Possibili Strategie di Server e Supervisor

Lapo Frati

UNIVERSITÀ DI PISA

July 6, 2014

Contents

1 Minimo

La più semplice strategia che può essere adottata consiste nel far si che i Server inviino al Supervisor soltanto il minimo delle loro misurazioni ed il Supervisor selezioni il minimo tra i dati ricevuti. Ma quale è la probabilità di successo di questa strategia? Gli intervalli di tempo che intercorrono tra le ricezioni dei messaggi da parte dei server sono sempre multipli del secret dei client che li inviano. Il secret di un client viene correttamente individuato se questi invia un messaggio per almeno due volte consecutive allo stesso server ,poichè esattamente un secret è il minimo valore misurabile. Quale è la probabilità che questo avvenga? Supponiamo che i server siano palline numerate in un'urna e che scegliere il server a cui inviare un messaggio equivalga ad estrarre una pallina da quest'urna. Supponiamo quindi di effettuare n estrazioni con rimpiazzo di palline numerate da 0 a k e sia W

il numero di estrazioni consecutive dello stesso numero nelle n estrazioni:

$$\begin{aligned} P(successo) &= P(EstarreAlmeno2VolteDiFilaLoStessoNumero) \\ &= P(W \geq 2) \\ &= 1 - P(W = 1) \end{aligned}$$

Calcoliamo allora $P(W = 1)$ come il rapporto tra i casi favorevoli e quelli totali:

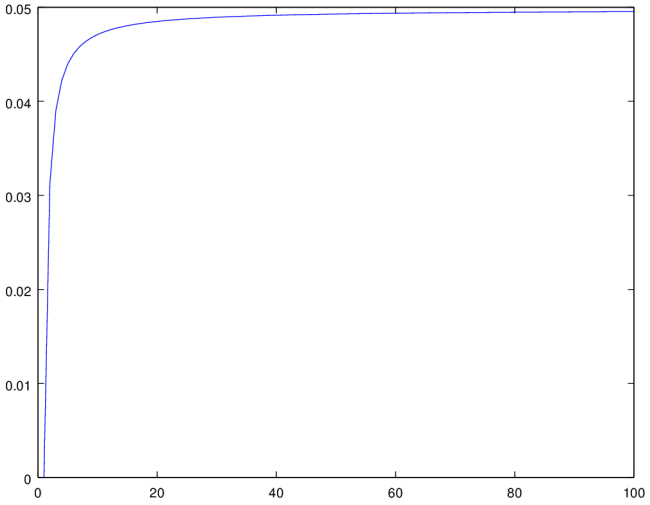
- Casi favorevoli: ho k possibili scelte per il primo numero estratto e *k-1* scelte per i restanti *n-1*, quindi in totale ho $k(k-1)^{(n-1)}$ possibili scelte.
- Casi totali: ho k scelte per ognuns delle n estrazioni, quindi ho k^n possibli scelte.

Pertanto:

$$\begin{aligned} P(W = 1) &= \frac{k(k-1)^{(n-1)}}{k^n} \\ &= \left(\frac{k-1}{k}\right)^{(n-1)} \end{aligned}$$

Poichè $\frac{k-1}{k} < 1$ e $n > 3k$ allora $\left(\frac{k-1}{k}\right)^{3k-1} > \left(\frac{k-1}{k}\right)^{n-1}$
Possiamo quindi studiare il comportamento di

$$f(k) = \left(\frac{k-1}{k}\right)^{3k-1}$$



Come si evince facilmente dal grafico la funzione è monotona crescente, calcoliamone quindi il limite all'infinito per trovare il massimo valore assunto da $f(k)$:

$$\begin{aligned}\lim_{k\to+\infty}\left(\frac{k-1}{k}\right)^{3k-1} &= \lim_{k\to+\infty}\frac{\left[\left(\frac{k-1}{k}\right)^{k}\right]^{3}}{\left(1-\frac{1}{k}\right)} \\ &= \lim_{k\to+\infty}\frac{\left[\left(1-\frac{1}{k}\right)\frac{1}{\left(1+\frac{1}{k-1}\right)^{(k-1)}}\right]^{3}}{\left(1-\frac{1}{k}\right)} \\ &= \frac{1}{e^3}\end{aligned}$$

poichè

$$\lim_{k\to+\infty}\left(1-\frac{1}{k}\right)=1$$

e

$$\lim_{k\to+\infty}\left(1+\frac{1}{k-1}\right)^{(k-1)}=e$$

Dato che $\frac{1}{e^3} \approx 0.05$

$$P(W = 1) \leq 0.05$$

e quindi

$$\begin{aligned} P(Successo) &= 1 - P(W = 1) \\ &\geq 1 - 0.05 \\ &\geq 95\% \end{aligned}$$

Inoltre, con una piccola accortezza, è possibile migliorarla ulteriormente; infatti se la nostra stima fallisce sarà altamente probabile che individui il doppio del secret, pertanto se otteniamo una stima che è maggiore di 3000, dividendola per due è molto probabile ottenere proprio il secret.

2 Tempo Totale

E' possibile portare la probabilità di successo al 100% ?

Si, una possibile strategia potrebbe essere: facciamo si che ogni server tenga traccia del tempo di arrivo del primo e dell'ultimo messaggio ricevuti e che conti il numero totale dei messaggi ricevuti.

Con queste informazioni il supervisor seleziona il massimo dei tempi di arrivo dell'ultimo messaggio ricevuto da ciascun server ed il minimo dei tempi di arrivo dei primi messaggi. La differenza tra questi due valori rappresenta il totale di secret inviati dal client.

Supponiamo: che si abbiano n server, che dal client i siano stati inviati in totale k_i messaggi ogni $secret_i$ millisecondi e che $count(i, j)$ sia il numero di messaggi che il server j ha ricevuto dal client i.

La differenza tra il massimo dei tempi di arrivo dell'ultimo messaggio del client i ed il minimo dei tempi di arrivo del primo messaggio equivale a $(k_i - 1)secret_i$ (il -1 è dovuto al fatto che un client comincia ad attendere solo dopo il primo invio).

Il supervisor conosce quindi $(k_i - 1)secret_i$ e può calcolare:

$$k_i = \sum_{j=1}^{n} count(i, j)$$

Pertanto

$$secret_i = \frac{(maxLast - minFirst)}{(\sum_{j=1}^{n} count(i, j)) - 1}$$

L'unica pecca di questa strategia sarebbe che se i server utilizzano la stessa strategia del supervisor per fornire una propria stima, questa risulta piuttosto approssimativa.

3 Minimo Migliorato

Un' ultima possibile strategia è una versione raffinata della strategia del Minimo.

Supponiamo che il supervisor riceva dai server, per ogni client, sia il minimo dei tempi tra i messaggi, sia il tempo di ricezione dell'ultimo messaggio di quel client e consideriamo i possibli scenari per l'invio degli ultimi due messaggi da parte di un client i:

- Entrambi i messaggi vengono inviati allo stesso server: in questo caso il minimo calcolato da quel server è esattamente $secret_i$.
- I due messaggi vengono inviati a due server differenti: in questo caso la differenza fra i tempi di ultima ricezione di quei server è esattamente $secret_i$.

Pertanto il supervisor può stimare il $secret_i$ calcolando il minimo tra i minimi ricevuti dai server e la differenza dei i due maggiori tempi di ultima ricezione, individuandolo con una precisione del 100% . I server invece potranno fornire una stima del secret usando il minimo da loro stimato.