Sistemas Distribuídos

Resumo

Rafael Rodrigues

LEIC
Instituto Superior Técnico
2022/2023

Contents

1 Introdução

Sistema Distribuído - Sistema de componentes software ou hardware localizados em computadores ligados em rede que comunicam e coordenam as suas ações através de troca de mensagens.

2 Remote Procedure Call (RPC)

2.1 Arquitetura cliente-servidor

- Servidores mantêm recursos e server pedidos de operações sobre esses recursos.
- Servidores podem ser clientes de outros servidores.
- Simples e permite distribuir sistemas centralizados muito diretamente.
- Limitado pela capacidade do servidor e pela rede que o liga aos clientes.

3 Fundamentos: Tempo

3.1 Sincronização de Relógios

- **Externa** - relógios dos processos são sincronizados através de uma **referência externa**
- **Interna** - relógios dos processos de um sistema **sincronizam-se entre si**

3.1.1 Algoritmo de Cristian

Os relógios dos clientes são sincronizados pelo relógio de um **servidor de tempo** (sincronização externa).

1. Servidor S lê o valor dos outros relógios.
   $T_{S_i} = T_{env_i} + T_{rec_i}/2$
   $delta_i = T_S - T_i$
   $erro_i = \pm\ RTT_i/2$
2. Indica a todos os participantes para ajustarem o seu relógio (incluindo o seu).
   $ajuste_i = \overline{T} + delta_i$

Diferença máxima = Soma dos dois maiores valores de erro

3.1.2 Algoritmo de Berkeley

1. É escolhido um líder através de um processo de eleição.
2. O líder pergunta os tempos aos seus servidores.
3. O líder calcula o tempo de cada máquina tendo em atenção o RTT.
4. O líder calcula a média dos tempos, ignorando os outliers.
5. Envia o valor (positivo ou negativo) que cada máquina deve ajustar.

3.1.3 Network Time Protocol (NTP)

3.2 Relógios Lógicos

Relacão happens-before/aconteceu-antes ($\rightarrow$)

1. se a e b são eventos do mesmo processo, se a ocorre antes de b, então $a \rightarrow b$
2. se a indica um evento envio de mensagem, e b o evento da receção dessa mensagem, então $a \rightarrow b$

Transitividade: se $a \rightarrow b$ e $b \rightarrow c$, então $a \rightarrow c$
Eventos concorrentes: se nem $a \rightarrow b$, nem $b \rightarrow a$, então $a \parallel b$

3.2.1 Lamport

1. se $a \rightarrow b$, então $C(a) < C(b)$
   - se os eventos ocorrerem no mesmo processo, e a ocorre *antes* de b, então $C(a) < C(b)$
   - se a for o evento envio de mensagem e b a sua receção, então $C(a) < C(b)$
2. o valor de $C(e)$ *nunca decresce*
   - As correções ao relógio devem ser feitas sempre por incrementos

3.2.2 Vector clocks

1. se $C(a) < C(b)$, então $a \rightarrow b$
   - $V_a < V_b$ se pelo menos um elemento de V_a for menor e nenhum for maior que V_b

4 Fundamentos: Coordenação

4.1 Exclusão Mútua

4.1.1 Algoritmo de Maekawa

4.2 Eleição de líder

4.2.1 Algoritmo baseado em anel

4.2.2 Algoritmo "bully"

5 Replicação

5.1 Coerência fraca

5.1.1 Gossip

5.1.2 Bayou

5.2 Coerência forte

5.2.1 Primary-backup

5.2.2 Replicação de máquina de estados

5.2.3 Registo distribuído coerente