Introdução

Relevante pois em diversas aplicações o tempo é extremamente crítico por conta da sincronia.

- A necessidade de sincronização em algoritmos distribuídos

  Quando uma transação ocorre, tem ações no computador do cliente e do banco, a informação sobre tempo da transação é fundamental, bem como a sincronia entre as máquinas.
  - Ex: Transações de comércio eletrônico
- Dificuldade de registrar o tempo em diferentes nós

  A garantia que o tempo seja exatamente o mesmo.
- Não existe um tempo global absoluto

  Não existe um mesmo tempo para todos os computadores para um SD, busca-se apenas o máximo de sincronia/precisão, uma diminuição de defasagem.

Relógios, eventos e estados de processo

- Relógio: Utiliza cristal de quartzo

  O cristal oscila em frequência constante quando submetido a uma corrente elétrica, a oscilação é usada para contabilizar o tempo.

- A distorção entre relógios

  Contudo há variações nessas frequências de cada computador, assim têm-se distorções.
  - **Taxa de desvio:** distância instantânea entre as leituras de dois relógios.

    O quão defasado o tempo de um relógio está em relação a outro.
  - **Taxa de Derivação:** Diferença na velocidade de contagem do tempo em função de diferenças físicas nos osciladores de cristal.

    Relacionada a velocidade com a

qual um computador vai contar o tempo baseando-se no cristal.

- Taxa de derivação:

  cerca de 10^{-6}s

  Dois relógios tem uma diferença de 1s entre eles a cada 11,6 dias.

- Tempo universal coordenado
  - Padrão internacional para contagem do tempo.
  - Obtido a partir de relógios atômicos com inclusão do ano bissexto.
    Mantido por um conjunto de equipamentos espalhados pelo mundo inteiro, baseados na oscilação do átomo.
  - Taxa de derivação de relógios atômicos: 10-13s.

Sincronização de relógios físicos

- **Sincronização externa:** sincronização com uma fonte externa.
  Por exemplo, quando usa-se a fonte de tempo UTC.
- **Sincronização interna:** sincronização entre máquinas dos seus relógios locais.
  Não necessariamente necessita de uma fonte externa.
- Um relógio H é correto se a taxa de derivação está dentro de um limite.
  Se a taxa de derivação do relógio é maior que a taxa de derivação limiar, ele não está correto.
- Monotonicidade de um relógio de software C
  Um relógio deve ter a monotonicidade garantida quando ele está sempre avançando no seu tempo.
  - $t>t$ implica que $C(t')>C(t)$
- Exemplo: Comando make

Mecanismo usado para otimizar o processo de compilação de código. Por exemplo: verifica se o código foi compilado e se sim e a hora do arquivo fonte for anterior à hora do código objeto, ele não irá compilar aquele arquivo.
Ou seja, só serão recompilados os códigos fontes cuja alteração foi realizada após a última compilação.

- Compara arquivo fonte com seu arquivo objeto.
- Compila apenas os modificados
- **Possível erro:** relógio adiantado durante uma compilação, depois atrasado para corrigir. Se o arquivo fosse modificado, o make não compilaria.

Porque o executável seria mais novo que o código fonte. Percebe-se a importância da monotonicidade.

Possíveis falhas

- A necessidade de correção em software do tempo fornecido pelo hardware.
  A hora que o hardware fornece nem sempre é a correta, assim às vezes são necessários cálculos para se garantir a hora correta.
- Um relógio sem condições de correção é definido como falho.
  Um relógio que não obedece a monotonicidade e sua taxa de derivação é superior ao limite.
- **Falha de colapso:** Para completamente.
  Falta de energia ou pane.
- **Falha arbitrária:** Bug do ano 2000, bateria fraca.
  Exemplo: falha na projeção dos relógios.
- Relógios não precisam ser precisos para serem corretos.
  Grau de precisão nem sempre é importante, como na sincronização interna.
- Na sincronização interna, a preocupação é apenas com o funcionamento

correto do relógio, não com sua configuração absoluta.
A ideia é simplesmente garantir a sincronia interna, então a hora externa é irrelevante.

Sincronização em sistema síncrono

Processos trabalhando sincronizados em relação ao tempo.

Exemplo:

- Dois processos em um sistema síncrono distribuído
  Um processo envia seu tempo "t" atráves de uma mensagem "m", o segundo processo obtém a hora e atualiza seu relógio.
- Dados conhecidos
  - Taxa de derivação dos relógios
  - Atraso máximo de transmissão de mensagens
    Tempo de envio de um processo para outro.
  - Tempo de execução de cada etapa de um processo
    Tempo que cada processo leva para enviar seus dados.
- Considere
  - t - tempo atual do relógio enviado numa mensagem m entre duas máquinas.
  - T_{trans} – tempo gasto na transmissão da mensagem
  - Tempo do receptor da mensagem = $t + T_{trans}$
    T_{trans} no valor mínimo está no melhor caso, sem obstáculos, e o máximo com problemas. Deve-se calcular a incerteza/desvio máximo do relógio, subtraindo o máximo do mínimo.
- Ttrans pode variar entre um mínimo e um máximo.
- No entanto, a maioria dos sistemas é assíncrono: Internet.
  A internet por natureza é assíncorna.

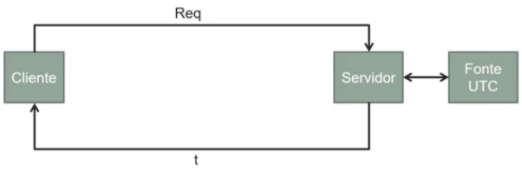
Método de Cristian

Método usado para sistemas assíncronos.

- Sincronização externa em sistemas assíncronos
  Usa-se uma fonte externa de tempo para sincronizar o tempo entre duas ou mais máquinas em sistemas assíncronos.
- Um servidor de tempo para sincronização externa.
  Usa-se um servidor de tempo para realizar a sintonia.

  Um servidor sincronizado com uma fonte externa manda sua hora para um cliente assim que requisitado.

  É feito o cálculo de tempo no Cliente (hora que recebeu + tempo de viagem dividido por dois (pois ida e volta))



RTT= round trip time(requisição+resposta)

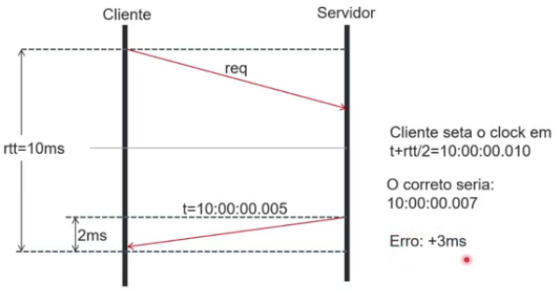
Clock do cliente = t+RTT/2

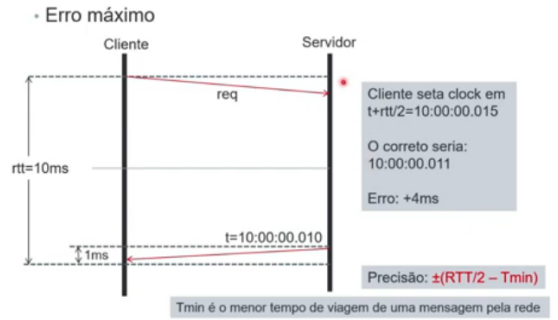
Método de Cristian

- Adequado para sistemas onde os RTTS são menores que a precisão desejada

- Existe um fator de erro na precisão do tempo calculado.
  Contudo, o uso de RTT/2 possui falhas pois a ida pode ser mais rápida que a volta ou vice-versa.

Método de Cristian: Precisão





Conclusões

- Quanto mais o RTT se aproxima de Tmin, maior será a precisão.

  Tmin: menor tempo necessário para que uma mensagem viaje pela rede.

  A precisão se aproxima de 0.

Problemas

- Uso de um servidor central para sincronizar.

  Existe um único ponto de falha, se ele falhar traz problemas para um conjunto de computadores.
- Existência de servidores intrusos difundindo hora falsa.

Algoritmo de Berkeley

Os nós são divididos entre mestres e escravos.

- A cada "t" unidades de tempo, o mestre solicita aos escravos o valor de seus clocks.
- Mestre estima o valor dos clocks de cada cliente pelo RTT

  Garantem a sincronização dos relógios dos escravos. De tempos em tempos o mestre requer o horário de seu escravo, estima a hora correta de cada relógio dos escravos, calcula a média de todos os relógios e elimina as horas acima de determinado limiar.
- Calcula a média dos valores recebidos, incluindo seu próprio relógio e eliminando todas as leituras maiores que um limiar.
- O mestre envia o valor ajustado aos escravos.
- Se o mestre falhar, outra máquina é eleita para substituí-la.

  Através de um algoritmo de eleição.

Algoritmo de Berkeley - Exemplo

LIMIAR DO RTT = 10ms

Maquina	Leitura	RTT	ESTIMADO	MÉDIA	AJUSTES
MESTRE	4:00:08.020	0	4:00:08.020		+0.003
P1	.043	8	.047	4:00:08.023	-0.024
P2	.000	4	.002		+0.021
P3	.085	16	.093	Desconsidera	-0.070

NTP: Network Time Protocol

Estratégia específica para internet.

- Christian e Berkeley são utilizados em intranets.
- NTP é um serviço para a Internet.
- Clientes sincronizam um com uma fonte UTC

  Os relógios atômicos continuam sendo a base da sincronização.
- Adequado para sistemas síncronos e para sincronização externa

- Característica
  - Confiável

    Mesmo havendo perda de conectividade, deve continuar em pleno funcionamento (tolerante a falhas). Usa-se uma cadeia de servidores e caminhos redundantes.
  - Escalável

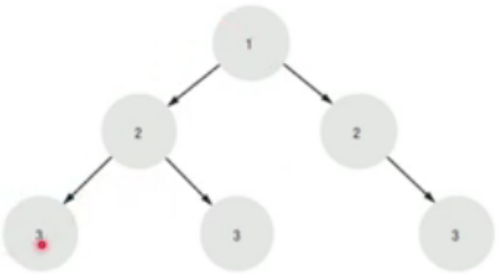
    Suporta um grande número de clientes até mesmo pela forma de sua hierarquia.
  - Seguro

    Proteção contra interferência no serviço de tempo (mal intencionada ou acidental). Há uma verificação de autenticidade a cada alteração de hora.

NTP: Strata

- Cada Strata ou camada possui vários servidores.
- Os servidores são conectados a uma sub-rede de sincronização.
- Os servidores do stratum 1 são conectados a uma fonte UTC

  Os servidores primários sincronizam com o utc e seus filhos com seus pais e assim respectivamente.



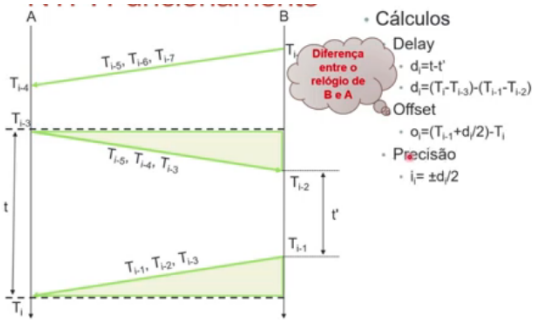
NTP: sincronização entre servidores

- Multicast
  - Usado em LANS de alta velocidade.
  - Um ou mais servidores enviam periodicamente o valor do clock para os outros servidores.
  - Precisão relativamente baixa
    Por usar multicast.
- Por solicitação
  - Quando uma precisão maior que o multicast é necessária.
  - Um servidor responde aos pedidos de timestamp (método Christian)
- Simétrico
  - Utilizado entre servidores nas camadas mais próximas da fonte UTC da subrede de comunicação.

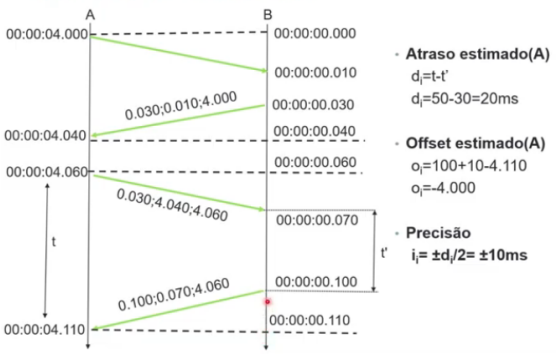
NTP: Funcionamento

- Servidores trocam pares de mensagens.
- Cada mensagem carrega os timestamps das duas últimas mensagens e da mensagem atual.
  Na mensagem de cada nó é a hora atual do relógio e a hora das duas últimas mensagens recebidas.
- Estima a diferença entre os dois relógios e a precisão da estimativa.
- Mensagens são trocadas com vários servidores.
- O servidor seleciona os valores mais confiáveis.
- Precisão: ±1ms em LANS

NTF: Funcionamento



Aula 16 - Tempos globais pt.1



- O algoritmo guarda os oito últimos pares <o_i,d_i>
  Oito servidores fazem o cálculo de offset e atraso, verifica-se dentro do conjunto de oito servidores o que possui menor atraso.

- Para setar o clock, NTP utiliza o valor do offset que apresenta o menor d_i (ou seja, maior precisão);
- Outros exemplos: https://ntp.br/ntp.php#O_Funcionamento do NTP

Sincronização de relógios físicos