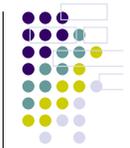




Arquitecturas Multithread

Filipe Campos



Sumário

- Introdução
- Arquitecturas:
 - Multiprocessadores
 - Multiprocessamento Simétrico
- Hyper Threading
 - Arquitectura:
 - Pipeline
 - Cache
 - barramento
 - Benchmarks
- Conclusão

Introdução

Durante muitos anos, a chave para aumentar o desempenho geral de um processador consistia em aumentar a sua frequência de relógio.

Nos dias de hoje, aumentar ainda mais a frequência de relógio, implica:

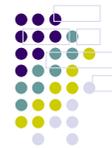
- Elevados custos de implementação.
- Aumento da complexidade de implementação.
- Aumento de desempenho geral pouco significativo.

Problema:

Diferença de velocidade de funcionamento entre a memória e o processador, cria um aumento de "caches misses" reduzindo o tempo de processamento útil.

Objectivo:

Criar mecanismo de tolerância a memória, permitindo assim aumentar a performance dos processadores sem aumentar muito a complexidade e custo de implementação.



Arquitecturas

Multiprocessadores (CMP)

- Várias unidades de processamento num só máquina (Exemplo: dual processador).
- Permite que diferentes threads sejam executadas em paralelo.

Multiprocessamento Simétrico (SMT)

- Combina a ideia da arquitectura CMP e arquitecturas superescalares.
- Permite que diferentes threads sejam executadas em paralelo.
- Permite que sejam executadas várias instruções por ciclo de relógio.



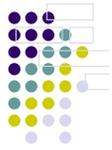


Hyper Threading

- Melhoria da arquitectura Multiprocessamento Simétrico(SMT).
- Usa os mesmos princípios da arquitectura SMT, mas permite que o processador tenha duas threads.
- Um processador físico aparece como dois processadores lógicos para o sistema operativo.

Resultado teorico:

- ▢ Aumenta o desempenho geral, através da mudança de utilização dos recursos do CPU entre duas threads. Muda a utilização dos recursos do processador da thread que está a ser executada para a segunda, quando a primeira inicia uma operação de longa duração.

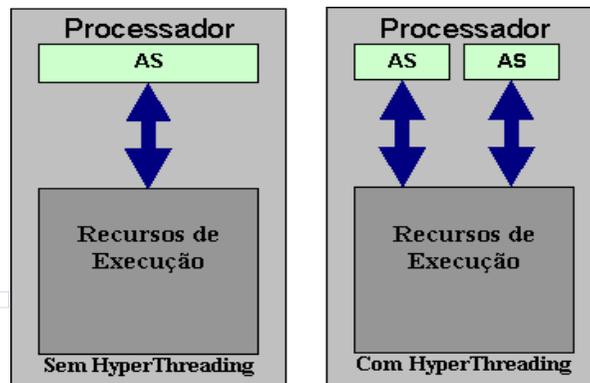


Arquitectura

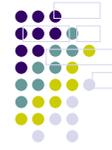
Para operar duas threads num único processador é necessário ter em consideração:

- O estado de cada uma das threads (*architecture state - AS*) seja mantido no chip.
- A área adicionada no chip para a sua implementação tem que ser redizida.
- Não perturbar muito o tempo de cada ciclo de relógio.

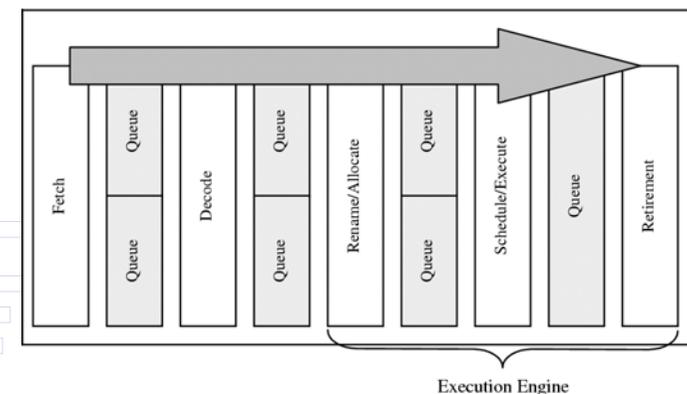
Arquitectura (2)



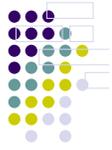
AS (*architecture state*): registos e controlador de interrupção



Pipeline



Duplicado: branch prediction, instrution pointer, Decode stage Buffer



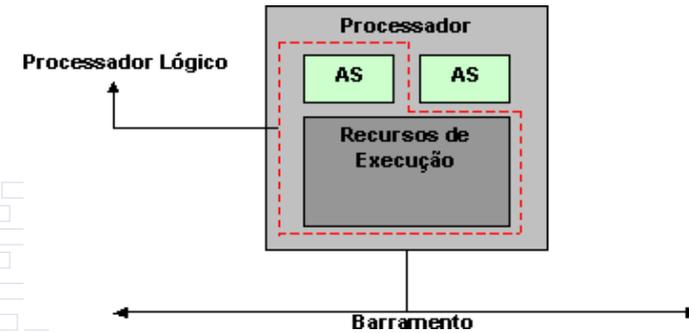
Cache

Cache é partilhada pelo dois processadores lógicos.

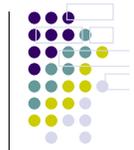
- Foi adicionado um mecanismo de identificação de threads nos acessos a memória.
- Cada thread tem os seus registos de reserva da memória.



Barramento

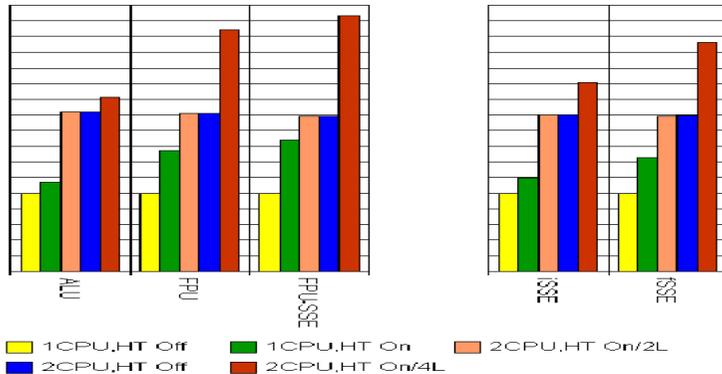


Barramento partilhado entre os dois processadores lógicos.

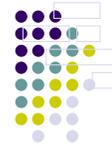


Benchmarks

Benchmarks do Dell Precision 530 Workstation usando o software SiSoft Sandra Benchmarks

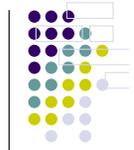


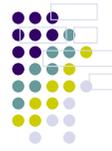
1 CPU, HT Off: 1 CPU = 1 processador lógico.
1 CPU, HT On: 1 CPU e hyper threading = 2 processadores logicos.
2 CPU, HT Off: 2 CPU = 2 processadores logicos.
2 CPU, HT On/2L: 2 CPU e hyper threading = 2 processadores logicos.
2 CPU, HT On/4L: 2 CPU e hyper threading = 4 processadores logicos.



Vantagens

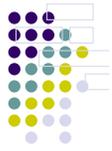
- Aumento significativo do desempenho, em algumas situações mais de 50% .
- O aumento de área do chip para o suporte de hyper threading e inferior a 5%.
- Impacto quase nulo sobre o tempo de ciclo (menor que 1%).
- Baixo custo de implementação.





Desvantagens

- sistemas operativos e aplicações têm que suportar multithreading, aumentando a complexidade e custo de implementação do software.
- Partilha de recursos, como por exemplo a cache pode introduzir erros de programação paralela, como conflitos de escrita/leitura.
- Ao permitir que uma thread use todos os recursos do processador, em caso de espera (exemplo: ciclo de espera) a segunda thread ficará parada também, conduzindo a uma diminuição da performance.
- No caso de um processador normal, o sistema operativo é que decide qual das duas threads é que tinha prioridade de execução.

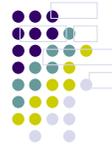


Conclusão

Dois processadores lógicos são melhores do que um?

A resposta é positiva na maior parte das situações, existindo um aumento substancial de desempenho, permitindo um melhor uso dos recursos do processador.

Na execução de software não computacionalmente intensivo ou que não suporte multithreading como por exemplo word, e-mail que representa a maior fatia das aplicações usadas pelo utilizador normal, o uso desta tecnologia não traz nenhuma melhoria no desempenho, podendo até em algumas situações diminuir.



Duvidas ?