

Sistemas Distribuídos (DCC/UFRJ)

Aula 1: Apresentação e motivação da disciplina

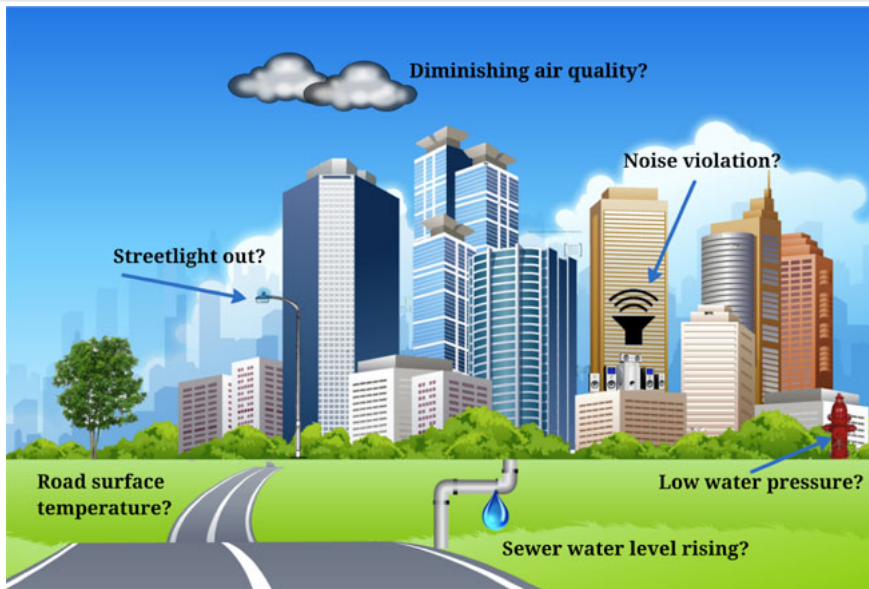
Prof. Silvana Rossetto

4 de abril de 2016

- 1 Conceitos básicos sobre sistemas distribuídos
 - Definição e motivação
 - Características e desafios

- 2 Visão geral da disciplina

Computação distribuída



Computação distribuída

- A **computação distribuída** envolve o projeto, implementação e avaliação de **algoritmos que executam em processadores distintos, interconectados por meio de canais de comunicação**
- Cenários de SDs: telecomunicações, serviços Web, computação científica, controle de processos de tempo real, “cidades inteligentes”, etc.

Os **SDs devem funcionar de forma correta**, mesmo com processadores individuais e canais de comunicação operando em diferentes velocidades, e mesmo na ocorrência de falhas em parte dos componentes do sistema

Motivação para sistemas distribuídos

- **Compartilhamento de recursos:** impressoras, páginas Web, registros de bases de dados, capacidade de processamento e armazenamento, etc
- **Gerenciamento de recursos por atores especiais:** servidores gerenciam recursos acessados por clientes remotos
- **Integração de diferentes agentes e fontes de informação:** tecnologias de sensoriamento, mobilidade, atuação, etc

Comunicação entre processos

Processos na mesma máquina usam primitivas de comunicação inter-processos (providas pelo SO), em máquinas distintas é preciso usar “troca de mensagens”:

- **Sockets:** processos enviam e recebem mensagens através do seu socket (interface entre a camada de aplicação e a camada de transporte)
- **Endereçamento de processos:**
 - nome ou endereço da máquina destino
 - identificador do processo na máquina destino (número da porta, ex., porta 80 para um servidor WEB)

Características de sistemas distribuídos

Características fundamentais

- Consistem de **componentes autônomos**
- Os usuários (ou programas) **vêm o sistema como um todo** (i.e., de uma forma ou de outra os componentes precisam colaborar)
- **Como estabelecer essa colaboração é o núcleo do desenvolvimento de sistemas distribuídos**

Consequências

- Execução de tarefas concorrentes
- Ausência de relógio global
- Independência de falhas (cada componente pode falhar isoladamente, enquanto os demais permanecem em execução)

Desafios principais para sistemas distribuídos

- 1 Heterogeneidade
- 2 Segurança
- 3 Escalabilidade
- 4 Tratamento de falhas
- 5 Concorrência
- 6 Transparência
- 7 Formas de acesso aos serviços

Heterogeneidade

- SDs são construídos sobre uma variedade de redes, SOs, plataformas de hardware e LPs
- Em geral, nenhuma premissa deve ser feita com relação ao tipo dos computadores (PCs, mainframes) e ao modo de interconexão (cabeada, sem fio)

Segurança

- **Segurança** em SDs envolve proteger os recursos compartilhados e as informações trocadas via mensagens

Escalabilidade

Dimensões de escalabilidade

Escalabilidade em SDs pode ser medida em três dimensões:

- 1 **Tamanho:** medida da dificuldade e custo de adicionar recursos e usuários ao sistema
- 2 **Geografia:** até que ponto usuários e recursos podem estar distantes uns dos outros
- 3 **Administração:** medida da dificuldade de gerenciar o sistema, especialmente quando envolve várias organizações

Escalabilidade

Técnicas de escalabilidade

- **Comunicação assíncrona:** oculta a latência de comunicação
- **Distribuição:** particiona um componente em partes menores e espalha essas partes pelo sistema
- **Replicação:** aumenta a disponibilidade e permite balanceamento de carga (pode gerar problema de consistência)

Tratamento de falhas

- Processos, rede e hardware podem falhar independentemente dos outros elementos
- Cada componente precisa ser projetado de forma a saber lidar com as possíveis falhas

Concorrência

- O desenvolvimento de um SD consiste em **dividir a aplicação em tarefas que podem ser submetidas à execução concorrente**
- Cada componente precisa ser projetado de forma a saber lidar com as questões referentes à concorrência (**exclusão mútua, estratégia de interação entre os elementos da aplicação e de controle da evolução da aplicação**)

Transparência

O conceito de **transparência** pode ser aplicado a vários aspectos de um SD:

- 1 **Acesso:** oculta diferenças na representação dos dados e no modo de acesso a um recurso
- 2 **Localização:** oculta o local que o recurso está armazenado
- 3 **Realocação:** oculta a possibilidade de movimentação do recurso enquanto em uso
- 4 **Replicação:** oculta o fato do recurso ser replicado
- 5 **Concorrência:** oculta o fato do recurso poder ser acessado por vários usuários simultaneamente
- 6 **Falha:** oculta a falha e recuperação de um recurso

Transparência

- Embora a transparência seja em geral desejada, **há situações nas quais ocultar completamente dos usuários o aspecto da distribuição não é a melhor opção**
- ex., atualizações que envolvem **várias réplicas em continentes distintos**: é melhor o usuário saber o custo para atualizar as informações
- outro ex., com a mobilidade dos dispositivos, a **noção de localização e percepção de contexto** pode ser útil e necessária

Formas de acesso aos serviços

- Um SD que oferece serviços conhecidos deve adotar regras padronizadas de sintaxe e semântica para garantir o acesso remoto

Objetivos da disciplina

Objetivo geral

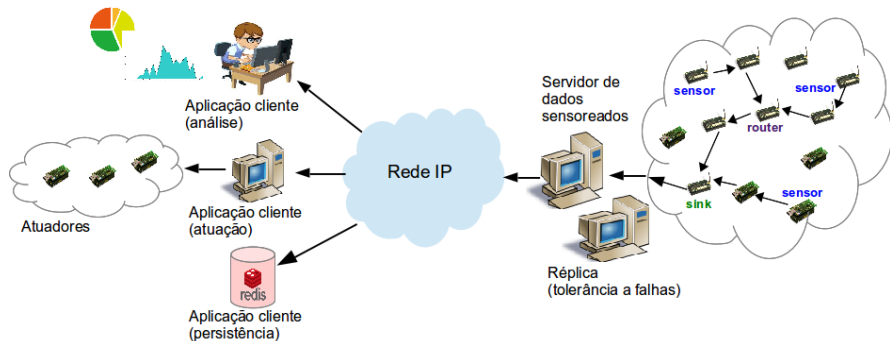
Apresentar as principais motivações e desafios da área de **computação distribuída**, e os algoritmos e modelos arquiteturais fundamentais para o desenvolvimento de sistemas distribuídos

[www.dcc.ufrj.br/\(til\)silvana/sd](http://www.dcc.ufrj.br/(til)silvana/sd)

Metodologia

Aulas teóricas intermeadas com exercícios práticos de implementação

Aplicação alvo



Programa da disciplina

- Conceitos fundamentais sobre sistemas distribuídos
- Comunicação entre processos via troca de mensagens
- Paradigmas de sistemas distribuídos (cliente/servidor, p2p)
- Modelos (de interação, de falhas de canais e processos)
- Sincronização e coordenação (relógios lógicos, causalidade, estados globais, eleição de líder, acordo distribuído, exclusão mútua distribuída)
- Tolerância a falhas
- Medidas de eficiência
- Abstrações de programação (RPC, publish/subscribe)
- Redes de sensores

Avaliação

- O cálculo da média final do aluno será feito através da seguinte equação:
$$M_f = (N_p * 0.5) + (T * 0.5),$$
 onde N_p equivale à nota de participação do aluno (frequência, participação nas aulas, implementação correta dos exercícios propostos).
- Se o aluno obtiver média final (M_f) **igual ou superior** a 7.0 ele será dispensado da prova suplementar e sua nota final será a média obtida ($N_f = M_f$).
- Caso contrário, se o aluno obtiver média final (M_f) **maior ou igual a 3.0 e menor que 7.0**, ele **deverá** fazer a P_s e sua nota final será dada pela equação:
$$N_f = (M_f + P_s)/2$$
- Os alunos que alcançarem nota final (N_f) igual ou superior a 5.0 e frequência mínima de 75% serão aprovados.

Referências bibliográficas

- ❶ *Sistemas Distribuídos - Princípios e Paradigmas*, A. S.Tanenbaum e M. Van Steen, 2ed., Pearson Prentice Hall, 2007
- ❷ *Introduction to Reliable and Secure Distributed Programming*, C. Cachin, R. Guerraoui and L. Rodrigues, Springer, 2011
- ❸ *Distributed Systems: Concepts and Design*, G. Coulouris, J. Dollimore and T. Kindberg, Addison Wesley, 4 ed., 2005
- ❹ *Redes de Computadores e a Internet*, Kurose and Ross, Pearson, 2010
- ❺ Artigos científicos