Resolución de conflictos en sistemas distribuidos

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Resolución de conflictos en sistemas distribuidos

Tenemos n procesos $\{P_1,..,P_n\}$

Cada proceso no tiene comunicación con el resto.

Pueden solicitar acceso a un mismo recurso (ej: base de datos)

Las solicitudes se realizan en rondas discretas.

Si solo 1 proceso pide el recurso se concede el acceso

Si mas de un proceso solicita acceso, ninguno lo obtiene.

Cómo podemos resolver los accesos y maximizar la probabilidad de obtener el recurso?

Quiebre de simetría

Si nadie solicita el recurso

El turno se “pierde”

Si solo 1 proceso solicita el recurso

Lo obtendrá con éxito

Si mas de 1 proceso solicita el recurso

NINGUNO lo obtiene y deberá volver a solicitarlo

Si cada vez que un proceso falla,

Solicita el proceso inmediatamente en el turno siguiente

Se provocará un atasco. ← queremos evitar que esto ocurra

Una solución mediante algoritmo randomizado

Sea p >0

- la probabilidad de que un proceso solicita acceso en un determinado turno
- La probabilidad es independiente entre cada proceso.

En cada ronda los procesos deciden acceder o no en base a p

- Si en un turno un proceso pide el recurso y falla,
- No lo pedirá deterministicamente en el siguiente.
- Se aplicará la misma probabilidad “p” para solicitarlo.

Cómo se comporta el sistema? Cual deber ser el valor de p?

Evento de intento de acceso

Llamaremos al evento A[i,t]

Intento del proceso P_i de acceder al recurso en la ronda t

Sabemos que estará signado por la probabilidad p

Podemos determinar tanto el intento como el no-intento:

$Prob(A[i,t]) = p$

$\overline{Prob(A[i,t])} = 1- p$

Evento de acceso con éxito

Llamaremos S[i,t] al evento de éxito en el acceso

- Implica que el proceso Pi intento acceder al recurso en la ronda t
- Ningún otro proceso lo intento en esa misma ronda

Podemos expresarlo como:

$$S[i,t] = A[i,t] \cap \left(\cap \overline{A[j,t]}\right),\ j \neq i$$

Su probabilidad:

$$Prob(S[i,t]) = Prob(A[i,t]) \cdot \prod_{J \neq i} Prob\left(\overline{A[j,t]}\right) = p * (1-p)^{(n-1)}$$

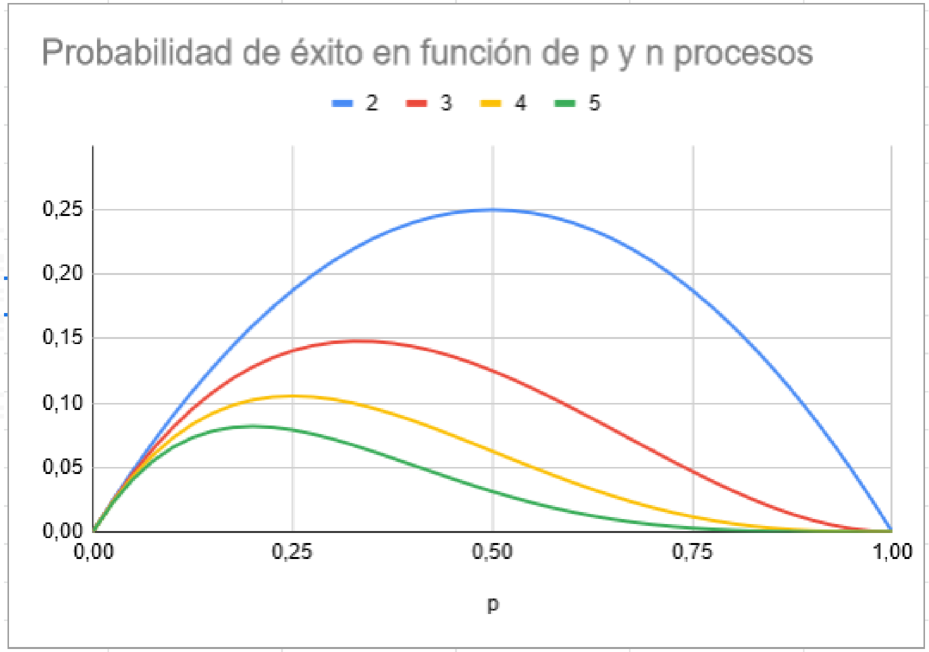
Maximizar la probabilidad de éxito

Sea $f(p) = p * (1-p)^{(n-1)}$

P es un valor entre 0 y 1

f(p=0) = 0 ← Nunca intento

f(p=1) = 0 ← Siempre intento



Maximización de f(p),

cálculo de la derivada $f'(p) = (1-p)^{n-1} - (n-1)\,p\,(1-p)^{n-2}$

La derivada

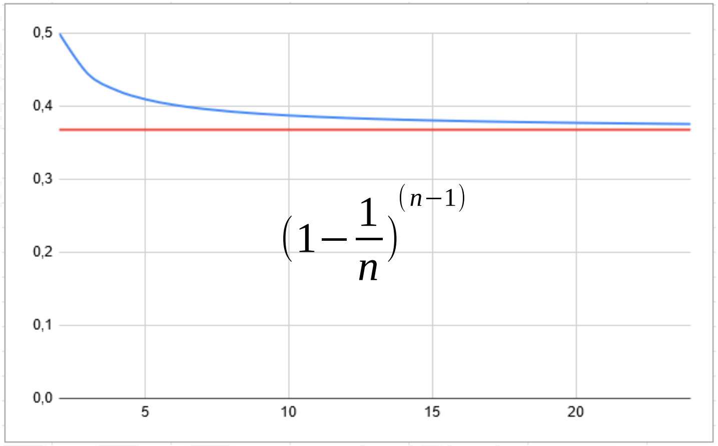
tiene un cero en p= 1/n → con ese valor maximizamos

Maximizar la probabilidad de éxito (cont.)

Una vez que tenemos p

Podemos reemplazarlo en nuestra probabilidad

$$Prob(S[i,t]) = \frac{1}{n} * \left(1 - \frac{1}{n}\right)^{(n-1)}$$



Como se comporta esta función

En función de n?

Si analizamos $\left(1 - \frac{1}{n}\right)^{(n-1)}$

Veremos que (para n>1) inicia en ½ y converge monotonamente a 1/e

Por lo que podemos acotar

$$\frac{1}{en} \leq Prob(S[i,t]) \leq \frac{1}{2n} \Rightarrow Prob(S[i,t]) = \Theta\left(\frac{1}{n}\right)$$

¿Cuánto tardará un proceso en tener éxito?

Si tomamos el proceso Pi

¿Cuanto tardara en acceder al recurso?

Si n es grande

Difícilmente lo consiga en su primer intento

Llamaremos F[i,t] Fallo (del protocolo) para el proceso i

Si luego de t rondas el proceso i aun no pudo acceder al recurso

¿Cuánto tardará un proceso en tener éxito? (cont.)

Podemos calcular

$$Pr(F[i,t]) = \overline{Pr(S[i,1])} * \overline{Pr(S[i,2])} * \ldots * \overline{Pr(S[i,t])}$$

Con

$$\overline{Prob(S[i,t])} = 1 - Prob(S[i,t])$$

$$\frac{1}{en} \leq Prob(S[i,t]) \leq \frac{1}{2n}$$

Por lo tanto

$$Pr(F[i,t]) \leq \left(1 - \frac{1}{en}\right)^t$$

¿Cuánto tardará un proceso en tener éxito? (cont.)

Si elegimos t=⌈en⌉

$$Pr(F[i,t]) \leq \left(1-\frac{1}{en}\right)^{\lceil en \rceil} \leq \left(1-\frac{1}{en}\right)^{en} \leq \frac{1}{e}$$

La probabilidad de que el proceso Pi no tenga éxito entre t=1 y ⌈en⌉

Está acotada por e^{-1} (de forma independiente a n)

¿Cuánto tardará un proceso en tener éxito? (cont.)

Si elegimos t ligeramente superior a ⌈en⌉

Por ejemplo ⌈en⌉*(c*ln n)

$$Pr(F[i,t]) \leq \left(\left(1 - \frac{1}{en}\right)^{\lceil en \rceil} \right)^{c * \ln n} \leq e^{-c \ln n} = n^{-c}$$

La probabilidad de que el proceso Pi no tenga éxito entre t=⌈en⌉ y Θ(n ln n)

Desciende precipitadamente

¿Cuánto tardará un proceso en tener éxito? (cont.)

En conclusión

Antes de $\Theta(n)$ rondas la probabilidad de fracaso está acotada por una constante

Luego, entre $\Theta(n)$ y $\Theta(n \ln n)$ rondas la probabilidad cae, delimitado por un polinomio inverso en n.

¿Cuánto tardarán todos los procesos en tener éxito?

Llamaremos fallo general del protocolo F[t]

Si luego de t rondas al menos un proceso aun no pudo acceder al recurso

¿Qué valor debe tener t

Para que la probabilidad de fallo general del protocolo sea razonablemente pequeña?

$$Pr(F[t]) = Pr(F[1,t] \cup F[2,t] \cup \ldots \cup F[n,t])$$

Es la unión de eventos no independientes. Es complejo de calcular. Pero podemos determinar una cota

¿Cuánto tardarán todos los procesos en tener éxito?

Se puede ver que

$$Pr(F[t]) = Pr(F[1,t] \cup F[2,t] \cup \ldots \cup F[n,t]) \leq \sum_{i=1}^{n} Pr(F[i,t])$$

Las probabilidades de fallo

de cada proceso son iguales

Tenemos

n procesos → sumamos n veces F[i,t]

Para que la probabilidad de fracaso sea pequeña

F[i,t] tiene que ser significativamente menor a 1/n

¿Cuánto tardarán todos los procesos en tener éxito?

Antes de Θ(n) rondas la F[i,t] está acotada por una constante

$$Pr(F[t]) \leq \sum_{i=1}^{n} Pr(F[i,t]) \leq nc$$

No logramos la cota requerida

Si tomamos t=⌈en⌉*(2*ln n) ← c=2

$$Pr(F[t]) \leq \sum_{i=1}^{n} Pr(F[i,t]) \leq n * n^{-2} = \frac{1}{n}$$

logramos la cota que nos solicitamos

¿Cuánto tardarán todos los procesos en tener éxito?

En conclusión

Con probabilidad de al menos $1 - 1/n$

Todos los procesos

Tienen éxito en acceder al recurso al menos 1 vez

En no mas de

$t=\lceil en \rceil * (2 * \ln n)$ rondas

Presentación realizada en Junio de 2020