Aula 15

- Avaliação de desempenho de sistemas computacionais
  - Definições e medição de desempenho
  - Desempenho do CPU
  - Lei de Amdahl
  - Métodos de avaliação do desempenho
  - Os *benchmarks* SPEC
  - Desempenho, potência, eficiência energética e custo

Bernardo Cunha, José Luís Azevedo, Arnaldo Oliveira, Tomás Oliveira e Silva

Introdução

- O que significa um computador apresentar melhor desempenho que outro?
  - Qual a métrica usada?
- Analogia com modelos de aviões. Possíveis métricas
  - Capacidade, alcance, velocidade ou débito?

Modelo do avião	Capacidade (passageiros)	Alcance (Km)	Vel. Cruzeiro (Km/h)	Débito (passageiros x Km/h)
Boeing 747	470	6700	980	460.600
Concorde	132	6500	2170	286.440
Douglas DC-8	146	14000	875	127.750

O que é o desempenho?

- O desempenho de um sistema pode definir-se relativamente a diversos factores:
  - Tempo de resposta
  - Número de operações executadas num dado intervalo de tempo
  - Consumo de energia (eficiência energética)
  - ...
- Nos computadores o **tempo de execução** dos programas está, normalmente, associado à definição de desempenho:
  - Quanto menor for o tempo de execução, melhor será o desempenho
- Para diferentes tipos de sistemas poderão, no entanto, ser apropriadas métricas de quantificação de desempenho distintas

O que é o desempenho? - exemplos

- Num ***data center*** que aceita pedidos de transacção de vários utilizadores
  - Um gestor de um *data center* está interessado em aumentar o débito (*throughput*) de processamento das transacções (trabalho total realizado num dado intervalo de tempo)
  - O computador mais rápido é aquele que concluir **mais transacções por unidade de tempo**
- Num sistema embutido (***embedded system***) com restrições de tempo-real, a resposta a certos eventos deve/tem de ocorrer dentro de um intervalo de tempo limitado (pré-definido)
  - Se o tempo de resposta estiver garantido, então os *designers* poderão aumentar o *throughput* ou reduzir o custo
- Conclui-se novamente que, dependendo do sistema, podem ser usadas diferentes métricas para a avaliação do desempenho

Qual a complexidade na caracterização do desempenho?

- Dificuldade da determinação do desempenho resultante da complexidade dos sistemas:
  - *Hardware* – os computadores utilizam, cada vez mais, técnicas complexas (**não determinísticas**) de melhoria de desempenho
  - *Software* – complexidade (e dimensão) dos programas
- Um computador é formado por vários blocos
  - Cada um deles afecta o desempenho global de modo distinto
  - Cada fabricante tem as suas próprias soluções de implementação
- Dependendo da complexidade do sistema pode ser impossível determinar o desempenho de forma analítica

Qual a complexidade na caracterização do desempenho?

- O desempenho global de um sistema é afectado quer pelo *hardware,* quer pelo *software*
- *Hardware*:
  - **Microprocessador** – é o elemento mais rápido do sistema: a melhoria do desempenho é obtida pelo aumento da frequência de relógio e pela minimização do número médio de ciclos de relógio necessários para a execução das instruções
  - **Memória** – mais lenta que o microprocessador: a melhoria do desempenho centra-se na minimização do tempo médio de acesso para leitura e escrita
  - **Periféricos** – podem realizar operações muito distintas: têm grande impacto no desempenho global, sobretudo em programas que os usam de forma intensiva
- *Software*:
  - **Compilador** – a qualidade do código gerado tem impacto no desempenho global: sequências de instruções usadas, utilização de registos internos, ...

Definição de desempenho baseada no **tempo de execução**

- O tempo é a medida mais fiável de avaliação de desempenho dos computadores
  - O computador que realizar uma dada quantidade de trabalho em menos tempo é o mais rápido
- Tempo de execução
  - Tempo entre o início e o fim da execução de uma tarefa
- Se um mesmo programa for executado em dois computadores distintos (em igualdade de circunstâncias)
  - O computador que terminar a sua execução em menos tempo é o mais rápido

$$Desempenho_X = \frac{1}{T_{Execução_X}}$$

$$Desempenho_X > Desempenho_Y$$

$$\frac{1}{T_{Execução_X}} > \frac{1}{T_{Execução_Y}}$$

$$T_{Execução_X} < T_{Execução_Y}$$

$$\frac{Desempenho_X}{Desempenho_Y} = \frac{T_{Execução_Y}}{T_{Execução_X}} = n$$

Métrica MIPS (Milions of Instructions Per Second)

- Medida da velocidade de execução do programa baseada no número de instruções executadas por unidade de tempo

$$\text{MIPS} = \frac{\#\,\text{Instruções Executadas}}{\text{Tempo de Execução} \cdot 10^6}$$

- Problemas
  - Não toma em consideração o tipo e a complexidade das instruções
  - Não permite a comparação de computadores com diferentes ISA uma vez que o número total de instruções para a mesma aplicação será diferente
  - Varia entre programas que executam no mesmo computador
  - Tem um comportamento anómalo em certas situações: MIPS pode variar inversamente com o desempenho – ver exemplo 4
  - Os fabricantes divulgam esta métrica assumindo que o processador executa repetidamente a instrução mais rápida (normalmente a instrução “nop”)

Parâmetros temporais para a medição do desempenho

- **Response time** (latency ou execution time ou elapsed time)
  - O tempo que decorre entre o início e o fim de uma dada tarefa
  - Tempo de execução no CPU incluindo acessos à memória, acesso a unidades de Input/Output (I/O), carga do sistema operativo – tudo
  - Pode ser impossível de avaliar analiticamente
- **CPU time** (ou CPU execution time)
  - O tempo que o CPU demora a executar uma dada tarefa, em situação "ideal". Não inclui:
    - ✓O tempo em que o programa está suspenso à espera que uma unidade de I/O (periférico) esteja pronta para transferir informação
    - ✓O tempo extra necessário para aceder à memória
    - ✓O tempo gasto a executar outros programas
  - O *CPU time* pode ser medido executando o programa

Desempenho do CPU (baseado no *CPU execution time*)

$$Desempenho_{CPU} = \frac{1}{Tempo_{CPU}}$$

$$Tempo_{CPU} = Ciclos_{CPU} \times T_{ciclo}^{CPU} = \frac{Ciclos_{CPU}}{Frequência_{CPU}}$$

Exemplo 1

Considere um computador com um CPU a funcionar com uma frequência de 4 GHZ. O *CPU Execution Time* medido na execução de um dado programa foi de 10s.

Pretende-se desenvolver um novo CPU que execute o mesmo programa em 6s. O *hardware designer* verificou que é possível um aumento da frequência de trabalho do CPU, mas isso acarreta um acréscimo do número total de ciclos de relógio de 1,2 vezes relativamente ao existente.

Qual a frequência de trabalho que deverá ter o novo CPU?

Desempenho do CPU (baseado no CPI)

- **CPI - Ciclos por Instrução:** número médio de ciclos de relógio que cada instrução de um programa demora a executar. Uma vez que instruções diferentes podem apresentar diferentes tempos de execução, o CPI é uma média de todas as instruções executadas no programa.

$$Ciclos_{CPU} = \#Instruções \times CPI \qquad Tempo_{CPU} = \frac{\#Instruções \times CPI}{Frequência_{CPU}} \quad (2)$$

- A expressão (2) evidencia os três aspectos-chave que afectam o desempenho. Como obter estes valores?
  - A frequência do CPU - é conhecida
  - O número de instruções - pode ser obtido por *profiling* (obtenção, durante a execução, de estatísticas de execução do programa) ou através de um simulador da arquitectura
  - O CPI - pode ser obtido por simulação ou através de contadores *hardware* (quando o CPU já está operacional)

Desempenho do CPU – Exemplo 2

- Considerem-se duas máquinas com implementações distintas da mesma arquitectura do conjunto de instruções (ISA). Para um dado programa,
  - Máquina A: Clock_cycle = 350 ps; CPI = 2,0
  - Máquina B: Clock_cycle = 400 ps; CPI = 1,5

- Qual a máquina mais rápida? Qual a relação de desempenho?

Desempenho do CPU – Exemplo 3

- O projectista de um compilador para uma dada máquina está a tentar decidir entre duas sequências de código para a tradução de uma instrução de alto nível.
  - Do conhecimento da implementação *hardware* do processador sabe-se que há 3 classes de instruções A, B e C cuja execução requer 1, 2 e 3 ciclos de relógio, respectivamente.
  - A primeira sequência de código tem 10 instruções: 5 do tipo A, 2 do tipo B e 3 do tipo C.
  - A segunda sequência de código tem 12 instruções: 8 do tipo A, 3 do tipo B e 1 do tipo C.
- Q1: Qual das duas sequências será mais rápida? Qual o factor?
- Q2: Qual o CPI de cada sequência?

Desempenho do CPU – Exemplo 4

- Dois compiladores diferentes estão a ser testados numa máquina com frequência de CPU de 2 GHz. Estão disponíveis 3 classes de instruções A, B e C cuja execução requer 1, 2 e 3 ciclos de relógio, respectivamente.
- Os dois compiladores são utilizados para produzir código para um dado programa :
  - O 1º compilador gera código que contém: 5 milhões de instruções da classe A, 1 milhão da classe B e 1 milhão da classe C
  - O 2º compilador gera código que contém: 10 milhões de instruções da classe A, 1 milhão da classe B e 1 milhão da classe C
- Q1: Qual das duas sequências produzidas será mais rápida de acordo com a métrica MIPS?
- Q2: Qual será mais rápida de acordo com a métrica “tempo de execução”?

Lei de Amdahl

- Quantifica a melhoria que se pode esperar no tempo de execução de um programa num dado computador, se se melhorar de N vezes um dos factores que determina esse tempo
- Considerando que o tempo de execução, antes da optimização, corresponde a:

$$T_{Exec} = T_{Exec}^{Afectado} + T_{Exec}^{NãoAfectado}$$

- O tempo de execução após a optimização fica:

$$T_{Exec}^{Melhorado} = \frac{T_{Exec}^{Afectado}}{N} + T_{Exec}^{NãoAfectado}$$

Lei de Amdahl - exemplo

- Se o tempo de execução do programa for dado por:
  - **$T_{EXEC} = T_{CPU} + T_{MEM} + T_{PERIF}$**
- Considere-se que: T_{CPU} = 1s, T_{MEM} = 4s, T_{PERIF} = 5s (T_{EXEC} = 10s)
- Se se passar a frequência do CPU para o dobro, então T_{CPU} será metade e o tempo de execução melhorado será:
  - **$T_{EXEC\_MELHORADO} = (T_{CPU} / 2) + T_{MEM} + T_{PERIF} = (1 / 2) + 4 + 5 = 9.5s$**
- O aumento para o dobro da frequência de relógio resultou, assim, numa melhoria no desempenho de:
  - **$Melhoria\_desempenho = T_{EXEC} / T_{EXEC\_MELHORADO} = 10 / 9.5 \cong 1,05$ (5%)**
- A melhoria global é inferior à melhoria de apenas 1 dos factores
- O impacto da melhoria é tanto maior quanto maior for o peso do factor objecto de optimização no tempo total de execução:
  - Só interessa optimizar o(s) caso(s) mais frequente(s)

Lei de Amdahl – Exemplo 5

- Considere um programa que executa em 100 s numa dada máquina, sendo que a unidade de multiplicação é responsável por 80 s desse tempo.
- Q1: Qual o factor de melhoria que é necessário introduzir na unidade de multiplicação para que a execução do programa seja 4 vezes mais rápida?
- Q2: Se se pretender uma melhoria do desempenho global de 5 vezes, qual o factor de melhoria a introduzir na unidade de multiplicação?

Métodos usados na avaliação de desempenho de ArqCs

- O que faz sentido, para o utilizador final, é o conhecimento do desempenho do computador como um todo, isto é, a executar programas reais, em cenário de utilização real
- A metodologia usada consiste na definição de programas de avaliação normalizados - ***benchmarks***
  - Cada fabricante pode testar os seus computadores de acordo com as normas do *benchmark* usado, e divulgar os resultados
  - Estes resultados permitem comparação entre diferentes computadores
- Métodos usados
  - *Benchmarks* sintéticos
  - *Workloads*
  - *Benchmarks* baseados em aplicações reais

Avaliação do desempenho - ***Benckmarks sintéticos***

- Whetstone e Dhrystone
- Pequenos programas com sequências de instruções escolhidas com base na frequência estatística de ocorrência dessas instruções em programas reais
- Muito usadas no passado (e em fases iniciais de desenvolvimento de uma nova arquitectura)
- Problemas:
  - Estas sequências de código podem não ser representativas das características dos programas que efectivamente vão ser executados
  - Encorajam o recurso, por parte dos fabricantes, a optimizações especializadas (não gerais) dos compiladores e das arquitecturas de modo a obter bons resultados com estes *benchmarks* (as sequências de instruções são conhecidas)

Avaliação do Desempenho

- ***Workload***
  - Conjunto representativo dos programas (e frequências relativas) que vão de facto executar no computador
  - A avaliação é efectuada com base no tempo de execução da *workload*
- ***Benchmarks***
  - Um *benchmark* é composto por um conjunto de programas reais de uma dada classe de aplicações
    - ✓Computação de alto desempenho
    - ✓Gráficos
    - ✓Servidores, ...
  - O *benchmark* destina-se a comparar o desempenho dos vários sistemas dos computadores nessas classes de aplicações

Os *benchmarks* **SPEC** (www.spec.org)

- **S**ystem **P**erformance **E**valuation **C**orporation
  - Fundada em 1989 por fabricantes de computadores
  - Disponibiliza *benchmarks* standard baseados em aplicações reais para avaliação de computadores pessoais e servidores
  - Define as regras de execução das aplicações e apresentação dos resultados
- Classes de *benchmarks*
  - CPU (SPEC CPU2006)
  - Gráficos, Computação de alto desempenho
  - Aplicações Java
  - Servidores de e-mail e Web
  - Sistemas de ficheiros
  - Potência
  - etc.

Desempenho, potência, eficiência energética e custo

- O consumo de potência é, cada vez mais, uma limitação chave no desempenho dos processadores, especialmente em sistemas embutidos e equipamentos portáteis
  - Alimentação a baterias e sistemas de dissipação passiva
  - **O consumo de potência é um aspecto tão importante como o desempenho ou o custo**
  - **Técnicas para redução do consumo:** redução da frequência (o consumo é proporcional à frequência), redução da tensão de alimentação, suspensão (parcial) do funcionamento

Desempenho, potência, eficiência energética e custo

- O desempenho é avaliado em diferentes modos de funcionamento
  - Potência máxima – alto desempenho
  - Potência mínima – maximização da autonomia
  - Potência intermédia – redução do consumo (e do desempenho)
- Em aplicações com limitações de consumo de potência, a eficiência energética, é provavelmente a métrica mais importante:
  - **Eficiência Energética = Desempenho / Energia consumida**

Conclusão

- Projecto realista de um sistema computacional deve ter em conta
  - Desempenho e a funcionalidade requeridas pelas aplicações alvo
  - O consumo de potência
  - O custo
- Tipos de computadores com diferentes restrições e compromissos
  - Computadores de alto desempenho e servidores da gama alta
  - Computadores de secretária e servidores da gama baixa
  - Equipamentos portáteis e sistemas embutidos (*embedded*)
- O único método fiável de medir e reportar o desempenho é usar como métrica o tempo de execução de aplicações reais
- Os três factores da expressão em conjunto é que determinam o desempenho

$$Tempo_{CPU} = \#Instruções \times CPI \times \frac{1}{Frequência_{CPU}}$$