

Sistemas Distribuídos

Coordenação Distribuída

7

Universidade Estácio de Sá
Professor Welsing M. Pereira
www.professorwelsing.webnode.com



Coordenação Distribuída



- Introdução
 - Os processos distribuídos frequentemente precisam coordenar suas atividades.
 - Se um conjunto de processos compartilham um recurso, ou uma coleção de recursos, então, frequentemente, a exclusão mútua é exigida para evitar interferência e garantir a consistência ao acessar esses recursos.
 - Esse problema é denominado sessão crítica (região crítica).
 - Que em Sistemas distribuídos para resolvê-lo precisamos de uma solução para a exclusão mútua distribuída.

2

Coordenação Distribuída



- Algoritmos de exclusão mútua
 - Consideremos um sistema de N processos p_i , $i = 1, 2, 3, \dots, N$.
 - Os processos acessam recursos comuns, mas fazem isso em uma sessão crítica.
 - Por simplicidade, admitamos que existe apenas uma seção crítica.
 - Suponhamos que o sistema seja assíncrono, que os processos não falham e que o envio de mensagens é confiável.

3

Coordenação Distribuída



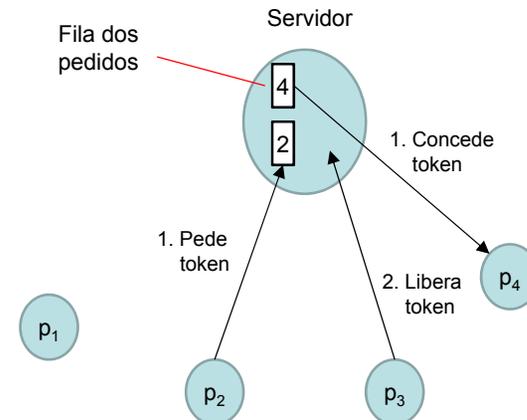
- Algoritmos de exclusão mútua
 - O protocolo em nível de aplicativo para executar uma seção crítica é o seguinte:
 - `enter()` //entra na seção crítica – bloqueia se necessário.
 - `resourceAcesses()` // acessa recurso compartilhado na seção crítica.
 - `exit()` // sai da seção crítica – outros processos podem entrar agora.

4

- Algoritmos de exclusão mútua
 - Requisitos básicos para entrar na seção crítica.
 - EM1(segurança)
 - No máximo um processo por vez pode ser executado na seção crítica.
 - EM2 (subsistência)
 - Os pedidos para entrar e sair têm sucesso. Ausência de inanição (starvation) e de impasse (deadlock).
 - EM3(ordenação)
 - Se um pedido para entrar na seção crítica aconteceu antes de outro, então a entrada na seção crítica é garantida nessa ordem.
 - Enquanto existir pedido que ainda não entrou na seção crítica um pedido não pode acessá-la mais de uma vez.

- O algoritmo do servidor central
 - Um servidor concede o direito de um processo acessar a seção crítica.
 - Para entrar na sessão crítica um processo envia uma mensagem de pedido para o servidor e espera uma resposta.
 - Conceitualmente, a resposta constitui um **token** significando permissão para entrar na seção crítica.
 - Se nenhum outro processo tiver o token no momento do pedido, então o servidor responderá imediatamente, concedendo o token.
 - Se o token estiver de posse de outro processo, então o servidor não responderá, mas enfileirá o pedido.
 - Na saída da seção crítica, uma mensagem é enviada para o servidor, devolvendo o token a ele.

- O algoritmo do servidor central
 - Se a fila de processos em espera não estiver vazia, o servidor escolherá a entrada mais antiga, a removerá e responderá para o processo correspondente. Então o processo escolhido terá a posse do token.



Coordenação Distribuída



- O algoritmo do servidor central
 - Dada nossa suposição de que não ocorrem falhas, é fácil ver que as condições de segurança e subsistência são satisfeitas por esse algoritmo.
 - Entretanto, não satisfaz a propriedade EM3.

9

Coordenação Distribuída



- O algoritmo do servidor central
 - Desempenho
 - Para obter a entrada na seção crítica:
 - É exigido apenas 2 mensagens (um pedido e uma concessão).
 - Consumo da largura de banda.
 - Apenas quando há solicitação ou liberação da seção crítica pelos processos.
 - O atraso experimentado por um processo que esteja solicitando a entrada na seção crítica está entre:
 - O tempo de ida e volta de uma mensagem (uma mensagem de pedido seguido de uma concessão).
 - Sair da região crítica:
 - Exige apenas uma mensagem (liberação).
 - O atraso de sincronização entre a saída de um processo da seção crítica e a entrada do processo seguinte:
 - O tempo de ida e volta de uma mensagem (mensagem de liberação e uma de concessão para o próximo processo).

10

Coordenação Distribuída



- Um algoritmo baseado em anel
 - Uma das maneiras mais simples de construir a exclusão mútua entre os N processos, sem exigir um processo adicional, é organizá-los em um anel lógico.
 - Isso exige apenas que cada processo p_i tenha um canal de comunicação com o processo seguinte no anel, $p_{(i+1) \bmod N}$.
 - A idéia é que a exclusão seja concedida pela obtenção de um token, na forma de uma mensagem passada de processo para processo, em uma única direção (digamos, no sentido horário) em torno do anel.

11

Coordenação Distribuída



- Um algoritmo baseado em anel
 - Se um processo não pede para entrar na seção crítica ao receber o token, então ele encaminha imediatamente o token para seu vizinho.
 - Um processo que solicite o token espera até recebê-lo, mas o mantém.
 - Para sair da seção crítica, o processo envia o token para seu vizinho.
 - As condições EM1 e EM2 são satisfeitas por esse algoritmo, mas o token nem sempre é obtido primeiramente por quem o deseja primeiro (não satisfazendo EM3)

12

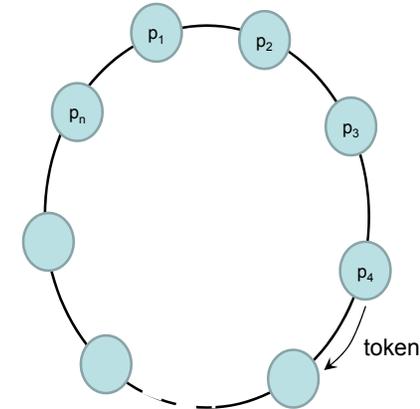
Coordenação Distribuída



- Um algoritmo baseado em anel
 - Desempenho
 - Para obter a entrada na seção crítica:
 - É exigido de 0 a N transmissões de mensagem (troca do token).
 - Consumo contínuo da largura de banda.
 - Mensagens com a passagem do token são enviadas a todo instante.
 - Exceto quando um processo está dentro da seção crítica.
 - O atraso experimentado por um processo que esteja solicitando a entrada na seção crítica está entre:
 - O tempo de 0 a N mensagens.
 - 0 quando ele acabou de receber o token.
 - O tempo de N mensagens quando ele acabou de passar o token.
 - Sair da região crítica:
 - Exige apenas uma mensagem (a passagem do token à frente).
 - O atraso de sincronização entre a saída de um processo da seção crítica e a entrada do processo seguinte:
 - Está entre 1 e N transmissões de mensagem.

13

Coordenação Distribuída



14

Coordenação Distribuída



- Um algoritmo usando multicast e relógios lógicos
 - Processos que solicitam a entrada em uma seção crítica difundem seletivamente (multicast) uma mensagem de pedido e só podem entrar nela quando todos os outros processos tiverem respondido a mensagem.
 - As condições sob as quais um processo responde a um pedido são projetadas de forma a garantir que as condições EM1, EM2 e EM3 sejam satisfeitas.
 - Os processos p1, p2, ..., pn apresentam identificadores numéricos distintos.
 - Presume-se que eles possuam canais de comunicação de uma para o outro e que cada processo pi mantenha um relógio atualizado.
 - As mensagens que solicitam entrada são da forma $\langle T, p_i \rangle$, onde T é a indicação de tempo do remetente e pi é o identificador do remetente.

15

Coordenação Distribuída



- Um algoritmo usando multicast e relógios lógicos
 - Cada processo registra seu estado de estar *fora* (**RELEASED**) da seção crítica, querendo *entrar* (**WANTED**) ou *estar* (**HELD**) na seção crítica, em uma **variável de estado**.
 - Se um processo solicita entrada, e o estado dos outros processos é **RELEASED**, então todos responderão imediatamente ao pedido e o solicitante obterá a entrada.
 - Se algum processo estiver no estado **HELD**, então este processo não responderá aos pedidos até que tenha terminado com a seção crítica; portanto, o solicitante não poderá entrar nesse meio tempo.
 - Se dois ou mais processos solicitam a entrada ao mesmo tempo, o pedido do processo que apresentar a identificação de tempo mais baixa será o primeiro a coletar N-1 respostas, garantindo a próxima entrada.

16

- Um algoritmo usando multicast e relógios lógicos
 - Se os pedidos apresentarem indicações de tempo iguais, serão ordenados de acordo com os identificadores correspondentes dos processos.
 - Note que, quando um processo solicita a entrada, ele retarda o processamento dos pedidos de outros processos até que seu próprio pedido tenha sido enviado e ele tiver gravado a indicação de tempo T do pedido.

- Um algoritmo baseado em anel
 - Desempenho
 - Para obter a entrada na seção crítica:
 - É exigido $2(N - 1)$ mensagens: $N - 1$ para o pedido + $N - 1$ para respostas.
 - Consumo contínuo da largura de banda.
 - Mensagens são enviadas apenas quando há o desejo de acesso à seção crítica
 - O atraso experimentado por um processo que esteja solicitando a entrada na seção crítica está entre:
 - O tempo de ida (pedido) e volta (resposta) de uma mensagem.
 - Sair da região crítica:
 - Exige apenas uma mensagem.
 - O atraso de sincronização entre a saída de um processo da seção crítica e a entrada do processo seguinte:
 - É o tempo da transmissão de uma mensagem.

- Um algoritmo usando multicast e relógios lógicos

Exemplo:
 Considere uma situação envolvendo três processos, p1, p2 e p3. Vamos supor que p3 não esteja interessado em entrar na seção crítica e que p1 e p2 solicitam a entrada correspondente. A indicação de tempo do pedido de p1 é 41 e a de p2 é 34. Quando p3 recebe seus pedidos, responde imediatamente. Quando p2 recebe o pedido de p1, verifica que seu próprio pedido tem a indicação de tempo mais baixa e, portanto, não responde, detendo p1. Entretanto, p1 verifica que o pedido de p2 tem uma indicação de tempo mais baixa do que a de seu próprio pedido e, portanto, responde imediatamente. Ao receber essa segunda resposta, p2 pode entrar na seção crítica. Quando p2 sair da seção crítica, responderá ao pedido de p1 e, portanto, garantirá sua entrada.

