

SINCRONIZAÇÃO E COORDENAÇÃO

DCE540 - Computação Paralela e Distribuída

Atualizado em: 27 de janeiro de 2022

Iago Carvalho

Departamento de Ciência da Computação



SINCRONIZAÇÃO E COORDENAÇÃO

Por sincronização, pode-se entender que

- um processo aguarda a conclusão de outro
- ambos são executados em sincronia
- sincronização de dados
 - garantir que os dados de dois ou mais processos são os mesmos

Por coordenação, pode-se entender que

- deve-se coordenar as interações e dependências entre dois ou mais processos

Pode-se dizer que coordenação encapsula sincronização

SINCRONIZAÇÃO POR *CLOCK*

Uma das formas mais simples de obtermos sincronização em sistemas distribuídos é realizando sincronização por *clock*

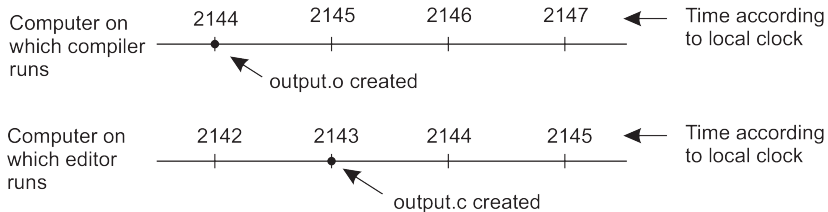
- Aqui, *clock* se refere a horário e não ao ciclo de *clock* do computador

Em sistemas não-distribuídos, o *clock* é uma ótima forma de realizar sincronização

- Exemplo: *make* em sistemas UNIX

Entretanto, em sistemas distribuídos, não existe necessariamente o conceito de um relógio global

UM MAKE EM SISTEMAS DISTRIBUÍDOS



Todo computador possui um *clock* físico

- um cristal de quartzo localizado na placa mãe

Este cristal é alimentado por uma bateria externa e possui uma RAM própria

Quando alvo de uma corrente elétrica, o cristal produz uma oscilação

- frequência da oscilação é estável e bem definida
- a cada n oscilações, gera-se uma interrupção de sistema
- esta interrupção conta um segundo

SINCRONIZAÇÃO DE *CLOCK*

Pode-se considerar o UTC (Universal Coordinated Time) como um *clock* global

Caso um computador seja sincronizada com o UTC

- Outros computadores podem sincronizar seus *clocks* com este primeiro

Seja t o tempo dado pelo UTC. Além disso, seja $C_p(t)$ o tempo atual em um computador p . O objetivo de um algoritmo de sincronização de *clock* é garantir que

$$\forall t, \forall p, q : |C_p(t) - C_q(t)| \leq \pi,$$

onde π é uma medida de **precisão** mínima

SINCRONIZAÇÃO DE *CLOCK*

A precisão anterior refere-se unicamente a diferença de tempo entre dois computadores parte de um mesmo sistema distribuído.

Quando medimos a discrepância do tempo entre um computador e o UTC, estamos medindo sua **acurácia**, que é dada por

$$\forall t, \forall p : |C_p(t) - t| \leq \alpha,$$

onde α é a acurácia mínima desejada

O objetivo de algoritmos de sincronização de *clock* é garantir

- Precisão mínima entre os computadores de um sistema distribuído
- Acurácia dos *clocks* de cada computador em relação ao UTC

DESVIO DE *CLOCK*

Um *clock* baseado em cristal de quartzo possui um desvio médio de 10^{-6} segundos por segundo

- Equivalente a 31,5 segundos por ano, aproximadamente

Todo sistema de *hardware* que faz medição de *clock* possui um desvio máximo de *clock* ρ especificado

Seja $F(t)$ a frequência de oscilação do *clock* de uma máquina em um instante t . Além disso, seja F a frequência ideal de oscilação. Podemos dizer que o desvio de *clock* de uma máquina obedece a equação

$$\forall t : (1 - \rho) \leq \frac{F(t)}{F} \leq (1 + \rho)$$

DESVIO DE *CLOCK*

Considere o *clock* de dois computadores estejam desviando-se do UTC em direções opostas

- O primeiro sempre mais rápido
- O segundo sempre mais lento

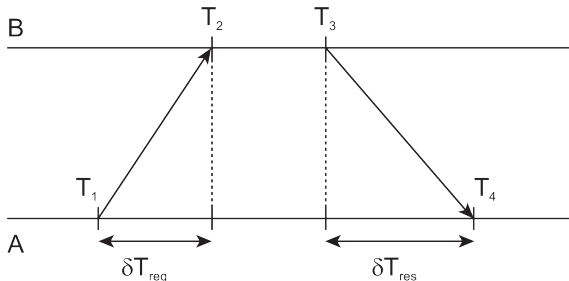
A **precisão** destes *clocks* será até de

$$2\rho \Delta t$$

Para se garantir uma precisão π , os *clocks* devem ser resincronizados pelo menos a cada

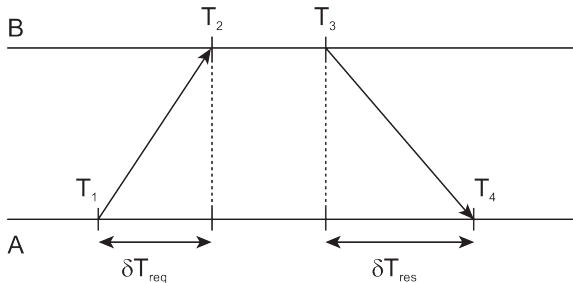
$$\frac{\pi}{2\rho} \text{ segundos}$$

NETWORK TIME PROTOCOL (NTP)



1. A manda requisição para B no tempo T_1
2. B recebe a requisição no tempo T_2
3. No tempo T_3 , B responde A enviando o valor T_2
4. A recebe a resposta no tempo T_4

NETWORK TIME PROTOCOL (NTP)

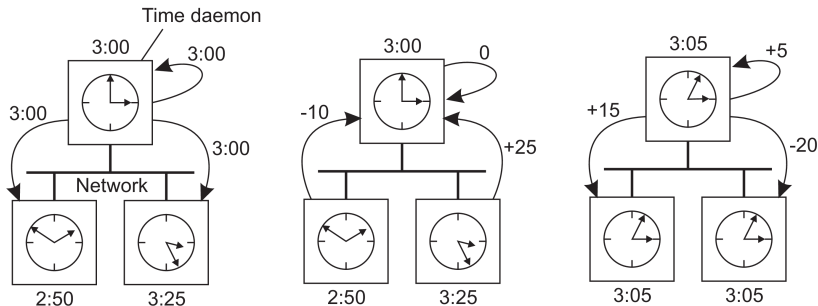


$$\text{offset } \theta = T_3 + \frac{(T_2 - T_1) + (T_4 - T_3)}{2} - T_4 = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

$$\text{delay } \delta = \frac{(T_4 - T_1) - (T_3 - T_2)}{2}$$

ALGORITMO DE BERKLEY

Um servidor tenta continuamente atualizar os *clocks* dos computadores que fazem parte de um sistema distribuído



Clock lógico refere-se ao fato de que não é necessário sincronizar os *clocks* das máquinas

- Ao invés disso, deve-se saber somente a ordem em que os eventos no sistema ocorreram

Este esquema de sincronização também diz que não é necessário que o *clock* de todas as máquinas do sistema estejam sincronizados

- Só é necessário sincronizar o *clock* daquelas máquinas ou dispositivos que interagem umas com as outras

CLOCK LÓGICO - ALGORITMO DE LAMPORT

