Hierarquia de Memória

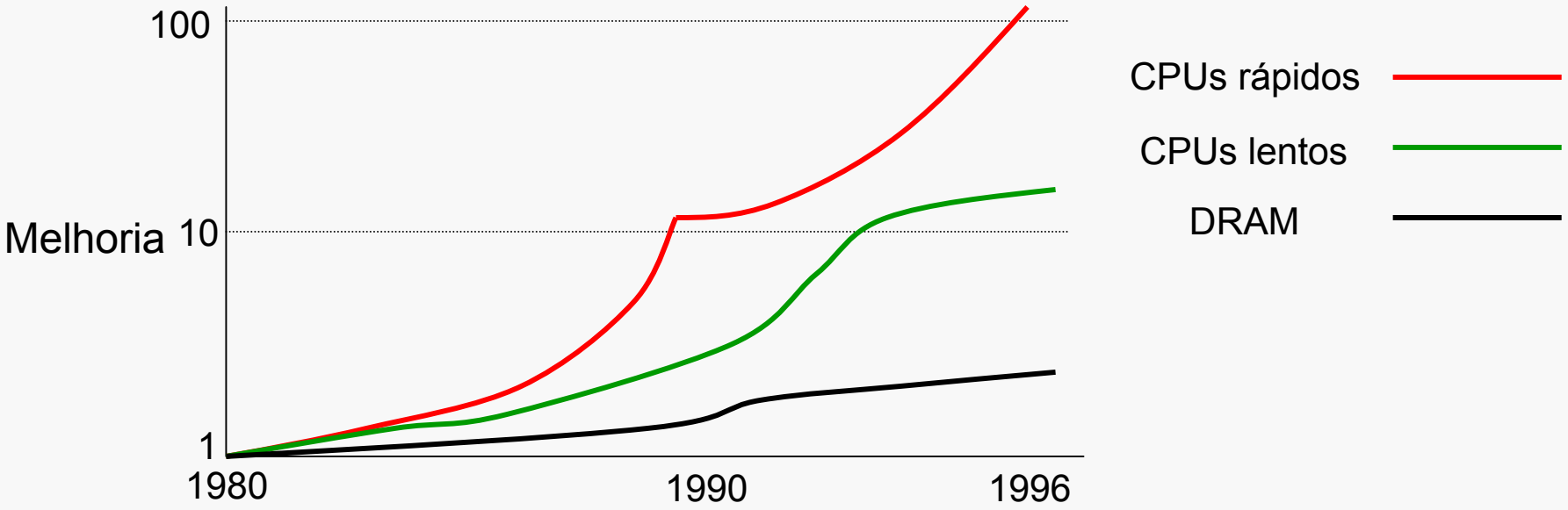
Introdução e Análise do Desempenho

Hierarquia de Memória

A velocidade dos processadores tem aumentado muito mais rapidamente do que a velocidade das memórias (DRAM).

$$f_{CPU} = 1.0 .. 3.0 \text{ GHz} \Rightarrow Tcc = 1.0 .. 0.33 \text{ ns}$$

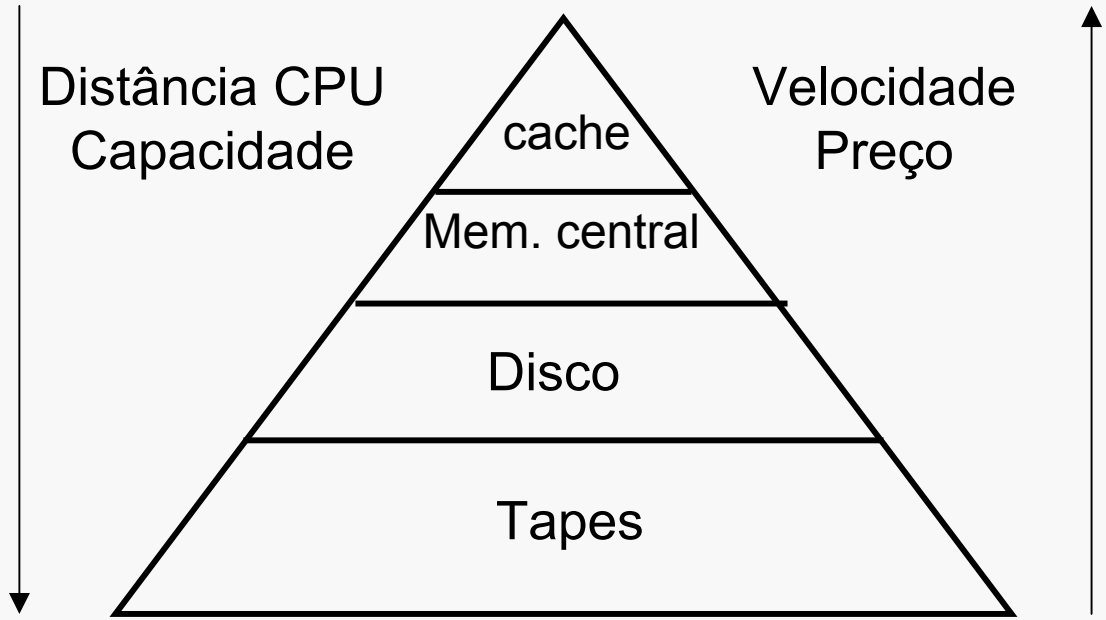
O tempo de acesso à DRAM ronda os 5 .. 60 ns



Como manter o processador alimentado com dados e instruções?

Hierarquia de Memória

Solução: Dotar a máquina de vários níveis de memória, tão mais rápidos (e mais caros e com menor capacidade) quanto mais perto se encontram do processador.
Cada nível contêm uma cópia do código e dados mais usados em cada instante.



SRAM – Static RAM
Extremamente rápida e cara

DRAM – Dynamic RAM
Mais lenta e barata

Localidade

É o **princípio da localidade**, exibido pela maior parte dos programas no acesso à memória, que permite acelerar os acessos à mesma com a hierarquia

O **princípio da localidade** divide-se em 2 componentes:

- **Localidade temporal**
- **Localidade espacial**

Localidade Temporal

Localidade Temporal – um elemento de memória acedido pelo CPU será, com grande probabilidade, acedido de novo num futuro próximo.

Exemplos: tanto as instruções dentro dos ciclos, como as variáveis usadas como contadores de ciclos, são acedidas repetidamente em curtos intervalos de tempo.

Consequência – a 1ª vez que um elemento de memória é acedido deve ser lido do nível mais baixo (por exemplo, da memória central).

Mas da 2ª vez que é acedido existem grandes hipóteses que se encontre na *cache*, evitando-se o tempo de leitura da memória central.

Localidade Espacial

Localidade Espacial – se um elemento de memória é acedido pelo CPU, então elementos com endereços na proximidade serão, com grande probabilidade, acedidos num futuro próximo.

Exemplos: as instruções são acedidas em sequência, assim como, na maior parte dos programas os elementos dos *arrays*.

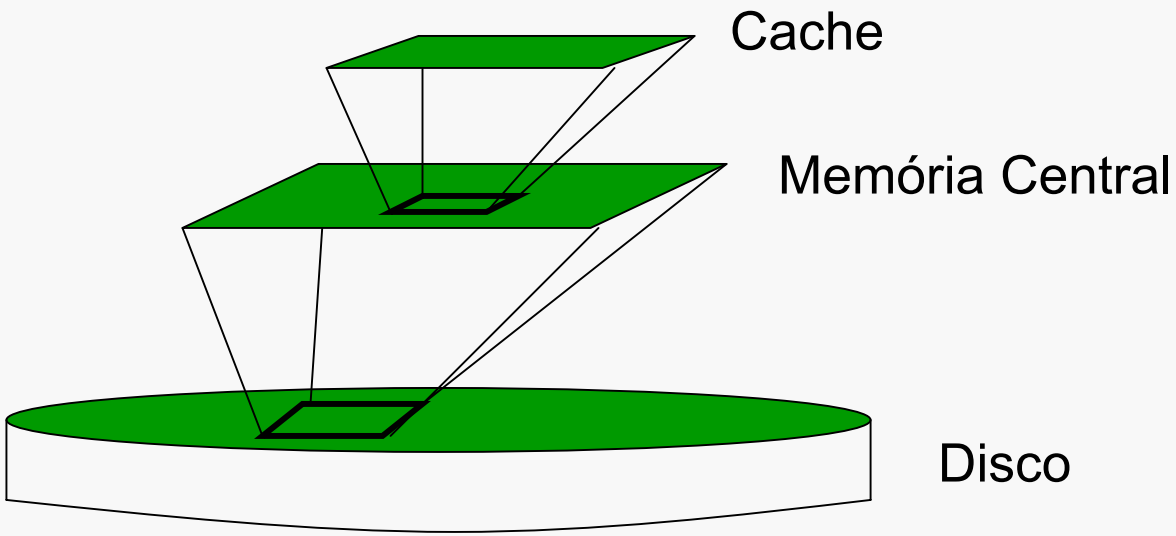
Consequência – a 1ª vez que um elemento de memória é acedido, deve ser lido do nível mais baixo (por exemplo, memória central) não apenas esse elemento, mas sim um **bloco** de elementos com endereços na sua vizinhança.

Se o processador, nos próximos ciclos, acede a um endereço vizinho do anterior (ex.: próxima instrução ou próximo elemento de um *array*) aumenta a probabilidade de esta estar na *cache*.

Inclusão

Os dados contidos num nível mais próximo do processador são sempre um sub-conjunto dos dados contidos no nível anterior.

O nível mais baixo contem a totalidade dos dados.



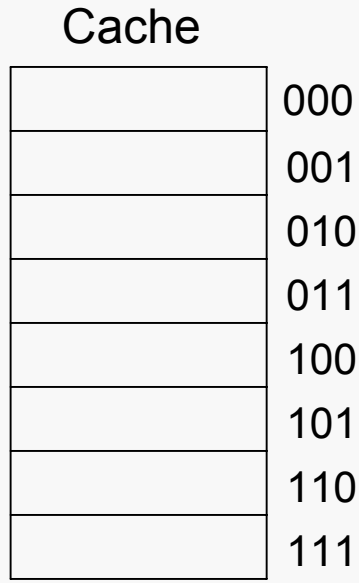
Terminologia (*cache* – memória central)

Linha – a cache está dividida em linhas. Cada linha tem o seu endereço (índice) e tem a capacidade de um bloco

Bloco – Quantidade de informação que é transferida de cada vez da memória central para a cache. É igual à capacidade da linha.

***Hit* –** Diz-se que ocorreu um *hit* quando o elemento de memória acedido pelo CPU se encontra na cache.

***Miss* –** Diz-se que ocorreu um *miss* quando o elemento de memória acedido pelo CPU não se encontra na cache, sendo necessário lê-lo da memória central.



Terminologia (*cache* – memória central)

Hit rate – Percentagem de *hits* ocorridos relativamente ao total de acessos à memória.

Miss rate – Percentagem de *misses* ocorridos relativamente ao total de acessos à memória. *Miss rate = (1 – hit rate)*

Hit time – Tempo necessário para aceder à cache, incluindo o tempo necessário para determinar se o elemento a que o CPU está a aceder se encontra ou não na cache.

Miss penalty – Tempo necessário para carregar um bloco da memória central para a cache quando ocorre um *miss.*

Hierarquia da memória - Desempenho

$$T_{exec} = \#I * CPI * T_{cc}$$

Como é que a hierarquia de memória influencia Texec?

#I – O número de instruções a executar depende do algoritmo, do conjunto de instruções e do compilador.

Tcc – é fixo para cada máquina. Não pode ser alterado modificando a organização da memória.

Hierarquia da Memória – Desempenho

$$T_{exec} = \#I * CPI * T_{cc}$$

$$CPI = CPI_{CPU} + CPI_{MEM}$$

CPI_{CPU} – nº de ciclos que o processador necessita, em média, para executar cada instrução;
O *hit time* considera-se incluído no CPI_{CPU}

CPI_{MEM} – nº de ciclos que o processador pára, em média, à espera de dados da memória central, por que não encontrou estes dados na cache. Estes são vulgarmente designados por ***memory stall cycles*** ou ***wait states***.

$$T_{exec} = \#I * (CPI_{CPU} + CPI_{MEM}) * T_{cc}$$

Hierarquia da Memória – Desempenho

$$CPI_{MEM} = \%acessosMem * missrate * misspenalty$$

Os acessos à memória devem-se ao *fetch* de instruções e ao acesso a dados. Como estes têm comportamentos diferentes usam-se diferentes percentagens de acesso à memória e miss rate para os dois casos.

Instruções – Todas as instruções são lidas da memória, logo a % de acesso à memória é de 100%. ***missrate*$_I$** refere-se ao acesso às instruções. Esta é geralmente menor que a dos dados devido à localidade espacial.

Dados – Apenas uma determinada percentagem de instruções acede à memória (%Mem). **missrate$_D$** refere-se ao acesso a dados.

$$CPI_{MEM} = (missrate_I + \%Mem * missrate_D) * misspenalty$$

Hierarquia da Memória – Desempenho

Abreviando *missrate* por *mr* e *misspenalty* por *mp* temos

$$T_{exec} = \#I * (CPI_{CPU} + CPI_{MEM}) * T_{cc}$$

$$CPI_{MEM} = (mr_I + \%Mem * mr_D) * mp$$

substituindo

$$T_{exec} = \#I * [CPI_{CPU} + (mr_I + \%Mem * mr_D) * mp] * T_{cc}$$

NOTA: A *miss penalty (mp)* tem que ser expressa em ciclos do *clock*.

Hierarquia da Memória – Desempenho

Considere uma máquina com uma *miss rate* de 4% para instruções, 5% para dados e uma *miss penalty* de 10 ciclos. Assuma ainda que 40% das instruções são *loads* ou *stores*, e que o CPI_{CPU} é 2. Qual o CPI total?

$$CPI = CPI_{CPU} + CPI_{MEM} = CPI_{CPU} + (mr_I + \%Mem * mr_D) * mp$$

$$CPI = 2 + (0.04 + 0.4 * 0.05) * 10 = 2 + 0.6 = 2.6$$

Se a frequência do relógio for de 800 MHz e o programa executar 10^9 instruções qual o tempo de execução?

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 2.6 * \frac{1}{800 * 10^6} = 3.25s$$

Hierarquia da Memória - Desempenho

Considere um programa com as características apresentadas na tabela, a executar numa máquina com memória de tempo de acesso 0. Se a frequência do processador for 1 GHz, qual o CPI médio e o tempo de execução?

Instrução	Nº Instruções	CPI
Cálculo	$3*10^8$	1,1
Acesso à Mem.	$6*10^8$	2,5
Salto	$1*10^8$	1,7
TOTAL:	10^9	

$$CPI = CPI_{CPU} + CPI_{MEM} = (3*1.1 + 6*2.5 + 1*1.7)/10 + 0 = 2$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 2 * \frac{1}{10^9} = 2s$$

Hierarquia da Memória – Desempenho

Considere o mesmo programa e máquina do acetato anterior, mas agora com um tempo de acesso à memória de 10 ns (por palavra ou instrução). Suponha ainda que esta máquina não tem cache.
Qual o CPI efectivo e o tempo de execução?

$$CPI = CPI_{CPU} + CPI_{MEM} = CPI_{CPU} + (mr_I + \%Mem * mr_D) * mp$$

Se a máquina não tem cache, então $mr_I = mr_D = 100\%$.

Da tabela tiramos que *%Mem* = 60%.

mp expresso em ciclos do relógio é 10/1 = 10 ciclos (f=1 GHz)

$$CPI = CPI_{CPU} + CPI_{MEM} = 2 + (1 + 0.6 * 1) * 10 = 2 + 16 = 18$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 18 * \frac{1}{10^9} = 18s$$

Hierarquia da Memória – Desempenho

Considere agora que existe uma *cache* com linhas de 4 palavras; a *miss rate* de acesso às instruções é de 6% e de acesso aos dados é de 10%; o tempo de acesso à memória central é constituído por uma latência de 40 ns mais 10 ns por palavra. Qual o CPI médio e o tempo de execução?

mp = 40 + 10*4 = 80 ns ; em ciclos *mp* = 80/1 = 80 ciclos

$$CPI = CPI_{CPU} + CPI_{MEM} = 2 + (0.06 + 0.6 * 0.1) * 80 = 2 + 9.6 = 11.6$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 11.6 * \frac{1}{10^9} = 11.6s$$

Hierarquia da Memória – Desempenho

Suponha que a capacidade da *cache* é aumentada para o dobro, resultando numa *miss rate* de acesso às instruções de 3.2% e acesso aos dados de 8%. No entanto, o tempo de acesso à cache (*hit* time) também aumenta, resultando num CPI_{CPU} de 2.5 . Qual o CPI médio e o tempo de execução?

$$CPI = CPI_{CPU} + CPI_{MEM} = 2.5 + (0.032 + 0.6 * 0.08) * 80 = 2.5 + 6.4 = 8.9$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 8.9 * \frac{1}{10^9} = 8.9s$$

Hierarquia da Memória – Desempenho

Para tirar maior partido da localidade espacial aumentou-se o número de palavras por linha de 4 para 8, reduzindo a *miss rate* de instruções para 1% e de dados para 6%. O tempo de acesso à memória central é composto por uma latência de 40 ns mais 10 ns por palavra. Qual o CPI médio e o tempo de execução?

mp = 40 + 10*8 = 120 ns ; em ciclos *mp* = 120/1 = 120 ciclos

$$CPI = CPI_{CPU} + CPI_{MEM} = 2.5 + (0.01 + 0.6 * 0.06) * 120 = 2.5 + 5.52 = 8.02$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 8.02 * \frac{1}{10^9} = 8.02s$$

Hierarquia da Memória – Desempenho

Para reduzir a *miss penalty* a memória central foi substituída por outra com uma latência de 40 ns e 5 ns por palavra. Qual o CPI médio e o tempo de execução?

mp = 40 + 5*8 = 80 ns ; em ciclos *mp* = 80/1 = 80 ciclos

$$CPI = CPI_{CPU} + CPI_{MEM} = 2.5 + (0.01 + 0.6 * 0.06) * 80 = 2.5 + 3.68 = 6.18$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 6.18 * \frac{1}{10^9} = 6.18s$$

Hierarquia da Memória – Desempenho

Finalmente o processador foi substituído por outro com uma frequência de 3 GHz, sem que a memória tenha sofrido qualquer alteração. Qual o CPI médio e o tempo de execução?

O ciclo do relógio é agora de 0.33 ns, logo *mp* = 80/0.33=240 ciclos

$$CPI = CPI_{CPU} + CPI_{MEM} = 2.5 + (0.01 + 0.6 * 0.06) * 240 = 2.5 + 11.04 = 13.54$$

$$T_{exec} = \#I * CPI * T_{cc} = 10^9 * 13.54 * \frac{1}{3 * 10^9} = 4.513s$$

Sumário

Tema	H & P
Hierarquia de memória	Sec. 7.1
Localidade	Sec. 7.1, 7.2
Hierarquia de memória – Desempenho	Sec. 7.3