

# Resolução/discussão da AF6 : Parallelismo

1.

- a) Uma vez que todos os cores funcionam em simultâneo, e que as variáveis são alojadas em memória partilhada, o resultado depende do timing de acesso aos valores (leitura e escrita) pelos vários processadores.

Uma vez que os valores de x e y só são alterados uma vez pelos cores 1 e 2, respetivamente, os seus valores finais serão sempre 2. No entanto, w e z assumirão valores diferentes consoante x e y já tenham sido alterados ou não. Os valores finais possíveis serão então:

$$\begin{aligned}w: & 1(0+0+1), 3(0+2+1 \text{ ou } 2+0+1) \text{ ou } 5(2+2+1) \\z: & 0(0+0), 2(0+2 \text{ ou } 2+0) \text{ ou } 4(2+2)\end{aligned}$$

Há portanto 9 resultados finais possíveis, correspondendo às combinações de resultados de w e z:

x:2, y:2, w: 1, z:0  
x:2, y:2, w: 1, z:2  
x:2, y:2, w: 1, z:4  
x:2, y:2, w: 3, z:0  
x:2, y:2, w: 3, z:2  
x:2, y:2, w: 3, z:4  
x:2, y:2, w: 5, z:0  
x:2, y:2, w: 5, z:2  
x:2, y:2, w: 5, z:4

- b) Para tornar a execução mais determinista, pode-se forçar que as instruções executem numa determinada sequência. No entanto, essa abordagem elimina todas as vantagens do processamento paralelo, transformando-o em sequencial.

Analisando o programa, os cores 1 e 2, que lidam com variáveis diferentes, podem correr em paralelo. Os cores 3 e 4 também, uma vez que escrevem em variáveis diferentes. Mas é importante que as instruções que afetam as variáveis x e y corram antes das que as lêem, para que o resultado final seja único. O mecanismo de sincronização para este efeito é o "lock", que permite que os cores 1 e 2 "tranquem" as variáveis x e y até serem alteradas. Dessa forma, os cores 3 e 4 só terão acesso ao valor final das mesmas, garantindo assim o resultado único.

2. O objetivo desta atividade é pesquisar na web e comparar dois hypervisors existentes no mercado. Dois dos mais utilizados são o VMware ESXi e o Hyper-V da Microsoft.

- a) Num ambiente de virtualização, várias máquinas virtuais partilham CPU e memória. Enquanto que o CPU pode ser partilhado no tempo, i.e., cada máquina consome algum tempo do CPU, consoante as necessidades em cada momento, as porções de memória necessárias por cada máquina têm de ser alocadas de forma mais permanente, sob pena de se perderem dados. Tipicamente, as máquinas virtuais, tanto no VMware como no Hyper-V são configuradas inicialmente com uma quantidade fixa de memória (virtual). A forma como a gerem é que é diferente. O VMware usa várias técnicas para disponibilizar páginas de memória não usadas numas máquinas a outras que delas necessitam, sendo que para o sistema operativo da máquina virtual,

a quantidade de memória virtualmente disponível é a que foi definida inicialmente - muitas vezes a memória física disponível é inferior à soma das que foram definidas em todas as máquinas, mas espera-se que seja suficiente para suprir as necessidades em cada momento. O Hyper-V, perante a necessidade de mais memória numa máquina virtual, atribui efetivamente essa memória à máquina, como se fosse adicionada "a quente" numa máquina física. Esta abordagem tem a desvantagem de o SO ou as aplicações poderem não reconhecer esse aumento de memória, mas, se for reconhecido, é mais rápida que a abordagem do VMware.

**b)** Num ambiente de um único processador, as máquinas virtuais dispõem de processadores virtuais que, na realidade, correspondem a fatias de tempo do processador físico. Havendo mais do que um CPU (ou um CPU com vários "cores"), essas fatias de tempo podem ser distribuídas pelos vários processadores, aumentando a eficiência. Mais, passa a ser possível definir máquinas virtuais multi-core, embora o nº de cores virtuais não possa ultrapassar o nº de cores físicos. A tecnologia de hyper-threading, desenvolvida pela Intel, faz com que um core pareçam dois ao sistema operativo. Na prática, o que a tecnologia faz é colocar um segundo "thread" a correr sempre que o primeiro fica a aguardar o acesso a outros recursos (ex. memória), ganhando alguma performance.

A vantagem de ter CPUs com hyper-threading num hypervisor é altamente dependente das aplicações: se forem aplicações de processamento intensivo, não tem vantagem, a performance pode mesmo degradar-se devido ao overhead necessário. Por outro lado, se as aplicações dependerem mais fortemente de recursos externos, podem partilhar o CPU com vantagem, pois grande parte do tempo estarão a aceder a esses outros recursos.

## Referências:

- [1] <https://www.vcloudnine.de/memory-management-vmware-esxi-vs-microsoft-hyper-v/>
- [2] <https://www.altaro.com/hyper-v/hyper-v-virtual-cpus-explained/>
- [3] <https://virtualizationreview.com/articles/2016/12/08/should-you-use-hyperthreading-with-vmware.aspx>

Última alteração: Quarta, 5 Janeiro 2022, 15:46