

# Avaliação de Desempenho, Custo e Eficiência Eenergética de Linguagens de Programação

Discente: Michel Tavares de Oliveira  
Orientador: Jean Carlos Teixeira de Araújo

Universidade Federal do Agreste de Pernambuco

Maio de 2021



- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Relacionados
- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
- 6 Resultados
- 7 Conclusão



# Frame Title

- A
- B
- C
- D

1 Motivacao

2 Questões de pesquisa

3 Trabalhos Realacionados

4 Fundamentação teorica

5 Materiais,métodos e execução

6 Resultados

7 Conclusão

## Questões de pesquisa

- Questão de pesquisa 1. Qual linguagem de programação é mais eficiente energeticamente?
- Questão de pesquisa 2. A linguagem mais rápida, é mais eficiente energeticamente?
- Questão de pesquisa 3. A linguagem que possui menor potência em Watts, foi a mais eficiente?

- 1 Motivação
- 2 Questões de pesquisa
- 3 **Trabalhos Realacionados**
- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
- 6 Resultados
- 7 Conclusão

# Energy Efficiency across Programming Languages: How Do Energy, Time, and Memory Relate?

- A
- B
- C
- D





1 Motivacao

2 Questões de pesquisa

3 Trabalhos Realacionados

4 **Fundamentação teorica**

Avaliação de desempenho de sistemas

Green Software

Eficiência energética

Métricas energéticas

5 Materiais,métodos e execução

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Relacionados
- 4 **Fundamentação teórica**
  - Avaliação de desempenho de sistemas
  - Green Software
  - Eficiência energética
  - Métricas energéticas
- 5 Materiais, métodos e execução

## Avaliação de desempenho de sistemas

- A
- B
- C
- D

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Relacionados
- 4 **Fundamentação teórica**
  - Avaliação de desempenho de sistemas
  - Green Software**
  - Eficiência energética
  - Métricas energéticas
- 5 Materiais, métodos e execução

# Green Software

- A
- B
- C
- D

1 Motivacao

2 Questões de pesquisa

3 Trabalhos Realacionados

4 **Fundamentação teorica**

Avaliação de desempenho de sistemas

Green Software

**Eficiência energética**

Métricas energéticas

5 Materiais,métodos e execução

## Eficiência energética

- A
- B
- C
- D



- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Relacionados
- 4 **Fundamentação teórica**
  - Avaliação de desempenho de sistemas
  - Green Software
  - Eficiência energética
  - Métricas energéticas**
- 5 Materiais, métodos e execução

- É a unidade de energia no Sistema Internacional de Unidades, utilizada para medir energia mecânica ou térmica
- Na energia mecânica 1 Joule equivale a energia necessária aplicar força 1 Newton por 1 metro
- Na energia térmica 1 Joule equivale a energia necessária para aumentar a temperatura da água a 1 grau

- $$P = \frac{E}{t} \quad (1)$$

- $$\text{Potência em quilowatts (kW)} = \frac{\text{Energia em kilojoules (kJ)}}{\text{Tempo em horas (h)}} \quad (2)$$

## Quilowatt-hora

- Referente a energia produzida ou consumida no período de 1 hora

$$\text{Energia total (kWh)} = \text{Potência em Quilowatts (kW)} \times \text{Tempo total em horas (h)} \quad (3)$$

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Relacionados
- 4 Fundamentação teórica
- 5 **Materiais, métodos e execução**
  - Intel RAPL
  - Utilitários Linux
  - The Computer Language Benchmark Game
  - Linguagens de Programação

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Realacionados
- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
  - Intel RAPL
  - Utilitários Linux
  - The Computer Language Benchmark Game
  - Linguagens de Programação





# RAPL Power Domain

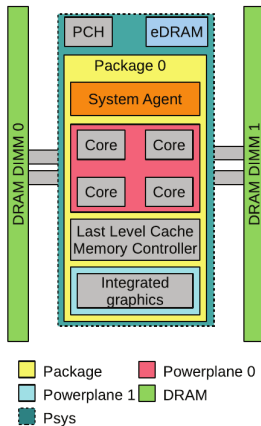


Figure 1: Intel RAPL Power Domains. Fonte: Khan at al 2018, [1]

## Contadores de energia e limitações

- Model-Specific Registers (MSRs) são registradores que fornecem acesso a diversas características e funcionalidades nos processadores x86
- MSR\_RAPL\_POWER\_UNIT de 32 bits sem sinal (0 a 4,294,967,295)
- O registrador começa a contar a partir da inicialização do computador
- Quando este valor limite é atingido, o valor do registrador é reinicializado para 0 novamente.
- É de grande importância considerar a reinicialização dos registradores para não obter dados incorretos durante os experimentos

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Realacionados
- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
  - Intel RAPL
  - Utilitários Linux
  - The Computer Language Benchmark Game
  - Linguagens de Programação

## Power Capping Framework

- Ferramenta integrada ao Kernel Linux
- Permite expor informações de energia via *sysfs* exportando informações sistemas de arquivos
- O framework cria de forma automática, uma árvore de diretórios com diversos objetos referente a interface de energia utilizada (The Linux Kernel Archives, 2024).

# Árvore de Diretórios Power Capping Framework

```

/sys/devices/virtual/powercap
└─ intel-rapl
    ├── enabled
    ├── intel-rapl:0
    │   ├── constraint_0_max_power_uw
    │   ├── constraint_0_name
    │   ├── constraint_0_power_limit_uw
    │   ├── constraint_0_time_window_us
    │   ├── constraint_1_max_power_uw
    │   ├── constraint_1_name
    │   ├── constraint_1_power_limit_uw
    │   ├── constraint_1_time_window_us
    │   ├── device -> ../../intel-rapl
    │   └── enabled
    └── ...

```

Figure 2: Árvore de Diretórios Power Capping Framework

# Shell

- Interface entre o usuário e o sistema operacional
- O Shell é uma ferramenta essencial quando o foco é ter mais controle sobre o sistema operacional (RAYMOND, 2003)
- Funciona como um intermediário entre usuário e SO
- Essa interação pode ocorrer de forma iterativa e não iterativa

# Shell Script

- É uma linguagem de script voltada para automatização de tarefas em sistemas operacionais, sendo ela interpretada por um interpretador Shell
- Permite realizar diversas tarefas executando apenas um arquivo de script
- O interpretador analisa linha por linha e executa os comandos encontrados de forma sequencial
- Bastante útil ao executar diversos comandos, assim como a possibilidade de realização de tarefas repetitivas e automáticas

# Bash

- Bash é um shell desenvolvido por Brian Fox no Projeto GNU
- Atualmente é o Shell padrão de diversas distribuições Linux, como Ubuntu, Debian e Manjaro.



# GNU Time

- Utilizada para medir tempo e recursos consumidos por uma aplicação durante sua execução
- Utilização é bastante simples, com opções salvar a saída dos resultados em arquivo de texto

```
/usr/bin/time ./meu_programa > output.txt
```

1 Motivacao

2 Questões de pesquisa

3 Trabalhos Realacionados

4 Fundamentação teorica

5 Materiais,métodos e execução

Intel RAPL

Utilitários Linux

The Computer Language Benchmark Game

Linguagens de Programação



# CLBG Benchmarks

Benchmarks	Descrição
fannkuch-redux	Acesso indexado a minúsculas sequências inteiras
n-body	Dupla precisão para cálculo de N-body
spectral-norm	Autovalor usando o método da potência
pidigits	Streaming de aritmética de precisão arbitrária
regex-redux	Combina DNA e substitui por padrões mágicos
fasta	Gerar e escrever sequências aleatórias de DNA
k-nucleotide	Atualiza hashtable e sequências de k-nucleotídeos
reverse-complement	Complemento reverso de sequências de DNA
binary-trees	Aloca e desaloca muitas árvores binárias
mandelbrot	Gera um conjunto de Mandelbrot em arquivo de bitmap portátil

- 1 Motivação
- 2 Questões de pesquisa
- 3 Trabalhos Realacionados
- 4 Fundamentação teórica
- 5 **Materiais, métodos e execução**
  - Intel RAPL
  - Utilitários Linux
  - The Computer Language Benchmark Game
  - Linguagens de Programação

# Linguagens de Programação

Linguagem	Versão	Compilador Open Source (Ubuntu 22.04)
Ada	10.5.1	GNAT GPL Compiler
C	11.4.0	GCC
C#	7.0.115	Mono
C++	11.4.0	GCC
Chapel	1.29.0	Chapel Compiler
Dart	3.2.6	Dart SDK
Erlang	26.2.2	Erlang OTP
F#	7.0.115	F# Compiler
Fortran	11.4.0	GFortran
Go	1.18.1	Go Compiler
Haskell	8.8.4	GHC Haskell Compiler
Java	19.0.2	OpenJDK
Javascript	18.19.0	V8
Julia	1.9.3	Julia Compiler
Lua	5.3.0	LuaJIT
OCaml	4.13.1	OCaml Compiler
Perl	5.34.1	Perl Compiler
Php	8.2.15	PHP Compiler
Python	3.10.12	Python Interpreter
Racket	8.2.0	Racket Compiler
Ruby	3.0.2	Ruby Compiler
Rust	1.75.0	Rustc Compiler
Swift	5.9.0	Swift Compiler

1 Motivacao

2 Questões de pesquisa

3 Trabalhos Realacionados

4 Fundamentação teorica

5 Materiais,métodos e execução

6 Resultados

7 Conclusão

# Frame Title

- A
- B
- C
- D



- 1 Motivacao
- 2 Questões de pesquisa
- 3 Trabalhos Realacionados
- 4 Fundamentação teorica
- 5 Materiais,métodos e execução
- 6 Resultados
- 7 Conclusão

# Frame Title

- A
- B
- C
- D

- [1] Kashif Nizam Khan et al. "RAPL in Action: Experiences in Using RAPL for Power Measurements". In: *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3.2 (2018). ISSN: 2376-3639. DOI: 10.1145/3177754. URL: <https://doi.org/10.1145/3177754>.

EOF