

Tópicos de Desenvolvimento de Software

Powercap e Codecarbon

28.03.2025

Hugo Gomes, Lara Ferreira & Luís Ferreira

1 Powercap

Powercap



O *Powercap* é uma tecnologia que combina a eficiência e a gestão de energia de modo a limitar o consumo e tornar programas mais energeticamente sustentáveis. Ter um *Powercap* demasiado baixo teremos um consumo reduzido mas também haverá um enorme prejuízo à velocidade do programa.

Assim sendo, é importante atingir um valor de *Powercap* que consiga um melhor consumo mas sem afetar em demasia a velocidade do programa.

Valor de Powercap utilizado



Para este trabalho fomos testando diferentes valores para o *Powercap* até termos o valor que nos dá o melhor balanço entre *GreenUp* e *SpeedUp*.

Esse valor foi 13, e nos próximos slides mostraremos as diferenças na utilização ou não do *Powercap*. Para as duas funções criadas no trabalho anterior mostraremos a diferença no *SpeedUp* e no *GreenUp*.

O valor previamente referido é obtido através de um ficheiro *powercap.py* que calcula o resultado mais eficiente consoante os valores obtidos no ficheiro *measurements_Powercap.csv*.

De seguida, pegamos nos valores recolhidos com o *powercap* 13 e -1, ou seja sem qualquer restrição, são comparados, de forma a podermos criar vários gráficos *GPS-UP*.

Medição dos valores - Fibonacci Linear



Podemos verificar que o *Powercap* aumentou o *Speedup* mas diminuiu o *Greenup*, o que significa que o programa executou mais rapidamente mas teve um maior consumo.

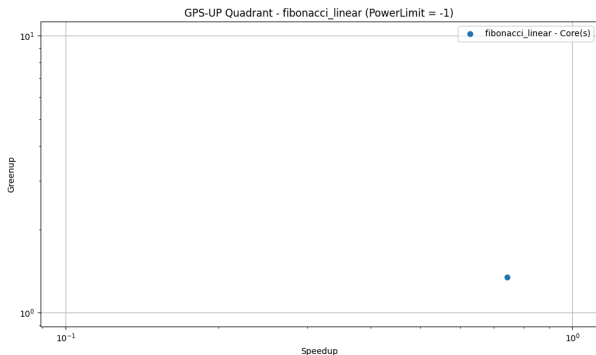


Figure 1: Energia (Core(s)), sem Powercap

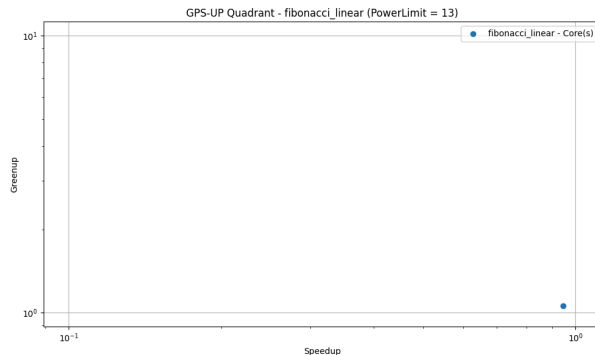


Figure 2: Energia (Core(s)), com Powercap

Medição dos valores - Fibonacci Linear



Aqui podemos ver que em termos de energia consumida, o *Powercap* tornou o programa bastante mais poupado, no entanto também baixou bastante o desempenho do programa.

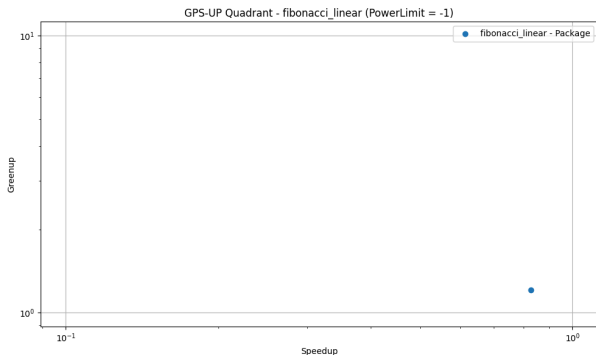


Figure 3: Energia (Package), sem Powercap

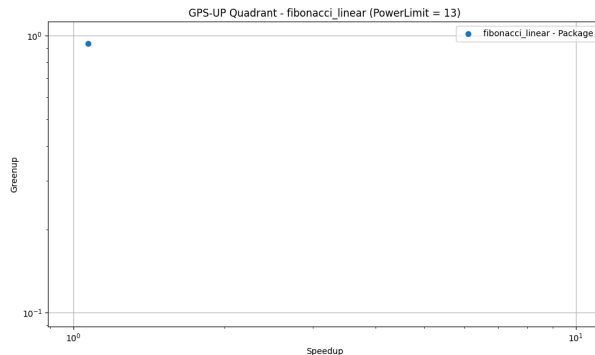


Figure 4: Energia (Package), com Powercap

Medição dos valores - Fibonacci Linear



Uma vez que aqui a unidade comparada é o tempo, seria de esperar que com a utilização do *Powercap* o *Greenup* aumentaria, no entanto aconteceu o oposto.

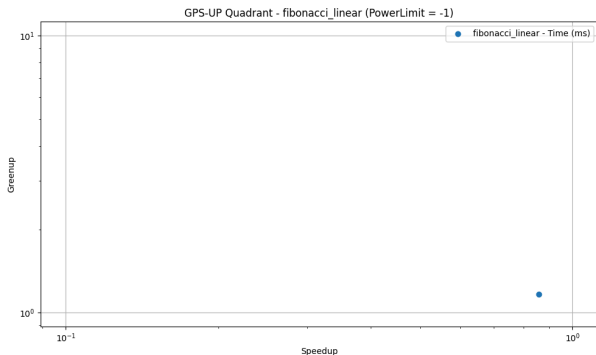


Figure 5: Tempo de execução, sem Powercap

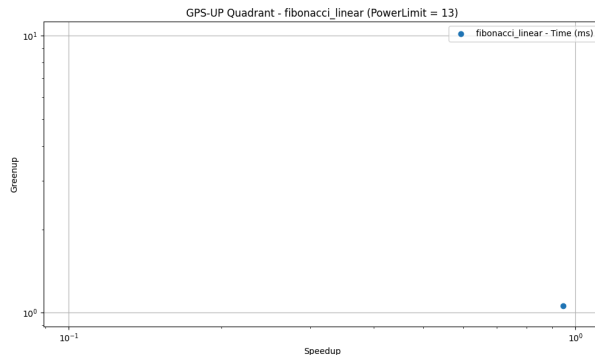


Figure 6: Tempo de execução, com Powercap

Medição dos valores - Fibonacci Recursivo



Na função recursiva o uso do *Powercap* pouco afetou a função recursiva a nível do processador.

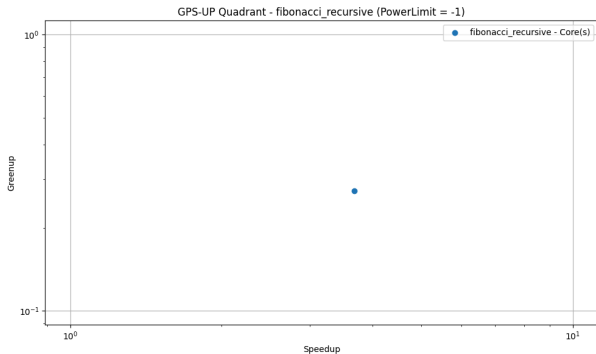


Figure 7: Energia (Core(s)), sem Powercap

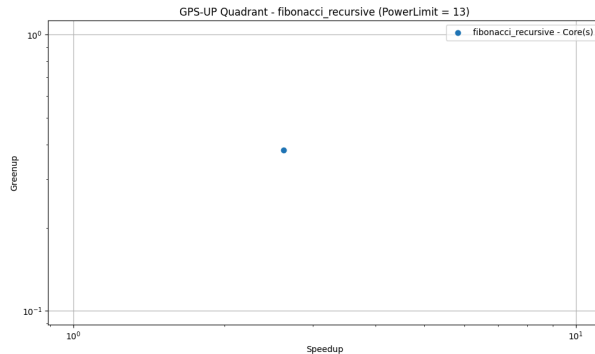


Figure 8: Energia (Core(s)), com Powercap

Medição de valores - Fibonacci Recursivo



Mais uma vez, apesar de haver uma ligeira melhoria no desempenho, é uma melhoria bastante pequena.

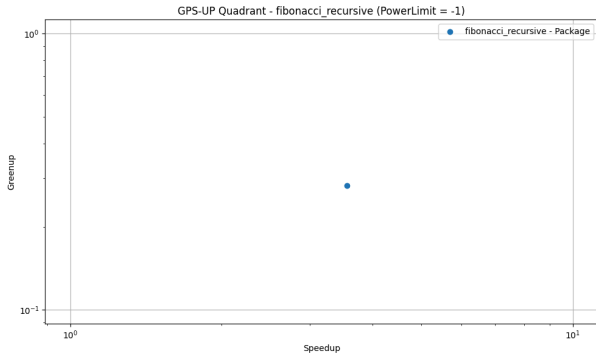


Figure 9: Energia (Package), sem Powercap

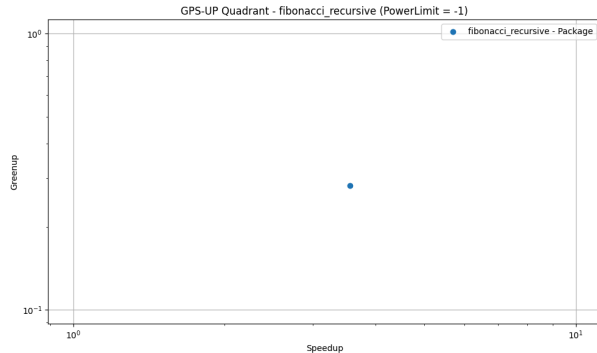


Figure 10: Energia (Package), com Powercap

Medição de valores - Fibonacci Recursivo



Novamente, o padrão mantém-se com a execução recursiva, ligeira melhoria, porém desta vez no *Greenup*.

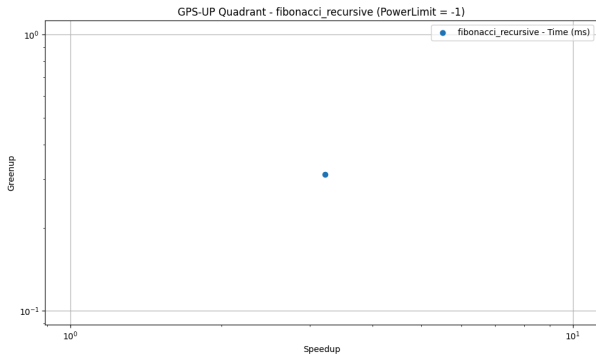


Figure 11: Tempo de execução, sem Powercap

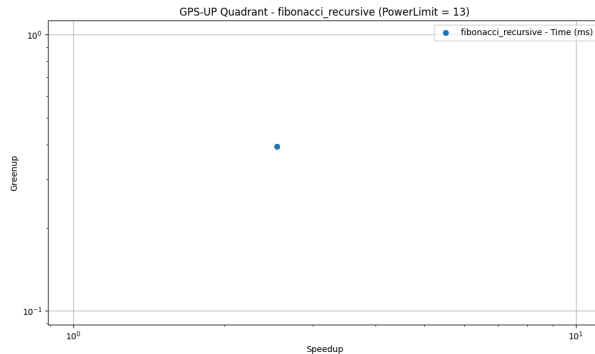


Figure 12: Tempo de execução, com Powercap

Consumo de energia e tempo de execução



A seguir, estão dois gráficos que representam a comparação de valores médios de tempo de execução de ambas as implementações de Fibonacci, para os três números de teste (10, 30 e 40). É fácil de visualizar que utilização de otimizações é extremamente importante para poupança de recursos, como neste caso: tempo e energia.

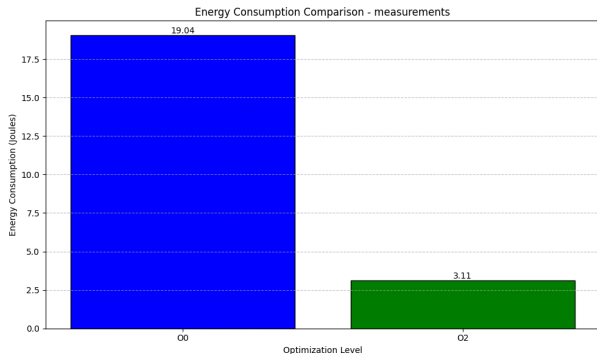


Figure 13: Consumo de Energia

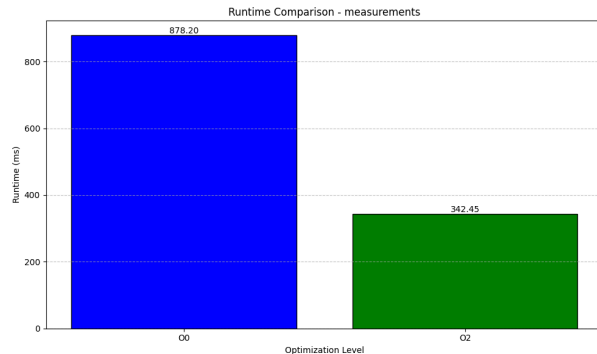


Figure 14: Tempo de execução

Powercap



Podemos concluir que o Powercap apresenta melhorias significativas no programa sequencial, não sendo tão relevante quando o programa é recursivo. Isto pode-se dever ao facto de que o programa recursivo não tem muito por onde ser melhorado e reduzir o seu consumo. Como o Powercap reduz o consumo, torna-se necessária a otimização da energia gasta.

2 CodeCarbon

O *CodeCarbon* é uma ferramenta inovadora que permite monitorizar e reduzir a pegada de carbono associada à execução de projetos computacionais. Ao medir o consumo energético dos processos e estimar as emissões de CO₂, o *CodeCarbon* posiciona-se como uma solução útil para desenvolvedores e investigadores preocupados com a sustentabilidade digital.

No entanto, para que esta ferramenta seja realmente eficaz, é essencial avaliar a sua precisão em comparação com métodos estabelecidos. Um dos padrões de referência na medição do consumo energético é o *RAPL* (*Running Average Power Limit*), uma tecnologia integrada nos processadores Intel que fornece medições detalhadas de consumo de energia, como já estudado previamente.

Com este trabalho, propomos investigar até que ponto os resultados obtidos pelo *CodeCarbon* são fiéis às medições registadas pelo *RAPL*. Para isso, analisamos diferentes implementações do cálculo da sequência de Fibonacci, comparando tempos de execução, energia e impactos das optimizações compiladas em diferentes níveis.

Comparação com o RAPL - Energy



Na figura 15, não há diferença significativa entre as medições de *RAPL* e *CodeCarbon* em ambos os níveis de otimização O0 e O2, com valores praticamente idênticos para ambas as ferramentas. Na figura 16, a ferramenta *CodeCarbon* apresenta valores ligeiramente superiores aos de *RAPL* em ambos os níveis de otimização, onde os valores são superiores devido à complexidade da função recursiva.

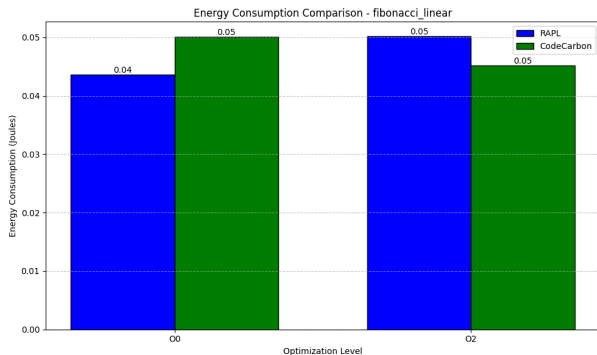


Figure 15: Fibonacci Linear

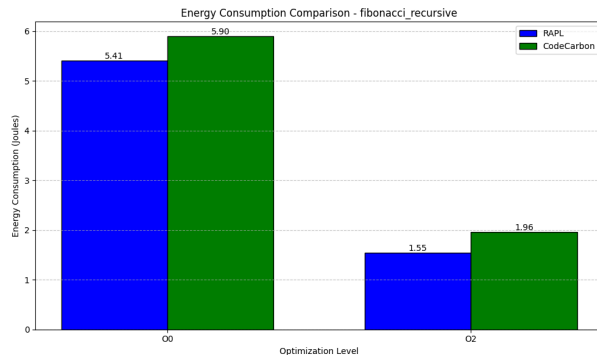


Figure 16: Fibonacci Recursiva

Comparação com o RAPL - Runtime



Na figura 17, a diferença entre os níveis de otimização é mínima, o CodeCarbon apresenta tempos ligeiramente menores. Na figura 18, a ferramenta CodeCarbon apresenta tempos de execução consistentemente menores do que RAPL, o que revela para valores superiores, o CodeCarbon é menos preciso a medir o tempo de execução.

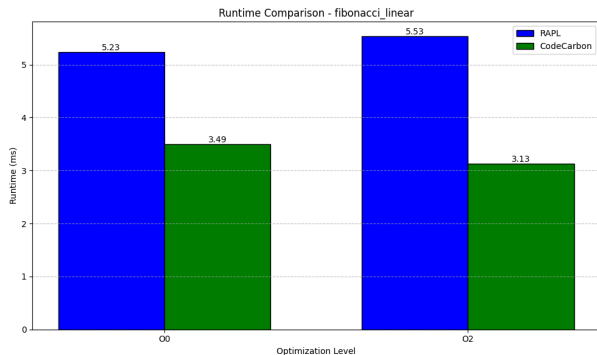


Figure 17: Fibonacci Linear

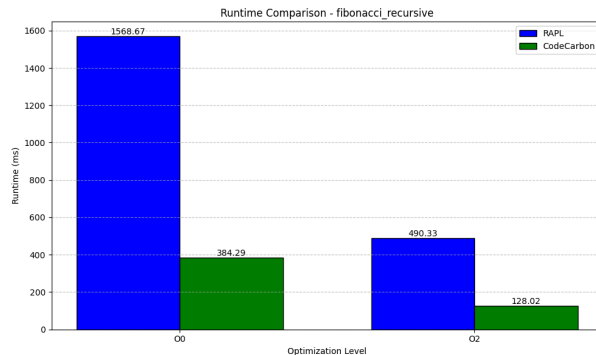


Figure 18: Fibonacci Recursiva

Conclusão



Segundo os nossos testes, o CodeCarbon e RAPL não são idênticos em medições, especialmente para cargas mais intensivas como Fibonacci recursivo, onde a diferença é expressiva.

Para cargas leves e iterativas, CodeCarbon apresenta valores mais similares com RAPL.

Caso o objetivo for medir consumo de energia e tempo com alta precisão, especialmente em workloads complexos, RAPL será uma escolha mais confiável.

Por sua vez, o CodeCarbon pode ser útil para estimativas rápidas, mas deve ser validado com ferramentas mais precisas, como RAPL, caso medições rigorosas sejam necessárias.