

Sincronização de Relógios em Sistemas Distribuídos

Marcelo Ambrosio Stefanello¹, Cristian Cleder Machado¹

¹Curso de Ciência da Computação – Universidade Regional Integrada (URI)
Caixa Postal 709 – 98.400-000 – Frederico Westphalen – RS – Brasil

inf17155@uri.edu.br, cristian@cristian.com.br

Resumo. *Em um sistema distribuído, cada computador que compõe tal sistema possui sua própria estrutura de controle de tempo; não há um dispositivo centralizado que controle o tempo de todos os computadores. Baseado nessa premissa, cada computador precisa, periodicamente, ser ajustado com algum outro relógio externo, que seja precisamente confiável. Esse processo de ajuste dos temporizadores de processos, em computadores formadores de um sistema distribuído, dá-se o nome de sincronização de relógios. Este artigo propõe apresentar alguns conceitos de sincronização de relógios de maneira geral e quais os benefícios em utilizá-los.*

1. Introdução

Em sistemas distribuídos, um dos pontos que exige maior cuidado no controle é o Tempo. Isso se deve ao fato de o mesmo ser uma variável cuja medida deve ser a mais precisa possível.

Para saber, por exemplo, em que momento do dia ocorreu determinado evento em um computador específico, torna-se necessário sincronizar o relógio deste computador com algum relógio externo, que seja confiável e aceito por todos, a fim de se estabelecer a devida precisão, citada anteriormente.

A tarefa de sincronização em sistemas distribuídos não se apresenta de forma tão simples, em virtude de algumas de suas características específicas, como: distribuição das informações em múltiplos pontos ao longo da rede do sistema; processos tomam decisões amparados pelas informações locais; e o relógio global pode não existir ou ser impreciso.

Infelizmente, a sincronização em sistemas distribuídos tende a ser bem mais complexa e trabalhosa se comparada à sincronização em sistemas centralizados, onde, quando um determinado processo deseja saber a hora, ele simplesmente realiza uma chamada de sistema, e o núcleo responde. Há também o fato de não ser viável existir um único processo, centralizado, que controle o tempo de todo o conjunto de computadores, e que tome todas as decisões referentes aos processos correntes. Uma falha neste processo poderia comprometer todo o funcionamento do sistema distribuído. Dessa forma, esse seria considerado um ponto do sistema não confiável.

Admite-se, então, que cada unidade formadora do sistema tenha seu próprio relógio local, mas que possa ser sincronizado a um relógio padrão. E é essa sincronização de relógios que vem como tema norteador do artigo.

2. Considerações Gerais Sobre o Tempo

Apesar de parecer um tanto simples, medir o tempo de forma precisa é uma tarefa que envolve constantes estudos e alta tecnologia. A hora terrestre, por exemplo, é medida usando-se duas formas de contagem de tempo distintas, porém complementares: o modo astronômico e o modo atômico.

A hora astronômica baseia-se num evento chamado “trânsito solar”, ou seja, é a passagem do Sol pelo ponto aparentemente mais alto no céu. O dia solar, então, é o intervalo entre dois trânsitos solares consecutivos. Como há 86.400 segundos em 1 dia, o segundo solar é definido exatamente como $1/86.400$ de um dia solar. (TANENBAUM, STEEN, 2006, p. 142)

A segunda forma de contagem do tempo, citada anteriormente, é a hora atômica. Esse método se baseia no número de transições que o átomo de Césio 133 faz em um segundo – 9.192.631.770 transições.

“Hoje, vários laboratórios ao redor do mundo têm relógios de Césio 133, e cada um deles informa periodicamente ao Bureau International de l'Heure (BIH), em Paris, quantas vezes seu relógio pulsou. O BIH calcula a média desses valores e produz a hora atômica internacional (International Atomic Time) ou TAI.” (TANENBAUM, STEEN, 2006, p. 142)

Infelizmente a hora TAI não é perfeitamente equivalente à hora solar: 86.400 segundos TAI são equivalentes a cerca de 3 milissegundos (ms) a menos do que um dia solar. A fim de resolver esse problema, o BIH passou a inserir segundos extras sempre que houver uma defasagem de 800 ms entre a hora solar e a TAI. Esse processo forma a hora coordenada universal (UTC, no francês), que serve de base para a hora civil mundial, atualmente.

“O Tempo Universal Coordenado – abreviado como UTC (do seu equivalente em francês) – é um padrão internacional para contagem de tempo. Ele é baseado no tempo atômico, mas ocasionalmente é inserido – ou, mais raramente excluído – o conhecido segundo bissexto, para manter a sincronização com o tempo astronômico.” (COULOURIS, 2007, p. 382)

3. Relógios Físicos

Os computadores atuais possuem mecanismos para controle do tempo. São chamados comumente de “relógios”, ou mais corretamente, “temporizadores”.

“Um temporizador de computador usualmente é um cristal de quartzo lapidado e usinado com precisão. Quando mantido sob pressão, cristais de quartzo oscilam a uma frequência bem definida que depende do tipo de cristal, de como ele foi lapidado e da magnitude da tensão.” (TANENBAUM, STEEN, 2006, p. 141).

Cada cristal possui dois registradores: um contador e um retentor. O registrador contador vai decrementando uma unidade a cada oscilação do cristal. Já o registrador retentor recarrega o contador cada vez que este chega a zero (quando ocorre uma interrupção). Essa interrupção é chamada de “ciclo de relógio”.

Em um único computador, não há nenhum problema crítico em ter o temporizador local com horário um pouco defasado, pois todos os processos irão requisitar informações de tempo do mesmo relógio, assim será mantida a consistência. Mas ao contrário, quando há várias CPUs, cada qual com seu próprio mecanismo controlador de tempo, é comum haver uma defasagem de relógio. Isso ocorre pelo fato de ser impossível que todos os cristais localizados nos vários computadores trabalhem numa frequência exatamente igual. Tal fato implica em saída de sincronia dos temporizadores, e conseqüentemente, obtenção de diferentes valores na leitura.

3.1. Método de Cristian para Sincronização de Relógios

O método de Cristian – 1989 – consiste em fazer uso de servidores de tempo, que utilizem dados UTC. No entanto, esse processo gera uma defasagem em virtude da troca de mensagens entre cliente e servidor.

Tanenbaum (2006, p. 145) cita um exemplo de como Cristian resolveu esse problema de defasagem: o processo A envia uma requisição a B, com uma marca de tempo T1. B registrará a hora em que recebeu a mensagem (T2, obtido pelo seu relógio local), e retornará uma resposta com uma marca de tempo T3, enviando também o valor T2 (horário de recebimento da mensagem T1). Por fim, A registra a hora de chegada da resposta T4. Esses valores de tempo são utilizados pelo cliente para descontar o tempo de viagem, e assim assumir um valor mais próximo possível da realidade.

Esse método, como descrito até agora, possui outro problema: haver somente um servidor UTC.

“[...] o método de Cristian sofre do problema associado a todos os serviços implementados por um único servidor: o de que o único servidor de tempo pode falhar e, assim, tornar a sincronização temporariamente impossível. Por isso Cristian sugeriu que o tempo fosse fornecido por um conjunto de servidores de tempo sincronizados, cada um com um receptor de sinais de tempo UTC.” (COULOURIS, 2007, p. 385).

3.2. Algoritmo Berkeley

Segundo Coulouris (2007, p. 385 – 386), em conjuntos de computadores executando um algoritmo Berkeley, um computador é escolhido para atuar como coordenador (mestre). O mestre, então, faz consultas sequenciais periodicamente nos outros computadores (escravos) cujos relógios devem ser sincronizados. Os computadores escravos retornam seus valores de relógio, e o mestre faz uma estimativa dos tempos locais desses relógios, levando em consideração os tempos de ida e volta. Por fim, faz a média dos valores obtidos, incluindo o valor de seu próprio relógio.

O diferencial deste algoritmo é que o mestre envia o valor individual de ajuste para cada relógio escravo, em vez de simplesmente enviar para todos os computadores o tempo corrente atualizado, o que geraria mais incerteza, devido aos atrasos de troca de mensagens. (COULOURIS, 2007)

4. Relógios Lógicos (Relógio de Lamport)

Lamport – 1978 – lançou uma teoria sobre sincronização de relógios, em que diz que mesmo sendo possível essa sincronização, há casos em que ela não necessariamente tem de ser absoluta, ou exata.

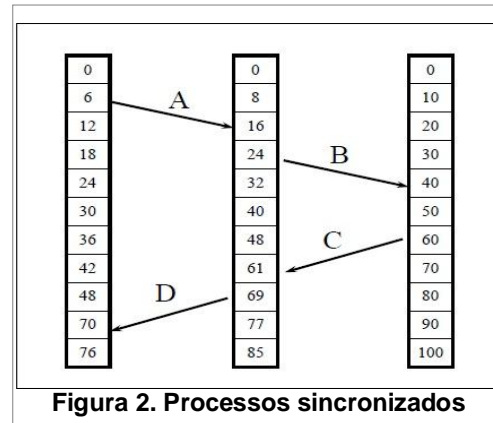
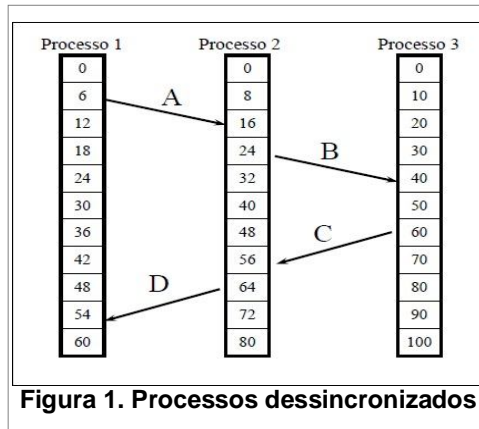
“Se dois processos não interagirem, não é necessário que seus relógios sejam sincronizados porque a falta de sincronização não seria observável e, portanto, não poderia causar problemas.” (TANENBAUM, STEEN, 2006, p. 148)

Lamport sugere que o importante é que os processos concordem com a ordem em que ocorrem os eventos, e não com a hora exata dos relógios. Assim, estabeleceu uma relação chamada “acontece antes”. Isso significa que, por exemplo, primeiro ocorre um evento A, e após, um evento B. Essa relação pode ser percebida em duas situações: quando um evento ocorre antes do outro; e no caso de uma mensagem, cujo o envio se dá antes do recebimento por parte do outro processo (uma mensagem nunca será recebida antes de ser enviada).

Exemplo da solução de Lamport: a Figura 1 representa três processos em execução, cada um em uma máquina diferente, com seu próprio relógio, que opera na sua própria velocidade, à uma taxa constante, porém diferentes uma da outra. Basta observar que, quando o relógio do processo 1 (p1) pulsa 6 vezes, o relógio do processo 2 (p2) pulsou 8, e assim por diante.

No exemplo, A, B, C e D representam mensagens que são transmitidas entre os processos. Pode-se observar que a mensagem A sai de p1 no pulso 6, e chega até p2 no

pulso 16. Se a mensagem transportar o valor do pulso de p1, p2 saberá que a mesma demorou teoricamente 10 pulsos para chegar chegar até ele. Da mesma forma ocorre com a mensagem B entre p2 e p3. Até aí tudo dentro da normalidade. No entanto, o grande problema ocorre na resposta das mensagens, ou seja, C e D. Percebe-se que a mensagem C sai de p3 no instante 60, e chega a p2 no instante 56. Isso não é possível, pois um fato não pode iniciar em um determinado momento e terminar em um momento anterior. É nesse tipo de situação que a solução de Lamport é usada, baseada no conceito “acontece antes”.



Como a mensagem transporta o tempo de envio do remetente, ao chegar em p2 o processo perceberá que o pulso da chegada é anterior ao tempo de envio na saída. Então o receptor adianta seu relógio para ficar uma unidade à frente do tempo de envio, nesse caso, 61. Semelhante a isso ocorre com a mensagem D. O estado final dos relógios dos processos envolvidos no exemplo ficam como mostrado na Figura 2.

5. Conclusão

O estudo realizado neste artigo mostra, como tema principal, o quão importante são a sincronização e a medição precisa do tempo, especialmente em computação distribuída. “Precisa” não no sentido de perfeição, mas sim no que tange a ordem cronológica dos fatos, no caso, eventos de processos, como demonstrado há pouco, no exemplo da solução de Lamport para relógios lógicos.

Foram apresentados também, estudos sobre outros algoritmos de otimização de sincronização, como Berkeley e o método de Cristian, ambos direcionados para relógios físicos. Além disso, o assunto tratado abre outras discussões, além do tema central - sincronização de relógios - mas também em questões de melhora de desempenho, como citado a defasagem de tempo no processo de troca de mensagens, e até mesmo instiga pesquisas sobre as medidas de tempo astronômico e atômico, que atualmente definem o horário padrão mundial, e melhoras na exatidão dos atuais relógios, principalmente dos computadores que usamos.

Referências

- Tanenbaum, A.S. e Steen M.V. (2006) “Sistemas Distribuídos Princípios e Paradigmas”, 2ª Edição, Editora Person Education, 686 p.
- Coulouris, G. et al. (2007) “Sistemas Distribuídos Conceitos e Projetos”, 4ª Edição, Porto Alegre, Editora Bookman, 780 p.

Universidade Federal de Santa Maria, “Sincronização em Sistemas Distribuídos”
Disponível em: <<http://www-usr.inf.ufsm.br/~ceretta/elc1018/sincronizacao.pdf>>,
Acesso em: 15 set. 2012.