Memoria CACHE

Definição

- Na área da computação, cache é um dispositivo de acesso rápido, interno a um sistema, que serve de intermediário entre um operador de um processo e o dispositivo de armazenamento ao qual esse operador acede.
- A vantagem principal na utilização de uma cache consiste em evitar o acesso ao dispositivo de armazenamento - que pode ser demorado -, armazenando os dados em meios de acesso mais rápidos

Definição

- Um cache é um bloco de memória para o armazenamento temporário de dados que possuem uma grande probabilidade de serem utilizados novamente.
- Uma definição mais simples de cache poderia ser:
  - Uma área de armazenamento temporária onde os dados frequentemente acedidos são armazenados para acesso rápido.

Operação

- Uma cache é feita de uma fila de elementos. Cada elemento tem um dado que é a cópia exacta do dado presente em algum outro local (original). Cada elemento tem uma etiqueta que especifica a identidade do dado no local de armazenamento original, que foi copiado.

☺ cache hit

- Quando o cliente da cache (CPU, navegador etc.) deseja aceder a um dado que acredita estar no local de armazenamento, primeiramente ele verifica a cache. Se uma entrada for encontrada com uma etiqueta correspondente ao dado desejado, o elemento da cache é então utilizado ao invés do dado original. Essa situação é conhecida como cache hit (acerto do cache).
  - Como exemplo, um navegador poderia verificar a sua cache local no disco para ver se tem uma cópia local dos conteúdos de uma página Web numa URL particular. Nesse exemplo, a URL é a etiqueta e o conteúdo da página é o dado desejado. A percentagem de acessos que resultam em cache hits é conhecida como a taxa de acerto (hit rate ou hit ratio) da cache.

☹ *cache miss*

- Uma situação alternativa, que ocorre quando a cache é consultada e não contém um dado com a etiqueta desejada, é conhecida como *cache miss* (erro do cache). O dado então é copiado do local original de armazenamento e inserido na cache, ficando pronto para o próximo acesso.

LRU (least recently used)

- Se a cache possuir capacidade de armazenamento limitada (algo comum de acontecer devido ao seu custo), e não houver mais espaço para armazenar o novo dado, algum outro elemento deve ser retirado dela para que liberte espaço para o novo elemento.
  - A forma (heurística) utilizada para seleccionar o elemento a ser retirado é conhecida como política de troca (replacement policy). Uma política de troca muito popular é a LRU (least recently used), que significa algo como "elemento recentemente menos usado".

write policy

- Quando um dado é escrito na cache, ele deve ser gravado no local de armazenamento em algum momento. O momento da escrita é controlado pela política de escrita (write policy).

write-through

- A política de write-through (algo como "escrita através") funciona da seguinte forma: a cada vez que um elemento é colocado no cache, ele também é gravado no local de armazenamento original.

write-back

- Alternativamente, pode ser utilizada a política de write-back (escrever de volta), onde as escritas não são directamente espelhadas no armazenamento. Ao invés, o mecanismo de cache identifica quais de seus elementos foram sobrepostos (marcados como sujos) e somente essas posições são colocadas de volta nos locais de armazenamento quando o elemento for retirado do cache. Por essa razão, quando ocorre um cache miss (erro de acesso ao cache pelo fato de um elemento não existir nele) em um cache com a política write-back, são necessários dois acessos à memória: um para recuperar o dado necessário e outro para gravar o dado que foi modificado no cache.

... write-back

- O mecanismo de write-back pode ser acionado por outras políticas também. O cliente pode primeiro realizar diversas mudanças nos dados do cache e depois solicitar ao cache para gravar os dados no dispositivo de uma única vez.
- Os dados disponíveis nos locais de armazenamento original podem ser modificados por outras entidades diferentes, além do próprio cache. Nesse caso, a cópia existente no cache pode se tornar inválida. Da mesma forma, quando um cliente atualiza os dados no cache, as cópias do dado que estejam presentes em outros caches se tornarão inválidas. Protocolos de comunicação entre gerentes de cache são responsáveis por manter os dados consistentes e são conhecidos por protocolos de coerência.

Princípio da localidade de referência

- É a tendência de o processador ao longo de uma execução referenciar instruções e dados da memória principal localizados em endereços próximos. Tal tendência é justificada devido as estruturas de repetição e as estruturas de dados, vetores e tabelas utilizarem a memória de forma subseqüente (um dado após o outro). Assim a aplicabilidade da cache internamente ao processador fazendo o intermédio entre a memória principal e o processador de forma a adiantar as informações da memória principal para oprocessador.

Tipos de memória cache

- Os tipos de memória cache mais conhecidos são:
  - Mapeamento direto,
  - Totalmente associativa
  - Associativa por conjunto (N-way).

Cache em níveis

- Com a evolução na velocidade dos dispositivos, em particular nos processadores, o cache foi dividido em níveis, já que a demanda de velocidade a memória é tão grande que são necessários caches grandes com velocidades altíssimas de transferencia e baixas latências. Sendo muito difícil e caro construir memórias caches com essas características, elas são construídas em níveis que se diferem na relação tamanho X desempenho.

Cache L1

- Uma pequena porção de memória estática presente dentro do processador.
- Em alguns tipos de processador, como o Pentium 2, o L1 é dividido em dois níveis:
  - dados e instruções (que "dizem" o que fazer com os dados).
  - A partir do Intel 486, começou a se colocar a L1 no próprio chip [processador].
  - Geralmente tem entre 16KB e 128KB; hoje já encontramos processadores com até 16MB de cache.

Cache L2

- Possuindo o Cache L1 um tamanho reduzido e não apresentando uma solução ideal, foi desenvolvido o cache L2, que contém muito mais memória que o cache L1.
- Alguns processadores colocam essa cache fora do processador, por questões econômicas, pois uma cache grande implica num custo grande, mas apartir do Pentium II, costuma que as caches L1 e L2 estão no mesmo cartucho que está o processador.

Cache L2

- A memória cache L2 é, sobretudo, um dos elementos essenciais para um bom rendimento do processador mesmo que tenha um clock baixo.
  - Um exemplo prático é o caso do Intel Itanium 9152M (para servidores) que tem apenas 1.6 GHz de clock interno e ganha de longe do atual Intel Extreme, pelo fato de possuir uma memória cache de 24Mb. *"Quanto mais alto é o clock do processador, mais este aquece e mais instável se torna."*
  - Os processadores Intel Celeron tem tão fraco desempenho por possuir menor memória cache L2.
  - Um Pentium M 730 de 1.6 GHz de clock interno, 533 MHz de FSB e 2 MB de cache L2, tem rendimento semelhante a um Intel Pentium 4 2.4 GHz, aquece muito menos e torna-se muito mais estável e bem mais rentável do que o Intel Celeron M 440 de 1.86 GHz de clock interno, 533 MHz FSB e 1 MB de cache L2.

Cache L3

- Terceiro nível de cache de memória. Inicialmente utilizado pelo AMD K6-III (por apresentar o cache L2 integrado ao seu núcleo) utilizava o cache externo presente na placa-mãe como uma memória de cache adicional. Ainda é um tipo de cache raro devido a complexidade dos processadores atuais, com suas áreas chegando a milhões de transístores por micrómetros ou picómetros de área. Ela será muito útil, é possível a necessidade futura de níveis ainda mais elevados de cache, como L4 e assim por diante.
- Hoje está de volta dentro das novas CPUs da Intel i3, i5 ou i7.

Caches inclusivos e exclusivos

- Caches Multi-level introduzem novos aspectos na sua implementação. Por exemplo, em alguns processadores, todos os dados no cache L1 devem também estar em algum lugar no cache L2. Estes caches são estritamente chamados de inclusivos.
- Outros processadores (como o AMD Athlon) têm caches exclusivos - os dados podem estar no cache L1 e L2, nunca em ambos. Ainda outros processadores (como o Pentium II, III, e 4 de Intel), não requerem que os dados no cache L1 residam também no cache L2, embora possam frequentemente fazê-lo.

Não há nenhum nome universal aceitado para esta política intermediária, embora o termo inclusivo seja usado.

☺ Caches exclusivos

- A vantagem de caches exclusivos é que são capazes de armazenarem mais dados.
  - Esta vantagem é maior quando o cache L1 exclusivo é de tamanho próximo ao cache L2, e diminui se o cache L2 for muitas vezes maior do que o cache L1.
  - Quando o L1 falha e o L2 acerta acesso, a linha correta do cache L2 é trocada com uma linha no L1. Esta troca é um problema, uma vez que a quantidade de tempo para tal troca ser realizada é relativamente alta.

☺ Caches inclusivos

- Uma das vantagens de caches estritamente inclusivos é que quando os dispositivos externos ou outros processadores em um sistema multiprocessado desejam remover uma linha do cache do processador, necessitam somente mandar o processador verificar o cache L2.

Nas hierarquias de cache exclusiva, o cache L1 deve ser verificado também.

- Uma outra vantagem de caches inclusivos é que um cache maior pode usar linhas maiores do cache, que reduzes o tamanho dos Tags do cache L2. (Os caches exclusivos requerem ambos os caches teres linhas do mesmo tamanho, de modo que as linhas do cache possam ser trocadas em uma falha no L1 e um acerto no L2).

Tamanho da cache

- Quando é feita a implementação da memória cache, alguns aspectos são analisados em relação a seu tamanho:
  - A relação acerto/falha
  - Tempo de acesso a memória principal
  - O custo médio, por bit, da memória principal, da cache L1 e L2
  - O tempo de acesso da cache L1 ou L2
  - A natureza do programa a ser executado no momento