Sumário

- Níveis de abstracção.
- Sistemas síncronos e assíncronos.
- Partições.
- Comunicação e filiação em grupo.
- Primitivas de comunicação e a sua concretização.

Heterogeneidade de redes e protocolos

- Muitos tipos de redes e protocolos.

  Como conceber aplicações distribuídas?
- Soluções específicas para um dado tipo de rede.

  Desempenho optimizado mas pouca portabilidade.
- Soluções genéricas.

  Possível limitação no desempenho.

  Que tipo de primitivas utilizar.

Primitivas fracas e fortes

- Primitivas bastante fracas:

  Exemplo: comunicação ponto-a-ponto não fiável.

  Bom desempenho mas grande complexidade na concepção da aplicação.
- Primitivas fortes:

  Difusão fiável e totalmente ordenada.

  Menor desempenho mas alivia a complexidade da aplicação.
- Pergunta chave:

  Esta primitiva é indispensável do ponto de vista lógico?

A primitiva mais fraca

- Comunicação ponto-a-ponto não fiável.
- A aplicação tem de tratar explicitamente da:

  Perca de mensagens.

  Duplicação de mensagens.

  Geralmente também não oferece fragmentação.

Uma hipótese mais útil e realista

- Ligações fiáveis e ordenadas ponto-a-ponto.

  Geralmente concretizadas por um protocolo de transporte.

  Bastante genérico e fácilmente concretizável.

  Custo de estabelecer uma ligação.

- Manutenção da ligação.

  Quando quebrar a ligação?

- Leva-nos a dois aspectos fundamentais:

  Sistemas síncronos e assíncronos e partições.

Sistemas síncronos e assíncronos.

- Sistemas síncronos:

  Caracterizados por atrasos limitados no processamento, na troca das mensagens e na velocidade dos relógios.

- Sistemas assíncronos:

  Em que não é possível definir limites aos atrasos na troca de mensagens ou no processamento (mais precisamente, sempre que estes limites, mesmo quando existem, são demasiado elevados para poderem ser utilizados na prática).

Sincronia: impacto nos protocolos

- Num sistema assíncrono é impossível distinguir uma falha de um atraso.
- Como resultado, é impossível GARANTIR algumas coisas úteis, como por exemplo:

  Que um conjunto de processos chega a acordo acerca de um determinado valor.

  Um conjunto de resultados teóricos conhecidos por resultados de impossibilidade.

Sincronia e noção de tempo

- De um modo geral, num sistema assíncrono também não é possível sincronizar relógios.

  Isto pode ser superada recorrendo ao sistema GPS.

- Tipicamente, recorre-se a relógios lógicos.
- Ou relógios sincronizados com muito pouca precisão.

Partições

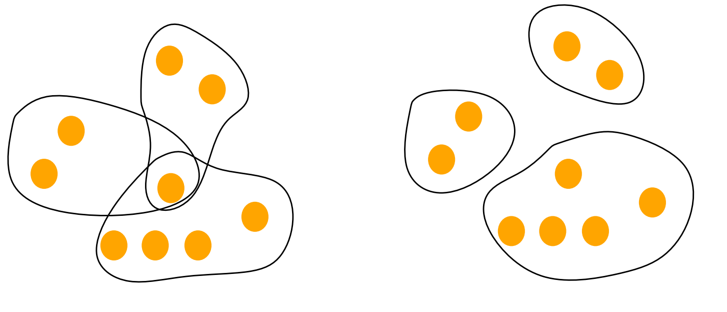
- Partição na rede.

  Os processos numa partição podem comunicar entre si mas não conseguem comunicar com nenhum nó noutra partição.

- Impossível garantir a coerência entre processos em partições distintas sem bloquear uma (ou todas as partições).

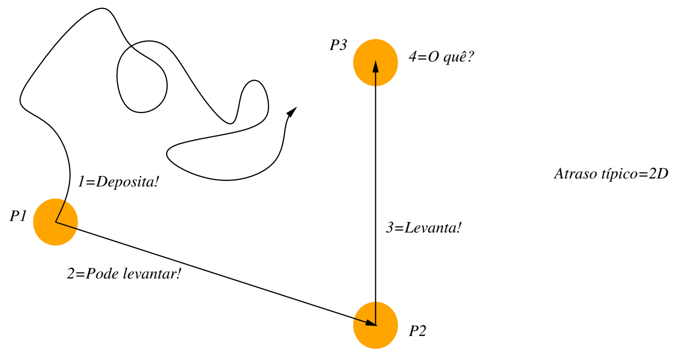
  Tenta-se escolher uma única partição onde se mantém a actividade.

Partições: ilustração

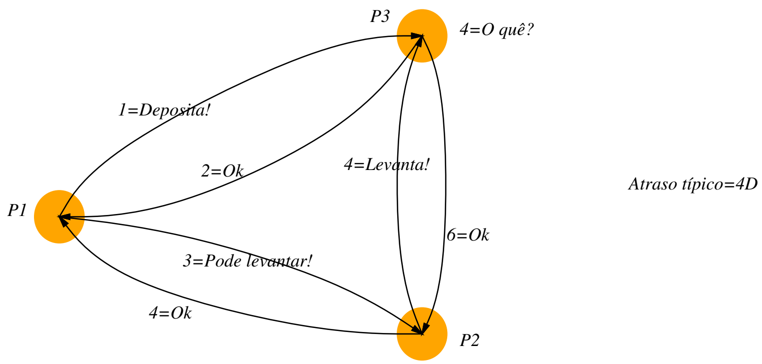


Sistema não particionado *Sistema particionado*

Ordenação para além de FIFO: causal



Solução: aumentar o sincronismo



Outra solução

- Utilizar uma primitiva de comunicação causal.

  Entrega as mensagens respeitando a ordem causal estabelecida por eventos de envio/recepção.

  - m_1 *e* m_2 *foram emitidas pelo mesmo processo e* m_2 *foi enviada após* m_1 *ou;*
  - m_1 *foi entregue ao emissor de* m_2 *antes de* m_2 *ter sido emitida ou;*
  - *existe* m_3 *tal que* $m_1 \rightarrow m_3$ *e* $m_3 \rightarrow m_2$.

Concretização de ordem causal

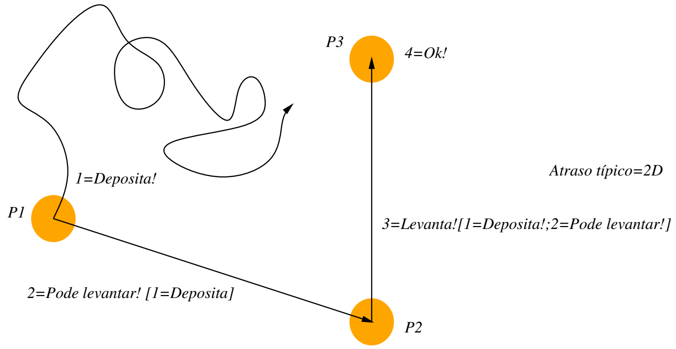
- Também tem um custo.
- É necessário manter informação acerca do passado.

  No pior caso, n^2 *elementos.*

- É necessário recolher confirmações:

  Para evitar a ocorrência de "buracos" no passado.

Concretização ordem causal (solução 1)



Problemas com a solução anterior

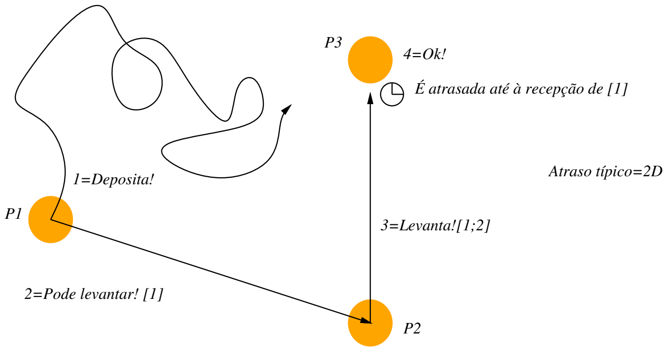
- As mensagens ficam muito grandes.

  É necessário eliminar as mensagens que já não é necessário transportar.

- Outra solução:

  Enviar só um identificador das mensagens.

Enviar só o identificador



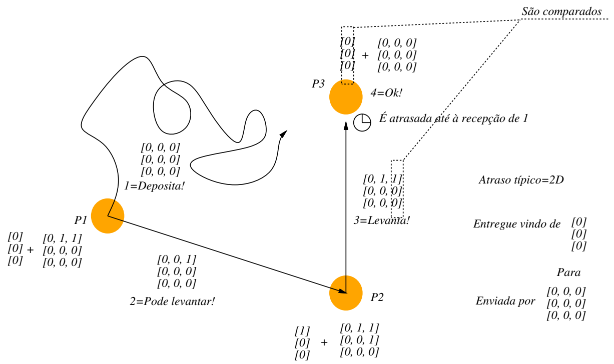
Problemas com a solução anterior

- Continua a ser necessário eliminar identificadores já obsoletos.
- Soluções:

  Relógio matricial.

  Usar informação acerca da topologia da rede.

Relógio matricial



Limitações da comunicação ponto-a-ponto

- Por vezes é necessário enviar uma mensagem para vários nós.

  Fazê-lo ao nível da aplicação não é uma solução modular.

- Por vezes é necessário que diversos processos estejam de acordo acerca da falha de um outro processo.

  É necessário executar um protocolo de filiação.

Filiação e comunicação em grupo

- Serviço de filiação em grupo:

  Permite aos participantes organizarem-se em grupos e receber informação actualizada de quais os elementos acessíveis (esta informação designa-se por vista de grupo*).*

- Serviço de comunicação em grupo:

  Permite aos elementos do grupo trocarem informação, sendo-lhes oferecidas garantias de entrega e de ordenação que variam de serviço para serviço.

Propriedades de um serviço de filiação

- Exactidão:

  A informação entregue reflecte a actual situação física.

  Só pode ser aproximada num sistema assíncrono.

- Coerência:

  A informação é idêntica em todos os processos na mesma vista.

Detecção de falhas

- Detector perfeito:

  O próprio processo que falha, avisa os outros que falhou no momento da sua recuperação.

  Nem sempre muito útil, pois impede o progresso da aplicação.

- Detector mais prático:

  Baseado em testes de actividade e temporizadores.

  Por vezes, fornece indicações inexactas.

- Podem ser classificados de acordo com diversos critérios.

Serviços de filiação

- Serviço linear:

  Só entrega vistas numa dada partição do sistema.

- Serviço parcial:

  Entrega vistas em todas as partições.

  Parcial forte: as vistas não se interseptam (mas não existem soluções não bloqueantes para oferecer este serviço).

  Parcial fraca: as vistas interseptam-se (mas o que se pode fazer com este serviço?).

Concretização da filiação

- É necessário que os processos troquem informação entre sí até chegarem a acordo acerca de uma nova vista.
- Dado que processos podem recuperar ou falhar durante a execução do algoritmo, é necessário arranjar maneira de estabilizar uma vista.

Comunicação em grupo

- Enviar uma mensagem para um grupo de processos.
- Endereço lógico:

  Enviar para o grupo sem indicar explicitamente a identificação (ou o número) dos membros.
- Difusão fiável:

  Se uma mensagem chega a um membro do grupo, deve chegar a todos.

  Quem são "todos"? Difusão fiável indissociável da filiação.

Sincronia virtual

Difusão virtualmente síncrona: Considere-se um grupo $\mathcal{G}$, uma vista $V^i(\mathcal{G})$, e uma mensagem m difundida para $\mathcal{G}$. A difusão m é virtualmente síncrona na vista $V^i(\mathcal{G})$ se e só se: para todo o p pertencente a $V^i(\mathcal{G})$ tal que p tenha entregue m na vista $V^i(\mathcal{G})$ e tenha instalado a vista $V^{i+1}(\mathcal{G})$, então todos os processos $q \in V^i(\mathcal{G})$ que tenham instalado $V^{i+1}(\mathcal{G})$, também entregaram m antes de instalar $V^{i+1}(\mathcal{G})$.

Dificuldades da difusão fiável

- Uma mensagem pode perder-se e só ser recebida por um sub-conjunto dos membros.

  Confirmação de recepção e retransmissão de mensagens.

  O responsável pela retransmissão, pode ser o processo a falhar.
- Ao mudar a vista, é necessário ver quais as mensagens que foram entregues.

Difusão uniforme

Difusão uniforme virtualmente síncrona: Considere-se um grupo $\mathcal{G}$, uma vista $V^i(\mathcal{G})$, e uma mensagem m difundida para $\mathcal{G}$. A difusão m é virtualmente síncrona na vista $V^i(\mathcal{G})$ se e só se: para todo o p pertencente a $V^i(\mathcal{G})$ tal que p tenha entregue m na vista $V^i(\mathcal{G})$ (mesmo que não tenha instalado a vista $V^{i+1}(\mathcal{G})$), então todos os processos $q \in V^i(\mathcal{G})$ que tenham instalado $V^{i+1}(\mathcal{G})$, também entregaram m antes de instalar $V^{i+1}(\mathcal{G})$.

Problemas da difusão uniforme

- Não se pode entregar uma mensagem antes de ter a certeza de que vai ser entregue.

  É necessário um turno adicional.

  Pode bloquear no caso de haver partições.

Ordenação de mensagens em difusão

- Ordenação causal.

  Semelhante à comunicação ponto-a-ponto.

- Ordenação total.

  Todas as mensagens são recebidas pela mesma ordem em todos os processos.

  Quem são todos os processos?

Concretização de ordem total

- Usar um processo como sequenciador.
- Usar relógios lógicos ou relógios físicos.

  Ordenar as mensagens concorrentes de acordo com um critério deterministico.

Resumo

- Abstracções.
- Detectores de falhas.
- Filiação em grupo.
- Primitivas de comunicação em grupo.