Avaliação de desempenho, custo e eficiência energética de linguagens de programação

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

Universidade Federal do Agreste de Pernambuco

Maio de 2024





- Motivação
- Questões de pesquisa
- Trabalhos ralacionados
- 4 Fundamentação teórica
- 6 Materiais.métodos e execução
- 6 Resultados
- Conclusão



- Motivação
- Questões de pesquisa
- Trabalhos ralacionados
- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
- 6 Resultados
- Conclusão

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais, métodos e execução Resultados Conclusão Reference

Frame Title

- Escolher linguagens mais eficientes energeticamente pode resultar em menor consumo de energia, custos operacionais reduzidos e menor impacto ambiental.
- O impacto do consumo excessivo de energia nos custos operacionais, no preço dos serviços para o usuário final e no meio ambiente
- Recursos finitos como energia

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

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

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?

- Trabalhos ralacionados

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

- Energy Efficiency Across Programming Languages: How Do Energy, Time, and Memory Relate?
- Towards a Green Ranking for Programming Languages
- Analyzing Programming Languages's Energy Consumption: An Empirical Study
- D

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

- A
- B
- D

- 1 Motivaçã
- Questões de pesquisa
- 3 Trabalhos ralacionados
- 4 Fundamentação teórica
 - Avaliação de desempenho de sistemas
 - Eficiência energética
 - Métricas energéticas
- 6 Materiais, métodos e execução

- Motivação
- Questões de pesquisa
- 3 Trabalhos ralacionados
- 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
- (
- D

- 4 Fundamentação teórica

 - Green software



Green software

Desenvolvimento e uso de software de forma sustentável.

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

Eficiência energética

- A
- B
- D

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



Joule

- 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 nescessária aplicar força 1 Newton por 1 metro
- Na energia termica 1 Joule equivale a energia nescessária para aumentar a temperatuda da água a 1 grau

Watt

- É a unidade de potencia no Sistema Internacional de Unidades, sendo a potencia media da quantidade de energia em um determinado tempo
- 1 Watt equivale a 1 Joule por segundo, logo, um dispositivo que consome 1 Watt está consummindo um Joule por segundo

$$P = \frac{E}{t} \tag{1}$$

Quilowatt

- É uma unidade de potência referente a 1000 Watts
- Utilizada para medir portencia eletrica em aplicações residenciais e comerciais e industriais

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

Quilowatt-hora

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

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

- Materiais.métodos e execução

The Computer Language Benchmark Game



- Materiais.métodos e execução Intel RAPL

The Computer Language Benchmark Game



Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

Intel RAPL

- Otimizar o gerenciamento energético dos processadores Intel
- Monitoramento de alguns parametros como temperatura, potência e consumo energético
- \bullet Foi implementado a nível de hardware a partir da $6^{\rm a}$ geração dos processadores Intel
- Segundo Khan at al 2018 [1], a precisão do RAPL é bastante promisora e os valores reportados são precisos suficientemente para prever e modelar sistemas.

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

RAPL Power Domain

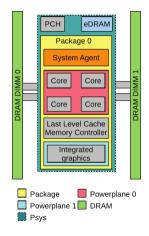


Figure 1: Intel RAPL Power Domains. Fonte: Khan at al 2018 [1]

Contadores de energia e limitações

- 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

- Materiais.métodos e execução

Utilitários Linux

The Computer Language Benchmark Game

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

```
/svs/devices/virtual/powercap
L— 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
```

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

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

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
```



- Materiais.métodos e execução

The Computer Language Benchmark Game



The Computer Language Benchmark Game

- É um projeto de software livre que fornece um repositório com uma variedade de algorítimos simples que podem ser implementados em diversas linguagens de programação
- Um web site que centraliza todos os dados sobre códigos fontes, execução de teste e resultados
- O projeto fornece, além do código fonte, informações sobre compilação e execução dos algorítimos

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

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



- Materiais.métodos e execução

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	
С	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	



- Materiais.métodos e execução

The Computer Language Benchmark Game

Ambiente de testes



Hardware

Especificação	Descrição	
Modelo	Intel® Core TM i5-10210U	
Geração	10ª Geração	
Codinome	Comet Lake	
Nº de núcleos	4	
Threads	8	
Frequência base	1.60 GHz	
Frequência turbo max	4.20 GHz	
Cache	6 MB Intel® Smart Cache	
Litografia	14 nm	
TDP	15 W	



Hardware

Especificação	Descrição
Memória DRAM	8GB (Dual Channel)
Frequência da Memória DRAM	2666MHz
Tipo de Memória DRAM	DDR4
Armazenamento Interno	256GB
Tipo do Armazenamento Interno	PCIe NVMe M.2
Potência Máxima da Fonte de Alimentação	45W



Sistema operacional

Especificação	Descrição	
Distribuição	Ubuntu	
Versão	22.04 LTS	
Codinome	Jammy Jellyfish	
Ambiente de desktop	KDE Plasma	
Kernel	Linux 5.15	
Arquitetura	x86 64 bits	
Tipo de instalação	Desktop	



- 6 Resultados

- 6 Resultados

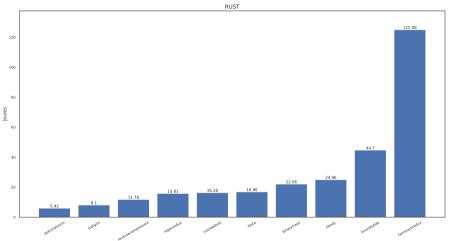
Análise das métricas por linguagem de programação



Análise das métricas por linguagem de programação

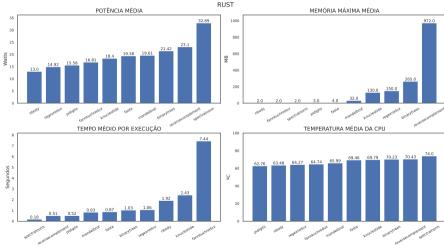
- Potência média
- Memória máxima média
- Tempo médio por execução
- Temperatura média da CPU

Consumo em Joules por problema





Potência, memória, tempo e temperatura



- 6 Resultados

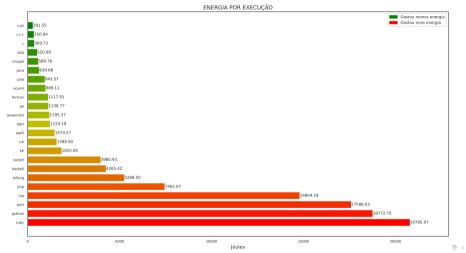
Eficiência energética



Avaliação de desempenho, custo e eficiência energética de linguagens de programação

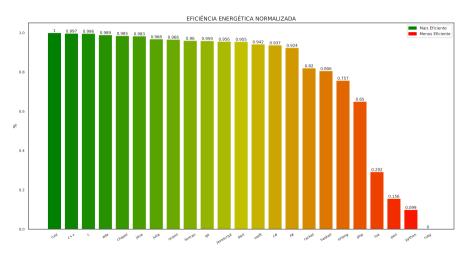
Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais,métodos e execução Resultados Conclusão Reference

Energia por execução



Discente: Michel Tavares de Oliveira Orientador: Jean Carlos Teixeira de Araújo Avaliação de desempenho, custo e eficiência energética de linguagens de programação

Eficiência energética





- 6 Resultados

Análise do custo financeiro



Análise do custo financeiro por 1000 execuções

- A tarifa média considerada foi de 0.73 centavos por kWh
- Utilizamos o consumo total em Joules e convertemos para kWh
- Multiplicamos o consumo total em kWh por 1000 execuções

Consumo em kWh =
$$\frac{\text{Consumo em Joules}}{3.600.000}$$
 (4)

Linguagens de programação

Linguagem	Joules	kWh	kWh por mil execuções	Custo por mil execuções (R\$)
Rust	291,55	0,000081	0,081	0,05913
C++	350,94	0,000098	0,098	0,07154
С	369,72	0,000103	0,103	0,07519
Ada	520,69	0,000145	0,145	0,10585
Chapel	589,76	0,000164	0,164	0,11972
Java	639,68	0,000178	0,178	0,12994
Julia	943,57	0,000262	0,262	0,19126
Ocaml	989,11	0,000275	0,275	0,20075
Fortran	1117,91	0,000310	0,310	0,2263
Go	1138,77	0,000316	0,316	0,23068
Javascript	1195,37	0,000332	0,332	0,24236
Dart	1224,19	0,000340	0,340	0,2482
Swift	1479,57	0,000411	0,411	0,30003
C#	1588,90	0,000441	0,441	0,32193
F#	1855,95	0,000516	0,516	0,37668
Racket	3980,93	0,001106	1,106	0,80738
Haskell	4265,42	0,001184	1,184	0,86432
Erlang	5268,50	0,001463	1,463	1,06799
PHP	7462,67	0,002073	2,073	1,51329
Lua	14804,18	0,004112	4,112	3,00176
Perl	17596,03	0,004888	4,888	3,56824
Python	18772,75	0,005214	5,214	3,80622
Ruby	20795,07	0,005777	5,777	4,21721



- 1 Motivação
- Questões de pesquisa
- Trabalhos ralacionados

- 4 Fundamentação teórica
- 5 Materiais, métodos e execução
- 6 Resultados
- Conclusão

Qual linguagem de programação é mais eficiente energeticamente?

- Rust (291.55 Joules)
- C++ (350.94 Joules)
- C (369.72 Joules)

Motivação Questões de pesquisa Trabalhos ralacionados Fundamentação teórica Materiais, métodos e execução Resultados **Conclusão** Reference

A linguagem mais rápida, é mais eficiente energeticamente?

- Na grande maioria dos casos sim, mas
- Julia apresentou maior tempo de execução em relação a Go e Chapel
- Conseguiu ser mais eficiente mesmo sendo mais lenta

A linguagem que apresentou menor potencia em watts, foi a mais eficiente?

- Não
- Pontência menor resulta em um maior tempo de execução
- Linguagens com maior potência nescessitavam de um menor tempo de execução

https://doi.org/10.1145/3177754.

EOF