

# Aspectos de Projeto de Sistemas Distribuídos

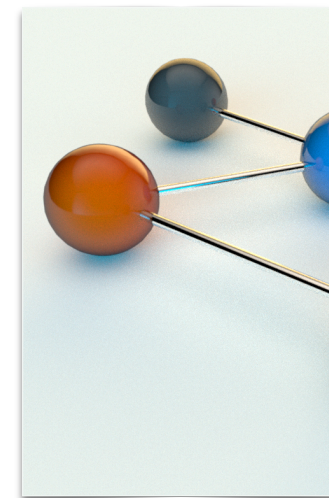


Arquitetura de Sistemas Distribuídos,  
Paralelos e Concorrentes  
Escola Politécnica – PUCPR  
Luiz A. de P. Lima Jr. • [luiz.lima@pucpr.br](mailto:luiz.lima@pucpr.br)

1

## Aspectos de Projeto de SD

Transparência  
Flexibilidade  
Confiabilidade  
Desempenho  
Escalabilidade



2

2



## Transparência

Aspectos de Projeto de SDs

3



## Transparência

### ● Objetivo

- ✦ fornecer aos usuários uma **imagem única e abstrata** do sistema computacional

### ● Níveis de transparência

- ✦ Nível de **usuário**: O usuário tem a impressão de estar usando um sistema centralizado.
- ✦ Nível de **programador**: O programador tem a ilusão de programar um sistema centralizado.
  - *Sintaxe e semântica das chamadas e mensagens remotas devem ser semelhantes.*

4

4

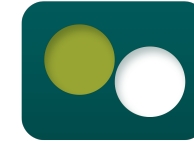


## Tipos de Transparências

- **Localização**
  - ✦ os usuários não precisam conhecer a localização dos recursos
- **Migração**
  - ✦ os recursos (dados, computação, processos) podem se mover no sistema sem alterar seus nomes (ou referências)
- **Replicação**
  - ✦ os usuários não sabem quantas cópias de um recurso existem
- **Concorrência**
  - ✦ múltiplos usuários podem compartilhar um recurso sem o perceber (e sem conflitos)
- **Paralelismo**
  - ✦ atividades podem ocorrer em paralelo sem que o usuário tenha de explicitá-las

5

5



ASDPC

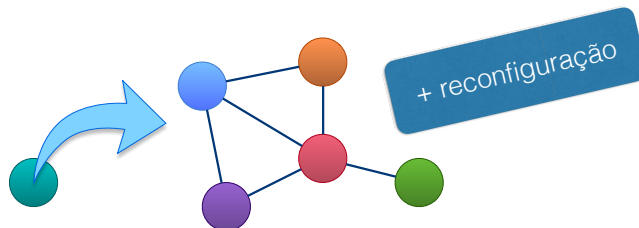
## Flexibilidade

Aspectos de Projeto de SDs

6

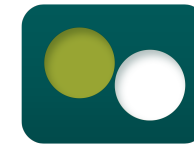
## Flexibilidade/Abertura (“Openness”)

- A **inserção** de novas entidades/serviços no sistema deve ser uma tarefa **simples**.



7

7



ASDPC

## Confiabilidade

Aspectos de Projeto de SDs

8

## Confiabilidade

- Em teoria:
  - ✦ Se uma máquina falhar, outra pode assumir suas tarefas.
    - = "confiabilidade"
- Na prática:
  - ✦ Alguns componentes ou serviços são mais **vitalis** para o sistema.
  - ✦ Caso parem, todo o sistema pode cair.
- Aspectos da confiabilidade
  - ✦ **Disponibilidade**
  - ✦ **Segurança**
  - ✦ **Tolerância a faltas**

9

9

## Confiabilidade: Disponibilidade

- **Fração de tempo** em que o sistema está **disponível** para uso.
- Alcançada através de:
  - ✦ **redundância** de componentes críticos:
    - se um componente falhar, pode ser substituído
- **Técnicas** geralmente utilizadas:
  - ✦ redundância de **hardware**
    - processadores, discos, memória, links de comunicação, ...
  - ✦ redundância de **software**
    - dois programas distintos efetuando a mesma função

10

10

## Confiabilidade: Segurança

- **Autenticidade**
  - ✦ Os usuários comprovam suas **identidades**
  - ✦ senhas, chaves, etc.
- **Autorização**
  - ✦ estabelecimento de **controles de acesso** aos recursos
  - ✦ **listas** de controle de acesso
- **Privacidade**
  - ✦ As informações somente podem ser lidas por quem tiver direito.
  - ✦ mecanismos de **criptografia**
- **Integridade**
  - ✦ os dados não podem ser destruídos ou corrompidos por terceiros
- **Não-repudição**
  - ✦ Todas as ações podem ser imputadas a seus **autores**.
  - ✦ mecanismos de **auditoria**

11

11

## Confiabilidade: Tolerância a Faltas

- O que fazer em caso de **falha** de um servidor?
- Sistemas distribuídos podem ser projetados para  **mascarar falhas**.
- Abordagens:
  - ✦ **replicação** de servidores
  - ✦ execução sem estado (**stateless execution**)

12

12

## Faltas, erros e falhas

### ● Faltas

- ✦ **Situações incorretas** no estado interno de um sistema
- ✦ Ex: um *bit* de memória inválido, um cabo de rede rompido

### ● Erro

- ✦ **Decorrencia da falta**
- ✦ Estado interno incorreto do *software*
- ✦ Ex: queda de uma conexão TCP, variável com valor errado

### ● Falha

- ✦ **Decorrencia do erro**
- ✦ Serviço oferecido ao usuário não cumpre sua especificação
- ✦ Ex: banco de dados fora do ar, aplicação mostrando dados incorretos

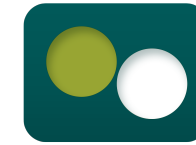
### ● Portanto:

- ✦



13

13



ASDPC

## Desempenho

Aspectos de Projeto de SDs

14

## Desempenho

### ● Métricas para medir desempenho:

- ✦ **Número de mensagens** trocadas
- ✦ **Tempo** de resposta
- ✦ **Throughput** (número de tarefas executadas / tempo)
- ✦ **Utilização** do sistema

### ● Em um sistema qualquer:

- ✦ + processadores, + memória, + capacidade de armazenamento ⇒ **melhoria do desempenho**

### ● Ao se distribuir os processos entre os processadores na rede:

- ✦ + **velocidade** final de computação?
- ✦ + custo de **comunicação**!

15

15

## Desempenho: Custos de Comunicação

### ● Componentes do **custo de comunicação**:

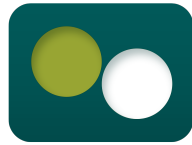
- ✦ Tempo de **processamento** do protocolo/*middleware*
- ✦ Tempo de **latência** do *hardware* e *software* de rede
- ✦ Tempo de **transmissão** da mensagem

### ● Para obter um **bom desempenho**:

- ✦ **Reduzir a comunicação** entre as entidades
- ✦ Buscar manter um bom nível de **paralelismo**
- ✦ Encontrar um ponto de **equilíbrio** entre ambos:
  - evitando: *sobrecarga da máquina e comunicação excessiva*

16

16



ASDPC

# Escalabilidade

Aspectos de Projeto de SDs

17



ASDPC

## Escalabilidade

- Noção intuitiva:
  - ✦ Um sistema distribuído que opera bem com 10 máquinas também deve funcionar bem com 10.000 máquinas.
- O desempenho do sistema não deve ser **degradado de forma acelerada** à medida em que o número de processos distribuídos cresce.

18



18



ASDPC

## Inimigos da Escalabilidade

- Componentes **centralizados**
  - ✦ por exemplo, um único servidor para todos os usuários
- Tabelas/BD **centralizadas**
  - ✦ por exemplo, um único arquivo com as informações acessadas pelo servidor
- Algoritmos **centralizados**
  - ✦ por exemplo, o roteamento de mensagens baseado em informações completas de caminho

19

19



ASDPC

## Melhorando a Escalabilidade

- Algoritmos **descentralizados** com as seguintes características:
  - ✦ Nenhuma máquina possui **informações completas** sobre o estado do sistema.
  - ✦ Máquinas tomam decisões baseadas apenas nas informações disponíveis **localmente**.
  - ✦ Falha de **uma das máquinas** não impede o funcionamento do algoritmo.
  - ✦ Não existe um **relógio global** implícito.

20

20





## Melhorando a Escalabilidade

### ● Sistemas **escaláveis**

#### ◆ **Servidores** distribuídos:

- *vários servidores cooperam para a execução de um serviço*

#### ◆ **Estruturas de dados** distribuídas:

- *divididas em partes e armazenadas em vários locais do sistema*

#### ◆ **Algoritmos** distribuídos:

- *cada servidor executa uma parte do algoritmo.*

