Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Licenciatura/Mestrado de Engenharia Informática e de Computadores

Sistemas Distribuídos

Semestre Verão 2011/2012

Luís Assunção

Objectivos

Aprender conceitos fundamentais de sistemas distribuídos, nomeadamente, os principais componentes e serviços necessários ao desenvolvimento de aplicações distribuídas sobre plataformas tecnológicas existentes.

Resultados de aprendizagem

Os alunos que terminam com sucesso esta unidade curricular serão capazes de:

1. **Descrever e discutir as vantagens, os problemas e os desafios que se colocam no desenvolvimento de sistemas distribuídos;**
2. **Desenvolver aplicações distribuídas usando mecanismos de comunicação e interacção de objectos e serviços distribuídos, identificando diferenças relativas ao paradigma de programação orientado a objectos (POO);**
3. **Descrever e discutir os componentes e serviços existentes nas plataformas (*middleware's*) mais conhecidas;**
4. **Analisar e discutir novos paradigmas emergentes na área de Sistemas Distribuídos.**

Avaliação dos resultados de aprendizagem

- **Os resultados de aprendizagem (1) e (3) são avaliados individualmente através de um exame final.**
- **O resultado de aprendizagem (2) é avaliado com base em trabalhos laboratoriais realizados em grupo e uma discussão final desses trabalhos. A qualidade dos relatórios (organização e clareza sobre as decisões tomadas) é um factor importante com peso na nota final;**
- **O resultado de aprendizagem (4) é avaliado através de um relatório de síntese e uma apresentação em sala de aula sobre um tópico emergente na área de Sistemas Distribuídos.**

Sobre o exame final

O exame final terá uma duração de 2 Horas com a seguinte estrutura:

1ª Hora – [Sem Consulta] Questões sobre conceitos e aspectos fundamentais das tecnologias utilizadas;

2ª Hora – [Com Consulta] Questões práticas de aplicação de conceitos e conhecimento de aspectos relacionados com as tecnologias utilizadas.

Nota Final:

(50%) Nota de exame >= 10 valores
(50%) Nota de avaliação de trabalhos práticos

Conteúdos Programáticos

- **O que é um Sistema Distribuído? Características fundamentais dos Sistemas Distribuídos. Conceito de *Middleware*. Modelos e arquitecturas de sistemas distribuídos: Cliente/servidor; Web; *n-tier*; *Peer-to-Peer*; Espaços de Tuplos; *Message Oriented Middleware (MOM);***
- **Evolução e caracterização da Programação Distribuída. Modelo *Socket* TCP/IP; modelo RPC (*Remote Procedure Call*) e o paradigma orientado a objectos em sistemas distribuídos: DCOM (Distributed Common Object Model); CORBA (*Common Object Request Broker Architecture*) e Java RMI (Remote Method Invocation);**
- **Objectos Distribuídos; Publicação de Objectos; Seriação de objectos; Objectos sem estado (*Stateless*) e com estado (*stateful*); Tempo de vida de objectos;**
- **Caso de estudo .NET Remoting: Canais de transporte TCP; HTTP e IPC; Objectos *Server Activated Objects* (SAO); Objectos *Client Activated Objects* (CAO) e publicação de Objectos (*Marshal*). Modelo de gestão do tempo de vida de objectos remotos em *.Net Remoting*. Chamadas assíncronas e *callbacks* em objectos remotos. Alojamento e configuração de objectos remotos no servidor Web *Internet Information System* (IIS);**

Conteúdos Programáticos (Cont.)

- Tempo e Coordenação em Sistemas Distribuídos. Dos relógios físicos aos relógios Lógicos de Lamport. Comunicação por grupos com *multicast* fiável e ordenação de eventos (caso de estudo: Spread Toolkit). Algoritmos distribuídos de exclusão mútua e de eleição.
- Segurança em Sistemas Distribuídos. Técnicas de implementação dos principais pilares da segurança: Privacidade; Integridade; Autenticação e Autorização.
- Arquitecturas orientadas ao Serviço (SOA). Introdução aos Web Services. Conceitos fundamentais: *XML Schema*; Estrutura das mensagens SOAP; *Web Service Definition Language* (WSDL);
- Exemplos de consolidação de conceitos em ambiente .NET Web Services. Acesso a Web Services existentes na Internet; Estratégias de Manutenção de Estado em Web Services;
- Interoperabilidade em sistemas distribuídos baseados em Serviços;
- Estudo de Caso: *Windows Communication Foundation* (WCF).

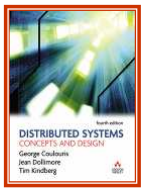
Bibliografia

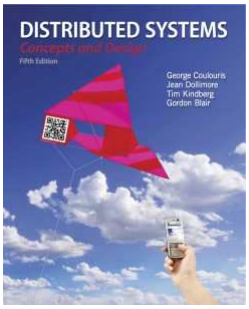
[B1] G. Coulouris, J. Dollimore, T. Kindberg, *Distributed Systems, Concepts and Design, Fourth Edition, ISBN 0321263545*, Addison-Wesley, 2005;

[B2] I. Ramer, *Advanced .NET Remoting, Second Edition, ISBN 1590594177*, Apress, 2005;

[B3-1] Juval Lowy, *Programming WCF Services, 3rd Edition*, O'Reilly, 2010

[B3-2] Steven Cheng, *Microsoft Windows Communication Foundation 4.0 Cookbook for Developing SOA Applications*, Packt Publishing, 2010;





George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair
Fifth Edition, published by Addison Wesley, May 2011

New to the fifth edition

New chapters:

Indirect Communication: Covering group communication, publish-subscribe and case studies on JavaSpaces, JMS, WebSphere and Message Queues.

Distributed Objects and Components: Covering component-based middleware and case studies on Enterprise JavaBeans, Fractal and CORBA.

Designing Distributed Systems: Devoted to a major new case study on the Google infrastructure.

Topics added to other chapters: Cloud computing, network virtualization, operating system virtualization, message passing interface, unstructured peer-to-peer, tuple spaces, loose coupling in relation to web services.

Other new case studies: Skype, Gnutella, TOTA, L2imbo, BitTorrent, End System Multicast.

More details about these and other changes are in the Preface.

Outras Referências

[R1] Distributed Systems, ***Principles and Paradigms***
Andrew S. Tanenbaum - Prentice-Hall

[R2] Reliable Distributed Systems
K. Birman, Springer, 2005

[R3] Distributed .NET Programming in C#
Tom Barnaby – Apress

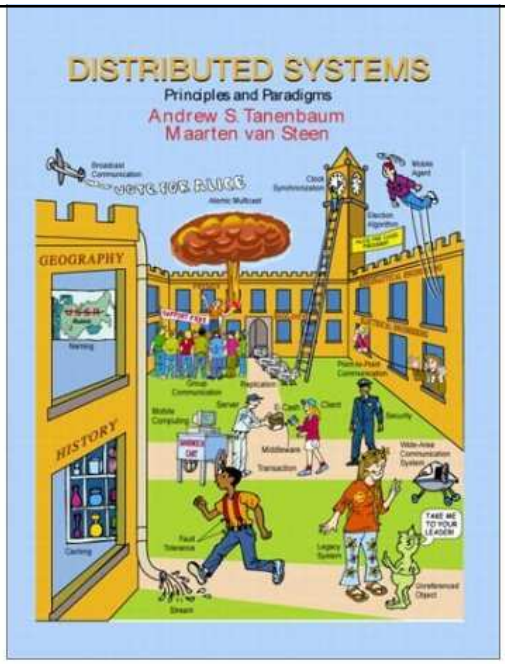
[R4] Microsoft .NET Remoting
Scott McLean, James Naftel, Kim Williams

[R5] Professional .NET Network Programming
Andrew Krowczyk e outros – Wrox

- **Site no Moodle**
  - **Artigos/Livros sobre temas específicos**
  - **Cópia dos transparentes de acompanhamento das aulas**

O que é um Sistema Distribuído ?

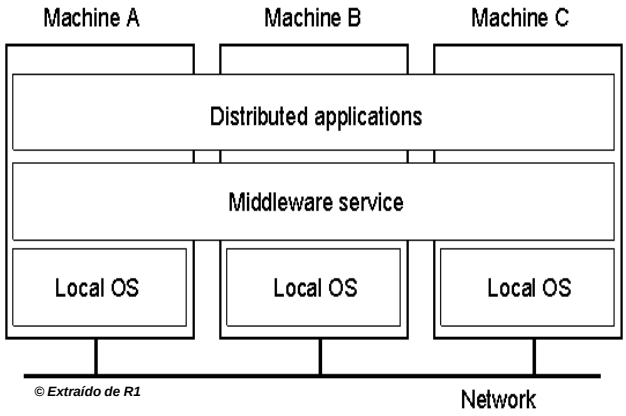
- **A confusão mais comum : “Uma rede de computadores é um sistema distribuído.”**
- **Uma rede de computadores é uma infra-estrutura servindo um conjunto de computadores interligados através de ligações físicas que suportam protocolos de comunicação.**
- **Um sistema distribuído é um sistema composto de vários computadores que comunicam através de uma rede e que alojam processos que usam protocolos distribuídos para assistirem a execução coerente de actividades distribuídas e cujo resultado pode ser visto como se só existisse um único computador.**
- **Um sistema distribuído partilha um estado e coopera para a obtenção de um objectivo comum. Em contrapartida os computadores de uma rede não interagem entre si limitando-se simplesmente a receberem ou enviarem mensagens ocasionais (por exemplo mail).**



Algumas definições

1. **"Um sistema distribuído é uma colecção de computadores independentes que aparecem aos utilizadores como um único sistema coerente – *Andrew Tanenbaum*"**

- **Esta definição revela dois aspectos: O primeiro relaciona-se com Hardware. O segundo relaciona-se com Software, fazendo com que os utilizadores "vejam" um sistema único.**



© *Extraído de R1*

Middleware
Camada de software que proporciona um modelo computacional de programação, mascarando a heterogeneidade do Hardware, Sistemas Operativos e Linguagens de Programação.
Normalmente disponibiliza um conjunto de blocos (building blocks) que facilitam o desenvolvimento de sistemas distribuídos.

2. **"Um sistema distribuído é um conjunto de blocos de hardware ou software, localizados em computadores de uma rede, que comunicam e coordenam as suas acções unicamente através de mensagens – *Coulouris*"**

Esta definição tem as seguintes consequências:

- **<u>Concorrência</u> – O trabalho é feito concorrentemente em cada computador, partilhando recursos (páginas Web, ficheiros, etc.) quando necessário. A capacidade do sistema pode aumentar adicionando recursos. Surge, assim, a necessidade de coordenar a concorrência entre programas que se executam em máquinas diferentes e que partilham recursos;**
- **<u>Não existe um relógio global</u> – Dado que é impraticável pensar que todos os computadores têm o mesmo valor de relógio, os programas que precisam de cooperar, coordenam as suas acções através de mensagens, não podendo ter em conta a noção de tempo dada pelo valor de um relógio;**
- **<u>Falhas independentes</u> – Qualquer computador na rede ou programa pode falhar, independentemente, sendo responsabilidade dos arquitectos do sistema distribuído, planear as consequências das possíveis falhas.**

3. **"Um sistema distribuído é um sistema que não permite que um utilizador trabalhe por causa da falha de um computador que o utilizador nunca tinha ouvido falar - *Leslie Lamport*"**

- **Embora muitas vezes se fale ou escreva sobre sistemas distribuídos, focando os benefícios da distribuição, é muito importante realçar que a dependência do sistema de uma colecção de máquinas interligadas numa rede tem os seus problemas.**

- **A famosa frase de *Lamport* pretende exactamente ilustrar isso.**
- **Se assumirmos que:**
  - **Cada computador pode falhar independentemente;**
  - **Que os computadores comunicam através de uma rede que pode falhar e introduzir atrasos;**
  - **Que os vários programas também podem falhar (terminações anormais);**

  é necessário ter em conta que, para além da colaboração entre as diversas partes, estas também podem falhar.

Uma questão não consensual: Será a Internet um sistema distribuído ?

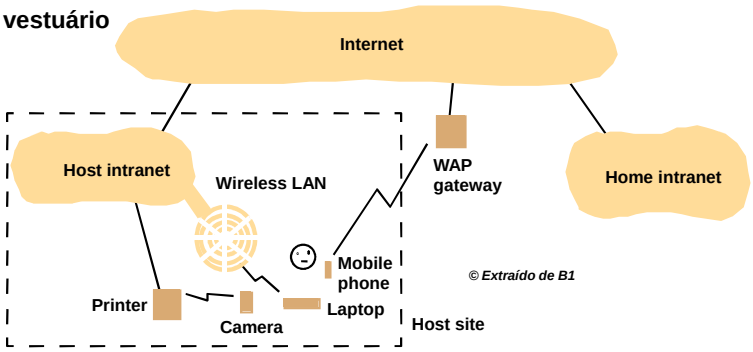
- **Não é um sistema distribuído se pensarmos na Internet como uma forma de milhões de computadores em todo o mundo comunicarem entre si através de protocolos standard (TCP/IP, http, ftp, smtp,, telnet, etc.);**

- **No entanto, existem hoje muitos sistemas distribuídos construídos no topo da Internet: Grandes Base de Dados (motores de pesquisa – Web Search); Comércio Electrónico; Aplicações *Groupware* (e-mail, *chats*, *forum*); Redes Sociais; Jogos online para milhares de jogadores; Messaging (MSN, Skype) etc.**

- **Por outro lado, hoje em dia, as grandes empresas (organizações) constroem os seus sistemas corporativos (*ERP-Enterprise Resource Planning*, Gestão Documental, *Workflow*, CRM, etc.) de forma distribuída, num contexto restrito (intranet), usando o paradigma da infra-estrutura subjacente à Internet.**

Computação Móvel e Ubíqua (*Mobile computing e Ubiquitous computing*)

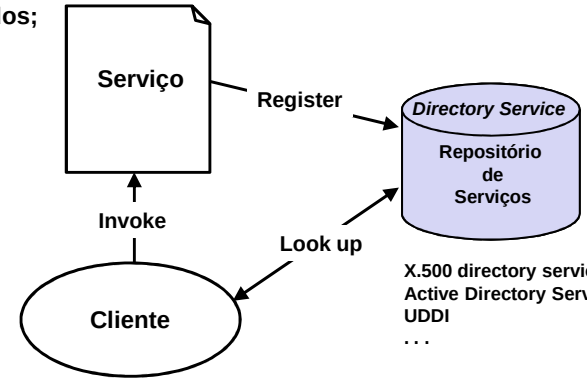
- A miniaturização e as comunicações sem fios (*wireless*) permitem a integração de dispositivos pequenos e portáveis nos sistemas distribuídos:
  - Computadores *Laptop*;
  - Dispositivos de bolso: PDAs (*Personal Digital Assistants*), telemóveis, câmaras fotográficas e de vídeo, relógios;
  - Dispositivos incorporados (embedded) em electrodomésticos, carros e até no vestuário



Partilha de Recursos

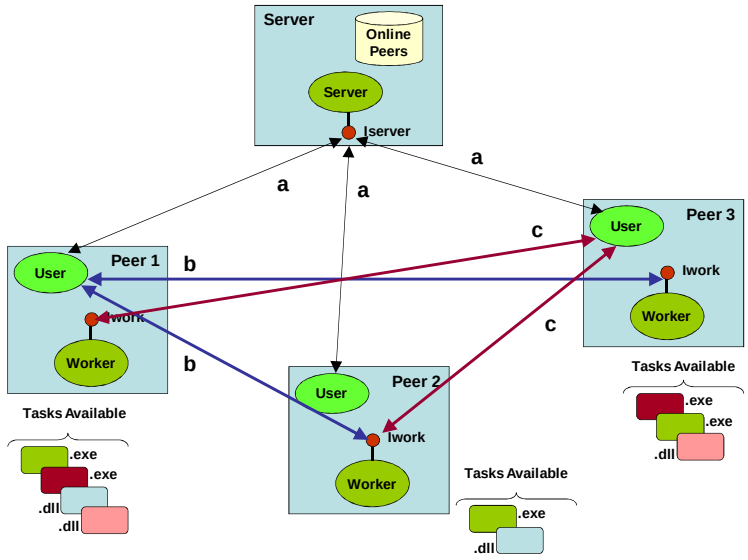
(Um objectivo dos sistemas distribuídos é a partilha de recursos)

- Recursos Hardware e software (computadores, impressoras, discos, ficheiros …)
- Partilha de dados (páginas Web, bases de dados …)
- Trabalho cooperativo (CSCW- *Computer Supported Cooperative Working* ou *GroupWare*): email; chats; forums
- Acesso a Serviços partilhados;



X.500 directory service;
Active Directory Service;
UDDI
. . .

Partilha de Recursos – Um exemplo



Grid Computing – *The Dream*

http://www.gridcafe.org/

"*Imagine-se muitos computadores, digamos vários milhões, dos quais alguns são PC's, mainframes e super computadores, mas também armazéns de dados, instrumentos e sensores meteorológicos e outros dispositivos de visualização.*

Imagine-se que todos eles estão situados em todo o mundo, apesar de pertencerem a diferentes pessoas (estudantes, médicos, engenheiros, etc.) ou organizações (empresas, universidades, hospitais, etc.).

Imagine-se que todos estes recursos estão ligados em rede, por exemplo a Internet.

Até aqui nada de novo, mas imaginemos que temos uma ferramenta mágica que coloca todos os recursos a actuar como um simples, enorme e potente computador. Este enorme computador é o que muitos sonhadores pensam que virá a ser a Grid".

http://www.eu-emi.eu/

European Middleware Initiative (EMI) - "is a unified, standardized, easy to install, software for distributed computing infrastructure"

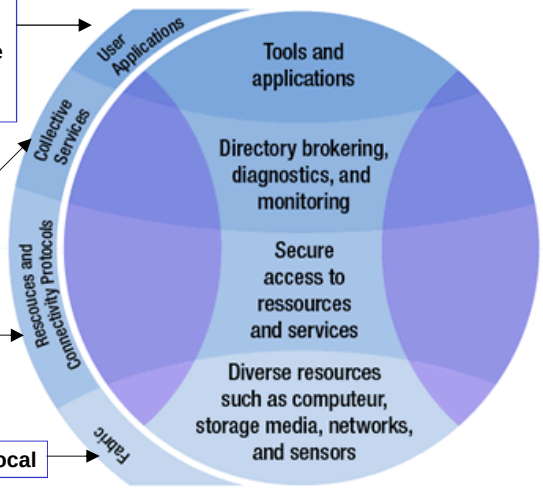
Arquitectura de uma *Grid*

Aplicações dos utilizadores, construídas usando os serviços disponibilizados pelas camadas inferiores, por exemplo, gestão de recursos, acesso a dados, descoberta de recursos etc.

Protocolos e serviços para encapsular interacções e comportamentos de partilha de recursos

Comunicação fácil e segura

Interfaces para controlo local



Open Grid Services Architecture (OGSA)

- Em 2001, surgiu o *Global Grid Forum* (GGF), como um corpo de definição de standards para as tecnologias *Grid*, tendo como objectivo produzir uma especificação standard designada por *Open Grid Services Architecture* (OGSA) com os seguintes requisitos:
  - Identificar e caracterizar as componentes funcionais que possam ser expressas numa forma reutilizável em diferentes configurações;
  - Orientação ao serviço garantindo a existência de um tratamento uniforme para todas as entidades, isto é, o comportamento da camada *Collective* possa ser expressa como a virtualização dos recursos existentes;
  - Alinhamento com o standard de facto *Web Services*, por exemplo poder expressar a interface de um *Grid Service* através da linguagem *Web Service Definition Language* (WSDL).

Globus Toolkit - Middleware que implementa o standard OGSA
http://www.globus.org/toolkit/

Cloud Computing

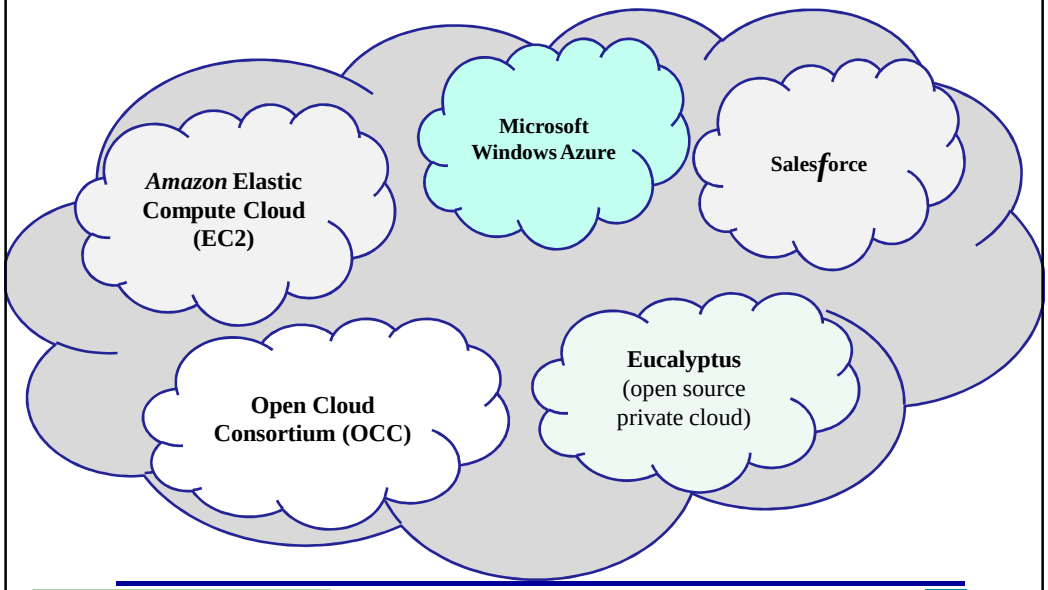
NIST National Institute of Standards and Technology
Information Technology Laboratory

" a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. "

Cloud Computing stack

Service Class	Access & Tools	Service contents
SaaS Software as a Service	Web Browser	Cloud Applications – Social Networks, email, Office suites (Google docs), CRM, Video processing
PaaS Platform as a Service	Development Environments	Cloud Platform – Programming languages, frameworks, Mashups editors, Data Storage models
IaaS Infrastructure as a Service	Virtualization Manager	Cloud Infrastructure – Computer servers, Data Storage, Firewall, Load Balancer, IP Addressing

Cloud of Clouds !



The Eight Fallacies of Distributed Computing

Peter Deutsch

Essentially everyone, when they first build a distributed application, makes the following eight assumptions. All prove to be false in the long run and all cause *big* trouble and *painful* learning experiences.

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

For more details, read the article by Arnon Rotem-Gal-Oz

http://nighthacks.com/roller/jag/resource/Fallacies.html

Características Fundamentais dos Sistemas Distribuídos

- **Heterogeneidade**
- **Sistemas Abertos**
- **Segurança**
- **Expansibilidade (*Scalability*)**
- **Tratamento de Falhas**
- **Concorrência**
- **Transparência**
- **Qualidade de Serviço (QoS-Quality of Service)**

Heterogeneidade

- **Diferentes topologias de rede;**
- **Hardware de computadores;**
- **Sistemas Operativos;**
- **Linguagens de Programação;**
- **Implementações em diferentes ambientes, usando diferentes *middlewares***

- **Através de várias organizações de normalização é, hoje, possível ter disponível um conjunto de standards e iniciativas abertas, que facilitam a interoperabilidade entre sistemas heterogéneos:**
  - **TCP/IP, HTTP, HTML**
  - **Conceito *Plug and Play***
  - **XML - Extensible Markup Language (XML)**
  - **WSDL-Web Services Definition Language**
  - **SOAP–Simple Object Access Protocol**
  - **ODBC - Open Database Connectivity**
  - **Código móvel (Java applets)**

Sistemas Abertos (Openess)

- Os sistemas distribuídos abertos são formados por sistemas heterogéneos (Hardware e Software);
- A especificação e documentação das interfaces chave é do domínio público, incluindo implementações de referência (Ex: Java Enterprise Edition);
- Em ambiente distribuído, são disponibilizados mecanismos de comunicação e publicação dos serviços para acesso aos recursos partilháveis (Ex: WSDL)
- A interoperabilidade assenta no esforço de grupos Nacionais e Internacionais de normalização, partindo de um modelo de cooperação, como é exemplo os RFCs (*Request For Comments*) (Ex: Protocolos TCP/IP, HTTP);
- Os sistemas distribuídos abertos são baseados em mecanismos de comunicação uniformes.

Segurança

- **Privacidade** - (confidencialidade) - protecção contra acessos não autorizados;
- **Integridade** - Protecção contra alterações e corrupção dos dados;
- **Autenticação** – Garantia de que os interlocutores são quem dizem ser;
- **Autorização** – Os utilizadores autenticados podem ter permissões diferentes para as diferentes actividades;
- **Disponibilidade** – Protecção contra interferências no acesso ao serviço;

Os desafios:

- Protecção contra ataques de negação do serviço (*denial of service*), por exemplo, “bombardeamento” de um serviço com pedidos, impossibilitando outros utilizadores de o usar;
- Segurança de código móvel

Expansibilidade (*Scalability*)

- Um sistema é expansível (*scalable*) se permanece efectivo (disponível, com desempenho e tempos de resposta aceitáveis) quando é aumentado um número aceitável de recursos e utilizadores;
- O desenho de sistemas distribuídos expansíveis levanta os seguintes desafios:
  - Custo controlado de recursos físicos (computadores, memória principal, memória secundária, periféricos, …)
  - Avaliação antecipada da evolução do sistema (volumes de informação, engarrafamentos, espaço de nomes (saturação de endereços IPv4 (32 bits), IPv6 (128 bits);
  - Controlo da perda de desempenho – Quais os indicadores;

Tratamento de falhas (Fault tolerance)

- As falhas num sistema distribuído são parciais, isto é, algumas componentes falham, enquanto outras continuam em função. No entanto, detectar e tratar falhas parciais é extremamente difícil;
- Técnicas para lidar com falhas:
  - Redundância do hardware (*clustering*, RAID, etc.)
  - Recuperação de erros por parte do software (mensagens podem ser retransmitidas e os dados de um ficheiro podem ser escritos em pares de discos);
  - Um utilizador deverá poder continuar a sua actividade numa estação de trabalho alternativa;
  - Replicação de recursos e servidores;
  - Reposição de estados anteriores após uma falha (*rolled back*);
  - Grupos de processos na implementação de aplicações distribuídas;
  - A disponibilidade (*Availability*) é maior num sistema distribuído - maior facilidade em disponibilizar recursos alternativos.

Concorrência

- Processos executados em concorrência
  - um processador versus múltiplos processadores
- Processos executados em paralelo
  - estação de trabalho com N processadores (N > 1) - Multi-processador
  - múltiplos servidores distribuídos - Multi-computador
- Servidores concorrentes na resposta a clientes (múltiplas *Threads* para atender os pedidos ao servidor)
- Controlo da Concorrência
- Mecanismos de sincronização, exclusão mútua, gestão de filas de espera (prioridades), Eleição; Chamadas assíncronas, etc.
- Transacções Distribuídas

Transparência

[Reference Model for Open Distributed Processing - ISO RM-ODP]

- Ao acesso
  - o acesso a recursos locais e remotos, usando operações idênticas.
- À localização
  - permite que recursos em diferentes locais, sejam acedidos do mesmo modo sem conhecimento da sua localização.
- À concorrência
  - permite que múltiplos processos concorrentes, usem recursos partilhados, sem interferência entre os processos. Por exemplo, um objecto/serviço servidor pode processar vários pedidos em concorrência.

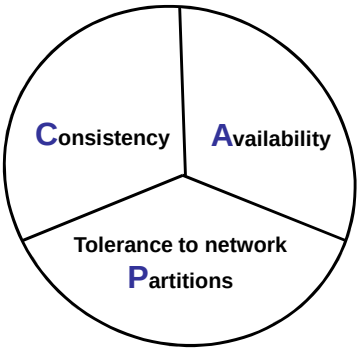
Transparência (cont.)

- **Às réplicas – Uso de múltiplas instâncias (réplicas) de um recurso para melhorar o desempenho, segurança e a tolerância a falhas, sem conhecimento por parte do utilizador ou programador da existência de réplicas;**
- **Às falhas - Disfarce de falhas, permitindo a conclusão de tarefas perante falhas de hardware ou de componentes de software;**
- **À migração - permite a deslocação de recursos ou processos sem afectar a operação dos utilizadores ou programas cliente;**
- **Ao desempenho - permite a reconfiguração de um sistema de modo a acompanhar as variações de carga;**
- **À expansibilidade - permite a extensão do sistema sem alteração da estrutura do sistema nem dos algoritmos das aplicações**

Qualidade de Serviço (QoS)

- **Garantir que os utilizadores têm acesso às funcionalidades de um serviço dentro de limites definidos;**
- **As principais características, não funcionais, envolvidas que afectam a QoS são a fiabilidade (*reliability*), segurança, desempenho e a capacidade de reconfigurar o sistema (*adaptability*).**
- **Nalguns sistemas, por exemplo, distribuição de multimédia o tempo de transmissão (*data-stream)*;**
- **A disponibilidade em tempo apropriado, dos recursos computacionais e de comunicação necessários.**

CAP Theorem



"Num sistema distribuído que partilhe dados só podemos ter 2 das 3 propriedades"

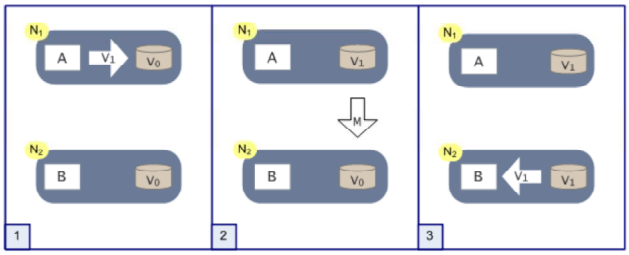
Eric A. Brewer - UC Berkeley, 2000

Consistency **– O sistema oferece um estado consistente para todos os observadores.**

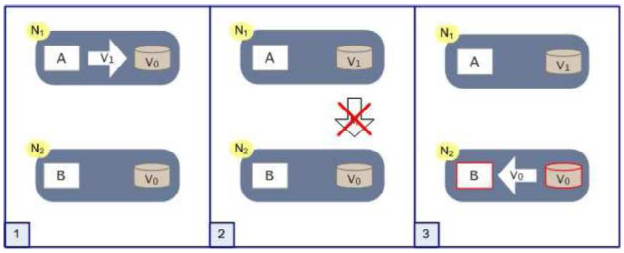
Availability **– O sistema continua a funcionar (eventual degradação na qualidade de serviço) na presença falhas ou falta de conectividade de alguns nós.**

Partitions Tolerance **- O sistema continua a funcionar em partes separadas, quando ocorrem falhas de conectividade na rede.**

Partitions Tolerance



2 partições consistentes



2 partições inconsistentes