**IESD**

Notas sobre a prática 1

**Objectivos:**

- Tomar contacto com a estruturação ISoS;

- Discussão sobre a estratégia de coordenação dos serviços;

- Fazer pequenas alterações ao código.

Caso: Acesso distribuído concorrente a um serviço que mantém um vector de inteiros com N elementos iniciados com valores arbitrários.

Modelo ISoS – Isystem -> CES -> Service

Validação-são admitidas unicamente transferências entre elementos -> Ocorrência de violação do invariante do somatório de elementos do vector[i] quando múltiplas entidades computacionais autónomas Service, enquanto clientes, acedem às operações de leitura e escrita sobre elementos do vector.

* **Propôr uma estratégia de coordenação que garanta o invariante.**

Questões para discussão:

. Modelo de interacção entre o Service cliente e servidor (implementa a interface vector). O servidor, neste momento, é um Webservice -> ver questões da aula de 19/3: se é directo, indirecto…

. Situação para um único cliente a aceder ao vector -> situação actual;

. Múltiplos clientes sem qualquer mecanismo de coordenação -> só assim é que se vai provocar a violação do invariante;

. Discussão do ou dos mecanismos de coordenação a considerar para que seja garantido o invariante do somatório;

- Atomicidade das transacções dos validadores;

. Qustionar a hipótese de um número muito elevado de elementos Service clientes do serviço vector (e.g. na ordem dos milhões, …) -> CMSP (10% mercado europeu são 50 milhões), problema da escala;

. Serviço crítico, com probabilidade de falha muito reduzida, podendo ser enquadrado enquanto serviço tolerante a falhas -> haverá ser uma probabilidade de falha;

. Importância para o caso CMSP (Collaborative Mobility SP) dos problemas, estratégias/técnicas de solução ou conceitos discutidos;

* **Caso de validação, Vector e CMSP.**

Tópicos relevantes -> Neste momento não interessa, é uma provocação

. Ordenação de eventos;

. Exclusão mútua;

. Atomicidade;

- 2PC ?

. Controlo de concorrência;

. Operações que podem gerar conflitos (*confliciting operations*);

. Bloqueios (*deadlocks*);

. Algoritmos de eleição;

. Procura de acordos (*agreements*);

. Modelo de transacções X/Open;

. Protocolo *two-phase-commit* (2PC).

- Atomic commitment protocol;

**Alterações ao código**

O cliente repete sucessivamente transferências entre elementos (e.g. transferir do índice 1 para o índice 3 e, depois, do 3 para o 1 e repetir); Convém que os valores não cheguem a zero ou, caso cheguem a zero, haja uma transferência de sentido contrário para repor o valor.

Os clientes garantem sempre que, quando retiram um valor determinado de uma posição do vector, adicionam-no noutra posição do vector. Ou seja, só há transferências entre elementos.

No servidor (pode ser no main() ), há uma thread sempre a verificar se o invariante se alterou. Caso tenha havido uma violação do invariante, aparece um alerta na consola (para já). Provoca-se a situação de inconsistência para verificar se aparece o aviso.

6/4/2021

Grupo 4