

Paralelismo, Concorrência e Coordenação Distribuída



Arquitetura de Sistemas Distribuídos,
Paralelos e Concorrentes
Escola Politécnica – PUCPR
Luiz A. de P. Lima Jr. • luiz.lima@pucpr.br

1

Paralelismo, Concorrência e Coordenação Distribuída

1. Servidores e Concorrência
2. Processos, threads e controle de concorrência
3. Controle Distribuído de Concorrência

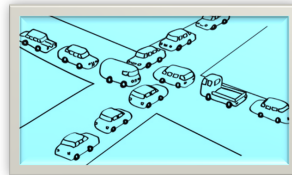


2

2



ASDPC



Concorrência em Sistemas Distribuídos

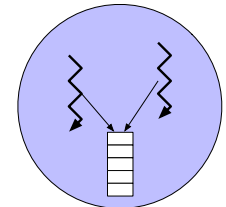
Paralelismo, Concorrência e Coordenação Distribuída

3



Concorrência em SDs

- Sistemas distribuídos são **inerentemente concorrentes**:
 - ✦ consequência da distribuição: vários usuários
 - ✦ a concorrência é necessária à **escalabilidade**
- A concorrência por **recursos compartilhados**:
 - ✦ é potencial fonte de **problemas...**
- **Necessidades**:
 - ✦ **Coordenação** do acesso aos recursos:
 - aspecto *recorrente* em *Sistemas Distribuídos*
 - ✦ **Exclusão Mútua** de Execução:
 - Monitores
 - Semáforos
 - ...
- Tratamento de deadlocks



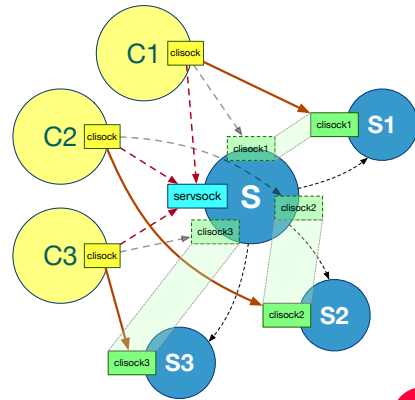
4

4



Servidores Concorrentes

```
int servsock = socket(...);
bind(servsock, addr); // addr = porta
while(true) {
    int clisock = accept(servsock, ...);
    pid_t pid = fork();
    if (pid == 0) { /* filho? */
        close(servsock);
        processa(clisock);
        close(clisock);
        exit(0);
    }
    close(clisock);
}
```

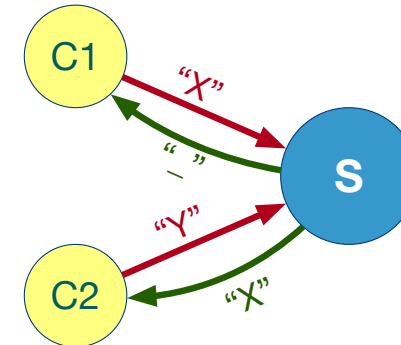


5

5

Servidores Concorrentes

- Como fazer com que o servidor responda sempre com a **última mensagem** enviada por algum cliente?

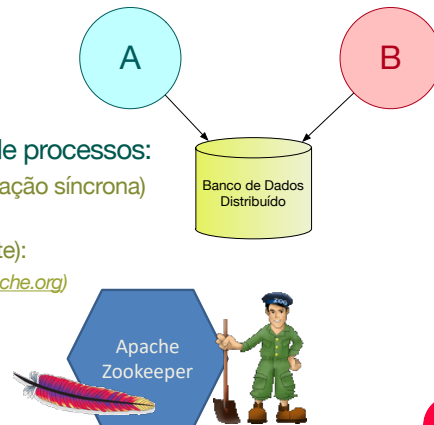


6

6

Concorrência em SDs

- Coordenação de processos:
 - Não apenas processos locais:
 - COORDENAÇÃO DISTRIBUÍDA
- Para coordenação distribuída de processos:
 - recebimento bloqueante (comunicação síncrona)
 - complexo...
 - utilização de plataforma (de suporte):
 - e.g., Apache ZooKeeper (zookeeper.apache.org)



7

7



ASDPC

ZooKeeper

Paralelismo, Concorrência e Coordenação Distribuída

8

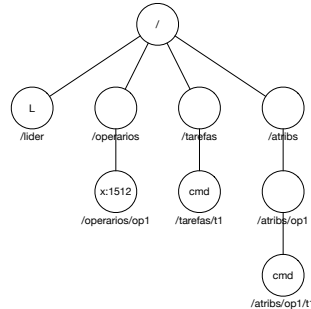
ZooKeeper: Visão Geral

● ZooKeeper (<https://zookeeper.apache.org>)

- semelhante a uma **API para sistema de arquivos**:
 - primitivas para criação, remoção, status, leitura (*get*), escrita (*set*), ...
- aplicações criam "**receitas**":
 - locks distribuídos, barreiras, eleição de líder, etc.*

● znodes:

- nós de **dados** (sequência de bytes) organizados em árvore



9

9

ZooKeeper: Visão Geral

● API básica:

- create** /path data
- delete** /path
- exists** /path
- setData** /path data
- getData** /path
- getChildren** /path

● ZooKeeper:

- garante de que não haverá leituras/escritas parciais de dados

10

10

ZooKeeper: Visão Geral

● znodes:

- persistentes**:
 - deletados apenas explicitamente (comando "*delete*")
- efêmeros**:
 - deletados quando cliente que os criou encerra sua sessão
 - (e.g., operário)
 - não podem ter filhos (pois são deletados "automaticamente")
- sequenciais**:
 - aos quais associa-se valores sequenciais crescentes
 - exemplo: `/tarefas/tarefa-0000000012`
 - para se obter nomes únicos
 - para saber a ordem de criação

11

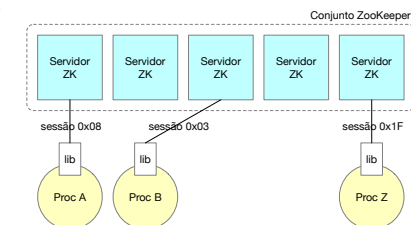
11

ZooKeeper: Visão Geral

● waches:

- para evitar "*polling*"
- watch**:
 - para uma única notificação
 - (se outras são necessárias, é necessário registrar **novo watch** a cada notificação)

● Arquitetura:



12

12



Programação Paralela X Programação Distribuída

13



Programação Paralela x Distribuída

	Programação Paralela	Programação Distribuída
Objetivos	desempenho (que um único processador não poderia atingir)	conveniência (disponibilidade, confiabilidade) + desempenho (geralmente, em 2º plano)
Interação	frequente, de alta granularidade, de baixo overhead e confiável	não tão frequente, grande volume de dados e não confiável

14

14



Programação Paralela x Distribuída

- A diferença em alguns aspectos se reduz a uma questão de:
 - ✦ grau apenas (e.g., frequência de interação); ou de
 - ✦ ênfase (desempenho x confiabilidade)
- PP e PD representam pontos distintos de um mesmo “espaço multidimensional”:
 - ✦ Aprender PP ajuda na PD
 - ✦ Aprender PD ajuda na PP

15

15

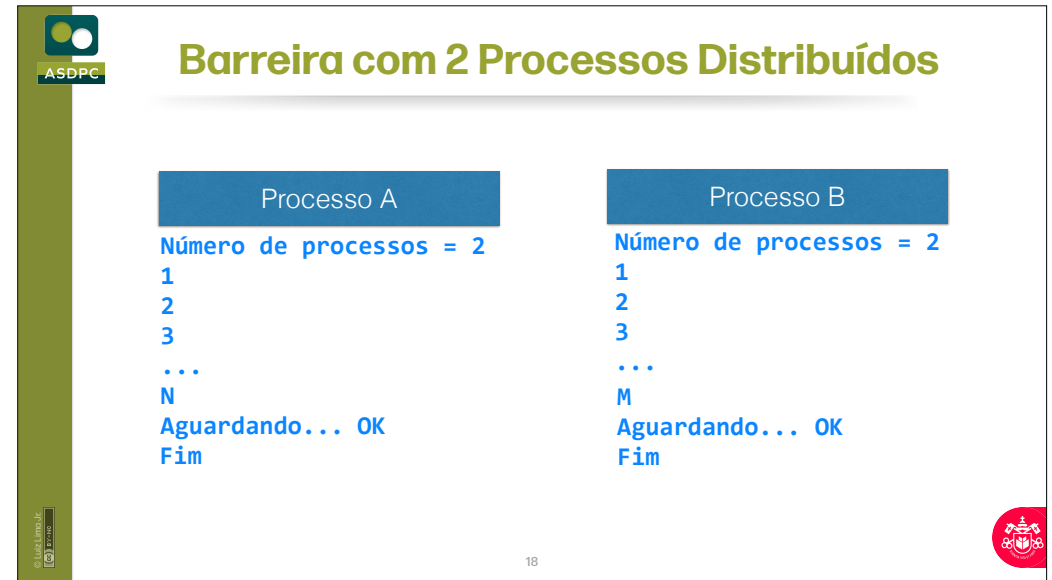


TDE 1 – Coordenação Distribuída de Processos com a Plataforma Zookeeper

16



17



18

Bibliografia

G. Coulouris et al. **"Distributed System: Concepts and Design"**, 5th edition, Addison Wesley, 2012.

Lin, C.; Snyder, L. **"Principles of Parallel Programming"**, Addison-Wesley, 2009.

F. Junqueira, F.; Benjamin, R. **Zookeeper**, O'Reilly 2014.

19

19