

## **Distributed Systems – Fault Tolerance**

## **Quorum Replication**

André Carvalho da Cunha Martins PG47009 Luís Carlos da Costa Salazar Martins PG47412 José Pedro Castro Ferreira PG47375







**Fig. 1.** PG47009

**Fig. 2.** PG47412

**Fig. 3.** PG47375

18 de março de 2022

## 1 Justificação para necessidade de lock na operação CAS

Durante a realização deste guião prático, foram várias vezes confrontados com problemas de controlo de concorrência. Uma vez que os testes eram realizados com múltiplos servidores e múltiplos clientes, era necessário ter um controlo de concorrência eficaz e que nos permitisse obter as respostas corretas e na ordem correta.

Posto isto, a operação CAS (Compare and Swap) envolvia tanto as operações de leitura como as operações de escrita, ou seja, era necessário assegurar a integridade e garantir que as trocas eram feitas com os valores corretos. Ao implementar um lock durante a operação de CAS, os servidores ficavam inalteráveis até à conclusão, ou envio de erro, da operação de troca, garantindo assim que todos os valores analisados e alterados seriam os corretos.

## 2 Contra-exemplo obtido com Maelstrom

Como contra-exemplo apresentado, foi definida uma versão sem locks e, recorrendo à ajuda do Maelstrom, é possível observar erros obtidos quando se fazem dois pedidos, um de escrita e um de CAS, que iriam afetar os valores pretendidos e não iriam retornar o resultado correto.

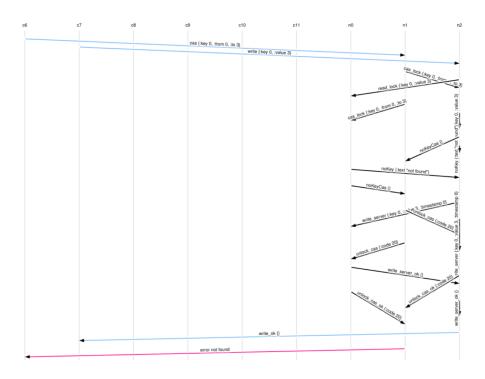


Fig. 4. Contra-exemplo Maelstrom