

Sincronização em Sistemas Distribuídos – parte 1

Raquel Mini - raquelmini@pucminas.br

Sincronização de Processos

O tempo é um problema prático importante:

- Precisamos que os computadores espalhados pelo mundo informem o horário das transações de comércio eletrônico de modo consistente
- Cada computador tem seu próprio relógio físico e, havendo vários computadores, os relógios diferem entre si e é muito difícil sincronizá-los perfeitamente
- Qual é o estado do processo A quando o processo B está em determinado ponto de sua execução

Sincronização de Processos

Sincronização externa

- Sincronizar os relógios de todos os computadores do sistema distribuído com uma fonte de tempo externa confiável e aceita por todos

Sincronização externa: para um limite de sincronização $D > 0$ e para uma fonte S de tempo UTC, a relação $|S(t) - C_i(t)| < D$, é verdadeira para $i = 1, 2, \dots, N$ e para todos os tempos reais t em I . Outra maneira de expressar isso é que os relógios C_i são *precisos* dentro do limite D .

Sincronização de Processos

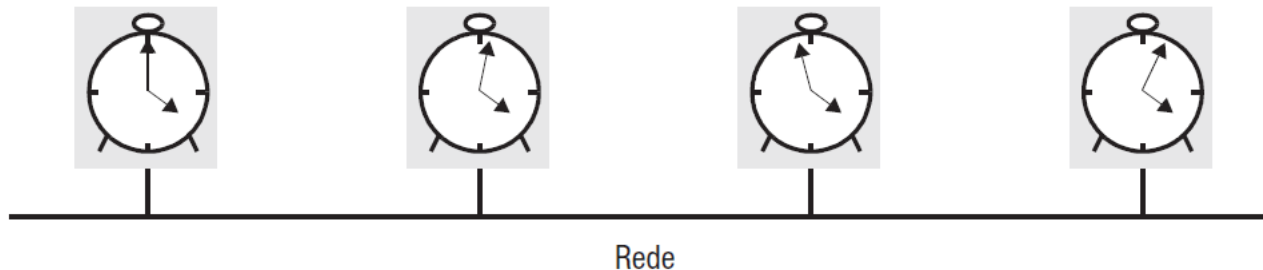
Sincronização interna

- Sincronizar os relógios de todos os computadores entre si com um grau de precisão conhecido

Sincronização interna: para um limite de sincronismo $D > 0$, a relação $|C_i(t) - C_j(t)| < D$ é verdadeira para todo $i, j = 1, 2, \dots, N$ e para todos os tempos reais t em I . Outra maneira de expressar isso é que os relógios C_i *concordam* dentro do limite D .

Sincronização de Processos

Os relógios de computador tendem a não estar em perfeito acordo



A diferença instantânea entre as leituras de quaisquer dois relógios é chamada de **desvio** (*skew*)

Os relógios baseados em cristal (usados nos computadores) estão sujeitos a **deriva de relógio** (*drift*) que significa que eles contam o tempo com diferentes velocidades e, portanto, divergem

Sincronização de Processos

Como saber se certos estados ocorreram ao mesmo tempo?

- Os relógios de computador podem ser sincronizados com fontes externas de tempo altamente precisas

Tempo oficial (padrão internacional):

- UTC: Tempo Universal Coordenado
- Fornecido por relógios atômicos
- Várias fontes: estações de rádio, satélites etc

Sincronização de Processos

Sincronização de relógios físicos

- Métodos pelos quais relógios de computador podem ser aproximadamente sincronizados, usando passagem de mensagens

Relógios lógicos

- Definir uma ordem de eventos, sem medir o tempo físico em que eles ocorreram

Sincronização de Relógios Físicos

Objetivo: manter os relógios físicos de um conjunto de máquinas “acertados”

- $|C_i - C_j| < \epsilon$, para todo i e j

Para saber em que hora do dia os eventos ocorrem nos processos de nosso sistema distribuído, é necessário sincronizar os relógios dos processos com uma fonte de tempo externa de referência

Algoritmo de Cristian

Proposto em 1989

Existem dois tipos de estações:

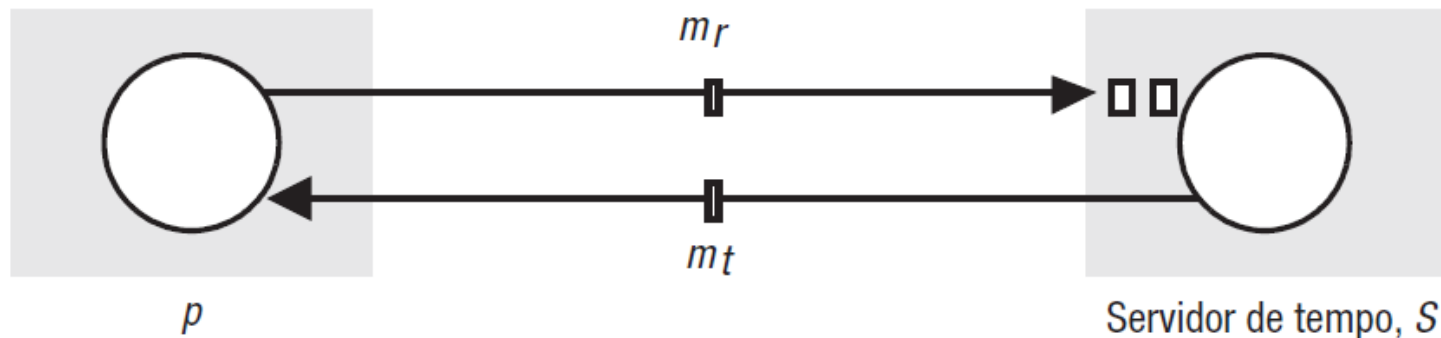
- Um servidor de tempo (*time server*)
- Clientes deste servidor de tempo

Ideia básica:

- Clientes periodicamente requisitam o tempo ao servidor
- O servidor de tempo está conectado a um dispositivo que recebe sinais de uma fonte UTC
- Ao receber uma requisição, o processo servidor S fornece o tempo, de acordo com seu relógio

Algoritmo de Cristian

Sincronização de relógio usando um servidor de tempo



Algoritmo de Cristian

Não existe um limite superior para os atrasos na transmissão de mensagens em um sistema assíncrono

- Mas, esses atrasos são razoavelmente curtos (uma pequena fração de segundo)

O método só obtém sincronização se os tempos de viagem de ida e volta observados entre cliente e servidor forem suficientemente curtos, comparados com a precisão exigida

Algoritmo de Cristian

Um processo p solicita o tempo em uma mensagem m_r e recebe o valor de tempo t em uma mensagem m_t

- t é inserido em m_t no último ponto possível antes da transmissão do computador de S

p registra o tempo de ida e volta total T_{viagem}

Uma estimativa simples do tempo para o qual p deve configurar seu relógio é $t + T_{viagem}/2$

- O tempo decorrido é dividido igualmente antes e depois de S ter colocado t em m_t

Algoritmo de Cristian

Se o valor do tempo de transmissão mínimo min for conhecido

- O mais cedo que S poderia ter colocado o tempo em m_t seria min após p ter enviado m_r
- O mais tarde que ele poderia ter feito isso seria min antes que m_t chegasse a p
- O tempo está no intervalo $[t + min, t + T_{viagem} - min]$

A variabilidade pode ser tratada fazendo-se várias requisições para S e pegando-se o valor mínimo de T_{viagem} para fazer a estimativa mais precisa de min

Algoritmo de Cristian

Se $C_{UTC} < C_{cliente}$ (ou seja, cliente está adiantado)

- Pode trazer consequências inesperadas para algumas aplicações
- Solução: “atrasar” o relógio gradativamente, até que a correção seja implementada

Algoritmo de Cristian

Cristian sugeriu que o tempo fosse fornecido por um conjunto de servidores de tempo sincronizados

O método de Cristian se destina principalmente ao uso dentro de intranets

O NTP (*Network Time Protocol*) define um protocolo para distribuir informações de tempo pela internet

NTP

Define um serviço de tempo e um protocolo para distribuir informações de tempo pela Internet

Objetivos:

- Fornecer um serviço que permita aos clientes na Internet serem sincronizados precisamente com o UTC
- Fornecer um serviço confiável que possa sobreviver a longas perdas de conectividade
- Permitir que os clientes sejam sincronizados de forma suficientemente frequente para compensar as taxas de deriva encontradas na maioria dos computadores
- Fornecer proteção contra interferência no serviço de tempo, seja mal-intencionada ou acidental

NTP

O serviço NTP é fornecido por uma rede de servidores localizados na Internet

Os **servidores primários** são conectados diretamente a uma fonte de tempo

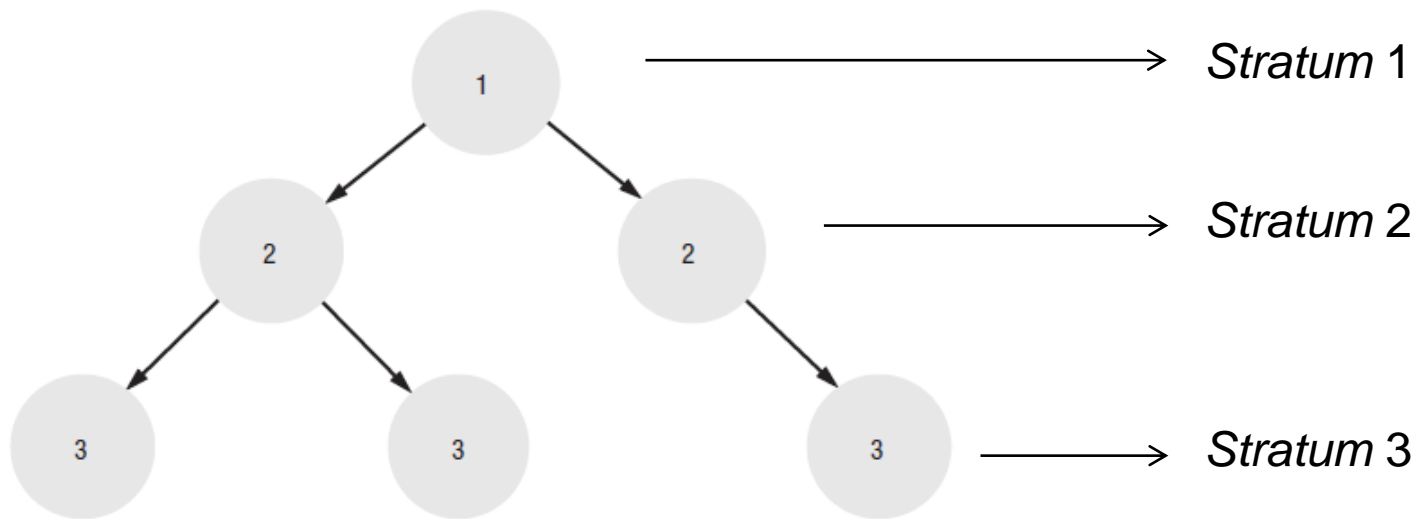
- Como um relógio de rádio recebendo UTC

Os **servidores secundários** são sincronizados com os servidores primários

Os servidores são conectados em uma hierarquia lógica chamada de subrede de sincronização

NTP

Um exemplo de subrede de sincronização em uma implementação de NTP



Os servidores de nível mais baixo (folha) são executados nas estações de trabalho dos usuários

NTP

Os relógios pertencentes aos servidores com número de *stratum* mais altos estão sujeitos a serem menos precisos do que aqueles com número de *stratum* mais baixos

A subrede de sincronização pode ser reconfigurada quando servidores se tornam inatingíveis ou quando ocorrem falhas

As mensagens NTP são enviadas de maneira não confiável, usando o protocolo de transporte UDP

NTP

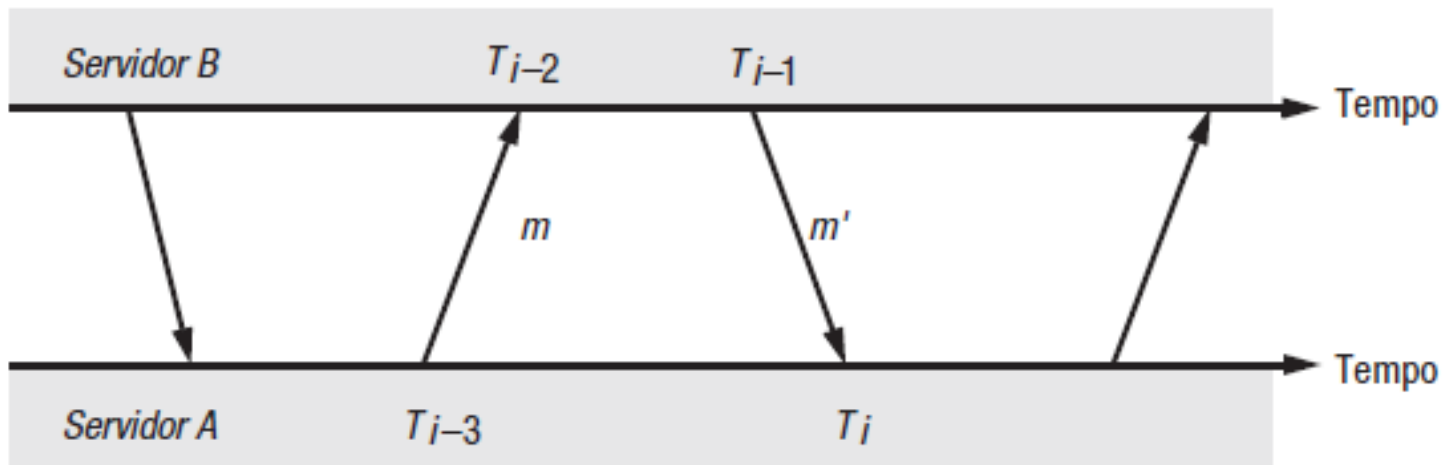
As mensagens carregam carimbos de tempo de eventos de mensagem recentes

- Os tempos locais de quando a mensagem NTP anterior entre o par foi enviada (T_{i-3}) e recebida (T_{i-2})
- O tempo local de quando a mensagem corrente foi transmitida (T_{i-1})

O destinatário da mensagem NTP anota o tempo local ao receber a mensagem (T_i)

NTP

Mensagens trocadas entre dois pares NTP



NTP

Para cada par de mensagens enviadas entre dois servidores, o NTP calcula

- Compensação c_i : estimativa da compensação real entre os dois relógios
- Atraso a_i : tempo de transmissão total das duas mensagens

Se o valor real da compensação do relógio entre B relativo à A for c e os tempos de transmissão reais de m e m' forem t e t' , respectivamente, então

$$T_{i-2} = T_{i-3} + t + c \text{ e } T_i = T_{i-1} + t' - c$$

$$a_i = t + t' = T_{i-2} - T_{i-3} + T_i - T_{i-1}$$

$$c = c_i + (t' - t)/2, \text{ onde } c_i = (T_{i-2} - T_{i-3} + T_{i-1} - T_i)/2$$

NTP

Como t e $t' \geq 0$

$$c_i - a_i/2 \leq c \leq c_i + a_i/2$$

Assim c_i é uma estimativa da compensação e a_i é uma medida da precisão dessa estimativa

Os servidores NTP aplicam um algoritmo de filtragem de dados em sucessivos pares $\langle c_i, a_i \rangle$ para fazer uma estimativa da compensação c e calcular a qualidade dessa estimativa

Uma dispersão relativamente alta representa dados relativamente não confiáveis

NTP

O NTP aplica um algoritmo de seleção de par

Os pares com números de *stratum* mais baixos são mais favorecidos do que os que possuem números de *stratum* mais altos, pois estão mais próximos das fontes de tempo principais

Os pares com a dispersão de sincronização mais baixa são relativamente favorecidos

Tarefa 4 – postar no Canvas até 21/03/2021

1. Um esquema para implementar envio de mensagens confiáveis no máximo uma vez usa relógios sincronizados para rejeitar mensagens duplicadas. Os processos colocam o valor de seus relógios locais (um “carimbo de tempo”) nas mensagens que enviam. Cada receptor mantém uma tabela fornecendo, para cada processo remetente, o maior carimbo de tempo de mensagem que observou. Suponha que os relógios estejam sincronizados dentro de 100 ms e que as mensagens podem chegar no máximo 50 ms após a transmissão.
 - a) Quando um processo pode ignorar uma mensagem contendo o carimbo de tempo T , se tiver registrado a última mensagem recebida desse processo como tendo o carimbo de tempo T' ?
 - b) Quando um receptor pode remover de sua tabela o carimbo de tempo de 175.000 (ms). (Dica: use o valor do relógio local do receptor.)
 - c) Os relógios devem ser sincronizados internamente ou externamente?

Tarefa 4 – postar no Canvas até 21/03/2021

2. Um cliente tenta sincronizar com um servidor de tempo. Ele grava os tempos de viagem de ida e volta e os carimbos de tempo retornados pelo servidor na tabela a seguir.

<i>Viagem de ida e volta (ms)</i>	<i>Tempo (hr:min:seg)</i>
22	10:54:23.674
25	10:54:25.450
20	10:54:28.342

Quais desses tempos ele deve usar para configurar seu relógio? Com que tempo ele deve ser configurado? Faça uma estimativa da precisão da configuração com relação ao relógio do servidor. Se for conhecido que o tempo entre o envio e o recebimento de uma mensagem no sistema envolvido é de pelo menos 8 ms, suas respostas mudarão?

Tarefa 4 – postar no Canvas até 21/03/2021

3. No sistema do exercício anterior, exige-se que se mantenha a sincronização do relógio de um servidor de arquivos dentro de ± 1 milissegundo. Discuta isso em relação ao algoritmo de Cristian.
4. Um servidor NTP B recebe uma mensagem do servidor A às 16:34:23.480, contendo o carimbo de tempo 16:34:13.430, e dá uma resposta. O servidor A recebe a mensagem às 16:34:15.725, contendo o carimbo de tempo de B, 16:34:25.7. Faça uma estimativa da compensação de B e A e a sua precisão.