os resultados para diferentes tipos de hardware e arquiteturas. O uso de máquinas virtuais e um único modelo de processador limitam a aplicabilidade dos achados a outros contextos. Futuras pesquisas devem expandir a análise para uma variedade maior de plataformas e cenários de uso, a fim de proporcionar uma compreensão mais ampla das tendências de consumo energético em softwares.

A partir dos resultados desse estudo, trabalhos futuros poderão investigar as causas e pesquisar métodos para limitar o aumento no consumo de energia sem retroceder as funcionalidades das aplicações utilizadas cotidianamente pelos usuários.

7. Agradecimentos

Esta pesquisa foi financiada pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), projetos 2019/26702-8, 2021/06867-2, 2023/00811-0 e 2024/01115-0, e pelo Programa Unificado de Bolsas de Estudos para Apoio à Formação de Estudantes de Graduação da Universidade de São Paulo (PUB–USP).

Referências

- Bourdon, A., Noureddine, A., Rouvoy, R., and Seinturier, L. (2013). Powerapi: A software library to monitor the energy consumed at the process-level. *ERCIM News*, 2013(92).
- David, H., Gorbatov, E., Hanebutte, U. R., Khanna, R., and Le, C. (2010). RAPL: Memory power estimation and capping. In *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, pages 189–194.
- IEA (2024). Electricity 2024. Technical report, International Energy Agency, Paris, France. <https://www.iea.org/reports/electricity-2024>.
- Mathew, M. (2022). Nuclear energy: A pathway towards mitigation of global warming. *Progress in Nuclear Energy*, 143:104080.
- Ournani, Z., Rouvoy, R., Rust, P., and Penhoat, J. (2021). Tales from the Code #1: The Effective Impact of Code Refactorings on Software Energy Consumption. In *ICSOFT 2021 - 16th International Conference on Software Technologies*, Virtual, France.
- Pang, C., Hindle, A., Adams, B., and Hassan, A. E. (2016). What do programmers know about software energy consumption? *IEEE Software*, 33(3):83–89.
- Raffin, G. and Trystram, D. (2024). Dissecting the software-based measurement of cpu energy consumption: a comparative analysis. arXiv preprint arXiv:2401.15985.
- Sun, Q., Li, H., Ma, Z., Wang, C., Campillo, J., Zhang, Q., Wallin, F., and Guo, J. (2016). A comprehensive review of smart energy meters in intelligent energy networks. *IEEE Internet of Things Journal*, 3(4):464–479.