

Algoritmos distribuídos: Consenso hierárquico

André Régis
Cauê Baasch
Matheus Roque

Sumário

- O que é consenso em sistemas distribuídos?
- O que é consenso hierárquico?
- Como o consenso hierárquico funciona?
- Implementação do consenso hierárquico
- Aplicação utilizando o algoritmo

O que é consenso em sistemas distribuídos?

Module 5.1 Interface and properties of consensus

Module:

Name: (regular) Consensus (c).

Events:

Request: $\langle cPropose \mid v \rangle$: Used to propose a value for consensus.

Indication: $\langle cDecide \mid v \rangle$: Used to indicate the decided value for consensus.

Properties:

C1: Termination: Every correct process eventually decides some value.

C2: Validity: If a process decides v , then v was proposed by some process.

C3: Integrity: No process decides twice.

C4: Agreement: No two correct processes decide differently.

O que é consenso hierárquico?

Algorithm 5.2 Hierarchical Consensus

Implements:

Consensus (c).

Uses:

BestEffortBroadcast (beb);
PerfectFailureDetector (\mathcal{P}).

upon event $\langle \text{Init} \rangle$ **do**

detected := \emptyset ; round := 1;
proposal := \perp ; proposer := 0;
for $i = 1$ **to** N **do**
 delivered[i] := broadcast[i] := false;

upon event $\langle \text{crash} \mid p_i \rangle$ **do**

detected := detected $\cup \{\text{rank}(p_i)\}$;

upon event $\langle \text{cPropose} \mid v \rangle \wedge (\text{proposal} = \perp)$ **do**

proposal := v ;

upon (round = rank (self)) \wedge (proposal $\neq \perp$) \wedge (broadcast[round] = false) **do**

 broadcast[round] := true;

trigger $\langle \text{cDecide} \mid \text{proposal} \rangle$;

trigger $\langle \text{bebBroadcast} \mid [\text{DECIDED}, \text{round}, \text{proposal}] \rangle$;

upon (round \in detected) \vee (delivered[round] = true) **do**

 round := round + 1;

upon event $\langle \text{bebDeliver} \mid p_i, [\text{DECIDED}, r, v] \rangle$ **do**

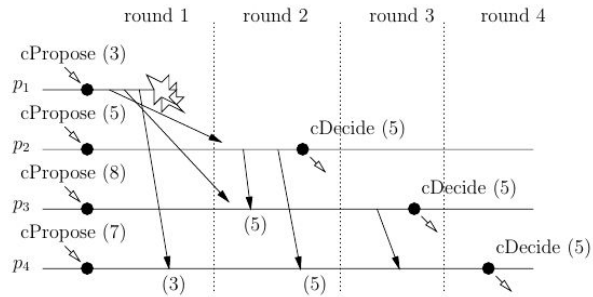
if ($r < \text{rank}(\text{self})$) \wedge ($r > \text{proposer}$) **then**

 proposal := v ;

 proposer := r ;

 delivered[r] := true;

Como o consenso hierárquico funciona?



Prós:

- O valor decidido está disponível assim que o primeiro processo decide
- O número de mensagens necessárias é consideravelmente menor do que o consenso por *flooding*
- O número de rodadas para consenso é constante, mesmo em casos sem falhas

Cons:

- Sempre é necessário N rodadas para N processos decidirem

Implementação do consenso hierárquico

Aplicação utilizando o algoritmo