Analisando Técnicas de Gestão de Energia em Aplicações Aceleradas por GPU em um Sistema *Exascale*

Mariana T. Costa[1], **Antônio Tadeu A. Gomes**[2], **Philippe O. A. Navaux**[1]
Bronson Messer[3], **Arthur F. Lorenzon**[1]

[1]Instituto de Informática – UFRGS – Porto Alegre – RS – Brasil

[2]Laboratório Nacional de Computação Científica – Petrópolis – RJ – Brasil

[3]Oak Ridge National Laboratory – Oak Ridge – EUA

Resumo. *A gestão de energia em sistemas de computação de alto desempenho (HPC) baseados em GPUs é um dos maiores desafios da era Exascale, dada sua influência direta sobre custos operacionais e sustentabilidade ambiental. Entre as técnicas já suportadas pelas arquiteturas modernas, destacam-se o power capping, que limita dinamicamente a potência consumida, e o frequency capping, que impõe tetos estáticos de frequência. Apesar de ambos terem o mesmo objetivo, seus efeitos diferem significativamente conforme o perfil da aplicação, o que torna sua comparação essencial para orientar o uso em escala. Este trabalho apresenta uma avaliação de power capping e frequency capping em três aplicações científicas representativas, executadas no supercomputador Frontier com até 256 GPUs AMD MI250X. Foram exploradas 13 configurações de frequência e 9 limites de potência em cenários de execução em nó único e multinó. Os resultados revelam que: (i) em cargas limitadas por memória, frequency capping em níveis intermediários reduziu o consumo energético em até 25% com impacto inferior a 3% no tempo de execução; (ii) em workloads balanceados entre computação e comunicação, frequency capping manteve desempenho competitivo e superou o power capping em eficiência; e (iii) em cargas intensivas de computação, ambas as técnicas apresentaram ganhos energéticos modestos, mas acompanhados de penalidades significativas de desempenho.*

1. Introdução

À medida que os sistemas de computação de alto desempenho (HPC - *high-performance computing*) avançam em direção à era *exascale*, compreender a relação entre desempenho e consumo energético torna-se um requisito central para pesquisadores e projetistas de arquiteturas. Os supercomputadores atuais dependem fortemente do uso de Unidades de Processamento Gráfico (GPUs - *graphics processing units*) para sustentar altas taxas de operações em ponto flutuante. Contudo, esse ganho em capacidade de processamento costuma vir acompanhado de um aumento expressivo no gasto energético, além de oferecer ao usuário poucas alternativas de controle direto sobre a eficiência em tempo de execução. Nesse cenário, estratégias de avaliação de desempenho precisam ser combinadas a mecanismos de gestão de energia, de modo a equilibrar o tempo de execução e o custo energético da solução. Assim, analisar como diferentes técnicas de controle de hardware impactam a execução das aplicações configura-se como um desafio estratégico para o co-projeto entre software e hardware [Navaux et al. 2023].

Entre os mecanismos disponíveis em plataformas com GPU, destacam-se o *power capping* e o *frequency capping* [Komoda et al. 2013, Le Sueur and Heiser 2010]. O primeiro limita dinamicamente a potência consumida pelo dispositivo, ajustando voltagem e frequência de forma reativa a partir de políticas internas do hardware. Já o segundo impõe um teto estático à frequência de operação, garantindo previsibilidade, mas podendo restringir o desempenho. Embora ambos busquem reduzir o gasto energético ou manter a execução dentro de orçamentos de potência, seus efeitos práticos diferem, pois atuam por caminhos de controle distintos, podendo resultar em variações significativas de desempenho, eficiência e utilização do hardware.

O impacto de cada abordagem depende fortemente do perfil da aplicação. Programas com alta intensidade computacional tendem a ser mais sensíveis a reduções de frequência, enquanto *workloads* limitados por acesso à memória geralmente toleram reduções ("*throttling*") agressivas sem perdas expressivas de desempenho. Da mesma forma, aplicações com comportamento irregular e picos de atividade podem se beneficiar de mecanismos adaptativos de potência, ao passo que *workloads* mais estáveis podem apresentar melhor aproveitamento sob limites de frequência moderados. Essa diversidade de cenários reforça a necessidade de análises experimentais detalhadas em ambientes de HPC heterogêneos e em escala de produção.

Neste trabalho, apresentamos uma análise de desempenho e consumo energético aplicando técnicas de limitação de potência e de frequência em três aplicações científicas aceleradas por GPU: *Cholla*, *HACC*, e *LAMMPS*. Estas aplicações foram escolhidas pois são representativas em termos de (*i*) característica computacional (intensiva em memória e computação) e (*ii*) aplicações com mais horas de processamento agregado. Portanto, o principal objetivo é compreender como essas estratégias de limitação de frequência e potência afetam o tempo de execução, a energia total consumida e métricas de eficiência energética em diferentes perfis de aplicações e configurações de sistema. Foram avaliados 9 níveis de potência e 13 valores de frequência em diferentes escalas (1 e 32 nodos) no supercomputador Frontier, instalado no *Oak Ridge Leadership Computing Facility* (OLCF). A partir dessa investigação, destacamos os seguintes resultados:

- Em aplicações limitadas por largura de banda de memória (e.g., *Cholla*), limitar a frequência de operação em níveis intermediários mostra uma redução no consumo de energia de até 25% com impacto inferior a 3% no tempo de execução. Por outro lado, o *power capping* se mostrou menos eficiente em larga escala (e.g., 32 nodos), elevando o consumo de energia e tempo de execução.
- Para *workloads* mais equilibradas entre computação e comunicação, a modulação de frequência em faixas intermediárias proporciona economias próximas de 11% no consumo energético com degradação mínima no tempo de execução ($\approx 2-3\%$), superando os resultados obtidos via *power capping*, que não apresentou ganhos nesse cenário.
- Em aplicações intensivas de computação, ambas as técnicas produziram efeitos semelhantes: reduções pequenas de energia (e.g., 8%) acompanhadas de aumento expressivo no tempo de execução (e.g., 28%). Nesses casos, a configuração padrão é sempre a melhor escolha quando se quer equilibrar desempenho e energia.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os mecanismos de gestão de potência em GPU juntamente com os trabalhos relacionados.

A Seção 3 descreve a metodologia utilizada para a experimentação. Os resultados de desempenho e eficiência energética são discutidos na Seção 4, enquanto a Seção 5 reúne as conclusões finais.

2. Fundamentação Teórica

Nesta seção, apresentamos dois mecanismos de controle de energia em GPUs: *frequency capping* e *power capping*. Ambos são amplamente suportados pelas arquiteturas modernas, mas diferem quanto ao nível de controle, previsibilidade e impacto no desempenho.

2.1. Limitação de Frequência (*Frequency Capping*)

Também conhecida como *Dynamic Voltage and Frequency Scaling* (DVFS), essa técnica restringe as frequências de operação dos domínios de núcleo e memória da GPU, selecionando pares frequência–tensão a partir de estados de desempenho pré-definidos (*P-states*) [Le Sueur and Heiser 2010]. A redução da frequência diminui o consumo dinâmico e estático de energia, conforme descrito pela equação de potência $P_{dyn} = C \cdot V^2 \cdot f$, onde C é a capacitância de comutação, V é a tensão de alimentação e f é a frequência de operação. O controle pode ser feito via ferramentas como `nvidia-smi` ou `rocm-smi`, permitindo definir limites globais ou específicos por domínio. Como a transição entre estados ocorre em microssegundos, o DVFS pode ser usado de forma pró-ativa e em fases curtas de execução. Entretanto, em aplicações *compute-bound*, quedas abruptas de frequência podem degradar significativamente o desempenho.

2.2. Limitação de Potência (*Power Capping*)

O *power capping* estabelece um teto para o consumo de energia, expresso em watts, atuando em tempo de execução sem alterações no código da aplicação. Esse limite é monitorado pelo controlador de potência da GPU, que ajusta dinamicamente a frequência, tensão ou unidades ativas sempre que o consumo ultrapassa o valor configurado [Komoda et al. 2013]. Ao contrário do *frequency capping*, que impõe um valor fixo de frequência, o *power capping* trabalha com um envelope de potência, tornando-se mais flexível, mas menos determinístico. Em sistemas de HPC, essa técnica é comumente integrada a *job schedulers*, permitindo impor limites energéticos globais e reduzir a variabilidade de consumo entre aplicações.

2.3. Trabalhos Relacionados

Nos últimos anos, diferentes trabalhos em HPC acelerado por GPU investigaram estratégias de gerenciamento de energia, desde limites estáticos até mecanismos adaptativos orientados por feedback. Nesta subseção, revisamos alguns desses esforços e a evolução em direção a soluções mais inteligentes e de maior abrangência.

Um dos primeiros estudos em larga escala foi realizado por Tapasya et al., que executaram mais de 5.300 rodadas da metodologia MUMMI, a qual emprega o código de dinâmica molecular ddcMD [Tapasya et al. 2019]. Eles observaram que a redução do limite de potência de 300 W para 170 W não impactou significativamente o desempenho, enquanto a limitação de frequência gerou maior variabilidade. Já Allen et al. identificaram que os limites de potência padrão impostos pelo hardware resultavam em até 35% de perda de desempenho, sugerindo que políticas conscientes da aplicação poderiam

distribuir melhor a energia entre núcleos de processamento e subsistemas de memória [Allen et al. 2020].

Com base nessas limitações, surgiram propostas de frameworks dinâmicos. Kryzwaniak et al. apresentaram o DEPO, um sistema de execução que ajusta os limites de potência de GPUs da NVIDIA de acordo com o comportamento da aplicação, alcançando até 30% de economia energética com menos de 5% de sobrecarga [Krzywaniak et al. 2023]. Simmendinger et al. seguiram abordagem semelhante, com um mecanismo sensível às fases da execução, obtendo 20% de redução de consumo sem comprometer o tempo de execução [Simmendinger et al. 2024]. Karimi et al., por sua vez, analisaram três meses de *logs* operacionais no Frontier [Karimi et al. 2025] e mostraram que a decomposição dos modos de operação da GPU poderia gerar até 8,5% de economia em escala real, equivalente a mais de 1.400 MWh. Em aplicações específicas, Acun et al. verificaram que rodar o MILC no Perlmutter, um supercomputador instalado no NERSC, com um limite de 200 W (50% do TDP da A100) reduziu o consumo em 28% com perda de desempenho inferior a 15% [Acun et al. 2024]. Em nível arquitetural, Patrou et al. investigaram o *Locally Self-Consistent Multiple Scattering* (LSMS) no superchip GH200 da NVIDIA, aplicando otimização multiobjetivo para orientar ajustes dinâmicos de potência [Patrou et al. 2025].

Pesquisas mais recentes têm avançado para modelos inteligentes baseados em aprendizado de máquina. Yiming et al. propuseram o DRLCap, que utiliza aprendizado por reforço profundo para ajustar dinamicamente frequências em diferentes arquiteturas de GPU, alcançando 22% de economia de energia em GPUs NVIDIA e 10% em AMD [Yiming et al. 2024]. Outra vertente são abordagens cooperativas e centradas no usuário: Angelelli et al. sugeriram um "modo ecológico", no qual usuários optam voluntariamente por limites de potência. Simulações com rastros reais de um sistema Top500 (*Marconi100*) mostraram que, com adesão de cerca de 30%, reduzem-se falhas de *jobs* sob restrições rígidas de energia, mantendo a vazão do sistema [Angelelli et al. 2024].

Modelos preditivos também vêm ganhando destaque. Antici et al. aplicaram aprendizado de máquina para prever consumo em *jobs* de CPU no Fugaku, supercomputador instalado no RIKEN, atingindo 90% de acurácia [Antici et al. 2025]. Ding et al. desenvolveram um modelo unificado que combina variabilidade de potência, eficiência e métricas de desempenho para orientar decisões de *capping* em HPC [Ding et al. 2025]. Já no contexto de *workloads exascale* em GPU, Lorenzon et al. apresentaram o V-FORGE, um *framework* de escalonamento que considera frequência e variabilidade [Lorenzon et al. 2025]. Seus resultados em 400 GPUs AMD MI250X indicaram ganhos de até 41% em *energy-delay product* (EDP), evidenciando a importância da co-otimização hardware–software.

Contribuições deste trabalho. Diferente de propostas que introduzem novos *frameworks* ou modelos de aprendizado para gestão de energia em GPUs, nossa abordagem é de análise exploratória e com enfoque mais abrangente. O objetivo é fornecer orientações práticas a usuários e administradores de sistemas, caracterizando o comportamento das interfaces existentes de *power capping* e *frequency capping* em diversos cenários. Enquanto soluções como DEPO [Krzywaniak et al. 2023], DRLCap [Yiming et al. 2024] e V-FORGE [Lorenzon et al. 2025] priorizam adaptatividade em tempo de execução, este estudo complementa tais esforços com uma avaliação sistemática de diferentes níveis

de potência e configurações de frequência em três aplicações científicas representativas, contemplando *workloads* ligados a computação e memória. Diferente de análises focadas em aplicações específicas (como MILC na A100 [Acun et al. 2024] ou LSMS no GH200 [Patrou et al. 2025]), aqui consideramos múltiplas aplicações e escalas, incluindo execução em nodo único e multi-nodo no supercomputador Frontier.

3. Metodologia

Nesta seção, descrevemos as aplicações selecionadas e a infraestrutura experimental para avaliar desempenho e eficiência energética.

3.1. Aplicações Alvo

Foram escolhidas três aplicações científicas aceleradas por GPU, oriundas de diferentes áreas, de modo a cobrir padrões diversos de computação, intensidade aritmética e acesso à memória: *Cholla*, código hidrodinâmico multidimensional baseado no método parabólico por partes (PPM), voltado para simulações astrofísicas. É uma aplicação de alta demanda de memória e ponto flutuante em dupla precisão. Sua relevância decorre do fato de a astrofísica computacional ser um dos domínios que mais consome recursos em centros de HPC. *HACC–Hydro*, versão do *Hardware Accelerated Cosmology Code* dedicada à hidrodinâmica. Combina forte demanda de largura de banda com alta intensidade computacional [Habib et al. 2013]. Esse código é representativo por figurar entre as aplicações chave em estudos de cosmologia, e, por esse motivo, HACC é amplamente utilizado como *proxy* em avaliações de desempenho e escalabilidade em supercomputadores. *LAMMPS*, pacote de dinâmica molecular amplamente usado em ciência de materiais. Na versão GPU, descarrega cálculos de forças e construção de listas de vizinhança. Está entre os *workloads* mais executados em HPC, como no supercomputador brasileiro Santos Dumont, pela ampla base de usuários, flexibilidade e versões otimizadas para GPU.

3.2. Sistema de Testes

Os experimentos foram realizados no supercomputador Frontier (OLCF) [Atchley et al. 2023]. Cada nodo de computação possui um processador AMD EPYC 7763 de 64 núcleos (com 2 *threads* por núcleo) e 512 GB de memória DDR4. Cada nodo também contém quatro aceleradores AMD MI250X, cada um composto por dois *Graphics Compute Dies* (GCDs), resultando em oito GCDs por nodo, cada um com 64 GB de memória HBM3E. Para este estudo, cada GCD foi tratado como uma GPU independente, totalizando 8 GPUs por nodo. As GPUs MI250X possuem *themal design power* (TDP) de 560W e permitem variação de frequência entre 500 e 1700 MHz. Foram testadas 13 configurações de frequência (incrementos de 100 MHz) e 9 configurações de potência (200W a 560W, em passos de 40W). Todas as definições foram aplicadas estaticamente antes da execução via ferramentas ROCm. As aplicações foram compiladas com ROCm 6.2.4, utilizando `hipcc` e as opções `-O3` e `-offload-arch=gfx90a`. Os experimentos foram executados com os módulos `craype-accel-amd-gfx90a` e `rocm/6.2.4` disponíveis no Frontier.

Foram considerados dois cenários experimentais: (*i*) execução em nodo único, utilizando 8 GPUs, e (*ii*) execução em larga escala, distribuída em 32 nodos com um total de 256 GPUs. As aplicações foram configuradas com parâmetros de entrada adequados a cada escala, de forma a refletir práticas usuais de execução em supercomputadores.

No caso da aplicação *Cholla*, adotou-se o regime de *strong scaling*, em que o *workload* permanece fixo enquanto o número de recursos aumenta, permitindo avaliar eficiência paralela e *overheads* de comunicação. Para *HACC* e *LAMMPS*, optou-se pelo regime de *weak scaling*, no qual o *workload* cresce proporcionalmente ao número de recursos computacionais, refletindo cenários típicos de produção em HPC e permitindo analisar a capacidade de cada aplicação em manter desempenho com o aumento de escala.

Para analisar o desempenho, considerou-se o tempo de execução reportado pela própria aplicação. Já para o consumo de energia, o mesmo foi obtido através do *Omnistat*, uma ferramenta *open-source* e com baixo *overhead* de monitoramento [Omnistat 2025]. Cada combinação de aplicação, número de nodos e configuração de frequência e potência foi executada cinco vezes, resultando em desvios padrão inferiores a 0,5%. Esse número relativamente baixo de repetições é justificado por dois fatores principais: (*i*) a baixa variabilidade observada no sistema, que garante estabilidade estatística mesmo com poucas execuções; e (*ii*) o elevado custo computacional associado à realização de experimentos em larga escala, envolvendo centenas de GPUs em um supercomputador de produção.

4. Resultados

Nesta seção, discutimos os efeitos das técnicas de *power capping* e *frequency capping* sobre o desempenho e consumo de energia, considerando execuções em nodo único e com 32 nodos.

4.1. Análise de Desempenho e Consumo de Energia

As Figuras 1, 3 e 4 mostram os resultados de consumo de energia (eixo y) e tempo de execução (eixo x) para a execução das aplicações *Cholla*, *LAMMPS* e *HACC* com 1 e 32 nodos, respectivamente. Nas figuras, cada marcador representa uma configuração de *power capping* (em círculo), frequência de operação (em triângulo) e configuração padrão, em que, quanto mais próximo da origem (e.g., tempo de execução e/ou energia igual a 0), melhor é o resultado atingido. Conforme observado, as três aplicações possuem comportamentos diferentes em termos de desempenho e consumo de energia quando configurações de gerenciamento de potência e frequência são aplicadas. Deste modo, cada aplicação é discutida em separado a seguir.

Ao considerar os resultados da aplicação *Cholla* (Figura 1) em nodo único (8 GPUs), tanto a frequência quanto o *power capping* afetam consumo e tempo. Para *power capping*, o menor consumo ocorre em *320 W* (52,3 kJ), o que representa –21,9% frente ao *Default* (66,9 kJ), com sobrecusto de tempo de $\approx 8\%$ (33,73s vs 31,8s). No ajuste de frequência, a faixa 0,9–1,0 GHz apresenta os melhores consumos (49,686–50,632 kJ), equivalendo a –25,8% a –24,3% em relação ao *Default*, com penalidades de tempo pequenas: 2,4% (31,878 s) em 0,9 GHz e 0,9% (31,423 s) em 1,0 GHz, indicando que o modo padrão é energeticamente superdimensionado e não otimiza desempenho nem eficiência.

Na execução em larga escala (32 nodos, 256 GPUs), os tempos com *power capping* ficam em 4,85–5,10 s, enquanto o *Default* é 4,03 s; portanto, há 20–27% de aumento de tempo. No consumo, todas as potências testadas (*200–520 W*) ficam acima do *Default* (138,5 kJ): o melhor caso de potência, *200 W*, consome 146,3 kJ ($\approx$+5,6%), e os demais variam de +11,8% (*400 W*) a +15,4% (*360 W*). Assim, em 32 nodos, *power capping* degrada desempenho e eficiência energética em relação ao modo de operação padrão. Já a

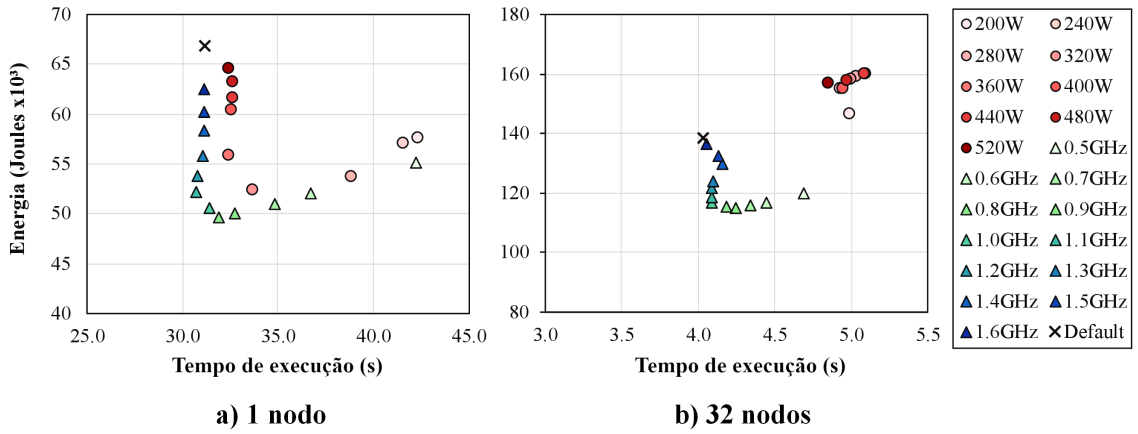


Figura 1. Resultados Cholla *Strong Scaling*.

redução de frequência é eficaz: em 0,9 GHz o consumo cai para 115,2 kJ (≈17%) com sobrecusto de apenas 3.7% no tempo de execução.

Comparando os dois cenários, observa-se que a aplicação apresenta boa escalabilidade, mantendo tempos de execução consistentes quando distribuída em múltiplos nodos. Entretanto, o impacto das técnicas de gerenciamento difere. Em nodo único, tanto a redução de frequência quanto o *power capping* permitem ganhos de eficiência energética relevantes, ainda que com diferentes custos de desempenho. Já em larga escala, o comportamento muda: enquanto o ajuste de frequência continua oferecendo economia de energia com impacto moderado no tempo, o *power capping* deixa de ser vantajoso, elevando o consumo e aumentando o tempo de execução.

A Figura 2 ilustra esse efeito por meio de *traces* de frequência das GPUs (média para cada GPU ID de todos os nodos – gráficos superiores) e potência (média para cada GCD de todos os nodos – gráficos inferiores) para 32 nodos. Com a frequência limitada em 980 MHz, as curvas permanecem praticamente constantes ao longo do tempo, indicando que frequência e potência operam de forma previsível e uniforme entre os nodos. Já com o *power capping* fixado em *200 W* (Figura 2b), a frequência apresenta oscilações frequentes durante a execução do *workload* na GPU (a partir de ≈ 320s), onde o *clock* sobe e desce para se ajustar ao limite de potência, provocando variação no consumo de potência. Essa instabilidade introduz desbalanceamento entre os nodos, já que diferentes processos podem executar em ritmos distintos, aumentando o tempo global de execução. Portanto, como esta é uma aplicação com características de acesso intensivo à memória, não há benefício em picos transitórios de frequência uma vez que grande parte do tempo é gasta aguardando dados. Nesse sentido, *frequency capping* favorece o consumo de energia mais eficiente e constante, enquanto o *power capping* induz variações que acabam penalizando a escalabilidade.

Na execução de *LAMMPS* em nodo único (Figura 3), observa-se que a aplicação já opera muito próxima do ponto ótimo no modo *Default*. Embora em alguns níveis o *power capping* tenha levado a redução no consumo de energia, essa vantagem não foi acompanhada por aumento no tempo de execução, resultando em eficiência semelhante ao padrão. O ajuste de frequência mostrou-se ainda menos eficaz nesse cenário, pois qualquer redução no nível de frequência levou a aumentos expressivos no tempo, por

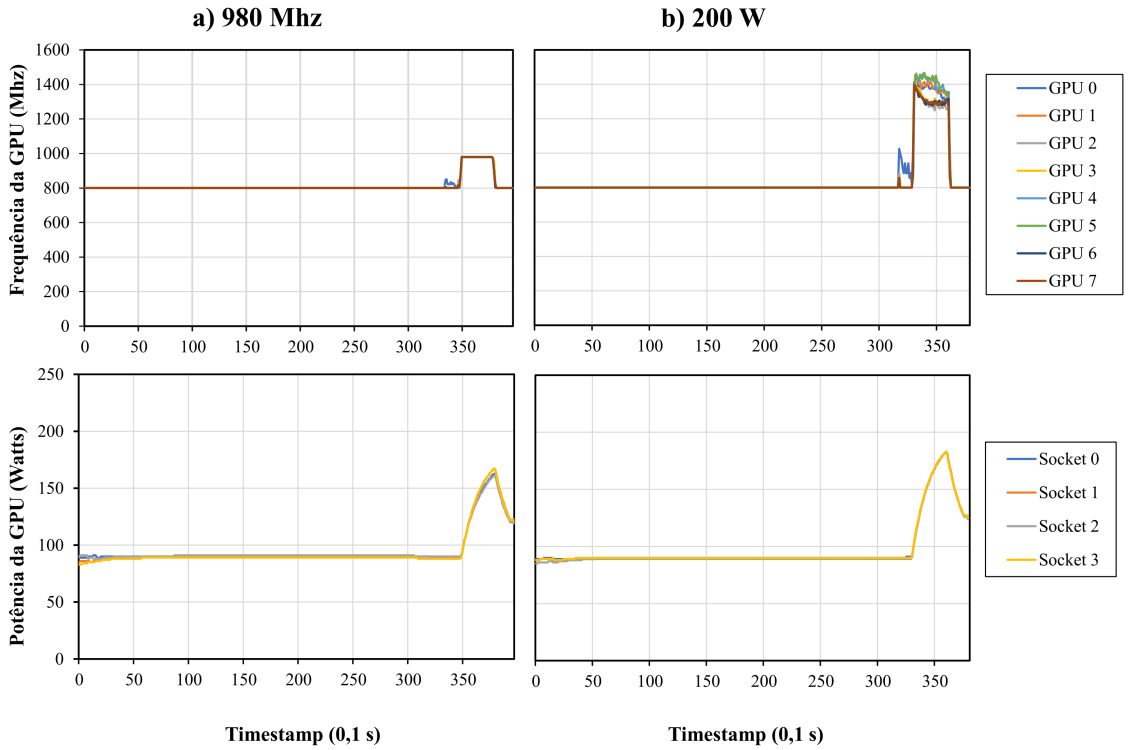
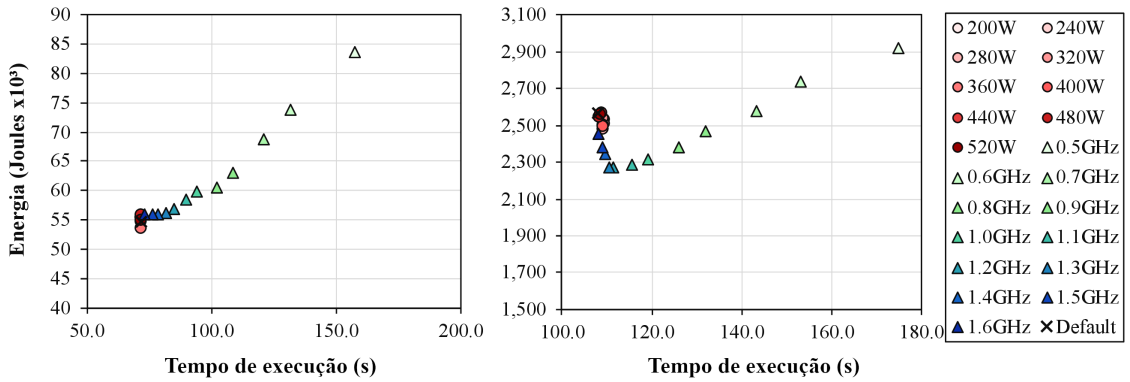


Figura 2. Comportamento de frequência de operação (média entre os GPUs IDs dos 32 nodos) e potência (média para os 4 GCDs dos 32 nodos) para a execução do Cholla em 32 Nodos.

exemplo, acima de 30% já em 0,9 GHz, sem ganhos energéticos proporcionais. Dessa forma, em um único nodo, tanto o limite de frequência quanto de potência não superam o desempenho e eficiência energética alcançado pela configuração padrão.

Em larga escala (32 nodos), o comportamento é distinto. O *power capping* novamente não se mostrou vantajoso, já que as pequenas reduções de energia observadas foram anuladas pelo aumento no tempo de execução. Em contrapartida, a modulação de frequência apresentou resultados mais favoráveis. Configurações intermediárias, entre 1,2 e 1,3 GHz, reduziram o consumo de energia em aproximadamente 11% em relação ao *Default*, com acréscimo de tempo de apenas 2-3%, indicando que, nesse regime, a aplicação não é totalmente limitada por frequência, e há espaço para explorar o equilíbrio entre tempo e consumo sem comprometer a escalabilidade global.

Comparando os dois cenários, observa-se que *LAMMPS* apresenta respostas distintas as duas técnicas. Em nodo único, a aplicação é fortemente dependente do desempenho de pico da GPU, e a redução de frequência provoca aumentos expressivos no tempo de execução, anulando eventuais economias energéticas. O *power capping*, por sua vez, praticamente não altera o comportamento: tanto o tempo quanto o consumo de energia permanecem próximos ao *Default*. Em múltiplos nodos, esse contraste fica ainda mais evidente. Enquanto o *power capping* continua sem trazer benefícios práticos, o *frequency capping* em níveis intermediários reduz o consumo energético de maneira consistente, com degradação de desempenho moderada graças à presença de fases de comunicação e sincronização.



a) 1 Nodo **b) 32 Nodos**

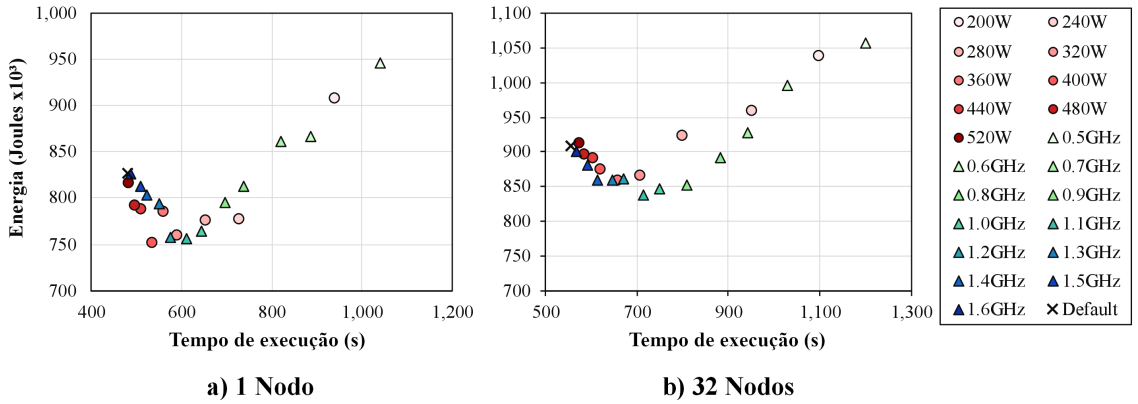
Figura 3. Resultados LAMMPS *Weak Scaling*.

Ao considerar os resultados obtidos para a aplicação *HACC* (Figura 4), na execução em 1 nodo, *power capping* e *frequency capping* traçam curvas tempo–energia muito semelhantes. O melhor ponto de limite de potência para energia foi *400 W*, o que representa uma economia de 9,2% frente ao *Default*, porém com acréscimo de 11,4% no tempo de execução (535,66 s vs 480,92 s). Pelo lado de frequência, o menor consumo ocorreu em 1,1 GHz (–8,5%), com uma penalidade de tempo maior (612 s; +27,3%). Em ambos os ajustes, portanto, a economia de $\approx$ 8-9% vem acompanhada de aumentos consideráveis no tempo e o *Default* permanece o ponto de melhor desempenho e competitivo em energia. Em 32 nodos, o padrão se repete. O melhor *power cap* em energia foi *360 W* (–5,5%), com +18,6% de tempo. No DVFS, o menor consumo ficou em 1,1 GHz (–7,7%), à custa de +28,3% no tempo. Ou seja, em escala, frequência reduzida poupa um pouco mais de energia, mas custa mais tempo. Já *power capping* economiza menos energia, porém com menor penalidade temporal. Em nenhum dos casos há melhora clara sobre o *Default* em termos de *trade-off* entre desempenho e energia.

Esse comportamento espelhado entre as duas técnicas decorre da natureza da aplicação. Isto é, aplicações com longas fases de computação intensiva e acesso à memória, limitar potência quanto reduzir frequência diminuem o *throughput* de forma aproximadamente proporcional. Como o consumo total de energia tende a acompanhar o tempo (Energia $\approx$ Potência média $\times$ Tempo) e a potência média também cai com a frequência/tensão, as duas estratégias acabam atingindo o mesmo comportamento. Assim, a ausência de grandes trechos ociosos ou limitados por comunicação pura (onde DVFS costuma ter maior impacto) e a forte dependência de FLOPs por largura de banda de memória explicam por que, no HACC, *power capping* e *frequency capping* produzem resultados similares e raramente superam o *Default* em termos de eficiência global.

4.2. Discussão

Os resultados discutidos na subseção anterior mostram que não existe uma configuração única de *power capping* ou *frequency capping* que funcione bem para todos os tipos de aplicações. O efeito dessas técnicas depende diretamente do perfil de cada programa. Aplicações que gastam muito tempo acessando memória se beneficiam de frequências intermediárias; programas que equilibram computação e comunicação podem ter ganhos moderados; já aplicações que usam intensamente a capacidade de cálculo da GPU, em



a) 1 Nodo **b) 32 Nodos**

Figura 4. Resultados HACC *Weak Scaling*.

geral, não apresentam melhorias relevantes em relação ao modo padrão. Isso significa que o ajuste de energia precisa levar em conta as características do programa antes de ser aplicado de forma generalizada.

Do ponto de vista do gerenciamento de sistemas, o ajuste de frequência resulta em um comportamento mais uniforme, mantendo os nodos do sistema sincronizados. Já a limitação de potência gera oscilações na frequência e no consumo de energia, o que pode causar diferenças de tempo entre nodos em execuções distribuídas. Em cenários de larga escala, essa instabilidade pode aumentar o tempo total da aplicação, mesmo que haja economia de energia em cada GPU. Deste modo, aplicar apenas *power capping* pode ser prejudicial em sistemas com muitos usuários e aplicações executando de forma concorrente (e.g., nuvem computacional).

Outro aspecto é que essas duas técnicas podem ser vistas como complementares. O *power capping* pode ser útil para impor limites de consumo em todo o sistema, garantindo que o uso total de energia não ultrapasse um certo valor. O *frequency capping*, por sua vez, é mais indicado para ajustar o consumo em aplicações específicas, especialmente quando há interesse em reduzir energia sem aumentar muito o tempo de execução. Uma abordagem combinada pode atender tanto às necessidades dos administradores, que precisam controlar o uso global de energia, quanto dos usuários, que desejam manter bom desempenho em suas aplicações.

5. Conclusão

Na era *Exascale*, a gestão de energia tornou-se um desafio central para equilibrar custos operacionais, sustentabilidade e desempenho em sistemas HPC acelerados por GPU. Entre os mecanismos disponíveis, *power capping* e *frequency capping* são amplamente suportados pelas arquiteturas modernas, mas apresentam impactos distintos dependendo do perfil da aplicação. Deste modo, este trabalho apresentou uma avaliação destas técnicas de gestão de energia em um supercomputador *exascale* de produção, o Frontier, equipado com GPUs AMD MI250X. A análise foi conduzida em aplicações científicas amplamente utilizadas neste sistema, abrangendo diferentes perfis de execução: aplicações limitadas por memória, balanceadas entre comunicação e computação e intensivas em capacidade de cálculo.

Através da execução destas aplicações com diferentes configurações de *power*

capping e *frequency capping* e número de nodos (até 256 GPUs), foi mostrado que, em aplicações limitadas por memória, o *frequency capping* em níveis intermediários reduziu o consumo de energia em até 25% com impacto mínimo em desempenho. Já em aplicações balanceadas, a modulação intermediária de frequência reduziu em 11% com aumento de apenas 3% no tempo, superando o *power capping*. Por fim, aplicações intensivas de computação, ambas as técnicas entregaram resultados similares. Como trabalhos futuros, pretende-se ampliar a análise para explorar estratégias híbridas que combinem *power capping* e *frequency capping* de forma coordenada. Além disso, pretende-se incluir modelos preditivos baseados em aprendizado de máquina, para identificar automaticamente o perfil da aplicação e recomendar a configuração energética mais adequada em tempo de execução.

Agradecimentos

Este trabalho foi parcialmente apoiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior- Brasil (CAPES)- Código de Financiamento 001, FAPERGS - PqG 24/2551-0001388-1, e CNPq. *This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.*

Referências

Acun, F., Zhao, Z., Austin, B., Coskun, A. K., and Wright, N. J. (2024). Analysis of power consumption and gpu power capping for milc. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1856–1861.

Allen, T., Feng, X., and Ge, R. (2020). Performance optimization in power-capped gpu computing. SC20. Poster, evaluating miniFE on Titan XP, showing up to 35 % performance loss under default GPU power capping and proposing application-aware SM/memory power allocation.

Angelelli, L., Carastan-Santos, D., and Dutot, P.-F. (2024). Run your hpc jobs innbsp;eco-mode: Revealing thenbsp;potential ofnbsp;user-assisted power capping innbsp;supercomputing systems. In *Job Scheduling Strategies for Parallel Processing: 27th International Workshop, JSSPP 2024, San Francisco, CA, USA, May 31, 2024, Revised Selected Papers*, page 181–196, Berlin, Heidelberg. Springer-Verlag.

Antici, F., Borghesi, A., Domke, J., and Kiziltan, Z. (2025). Uopc: A user-based online framework to predict job power consumption in hpc systems. In *ISC High Performance 2025 Research Paper Proceedings (40th International Conference)*, pages 1–12.

Atchley, S., Zimmer, C., Lange, J., Bernholdt, D., Melesse Vergara, V., Beck, T., Brim, M., Budiardja, R., Chandrasekaran, S., Eisenbach, M., Evans, T., Ezell, M., Frontiere, N., Georgiadou, A., Glenski, J., Grete, P., Hamilton, S., Holmen, J., Huebl, A., Jacobson, D., Joubert, W., Mcmahon, K., Merzari, E., Moore, S., Myers, A., Nichols, S., Oral, S., Papatheodore, T., Perez, D., Rogers, D. M., Schneider, E., Vay, J.-L., and Yeung, P. K. (2023). Frontier: Exploring exascale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23, New York, NY, USA. Association for Computing Machinery.

Ding, N., Antepara, O., Zhao, Z., Austin, B., Oliker, L., Wright, N. J., and Williams, S. (2025). Maximizing power-constrained supercomputing throughput. In *ISC High Performance 2025 Research Paper Proceedings (40th International Conference)*, pages 1–13.

Habib, S., Morozov, V., Frontiere, N., Finkel, H., Pope, A., and Heitmann, K. (2013). Hacc: Extreme scaling and performance across diverse architectures. In *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–10.

Karimi, A. M., Maiterth, M., Shin, W., Sattar, N. S., Lu, H., and Wang, F. (2025). Exploring the frontiers of energy efficiency using power management at system scale. In *SC-W*, SC-W '24, page 1835–1844. IEEE Press.

Komoda, T., Hayashi, S., Nakada, T., Miwa, S., and Nakamura, H. (2013). Power capping of cpu-gpu heterogeneous systems through coordinating dvfs and task mapping. In *2013 IEEE 31st International Conference on Computer Design (ICCD)*, pages 349–356.

Krzywaniak, A., Czarnul, P., and Proficz, J. (2023). Dynamic gpu power capping with online performance tracing for energy efficient gpu computing using depo tool. *Future Generation Computer Systems*, 145.

Le Sueur, E. and Heiser, G. (2010). Dynamic voltage and frequency scaling: The laws of diminishing returns. In *Proceedings of the 2010 international conference on Power aware computing and systems*, pages 1–8.

Lorenzon, A. F., Beck, A. C. S., Navaux, P. O. A., and Messer, B. (2025). Energy-efficient gpu allocation and frequency management in exascale computing systems. In *ISC High Performance 2025 Research Paper Proceedings (40th International Conference)*, pages 1–11.

Navaux, P. O. A., Lorenzon, A. F., and Serpa, M. D. S. (2023). Challenges in High-Performance Computing. *Journal of the Brazilian Computer Society*, 29(1):51–62.

Omnistat (2025). Omnistat: Scale-out cluster telemetry. Accessed: 2025-07-20.

Patrou, M., Wang, T., Elwasif, W., Eisenbach, M., Miller, R., Godoy, W., and Hernandez, O. (2025). Power-capping metric evaluation for improving energy efficiency in hpc applications.

Simmendinger, C., Marquardt, M., Mäder, J., and Schneider, R. (2024). Powersched - managing power consumption in overprovisioned systems. In *2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops)*, pages 1–8.

Tapasya, P., Frye, Z., Bhatia, H., Natale, F., Glosli, J., Ingólfsson, H., and Rountree, B. (2019). Comparing gpu power and frequency capping: A case study with the mummi workflow. pages 31–39.

Yiming, W., Hao, M., He, H., Zhang, W., Tang, Q., Sun, X., and Wang, Z. (2024). Drlcap: Runtime gpu frequency capping with deep reinforcement learning. *IEEE Transactions on Sustainable Computing*, PP:1–15.