

Lista de Exercícios - Hardware paralelo

1. Quando discutimos a adição de ponto flutuante, partimos da suposição simplificadora de que cada uma das unidades funcionais levava o mesmo tempo. Suponha que buscar (*fetch*) e armazenar (*store*) levem 2 nanossegundos cada e que as operações restantes levem 1 nanossegundo cada.
 - (a) Quanto tempo leva uma adição de ponto flutuante com essas suposições?
 - (b) Quanto tempo levará uma adição sem pipeline de 1.000 pares de *floats* com essas suposições?
 - (c) Quanto tempo levará uma adição em pipeline de 1.000 pares de *floats* com essas suposições?
 - (d) O tempo necessário para busca e armazenamento pode variar consideravelmente se os operandos/resultados forem armazenados em diferentes níveis da hierarquia de memória. Suponha que uma busca de um cache de nível 1 leve dois nanossegundos, enquanto uma busca de um cache de nível 2 leve cinco nanossegundos e uma busca da memória principal leve cinquenta nanossegundos. O que acontece com o pipeline quando há uma falha de cache de nível 1 na busca de um dos operandos? O que acontece quando há uma falha de nível 2?
2. Explique como uma fila implementada em hardware na CPU poderia ser usada para melhorar o desempenho de um *cache write-through*.
3. Dado o exemplo a seguir envolvendo leituras de cache de um array bidimensional.

```
double A[MAX][MAX], x[MAX], y[MAX];  
.  
.  
.  
// Initializa A e x; y = 0  
.  
. .  
// Primeiro par de loops/  
for (i = 0; i < MAX ; i++)  
    for (j = 0; j < MAX; j++)  
        y [i] += A[i] [j]x[j];  
.  
. .  
// y = 0  
.  
. .  
// Segundo par de loops  
for (j = 0; j < MAX ; j++)  
    for (i = 0; i < MAX; i++)  
        y [i] += A[i] [j]x[j];
```

- (a) Como o aumento do tamanho da matriz afetaria o desempenho dos dois pares de loops aninhados?
- (b) Como o aumento do tamanho da cache afetaria o desempenho dos dois pares de loops aninhados?
- (c) Quantas falhas ocorrem nas leituras de A no primeiro par de loops aninhados?
- (d) Quantas falhas ocorrem no segundo par?

Suponha que uma linha de cache consiga armazenar quatro elementos de A .

4. Responda as questões abaixo. Explique o por quê de cada resposta.

- (a) A adição de cache e memória virtual a um sistema von Neumann altera sua designação como sistema SISD?
- (b) E quanto à adição de pipeline?
- (c) Multiple issue?
- (d) Hardware multithreading?

5. Suponha que um programa deva executar 10^{12} instruções para resolver um problema específico. Suponha ainda que um sistema de processador único possa resolver o problema em 10^6 segundos (cerca de 11,6 dias). Portanto, em média, o sistema de processador único executa 10^6 instruções por segundo. Agora suponha que o programa tenha sido paralelizado para execução em um sistema de memória distribuída. Suponha também que o programa paralelo usa p processadores e, assim, cada processador executará $10^{12}/p$ instruções. Além disso, cada processador deverá enviar $10^9(p - 1)$ mensagens. Por fim, suponha que não haja *overhead* adicional na execução do programa paralelo. Ou seja, o programa será concluído depois que cada processador tiver executado todas as suas instruções e enviado todas as suas mensagens e não haverá atrasos devido a coisas como espera por mensagens.

- (a) Suponha que demore 10^{-9} segundos para enviar uma mensagem. Quanto tempo levará para o programa ser executado com 1000 processadores se cada processador for tão rápido quanto o processador único no qual o programa serial foi executado?
- (b) Suponha que demore 10^{-3} segundos para enviar uma mensagem. Quanto tempo levará para o programa rodar com 1000 processadores?

6. (a) Suponha que um sistema de memória compartilhada use coerência de cache (*snooping*) e caches *write-back*. Suponha também que o núcleo 0 tenha a variável x em seu cache e execute a atribuição $x = 5$. Finalmente, suponha que o núcleo 1 não tenha x em seu cache e, após a atualização do núcleo 0 para x , o núcleo 1 tente executar $y = x$. Qual valor será atribuído a y ? Por que?

(b) Suponha que o sistema de memória compartilhada da parte anterior utilize um protocolo baseado em diretório. Qual valor será atribuído a y ? Por que?

(c) Como os problemas encontrados nas duas primeiras partes podem ser resolvidos?

Referência

- PACHECO, Peter S. An introduction to parallel programming. Amsterdam Boston: Morgan Kaufmann, c2011. xix, 370 p. ISBN: 9780123742605.