

# **Sistemas Distribuídos**

Resumo

**Rafael Rodrigues**

LEIC

Instituto Superior Técnico

2022/2023

# Contents

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Remote Procedure Call (RPC)</b>	<b>3</b>
2.1	Arquitetura cliente-servidor . . . . .	3
<b>3</b>	<b>Sincronização de Relógios</b>	<b>4</b>
3.1	Algoritmo de Cristian . . . . .	4
3.2	Algoritmo de Berkeley . . . . .	4
3.3	Network Time Protocol (NTP) . . . . .	4
<b>4</b>	<b>Relógios Lógicos</b>	<b>5</b>
4.1	Lamport . . . . .	5
4.2	Vector clocks . . . . .	5
<b>5</b>	<b>Registos</b>	<b>5</b>
<b>6</b>	<b>Espaços de tuplos</b>	<b>5</b>
<b>7</b>	<b>Estados Globais</b>	<b>6</b>
<b>8</b>	<b>Eleição de líder</b>	<b>7</b>
8.1	Algoritmo baseado em anel . . . . .	7
8.2	Algoritmo "bully" . . . . .	7
<b>9</b>	<b>Exclusão Mútua</b>	<b>7</b>
9.1	Algoritmo de Ricart-Agrawala . . . . .	7
9.2	Algoritmo de Maekawa . . . . .	7
<b>10</b>	<b>Difusão em ordem total</b>	<b>8</b>
<b>11</b>	<b>Replicação</b>	<b>9</b>
11.1	Coerência forte . . . . .	9
11.1.1	Primary-backup . . . . .	9
11.1.2	Replicação de máquina de estados . . . . .	9
11.1.3	Registo distribuído coerente . . . . .	9
11.2	Coerência fraca . . . . .	9
11.2.1	Gossip . . . . .	9
11.2.2	Bayou . . . . .	9
<b>12</b>	<b>Consenso</b>	<b>10</b>
12.1	Floodset consensus . . . . .	10
<b>13</b>	<b>Transações distribuídas</b>	<b>11</b>
13.1	2-phase commit (2PC) . . . . .	11
<b>14</b>	<b>Segurança</b>	<b>12</b>

# 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)

Estrutura a programação distribuída com base na chamada (pelo clientes) de procedimentos que se executam remotamente (no servidor).

O cliente envia um pedido e o servidor envia uma resposta.

### 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 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 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).

$$ajuste_i = \bar{T} + delta_i$$

Diferença máxima = Soma dos dois maiores valores de erro

Precisão =  $\pm (RTT/2 - min)$

#### 3.2 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.

#### 3.3 Network Time Protocol (NTP)

## 4 Relógios Lógicos

Relação 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 recepção 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  $a \parallel b$

### 4.1 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 recepção, então  $C(a) < C(b)$
2. o valor de  $C(e)$  *nunca decresce*
  - As correções ao relógio devem ser feitas sempre por incrementos

### 4.2 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  $V_b$

## 5 Registos

## 6 Espaços de tuplos

## 7 Estados Globais

## 8 Eleição de líder

### 8.1 Algoritmo baseado em anel

### 8.2 Algoritmo "bully"

## 9 Exclusão Mútua

### 9.1 Algoritmo de Ricart-Agrawala

### 9.2 Algoritmo de Maekawa



## 10 Difusão em ordem total

# 11 Replicação

## 11.1 Coerência forte

### 11.1.1 Primary-backup

### 11.1.2 Replicação de máquina de estados

### 11.1.3 Registo distribuído coerente

## 11.2 Coerência fraca

### 11.2.1 Gossip

### 11.2.2 Bayou

## 12 Consenso

### 12.1 Floodset consensus

## 13 Transações distribuídas

Uma transação distribuída implica executar múltiplas suboperações em nós diferentes.

- Caso tudo corra bem, todas as suboperações são confirmadas (**commit**).
- Caso algo corra mal, todas as suboperações são anuladas (**abort**).

Propriedades ACID:

- Atomic - Atomicidade
- Consistent - Coerência
- Isolated - Isolamento
- Durable - Durabilidade

### 13.1 2-phase commit (2PC)

1. O coordenador envia uma mensagem denominada "prepare" a todos os participantes.
- 2.

## 14 Segurança