"Diga-me como me mede e te direi como me comporto [te engano]" (Goldratt, E.).

Métricas de Desempenho

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Métricas de desempenho

Temos dois computadores.

O computador *X* e o computador *Y*.

Como dizer qual é melhor?

Métricas de desempenho

Temos dois computadores.

O computador *X* e o computador *Y*.

Como dizer qual é melhor?

Podemos medir o tempo que X e Y demoram para completar uma tarefa. **Problemas?**

Métricas de desempenho

Temos dois computadores.

O computador *X* e o computador *Y*.

Como dizer qual é melhor?

Podemos medir o tempo que X e Y demoram para completar uma tarefa. **Problemas?**

Uma tarefa não representa todos os programas possíveis;

Computadores diferentes são especialistas (e mais rápidos) em diferentes tarefas;

Tempo não é a única métrica possível;

Quanta energia foi gasta?

Quantas tarefas conseguimos completar em um período de tempo (vazão ou throughput)?

...

Métricas de desempenho

Considere que vamos medir o tempo de execução de um conjunto de tarefas em X.

O tempo que podemos acompanhar com um cronômetro (ou relógio de parede) para completar a tarefa é chamado de ***wall-clock time***.

> O tempo que nós humanos vamos precisar esperar para completar a tarefa.
> Se um programa demorou um minuto para ser executado, ele ficou realmente executando por um minuto na CPU?

Métricas de desempenho

Considere que vamos medir o tempo de execução de um conjunto de tarefas em X.

O tempo que podemos acompanhar com um cronômetro (ou relógio de parede) para completar a tarefa é chamado de ***wall-clock time***.

O tempo que nós humanos vamos precisar esperar para completar a tarefa.

Mas os programas "param" esperando:

Entradas e Saídas;

O Sistema Operacional;

Outros programas que também competem por recursos;

...

O ***CPU time*** (tempo de CPU) considera apenas o tempo em que o programa ficou efetivamente sendo processado.

Wall-clock time versus CPU time

Então como medir o tempo?

Wall-clock time ou ***CPU time***?

Vai depender do seu problema.

CPU time parece mais justo;

Mas Wall-clock time é o que nós humanos efetivamente experienciamos.

Exemplo

O comando *time* no Linux mostra o tempo de execução de um programa.

Sintaxe: *time nomeProgramaMensurar*

O comando devolve

Real → O **Wall-clock** time do programa;
User → O **CPU time** do programa usado em espaço do **usuário**;
Sys → O C**PU time** do programa usado em espaço do **sistema operacional**.

Para obter o CPU time completo, some User+Sys.

Faça você mesmo

Abra um terminal e execute os comandos.

Mostrar quanto tempo o comando (programa) *ls* demora para executar.

time ls

Mostrar quanto tempo o comando (programa) *top* vai ficar em execução.

time top

Medindo o tempo de resposta

Após medir o tempo de execução para X e Y, podemos definir então que X é n vezes mais rápido que Y através de:

$$n = \frac{Tempo\ Exec_Y}{Tempo\ Exec_X}$$

Throughput

Podemos aplicar o mesmo raciocínio para o *throughput* (vazão).

Exemplo: o *throughput* de X é 2x maior do que o de Y se X completa 2x mais tarefas do que Y em determinado período de tempo.

Quando a vazão pode ser mais importante do que o tempo de execução?

Throughput

Podemos aplicar o mesmo raciocínio para o *throughput* (vazão).

Exemplo: o *throughput* de X é 2x maior do que o de Y se X completa 2x mais tarefas do que Y em determinado período de tempo.

Quando a vazão pode ser mais importante do que o tempo de execução?

Em servidores, por exemplo.

Desejamos completar o maior número de requisições em um determinado período.

Não nos importamos se, por exemplo, uma requisição individual para a carga de uma página Web vai demorar 0,1 ou 0,2 segundos.

Energia

Qual a importância da eficiência energética?

Energia

Qual a importância da eficiência energética?

Menos energia consumida = menor poluição.

Pense nas tartarugas!

Energia

Qual a importância da eficiência energética?

Menos energia consumida = menor poluição.

Pense nas tartarugas!

Em um celular/Notebook.

Duração da bateria, preço da bateria, custo energético, peso, refrigeração.

Em um servidor.

Custo energético, refrigeração.

Em um Desktop.

Refrigeração, custo energético.

...

Energia

O que é um Watt?

Energia

Potência (Watts) = 1 Joule por segundo.

Faz sentido medir a potência para o design do sistema de refrigeração.

- Comumente os fabricantes chamam de TDP (Thermal Design Power).
- Acesse em ark.intel.com e veja o TDP de alguns de seus processadores

TDP é o “consumo sustentado de potência”.

- Geralmente é maior do que a potência consumida média.
- Geralmente é 1,5x menor do que a potência máxima que pode ser consumida.

Energia

Potência (Watts) = 1 Joule por segundo.

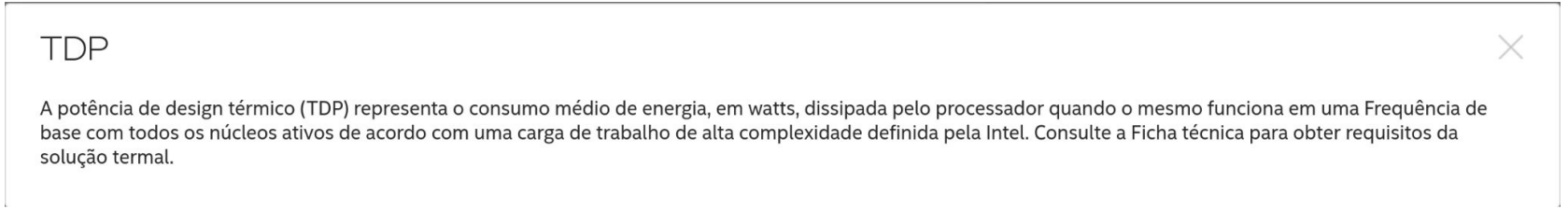
Faz sentido medir a potência para o design do sistema de refrigeração.

- Comumente os fabricantes chamam de TDP (Thermal Design Power).
- Acesse em ark.intel.com e veja o TDP de alguns de seus processadores

TDP é o "consumo sustentado de potência".

- Geralmente é maior do que a potência consumida média.
- Geralmente é 1,5x menor do que a potência máxima que pode ser consumida.

No final das contas, o TDP é uma carga de trabalho definida pelo fabricante, que obviamente não deixa claro que carga é essa.



Energia

Potência (Watts) = 1 Joule por segundo.

Potência não é uma boa métrica para eficiência energética.

Por quê?

Energia

Exemplo:

Se X consome 20% mais potência (W) que Y, mas completa a tarefa em 70% do tempo que Y, temos que:

*Consumo = 1,2 * 0,7 = 0,84*

Lembrando que 1 Watt = 1 Joule por Segundo.

X consome mais Joules por Segundo, mas precisa de menos segundos para completar a tarefa.

Devemos comparar então a energia (Joules) gasta para as tarefas, e não a potência (Watts).

A potência pode nos levar a resultados incorretos ou contra intuitivos.

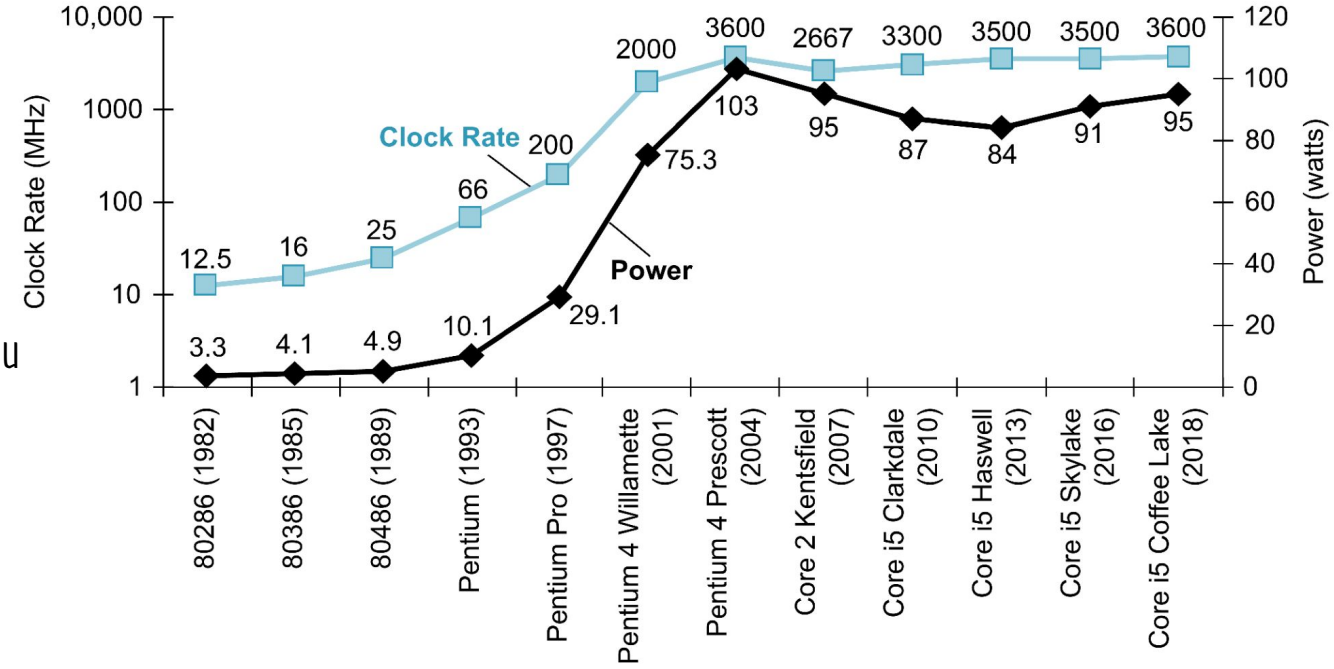
Veja detalhes em: HENNESSY, J; PATTERSON, D. Computer Architecture: A Quantitative Approach. Elsevier, 2019.

Mais Sobre Energia

Nossas CPUs ficaram sedentas por energia com o tempo.

Sedentas a ponto da energia ser um fator limitante do desempenho.

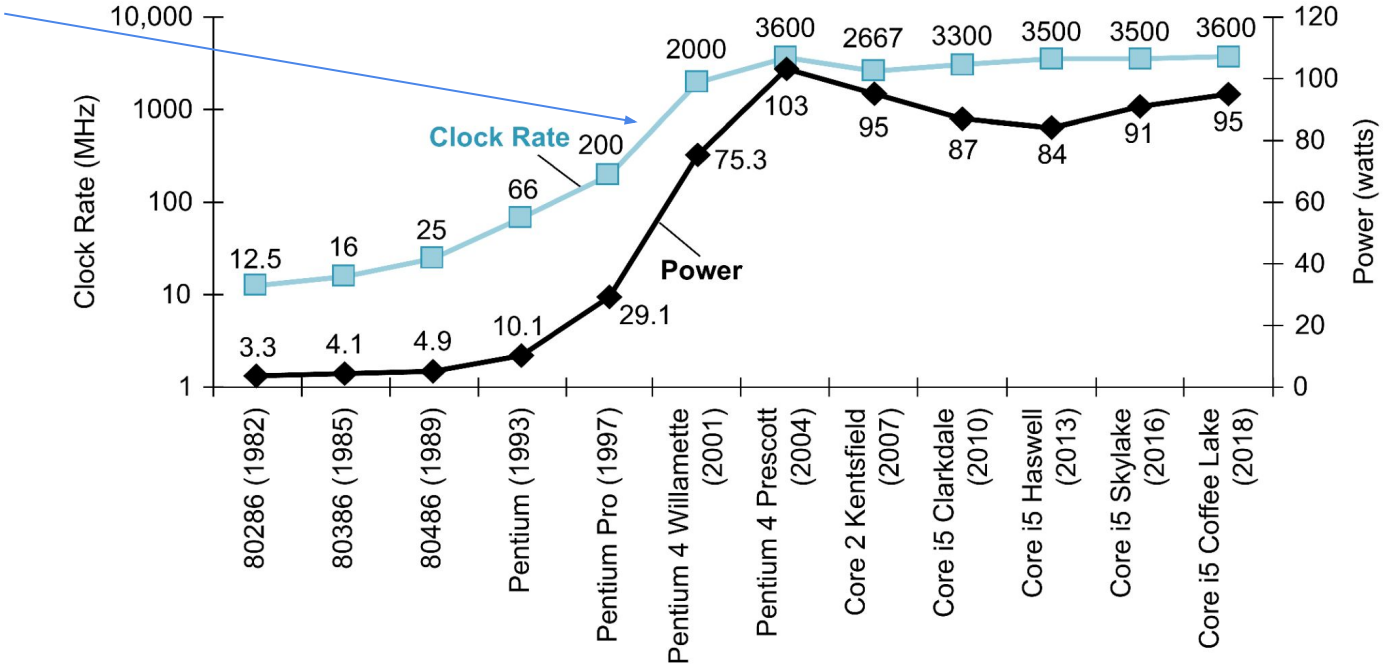
O problema é tão sério que ganhou um nome: **Power Wall**.



Patterson, Hennessy (2020).

Mais Sobre Energia

O aumento do consumo aumenta com a frequência!



Patterson, Hennessy (2020).

Mais Sobre Energia

A maioria dos processadores são feitos com portas lógicas CMOS e suas variantes.

Uma porta CMOS vaza (*leak*) uma quantidade muito pequena de energia quando estática (mantida em 0 ou 1).

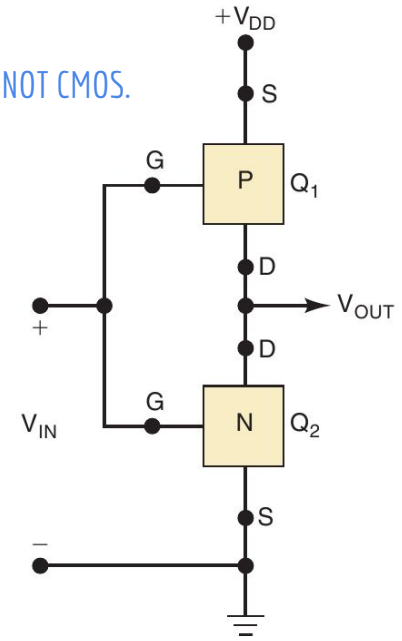
Quando trocamos de estado, temos dois problemas principais.

Troca de Estados

Durante uma troca de estados, ambos os transistores (P e N) podem ficar semi abertos por um curto período de tempo.

Caminho quase direto entre Vdd e o Terra.

Porta NOT CMOS.



Capacitância Parasita

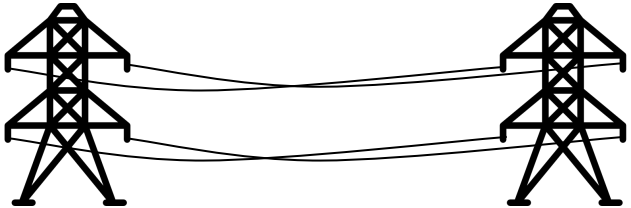
Todo circuito elétrico acumula carga como um capacitor.

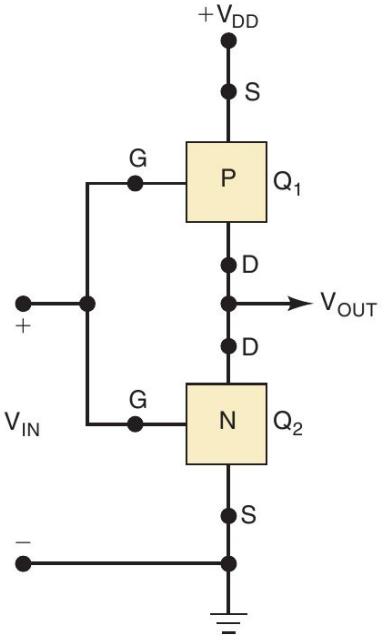
Capacitância parasita.

Quando a porta lógica troca de estado, precisamos dissipar a energia desse capacitor.

Custa tempo e energia.

Gera calor.



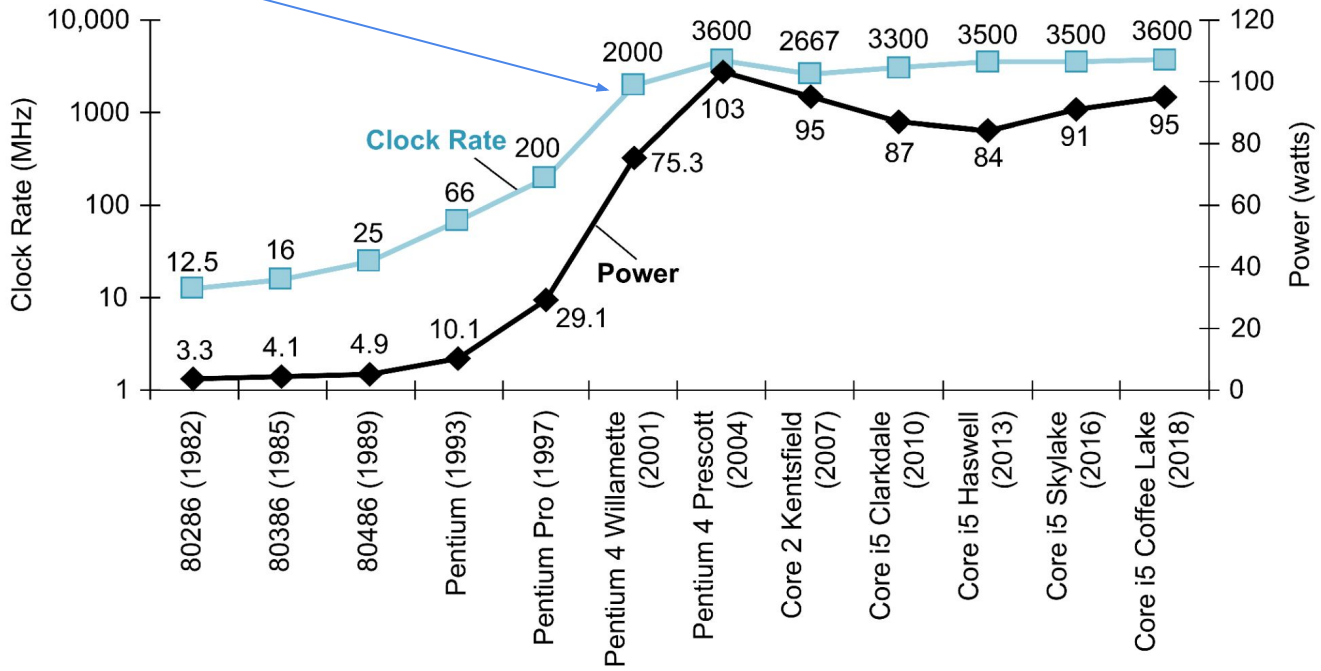


Mais Sobre Energia

$$\text{Energia} \propto \text{Carga Capacitiva} \times \text{Tensão}^2 \times \text{Frequência de Trocas}$$

Mais Sobre Energia

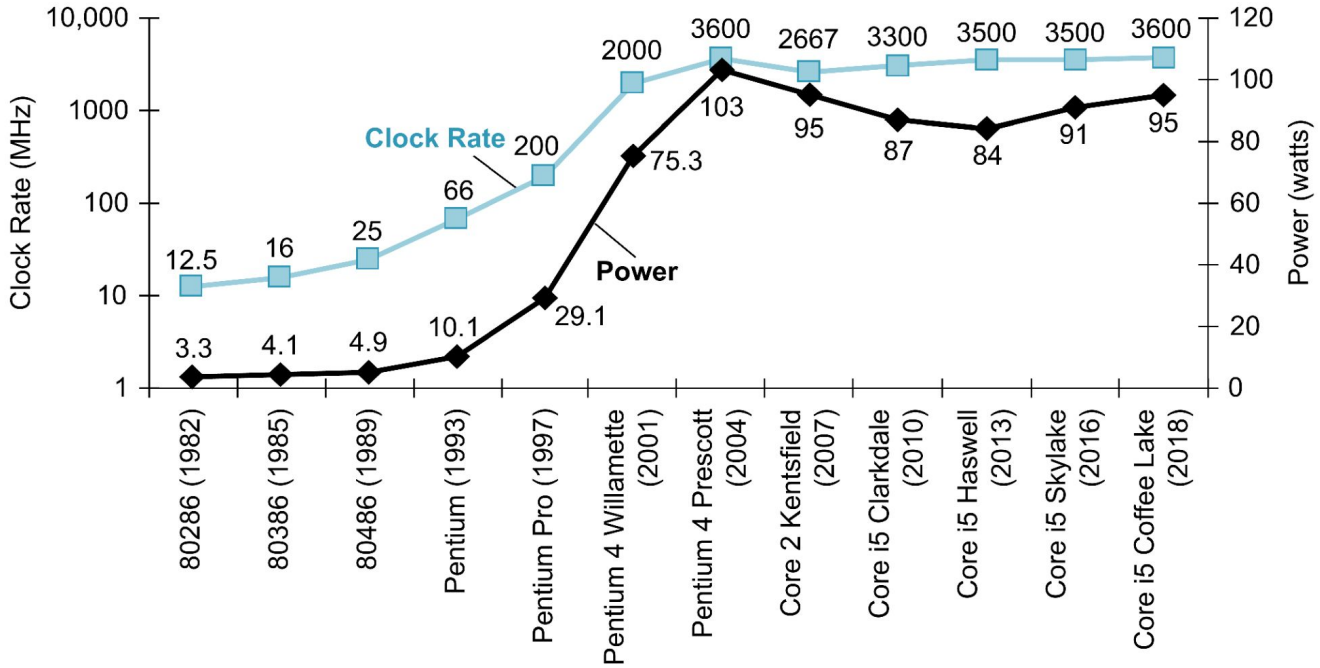
Energia $\propto$ Carga Capacitiva $\times$ Tensão^2 $\times$ Frequência de Trocas



Mais Sobre Energia

Energia $\propto$ Carga Capacitiva × Tensão^2 × Frequência de Trocas

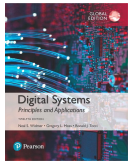
Cada nova geração tenta diminuir a carga capacitiva e a tensão para compensar pela frequência.



Mais Sobre Energia

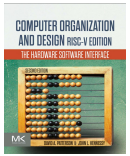
Leia mais sobre Power Wall e capacitâncias parasitas em:

Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Hennessy, Patterson. Arquitetura de Computadores: uma abordagem quantitativa. 2019.

Patterson, Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2020.



Faça você mesmo

Instale o stress-ng na sua máquina.

```
sudo apt-get install stress-ng
```

Execute em um terminal o comando `watch -n 1 lscpu` para verificar a frequência corrente da sua CPU.

Em outro terminado, execute o seguinte comando para “estressar” uma única CPU do seu sistema por 1 minuto.

```
stress-ng --cpu 1 -t 1m
```

Depois faça o mesmo teste, mas estressando todas as CPUS.

```
stress-ng --cpu 0 -t 1m
```

Faça você mesmo

Instale o stress-ng na sua máquina.

```
sudo apt-get install stress-ng
```

Execute em um terminal o comando `watch -n 1 lscpu` para verificar a frequência corrente da sua CPU.

Em outro terminado, execute o seguinte comando para "estressar" uma única CPU do seu sistema por 1 minuto.

```
stress-ng --cpu 1 -t 1m
```

Depois faça o mesmo teste, mas estressando todas as CPUS.

```
stress-ng --cpu 0 -t 1m
```

O clock pode ter diminuído quando você acionou todas as CPUs por questões de **Energia**.

- O sistema não consegue **arrefecer** todas as CPUs ao mesmo tempo.
- O sistema não consegue **alimentar** todas as CPUs ao mesmo tempo.

Faça você mesmo

Execute um programa qualquer com o Likwid, e meça a energia gasta para executar o programa na CPU.

No Likwid, os medidores de energia geralmente estão em um grupo ENERGY. Para listar os medidores, use *likwid-perfctr -a*

Exemplo:

likwid-perfctr -C 0 -g ENERGY NOME_PROGRAMA

Carga de trabalho

Para medir o desempenho do computador (utilizando alguma métrica) precisamos de uma carga de trabalho.

Programas para testar.

Carga de trabalho

Mas qual(is) programa(s) executar?

Encontrar esses programas pode ser simples se temos:

- Usuários bem comportados que usam sempre o mesmo conjunto de programas.
- Computadores criados para tarefas específicas.

Mas se desejamos um conjunto representativo “universal” de tarefas para medir o desempenho, as coisas se complicam.

Carga de trabalho

Benchmarks são conjuntos de programas que possuem uma carga de trabalho considerada “típica”.

São comumente utilizados para dizer se o computador *X* é melhor do que *Y*.

Alguns exemplos de benchmarks.

- Desempenho de processadores utilizando um Benchmark proprietário: www.cpubenchmark.net
- Benchmarks SPEC: www.spec.org
- Benchmarks TPC: www.tpc.org

Benchmarks

Problemas de um benchmark?

Benchmarks

Problemas de um benchmark:

- O que é uma carga de trabalho “típica”.
  - Pior, o que é “típico” para um usuário, não é para outro.
- Engenheiros podem criar sistemas que são otimizados para os benchmarks, mas não necessariamente otimizados para o uso real do sistema.
  - Diga-me como me medes e eu te direi como te engano.

Clock

Processadores utilizam sinais de clock para sincronismo.

O sinal pode ser definido pelo seu período:

Ex.: 1 ns

ou frequência (F):

Ex.: 1 GHz

Lembrando que F = 1/T.

Então temos duas novas formas de obter o CPU time:

CPU time = Número de ciclos utilizados pelo programa * período.

CPU time = Número de ciclos utilizados pelo programa / frequência.

Faça você mesmo

Considere um processador que utiliza um clock de 4GHz.

1. Qual o período do clock?
2. Considere que um programa utilizou 12×10^9 ciclos para ser executado na CPU. Qual o CPU time?

Contando instruções

Podemos contar quantas instruções foram executadas.

Instruction Count – **IC**.

Se também sabemos quantos ciclos de clock foram necessários, podemos facilmente obter o número médio de ciclos de clock necessários para cada instrução.

CPI: Clock Cicles per Instruction.

CPI = número total de ciclos de clock / total de instruções.

Faça você mesmo

Execute um programa qualquer com o Likwid, e meça o CPI.

No Likwid, os medidores de CPI geralmente estão em um grupo CLOCK.

Exemplo:

likwid-perfctr -C 0 -g CLOCK NOME_PROGRAMA

Contando instruções

Com o CPI, podemos calcular o CPU time como:

CPU time = total instruções * CPI * Período de clock.

Contando instruções

Com o CPI, podemos calcular o CPU time como:

CPU time = total instruções * CPI * Período de clock.

O CPU time de um programa pode ser melhorado por vários fatores distintos e diretamente dependentes da arquitetura da CPU. **Quais?**

Contando instruções

Com o CPI, podemos calcular o CPU time como:

CPU time = total instruções * CPI * Período de clock.

O CPU time de um programa pode ser melhorado por vários fatores distintos e diretamente dependentes da arquitetura da CPU. **Quais?**

A frequência de clock;

O formato das instruções.

Formatos diferentes podem requerem mais ou menos instruções para executar uma tarefa.

O número médio de ciclos de clock necessários para se executar uma instrução.

Uma CPU que sofre com muitos stalls vai aumentar o CPU time.

Exercício

1. Considere que você vai comparar duas CPUs, onde as informações das CPUs são dadas na tabela a seguir. Considere ainda que o CPI, e o número médio de instruções (quando o programa é compilado para a arquitetura da CPU A ou B) foram extraídos utilizando um benchmark. Baseado nessas informações, calcule o CPU time e indique qual CPU é melhor.

	CPU A	**CPU B**
Frequência (GHz)	4	5
CPI	2	3
Número médio de instruções em um programa.	$26x10^8$	$22x10^8$

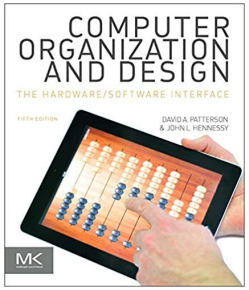
Exercício

2. Considere o mesmo exercício anterior. Mas agora considere que o Fabricante da CPU A reportou que sua CPU consome 105 Watts quando usada em plena capacidade, enquanto a CPU B consome 97 Watts. Qual CPU é mais energeticamente eficiente considerando a carga de trabalho da tabela?

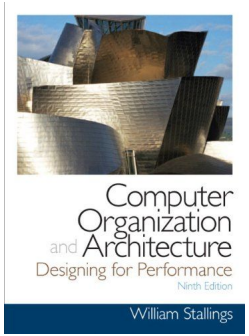
	CPU A	**CPU B**
Frequência (GHz)	4	5
CPI	2	3
Número médio de instruções em um programa.	$26x10^8$	$22x10^8$

Referências

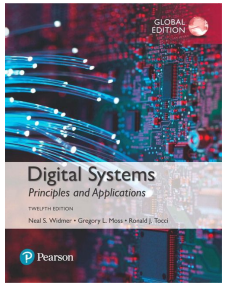
Patterson, Hennessy . Arquitetura e Organização de Computadores: A interface hardware/software. 2014.



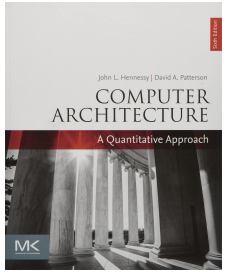
Stallings, W. Organização de Arquitetura de Computadores. 10a Ed. 2016.



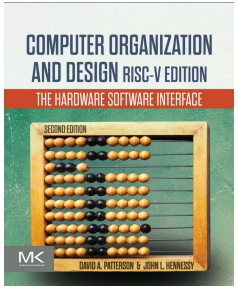
Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Hennessy, Patterson. Arquitetura de Computadores: uma abordagem quantitativa. 2019.



Patterson, Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2020.



Licença