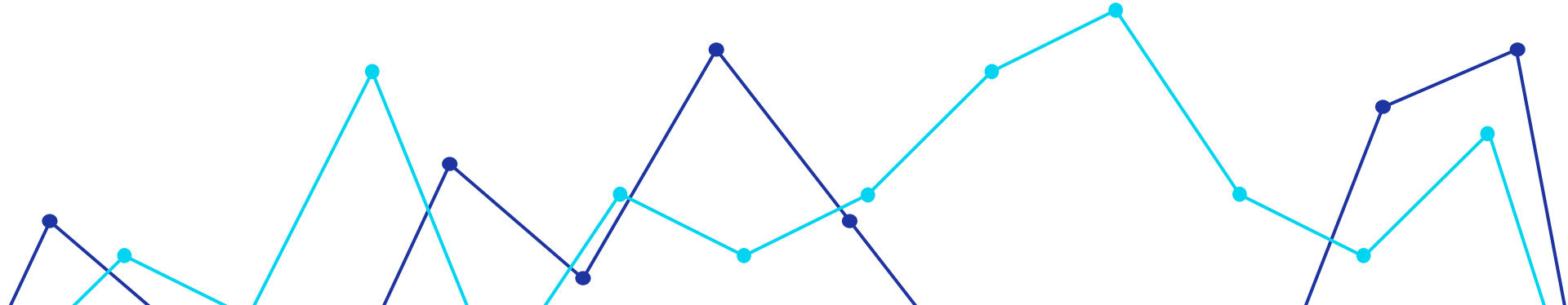
Monitorização da Performance de Linguagens de Programação

Experimentação em Engenharia de Software

Departamento de Informática
Universidade do Minho
2º Semestre | Ano Letivo 23/24

Daniel Du *pg53751*
José Fonte *a91775*
Ricardo Lucena *pg54187*



Motivação e Objetivos

Motivação

- Comparação de Desempenho
- Otimização de Código
- Utilização de acordo com Use Case
- Energia e Sustentabilidade
- Auxiliar a Tomada de Decisão
- Avaliar novas Tecnologias

Objetivos

- Correr diferentes *benchmarks* com cada linguagem de programação.
- Limitar performance em diferentes níveis.
- Criar gráficos para análise dos dados.
- Avaliar a performance e consumo de energia de cada *benchmark*.
- Encontrar o melhor ponto de relação performance - consumo de energia.

Metodologia de Trabalho

- Benchmarks em dois pcs diferentes.
- Linguagens de Programação : Python, Java e Haskell.
- Utilizar RAPL para avaliar tempo de execução,consumo de energia e memória.
- Utilizar POWERCAP para limitar os níveis de performance.

PC1

- Modelo: Lenovo Ideapad 5 14IIL05
- CPU: Intel i7-1065G7
  - Cores: 4
  - Threads:8
  - Base Frequency: 1.3GHz
  - Max Frequency: 3.9GHz
  - Cache: 8MB
  - Memory Support: DDR4-3200
- Integrated GPU: Intel Iris Plus Graphics G7
- External GPU: NVIDIA GeForce MX350
- RAM: 16GB
- OS: Kubuntu 22.04 LTS x86_64

PC2

- Modelo: Asus X550JX 1.0
- CPU: Intel i7-4720HQ
  - Cores: 4
  - Threads 8:
  - Base Frequency: 2.60 GHz
  - Max Frequency: 3.60 GHz
  - Cache: 6MB
  - Memory Support: DDR3L 1333/1600
- Integrated GPU: Intel HD Graphics 4600
- External GPU: NVIDIA GeForce GTX 950M
- RAM: 8GB
- OS: Linux Mint 21 x86_64

Metodologia de Trabalho

Comparação de Hardware

1. **CPU :** O Intel Core i7-1065G7 do PC1 é construído numa arquitetura mais recente (Ice Lake) e fabricado usando um processo de 10 nm, o que geralmente resulta em melhor eficiência de energia em comparação com o Intel Core i7-4720HQ do PC2 (arquitetura Haswell, processo de 22 nm).
2. **Memory :** A RAM DDR4 do PC1 geralmente consome mais energia em comparação com a RAM DDR3L do PC2, especialmente em velocidades mais altas.

No geral, o PC1 (Lenovo Ideapad 5 14IIL05) parece ter uma vantagem em termos de CPU, memória e GPU integrada, enquanto o PC2 (Asus X550JX 1.0) possui uma GPU externa mais poderosa. Isso significa que o desempenho geral e o consumo de energia do PC1 serão superiores.

Trabalho Desenvolvido

Conjuntos de Benchmarks

- Haskell : pyperformance (https://github.com/python/pyperformance)
- Python : NoFib (https://gitlab.haskell.org/ghc/nofib)
- Java : Dacapo (https://www.dacapobench.org/)

Estrutura da Análise de Cada Linguagem

- Versão do Compilador e dos *Benchmarks*
- *Benchmarks* executados
- Resultados de *Execution Time* e *Power Consumption* por cada *PowerCap*
- Análise dos Resultados

Haskell

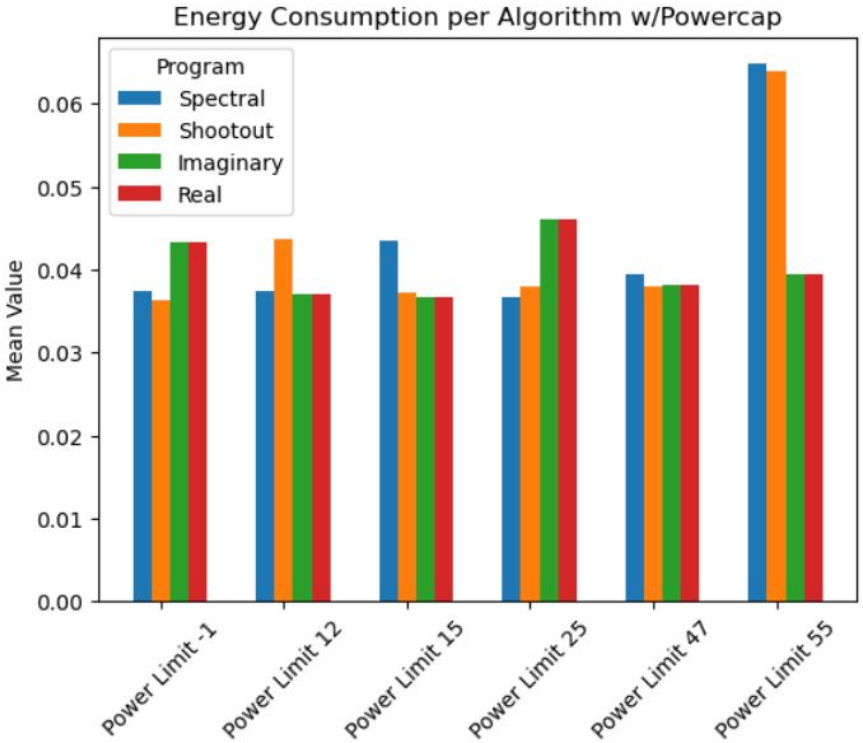
- Versão do Compilador : PC1 : 8.8.4 | PC2: 8.8.4
- Versão dos Benchmarks : nofib
- Powercaps: No Powercap, 12, 15, 25, 47, 55
- Resultados : Tempo de Execução, Consumo de Energia, Memória

Conjunto de Benchmarks

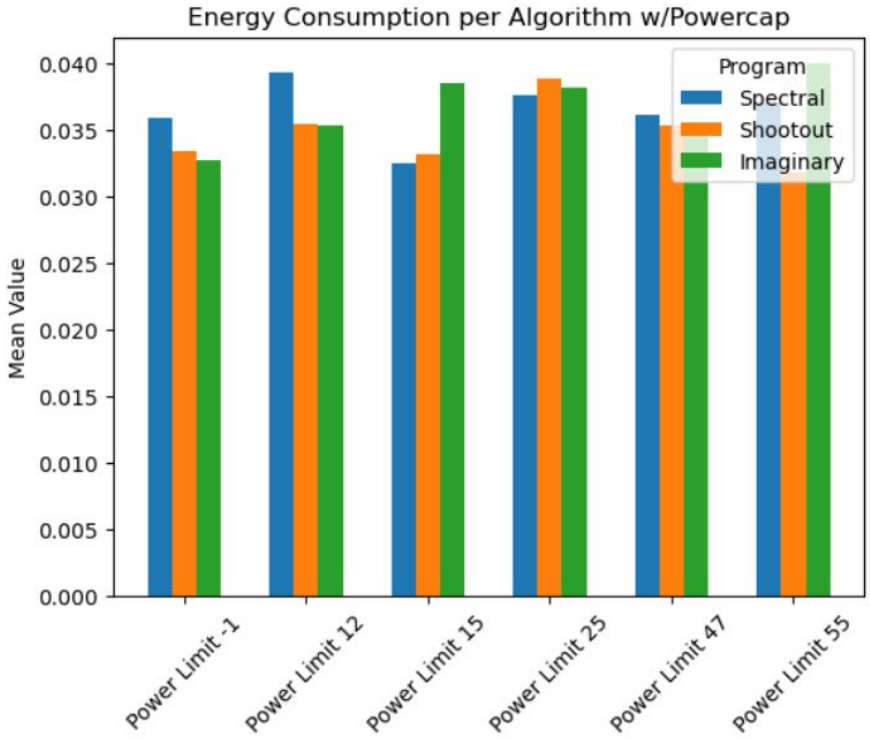
- **Imaginary** - Benchmarks direcionados à resolução de puzzles, como por exemplo n-queens.
- **Spectral** - Kernel Algoritmicos.
- **Real** - Aplicações reais, com interface de linhas de comando.
- **Shootout** - Benchmarks de um conjunto de benchmarks.

Haskell - Benchmarks (PC1 vs PC2)

PC1

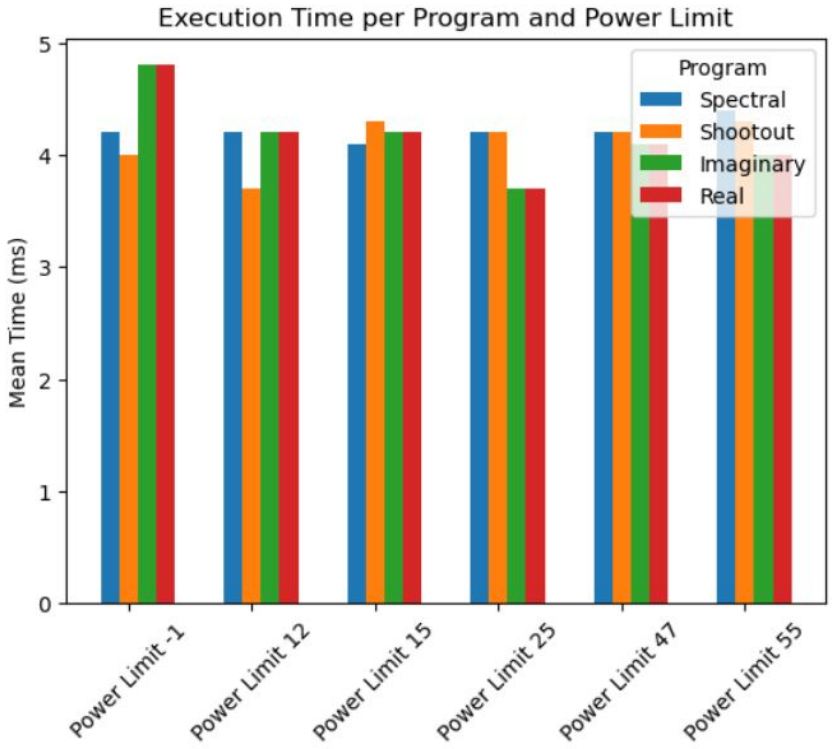


PC2

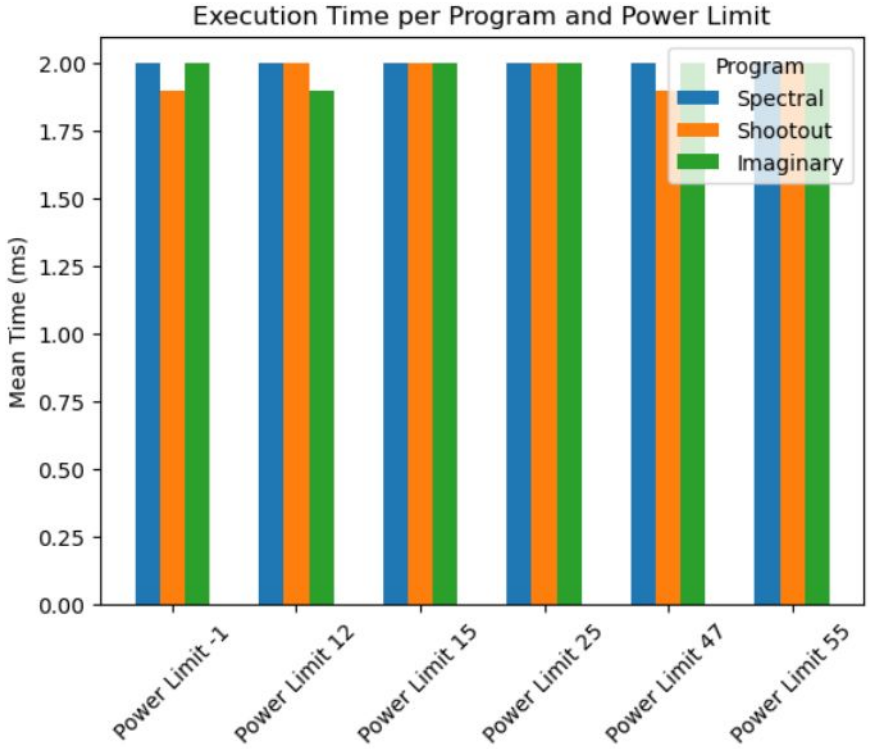


Haskell - Benchmarks (PC1 vs PC2)

PC1



PC2



Haskell - Análise de Resultados

Consumo de Energia

- O consumo de energia é constante nos diferentes powercaps - sendo que no global, o consumo do PC2 é ligeiramente inferior.
- Como o valores são constantes assume-se que o consumo é tão baixo que não se atinge os limites do PowerCap.
- Analisando cada benchmark, as variações entre os diferentes powercaps estão dentro do esperado.

Tempos de Execução

- Dado os benchmarks não atingirem os powercaps, os valores são iguais para todos os diferentes limites.
- Apesar das expectativas contrárias, observamos que o PC2 apresenta ligeiramente melhores resultados em comparação com o PC1.
- No entanto, as diferenças são mínimas, cerca de 2 a 3 ms. Isso pode ser atribuído ao OS, ao momento de execução, carga no PC de teste ou outros fatores.
- Deste modo, a conclusão é que o PC2 consegue melhor performance com menor consumo de energia.

Java

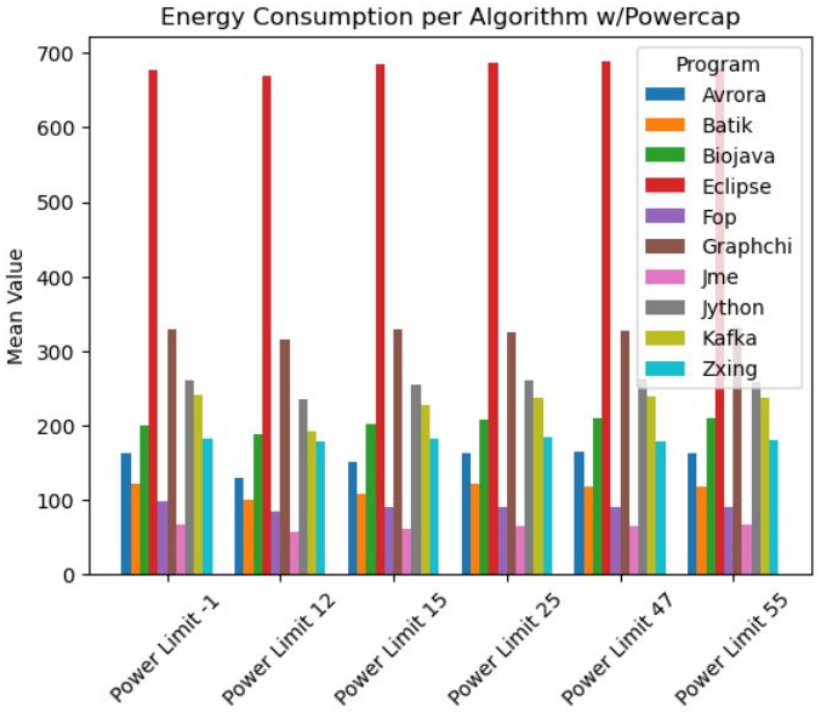
- Versão do Compilador : PC1 : 17.0.10 | PC2: 21.0.2
- Versão dos Benchmarks : Dacapo 23.11-chopin
- Powercaps: No Powercap, 12, 15, 25, 47, 55
- Resultados : Tempo de Execução, Consumo de Energia, Memória

Conjunto de Benchmarks

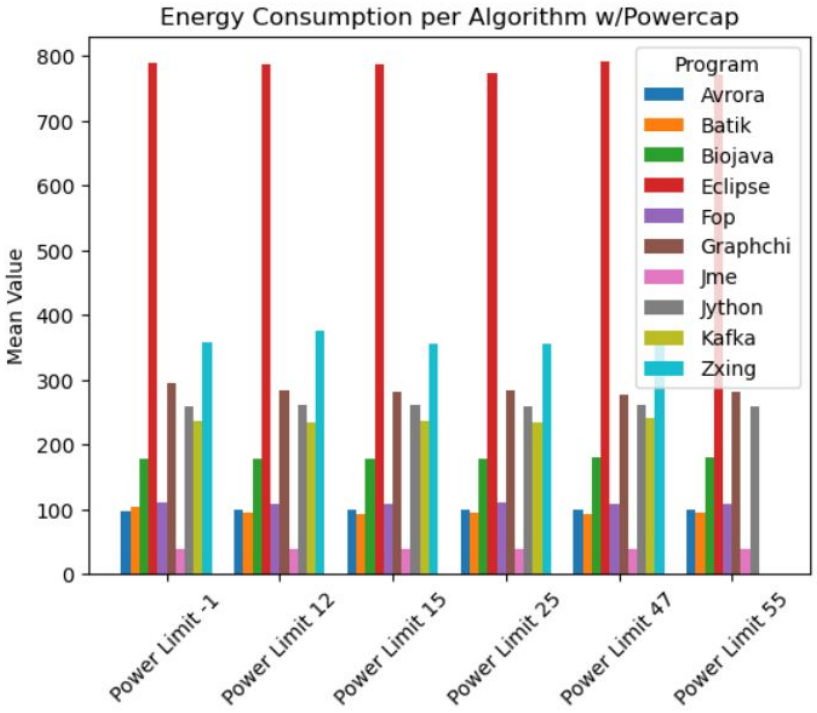
- **Simulation and Rendering:** Avrora, Sunflow, Batik
- **Data Processing and Analytics:** Graphchi, Kafka, H2O
- **Web Development and Server Management:** Tomcat, Eclipse, Spring
- **Database and Data Manipulation:** H2, Biojava
- **Search and Text Processing:** Luindex, Lusearch, PMD
- **XML Processing and Transformation:** Xalan, Fop
- **Targeting embedded systems and mobile devices:** JME
- **An implementation of Python in Java :** Jython

Java - Benchmarks (PC1 VS PC2)

PC1

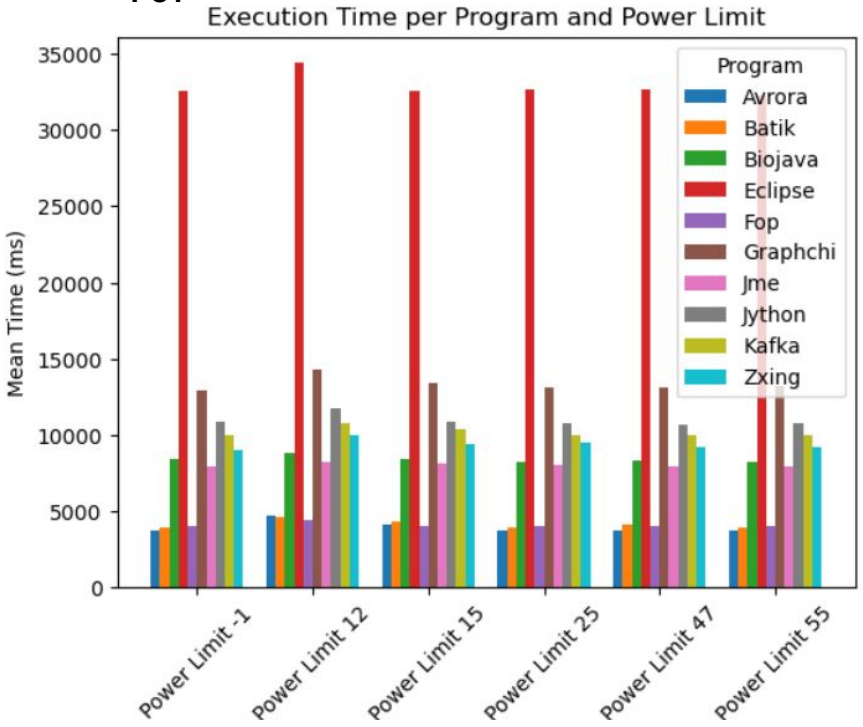


PC2

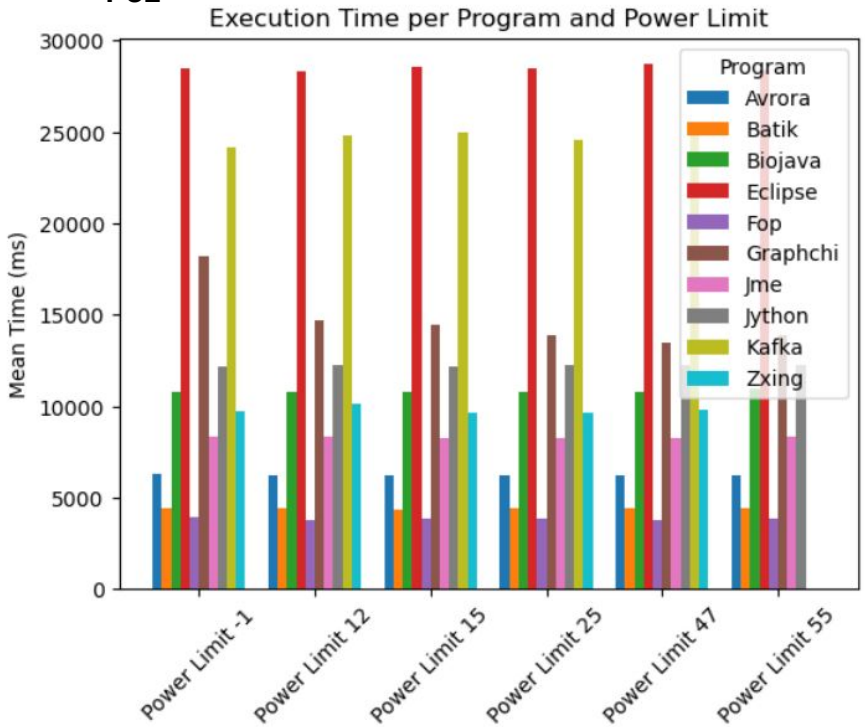


Java - Benchmarks (PC1 VS PC2)

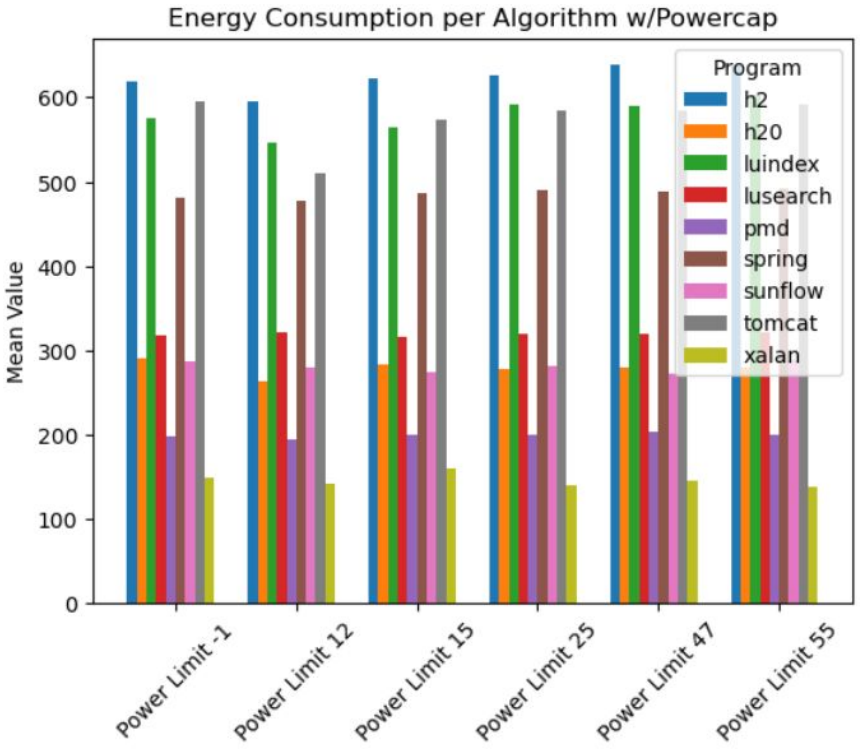
PC1

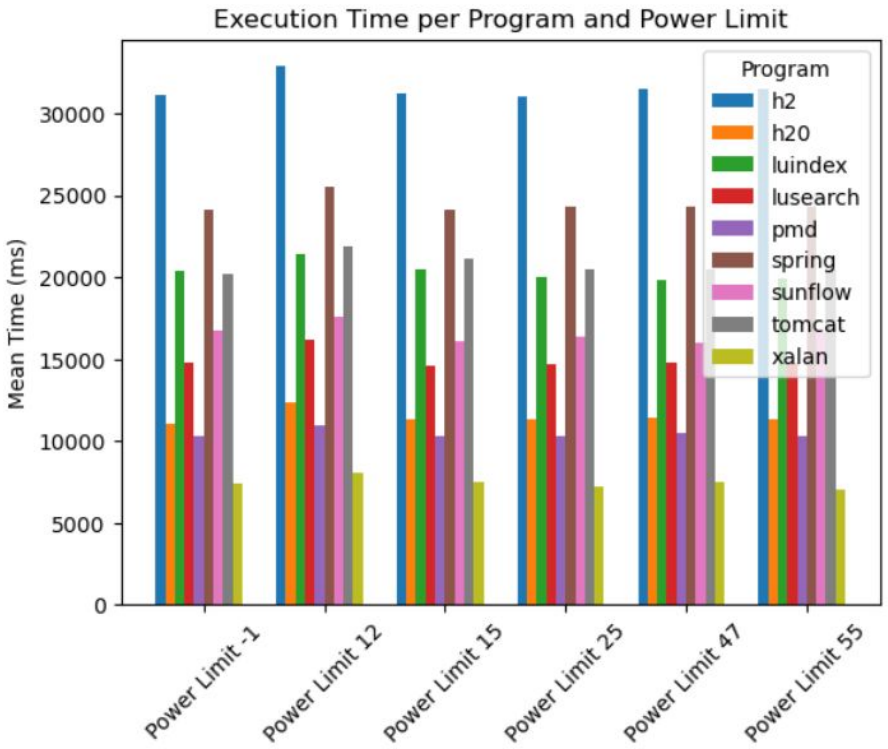


PC2



Java - Benchmarks (Apenas PC1)





Java - Análise de Resultados

Consumo de Energia

- Benchmarks mais pesados, em comparação ao Haskell.
- Diferenças de *Power Consumption* significativas entre Benchmarks.
- *Power Consumption* semelhantes apesar do PowerCap.
- Consumos mais altos no PC1, exceto nos *Benchmarks* Eclipse e Zxing.

Tempos de Execução

- *Benchmarks* Kafka, Graphchi muito diferentes entre PC1 e PC2.
- PC2 tende a obter resultados piores do que o PC1, exceto no *benchmark Eclipse* que é ligeiramente mais rápido.
- Para os *benchmarks* da categoria de "simulação e renderização" conseguimos observar que os tempos de execução do PC2 são maiores comparativamente o PC1, apesar do PC2 ter uma placa gráfica externa melhor. Desta forma conseguimos concluir que os *benchmarks* não requerem o uso de uma placa gráfica externa.
- No geral, todos os *benchmarks* apresentam resultados relativamente parecidos, independentemente do *powerlimit*.

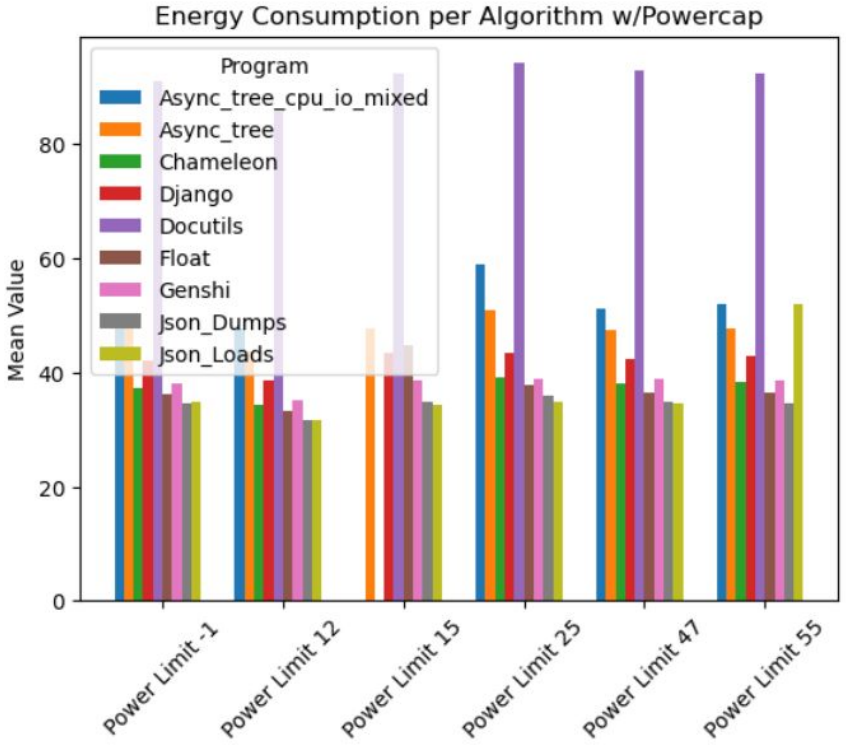
Python

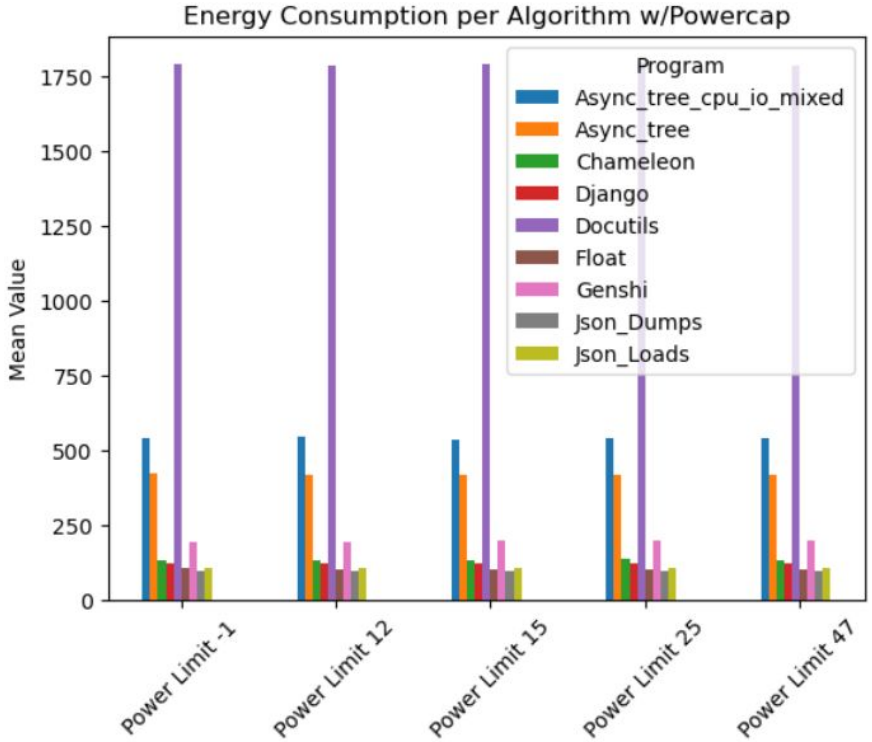
- Versão do Compilador : PC1 : 3.10.12 | PC2: 3.11.14
- Versão dos Benchmarks : pyperformance 1.11.0
- Powercaps: No Powercap, 12, 15, 25, 47, 55
- Resultados : Tempo de Execução, Consumo de Energia, Memória

Conjunto de Benchmarks

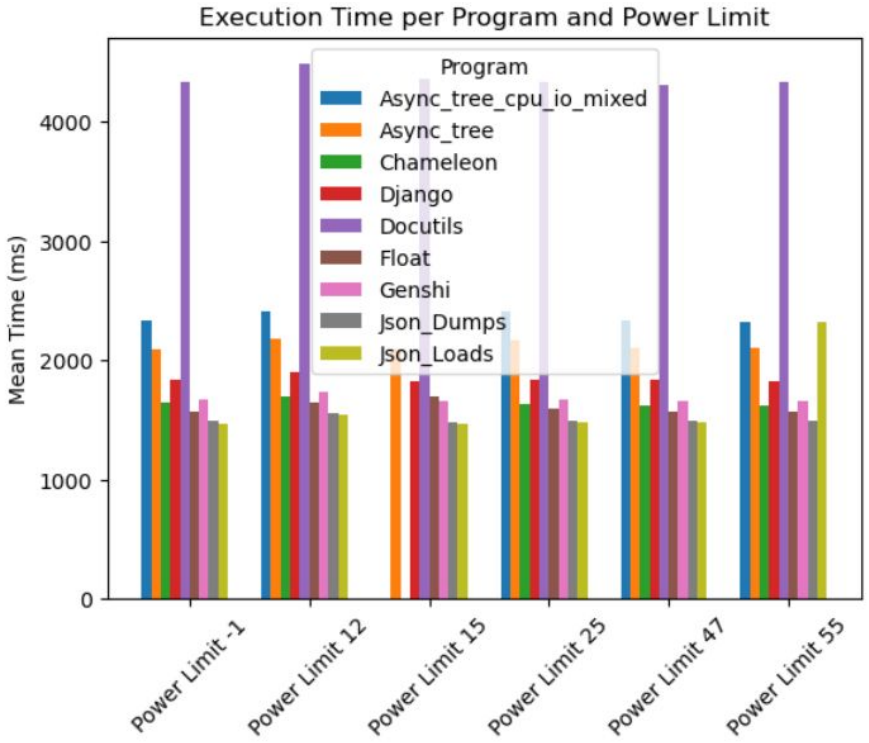
- **Apps:** 2to3, chameleon, docutils
- **Asyncio:** async_tree, async_tree_cpu_io_mixed, async_tree_cpu_io_mixed_tg
- **Math:** Float, Nbody, pidigits
- **Regex:** regex_compile, regex_dna, regex_effbot
- **Serialize:** json_dumps, json_loads, pickle
- **Startup:** python_startup, python_startup_no_site
- **Template:** django_template, genshi, mako

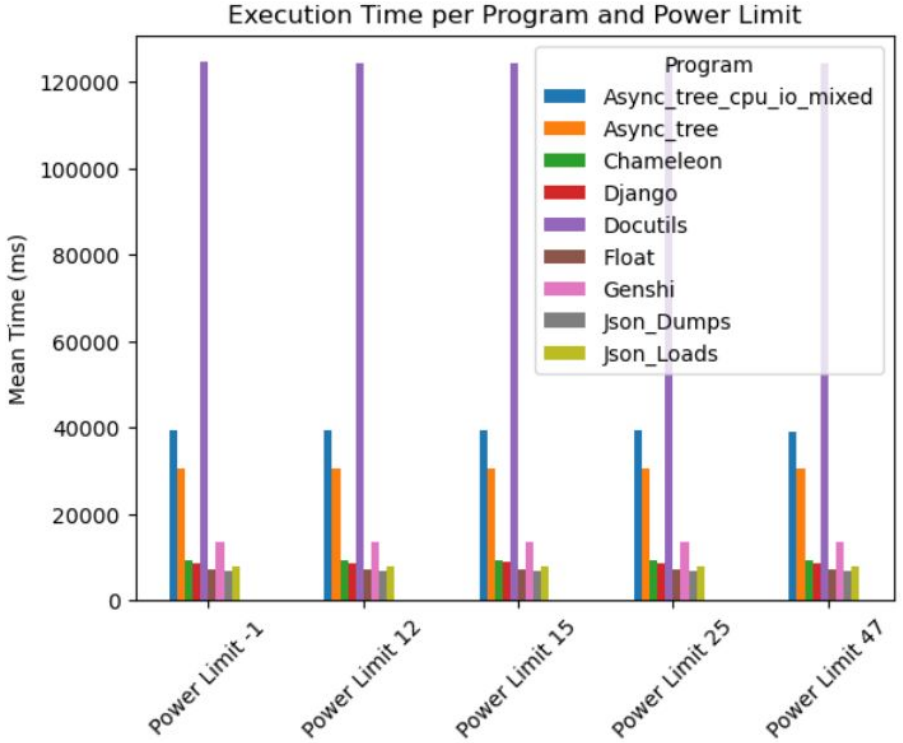
Python - Benchmarks (PC1 VS PC2)



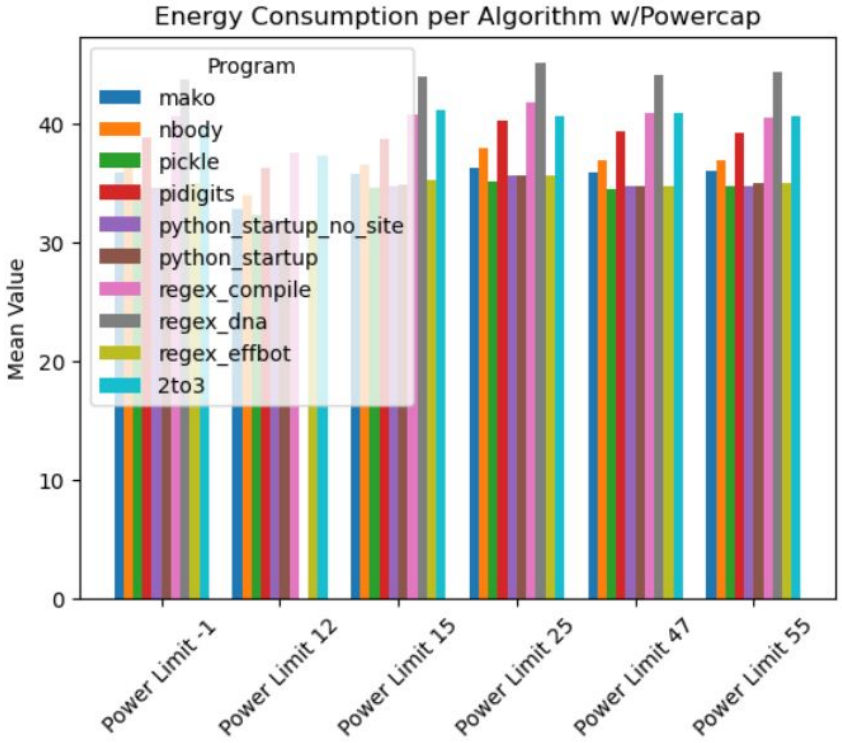


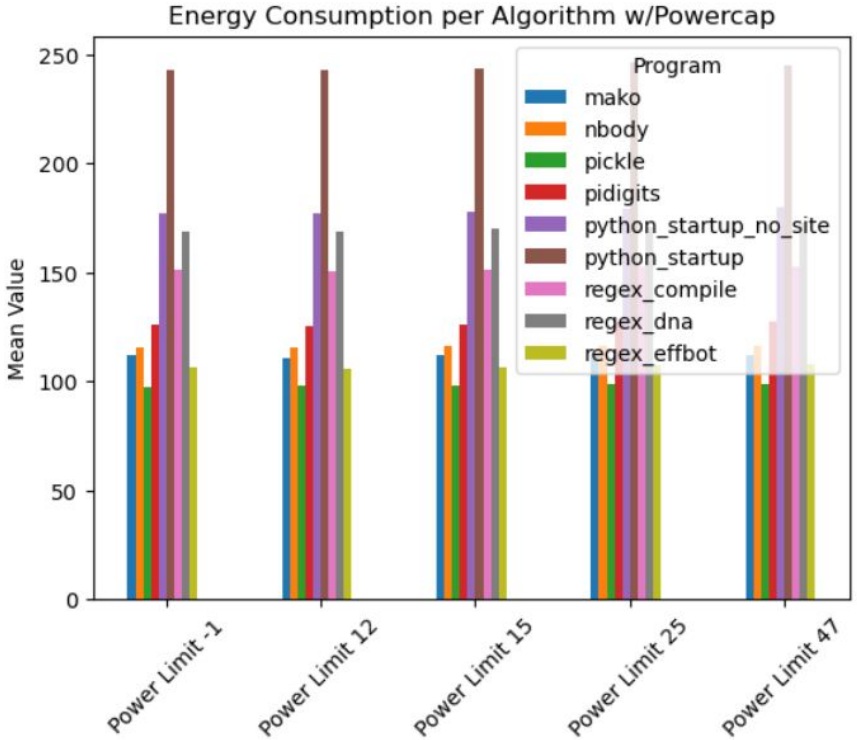
Python - Benchmarks (PC1 VS PC2)



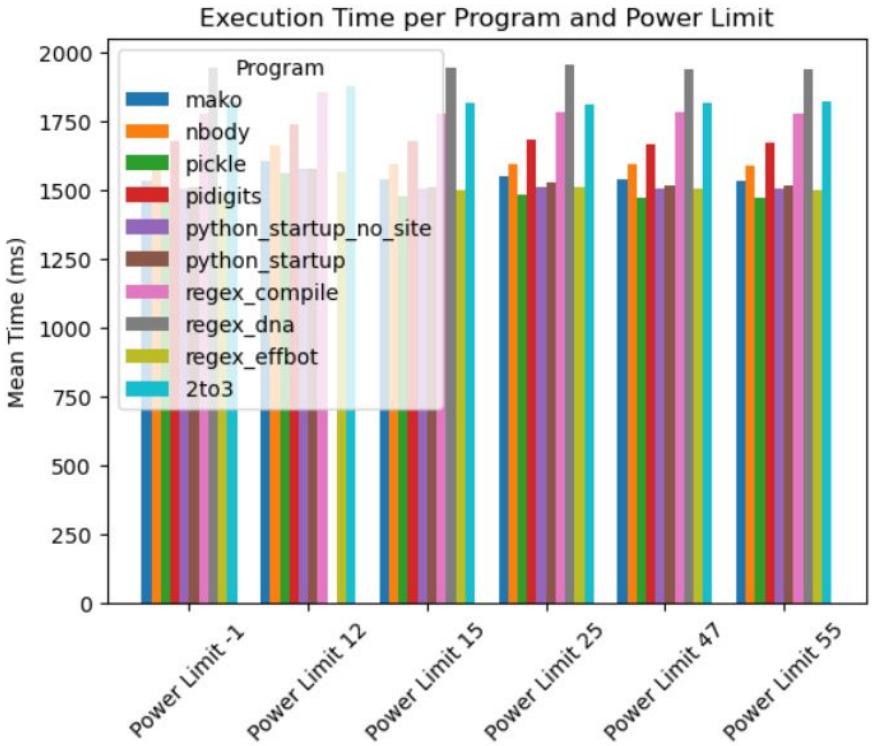


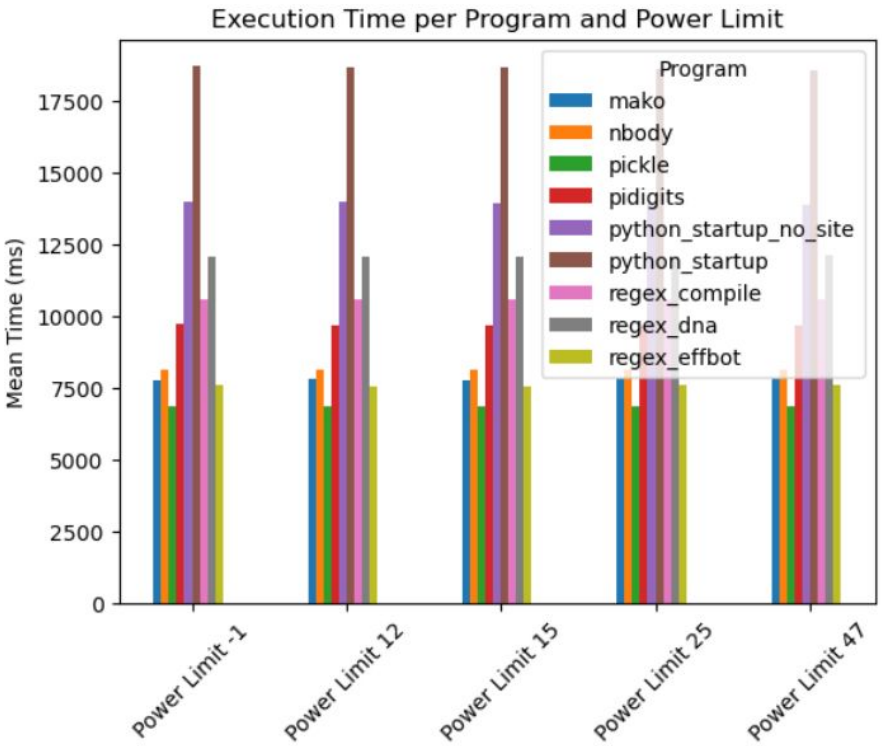
Python - Benchmarks (PC1 VS PC2)





Python - Benchmarks (PC1 VS PC2)





Python - Análise de Resultados

Consumo de Energia

- Docutils benchmark que mais energia consome para ambos os PCs.
- De forma geral, todos os benchmarks no PC2 consomem mais energia do que no PC1.
- Mais uma vez, conseguimos reparar que o powercap imposto pouca diferença faz uma vez que os resultados, ao contrário do que o esperado, muito parecidos.

Tempos de Execução

- Os tempos de execução no PC1 são todos menores do que no PC2.
- De todos os benchmarks, o *Docutils* é o que demora mais
- Todas as benchmarks no PC1 (com exceção do docutils) apresentam resultados relativamente parecidos, independentemente do powerlimit.

> man --help

De forma a conseguir correr os programas basta, na diretoria principal, correr o seguinte comando no terminal:

> ***Sudo sh measure.sh***

Através deste script é nos permitido executar os diversos benchmarks para todos os powercaps de forma automática e em simultâneo obter informações sobre o powercap imposto, a temperatura, tempos de execução e outras métricas.

Após ter o ficheiro measurements.csv com os resultados finais gerado, correr o testecsv.py (presente no zip “graficos” dentro da pasta “pc1” ou “pc2”) na mesma localização do CSV gerado anteriormente para separar o CSV grande em frações menores.

Depois de garantirmos que os *notebooks* estão com as diretorias dos csv’s corretas, podemos gerar os gráficos.

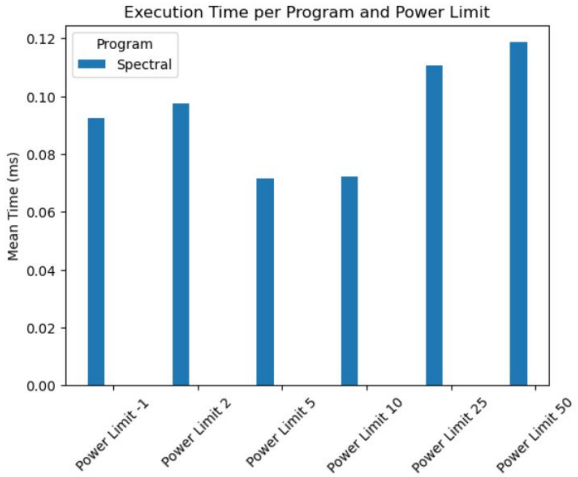
Alguns pontos a ter em conta

- Alguns *benchmarks* registaram valores negativos dado que o registo onde os valores eram guardados às vezes sofriam *overflow*. Como esses registos eram *outliers* o grupo decidiu removê-los por completo.
- Alguns *benchmarks* de Java do PC2 não correram com êxito, fazendo com que certos *benchmarks* só tenham valores do PC1.
- No PC1 há poucos *benchmarks* que não executaram com sucesso num determinado *powercap*. O grupo achou que a melhor forma a resolver este problema seria através da criação de uma linha com valores “0” para o determinado registo, ao invés de ter uma linha com valor “error”.
- Não existem registos do uso de *powercap* com valor 55 no PC2 como existe no PC1 para os *benchmarks* de Python, dado que o tempo de execução do programa inteiro estar a demorar demasiado tempo e o proprietário do mesmo viu-se forçado a interromper a execução do RAPL.
- Todos os *benchmarks* foram corridos 10 vezes.

Conclusões Finais

O grupo observou que não existe diferenças significativas no *Power Consumption* e no *Time Execution* entre *powercaps* diferentes, o que não se enquadra com as nossas expectativas e as do corpo docente. No entanto, esta análise só foi realizada próxima à data de entrega, devido ao longo tempo de execução dos benchmarks, não sendo assim viável uma nova execução dentro do prazo estipulado.

Para tentar identificar os possíveis problemas, o grupo realizou um *benchmark* simples para verificar se há de facto diferenças entre os limites de energia, onde concluiu que o problema persistia mesmo em problemas de menor dimensão.



Monitorização da Performance de Linguagens de Programação

Experimentação em Engenharia de Software

Departamento de Informática
Universidade do Minho
2º Semestre | Ano Letivo 23/24

Daniel Du *pg53751*
José Fonte *a91775*
Ricardo Lucena *pg54187*