Sistemas Distribuídos
Sincronização

Disciplina: Sistemas Distribuídos
Prof.: Edmar Roberto Santana de Rezende

Faculdade de Engenharia de Computação
Centro de Ciências Exatas, Ambientais e de Tecnologias
Pontifícia Universidade Católica de Campinas

Sincronização

Clocks Físicos

- Algoritmo de Lamport:
  - provê uma forma de ordenação de eventos não ambígua
  - → os instantes de tempo associados a cada evento não são necessariamente próximos dos tempo em que os eventos realmente aconteceram
- Em alguns sistemas:
  - o conhecimento do instante de tempo real é muito importante

    Ex: Sistemas de tempo real
  - → para tais sistemas são necessários clocks físicos externos
  - é desejável a existência de mais de um clock externo
  - → eficiência e redundância
- Problemas:
  - como sincronizá-los com o clock de tempo real?
  - como sincronizar esses clocks entre si?

Sincronização

Medida do tempo

- Até a invenção dos relógios mecânicos no século XVII:
  - o tempo era medido com o auxílio dos astros
  - a cada dia:
    - o Sol nascia no horizonte a leste
    - alcançava uma altura máxima no céu
    - se punha no oeste
  - **trânsito do Sol:** evento do Sol alcançando seu ponto de maior altura
  - **dia solar:** intervalo entre dois trânsitos consecutivos do Sol
  - **segundo solar:** 1/86.400 avos do dia solar (24h x 3.600s)

- Por volta de 1940:
  - estabelecido que o período de rotação da Terra não é constante
  - desaceleração gradativa em função das marés e do atrito da atmosfera
  - há 300 milhões de anos o ano tinha cerca de 400 dias
    → o tamanho do ano não parece ter mudado, os dias ficaram mais longos

Sincronização

Medida do tempo

- Além dessas variações de muito longo prazo:
  - ocorrem também pequenas variações no comprimento do dia
  - → devido provavelmente à turbulência no núcleo central do globo terrestre
- Tais revelações levaram os astrônomos a calcular o comprimento de um dia:
  - medindo a duração de um grande número de dias e
  - **segundo solar médio:** resultado da divisão da duração média do dia por 86.400
- Em 1948:
  - invenção do relógio atômico
  - contagem das transições do átomo de césio 133
  - → tornou-se possível:
    - medir o tempo com um grau de precisão muito maior
    - de maneira independente das condições do globo terrestre e da atmosfera que o cerca

Sincronização

Medida do tempo

- Relógio atômico:
  - **segundo:** tempo necessário para que o átomo de césio 133 faça extamente **9.192.631.770** transições
  - → a escolha deste valor é devido à necessidade de torná-lo igual ao segundo solar médio no ano em que o segundo atômico foi introduzido

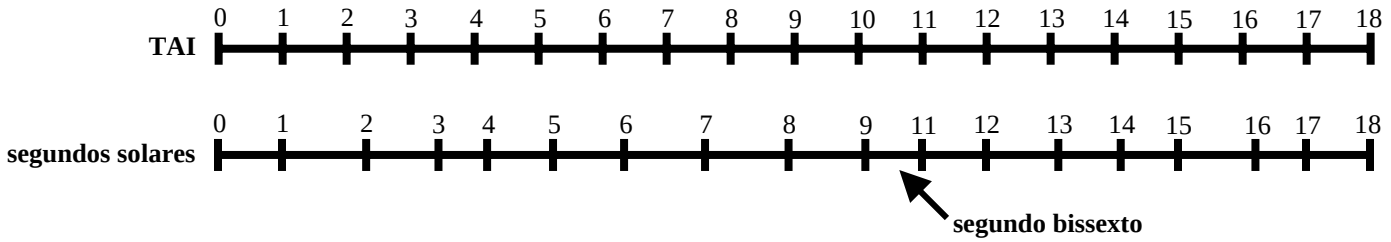
- Atualmente:
  - existem cerca de 50 laboratórios no mundo inteiro com relógios de césio 133
  - periodicamente, cada laboratório informa ao BIH (*Bureau International de l'Heure*), em Paris:
    - → quantas vezes seu relógio oscilou desde 1 de janeiro de 1958
  - **TAI (Tempo Atômico Internacional)**
    - → média do número de oscilações de um conjunto de relógios de césio 133 dividida por 9.192.631.770

Sincronização

Medida do tempo

- TAI (Tempo Atômico Internacional):
  - apesar do TAI ser extremamente preciso existe um problema:
  - → 86.400 TAI segundos é cerca de 3 ms menor que um dia solar atual
    (o dia solar médio está durando mais com o passar do tempo)
  - → com o passar dos anos o meio dia vai acontecer cada vez mais cedo
    (Em 1582 o Papa Gregório XIII suprimiu por decreto 10 dias do calendário)

- Solução:
  - **segundos bissextos:**
    - sempre que a diferença entre o tempo solar e o TAI ultrapassar a casa dos 800 ms



Sincronização

Medida do tempo

- Tempo Universal Coordenado (UTC):
  - esta correção deu origem a um sistema de tempo baseado em segundos TAI constantes
  - → mas que permanecem em fase com o movimento aparente do sol
  - é a base de todos os sistemas de medida de tempo civil adotados atualmente
  - → aos poucos ele está substituíndo os padrões antigos como o **GMT** (Tempo Médio de Greenwich) que é o tempo astronômico
  - Para tornar o UTC acessível:
    - NIST (*National Institute of Standard Time*):
      - → opera estação de rádio em ondas curtas com o prefixo WWV
      - → a WWV envia em broadcast um pulso no início de cada segundo UTC
      - → precisão de 1ms (podendo chegar à 10ms)
    - GEOS (*Geostationary Environment Operational Satellite*):
      - → vários satélites também oferecem o serviço UTC
      - → precisão de 0,5 ms (ou até mais precisos)

Sincronização

Algoritmos para sincronização de clock

- Se uma das máquinas em um sistema distribuído possuir um receptor WWV:
  - o problema se resume a manter todas as demais máquinas do sistema em sincronismo com ela
- Se nenhuma máquina do sistema possuir um receptor WWV:
  - então cada uma delas deve cuidar de seu próprio tempo
  - o problema passa a ser o de manter tais máquinas juntas
- Propostos diversos algoritmos para realizar essa sincronização:
  - Todos baseados no mesmo modelo de sistema:
    - todas as máquinas têm um temporizador que gera uma interrupção H vezes por segundo
    - quando esse temporizador expira, um software adiciona 1 unidade ao clock **C** conhecido por todas as máquinas do sistema
    - quando o tempo UTC for ***t*** o valor do clock em uma máquina ***p*** é $\mathbf{C_p(t)}$

Sincronização

Algoritmos para sincronização de clock

- Em condições ideais:
  - $C_p(t) = t$, para qualquer valor de p e t
  - → a primeira derivada de **C** em relação ao tempo deve ser sempre 1
  - → condições impossíveis de serem obtidas na prática
    - temporizadores com erro relativo de ~10^{-5}

- Soluções:
  - Algoritmo de Cristian
  - Algoritmo de Berkeley
  - Algoritmos baseados na média
  - Sistemas com várias fontes externas

Sincronização

Algoritmo de Cristian

- Baseado no trabalho de Cristian (1989)
- Servidor de tempo:
  - máquina com receptor WWV
- Periodicamente:
  1. cada máquina envia uma mensagem para o servidor de tempo perguntando pelo tempo corrente
  2. o servidor de tempo responde o mais rápido possível, com uma mensagem contendo o tempo corrente C_{UTC}
  - quando o transmissor obtém uma resposta pode simplesmente ajustar seu clock
    - → Problemas:
      - o tempo nunca pode andar para trás
      - a consulta ao servidor de tempo gasta um tempo não-nulo (o retardo pode vir a ser grande)

Sincronização

Algoritmo de Cristian

- Soluções:
  - mudança no clock deve ser feita gradativamente

    Ex: se deve adicionar 10 ms, basta adicionar 9 ms (para atrasar) ou 11 ms (para adiantar) o clock
  - tentar medir o tempo de transmissão:

    T_0 → envio da requisição

    T_1 → recebimento da resposta

    $(T_1 - T_0)/2$ → tempo aproximado de propagação da mensagem
  - se soubermos:

    I → tempo de tratamento da interrupção

    $(T_1 - T_0 - I)/2$ → melhor estimativa do tempo de propagação da mensagem
  - melhor estimativa ainda:
    - realizar uma série de medidas, em vez de uma única
    - calcular a média de tais medidas

      (possivelmente descartando valores fora de um limite)

Sincronização

Algoritmo de Berkeley

- Algoritmo de Cristian:
  - o servidor de tempo é completamente passivo
  - → no Unix de Berkeley foi adotada uma estratégia oposta
- Algoritmo de Berkeley:
  - servidor de tempo é uma entidade ativa:
    - consulta periodicamente cada uma das máquinas do sistema para saber o tempo corrente em cada uma delas
    - baseado nas respostas:
      1. calcula o tempo médio
      2. informa a todas as outras máquinas para adiantar ou atrasar seus clocks, tornando-se iguais ao tempo médio calculado
  - método adequado quando nenhuma das máquinas tem um receptor WWV
  - o ajuste do servidor de tempo deve ser feito periodicamente por um operador (de forma manual)

Sincronização

Algoritmos Baseados na Média

- Algoritmos de Cristian e de Berkeley:
  - são altamente centralizados
- Algoritmos Baseados na Média:
  - no iníco de intervalos fixos:
    - cada máquina envia em broadcast seu tempo corrente
      (as mensagens não serão enviadas todas ao mesmo tempo)
    - em seguida, inicia um temporizador local para coletar toda as outras mensagens que chegarem em um tempo ***S***
    - quando todas as mensagens forem recebidas:
      - basta executar um algoritmo para calcular novo tempo corrente
        (pode ser a média em um caso simples ou algo mais sofisticado)

Sincronização

Sistemas com Várias Fontes Externas de Tempo

- Problemas:
  - valores diferentes serão produzidos
- Solução:
  - sincronizar todas as fontes de alguma forma:
    - algoritmos vistos anteriormente
    - organizar a rede em anel:
      - cada máquina troca informações de tempo com seus vizinhos