

Aula 14 - Broadcasts (cont.)

Monday, April 25, 2016 13:40

Propriedades do reliable broadcast

1. Acordo:
 - a. Se um processo correto executa `deliver(msg)`, então todo processo correto executa `deliver(msg)`.
2. Validade:
 - a. Se um processo correto executa `broadcast(msg)` então todos os processos corretos executam `deliver(msg)`
3. Integridade
 - a. Não há mensagens espúrias

Entrega ordenada de mensagens

Vamos estudar 3 tipos de broadcast:

- FIFO Broadcast
- Causal Broadcast
- Atomic Broadcast

FIFO Broadcast

FIFO: First in, First Out

O broadcast FIFO garante a ordem das mensagens transmitidas por um único processo p .

A propriedade que deve ser garantida é:

FIFO order (ordem FIFO):

[definição ingênua (naïve)]

Se um processo p transmite a mensagem m antes de transmitir a mensagem m' , então todos os processos corretos entregam m antes de entregar m' .

Problema desta definição:

`fbcast = FifoBroadcast`

a
 $bcast(m')$

(m^2)
 (m^3)

b
 deliver (m^1)
 (m^2)
 (m^3)

c
 deliver (m^1)
 (m^2)
 (m^3)

d
 deliver (m^1)
 (m^3)

Ou seja, mesmo que m_1, m_2 & m_3 sejam entregues, pode ser que nem todos cheguem.

Definição correta da FIFO order:

Se um processo p transmite a mensagem m antes de transmitir a mensagem m' , então todos os processos só entregam m' se antes tiverem entregado m .

Para ordenar as mensagens, basta usar os números de sequência de mensagens de cada processo $seq\#()$.

Causal Broadcast

Garante a ordem imposta pela relação "aconteceu-antes-de".

Para dois eventos e, f

1. Se e e f aconteceram no mesmo processo, podemos ordenar com o relógio local $e \rightarrow f$.
 - a. Neste caso específico, o que ordenamos?
 - b. Ex: $\text{deliver}(m) \rightarrow \text{broadcast}(m')$
2. Entre dois processos quaisquer, o $\text{broadcast}(msg)$ acontece-antes-de $\text{deliver}(msg)$
3. Transitividade.

Definição da ordem causal

Se o broadcast de uma mensagem m aconteceu-antes-do broadcast de uma mensagem m' ,

Então fazer duas definições:

- Ingênua
 - o Então, em todo todo processo correto, $\text{deliver}(m) \rightarrow \text{deliver}(m')$.
- Correta
 - o Nenhum processo correto executa $\text{deliver}(m')$ sem ter executado $\text{deliver}(m)$.

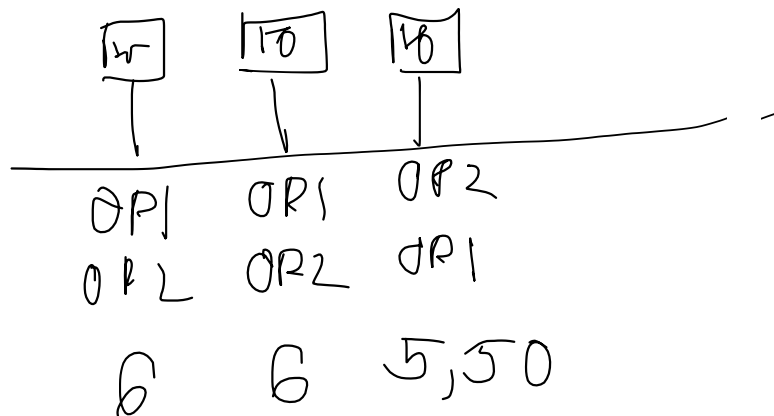
Broadcast atômico

O mais importante de todos! Necessário para implementar a replicação distribuída.

OP1: Executar $rep \times 10\% + rep$

OP2: Executar $rep = rep - 5$

As duas ações têm que ser executadas na mesma ordem em TODAS as réplicas para garantir a consistência.



Perdeu
Consistência!!!

Que tal aplicar a ordem FIFO (para garantir a consistência da replicação distribuída)?

Não! FIFO só garante a ordem de mensagens vindas de um mesmo processo.

Que tal aplicar ordem causal?

Não! A ordem causal é uma ordem parcial (eventos concorrentes não são ordenados)

No broadcast atômico, todos os processos entregam todas as mensagens na mesma ordem - garantindo a consistência distribuída

- Em outras palavras, todos os processos entregam a mesma sequência de mensagens

A propriedade do broadcast atômico é a:

- Ordem TOTAL - se dois processos corretos quaisquer entregarem as mensagens m e m' , então p entrega m antes de entregar m' se e somente se q entrega m antes de entregar m' .

Atomic broadcast é FIFO broadcast? Não!

Causal broadcast é atômico? Não!

Causal broadcast é FIFO? Sim!