CENTRO UNIVERSITÁRIO INSTITUTO DE EDUCAÇÃO SUPERIOR DE BRASÍLIA

TENÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

SINCRONISMO EM SISTEMAS DISTRIBUÍDOS

GERALDO LÚCIO CARVALHO DOS SANTOS - 2424290069

RAPHAEL FERRAZ CAMELO TAVEIRA - 2424290071

FERNANDO HENRIQUE DOS SANTOS FORTES - 2424290074

Brasília

2024

GERALDO LÚCIO CARVALHO DOS SANTOS
RAPHAEL FERRAZ CAMELO TAVEIRA
FERNANDO HENRIQUE DOS SANTOS FORTES
ADSNM1B-ADS042

SINCRONISMO EM SISTEMAS DISTRIBUÍDOS

Trabalho apresentado como complementação de nota para a aprovação na disciplina de Redes de Computadores e Sistemas Distribuídos

Orientador: Prof. Me. Carlos Henrique Bacellar Bon.

Brasília

2024

RESUMO

Será abordado brevemente sobre como alguns algoritmos de sincronização atuam em um sistema distribuídos. Abordando sobre clock físico e clock lógico, algoritmo de exclusão múltipla, de Lamport, Bully, Cristian e Berkeley.

Palavras-chave: Sincronização, Clock, Algoritmo.

ABSTRACT

It will be briefly discussed how some synchronization algorithms work in a distributed system. Addressing physical clock and logical clock, multiple exclusion algorithm, by Lamport, Bully, Cristian and Berkeley.

Keywords: Synchronization, Clock, Algorithm.

SUMÁRIO

Introdução

Para o bom funcionamento de um sistema, é necessário que todos os dispositivos estejam no mesmo tempo., isto é, que trabalhem usando o mesmo tempo. Pois caso haja uma falha entre os dispositivos, um pequeno atraso ou adiantamento no relógio um do outro, os processos não irão ocorrer da maneira correta.

Quando um processo deseja saber a hora, ele apenas se direciona para o núcleo do computador e ele dirá. Porém quando se há mais de um computador na jogada, a situação muda. É necessário que haja uma intervenção externa para poder sincronizar os dispositivos entre eles.

Imagine a situação onde ocorra um erro em um computador do setor de finanças. Então o responsável pelos problemas técnicos aparece e a primeira coisa que ele faz é verificar os *logs*. Caso a hora do computador seja diferente a do servidor, não será possível determinar o que o usuário estava fazendo e em que período do dia.

Por esse motivo que a sincronização em um sistema distribuído é tão importante. Ele é um dos pilares para que haja uma boa relação entre os dispositivos.

Antes do sincronismo

Antes do sincronismo, os sistemas distribuídos funcionavam de forma bastante limitada vejam algumas situações;

Isolamento

Cada computador trabalha de forma isolada, praticamente sem interação com outros sistemas. Dificultando por exemplo uma simples transferência de arquivos. Se você precisasse enviar um documento, contrato, ou seja, um texto qualquer isso deveria ser feito de forma não virtual, ou seja, por impressos, cartas, etc.

Mensagens manuais

A troca de informações entre os sistemas eram de forma manual. Por exemplo se fosse necessário enviar um arquivo de um computador para outro era necessária uma mídia física, como um disquete, por exemplo, onde se você precisasse transferir um arquivo de texto de um computador para outro seria necessário gravar as informações neste disco para que ele fosse posteriormente lido ou gravado na outra máquina. Muitas vezes até mesmo pela limitação de espaço nestes discos eram necessárias diversas unidades gravadas para transferir um arquivo de um computador para o outro.

Baixa eficiência

Devido a falta de coordenação, conectividade os sistemas enfrentavam dificuldades em tarefas tanto simples quanto complexas como manter um relógio e data sincronizadas. Imagine dentro de uma empresa com 10 computadores cada um poderia apresentar uma hora e data diferentes.

Clock lógico

Quando se fala de um sistema, deve ser decidido uma coisa: ele terá que trabalhar com dispositivos externos? Se a resposta for “não”, a sincronização deverá ocorrer apenas entre eles, naquele sistema interno. Ou seja, o tempo real não irá importar muito.

Clock físico

Descrevendo brevemente, o temporizador de um computador é determinado pela oscilação de um cristal de quartzo a uma determinada pressão.

Cada computador possui o próprio relógio. É ele quem determina o fluxo temporal naquela máquina em específica. Porém quando há mais de uma UCP (Unidade Central de Processamento), não pode haver uma grande defasagem de tempo entre eles.

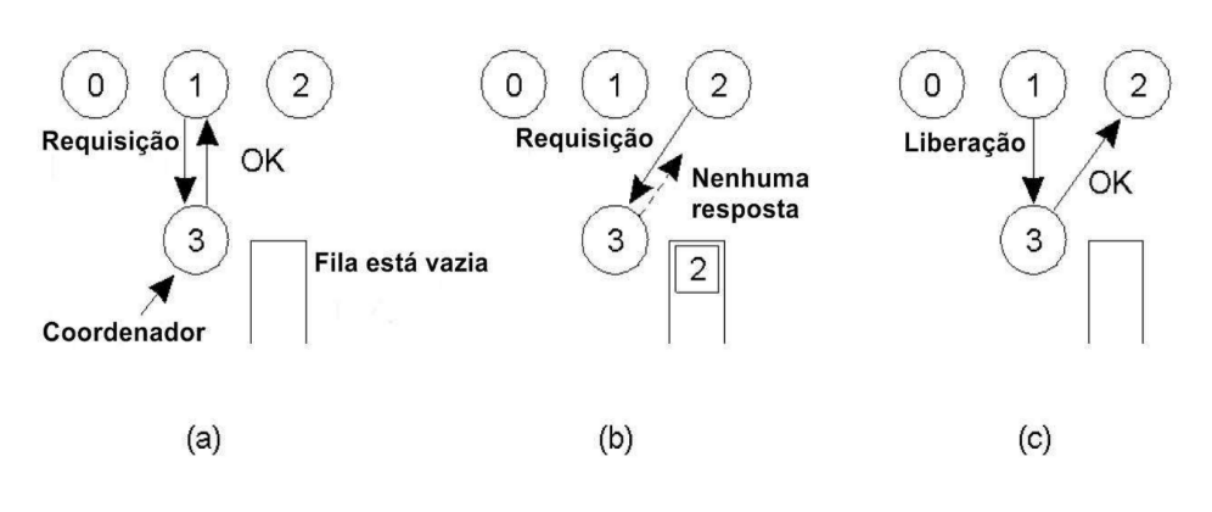
Mas cada um desses computadores, com seus próprios cristais de quartzo, pode, e vão, oscilar em frequências ligeiramente diferente. E é aí que entra os diversos métodos de sincronismo.

Algoritmo de Exclusão Mútua (1965)

O algoritmo da exclusão mútua foi desenvolvido por Edsger W. Dijkstra, em 1965. Ele diz que quando há o compartilhamento de um recurso em um sistema distribuídos, é escolhido um computador para ser o coordenador.

Agora sempre que um computador desejar usar um determinado recurso, ele deverá enviar um sinal ao coordenador perguntando se a vaga está disponível. Caso esteja, será enviado um OK liberando o acesso.

Porém caso a vaga esteja sendo utilizada, o requisitante receberá mensagem alguma do coordenador, assim indo para a fila. E assim que ele liberar o acesso novamente o requisitante em primeiro lugar da lista tomará uso.



Algoritmo de Lamport (1978)

Entre os anos de 1970 e 1980, Leslie Lamport apresentou a ideia de relógios lógicos que não medem o tempo real e isto foi essencial para implantação do sincronismo.

Leslie Lamport também desenvolveu uma solução para a diferença de tempos entre computadores. Aqui não há a necessidade de se eleger um coordenador. Todo o processo é realizado pelos próprios membros do sistema.

Cada processo em um sistema distribuído mantem um contador que incrementa a cada evento que ocorre no processo como exemplo o envio de uma mensagem. Quando um processo envia uma mensagem ele inclui um valor do seu contador. Quando o destinatário recebe a mensagem, ele ajusta seu próprio contador para garantir que ele é maior que o contador do remetente.

Primeiro cenário: o P1 (P = processo) envia um sinal A para o P2, sinalizando o momento em que foi enviado. Nesse caso, T6. Então P2 registra o momento de recebimento como T16. Agora P2 envia um sinal B para P3 com o tempo T24. P3 recebe B com o tempo T40

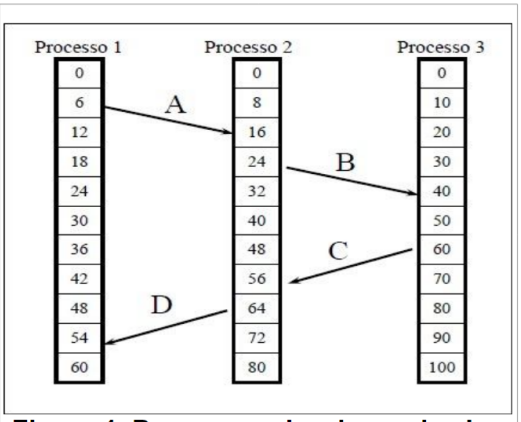


Figura 1. Processos dessincronizados

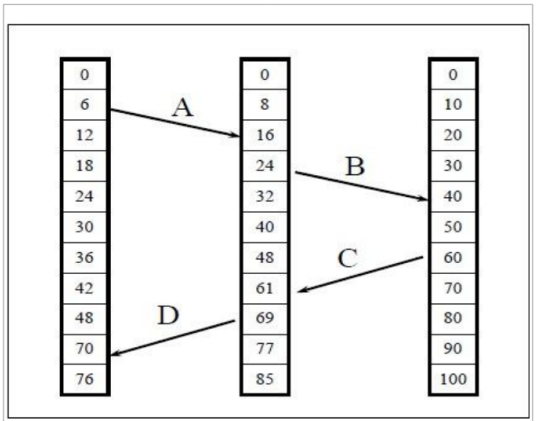


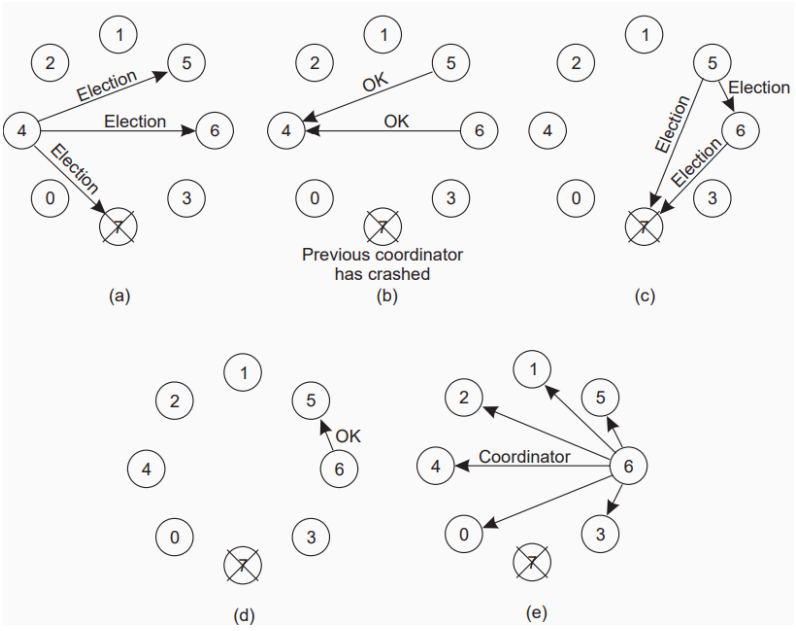
Figura 2. Processos sincronizados

Algoritmo de Bully (1982)

Desenvolvido em 1982 por Hector Garcia-Molina, nos Estados Unidos. Esse algoritmo funciona da seguinte forma: temos um processo como o coordenador. Ele quem gerencia o sistema. Mas caso esse processo entre em modo offline, um novo coordenador precisa ser eleito.

Ao notar que não há um coordenador, um processo qualquer pode iniciar uma votação ao enviar a mensagem ELECTION para todos os outros computadores. Caso nenhum outro processo de maior força responde, este quem enviou a mensagem se tornará o novo coordenador. Mas caso outro processo maior responda, este tomará o lugar.

A força de cada processor se dá pelo identificado único (PID) que cada um recebe, definido pelo escalonamento de processos. Quanto maior é esse número, maior é a “força” deste processo.



Algoritmo de Cristian (1989)

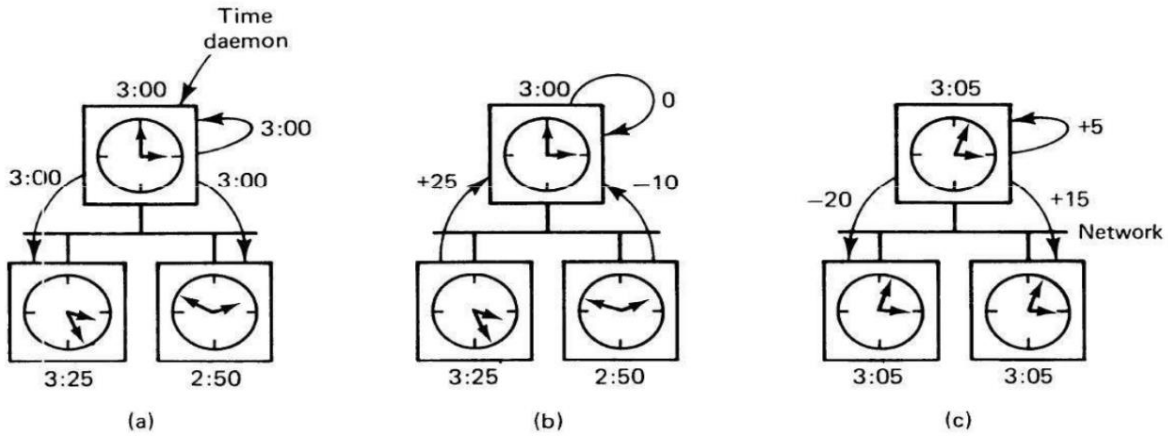
Esse método define uma máquina como um servidor de tempo no qual todas as outras máquinas da rede irão consultar para definirem a hora correta de seus relógios.

Essa máquina possui um fornecedor externo, um relógio atômico super preciso. Porém essa precisão não resolve todos os problemas. O tempo que leva da requisição do tempo oficial até a devida entrega, pode criar uma diferença nessa entrega. Esse é um dos pontos negativos desse algoritmo.

Algoritmo de Berkeley (1989)

Por fim, o algoritmo de Berkeley também possui um servidor de tempo, definido como o mestre. Nesse lado o mestre não se localiza externamente, é de dentro da própria rede. Então funciona da seguinte forma: o mestre possui uma determinada hora, que pode ser fornecida por um relógio atômico. Periodicamente as máquinas, os escravos, recorrem ao mestre para corrigir o horário. Nessa situação o servidor recebe todos os tempos e realiza um cálculo para definir uma hora conveniente a todos.

A hora de um pode ser maior e a hora de outro pode ser maior. Para resolver isso o servidor entrega a hora correta que devem definir. Claro esse mecanismo também leva em consideração o atraso que leva até a mensagem chegar ao dispositivo.



Protocolos de comunicação

Com o desenvolvimento de protocolos de comunicação como TCP/IP em torno dos anos 80 e 90, permitiu que sistemas distribuídos se comunicassem de forma mais eficiente. Permitiu a troca de dados de forma eficiente.

Conclusão

Podemos concluir que é uma necessidade fundamental, permitindo que diferentes componentes trabalhem em harmonia, mesmo estando fisicamente separados. Deixando de ser ineficaz com computadores trabalhando de forma isolada e iterações manuais que restringiam a transferência de dados e eficácia.

É possível notar que há diversos mecanismos para resolver um mesmo problema: a sincronização entre dispositivos (não necessariamente apenas temporal). Podemos ter máquinas centralizadas para coordenar um sistema. A falta de sincronia pode ter impactos severos na harmonia desse sistema.

Cada tipo oferece vantagens específicas e aplicações particulares, desde a precisão do sincronismo temporal usando relógios atômicos até a complexidade dos algoritmos de consenso necessários para a validação de transações em redes blockchain.

Resumindo, a evolução do sincronismo em sistemas distribuídos permitiu avanços significativos na forma como os computadores trabalham juntos, resultando em maior eficiência, confiabilidade e escalabilidade. Esses avanços possibilitaram a criação de aplicações modernas que dependem de uma operação coordenada e precisa, beneficiando empresas e usuários ao redor do mundo.

Em um mundo cada vez mais conectado e em constante desenvolvimento estudar e aprimorar essas tecnologias não apenas garante um funcionamento eficiente, mas também molda o futuro da inovação.

Referências bibliográficas

STEFANELLO, Marcelo Ambrosio; MACHADO, Cristian Cleder. **Sincronização de Relógios em Sistemas Distribuídos. Anais do Encontro Anual de Tecnologia da Informação**, v. 2, n. 1, p. 281-281, 2012.

VAN STEEN, M.; TANENBAUM, A. S. **Sistemas Distribuídos: princípios e paradigmas (Segunda Edição)**. Londres, England: Pearson Education, 2006.

Sincronização em Sistemas Distribuídos. Disponível em: <https://www.cin.ufpe.br/~avmm/arquivos/provas%20software/resuminho3.pdf>.

KUROSE, J. F.; ROSS, K. W. **Redes De Computadores E A Internet Uma Abordagem Top Down**. Londres, England: Pearson, 2021.

Aulas IESB – Aulas ministradas pelo professor CARLOS HENRIQUE BACELLAR BOM.