Arquitecturas Multi-core

Ricardo Gonçalves
up200804565@fc.up.pt

8 de Dezembro de 2013

Resumo

Actualmente, tem-se assistido à utilização massificada de processadores multi-core em diversas situações, como em computadores pessoais e muito recentemente em dispositivos móveis. Com este trabalho, o que se pretende é fazer um estudo das arquitecturas multi-core: compreender o que levou ao seu aparecimento, como têm evoluído ao longo do tempo, dar a conhecer várias arquitecturas existentes e qual a utilidade de cada uma delas. Também se dará a conhecer as estratégias necessárias para resolver os problemas de consistência das caches neste contexto.

1 Introdução

O aparecimento de processadores multi-core é recente. O primeiro processador multi-core de uso geral produzido foi o POWER4 pela IBM, em 2001. Mas só a partir de 2005, é que as grandes empresas produtoras de processadores, a Intel e a AMD, é que laçaram processadores com mais do que um núcleo. Desde então, a evolução tem sido constante, tanto no processo de fabrico como na quantidade processadores por *chip* e na capacidade de processamento.

Inicialmente, será apresentada uma noção do que é um processador multi-core e quais os motivos que levaram ao seu surgimento. De seguida, irá dar-se a conhecer a evolução dos processadores multi-core e também de algumas arquitecturas. Por fim, será discutido problemas de coerência de caches entre os vários processadores.

1.1 Processador Multi-core: o que é?

Um processador multi-core é um componente que contém mais do que uma unidade de processamento, denominado de *core*, tudo integrado no mesmo *chip*. Cada processador funciona de forma independente dos restantes. Isto permite que se execute várias instruções ao mesmo tempo, podendo assim distribuir-se as tarefas pelos vários processadores. Com um processador single-core, todas as tarefas tinham de ser executadas sequencialmente.

Embora ter várias unidades de processamento permite executar várias instruções simultaneamente, isto causa problemas de sincronização entre os vários processadores devido a acessos concorrentes a recursos do computador. Para resolver estes problemas, foi necessário criar arquitecturas específicas que os resolvesse, nomeadamente a nível das caches.

1.2 Do multi-processador ao multi-core

Apesar do surgimento dos multi-cores ser recente, a ideia de utilizar multi-processadores não é nova. Um computador multi-processador é um computador que está equipado com vários processadores, mas em *chips* separados fisicamente. Desde dos princípios da era dos processadores que se tem usado este tipo de arquitecturas, mas o seu uso era orientado para situações que exigiam maiores capacidades de processamento.

Com a constante necessidade de crescimento da capacidade de processamento, tornou-se difícil criar processadores single-core capazes de satisfazer a procura. Para aumentar a capacidade de

processamento, o que se fazia principalmente era aumentar a frequência de funcionamento do processador. Mas aumentar a frequência faz com que a quantidade de energia dissipada aumente também, tornando estes processadores bastante dispendiosos e mais difíceis de construir. Para além disso, estava-se a chegar ao limite da capacidade de funcionamento do próprio material.

Estes factores fizeram com se adoptasse uma estratégia diferente. A solução encontrada foi o processador multi-core. Os multi-cores relativamente ao multi-processadores permitem ter consumos energéticos mais baixos, comunicações melhores e mais rápidas entre processadores devido à proximidade e custos de produção mais baixos.

2 Evolução dos processadores multi-core

Os processadores multi-core tem evoluído muito rapidamente. O surgimento do primeiro processador dual-core foi o POWER4 da IBM em 2001. Só apenas em Maio de 2005 é que a Intel e a AMD é que lançaram processadores dual-core, o Pentium D e o Athlon 64 X2, respectivamente. Mas desde então que o progesso tem sido constante. Actualmente, de forma geral, os processadores são equipados com 4 a 8 núcleos para uso pessoal e com 20 núcleos para servidores. Existem, naturalmente, variadas configurações específicas, com mais ou menos núcleos, que cada fabricante adopta para as variadas situações.

Quanto ao futuro dos multi-cores, alguns especialistas prevêm que em 2017 possamos ter processadores embebidos com 4096 cores, processadores com 512 cores para servidores e processadores com 128 cores para uso pessoal.

3 Arquitecturas multi-core

As arquitecturas dos processadores multi-core estão intrinsecamente ligadas com o objectivo que se pretende alcançar. De seguida, serão dadas a conhecer diferentes arquitecturas multi-core. São bastante diferentes umas das outras, pois o problema que pretendem resolver também é diferente.

3.1 Processadores de uso geral

Um processador single-core é do tipo SISD (Single Instruction Single Data) pois só executa uma instrução a cada momento sobre um dado. Geralmente, um processador single-core recorre a dois níveis de cache (L1 e L2) e só depois é que recorre à memória principal.

Com a introdução dos processadores multi-core, o processador a passa a ser do tipo MIMD (Multiple Instruction Multiple Data), pois, a cada momento, estão a ser executadas várias instruções sobre vários dados. Isto porque, como já se referiu, passou-se a ter mais que um processador. Mas a processo de interacção entre as caches e os processadores é diferente. No exemplo da Figura 1, os processadores têm dois níveis de cache dedicados a cada processador acrescido de um terceiro nível de cache (L3) que é partilhado por todos eles e só depois então é que tem a memória principal. Este modelo é o mais utilizado actualmente em processadores para uso pessoal. No entanto, podem existir arquitecturas diferentes, como ter apenas dois níveis de cache ou então não ter cache partilhada entre processadores.

Este modelo de multi-cores com caches dedicadas cria problemas de coerência entre caches de diferentes processadores, que será explicado posteriormente. Para resolver este problema, terá de se utilizar um mecanismo de notificação para estas situações.

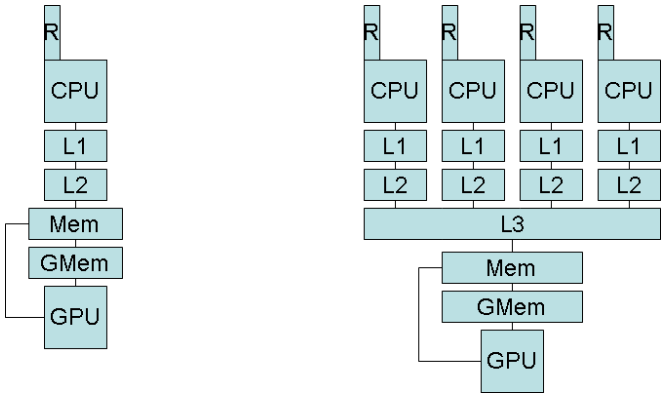

Figura 1: Do lado esquerdo está a arquitectura de um processador single-core. Do lado direito está a arquitectura de um processador quad-core. Na figura, R entende-se por Registos, L1, L2 e L3 por caches, Mem por Memória, GMem por Memória Gráfica.

3.2 Processadores gráficos

O processadores gráficos geralmente são do tipo SIMD (Single Instruction Multiple Data), o que por si só, poderá indicar que a arquitectura será diferente.

Na Figura 2 é apresentada a arquitectura de um processador gráfico CUDA da NVidia que se diferencia pela grande quantidade de pequenos processadores. Isto está relacionado com o facto desta arquitectura ser SIMD. O objectivo destes processadores é aplicar o mesmo conjunto de instruções a uma grande quantidade de dados. Tendo muitas unidades processamento, consegue-se mais rapidamente completar a computação devido ao paralelismo.

Em cada unidade processamento existe um ficheiro de registos e um grande número de threads que são eficientemente trocadas pelo escalonador implementado em hardware. Cada processador tem cache partilhada e que precisa de ser explicitamente gerida pelo programador.

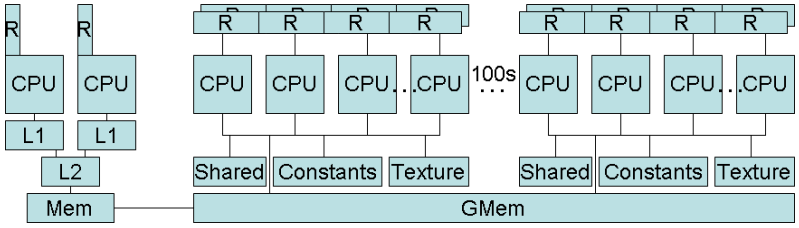

Figura 2: Arquitectura de um processador do tipo CUDA. Do lado esquerdo do diagrama está o processador do sistema que neste caso é um dual-core. Na figura, R entende-se por Registos, L1 e L2 por caches, Mem por Memória, GMem por Memória Gráfica.

Na Figura 3 estão as arquitecturas de dois processadores de consolas de jogos, a PlayStation 3 e a XBOX 360 que apresentam arquitecturas multi-core bastante diferentes.

O processador CELL da PlayStation 3 foi o que trouxe uma arquitectura que fez a diferença na geração de gráficos de videojogos. Possui uma série de co-processadores com memória dedicada para instruções e dados, denominada Local Store. As transferências de dados são feitas apenas por DMA (Direct Memory Access).

Já arquitectura da XBOX 360 é muito parecida à arquitectura de um processador multi-core normal. A principal vantagem desta arquitectura está no facto da memória gráfica e memória do sistema serem a mesma. É também permitido ao processador gráfico ter acesso à cache L2 do processador.

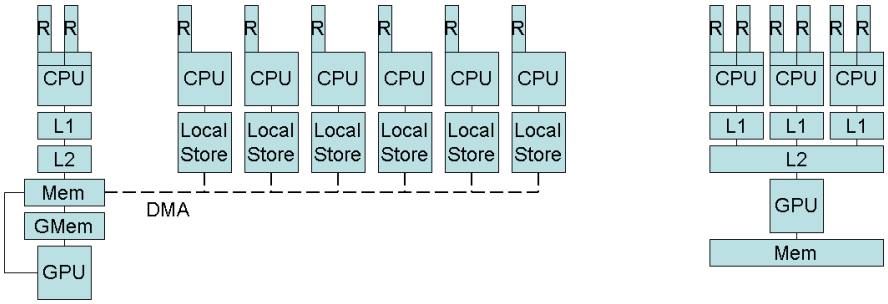

Figura 3: À esquerda está a arquitectura do processador CELL, da PlayStation 3. À direita está a arquitectura do processador da XBOX 360. Na figura, R entende-se por Registos, L1 e L2 por caches, Mem por Memória, GMem por Memória Gráfica.

4 Caches

Na secção sobre as arquitecturas, vimos que existiam caches que eram dedicadas a cada processador e caches que eram partilhadas por todos.

Quando a cache é dedicada, existe a vantagem do acesso ser mais rápido, pois ela encontra-se fisicamente mais próxima do processador. Por esse motivo, é que se usam as caches de mais baixo nível como dedicadas.

Já quando a cache é partilhada, as *threads* em diferentes processadores podem partilhar os mesmos dados da cache. E caso estejam a ser executadas poucas *threads*, estas ficam com mais espaço disponível tornando a sua execução mais rápida.

4.1 O problema da coerência das caches

Como já foi referido anteriormente, existe o problema da coerência das caches. Este problema acontece quando um processador modifica um valor de uma posição de memória e essa mesma posição de memória está presente na cache de outro processador como válida. Este processador quando for utilizar o seu valor, não irá obter o valor correcto. Em vez de utilizar o valor novo modificado pelo primeiro processador, irá usar o valor antigo antes da modificação.

Este problema não é restrito a processadores multi-core, mas também a computadores multi-processadores. Existem várias soluções, mas a mais simples é o protocolo de invalidação.

Quando um processador altera algum valor, este envia um pedido de invalidação através do barramento inter-core para os outros processadores. O barramento inter-core é um barramento que interliga todos os processadores e permite que comuniquem entre eles. Os processadores ao receberem o pedido de invalidação e se tiverem essa posição de memória na cache, marcam essas posições como inválidas obrigando a que se tenha que pedir o valor de novo à memória central. Este sistema implica que as caches sejam *write-through*, mas tem a vantagem gerar menos tráfego no barramento inter-core.

Existe uma pequena variação deste modelo, que em vez de enviar um pedido de invalidação, envia o próprio valor. Desta forma, não obriga a que se tenha que ir à memória principal buscar o valor actualizado, mas requer que se envie uma mensagem a cada alteração de um valor por todos os processadores.

Na implementação dos processadores, são usados protocolos mais sofisticados que requerem bits de estado adicionais.

5 Conclusão

Apesar dos problemas associados à utilização de processadores multi-core, tem sido essa a tendência de evolução dos processadores na busca de maior capacidade de processamento.

Ainda há poucos anos não havia nenhum processador multi-core, actualmente já faz parte do nosso uso quotidiano a sua utilização. Vão surgindo também novas arquitecturas que permitem tirar maior partido do processamento, arquitecturas estas que podem ser úteis em situações específicas, como é o caso dos processamento de gráficos.

Com a massificação dos multi-core, as técnicas de programação paralela têm ganho mais relevância. Só dessa forma é que se consegue tirar partido da capacidade dos dispositivos que nos rodeiam cada vez mais.

Referências

[1] Stephen W. Keckler, Kunle Olukotun, H. Peter Hofstee, *Multicore Processors and Systems*. Springer, 2009.

[2] Jernej Barbic, *Multi-core architectures*. 2007. URL: `http://www.cs.cmu.edu/~fp/courses/15213-s07/lectures/27-multicore.pdf`.

[3] *Multi-Platform Multi-Core Architecture Comparison (PC, Wii, Xbox 360, PS3, CUDA, Larrabee)*. 2008. URL: `http://beautifulpixels.blogspot.pt/2008/08/multi-platform-multi-core-architecture.html`.

[4] IBM. *Power 4: The First Multi-Core, 1GHz Processor*. URL: `http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/power4/`.

[5] Wikipedia. *Superscaler*. URL: `http://en.wikipedia.org/wiki/Superscalar`.

[6] Garrison Prinslow. 2011. *Overview of Performance Measurement and Analytical Modeling Techniques for Multi-core Processors*. URL: `http://www.cs.wustl.edu/~jain/cse567-11/ftp/multcore/`

[7] Bryan Schauer. 2008. *Multicore Processors - A Necessity*. URL: `http://www.csa.com/discoveryguides/multicore/review.pdf`

[8] *CUDA Programming - Week 4. Shared memory and register*. URL: `http://www.csa.com/discoveryguides/multicore/review.pdf`