Sistemas Distribuidos

14/12/16

* Processos executam em máquinas diferentes.
* Processos devem se organizar
* Coordenação
  + Algorítmos (estratégias) centralizadas
  + Algorítmos distribuídos
* Centralizados
  + Processo coordenador com uma única função
  + Muito simples
  + Só o coordenador sabe quem pode executar
  + Se o coordenador não funciona nada é feito
  + Ponto de falha severa
* Distribuídos
  + Não existe coordenador
  + Reporta-se a todos os processos com o mesmo interesse
  + O desempenho é ruim, pois é muito burocrático.
  + Se um processo falha o sistema pode continuar funcionando.
  + Processo de decisão é do processo.
  + Grupo é montado com processos de mesmo interesse.
    - Aberto – Fácil entrada
    - Fechado – Tem 1 coordenador
    - Confiável - Todas as mensagens são entregues
    - Não confiável
  + Funciona bem: grupo fechado e confiável.
  + Como saber quem pediu primeiro? Sincronização!
  + Desempenho centralizado é bem melhor que o distribuido.

09/01/17

* Relógio físico
  + Marca o tempo real
  + Relógio físico não é rápido suficiente para certas ações.
* Relógio lógico
  + Sequenciamento de ações
  + Ordenação de eventos
  + Contador de eventos

11/01/2017

* Sincronização externa
  + As máquinas se sincronizam com uma única fonte externa
  + Satélites geram pulsos UTC
  + Tolerancia de atraso depende de cada aplicação.
* Sincronização interna
  + Não depende de fonte externa
  + Todos os relógios devem marcar a mesam hora.
* Como fazer sincronização externa
  + Quanto mais preciso mais caro.
  + Antena para ler pulsos UTC.
  + Algoritmo de Cristian – Servidor de tempo
  + NTP – Com extratos e acesso pela internet.
* Como fazer sincronização interna
  + Acerta na mão
  + Algoritmo de Berkley, daemon de tempo colhe os tempos das maquinas do grupo e manda elas ajustarem de acordo com a média.

16/01/17

* Relógio lógico
  + Contador
  + Relógios coordena
  + Processos devem ser distinguiveis.
  + A acontece antes de B
  + Relógio lógico conta eventos a cada evento inrementa-se o relógio.
  + Na mesma máquina todos os processos usam o mesmo relógio lógico.
  + O que acontece em máquinas diferentes?
  + Algoritmo de Lamport para sincronização de relógios lógicos
    - Antes da ação incrementa o relógio
    - Ao receber compara o número do relógio atual com o recebido, pega o maior e acrescenta-se 1.
  + Algoritmo só garante sincronização na ação.
  + Relógio vetorial (todos os relógios do sistema).

18/01/17

* Problemas resolviveis com a coordenação
  + Exclusão mútua
    - É fácil com memória comum.
    - Na computação distribuida não existe
  + Exclusão mútua distribuida
    - Coordenador
      * Garante exclusivadade de um recurso.
      * Coordenador deve estar numa máquina bem disponível.
      * Coordenador é de cada recurso.
      * 3 mensagens: Request, Release e OK
      * Grupo deve ser confiável.
      * Apresenta um ponto de falha
    - Distribuir
      * Não tem coordenador
      * Pergunta todos os processos se estão usando o recurso.
      * Se 2 pedirem o mesmo recurso, ver por relógio lógico quem pediu primeiro.
      * Detentores de recursos sabem quem vai pgar o recurso após eles através de uma fila.
      * Burocrático e lento

23/01/17

* Deadlock
  + Espera circular por recursos.
  + Em abminentes distribuidos é mais difícil a detecção
  + Ao bloquear, o que aquentece?
  + Prevenção
    - Acontece quando o processo pede algo externo
    - Deadlock de recuros e deadlock de comunicação (espera por informações)
    - Considera o canal de comunicação como um recursos e tudo vira deadlock de recurso.
    - Nunca ter ciclo em comunicações e recursos.
    - Modelagem formal do sistema = modelo matemático com transformações transistivas.
    - Comportamento observável (ESP)
  + Tratamento
    - Deixa acontecer e trata depois
    - Usa coordenação para tratar
    - Tratamento centralizado
      * Coordenador é quem avalia
      * Mantém um grafo global
      * Perinta ao coordndor se haverá deadlock.
      * Falso deadlock = coordenador ver, mas grupo não
        + Grupo confiável
        + Relógio lógico
      * Mata processo e libera a fila.
    - Distribuída
      * Cada processo tenta resolver
      * Processos detectam o deadlock.
      * Cada processo que bloqueia lança mensagem no meio, se mensagem volta é pq teve deadlock
      * Quem acha o deadlock se mata

25/01/17

* Eleição
  + Processo mais próximo podem receber identificadores mais altos
  + Máquinas que ficam mais tempo no ar ganham identificadores mais altos.
  + Segundo processo com melhor id ganha posição de coordenador quando o primeiro cair.
  + Algoritmo Bully
    - Eleição entre quem tem a mesma capacidade
    - Eleição é iniciada quando o coordenador é solicidado e não responde.
    - Quem prercebe que o coordenador caiu é quem incia a eleição.
    - Manda mensagem para processos com maior id
    - Recebe OK
    - Quem recebe menagem de elição a propaga
    - Se passar tempo e o processo não receber os Oks necessários é pq ele é o cara com maior identificador logo ele será o coordenador.
    - Avisa a todos que ele é o coordenador.
    - No retorno do ex coordenador el retoma o poder.
    - Usado quando for importante ter prioridade no coordenador.
  + Ring
    - Cara que percebe que o coordenador caiu inicia a eleição.
    - Manda p o processo seguinte na ordem de identificadores fazendo uma lista
    - Passou por todos, verifica quem é o coordenado na lista e publica para todos quem é o coordenador.
    - Se o cara de maior id voltar ele não tira o controle.
    - Multiplas listas podem surgir e o ultimo número da lisa define se ela será dropada ou não.