

# Sistemas Distribuídos

Carlos Baquero

Departamento de Informática  
Universidade do Minho

MIETI/LCC 2019/2020

# Objectivos

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Obter uma perspectiva abrangente dos sistemas distribuídos e da sua ligação aos sistemas operativos e linguagens de programação.
- Obter conhecimento sobre mecanismos de geração de concorrência e controlo de concorrência.
- Desenvolver a capacidade prática de programação de prototipos de aplicações distribuídas com recurso à linguagem Java e primitivas de sistema com base na tradição Unix.

# Bibliografia

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*Distributed Systems, Concepts and Design*. George Colouris, Jean Dollimore and Tim Kindberg. Fourth Edition. Addison Wesley.  
Informação no site <http://www.cdk4.net/>

# Avaliação

Sistemas  
Distribuídos

©2015 cbm

**Apresentação**

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

A avaliação é composta por um trabalho prático e um exame escrito. Quer o trabalho prático, quer o exame, têm nota mínima de 8 em 20. Os pesos são respectivamente 75% e 25%.

# Observação

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*Notar que, para unidades curriculares com várias componentes de avaliação, em que uma delas seja uma componente prática/experimental sujeita a avaliação, independentemente do seu peso na nota final, os alunos só poderão ir a exame de recurso se obtiverem aproveitamento nessa componente prática/experimental.*

# Contactos

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Página pessoal: <http://haslab.uminho.pt/cbm>
- Email: [cbm@di.uminho.pt](mailto:cbm@di.uminho.pt)

# Sistema Distribuído, Teaser

Sistemas  
Distribuídos

©2015 cbm

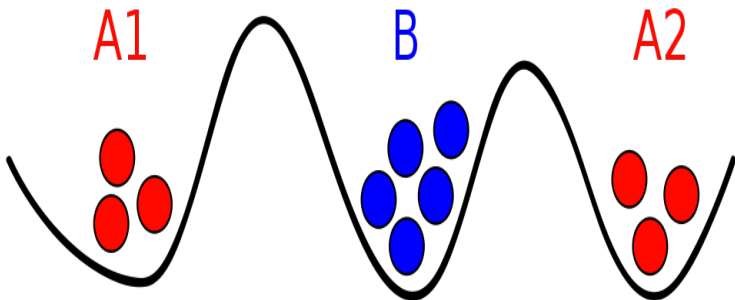
Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições



As duas divisões do exército vermelho são comandadas por dois generais e juntos têm mais tropas que o exército azul. Contudo, só podem comunicar por mensageiros que passam no território inimigo. Como coordenar um ataque para a mesma hora?

[http://en.wikipedia.org/wiki/Two\\_Generals'\\_Problem](http://en.wikipedia.org/wiki/Two_Generals'_Problem)

# Sistema Distribuído, Lamport quote

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*“A distributed systems is the one that prevents you from working because of the failure of a machine that you never heard of.”*



# Sistema Distribuído, C.D.K. quote

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*“We define a distributed system as one in which hardware and software components located at networked computers communicate and coordinate their actions only by passing messages.”*

# Sistema Distribuído, C.D.K. quote

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*“We define a distributed system as one in which hardware and software components located at networked computers communicate and coordinate their actions only by passing messages.”*

Esta definição (e a natureza destes sistemas) leva a um conjunto de consequências ...

# Execuções concorrentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Execução concorrente é a norma.

# Execuções concorrentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Execução concorrente é a norma.
- Os utilizadores vão efectuando as suas tarefas nas suas máquinas.
- Existe partilha de recursos e estes podem ser adicionados.

# Execuções concorrentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Execução concorrente é a norma.
- Os utilizadores vão efectuando as suas tarefas nas suas máquinas.
- Existe partilha de recursos e estes podem ser adicionados.
- A coordenação de execuções concorrentes assume um papel importante.

# Ausência de relógio global

Sistemas  
Distribuídos

©2015 cbm

Apresentação

**Caracterização**

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- A coordenação é feita por troca de mensagens.

# Ausência de relógio global

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- A coordenação é feita por troca de mensagens.
- Existe a necessidade de alguma noção partilhada de tempo, para apoio à coordenação.

# Ausência de relógio global

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- A coordenação é feita por troca de mensagens.
- Existe a necessidade de alguma noção partilhada de tempo, para apoio à coordenação.
- Mas, há limites ao grau de sincronização possível entre máquinas distribuídas.



# Ausência de relógio global

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- A coordenação é feita por troca de mensagens.
- Existe a necessidade de alguma noção partilhada de tempo, para apoio à coordenação.
- Mas, há limites ao grau de sincronização possível entre máquinas distribuídas.
- Não existe um tempo global único. *Nem na Física existe.*

# Falhas independentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Distintas fontes de alimentação, administradores, acessos à rede.

# Falhas independentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Distintas fontes de alimentação, administradores, acessos à rede.
- As máquinas falham, mas aqui falham independentemente.
  - É mau. Podemos não detectar a falha ou o regresso da máquina.
  - É bom. Com redundância podemos manter o serviço.

# Falhas independentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Distintas fontes de alimentação, administradores, acessos à rede.
- As máquinas falham, mas aqui falham independentemente.
  - É mau. Podemos não detectar a falha ou o regresso da máquina.
  - É bom. Com redundância podemos manter o serviço.
- Falhas na rede podem isolar segmentos que continuam a operar. Particionamento.

# Falhas independentes

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Distintas fontes de alimentação, administradores, acessos à rede.
- As máquinas falham, mas aqui falham independentemente.
  - É mau. Podemos não detectar a falha ou o regresso da máquina.
  - É bom. Com redundância podemos manter o serviço.
- Falhas na rede podem isolar segmentos que continuam a operar. Particionamento.
- Como distinguir um rede lenta de uma rede isolada ou simplesmente desligada?

# Exemplos de sistemas distribuídos

## Internet

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Trata-se de um sistema distribuído de grande escala.
- Disponibiliza vários serviços; WWW, email, ftp.
- Conjunto de subredes ligadas por *backbones*.
- Os ISPs dão conectividade aos utilizadores nas fronteiras da rede.
- Várias Intranets mantêm ligações à Internet.

# Exemplos de sistemas distribuídos

## Intranet

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Porção da Internet administrada separadamente e com uma fronteira de segurança com o resto da Internet.
- Os tamanhos variam mas existe sempre de algum modo um controlo administrativo comum à Intranet.
- Uso de *Firewalls* e NAT para mediação do acesso.
- Objectivos
  - Protecção da informação trocada internamente (registos de pacientes em hospitais, segredo comerciais, ...).
  - Relaxamento das regras internas de segurança.
  - Economia de recursos (web caching).
  - No limite podem não ter ligação à Internet, apesar de usar os seus protocolos.

# Exemplos de sistemas distribuídos

Mobilidade, Computação Ubíqua, Redes de Sensores

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Caso extremo de sistema distribuído.
- Ao contactar a rede em vários pontos surgem problemas de identidade.
- Possibilidade de particionamento permanente.
- Motivação para dependência da localização e contexto.
- Com a ubiquidade (com chips em objectos no nosso ambiente) surgem novas direcções e pressões.
- O mesmo se passa com as redes de sensores. **Enormes pressões na escala e pressupostos de comunicação.**



# Web e HTTP

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

HTTP é um dos protocolos suportados pelos browser, sendo do tipo *request/reply*. Envolve vários serviços no cliente e servidor, DNS, File Systems, External Viewers e MIME types, Execução de CGIs, Código Móvel.

# Web e HTTP

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

HTTP é um dos protocolos suportados pelos browser, sendo do tipo *request/reply*. Envolve vários serviços no cliente e servidor, DNS, File Systems, External Viewers e MIME types, Execução de CGIs, Código Móvel.

## Exemplos de URLs

- `http://www.cdk4.net/`
- `http://www.w3.org/Protocols/Activity.html#intro`
- `http://www.google.com/search?q=kindberg`

# Heterogeneidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Surge numa série de dimensões:

- Redes
- Computer Hardware
- Sistemas Operativos
- Linguagens de Programação
- Diferenças de implementação

# Heterogeneidade, Suporte

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Middleware

Mascara alguns aspectos de heterogeneidade, nomeadamente: Rede, Hardware (Little vs Big Endian), OS, Linguagens de Programação.

# Heterogeneidade, Suporte

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Middleware

Mascara alguns aspectos de heterogeneidade, nomeadamente: Rede, Hardware (Little vs Big Endian), OS, Linguagens de Programação.

## Mobile Code

O uso de byte codes e máquinas virtuais permite um aumento da portabilidade de código. Os problemas de segurança associados à distribuição podem ser diminuídos por *sandboxes*.

# Abertura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

**Caracterização**

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Trata-se de uma característica que determina até que ponto um sistema pode estendido e re-implementado. Nesse sentido as interfaces principais têm de ser públicas. A existência de componentes interoperantes com diferentes programadores lança novos desafios à construção de sistemas distribuídos.

Existem processos de estandardização informal, desenvolvidos desde a década de 80 e apoiados pela própria infraestrutura de Internet.

- Request for Comments (RFCs) guardados em [www.ietf.org](http://www.ietf.org).
- CORBA technical documents, em [www.omg.org](http://www.omg.org).
- ...

# Abertura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Características de sistemas abertos:

- Os interfaces principais são públicos.
- Uso de mecanismos de comunicação uniformes e com interfaces documentadas, no acesso a recursos partilhados.
- Capacidade de compor sistemas a partir de hardware heterogénio, potencialmente de diferentes vendedores. Cada componente terá de ser conforme à especificação.



# Abertura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Características de sistemas abertos:

- Os interfaces principais são públicos.
- Uso de mecanismos de comunicação uniformes e com interfaces documentadas, no acesso a recursos partilhados.
- Capacidade de compor sistemas a partir de hardware heterogénio, potencialmente de diferentes vendedores. Cada componente terá de ser conforme à especificação.

## Independência de Fornecedor

Em conjunto estes mecanismos dão aos clientes e integradores um elevado nível de independência de plataformas e vendedores específicos. Exemplo: Skype vs VOIP ?

Pode-se observar três componentes:

- **Confidencialidade.** Evitar o acesso à informação por entidades não autorizadas.

Pode-se observar três componentes:

- Confidencialidade. Evitar o acesso à informação por entidades não autorizadas.
- Integridade. Evitar alterações ou corrupção da informação.

Pode-se observar três componentes:

- **Confidencialidade.** Evitar o acesso à informação por entidades não autorizadas.
- **Integridade.** Evitar alterações ou corrupção da informação.
- **Disponibilidade.** Impedir interferências no acesso a recursos.

Pode-se observar três componentes:

- Confidencialidade. Evitar o acesso à informação por entidades não autorizadas.
- Integridade. Evitar alterações ou corrupção da informação.
- Disponibilidade. Impedir interferências no acesso a recursos.

Algumas propriedades são já controláveis pelo uso de técnicas criptográficas modernas, idealmente com robustez matemática e algoritmos públicos. Contudo a *Security by obscurity* ainda tarda a ser erradicada ...

Existem alguns desafios que ainda aguardam respostas totalmente adequadas:

- *Denial of service attacks*. Onde se tenta esgotar os recursos de um serviço distribuído com o objectivo de impedir os seu. Exemplos DDOS coordenados a sites de referência.

Existem alguns desafios que ainda aguardam respostas totalmente adequadas:

- *Denial of service attacks*. Onde se tenta esgotar os recursos de um serviço distribuído com o objectivo de impedir os seu. Exemplos DDOS coordenados a sites de referência.
- Segurança de código móvel. A execução local de código “desconhecido” levanta problemas de segurança.
  - Como permitir e monitorar o uso de recursos?
  - Como autenticar o código?

# Escalabilidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Um sistema é “Escalável” se se mantiver eficaz perante aumentos significativos no numero de recursos e utilizadores. Podem-se identificar várias dimensões neste problema.



# Escalabilidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Um sistema é “Escalável” se se mantiver eficaz perante aumentos significativos no numero de recursos e utilizadores. Podem-se identificar várias dimensões neste problema.

## Controlo do custo de recursos físicos

É desejável (no mínimo) que a manutenção de um serviço que servia  $x$  utilizadores e agora pretende servir o dobro, possa ser feita duplicando os recursos (espaço em disco, CPU disponível, ...). Tal evidenciaria uma evolução  $O(n)$ , mas contudo nem sempre tal é possível devido a fenomenos complexos de *diminished returns*.

# Escalabilidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Controlo de perdas de desempenho

O sistema inicial DNS passava pela distribuição a cada máquina da lista de nomes e endereços de todas as máquinas. Tal punha dados  $O(n)$  em cada máquina e induzia uma carga  $O(n^2)$  na rede. O sistema actual distribui a carga e sendo hierarquico tem características  $O(\log n)$ . Em geral os sistemas não devem exigir mais do que  $O(\log n)$  para serem escaláveis.

# Escalabilidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Controlo de perdas de desempenho

O sistema inicial DNS passava pela distribuição a cada máquina da lista de nomes e endereços de todas as máquinas. Tal punha dados  $O(n)$  em cada máquina e induzia uma carga  $O(n^2)$  na rede. O sistema actual distribui a carga e sendo hierarquico tem características  $O(\log n)$ . Em geral os sistemas não devem exigir mais do que  $O(\log n)$  para serem escaláveis.

## Evitar o fim de recursos finitos

Actualmente o IPv4 tem graves problemas de falta de endereços. Na verdade qualquer limite fixo é sujeito a problemas no futuro desde que ocorra expansão. Limites sobre-dimensionados têm inconvenientes na carga induzida em permanência.

## Evitar engarrafamentos

Falta de descentralização pode induzir engarrafamentos de performance ou mesmo administrativos. Existem recursos muito acedidos face a outros, como nos sistemas sujeitos a *power laws*, bem como modas ou *flash crowds*. Nalguns casos a replicação e *caching* podem ajudar, mas não é simples ....

# Gestão de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Detecção. Algumas falhas podem ser detectadas (ex: corrupção via checksums), outras não e apenas suspeitadas (ex: falha de servidores em sistemas assíncronos).

# Gestão de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Detecção. Algumas falhas podem ser detectadas (ex: corrupção via checksums), outras não e apenas suspeitadas (ex: falha de servidores em sistemas assíncronos).
- Máscaras. Algumas falhas podem ser escondidas ou tornadas menos severas. Ex: Retransmissão de mensagens, RAID, .... Contudo estas técnicas são tipicamente melhorias probabilísticas.

# Gestão de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- **Detecção.** Algumas falhas podem ser detectadas (ex: corrupção via checksums), outras não e apenas suspeitadas (ex: falha de servidores em sistemas assíncronos).
- **Máscaras.** Algumas falhas podem ser escondidas ou tornadas menos severas. Ex: Retransmissão de mensagens, RAID, .... Contudo estas técnicas são tipicamente melhorias probabilísticas.
- **Tolerância.** Por vezes as próprias falhas podem ser toleradas pelos utilizadores (não confundir com tolerância a faltas). E.g. Repetição em protocolos request/reply.

# Gestão de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- **Detecção.** Algumas falhas podem ser detectadas (ex: corrupção via checksums), outras não e apenas suspeitadas (ex: falha de servidores em sistemas assíncronos).
- **Máscaras.** Algumas falhas podem ser escondidas ou tornadas menos severas. Ex: Retransmissão de mensagens, RAID, .... Contudo estas técnicas são tipicamente melhorias probabilísticas.
- **Tolerância.** Por vezes as próprias falhas podem ser toleradas pelos utilizadores (não confundir com tolerância a faltas). E.g. Repetição em protocolos request/reply.
- **Recuperação.** Os efeitos de algumas falhas podem ser mitigados por recuperação. Ex: Rollbacks.



# Gestão de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- **Detecção.** Algumas falhas podem ser detectadas (ex: corrupção via checksums), outras não e apenas suspeitadas (ex: falha de servidores em sistemas assíncronos).
- **Máscaras.** Algumas falhas podem ser escondidas ou tornadas menos severas. Ex: Retransmissão de mensagens, RAID, .... Contudo estas técnicas são tipicamente melhorias probabilísticas.
- **Tolerância.** Por vezes as próprias falhas podem ser toleradas pelos utilizadores (não confundir com tolerância a faltas). E.g. Repetição em protocolos request/reply.
- **Recuperação.** Os efeitos de algumas falhas podem ser mitigados por recuperação. Ex: Rollbacks.
- **Redundância.** A tolerância a faltas, evitando a exposição de falhas, pode ser procurada com recurso à programação de redundância na arquitectura distribuída.

# Gestão de Falhas

## Alta Disponibilidade

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Os sistemas distribuídos devem exibir alta disponibilidade face a falhas de hardware. A disponibilidade de um sistema é uma medida da proporção de tempo em que um sistema se encontra disponível para uso. Quando ocorrem faltas nos componentes individuais devem ser tomadas medidas compensatórias para minimizar o seu impacto no sistema.

# Concorrência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Sendo a concorrência inerente aos sistemas distribuídos é comum a tentativa de acesso concorrente a recursos partilhados. O controlo de concorrência pode por vezes ser feito junto ao recurso partilhado, usando primitivas clássicas como semáforos, exclusão mútua, etc.

# Concorrência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Sendo a concorrência inerente aos sistemas distribuídos é comum a tentativa de acesso concorrente a recursos partilhados. O controlo de concorrência pode por vezes ser feito junto ao recurso partilhado, usando primitivas clássicas como semáforos, exclusão mútua, etc. Exemplo: Consulta de saldo concorrente com transferência.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.



# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.
- Na Replicação. Evitando expôr a presença de réplicas.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.
- Na Replicação. Evitando expôr a presença de réplicas.
- Nas Falhas. Escondendo-as.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.
- Na Replicação. Evitando expôr a presença de réplicas.
- Nas Falhas. Escondendo-as.
- Na Mobilidade. Permitindo-a sem que afecte o sistema.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.
- Na Replicação. Evitando expôr a presença de réplicas.
- Nas Falhas. Escondendo-as.
- Na Mobilidade. Permitindo-a sem que afecte o sistema.
- Performance. Permitindo reconfigurações para a sua recuperação.

# Transparência

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Pretende-se mostrar o sistema distribuído como um todo e não como um aglomerado de componentes individuais. Podem identificar-se vários tipos de transparência:

- No Acesso. Idêntico acesso a recursos remotos e locais.
- Na Localização. Um endereço DNS é dissociado do local.
- Concorrência. Criando a ilusão de acesso exclusivo a recursos.
- Na Replicação. Evitando expôr a presença de réplicas.
- Nas Falhas. Escondendo-as.
- Na Mobilidade. Permitindo-a sem que afecte o sistema.
- Performance. Permitindo reconfigurações para a sua recuperação.
- Na Escalabilidade. Permitindo-a sem mudanças aos algoritmos e estrutura.

# Arquitectura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

A arquitectura de um sistema distribuído prende-se com a forma como os seus componentes se encontram posicionados e organizados a nível funcional.

Pode-se observar uma divisão principal em sistemas **cliente-servidor** e abordagens **par-a-par**.

# Arquitectura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

A arquitectura de um sistema distribuído prende-se com a forma como os seus componentes se encontram posicionados e organizados a nível funcional.

Pode-se observar uma divisão principal em sistemas **cliente-servidor** e abordagens **par-a-par**.

O modelo client-servidor básico pode sofrer adaptações, nomeadamente:

- Pelo recurso a código móvel e/ou objectos móveis.

# Arquitectura

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

A arquitectura de um sistema distribuído prende-se com a forma como os seus componentes se encontram posicionados e organizados a nível funcional.

Pode-se observar uma divisão principal em sistemas **cliente-servidor** e abordagens **par-a-par**.

O modelo client-servidor básico pode sofrer adaptações, nomeadamente:

- Pelo recurso a código móvel e/ou objectos móveis.
- Pelo suporte de números variáveis de nodos e à descoberta de serviços.



# Middleware

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Disponibiliza abstracções para programação de aplicações distribuídas. Em particular existe um uso alargado de **object oriented middleware**:

- CORBA.
- Java RMI.
- web services.
- MS DCOM.

A colocação de muita funcionalidade no middleware é limitada pelo **end-to-end argument**.

# The end-to-end argument [Saltzer e al. 1984]

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

*Some communication-related functions can be completely and reliably implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that function as a feature of the communication systems itself is not always sensible.*

## Semântica junto à aplicação

Ex: Na transmissão de coordenadas num jogo faz sentido esquecer coordenadas antigas em favor de passar mais rapidamente as novas.

# Arquitetura cliente-servidor

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Trata-se de arquitetura historicamente mais importante e ainda a mais comum. Exemplos: DNS e WWW.

# Arquitetura cliente-servidor

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Trata-se de arquitetura historicamente mais importante e ainda a mais comum. Exemplos: DNS e WWW.
- É comum organizar-se hierarquicamente e/ou haver nodos que acumulam as funções de cliente e servidor. Exemplos: DNS e Crawlers.

# Arquitetura par-a-par

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Elimina-se a distinção cliente-servidor, cada nó é sempre cliente e servidor.

# Arquitetura par-a-par

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Elimina-se a distinção cliente-servidor, cada nó é sempre cliente e servidor.
- Arquitetura semi ou totalmente descentralizada.

# Arquitetura par-a-par

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Elimina-se a distinção cliente-servidor, cada nó é sempre cliente e servidor.
- Arquitetura semi ou totalmente descentralizada.
- Tira partido das capacidade na periferia da rede.

# Arquitetura par-a-par

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Elimina-se a distinção cliente-servidor, cada nó é sempre cliente e servidor.
- Arquitetura semi ou totalmente descentralizada.
- Tira partido das capacidade na periferia da rede.
- Abordagem economicamente leve a sistemas distribuídos de larga escalada.



# Arquitetura par-a-par

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Elimina-se a distinção cliente-servidor, cada nó é sempre cliente e servidor.
- Arquitetura semi ou totalmente descentralizada.
- Tira partido das capacidade na periferia da rede.
- Abordagem economicamente leve a sistemas distribuídos de larga escalada.
- Problemas na manutenção de qualidade de serviço.

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Multiplos Servidores

Partição do estado (ou mesmo replicação total) entre múltiplos servidores. Vários serviços pesados (Google, Ebay, ...) são com base em backends com clusters de máquinas.

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Multiplos Servidores

Partição do estado (ou mesmo replicação total) entre múltiplos servidores. Vários serviços pesados (Google, Ebay, ...) são com base em backends com clusters de máquinas.

## Proxys e Caches

As *caches* permitem guardar copias de objectos remotos em locais mais proximos do clientes. Podem ficar desactualizadas mas isso é facilmente detectável mesmo sem sincrononia de relógios. Permitem ganhos por localidade de referencias entre um grupo de utilizadores e atenuam *flash crowds*.

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Código móvel

Permite colmatar funcionalidades ausentes nos browsers ou reduzir as necessidades de comunicação. Por exemplo, pode implementar funcionalidade com base em *push* por parte do servidor (cotações de bolsa e triggers para acção de bolsa no cliente).

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Código móvel

Permite colmatar funcionalidades ausentes nos browsers ou reduzir as necessidades de comunicação. Por exemplo, pode implementar funcionalidade com base em *push* por parte do servidor (cotações de bolsa e triggers para acção de bolsa no cliente).

## Agentes móveis

Conjunto de código e dados móveis que permite situar a computação no local mais favorável no sistema distribuído. Por exemplo executando queries junto às bases de dados de vários servidores. Tal como o código móvel, levantam problemas óbvios de segurança.

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Network Computers

Simplificam a manutenção de máquinas clientes. Não mantêm estado local mas caches de dados e aplicações. Apesar de executarem operações localmente, dependem de servidores de ficheiros para carregamento dos sistema operativo, aplicações e dados.

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Network Computers

Simplificam a manutenção de máquinas clientes. Não mantêm estado local mas caches de dados e aplicações. Apesar de executarem operações localmente, dependem de servidores de ficheiros para carregamento dos sistema operativo, aplicações e dados.

## Thin Clients

Apenas executam localmente a apresentação no ecrã e interação com periféricos de interface com utilizador. Podem correr um processo servidor X11 (apesar de serem nodos clientes) e são suportados por uma máquina onde correm os processos e residem os dados. Tem alguma aplicabilidade em sistemas móveis

# Variantes ao modelo CS

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Network Computers

Simplificam a manutenção de máquinas clientes. Não mantêm estado local mas caches de dados e aplicações. Apesar de executarem operações localmente, dependem de servidores de ficheiros para carregamento dos sistema operativo, aplicações e dados.

## Thin Clients

Apenas executam localmente a apresentação no ecrã e interação com periféricos de interface com utilizador. Podem correr um processo servidor X11 (apesar de serem nodos clientes) e são suportados por uma máquina onde correm os processos e residem os dados. Tem alguma aplicabilidade em sistemas móveis

## Sistemas Móveis

Para além de exacerbarem algumas dificuldades no modelo cliente-servidor, permitem interações ad-hoc que se aproximam do modelo par-a-par. Ilhas de conectividade.



# Modelo de interacção

Canais de comunicação

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Latência

Atraso induzido pelo canal. os seus fatores incluem:

- Atraso no acesso aos canais. Nomeadamente espera de silêncio numa rede ethernet.
- Atraso na propagação. Longas distancias, ligações satélite.
- Atrasos computacionais nos routers e sistemas operativo das máquinas em cada extremo.

# Modelo de interacção

Canais de comunicação

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Latência

Atraso induzido pelo canal. os seus fatores incluem:

- Atraso no acesso aos canais. Nomeadamente espera de silêncio numa rede ethernet.
- Atraso na propagação. Longas distancias, ligações satélite.
- Atrasos computacionais nos routers e sistemas operativo das máquinas em cada extremo.

## Largura de Banda

Quantidade de informação trocada ao longo de um dado tempo. A partilha de canais limita a largura de banda individual.

# Modelo de interacção

Canais de comunicação

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Latência

Atraso induzido pelo canal. os seus fcatores incluem:

- Atraso no acesso aos canais. Nomeadamente espera de silêncio numa rede ethernet.
- Atraso na propagação. Longas distancias, ligações satélite.
- Atrasos computacionais nos routers e sistemas operativo das máquinas em cada estremo.

## Largura de Banda

Quantidade de informação trocada ao longo de um dado tempo. A partilha de canais limita a largura de banda individual.

## Jitter

Variação na cadência de chegada de mensagens. Particularmente importante em informação multimédia (vídeo e audio).

# Modelos Síncrono vs Assíncrono

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Síncrono

- Limite inferior e superior para cada passo de computação.
- Limites para a recepção de mensagens.
- Deriva de relógios dentro de limites conhecidos.

# Modelos Síncrono vs Assíncrono

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Síncrono

- Limite inferior e superior para cada passo de computação.
- Limites para a recepção de mensagens.
- Deriva de relógios dentro de limites conhecidos.

## Assíncrono

- Tempos arbitrários para cada passo de computação.
- Atrasos arbitrários na recepção de mensagens.
- Deriva de relógios ilimitada.

# Modelos Síncrono vs Assíncrono

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Síncrono

- Limite inferior e superior para cada passo de computação.
- Limites para a recepção de mensagens.
- Deriva de relógios dentro de limites conhecidos.

## Assíncrono

- Tempos arbitrários para cada passo de computação.
- Atrasos arbitrários na recepção de mensagens.
- Deriva de relógios ilimitada.

## Partilha

Sendo possível construir sistemas sincronos a partilha de recursos conduz rapidamente a modelos assincronos (mais económicos).

# Ordenação de mensagens

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Envios

Troca de mensagens entre utilizadores X,Y,Z e A numa lista:

- X manda mensagem com tema *Meeting*.
- Y e Z respondem ambos com tema *Re:Meeting*.

# Ordenação de mensagens

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Envios

Troca de mensagens entre utilizadores X,Y,Z e A numa lista:

- X manda mensagem com tema *Meeting*.
- Y e Z respondem ambos com tema *Re:Meeting*.

Devido a atrasos independentes nos percursos das mensagens estas podem ser recebidas por A numa ordem que não retrata a ordem do tempo real.



# Ordenação de mensagens

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

## Envios

Troca de mensagens entre utilizadores X,Y,Z e A numa lista:

- X manda mensagem com tema *Meeting*.
- Y e Z respondem ambos com tema *Re:Meeting*.

Devido a atrasos independentes nos percursos das mensagens estas podem ser recebidas por A numa ordem que não retrata a ordem do tempo real.

## Recebidas em A

- 1 De Z, *Re:Meeting*.
- 2 De X, *Meeting*.
- 3 De Y, *Re:Meeting*.

# Ordenação de mensagens

Sistemas  
Distribuídos

©2015 cbm

Apresentação

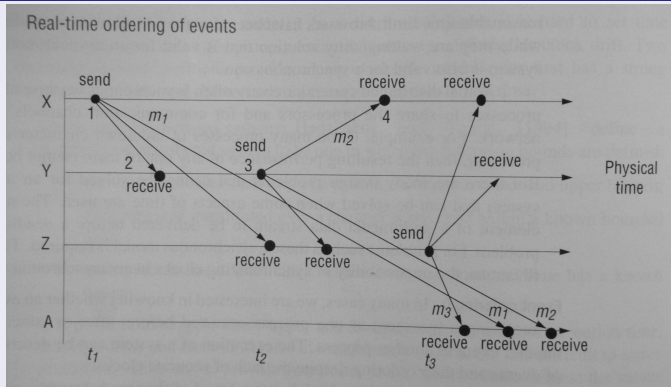
Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Figura 2.8



# Ordenação de mensagens

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Sendo difícil uma sincronização fina de relógios de tempo real num sistema assíncrono é contudo possível ter outras noções de tempo que relatam a ordem relativa de eventos no sistema. Tais conceitos, como **Lamport Clocks** e **Vector Clocks** serão aprofundados numa fase posterior.

# Modelos de Falhas

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

O modo como processos e canais falham permite construir uma classificação de falhas. A taxonomia de falhas de Hadzilacos e Toueg faz uma separação entre **falhas com omissão**, **falhas arbitrárias** e falhas relacionadas com o tempo (**timing failures**). Vamos aprofundar aqui as duas primeiras classes.

# Modelos de Falhas

## Falhas com Omissão

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Ocorrem quando os processos ou canais deixam de fazer acções que lhes compete fazer.

# Modelos de Falhas

## Falhas com Omissão

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Ocorrem quando os processos ou canais deixam de fazer acções que lhes compete fazer.

### Omissão em Processos

Um processo entra em **crash** quando falha deixando de executar para sempre passos do seu programa. É comum considerar-se que um processo ou executa correctamente ou entra em crash, parando. Outros processo podem tentar detectar um crash verificando de o processo deixa de responder a mensagens (**timeouts**). Num sistema assíncrono nunca há certezas ...

# Modelos de Falhas

## Falhas com Omissão

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Ocorrem quando os processos ou canais deixam de fazer acções que lhes compete fazer.

### Omissão em Processos

Um processo entra em **crash** quando falha deixando de executar para sempre passos do seu programa. É comum considerar-se que um processo ou executa correctamente ou entra em crash, parando. Outros processo podem tentar detectar um crash verificando de o processo deixa de responder a mensagens (**timeouts**). Num sistema assíncrono nunca há certezas ...

### Fail-stop

Num sistema síncrono é possível detectar com certeza se um processo falha, estipulando um timeout e desde que haja garantia de entrega nas mensagens. Neste caso temos um modelo **fail-stop**.

### Omissão na Comunicação

Vendo uma comunicação com base em primitivas **send** e **receive** podemos observar a decomposição em:

- Colocar a mensagem num **out buffer**
- O transporte do **out buffer** para um **in buffer** pelo canal de comunicação.
- Leitura do **in buffer** e entrega (**delivery**) ao processo destino.



### Omissão na Comunicação

Vendo uma comunicação com base em primitivas **send** e **receive** podemos observar a decomposição em:

- Colocar a mensagem num **out buffer**
- O transporte do **out buffer** para um **in buffer** pelo canal de comunicação.
- Leitura do **in buffer** e entrega (**delivery**) ao processo destino.

Esta decomposição leva, respectivamente, aos seguintes tipos de falhas:

### Omissão na Comunicação

Vendo uma comunicação com base em primitivas **send** e **receive** podemos observar a decomposição em:

- Colocar a mensagem num **out buffer**
- O transporte do **out buffer** para um **in buffer** pelo canal de comunicação.
- Leitura do **in buffer** e entrega (**delivery**) ao processo destino.

Esta decomposição leva, respectivamente, aos seguintes tipos de falhas:

- Omissão no envio.
- Omissão no canal.
- Omissão na recepção.

### Omissão na Comunicação

Vendo uma comunicação com base em primitivas **send** e **receive** podemos observar a decomposição em:

- Colocar a mensagem num **out buffer**
- O transporte do **out buffer** para um **in buffer** pelo canal de comunicação.
- Leitura do **in buffer** e entrega (**delivery**) ao processo destino.

Esta decomposição leva, respectivamente, aos seguintes tipos de falhas:

- Omissão no envio.
- Omissão no canal.
- Omissão na recepção.

As falhas até aqui descritas são consideradas **falhas benignas**

# Modelos de Falhas

## Falhas Arbitrárias

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Num modelo de falhas arbitrárias ou **bizantinas** encontra-se a pior semântica de falhas possível. Aqui são possíveis todos os tipos de comportamentos, incluindo respostas erradas nos processos, repetição e/ou criação de mensagens, ...

# Modelos de Falhas

## Falhas Arbitrárias

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Num modelo de falhas arbitrárias ou **bizantinas** encontra-se a pior semântica de falhas possível. Aqui são possíveis todos os tipos de comportamentos, incluindo respostas erradas nos processos, repetição e/ou criação de mensagens, ...

### Processos Bizantinos

- Podem ser omitidas passos arbitrários.
- Podem ser criados e executados passos arbitrários.

# Modelos de Falhas

## Falhas Arbitrárias

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Num modelo de falhas arbitrárias ou **bizantinas** encontra-se a pior semântica de falhas possível. Aqui são possíveis todos os tipos de comportamentos, incluindo respostas erradas nos processos, repetição e/ou criação de mensagens, ...

### Processos Bizantinos

- Podem ser omitidas passos arbitrários.
- Podem ser criados e executados passos arbitrários.

### Canais Bizantinos

Alguns exemplos do que pode ocorrer:

- Corrupção de mensagens.
- Criação de novas mensagens.
- Repetição de mensagens.

# Comunicação fiável ponto a ponto

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Mesmo na presença de falhas com omissão num canal de comunicação é possível um serviço que permita fiabilidade na comunicação sendo esta expressa em termos de **validade** e **integridade**.

**Validade** Qualquer mensagem no buffer de saída é inevitavelmente (**eventually**) entregue no buffer de entrada do destino.

**Integridade** A mensagem recebida é idêntica à enviada e nenhuma mensagem é entregue duas vezes.

# Comunicação fiável ponto a ponto

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Mesmo na presença de falhas com omissão num canal de comunicação é possível um serviço que permita fiabilidade na comunicação sendo esta expressa em termos de **validade** e **integridade**.

**Validade** Qualquer mensagem no buffer de saída é inevitavelmente (**eventually**) entregue no buffer de entrada do destino.

**Integridade** A mensagem recebida é idêntica à enviada e nenhuma mensagem é entregue duas vezes.

A retransmissão, checksum e atribuição de números de sequência às mensagens individuais ajuda alcançar estas propriedades.



# Exclusão Mútua Distribuída

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Num ambiente mono-processador a obtenção de exclusão mútua no acesso a recursos partilhados é obtida à custa de instruções de teste e alteração indivisível, sendo posteriormente abstraídas como variáveis mutex, monitores, etc.

# Exclusão Mútua Distribuída

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Num ambiente mono-processador a obtenção de exclusão mútua no acesso a recursos partilhados é obtida à custa de instruções de teste e alteração indivisível, sendo posteriormente abstraídas como variáveis mutex, monitores, etc.
- Num ambiente distribuído a exclusão mútua não é menos necessária e passa a ser obtida por algoritmos de coordenação distribuída. Por exemplo num sistema de partilha de sistema de ficheiros a obtenção de *locks* sobre ficheiros é tipicamente feita à custa de um mecanismo externo de exclusão mútua.

# Exclusão Mútua Distribuída

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Num ambiente mono-processador a obtenção de exclusão mútua no acesso a recursos partilhados é obtida à custa de instruções de teste e alteração indivisível, sendo posteriormente abstraídas como variáveis mutex, monitores, etc.
- Num ambiente distribuído a exclusão mútua não é menos necessária e passa a ser obtida por algoritmos de coordenação distribuída. Por exemplo num sistema de partilha de sistema de ficheiros a obtenção de *locks* sobre ficheiros é tipicamente feita à custa de um mecanismo externo de exclusão mútua.

## Exemplo: Parque de Estacionamento

Num parque de estacionamento com N entradas/saídas e um sistema embutido com N nodos e ligação WIFI em cada portão, o acesso a um contador partilhado de número de veículos pode ser feito à custa de exclusão mútua.

# Exclusão Mútua Distribuída

## Contexto

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Sistema assíncrono em que os processo não falham e a entrega de mensagens é fiável (sendo entregues intactas e uma única vez).

# Exclusão Mútua Distribuída

## Contexto

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Sistema assíncrono em que os processo não falham e a entrega de mensagens é fiável (sendo entregues intactas e uma única vez).

### Nível da Aplicação

**Enter** Acesso à região crítica com eventual espera.

... Acesso aos recursos protegidos.

**Exit** Indicação de saída de região crítica.

# Exclusão Mútua Distribuída

## Requisitos

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Os algoritmos de exclusão mútua devem cumprir alguns ou todos dos seguintes requisitos:

**Safety** Garantia de que no máximo um processo pode executar no interior da região crítica.

**Liveness** Os pedidos de entrada e indicações de saída da região crítica acabam inevitavelmente por ser bem sucedidos (as esperas não são intermináveis).

**Ordem Causal** Se um pedido de entrada anteceder outro na relação de causalidade de Lamport este deve entrar na região antes do outro.

Estas garantias de *Liveness* e *Ordem Causal* estão relacionadas com o conceito de *Fairness*.

# Exclusão Mútua Distribuída

## Algoritmo Centralizado

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

O acesso à região é obtido fazendo um pedido a um servidor central e esperando pela resposta. A saída é igualmente notificada ao servidor. O servidor mantém uma fila de pedidos pendentes de entrada por ordem de recepção de pedido.

# Exclusão Mútua Distribuída

## Algoritmo Centralizado

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

O acesso à região é obtido fazendo um pedido a um servidor central e esperando pela resposta. A saída é igualmente notificada ao servidor. O servidor mantém uma fila de pedidos pendentes de entrada por ordem de recepção de pedido.

### Propriedades

Na ausência de falhas as duas primeiras propriedades são respeitadas. Quanto à terceira propriedade, **Ordem Causal**, qual o comportamento ?



# Exclusão Mútua Distribuída

## Algoritmo Centralizado

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

O acesso à região é obtido fazendo um pedido a um servidor central e esperando pela resposta. A saída é igualmente notificada ao servidor. O servidor mantém uma fila de pedidos pendentes de entrada por ordem de recepção de pedido.

### Propriedades

Na ausência de falhas as duas primeiras propriedades são respeitadas. Quanto à terceira propriedade, **Ordem Causal**, qual o comportamento ?

É possível imaginar execuções em que esta propriedade é desrespeitada. Devemos também analisar o impacto do algoritmo em termos de mensagens trocadas.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.
- É mantido um testemunho no sistema.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.
- É mantido um testemunho no sistema.
- Um processo que receba um testemunho e não o necessite passa-o ao seu sucessor.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.
- É mantido um testemunho no sistema.
- Um processo que receba um testemunho e não o necessite passa-o ao seu sucessor.
- Um processo que queira aceder à região crítica deve aguardar pelo testemunho e retê-lo uma vez obtido.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.
- É mantido um testemunho no sistema.
- Um processo que receba um testemunho e não o necessite passa-o ao seu sucessor.
- Um processo que queira aceder à região crítica deve aguardar pelo testemunho e retê-lo uma vez obtido.
- À saída da região crítica o testemunho é passado ao seu sucessor.

# Exclusão Mútua Distribuída

## Algoritmo em Anel

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

- Os  $N$  processos são agrupados segundo um anel estando cada processo ligado ao seu sucessor e antecessor lógico.
- É mantido um testemunho no sistema.
- Um processo que receba um testemunho e não o necessite passa-o ao seu sucessor.
- Um processo que queira aceder à região crítica deve aguardar pelo testemunho e retê-lo uma vez obtido.
- À saída da região crítica o testemunho é passado ao seu sucessor.

### Propriedades

Sendo respeitadas as duas primeiras propriedades nada força a que os padrões de causalidade na comunicação entre os processos estejam de acordo com as passagens de testemunho. A comunicação envolve em média  $O(N)$  mensagens.

# Eleições

## Motivação e Propriedades

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Quando se torna necessário designar um dado processo para desempenhar um papel específico é comum recorrer-se a um mecanismo de eleição. Por exemplo para se recuperar um token perdido.



# Eleições

## Motivação e Propriedades

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Quando se torna necessário designar um dado processo para desempenhar um papel específico é comum recorrer-se a um mecanismo de eleição. Por exemplo para se recuperar um token perdido.

Uma propriedade importante é que um unico processo seja eleito, mesmo que vários iniciem o mecanismo de eleição. Mais formalmente:

# Eleições

## Motivação e Propriedades

Sistemas  
Distribuídos

©2015 cbm

Apresentação

Caracterização

Modelos de  
Sistemas

Coordenação e  
Acordo

Eleições

Quando se torna necessário designar um dado processo para desempenhar um papel específico é comum recorrer-se a um mecanismo de eleição. Por exemplo para se recuperar um token perdido.

Uma propriedade importante é que um unico processo seja eleito, mesmo que vários iniciem o mecanismo de eleição. Mais formalmente:

Cada processo  $p_i (i = 1, 2, \dots, N)$  possui uma variável *elected<sub>i</sub>* que virá a conter o ID do processo eleito. Todos os processos possuem IDs únicos e comparáveis num ordem total.

Quando se torna necessário designar um dado processo para desempenhar um papel específico é comum recorrer-se a um mecanismo de eleição. Por exemplo para se recuperar um token perdido.

Uma propriedade importante é que um unico processo seja eleito, mesmo que vários iniciem o mecanismo de eleição. Mais formalmente:

Cada processo  $p_i (i = 1, 2, \dots, N)$  possui uma variável *elected* <sub>$i$</sub>  que virá a conter o ID do processo eleito. Todos os processos possuem IDs únicos e comparáveis num ordem total.

- Safety. All  $p_i$  has *elected* = *nil* ou *elected* =  $P$  em que  $P$  é o ID mais elevado de entre os processo activos (non-crashed).
- Liveness. Todos os  $p_i$  participam e inevitavelmente chegam a um *elected*  $\neq$  *nil* ou terminam em crash.

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com *acted = no* e *elected = nil*.

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com *acted* = *no* e *elected* = *nil*.
- Qualquer  $p_i$  pode arrancar fazendo *acted* := *yes* e mandando *msg*( $ID_i$ ).

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com *acted* = *no* e *elected* = *nil*.
- Qualquer  $p_i$  pode arrancar fazendo *acted* := *yes* e mandando *msg*( $ID_i$ ).
- Ao receber *msg*( $ID_j$ ). Se  $ID_j > ID_i$  manda *msg*( $ID_j$ ). Se  $ID_j < ID_i$ : Se *acted* = *no* manda *msg*( $ID_i$ ) senão nada. Sempre que se manda *msg*(...) faz-se *acted* := *yes*.

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com *acted* = *no* e *elected* = *nil*.
- Qualquer  $p_i$  pode arrancar fazendo *acted* := *yes* e mandando *msg*( $ID_i$ ).
- Ao receber *msg*( $ID_j$ ). Se  $ID_j > ID_i$  manda *msg*( $ID_j$ ). Se  $ID_j < ID_i$ : Se *acted* = *no* manda *msg*( $ID_i$ ) senão nada. Sempre que se manda *msg*(...) faz-se *acted* := *yes*.
- Ao receber *msg*( $ID_i$ ) faz-se *elected* <sub>$i$</sub>  :=  $ID_i$  e manda *msg*(*elected*,  $ID_i$ ), fazendo *acted* := *no*.



Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com  $acted = no$  e  $elected = nil$ .
- Qualquer  $p_i$  pode arrancar fazendo  $acted := yes$  e mandando  $msg(ID_i)$ .
- Ao receber  $msg(ID_j)$ . Se  $ID_j > ID_i$  manda  $msg(ID_j)$ . Se  $ID_j < ID_i$ : Se  $acted = no$  manda  $msg(ID_i)$  senão nada. Sempre que se manda  $msg(...)$  faz-se  $acted := yes$ .
- Ao receber  $msg(ID_i)$  faz-se  $elected_i := ID_i$  e manda  $msg(elected, ID_i)$ , fazendo  $acted := no$ .
- Ao receber  $msg(elected, ID_j)$  faz-se  $elected_i := ID_j$  e  $acted := no$ .

Processos organizados em anel, como um canal para o vizinho horário e IDs únicos. Comunicação pode ser assíncrona, mas sem falhas.

- Todos com  $acted = no$  e  $elected = nil$ .
- Qualquer  $p_i$  pode arrancar fazendo  $acted := yes$  e mandando  $msg(ID_i)$ .
- Ao receber  $msg(ID_j)$ . Se  $ID_j > ID_i$  manda  $msg(ID_j)$ . Se  $ID_j < ID_i$ : Se  $acted = no$  manda  $msg(ID_i)$  senão nada. Sempre que se manda  $msg(\dots)$  faz-se  $acted := yes$ .
- Ao receber  $msg(ID_i)$  faz-se  $elected_i := ID_i$  e manda  $msg(elected, ID_i)$ , fazendo  $acted := no$ .
- Ao receber  $msg(elected, ID_j)$  faz-se  $elected_i := ID_j$  e  $acted := no$ .