



# Introdução aos Sistemas Distribuídos



**prof. Talles (CMP)**  
***[talles@ucg.br](mailto:talles@ucg.br)***

# Introdução aos Sistemas Distribuídos

## Objetivos da Aula:

- Conseguir responder à pergunta: o que é um Sistema Distribuído?
- Relembrar “as regras do jogo” descritas na aula passada.

## Tópicos Abordados:

- Definição do Objeto de Estudo
- Exemplos de SD
- CMP1190 - ~~2014/2015 (G11)~~
  - O que vamos estudar?
  - Avaliações : Provas e Trabalhos
  - Referências Bibliográficas Adotadas

## O que é um Sistema Distribuído



Silberschatz (x - 2007)

Um Sistema Distribuído é uma coleção de processadores que **não compartilham memória** ou **relógio**.

...coleção de processadores **pouco acoplados**, interconectados por uma rede de comunicação.



## O que é um Sistema Distribuído

- **Taxonomia de Flynn (1972):** fluxo de instruções e fluxo de dados:
  - SISD (ex. PC antigo)
  - SIMD (controle central comandando processadores vetoriais, matriciais e etc...)
  - MISD (modelo teórico)
  - MIMD (Sistemas Distribuídos e Sistemas Paralelos)

### Tanenbaum (1992)

Fracamente acoplado

Fortemente acoplado

Sem memória  
compartilhada

Memória compartilhada



## O que é um Sistema Distribuído

### Definições mais antigas...

#### 1. Sistemas Paralelos:

*“A computer system in which two or more CPUs share full access to a common RAM”*

(STEEN, V. M. TANENBAUM, A. S. , 2001)

- Exploram paralelismo interno
- Multiprocessadores

#### 2. Sistemas Distribuídos:

*“Tightly-coupled CPUs that do not share memory”*

(STEEN, V. M. TANENBAUM, A. S. , 2001)

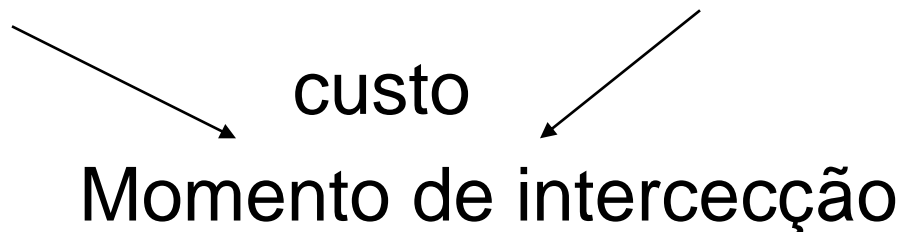
- Exploram paralelismo externo
- Multicomputadores



## O que é um Sistema Distribuído

### Definições mais antigas...

1. Processamento Paralelo: um conjunto de processadores trabalhando cooperativamente, onde cada qual atua num segmento do problema.
2. Paralelismo de tarefas
  - Alta granularidade: pouca ou nenhuma comunicação
  - Baixa granularidade: comunicação intensa
3. Processamento Paralelo X Multitarefa
4. Paralelismo Interno X Paralelismo Externo



**Ex. PVM**

## O que é um Sistema Distribuído



Coulouris (2005)

Um Sistema Distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens

## O que é um Sistema Distribuído



Tanenbaum (2007)

Um Sistema Distribuído é um conjunto de **computadores independentes** que se apresenta a seus usuários como **um sistema único e coerente**.



Colaboração



Componentes autônomos

Como tornar viável a colaboração?





## O que é um Sistema Distribuído

De acordo com Tanenbaum, principais **metas** (Coulouris chama de **Desafios**) que envolvem os projetos de SD:

- 1) Acesso a recursos compartilhados
  - *groupware*
- 2) Transparência
- 3) Abertura
- 4) Scaling (Escalabilidade)
- 5) Ciladas
  - Tolerância a falhas; segurança; heterogeneidade e topologia da rede; latência e largura de banda...

## O que é um Sistema Distribuído



## Acesso a recursos compartilhados

- Paradigmas de interação
  - Cliente/servidor
  - P2P
  - Modelo baseado em objeto
  - Publicar/Sobrescrever
  - Roteamento baseado em árvore
  - ...

*Veremos  
adiante...*

## O que é um Sistema Distribuído



Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Diferentes formas de **transparência** em um SD  
de acordo com ISO



## O que é um Sistema Distribuído

### Transparência: migração

- Migração de dados (como obter informações?)
  - Ftp automático (AFS original)
  - Transferência de partes de um arquivo (NFS, SMB)
- Migração de Computação
  - Hipótese : “Se o tempo para transferir os dados for maior do que o tempo para executar um comando remoto, o comando remoto deve ser utilizado”.
- Migração de Processos
  - Balanceamento de carga
  - Diminuição do tempo de Computação
  - Preferência de *hardware* específico
  - Preferência por *software* específico
  - Acesso a dados protegidos por domínio

## O que é um Sistema Distribuído



Ainda, de acordo com Coulouris:

***Performance transparency:*** allows the system to be reconfigured to improve performance as loads vary.

***Scaling transparency:*** allows the system and applications to expand in scale without change to the system structure or the application algorithms.



## O que é um Sistema Distribuído

## O contrário da Transparência? Translucidez

### Translucidez

```
import java.lang.reflect.*; → Meta-interface
public class DumpMethods {
    public static void main(String args[]) {
        try {
            Class c = Class.forName(args[0]);
            Method m[] = c.getDeclaredMethods();
            for (int i = 0; i < m.length; i++)
                System.out.println(m[i].toString());
        } catch (Throwable e) { System.err.println(e); }
    }
}
```

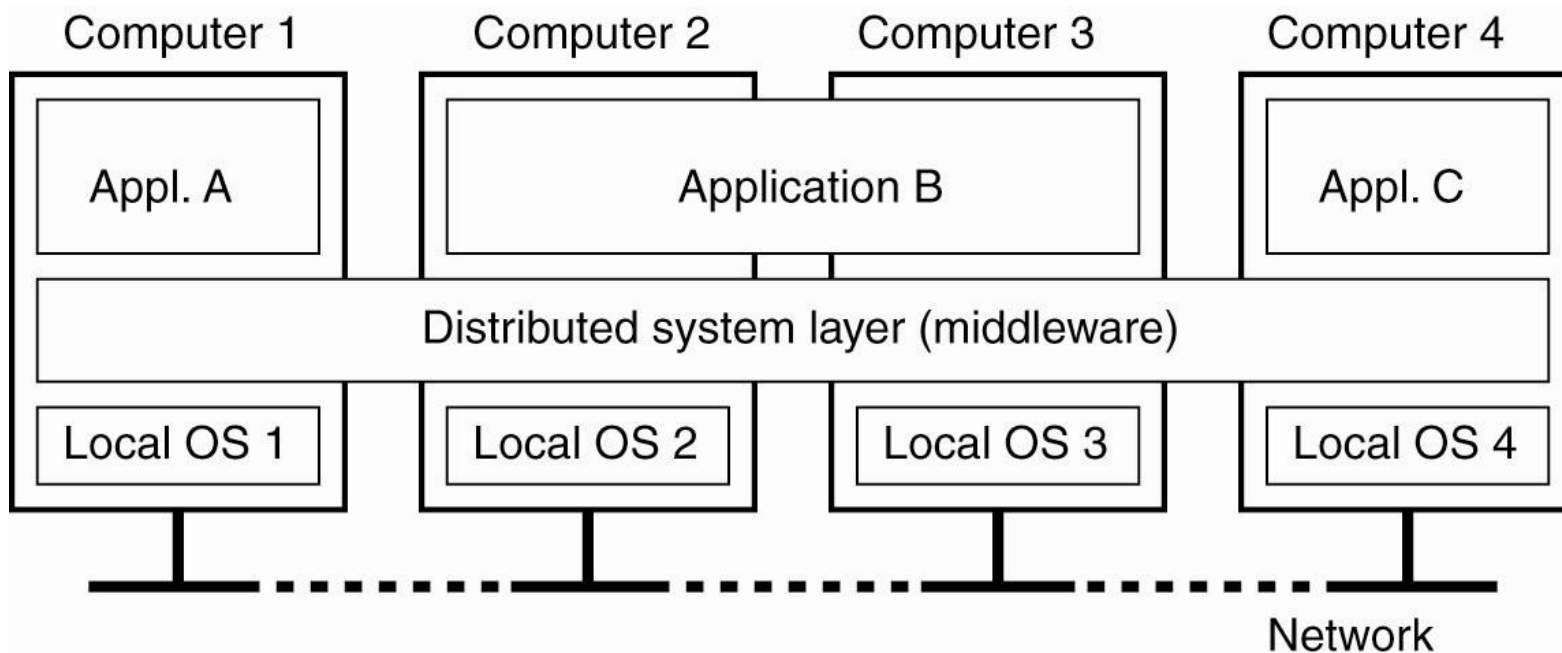
java DumpMethods java.util.Date

↓

```
public void java.util.Date.setMinutes(int)
public void java.util.Date.setDate(int)
public void java.util.Date.setYear(int)
public int java.util.Date.getMinutes()
public long java.util.Date.getTime()
public void java.util.Date.setTime(long)
public void java.util.Date.setMonth(int)
public void java.util.Date.setHours(int)
...
```

**Reflexão Estrutural**

## O que é um Sistema Distribuído



**Middleware:** organização mais utilizada em SD



## O que é um Sistema Distribuído

### Abertura:

“Um sistema distribuído aberto oferece serviços de acordo com regras padronizadas que descrevem a sintaxe e a semântica desses serviços”.

- Protocolos e Interfaces
- Interoperabilidade
- Portabilidade
- Extensibilidade
  - Adicionar e substituir componentes sem afetar os demais
- Separação entre política e mecanismo



## O que é um Sistema Distribuído



### **S.D.'s abertos : características**

- Sistemas que oferecem compartilhamento através de uma interface bem conhecida
  - IPC, RPC ou coisa parecida
- Possibilidade de heterogeneidade
  - Vários fornecedores
  - Vários clientes
  - Mesma interface



## O que é um Sistema Distribuído

### S.D.'s abertos : características

- Determina quanto o sistema pode ser estendido
  - Extensão de *hardware*
    - Adição de memória, periféricos...
  - Extensão de *software*
    - Recursos de *software*, protocolos, serviços...
- Grau de não interferência e não duplicidade quando da inclusão de algo novo





## O que é um Sistema Distribuído

### S.D.'s abertos : características

- Evolução para sistemas abertos
  - Ex. UNIX
  - Acesso às chamadas de sistema
    - Devem estar bem documentadas
  - `getpagesize()`, `fork()`, `waitpid()`...
- Abertura pela comunicação (anos 80)
  - Berkeley e o BSD UNIX
  - *Interprocess Communication*
    - Interface para comunicação entre processos
      - Locais
      - Remotos

Socket () ?!?!



## O que é um Sistema Distribuído

### Escalabilidade, de acordo com Tanenbaum:

- Em relação ao seu **tamanho**
  - Fácil adicionar mais usuários e recursos ao sistema
- Em termos **geográficos**
  - Usuários e recursos podem estar longe uns dos outros
- Em termos **administrativos**
  - Facilidade de gerenciamento mesmo que abranja muitas organizações administrativas diferentes

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Exemplos de limitações de escalabilidade



## O que é um Sistema Distribuído

### Tolerância a falhas: exemplo

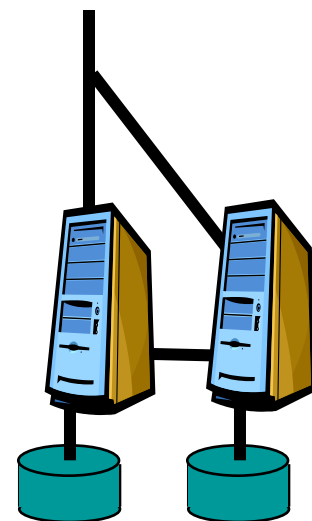
- Considere cinco S. Arquivos D's:
  - Probabilidade de cada um estar operacional a cada momento  $p = 0,95$ .
  - Probabilidade de todos estarem inoperantes  
 $= (1-p).(1-p).(1-p).(1-p).(1-p) = 0,05^5$   
Eventos indepedentes aos pares
  - Probabilidade de pelo menos um estar operante  
 $= p.2p.3p.4p = 24p^4 = 24(0,95)^4 > 0,95$



## O que é um Sistema Distribuído

### Tolerância a falhas: tipos de falhas

- Falhas previsíveis:
  - Redundância de *hardware*
    - Em S.Ds granularidade fina: replicar apenas os serviços críticos
  - *Software* tolerante?
    - Formas de recuperar o estado seguro
    - *Roll back; Shadow pages...*
- Falhas não-previsíveis
  - “Demissão do responsável pelo projeto”



## O que é um Sistema Distribuído



### Tolerância a falhas: Robusteza

- Capacidade de Detecção de falhas
- Resiliência
  - capacidade concreta de retornar ao estado natural de excelência, superando uma situação crítica.

## Exemplos de Sistemas Distribuídos:

De acordo com Tanenbaum,

### 1) Sistemas de Computação Distribuídos

*Cluster Computing Systems*

*Grid Computing Systems*

### 2) Sistemas de Informação Distribuídos

Sistemas de Processamento de Transações

Integração de Aplicações Empresariais

### 3) Sistemas Embarcados (ou embutidos) distribuídos ou Sistemas Distribuídos Pervasivos

Sistemas Domésticos

Sistemas Eletrônicos para o Tratamento da Saúde

Redes de Sensores

E a Internet?

Fez surgir  
vários outros  
SD lógicos  
ou virtuais

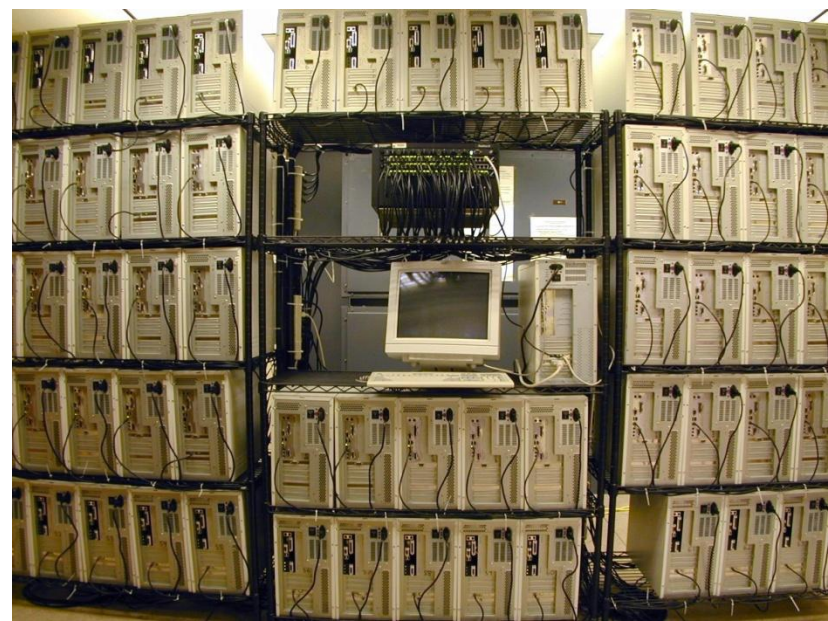
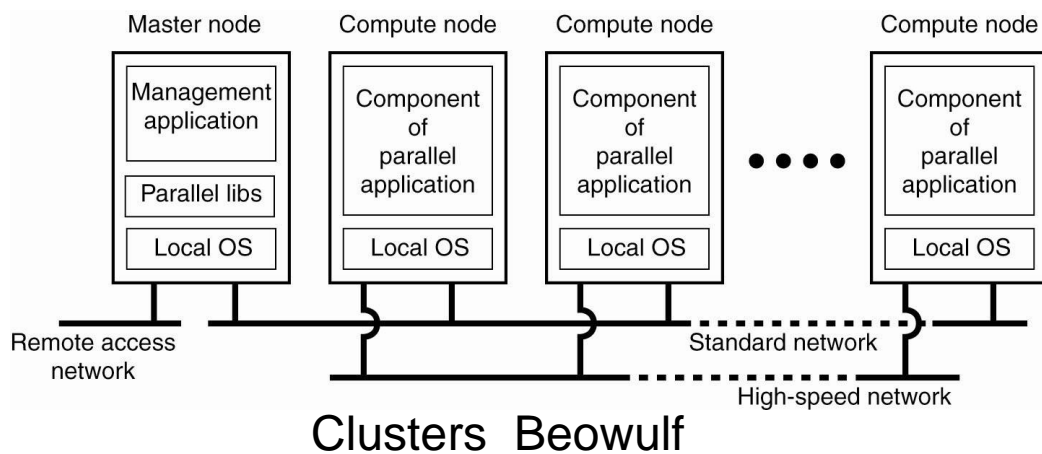




## Exemplos de Sistemas Distribuídos:

### 1) Cluster Computing Systems

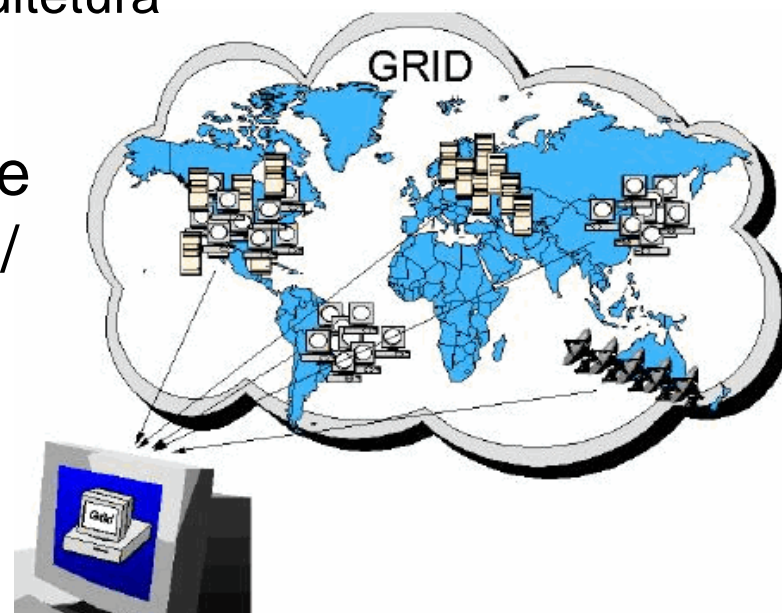
- Computação de alto desempenho (programação paralela)
- Homogeneidade
- Abordagem assimétrica (mestre-escravo)
  - Beowulf Linux
- Abordagem simétrica (MOSIX)



## Exemplos de Sistemas Distribuídos:

### 1) Grid Computing Systems

- Nós em domínios administrativos diferentes
  - Foco dirigido às questões da arquitetura
- Heterogeneidade
- Open Grid Services Architecture (OGSA) <http://www.globus.org/ogsa/>



## Exemplos de Sistemas Distribuídos:

### 2) Transaction Processing Systems

Atomicidade; Consistência; Isolamento e Durabilidade

Primitive	Description
BEGIN_TRANSACTION	Mark the start of a transaction
END_TRANSACTION	Terminate the transaction and try to commit
ABORT_TRANSACTION	Kill the transaction and restore the old values
READ	Read data from a file, a table, or otherwise
WRITE	Write data to a file, a table, or otherwise

Exemplos de primitivas de transações

## Exemplos de Sistema Distribuído:

### 2) Enterprise Application Integration: problema da heteogeneidade

- Diferentes plataformas de hardware
  - PCs, laptops, estações de trabalho, mainframes, palmtops, eletrodomésticos
- Diferentes sistemas operacionais
  - Windows Vista/XP/2000/NT/98/95, Solaris, Linux, MacOS X, PalmOS, PocketPC
- Diferentes tecnologias de rede
  - Internet TCP/IP, ATM
- Diferentes linguagens de implementação
  - C, C++, Java, Smalltalk, COBOL, ADA, VB, C#, ...

## 2) Enterprise Application Integration: problema da heteogeneidade

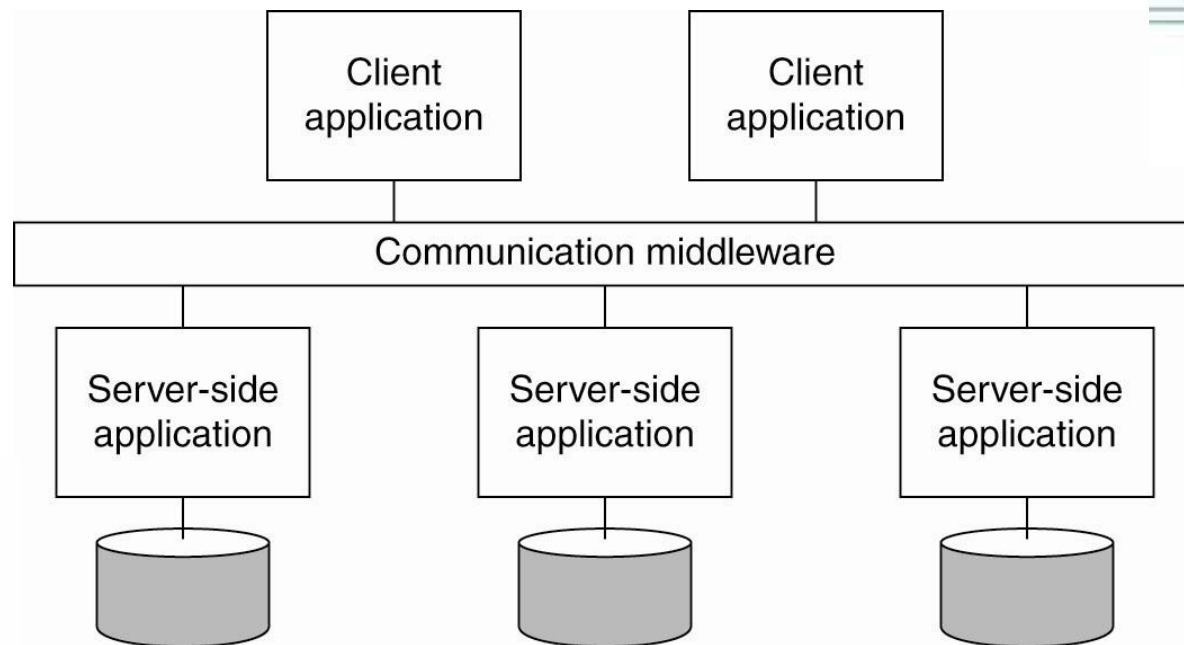
- Definição de regras de interoperabilidade padronizadas
  - Modelo de objetos comum (para desenvolvimento OO)
  - Protocolo de interação
  - Sintaxe de transferência de dados
- **Objetivo:** Permitir que aplicações distribuídas funcionem propriamente, a despeito da heterogeneidade

## 2) Enterprise Application Integration

- Consórcio que reúne cerca de 800 empresas envolvidas com tecnologia de objetos
  - fabricantes de middleware e aplicações
  - instituições de pesquisa
  - usuários
- Missão
  - “Criar um mercado de software baseado em componentes”
- Ênfase:
  - Re-uso de componentes
  - Interoperabilidade e portabilidade de componentes
  - Desenvolvimento de componentes padronizados



## 2) Enterprise Application Integration



Middleware como facilitador



## Exemplos de Sistema Distribuído:

### 3) Sistemas Domésticos

- Autoconfigurável
- Autogerenciável
- Gerenciamento do espaço pessoal
  - Recomendadores: identificar gostos (perfis) para configuração do espaço pessoal.

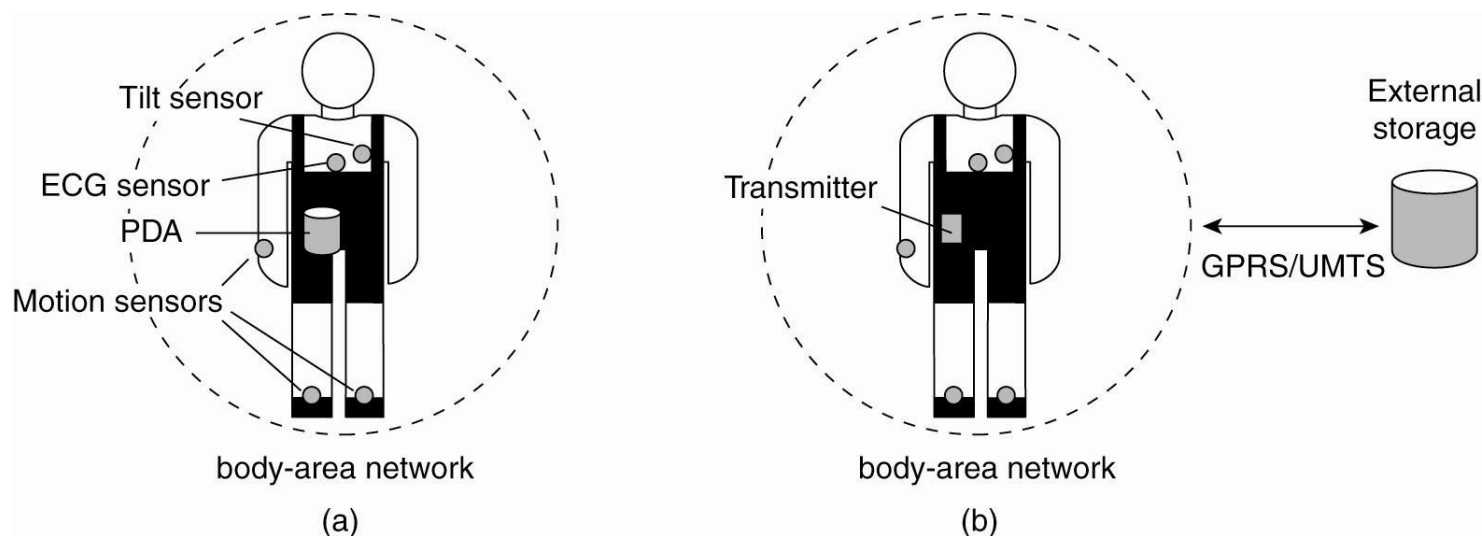


Fig. 1 : MeeSeries Product Overview



## Exemplos de Sistema Distribuído:

### 3) Sistemas Eletrônicos para o Tratamento da Saúde



*Body Sensor Networks - BSN*

## Exemplos de Sistema Distribuído:

### 3) Sistemas Eletrônicos para o Tratamento da Saúde

Alguns Desafios da perspectiva de Sistemas Distribuídos

- Onde e Como os dados monitorados deverão ser armazenados?
- Como evitar a perda de dados cruciais?
- Qual a infraestrutura necessária para gerar e transmitir sinais de alerta?
- Como os médicos podem dar o retorno *on-line*?
- Como pode ser alcançada a máxima robustez do sistema de monitoração?
- Quais são as questões de segurança e como as políticas adequadas podem ser impostas?
- ...
- Como alterar a configuração sem intervir na capacidade de operação autônoma?

## Exemplos de Sistema Distribuído:

### 3) Redes de Sensores

“Uma rede de sensores é um sistema distribuído composto por unidades autônomas (nós sensores) interconectados por um meio de comunicação”

- A maioria das aplicações utiliza enlaces **wireless**
- Na maioria das aplicações os nós sensores são **alimentados por baterias**
- A maioria das aplicações utiliza o roteamento **multihop**
- Requisitos do projeto são fortemente influenciados pelos requisitos da aplicação

Rede de computadores → fornecem serviços de comunicação

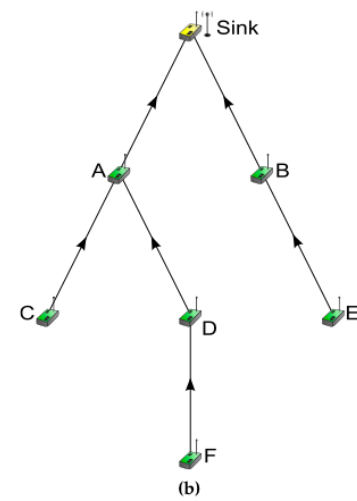
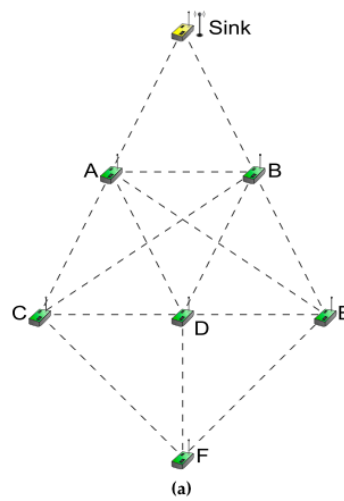
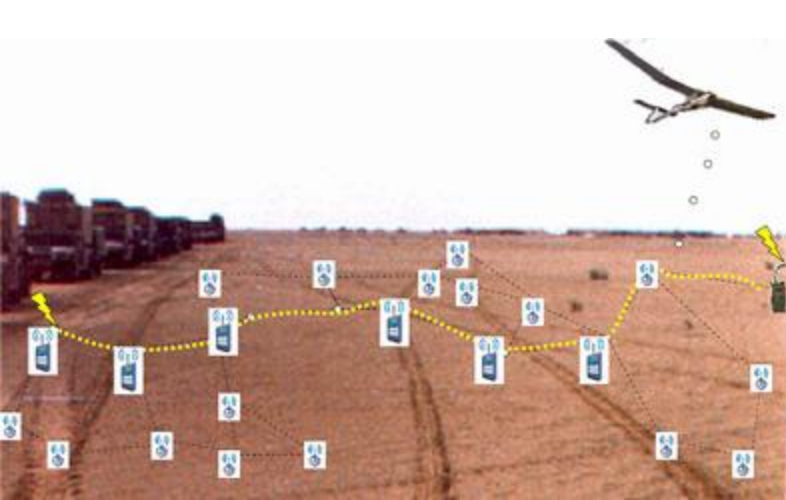
Redes de Sensores → são usadas para processar informações

## Exemplos de Sistema Distribuído:

### 3) Redes de Sensores

#### Como executar o processamento dentro da rede?

- Várias formas...
- Uma simples é repassar a consulta a todos os nós ao longo de uma árvore que abraja todos os nós distribuídos ao longo do evento de interesse ou de toda a rede. Na sequência, agregar as respostas em cada ramo da árvore.

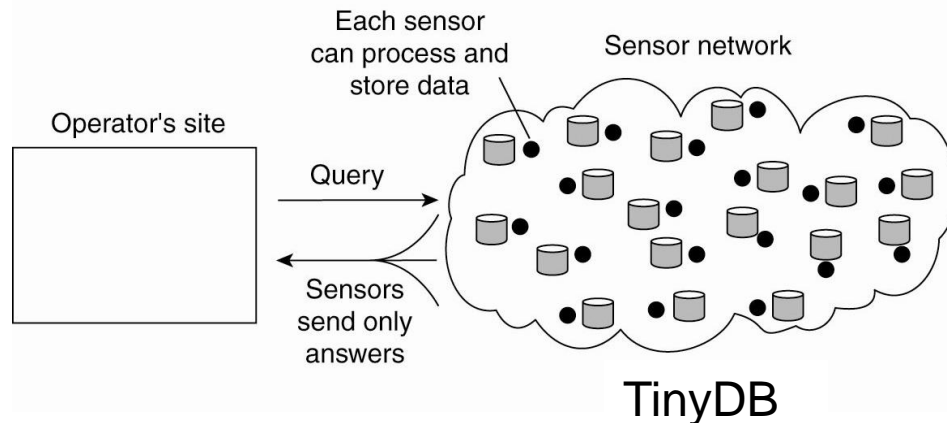
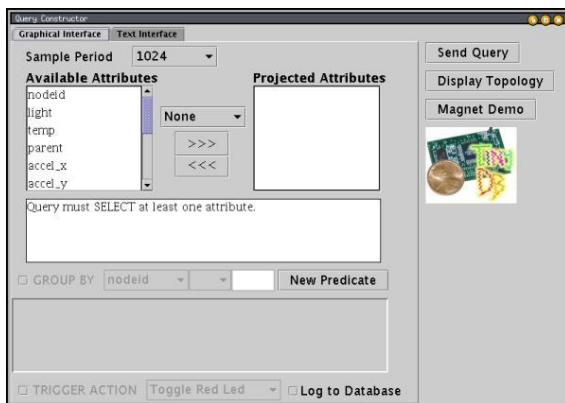


## Exemplos de Sistema Distribuído:

### 3) Redes de Sensores

#### Questões pertinentes para SD:

- Como montar dinamicamente uma árvore eficiente em uma rede de sensores?
- Como ocorre a agregação de resultados? Ela pode ser controlada?
- O que acontece quando os enlaces de rede falham?



## **Exercícios (para a próxima aula):**

- Do Tanenbaum, cap. 1: 13, 14 e 15.
  - Para a Prova 1, todos os exercícios do capítulo 1.
- Do Coulouris, cap. 1: 1.3, 1.5, 1.7 e 1.13

# Introdução aos Sistemas Distribuídos

**CMP1190 - ~~2011-1 (2011)~~**

## O que vamos estudar?

**UNIDADE 1: Apresentação do Curso, Motivação, Objetivos e Critérios Adotados para as Avaliações.**

**UNIDADE 2: Introdução**

Características e definições de Sistemas Distribuídos; Principais Desafios.

**UNIDADE 3: Modelos básicos**

Arquitetura cliente-servidor e sistemas *peer-to-peer*; sistemas multicamadas e *middleware*; código móvel e agentes.

# Introdução aos Sistemas Distribuídos

**CMP1190 - ~~2007-1 (001)~~**

## O que vamos estudar?

### **UNIDADE 4: Suporte do Sistema Operacional**

Comunicação interprocessos e o mecanismo de *sockets*; processos clientes e processos servidores; expressão da concorrência; threads e modelos para servidores concorrentes.

### **UNIDADE 5: Modelos de Middleware**

Chamada ao Procedimento Remoto; Middleware orientado a objetos e especificação CORBA; Tecnologia Java e o RMI; Middleware baseado em Serviços Web.



# Introdução aos Sistemas Distribuídos

**CMP1190 - ~~2017/2018 (30h)~~**

## O que vamos estudar?

### **UNIDADE 6: Sincronização Distribuída**

Sincronização com base no Tempo Físico; Relógio Lógico e o algoritmo de Lamport; Algoritmos para Garantia de Exclusão Mútua; Algoritmos para Eleição de Líder; Deadlocks em Sistemas Distribuídos; Análise da complexidade de algoritmos distribuídos.

### **UNIDADE 7: Outros serviços de middleware**

Transações: conceitos, propriedades, transações distribuídas, protocolo COMMIT, controle de concorrência, estudo de casos: CORBA OTS e JTS; Serviço de Nomes e Diretórios Distribuídos: conceituação; *name space*; estudo de casos; Sistemas de Arquivos Distribuídos: conceituação, questões de implementação: transparência de nomes, políticas de cache e replicação

# Introdução aos Sistemas Distribuídos

**CMP1190** ~~212044 (2011)~~

## O que vamos estudar?

### **UNIDADE 8: Tópicos Emergentes**

Redes de sensores e a plataforma SunSPOT; *Cloud Computing* e a plataforma GridGrain; TV Digital e o JavaTV; Mashup e Google Services APIs; Java FX Mobile; Bancos de Dados distribuídos e o Hibernate Search; Web 2.0 e o JSON; Sistemas Corporativos e a plataforma J2EE/EJB 3.1, Sistemas baseados em Coordenação e Jini/JavaSpaces,...

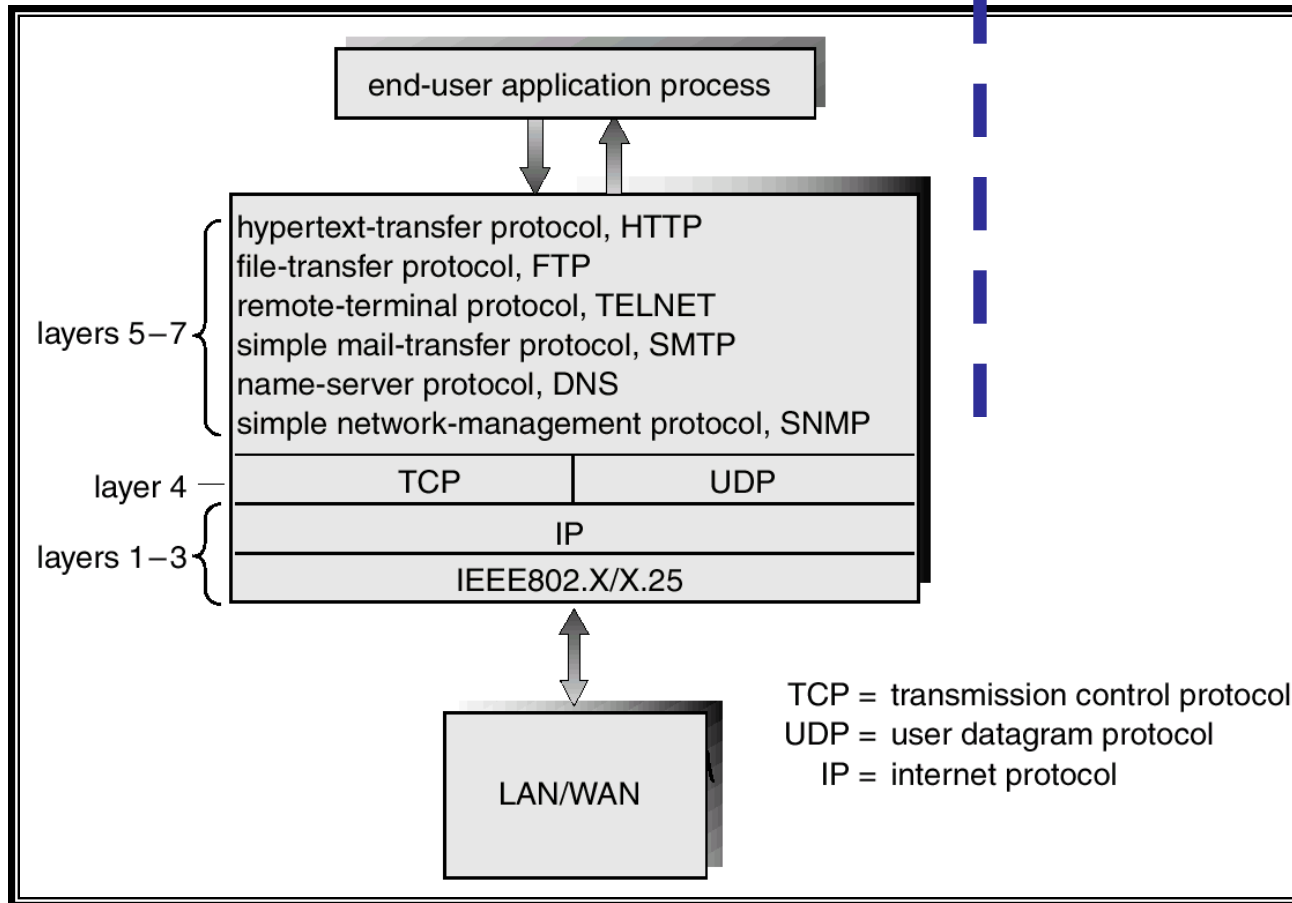
\*Podem ser incluídos outros assuntos de acordo com o interesse de vocês !

# Introdução aos Sistemas Distribuídos

CMP1190 - ~~2014/2015~~

Nosso curso do ponto de vista das tecnologias

## O que vamos estudar?



CMP1190 - ~~2024-1 (CMP1)~~

## Avaliações:

A Nota N1 é calculada pela seguinte expressão:

$$N1 = [\textit{Trabalho 1} + \textit{Prova 1}] / 2 + \textit{Atividades Complementares}$$

(máximo 2,0 pontos)

A Nota N2 é calculada pela seguinte expressão:

$$N1 = [\textit{Trabalho 2} + \textit{Seminário}] / 2 + \textit{Atividades Complementares}$$

(máximo 1,0 ponto)

# Introdução à Inteligência Artificial

CMP1190 - ~~2011/2012 (2011)~~



Entrega: ~~07/03/2013~~

## Avaliações:

O **Trabalho 1** é referente ao “**Projeto, desenvolvimento e avaliação de um SD para suporte a uma aplicação de propósito específico**”. É pré-requisito para a consecução do Trabalho 2, exigido para a composição da nota N2. A avaliação do Trabalho 1 é **individual**. Também, é exigido um **relatório por grupo**, no formato padronizado pela Sociedade Brasileira de Computação (SBC) ou pelo IEEE para artigos científicos. **Esses assuntos devem ser organizados em: I - Introdução, II – Materiais e Métodos, III - Resultados Preliminares e IV - Conclusões.**

# Introdução à Inteligência Artificial

CMP1190 - ~~2018/1 (C01)~~



## Avaliações:

A **Prova 1** tem o propósito de avaliar o **conteúdo cumulativo** ministrado durante o período da N1, definido pelo Calendário Acadêmico da PUC Goiás. É importante ressaltar que além do conteúdo teórico, abordado nas preleções, nessa avaliação, é também abordado todo o conteúdo referente à experimentação supervisionada pelo professor e todo o conteúdo das Atividades Complementares propostas.

# Introdução à Inteligência Artificial

**CMP1190 - ~~2014/1 (001)~~**



## **Avaliações:**

O **Seminário** é relativo aos **Tópicos Emergentes**. Tem o propósito de possibilitar que o aluno entre em contato com tecnologias e paradigmas não abordados durante o curso. É um trabalho em grupo, entretanto, a avaliação é individual. Para este trabalho é exigida uma apresentação, com os seguintes tópicos: I – Introdução (apresentação do tema, motivações, problema (s) em questão, justificativas); II – Tecnologias ; III – Ferramentas Disponíveis; IV – Protótipo (s) desenvolvido; V – Conclusões (análise crítica e fundamentada, comparação com os outros paradigmas e tecnologias estudadas no curso)

# Introdução à Inteligência Artificial

CMP1190 - ~~2011/12 (2011)~~

11 - ~~2011/12 (2011)~~

## Avaliações:

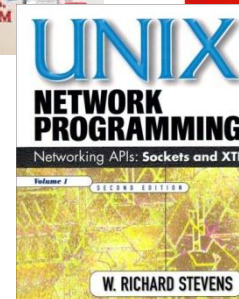
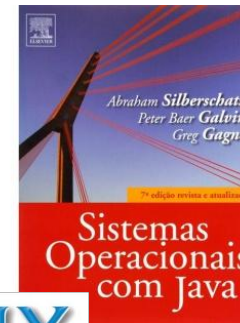
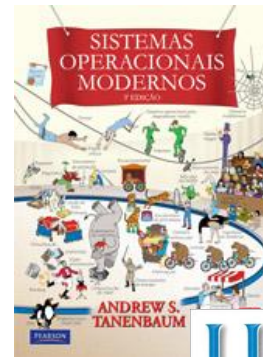
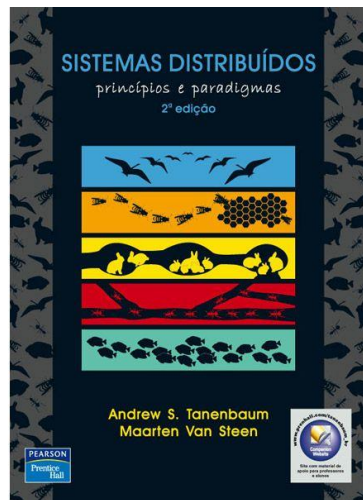
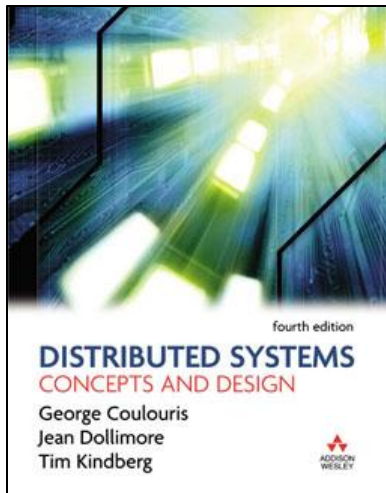
O **Trabalho 2** é a versão final referente ao tópico “**Projeto, desenvolvimento e avaliação de um SD para suporte a uma aplicação de propósito específico**”. O Trabalho 2 é avaliado por meio de uma **apresentação em grupo**, no formato de **seminário**, com a **demonstração do protótipo** desenvolvido ao longo do semestre. Também, é exigido um **relatório por grupo**, no formato padronizado pela Sociedade Brasileira de Computação (SBC) ou pelo IEEE para artigos científicos. Esse documento deve relatar todos os resultados do Trabalho, organizados em: **I - Introdução, II - Revisão Bibliográfica, III – Materiais e Métodos, IV - Tecnologias e Protótipos, V – Validação e Verificação e VI - Conclusões.**



# Introdução à Inteligência Artificial

CMP1190 - ~~24/07/2021~~ (2021)

## Referências Bibliográficas:



Bibliografia Bsica

Bibliografia  
Complementar

Apoio