

Introdução aos Sistemas Distribuídos

Sistemas Distribuídos 2023/24

O que é um sistema distribuído?

- ▶ Um sistema distribuído consiste num conjunto de componentes ligados através de uma rede.
- ▶ Estes componentes coordenam as suas ações por forma a resolver um determinado problema.
- ▶ A comunicação é feita unicamente através de troca de mensagens.

Motivação para usar sistemas distribuídos

- ▶ Partilhar dados e recursos entre utilizadores.
- ▶ Quando os recursos estão naturalmente em máquinas distintas (distribuição funcional).
- ▶ Distribuição e balanceamento de carga.
- ▶ Capacidade computacional.
- ▶ Separação física (segurança e fiabilidade).
- ▶ Razões económicas.

Desafios dos sistemas distribuídos

- ▶ Heterogeneidade
- ▶ Flexibilidade
- ▶ Segurança
- ▶ Escalabilidade
- ▶ Tratamento de falhas
- ▶ Concorrência
- ▶ Transparência
- ▶ Desempenho
- ▶ Integração de diferentes aplicações

Consequências práticas

- ▶ A concorrência é a norma.
- ▶ Existe não-determinismo na execução.
- ▶ Não existe um relógio global (sincronização).
- ▶ Latência na troca de mensagens e execução de procedimentos.
- ▶ É difícil manter conhecimento do estado global.
- ▶ Semântica de falha diferente (rede e máquinas).

Modelos arquiteturais

Quais são as entidades que comunicam?

- ▶ Processos, nós e threads
- ▶ Objetos distribuídos
- ▶ Web services

Que paradigma de comunicação é usado?

- ▶ Interprocess communication
- ▶ Remote invocation
- ▶ Indirect communication

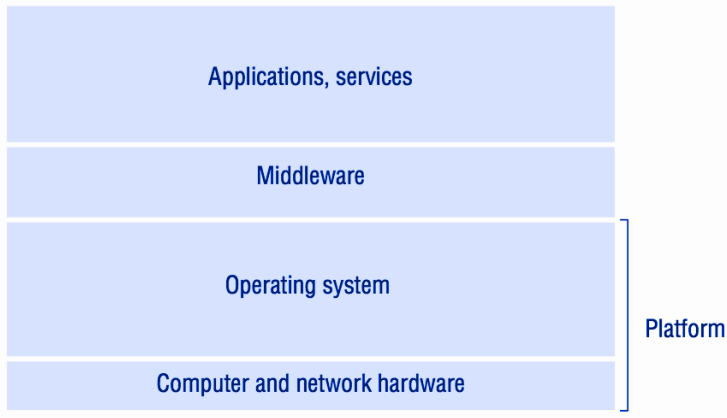
Qual são os papéis e as responsabilidades de cada entidade?

- ▶ Client-server
- ▶ Peer-to-peer

Como estão distribuídos fisicamente?

- ▶ Serviços em múltiplos servidores
- ▶ Caches, código móvel, agentes móveis

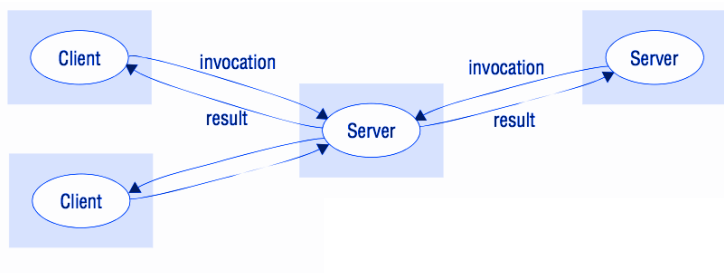
Camadas de serviços em sistemas distribuídos



Middleware

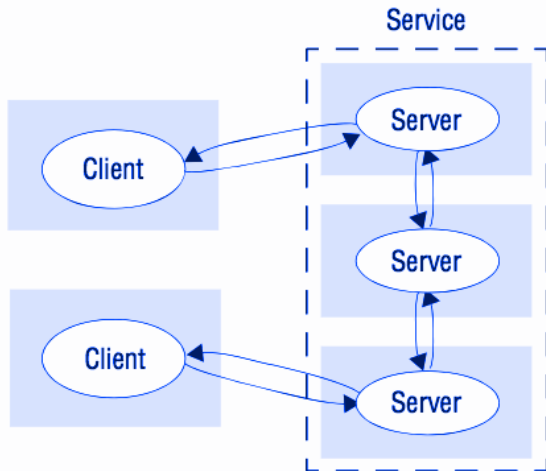
- ▶ É uma camada de software cujo objetivo é mascarar a heterogeneidade presente nos sistemas distribuídos e fornecer um modelo de programação conveniente aos programadores.
- ▶ Alguns exemplos:
 - ▶ Sun RPC (Remote Procedure Call)
 - ▶ OMG CORBA (Common Object Request Broker)
 - ▶ Microsoft DCOM (Distributed Component Object Model)
 - ▶ Java RMI
 - ▶ SOAP
 - ▶ REST
 - ▶ JBoss
 - ▶ JMS
 - ▶ ...

Modelo básico cliente-servidor



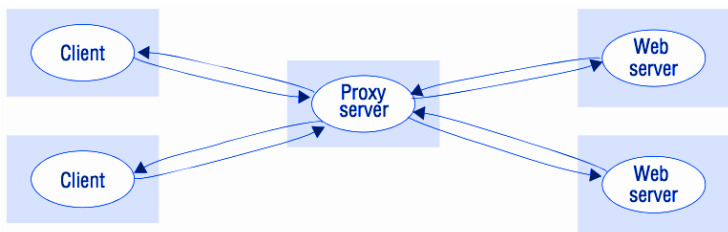
Processos cliente interagem com servidores individuais por forma a aceder a recursos ou dados. Os servidores podem, por sua vez, ser clientes de outros servidores.

Um serviço disponibilizado por múltiplos servidores



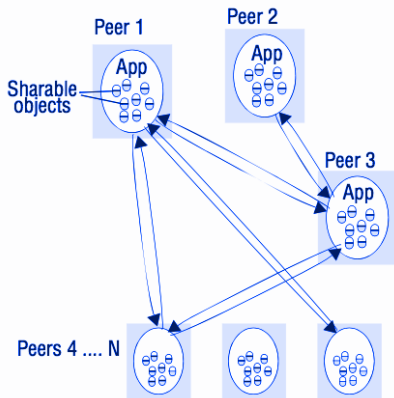
Um serviço pode ser implementado por diversos servidores (e.g., Web server clusters, parallel databases).

Proxies e caches



Uma cache é um armazém de dados usados recentemente, mais próxima de um cliente do que os próprios objetos (e.g., Web proxy servers).

Aplicações peer-to-peer



Todos os processos têm papéis semelhantes, cooperando sem distinção entre servidores e clientes (e.g., Gnutella, BitTorrent). Cada máquina é responsável por uma pequena parte da base de dados, do armazenamento, processamento e comunicação.

Classes de avarias

Classe de avarias	Afeta	Descrição
Fail-stop	Processo	O processo pára e permanece parado. Outros processos podem detetar este estado.
Crash	Processo	O processo pára e permanece parado. Outros processos poderão não detetar este estado.
Omission	Canal	Uma mensagem enviada nunca chega ao destinatário.
Send-omission	Processo	Um processo executa uma operação de envio, mas a mensagem nunca chega ao buffer de saída.
Receive-omission	Processo	Uma mensagem é colocada no buffer de receção de um processo, mas este nunca a recebe.
Arbitrary (Byzantine)	Processo ou canal	O processo ou o canal exhibe um comportamento arbitrário: pode enviar mensagens arbitrárias em momentos arbitrários; pode omitir mensagens; pode parar ou executar operações incorretas.

Avárias temporais

Classe de avarias	Afeta	Descrição
Relógio	Processo	O relógio local de um processo excede ritmo máximo de deriva do tempo real.
Desempenho	Processo	Um processo excede o intervalo máximo de tempo entre duas operações.
Desempenho	Canal	A transmissão de uma mensagem toma mais tempo do que o máximo estipulado.

Bibliografia

- ▶ George Coulouris *et al.*, Chapter 1, Distributed Systems: Concepts and Design, 5th edition, 2011.
- ▶ George Coulouris *et al.*, Chapter 2, Distributed Systems: Concepts and Design, 5th edition, 2011.