Avaliação de desempenho

Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

Introdução

- Desempenho em hardware
  - *Pipeline*
    - Aumento da taxa de execução
    - Melhor aproveitamento do hardware

Introdução

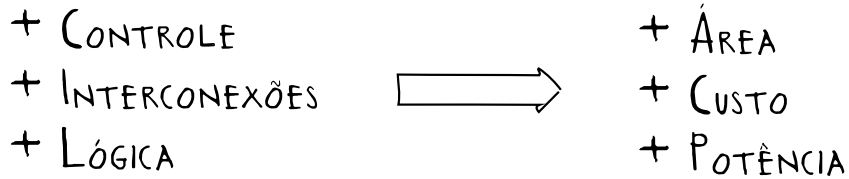
- Desempenho em hardware
  - *Pipeline*
    - Aumento da taxa de execução
    - Melhor aproveitamento do hardware
  - *Superescalar*
    - Paralelismo entre instruções
    - Aumento do desempenho de execução

Introdução

- Desempenho em hardware
  - *Pipeline*
    - Aumento da taxa de execução
    - Melhor aproveitamento do hardware
  - *Superescalar*
    - Paralelismo entre instruções
    - Aumento do desempenho de execução
  - Multiprocessamento
    - Paralelismo de processo e *thread*
    - O software precisa explorar o paralelismo

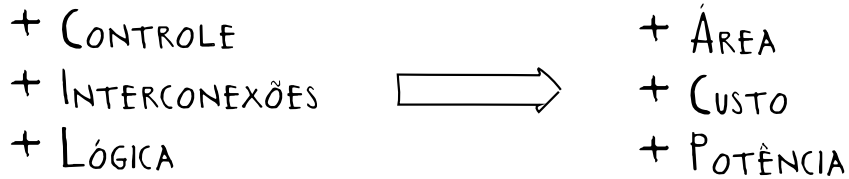
Introdução

- Aumento da complexidade do projeto

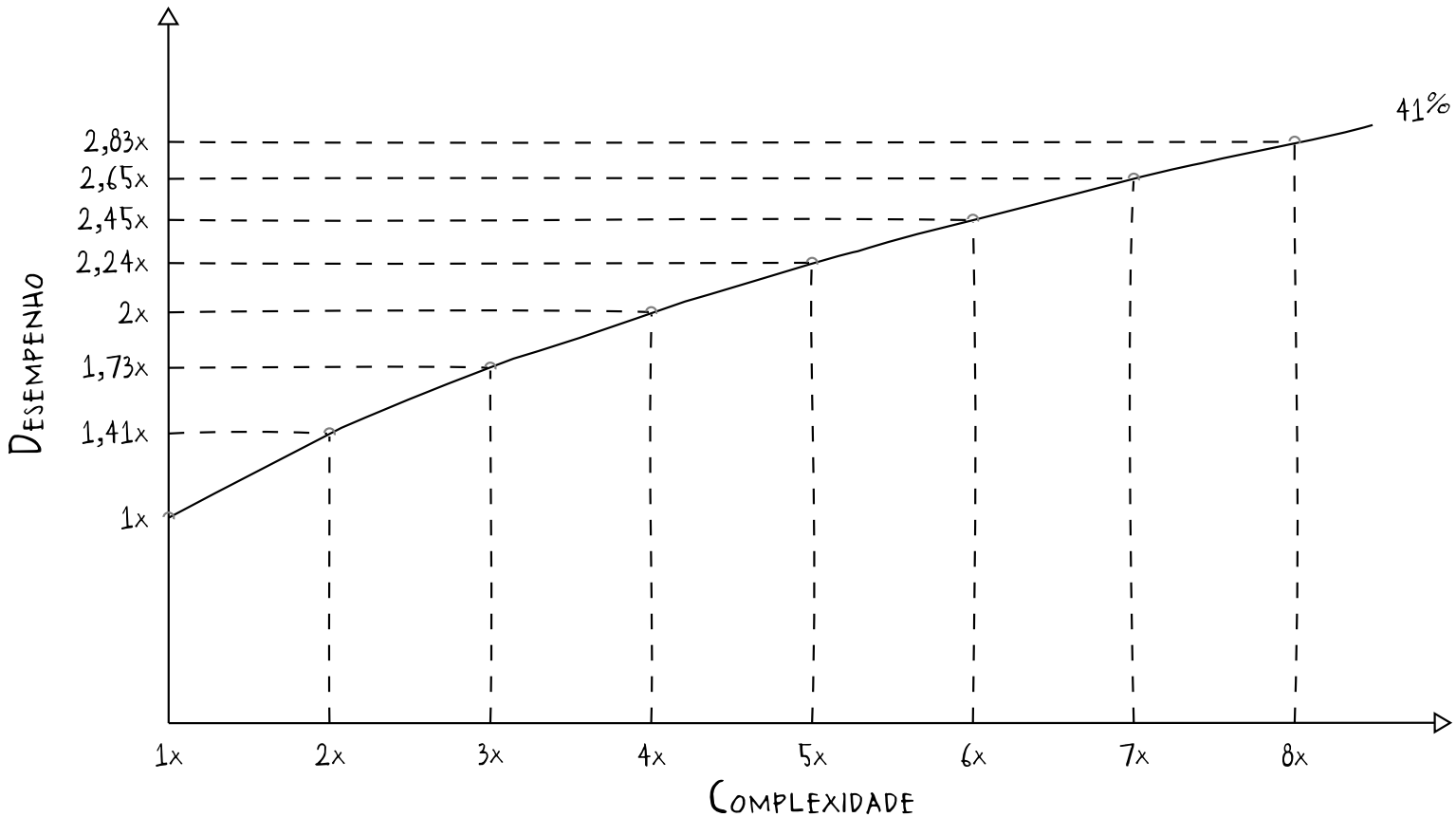


Introdução

- Aumento da complexidade do projeto



- Regra de Pollack



Introdução

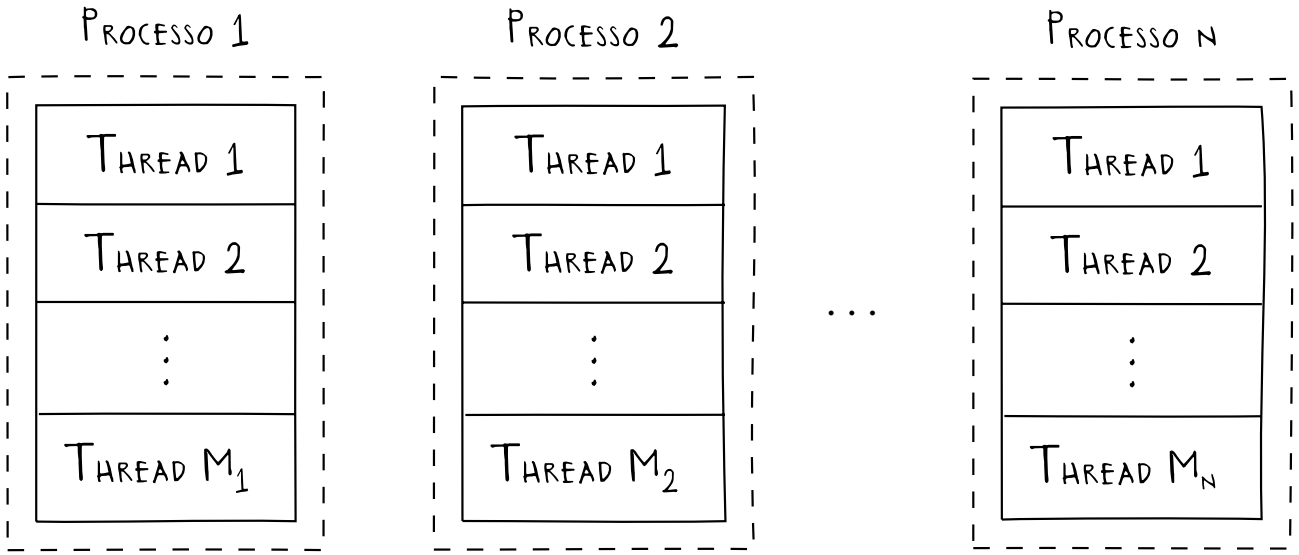
- Qual a diferença entre processo e *thread*?
  - Processo: é uma instância independente de uma aplicação que executa com escalonamento feito pelo SO, além de seu próprio contexto, memória virtual e recursos alocados

Introdução

- Qual a diferença entre processo e *thread*?
  - Processo: é uma instância independente de uma aplicação que executa com escalonamento feito pelo SO, além de seu próprio contexto, memória virtual e recursos alocados
  - *Thread:* só existe como parte de um processo e seu escalonamento pode ser feito pelo programador, compartilhando os mesmos recursos do processo

Introdução

- Qual a diferença entre processo e *thread*?
  - Processo: é uma instância independente de uma aplicação que executa com escalonamento feito pelo SO, além de seu próprio contexto, memória virtual e recursos alocados
  - *Thread:* só existe como parte de um processo e seu escalonamento pode ser feito pelo programador, compartilhando os mesmos recursos do processo



Introdução

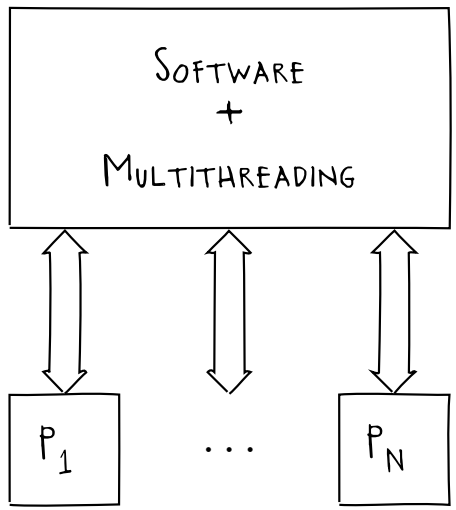
- Desempenho em software
  - Multiprogramação
    - Permite a execução concorrente de múltiplos processos durante um mesmo período de tempo
    - Em plataformas multiprocessadas, os processos podem ser executar paralelamente em cada processador

Introdução

- Desempenho em software
  - Multiprogramação
    - Permite a execução concorrente de múltiplos processos durante um mesmo período de tempo
    - Em plataformas multiprocessadas, os processos podem ser executar paralelamente em cada processador
  - *Multithreading*
    - Cria um ambiente de execução concorrente dentro do processo para maximizar o uso dos recursos
    - Um processo com *multithreading* pode ser paralelizado entre os núcleos de processamento

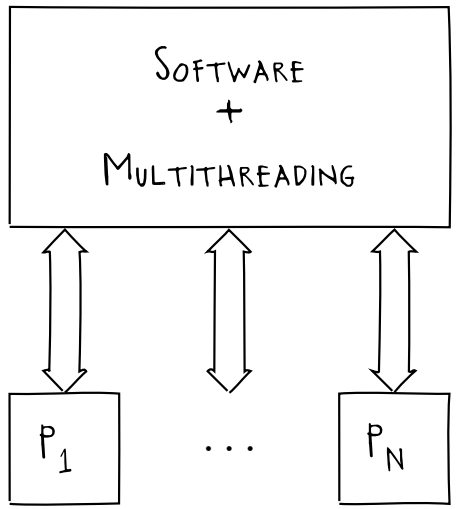
Introdução

- Qual é o limite de aumento do desempenho?
  - Hardware $\times$ Software



Introdução

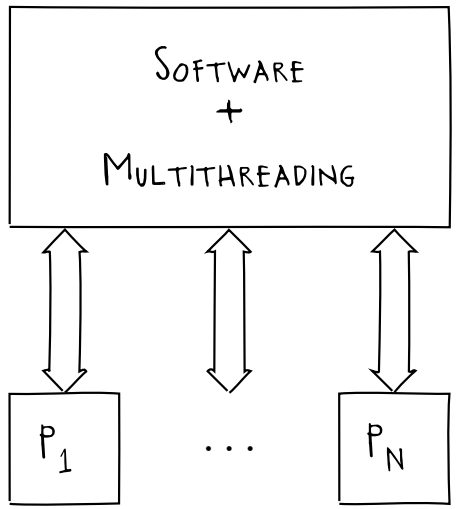
- Qual é o limite de aumento do desempenho?
  - Hardware $\times$ Software



$$\uparrow \textit{Processadores} \xrightarrow{?} \uparrow \textit{Desempenho}$$

Introdução

- Qual é o limite de aumento do desempenho?
  - Hardware $\times$ Software



$$\uparrow \textit{Threads} \xrightarrow{?} \uparrow \textit{Desempenho}$$

Introdução

- Lei de Amdahl
  - A melhoria de desempenho está limitada a parte S do software que é sequencial e não a parte $P = 1 - S$ que pode ser paralelizada em N processadores

$$Amdahl(N) = \frac{Uniprocessador}{Multiprocessador}$$

Introdução

- Lei de Amdahl
  - A melhoria de desempenho está limitada a parte S do software que é sequencial e não a parte $P = 1 - S$ que pode ser paralelizada em N processadores

$$
\begin{aligned}
Amdahl(N) &= \frac{Uniprocessador}{Multiprocessador} \\
&= \frac{S + P}{S + \frac{P}{N}}
\end{aligned}
$$

Introdução

- Lei de Amdahl
  - A melhoria de desempenho está limitada a parte S do software que é sequencial e não a parte $P = 1 - S$ que pode ser paralelizada em N processadores

$$
\begin{aligned}
Amdahl(N) &= \frac{Uniprocessador}{Multiprocessador} \\
&= \frac{S + P}{S + \frac{P}{N}} \\
&= \frac{S + (1 - S)}{S + \frac{P}{N}} \\
&= \frac{1}{S + \frac{P}{N}}
\end{aligned}
$$

Introdução

- Lei de Amdahl
  - Na análise de código de um software, foi detectado que 1% de seu fluxo de execução é sequencial
  - Para a execução do software podem ser utilizados um número infinito de unidades de processamento

$$\lim_{N \to \infty} Amdahl(N) = \frac{1}{S + \frac{P}{N}}$$

Introdução

- Lei de Amdahl
  - Na análise de código de um software, foi detectado que 1% de seu fluxo de execução é sequencial
  - Para a execução do software podem ser utilizados um número infinito de unidades de processamento

$$\begin{aligned}\lim_{N\to\infty} Amdahl(N) &= \frac{1}{S + \frac{P}{N}} \\ &= \frac{1}{0,01 + \frac{0,99}{N}} \\ &= \frac{1}{0,01 + \frac{0,99}{N} \to 0}\end{aligned}$$

Introdução

- Lei de Amdahl
  - Na análise de código de um software, foi detectado que 1% de seu fluxo de execução é sequencial
  - Para a execução do software podem ser utilizados um número infinito de unidades de processamento

$$\begin{aligned}
\lim_{N \to \infty} Amdahl(N) &= \frac{1}{S + \frac{P}{N}} \\
&= \frac{1}{0,01 + \frac{0,99}{N}} \\
&= \frac{1}{0,01 + \frac{0,99}{N} \to 0} \\
&= 100
\end{aligned}$$

Avaliação de desempenho

- Definição de desempenho de sistemas
  - Os desempenhos dos sistemas A e B dependem de seus tempos de execução para realizar as operações X

$$Desempenho_A = \frac{X}{Tempo\ de\ execução_A}$$

$$Desempenho_B = \frac{X}{Tempo\ de\ execução_B}$$

Avaliação de desempenho

- Definição de desempenho de sistemas
  - Os desempenhos dos sistemas A e B dependem de seus tempos de execução para realizar as operações X

$$Desempenho_A > Desempenho_B$$

Avaliação de desempenho

- Definição de desempenho de sistemas
  - Os desempenhos dos sistemas A e B dependem de seus tempos de execução para realizar as operações X

$$\begin{array}{ccc} Desempenho_A & > & Desempenho_B \\ \frac{X}{Tempo\ de\ execução_A} & > & \frac{X}{Tempo\ de\ execução_B} \\ Tempo\ de\ execução_B & > & Tempo\ de\ execução_A \end{array}$$

Avaliação absoluta de desempenho

Avaliação de desempenho

- Definição de desempenho de sistemas
  - Os desempenhos dos sistemas A e B dependem de seus tempos de execução para realizar as operações X

$$n = \frac{Desempenho_A}{Desempenho_B}$$

Avaliação de desempenho

- Definição de desempenho de sistemas
  - Os desempenhos dos sistemas A e B dependem de seus tempos de execução para realizar as operações X

$$
\begin{aligned}
n &= \frac{Desempenho_A}{Desempenho_B} \\
&= \frac{\frac{X}{Tempo\ de\ execução_A}}{\frac{X}{Tempo\ de\ execução_B}} \\
&= \frac{Tempo\ de\ execução_B}{Tempo\ de\ execução_A}
\end{aligned}
$$

Avaliação relativa de desempenho

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Para medir o desempenho de um sistema é preciso obter o tempo de processamento (*CPU time*) para executar uma determinada aplicação *X*

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Para medir o desempenho de um sistema é preciso obter o tempo de processamento (*CPU time*) para executar uma determinada aplicação *X*
  - Este é o tempo gasto na execução das operações pelo processador, desconsiderando o período de espera por escalonamento ou operações de E/S

$$\begin{aligned} Tempo\ de\ execução_X &= \#Ciclos_X \times Periodo\ de\ relógio \\ &= \frac{\#Ciclos_X}{Frequência\ de\ relógio} \end{aligned}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Em uma aplicação X com tempo de execução de 8 segundos em um processador com frequência de 5 GHz, quantos ciclos são consumidos?

$$Tempo_X = \frac{\#Ciclos_X}{Frequência}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Em uma aplicação X com tempo de execução de 8 segundos em um processador com frequência de 5 GHz, quantos ciclos são consumidos?

$$\begin{aligned} Tempo_X &= \frac{\#Ciclos_X}{Frequência} \\ 8 &= \frac{\#Ciclos_X}{5 \times 10^9} \end{aligned}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Em uma aplicação X com tempo de execução de 8 segundos em um processador com frequência de 5 GHz, quantos ciclos são consumidos?

$$
\begin{aligned}
Tempo_X &= \frac{\#Ciclos_X}{Frequência} \\
8 &= \frac{\#Ciclos_X}{5 \times 10^9} \\
&\downarrow \\
\#Ciclos_X &= 4 \times 10^{10}\ ciclos
\end{aligned}
$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Para reduzir o tempo de execução da aplicação X em 25%, qual deveria ser a frequência de operação do processador?

$$Tempo_Y = Tempo_X \times 0,75$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Para reduzir o tempo de execução da aplicação X em 25%, qual deveria ser a frequência de operação do processador?

$$\begin{aligned} Tempo_Y &= Tempo_X \times 0,75 \\ 6 &= \frac{4 \times 10^{10}}{Frequência} \end{aligned}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Para reduzir o tempo de execução da aplicação X em 25%, qual deveria ser a frequência de operação do processador?

$$
\begin{aligned}
Tempo_Y &= Tempo_X \times 0,75 \\
6 &= \frac{4 \times 10^{10}}{Frequência} \\
&\downarrow \\
Frequência &= \frac{4 \times 10^{10}}{6} \\
&\approx 6,66\ GHz
\end{aligned}
$$

$$\uparrow Frequência \longleftrightarrow \uparrow Desempenho$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Cada instrução da aplicação X possui um tempo médio de ciclos durante sua execução (CPI)

$$\#Ciclos_X = \#Instruções_X \times CPI_X$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Cada instrução da aplicação X possui um tempo médio de ciclos durante sua execução (CPI)

$$\begin{aligned} \#Ciclos_X &= \#Instruções_X \times CPI_X \\ &\downarrow \\ Tempo_X &= \frac{\#Instruções_X \times CPI_X}{Frequência} \end{aligned}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Cada instrução da aplicação X possui um tempo médio de ciclos durante sua execução (CPI)

$$\begin{aligned} \#Ciclos_X &= \#Instruções_X \times CPI_X \\ &\downarrow \\ Tempo_X &= \frac{\#Instruções_X \times CPI_X}{Frequência} \end{aligned}$$

$$\downarrow CPI_X \longleftrightarrow \uparrow Desempenho$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - A taxa de execução de instruções para uma determinada aplicação *X* define o desempenho em termos do número de operações que podem ser executadas por segundo

$$Taxa\ de\ execução_X = \frac{\#Instruções_X}{Tempo_X}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - A taxa de execução de instruções para uma determinada aplicação X define o desempenho em termos do número de operações que podem ser executadas por segundo

$$
\begin{aligned}
Taxa\ de\ execução_X &= \frac{\#Instruções_X}{Tempo_X} \\
&= \frac{\#Instruções_X}{\frac{\#Instruções_X \times CPI_X}{Frequência}}
\end{aligned}
$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - A taxa de execução de instruções para uma determinada aplicação *X* define o desempenho em termos do número de operações que podem ser executadas por segundo

$$
\begin{aligned}
Taxa\ de\ execução_X &= \frac{\#Instruções_X}{Tempo_X} \\
&= \frac{\#Instruções_X}{\frac{\#Instruções_X \times CPI_X}{Frequência}} \\
&= \frac{Frequência}{CPI_X}\ instruções/s
\end{aligned}
$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - A taxa de execução de instruções para uma determinada aplicação *X* define o desempenho em termos do número de operações que podem ser executadas por segundo

$$
\begin{aligned}
Taxa\ de\ execução_X &= \frac{\#Instruções_X}{Tempo_X} \\
&= \frac{\#Instruções_X}{\frac{\#Instruções_X \times CPI_X}{Frequência}} \\
&= \frac{Frequência}{CPI_X}\ instruções/s
\end{aligned}
$$

Esta taxa depende da frequência de operação
e do CPI médio das instruções do processador

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Executando uma aplicação X em um processador com frequência de 5 GHz e CPI médio de 0,25

$$Taxa\ de\ execução_X = \frac{Frequência}{CPI_X}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Executando uma aplicação X em um processador com frequência de 5 GHz e CPI médio de 0,25

$$\begin{aligned} Taxa\ de\ execução_X &= \frac{Frequência}{CPI_X} \\ &= \frac{5 \times 10^9}{0,25} \end{aligned}$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Executando uma aplicação X em um processador com frequência de 5 GHz e CPI médio de 0,25

$$
\begin{aligned}
Taxa\ de\ execução_X &= \frac{Frequência}{CPI_X} \\
&= \frac{5 \times 10^9}{0,25} \\
&= 20 \times 10^9\ instruções/s \\
&= 20\ GIPS
\end{aligned}
$$

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Uma métrica alternativa para avaliar o desempenho é a contabilização do número de operações de ponto flutuante realizadas pelo processador

Avaliação de desempenho

- Medição de desempenho de sistemas
  - Uma métrica alternativa para avaliar o desempenho é a contabilização do número de operações de ponto flutuante realizadas pelo processador
  - Geralmente é descrita em milhões de operações de ponto flutuante por segundo ou *Millions of Floating-point Operations Per Second* (MFLOPS)

$$MFLOPS = \frac{\#Operações\ de\ ponto\ flutuante}{Tempo\ de\ execução} \times 10^{-6}$$

Avaliação de desempenho

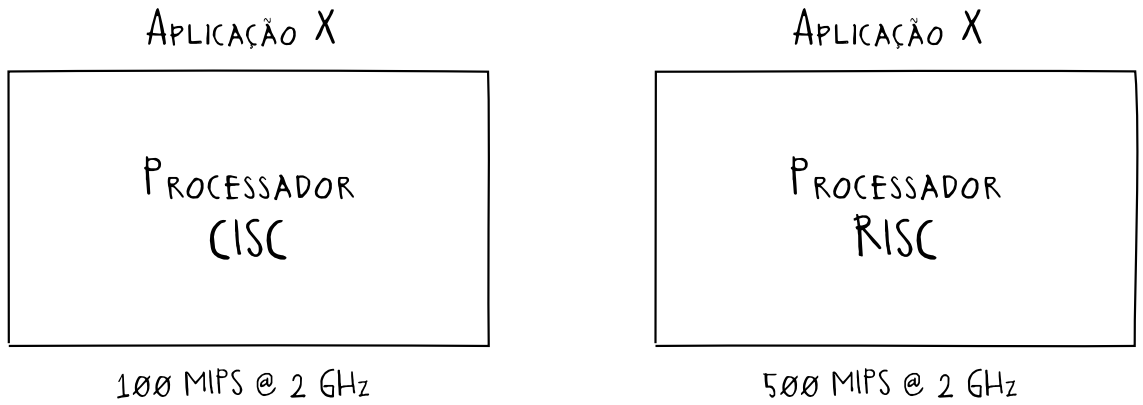
- Medição de desempenho de sistemas
  - Uma métrica alternativa para avaliar o desempenho é a contabilização do número de operações de ponto flutuante realizadas pelo processador
  - Geralmente é descrita em milhões de operações de ponto flutuante por segundo ou *Millions of Floating-point Operations Per Second* (MFLOPS)

$$MFLOPS = \frac{\#Operações\ de\ ponto\ flutuante}{Tempo\ de\ execução} \times 10^{-6}$$

É comumente utilizado na avaliação de desempenho de aplicações científicas, de jogos e de supercomputadores

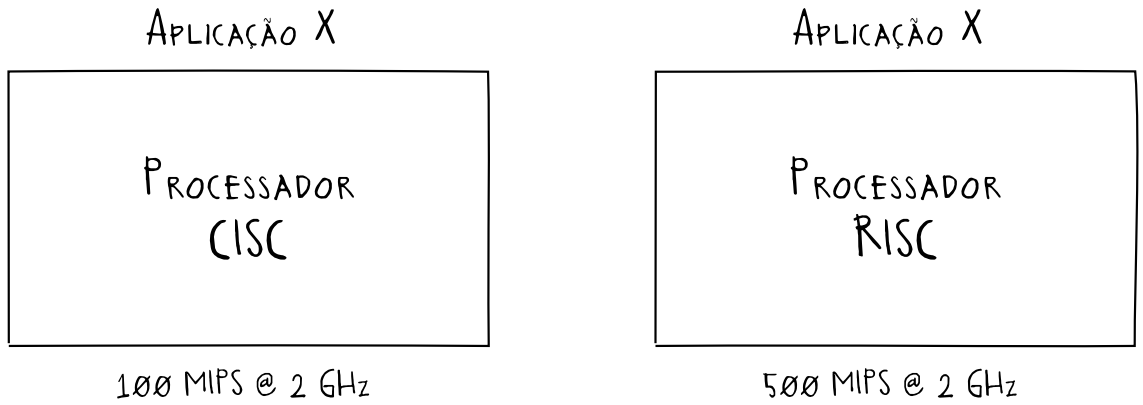
Avaliação de desempenho

- Como avaliar o desempenho de processadores com arquiteturas e repertório de instruções diferentes?



Avaliação de desempenho

- Como avaliar o desempenho de processadores com arquiteturas e repertório de instruções diferentes?



$$Desempenho_{CISC} \overset{?}{<} Desempenho_{RISC}$$

Avaliação de desempenho

- Realização da operação $A = A + 1$ com todos os dados armazenados na memória principal

```
1 add [A], 1
```

CISC

```
1 la t0, A
2 lw t1, 0(t0)
3 addi t1, t1, 1
4 sw t1, 0(t0)
```

RISC

Avaliação de desempenho

- Realização da operação $A = A + 1$ com todos os dados armazenados na memória principal

```
1 add [A], 1
```

CISC

```
1 la t0, A
2 lw t1, 0(t0)
3 addi t1, t1, 1
4 sw t1, 0(t0)
```

RISC

Avaliando o desempenho com a métrica MIPS, o processador RISC pareceria 5 vezes mais rápido que o CISC, entretanto, ambos realizam as mesmas operações no mesmo tempo

Avaliação de desempenho

- A indústria e a academia resolveram desenvolver um conjunto de aplicações de referência (*benchmarks*) para avaliar o desempenho dos sistemas, sem dependências tecnológicas de instruções ou de frequência de operação

Avaliação de desempenho

- A indústria e a academia resolveram desenvolver um conjunto de aplicações de referência (*benchmarks*) para avaliar o desempenho dos sistemas, sem dependências tecnológicas de instruções ou de frequência de operação
  - Estas aplicações são escritas em linguagens de programação de alto nível, como C ou C++, podendo ser compiladas para diferentes arquiteturas

Avaliação de desempenho

- A indústria e a academia resolveram desenvolver um conjunto de aplicações de referência (*benchmarks*) para avaliar o desempenho dos sistemas, sem dependências tecnológicas de instruções ou de frequência de operação
  - Estas aplicações são escritas em linguagens de programação de alto nível, como C ou C++, podendo ser compiladas para diferentes arquiteturas
  - Devem ser representativas para algum domínio da computação, como cálculo numérico, inteligência artificial ou processamento de imagem

Avaliação de desempenho

- A indústria e a academia resolveram desenvolver um conjunto de aplicações de referência (*benchmarks*) para avaliar o desempenho dos sistemas, sem dependências tecnológicas de instruções ou de frequência de operação
  - Estas aplicações são escritas em linguagens de programação de alto nível, como C ou C++, podendo ser compiladas para diferentes arquiteturas
  - Devem ser representativas para algum domínio da computação, como cálculo numérico, inteligência artificial ou processamento de imagem
  - É esperada uma distribuição ampla com código fonte aberto e resultados mensuráveis, para fins de comparação dos resultados gerados

Avaliação de desempenho

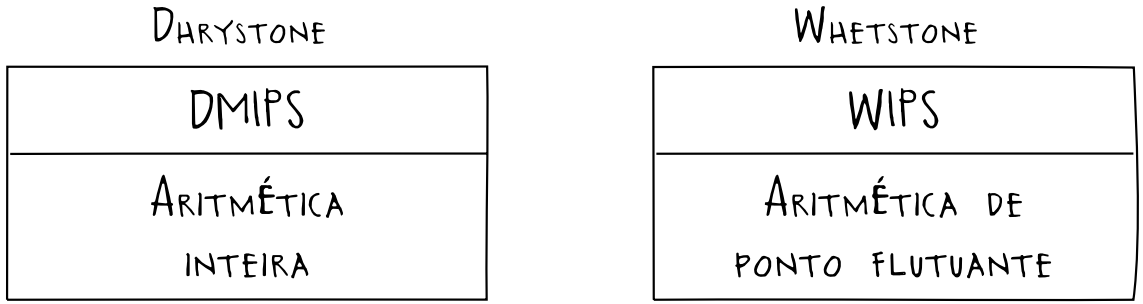
- Aplicações de referência (*benchmarks*)
  - Domínios $\times$ métricas de avaliação
    - Processamento de vídeo: taxa de quadros (fps)
    - Serviços: tempo de resposta ou latência (s)
    - Sistemas embarcados: potência consumida (Wh)
    - Telecomunicações e redes: taxa de transferência (bps)
    - ...

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Algoritmos sintéticos
    - São algoritmos criados para explorar um conjunto de operações que são utilizadas em diversas aplicações
    - As implementações são focadas em estabelecer estatísticas sobre as operações e os comportamentos

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Algoritmos sintéticos
    - São algoritmos criados para explorar um conjunto de operações que são utilizadas em diversas aplicações
    - As implementações são focadas em estabelecer estatísticas sobre as operações e os comportamentos



Os resultados gerados só permitem avaliar
o desempenho e não possuem nenhuma utilidade

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Dhrystone
    - Criado por Reinhold P. Weicker em 1984
    - Simples + Fácil utilização + Código aberto
    - A métrica fornece uma relação entre quantas iterações das funções e procedimentos foram realizadas durante o tempo de execução do software

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Dhrystone
    - Criado por Reinhold P. Weicker em 1984
    - Simples + Fácil utilização + Código aberto
    - A métrica fornece uma relação entre quantas iterações das funções e procedimentos foram realizadas durante o tempo de execução do software

$$1\ DMIPS = 1757\ iterações$$
$$\updownarrow$$
$$DEC\ VAX\ 11/780$$
$$@$$
$$1\ MIPS$$

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Whetstone
    - Criado pelo Laboratório Nacional de Física do Reino Unido em Whetstone no ano de 1972
    - Focado em avaliar o desempenho das operações aritméticas em ponto flutuante, através da métrica de *Whetstone Instructions Per Second* (WIPS)
    - Cerca de metade do seu tempo de execução é consumido executando funções de bibliotecas matemáticas

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Dhrystone $\times$ Whetstone

Arquitetura	DMIPS	MWIPS
Intel Core 2 E8400	12,41	23,29
AMD Phenom II X4 965	27,25	44,19
Intel Core i7 980X	72,08	108,56

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Deficiências dos algoritmos sintéticos
    - Como os algoritmos utilizados são sintéticos, as estruturas de códigos descritas não são usualmente utilizados em situações reais

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Deficiências dos algoritmos sintéticos
    - Como os algoritmos utilizados são sintéticos, as estruturas de códigos descritas não são usualmente utilizados em situações reais
    - Devido a sua concepção artificial e previsível, as técnicas de otimização dos compiladores são capazes de ampliar o desempenho obtido

Avaliação de desempenho

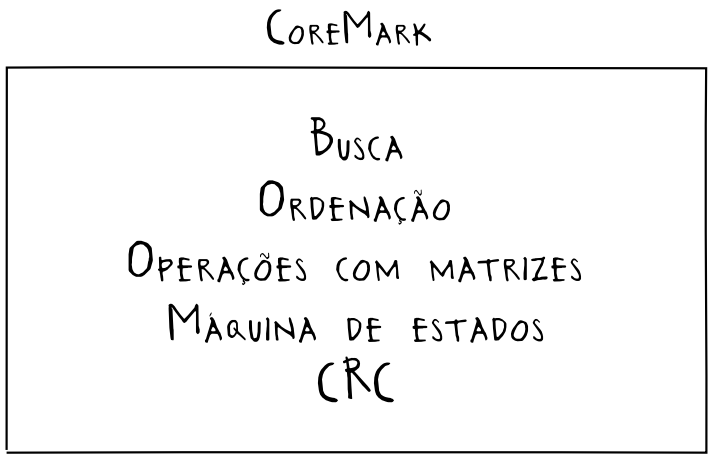
- Aplicações de referência (*benchmarks*)
  - Deficiências dos algoritmos sintéticos
    - Como os algoritmos utilizados são sintéticos, as estruturas de códigos descritas não são usualmente utilizados em situações reais
    - Devido a sua concepção artificial e previsível, as técnicas de otimização dos compiladores são capazes de ampliar o desempenho obtido
    - Por ser uma aplicação pequena, grande parte dos acessos realizados para buscar instruções e dados estarão disponíveis na cache

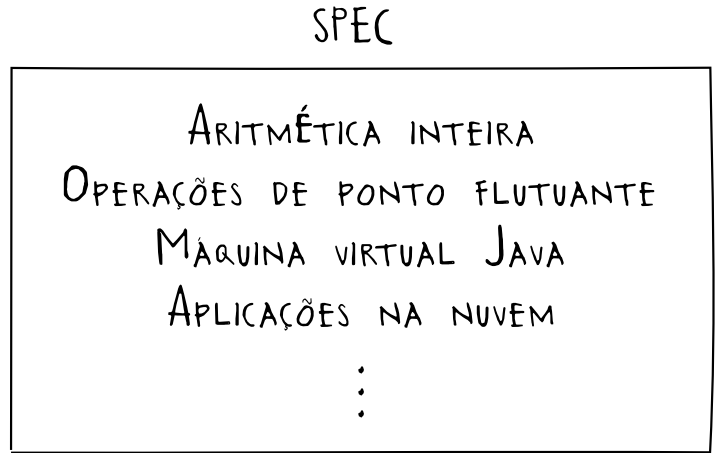
Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Algoritmos reais
    - São algoritmos baseados em problemas reais de diversas áreas, como multiplicação de matrizes, busca, processamento e ordenação de dados
    - O foco da implementação é incorporar os algoritmos em um contexto real de utilização

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - Algoritmos reais
    - São algoritmos baseados em problemas reais de diversas áreas, como multiplicação de matrizes, busca, processamento e ordenação de dados
    - O foco da implementação é incorporar os algoritmos em um contexto real de utilização





Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - CoreMark
    - Desenvolvido por Shay Gal-On da EEMBC em 2009
    - O objetivo principal de sua criação foi de transformar em um padrão para a indústria e substituir o Dhrystone
    - Combina as vantagens do Dhrystone com a utilização de algoritmos reais que fornecem uma métrica padronizada para avaliação de desempenho

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - CoreMark
    - Desenvolvido por Shay Gal-On da EEMBC em 2009
    - O objetivo principal de sua criação foi de transformar em um padrão para a indústria e substituir o Dhrystone
    - Combina as vantagens do Dhrystone com a utilização de algoritmos reais que fornecem uma métrica padronizada para avaliação de desempenho

1 *iteração dos algoritmos*

$\updownarrow$

1 *CoreMark*

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - *Standard Performance Evaluation Corporation* (SPEC)
    - É uma organização fundada em 1988 e tem como objetivo a criação a manutenção de um conjunto de aplicações de referência padronizadas
    - Possui diversas áreas de atuação, como sistemas de código aberto e computação de alto desempenho

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - *Standard Performance Evaluation Corporation* (SPEC)
    - É uma organização fundada em 1988 e tem como objetivo a criação a manutenção de um conjunto de aplicações de referência padronizadas
    - Possui diversas áreas de atuação, como sistemas de código aberto e computação de alto desempenho

$$Desempenho = \frac{Tempo_{Referência}}{Tempo_{Sistema}}$$

Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - *Standard Performance Evaluation Corporation* (SPEC)
    - É uma organização fundada em 1988 e tem como objetivo a criação a manutenção de um conjunto de aplicações de referência padronizadas
    - Possui diversas áreas de atuação, como sistemas de código aberto e computação de alto desempenho

$$
\begin{aligned}
Desempenho &= \frac{Tempo_{Referência}}{Tempo_{Sistema}} \\
&= \frac{Tempo_{Referência}}{\frac{1}{n}\sum_{i=1}^{n} Tempo_i}
\end{aligned}
$$

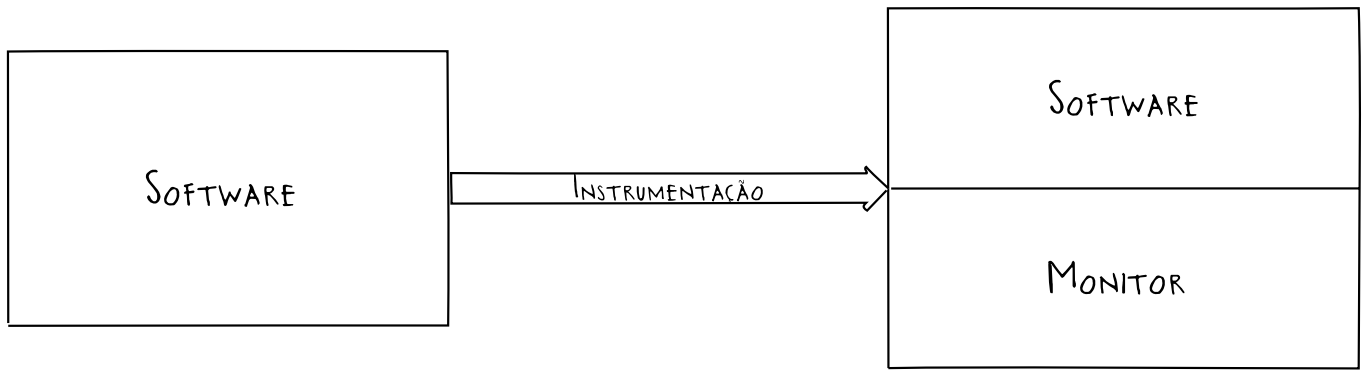
Avaliação de desempenho

- Aplicações de referência (*benchmarks*)
  - CoreMark $\times$ SPEC CPU 2006

Arquitetura	CoreMark	SPEC CPU 2006
Intel Core 2 E8400	20628,00	23,50
AMD Phenom II X4 910	24828,46	16,10
Intel Core i7-2600	99562,34	46,40

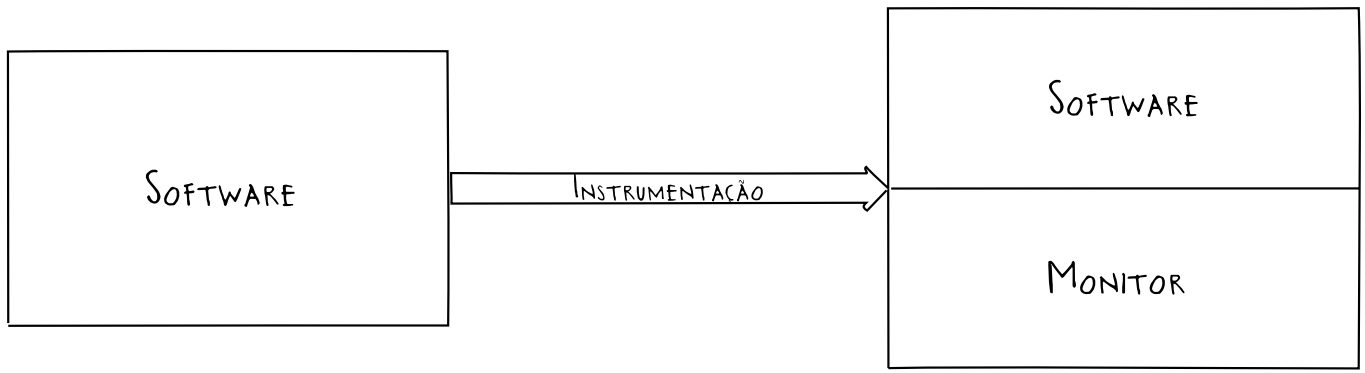
Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - É uma técnica de instrumentação do código fonte para análise das propriedades dinâmicas do software, como alocação de memória e tempo de execução das funções e dos procedimentos



Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - É uma técnica de instrumentação do código fonte para análise das propriedades dinâmicas do software, como alocação de memória e tempo de execução das funções e dos procedimentos



É feita a otimização do software com as estatísticas e informações obtidas

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)

```
1  // Tipos inteiros de tamanho fixo
2  #include <stdint.h>
…  ...
26 // Função principal
27 int main() {
28     // Variáveis auxiliares
29     uint64_t fat = 0, fib = 0;
30     // 10000 iterações
31     for(uint32_t i = 0; i < 10000; i++) {
32         // Chamadas de funções
33         fat = fat + fatorial(i);
34         fib = fib + fibonacci(i);
35     }
36     // Retorno sem erros
37     return 0;
38 }
```

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - GNU Profiler (gprof)

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - GNU Profiler (gprof)

```
$ gcc -Wall -g -pg exemplo.c -o exemplo.elf

$ time ./exemplo.elf
```

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - GNU Profiler (gprof)

```
$ gcc -Wall -g -pg exemplo.c -o exemplo.elf

$ time ./exemplo.elf

real 0m1.838s
user 0m1.838s
sys  0m0.000s
```

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - GNU Profiler (gprof)

```
$ gcc -Wall -g -pg exemplo.c -o exemplo.elf

$ time ./exemplo.elf

real 0m1.838s
user 0m1.838s
sys  0m0.000s

$ gprof exemplo.elf
```

Avaliação de desempenho

- Análise dinâmica de perfil (*software profiling*)
  - GNU Profiler (gprof)

```
$ gcc -Wall -g -pg exemplo.c -o exemplo.elf

$ time ./exemplo.elf

real 0m1.838s
user 0m1.838s
sys  0m0.000s

$ gprof exemplo.elf
Flat profile:


Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 55.10      0.33     0.33    10000    32.51    32.51  fatorial
 42.18      0.57     0.25    10000    24.89    24.89  fibonacci
  4.30      0.60     0.03                             frame_dummy
.
.
.
$
```