

Exercícios de Fixação – Chapter 06

Respostas

- 1) Hardware clock é o conjunto de componentes físicos, mecânicos e eletrônicos que geram, contam e armazenam clock ticks na forma de um temporizador (timer). Software clock são as rotinas e estruturas de dados que, por meio das interrupções (IRQs) geradas pelo hardware clock, mantêm a informações de data e hora atualizadas e disponíveis para os processos de aplicação; sua operação ocorre no contexto do sistema operacional. Os logical clocks, também implementados via software, são implementados pelo middleware para suporte aos processos distribuídos, em especial para se controlar a ordenação global de eventos distribuídos.
- 2) O NTP é baseado em uma hierarquia de servidores, onde cada nível da hierarquia é denominado de Stratum. Normalmente, são implementados 4 Stratums, onde os servidores do Stratum 1 (de mais alto nível) obtêm sua hora por meio de sinais provenientes de UTC receivers. Cada par de servidores NTP, seja no mesmo Stratum ou em Stratums diferentes, trocam informações de sincronização. Inicialmente, eles computam o offset entre os nós por meio do Algoritmo de Christian, resultando no valor de teta, o qual serve para indicar a defasagem entre os relógios de ambos os servidores. Além disso, eles calculam o delta que indica o delay médio entre os servidores, o que serve para ajustar os timestamps durante a sincronização entre eles. A posição na hierarquia determinará qual servidor ajustará o seu relógio ao do outro, onde o servidor de Stratum maior sempre ajusta seu relógio ao servidor de Stratum menor.
- 3) Com base na figura indicada, todos os processos inicializam seus relógios lógicos para o valor zero. A cada evento local, seu contador (relógio) troca de valor pelo incremento definido para aquele sistema. No caso, a figura indica os incrementos 6 (P1), 8 (P2) e 10 (P3). A cada troca de mensagem, caso haja uma defasagem entre os relógios, de forma que o relógio do processo destinatário tenha um “tempo lógico” anterior ao “tempo lógico” do emissor, então o relógio lógico do destinatário é atualizado para o valor do timestamp + 1. Por exemplo, P2 quando recebeu m3 de P3 atualizou o seu relógio lógico de 56 para 61, já que o timestamp de saída da mensagem em P3 foi 60. Desta forma, sempre o relógio do processo que recebe terá um valor superior ao valor lógico do relógio de Lamport do emissor, garantido assim a possibilidade de ordenação global dos eventos, neste caso, de envio e recebimento de mensagens.
- 4) Não permitem estabelecer qualquer relações entre dois eventos pela comparação dos seus relógios lógicos individuais, $C(i)$. Por exemplo, não é possível saber se um evento precede causalmente o outro apenas olhando nos seus respectivos relógios lógicos. Exemplo ilustrado no slide 39, em que olhando o evento de envio de m1 no tempo $C(6)$ e o evento de envio de m2 no tempo $C(20)$ não é suficiente para concluir que o primeiro evento aconteceu antes do segundo, já que seus relógios não necessariamente estão sincronizados e ambos não trocaram mensagem até então. Para solucionar esse problema foi proposto o uso dos relógios vetoriais.

- 5) Permission based (Fragilidades): O coordenador se torna um SPOF, aumentando o risco de indisponibilidade do sistema. Em termos de performance, em sistemas de grande porte o coordenador centralizado se torna um bottleneck. Permission based (pontos positivos): A simplicidade de implementação; menor número de mensagens trocadas.

Ricart & Agrawala (Fragilidades): Esse algoritmo é distribuído e envolve a troca de mensagem entre os vários processos, portanto, existem um número N de possíveis pontos de falha, aumentando bastante o risco de indisponibilidade do sistema em grandes grupos de processos. Em termos de performance, em sistemas de grande porte a troca de mensagem entre todos os nós implica em aumento do tráfego de rede; é mais apropriado para pequenos grupos de processos. Ricart & Agrawala (ponto positivo): Não depende de um único coordenador.

Token-ring (Fragilidades): A perda do token se torna um problema, então é preciso um mecanismo para detecção de token perdido, tal como timeout, contudo susceptível a falsos-positivo. Token-ring (ponto positivo): Garantia e determinismo na obtenção do recurso compartilhado.

Fully decentralized (Fragilidades): Se uma quantidade f de coordenadores reinicializam, haverá uma violação na votação quando um número $m = >N/2$ coordenadores concordam em liberar o recurso solicitado. Token-ring (ponto positivo): Não é preciso se comunicar com todos os coordenadores para obter aprovação na liberação do recurso, reduzindo uso da rede e tempo para acesso ao recurso.

- 6) Dado que o Algoritmo de Bully realiza múltiplos rounds de eleição até encontrar o vencedor e o algoritmo de Ring apenas um, o segundo é mais eficiente do que o primeiro em termos de troca de mensagens. Para exemplo, considerar os slides 65 e 66 para Bully e 68 para Ring.