Accordo Bizantino / 03-05

3 generali, 2 leali — decisione: A o W

(o senza scambiano messaggi sul da farsi: vista la guerra

ogni attesa è detta round

u

v z

se in disaccordo i leali finiscono male

bastano 2 "voti"

(es. se u e z (sleale) A ma v no è un problema)

messaggi non firmati

supposti u e v leali:

di chi riceve

mess mandati da	u	v	z
u		A	A
v	A		A
z	W	W	

u -> A

v -> A

v -> W

z può mandarli diversi

u e v ricevono A e W, non conosco chi invia (il non questo)

quindi maggioranza vince

se u e v d'accordo, lo sleale manda diverso

	u	v	z
u		A	A
v	W		W
z	A	W	

u -> A

v -> A

v -> W

u e v non sono d'accordo

z può inviare una volta

(diversi a entrambi anche)

Il consenso può essere realizzato se dati t processi infami realizzati:

ho $n \geq 3t+1$ (non arreco) sono il n° di processi tot, ne servono 4 x raggiungere un accordo con $n \geq 3t+1$ quindi $3t+1$ n.: $3 \neq 4$

Dato $i = 1, \dots, n$ processo di cui t inaffidabili
della V consenso $b(i) \in (0,1)$ bit i-esimo processo assume 1 o 0

[consenso: dopo un certo numero di round $b(i) = v$ $\forall j$ affidabile
validità: ci fosse unanimità su $v_0 \rightarrow v = v_0$ (se eravamo d'accordo vince quello)
↳ 2 vincoli:

$\forall$ proc affidabile j: input I: $b_0(j)$ / output O: con $p = \frac{1}{2}$ $b(j) = v$

PROT n-C | dalla maggioranza (quindi: bit è 1 o 0) della maj voti maggioritari della tally (con pla di voti)
↳ quanti 1 ci sono stati

es. su 301 201 mi fido, ios no
es se misuro 204 "1" gli altri non potranno misurare più di 201 "0" (per vincolo: soglia $h = 2t+1$, voti affidabili)
→ se ho maj di un tipo non posso averlo dell'altro

Algoritmo

while T // in 1 round ho $\frac{1}{2}$ di p di aver trovato accordo, in 2 ho $\frac{3}{4}, \frac{3}{8} \rightarrow \frac{7}{8} \dots$
- trasmetto $b(j)$; - ricevo gli altri bit - $maj(j) \leftarrow$ maggioranza
- $tally(j) \leftarrow$ n° voti maggioritari

- if $tally(j) \geq 2t+1$ (h affidabili) then $b(j) = maj(j)$ se tutto violerebbe il vincolo di validità
else if TESTA $b(j) = 1$ else $b(j) = 0$

PROT LV | $J : b_0(J)$ / $O : b(J) = v$ $\quad n = 8t+1$

$L = \lceil n/2 \rceil + t + 1 = 5t+1 \qquad H = L + t = 6t+1$

se uno supera la soglia grande → tutti superano la piccola

es. $tally(J^*) \leq L \qquad tally(J) < H$

(se sono tutti sotto la piccola, allora lo sono anche della grande) → con la doppia soglia ho depotenziato gli inaffidabili

es. se $J^* K^* \mid maj(J^*) \neq maj(K^*)$ allora $tally(J) < L$

Algoritmo

while True $\qquad 4t+1$

- trasmetti $b(J)$ • ricevi gli altri bit • forma maj e tally
- if TESTA then soglia = H else soglia = L
- if $tally(J) \geq$ soglia then $b(J) = maj(J)$ else $b(J) = 0$
- if $tally(J) \geq 7t+1$ then $b(J) = maj(J)$

$E[\#\text{passi}]$ = valore atteso di passi = 2.

ovvero la versione LoSVegas converge a un numero costante di iterazioni pari a 2