

Sistemas Distribuídos

INF2545

Noemi Rodriguez
`noemi@inf.puc-rio.br`



Sistemas Distribuídos

- o que são: "coleção de máquinas independentes que aparecem para o usuário como um único sistema coerente"
 - que tipo de usuário?
 - » programador é usuário?
 - o que é "coerente"?
 - » o conceito de transparência
- um sistema distribuído é uma coleção de máquinas independentes que são usadas em conjunto para executar uma tarefa ou prover um serviço.



mensagem circulada no DEC SRC em 1987:

Received: by jumbo.dec.com (5.54.3/4.7.34)
id AA09105; Thu, 28 May 87 12:23:29 PDT

Date: Thu, 28 May 87 12:23:29 PDT

From: **lamport** (Leslie Lamport)

To: src-t

Subject: distribution

There has been considerable debate over the years about what constitutes a distributed system. It would appear that the following definition has been adopted at SRC:

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

...



para que queremos SDs?

- custo e desempenho
 - mais explorado em “programação concorrente e paralela” mas tb aqui
- escalabilidade
 - facilidade de aumentar recursos
- distribuição inerente
 - dispersão geográfica e social
 - compartilhamento de recursos
- confiabilidade
 - redundância



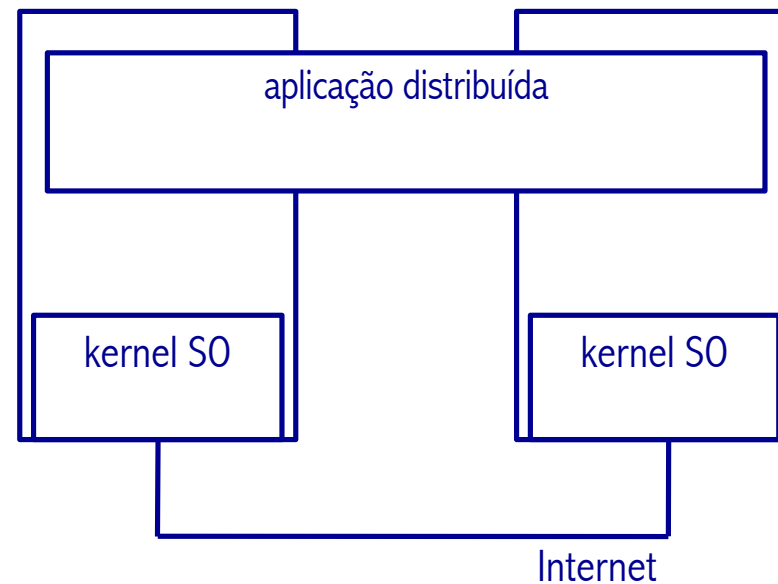
... mas...

- rede tem que ser levada em consideração
 - desempenho e falhas
- segurança
 - distribuição introduz problemas inexistentes em sistemas centralizados
- complexidade de software



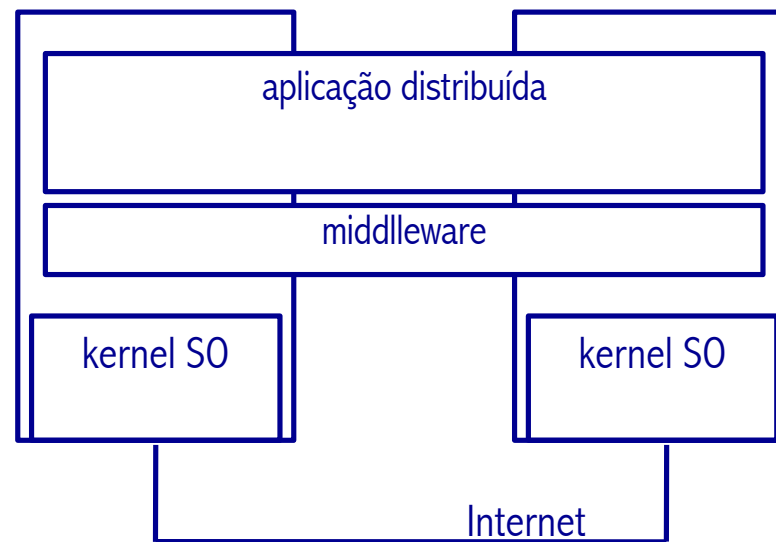
arquiteturas de interesse

- multicomputadores: cada um com sua memória e processadores, interligados por redes
 - construção de aplicação distribuída sobre recursos de redes de SO pode ser árdua



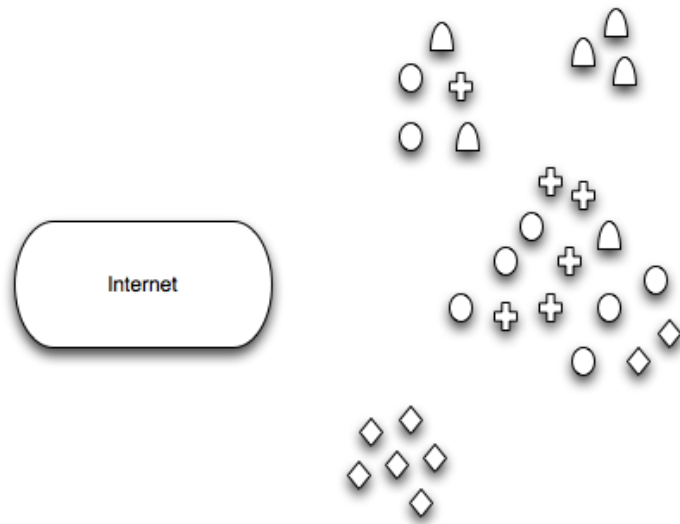
middleware

- serviços e abstrações que facilitam o desenvolvimento de aplicações distribuídas





IoT: arquiteturas



- interesse como sistema distribuído:
 - limitação de recursos traz “mistura de camadas”
 - insegurança
 - falhas



estudo de sistemas distribuídos

- livros clássicos de sistemas distribuídos
 - comunicação entre processos
 - sincronização e coordenação
 - replicação
 - sistemas de arquivos
 - segurança
 - confiabilidade
- nesse curso, ênfase também na **programação** de SDs e em graus de acoplamento distintos



programação

- requisitos de diferentes ambientes
 - redes locais e geográficas
 - baixo e alto acoplamento
 - linguagens e bibliotecas
 - protocolos abertos
- facilidades de programação importantes para diferentes classes de aplicações
 - ou mesmo para diferentes interações dentro da mesma aplicação
- necessidades
 - modelos de programação suportados:
 - » cliente-servidor, p2p, computação móvel, ...
 - comportamento diante de falhas
 - escala e acoplamento



programa do curso

- Comunicação: troca de mensagens. chamada remota de procedimentos e métodos. publish/subscribe. filas de mensagens. comunicação em grupo.
- Arquiteturas: cliente-servidor, p2p, eventos.
 - arquitetura interna: concorrência, threads e eventos
- Sincronização e Coordenação: relógios lógicos. ordenação de eventos, exclusão mútua. roteamento
- Replicação e consistência
- Tolerância a Falhas: comunicação confiável. recuperação de falhas. redundância.
- Outros:
 - localização
 - segurança
 - arquivos



discussão

- facilidade de desenvolvimento
- desempenho
- transparência
- escalabilidade
- flexibilidade



avaliação

- 4 trabalhos (implementação) – grupo ou individual
 - RPC
 - replicação (Raft)
 - roteamento/coordenação
 - integração de diferentes plataformas
- 4 resumos e críticas de artigos – individuais
- sobre prazos de entrega:
 - Cada aluno terá direito a 8 dias de atraso por semestre.



Bibliografia

- livros SDs:
 - M. van Steen e A. Tanenbaum. Distributed Systems. 2017. www.distributed-systems.net. [REQUISITAR CÓPIA DIGITAL]
 - Keong Lua, Heather Yu, John Buford, 2009. P2P Networking and Applications. Morgan Kaufmann.
 - Rachid Guerraoui, Luis Rodrigues. 2006. Introduction to Reliable Dist. Programming. Springer.
- surveys e artigos
 - alguns “clássicos”:
 - » Valeria Cardellini, Emiliano Casalicchio, Michele Colajanni, and Philip S. Yu. 2002. The state of the art in locally distributed Web-server systems. *ACM Computing Surveys*, 34(2), jun 2002.
 - » Amir Yahyavi and Bettina Kemme. 2013. Peer-to-peer architectures for massively multiplayer online games: A Survey. *ACM Computing Surveys* 46, 1, Article 9 (July 2013)
 - » outros...



introdução

- processos e comunicação

