SISTEM DE TIP COADĂ CU DOUĂ SERVERE LEGATE ÎN PARALEL

Variabilă de timp : $\boxed{t}$

Variabile de stare a sistemului (SS) : $\boxed{(n, i_1, i_2)}$

nr. de clienți din sistem (n)
clientul curent la serverul 1 (i_1)
clientul curent la serverul 2 (i_2)

Variabile contor : $\boxed{N_A}$ → nr. de sosiri până la momentul de timp t

$\boxed{C_j}$ → nr. de clienți serviți de serverul j, $j \in \{1, 2\}$ până la momentul de timp t

Variabile output : $\boxed{A(n)}$ → timpul la care sosește clientul n

$\boxed{D(n)}$ → timpul la care clientul n părăsește sistemul

Lista de evenimente : $\boxed{t_A, t_1, t_2}$

$\boxed{t_A}$ → timpul la care va sosi următorul client

$\boxed{t_i}$ → timpul la care clientul curent își încheie activitatea la serverul i, $i = \overline{1,2}$

OBS : (1) Dacă nu există client la serverul curent (i) atunci $t_i = \infty$, $i = \overline{1,2}$

(2) Lista de evenimente va conține mereu cele 3 variabile: t_A, t_1, t_2.

Schema de simulare

① Inițializare

$t = N_A = C_1 = C_2 = 0$

$SS = (0)$

Generăm T_0 și $t_A = T_0$, $t_1 = t_2 = \infty$

Pentru a actualiza sistemul ne deplasăm pe axa temporală până la momentul apariției următorului eveniment.

În continuare vom considera mai multe cazuri, în funcție de care element din lista de evenimente are valoarea mai mică.

$\boxed{\text{CONVENȚIE}}$: Y_i sunt v.a. cu funcția de repartiție G_i, $i = \overline{1,2}$

$SS = (n, i_1, i_2)$ dacă în sistem există n clienți și la serverul 1 clientul curent este i_1, iar la serverul 2 clientul curent este i_2

ex. $SS = (1, 0, j)$ → singurul client din sistem este clientul j și este servit de serverul 2

$SS = (0)$ → nu avem clienți în sistem

② Cazul 1: $\boxed{SS = (n, i_1, i_2) \text{ și } t_A = \min(t_A, t_1, t_2)}$

→ sosește un client, verificăm dacă poate fi servit imediat sau intră în coada de așteptare

$t = t_A$

$N_A = N_A + 1$

Folosim algoritmul ♡ → Generăm T_t și $t_A = T_t$

Output: $A(N_A) = t$

Dacă $SS = (0)$: $SS = (1, N_A, 0)$
Generăm Y_1 și $t_1 = t + Y_1$

Dacă $SS = (1, j, 0)$: $SS = (2, j, N_A)$
Generăm Y_2 și $t_2 = t + Y_2$

Dacă $SS = (1, 0, j)$: $SS = (2, N_A, j)$
Generăm Y_1 și $t_1 = t + Y_1$

Dacă $n > 1$: $SS = (n+1, i_1, i_2)$

③ Cazul 2: $\boxed{SS = (n, i_1, i_2) \text{ și } t_1 < t_A,\ t_1 \leq t_2}$

→ serverul 1 se eliberează înainte de sosirea unui client nou

$t = t_1$

$C_1 = C_1 + 1$

Output: $D(i_1) = t$

Dacă $n = 1$: $SS = (0)$
$t_1 = \infty$

Dacă $n = 2$: $SS = (1, 0, i_2)$
$t_1 = \infty$

Dacă $n > 2$: $m = \max(i_1, i_2)$ Generăm Y_1
$SS = (n-1, m+1, i_2)$ și $t_1 = t + Y_1$

$m+1$ este nr. de ordine al unui client nou

(4) Cazul 3: $\boxed{SS = (n, i_1, i_2) \text{ și } t_2 < t_A,\ t_2 < t_1}$

→ serverul 2 se eliberează înaintea serverului 1 și înainte de sosirea unui client nou

$t = t_2$

$C_2 = C_2 + 1$

Output: $D(i_2) = t$

Dacă $n = 1$: $SS = (0)$

$t_2 = \infty$

Dacă $n = 2$: $SS = (1, i_1, 0)$

$t_2 = \infty$

Dacă $n > 2$: $m = \max(i_1, i_2)$

$SS = (n-1, i_1, m+1)$

Generăm Y_2 și $t_2 = t + Y_2$

SISTEM DE TIP COADĂ CU DOUĂ SERVERE LEGATE ÎN SERIE

Variabilă de timp : $\boxed{t}$

Variabilele de stare ale sistemului (SS) : $\boxed{(n_1, n_2)}$

$\boxed{n_1}$ reprezintă nr. de clienți de la serverul 1 (cei din coadă și cel care este servit)

$\boxed{n_2}$ reprezintă nr. de clienți de la serverul 2.

Variabile contor : $\boxed{N_A}$ → nr. de sosiri până la momentul t

$\boxed{N_D}$ → nr. de plecări până la momentul t

Variabile output : $\boxed{A_1(n)}$ → momentul sosirii clientului n (nu la serverul 1, ci în sistem)

$\boxed{A_2(n)}$ → momentul sosirii clientului n la serverul 2

$\boxed{D(n)}$ → momentul plecării clientului n din sistem

Lista de evenimente : $\boxed{t_A}$ → momentul sosirii următorului client

$\boxed{t_i}$ → momentul la care clientul curent va finaliza interacțiunea cu serverul i, $i \in \{1, 2\}$

OBS : 1. Dacă nu există client la serverul i facem convenția că $t_i = \infty$, $i \in \{1, 2\}$

2. Lista de evenimente va conține mereu cele 3 variabile: t_1, t_2 și t_A.

Schema de simulare

Algoritm de generare pt. T_s

1. $t = s$
2. Generăm U_1, U_2
3. $t = t - \frac{1}{\lambda} \cdot \log U_1$
4. Dacă $U \le \frac{\lambda(t)}{\lambda}$ $T_s = t$ și STOP
5. Mergi la 2.

$s = 0$

$s = t$

① Inițializare

$t = N_A = N_D = 0$

$SS = (0,0)$

Generăm T_0 și atribuim $t_A = T_0$

$t_1 = t_2 = \infty$

Pentru a actualiza sistemul ne deplasăm pe axa temporală până la următorul eveniment.

În continuare vom considera mai multe cazuri, în funcție de care element din lista de evenimente are valoarea mai mică.

$\boxed{\text{CONVENȚIE}}$: Y_i sunt v.a. cu funcția de repartiție G_i, $i = \overline{1,2}$.

$SS = (n_1, n_2) \to$ starea sistemului

$EL = t_A, t_1, t_2 \to$ lista de evenimente

② Cazul 1 : $\boxed{t_A = \min(t_A, t_1, t_2)}$

↳ sosește un client nou și verificăm dacă serverul 1 este liber (altfel trebuie pus în coada de așteptare)

$t = t_A$

$N_A = N_A + 1$

$n_1 = n_1 + 1$

Generăm T_t și apoi $t_A = T_t$

Dacă $n_1 = 1$ atunci generăm Y_1 și $t_1 = t + Y_1$

Output: $A_1(N_A) = t$.

clientul care tocmai a sosit va fi servit imediat de serverul 1

③ Cazul 2 : $\boxed{t_1 < t_A,\ t_1 \le t_2}$

↳ serverul 1 se eliberează înainte de sosirea unui client nou

$t = t_1$

$n_1 = n_1 - 1$, $n_2 = n_2 + 1$

Dacă $n_1 = 0$ $t_1 = \infty$

Altfel, generăm Y_1 și $t_1 = t + Y_1$

Dacă $n_2 = 1$, generăm Y_2 și $t_2 = t + Y_2$

Output: $A_2(N_A - n_1) = t$

④ Cazul 3 : $\boxed{t_2 < t_A,\ t_2 < t_1}$

↳ serverul 2 se eliberează înainte de a sosi un client nou și înainte de finalizarea activității la serverul 1

$t = t_2$

$N_D = N_D + 1$

$n_2 = n_2 - 1$

Dacă $n_2 = 0$ $t_2 = \infty$

Dacă $n_2 = 1$ generăm Y_2 și $t_2 = t + Y_2$

Output: $\cancel{A_2(N_A - n_1) = t}$ $D(N_D) = t$.

Algoritm pentru generarea T_s

În toate modelele cu coadă de așteptare vom presupune că sosirea clienților se face conform cu un proces Poisson neomogen cu funcția de intensitate $\lambda(t)$ mărginită, $t>0$.

Fie λ o constantă a.î. $\lambda(t) \le \lambda \ \forall t>0$.

$T_s \to$ momentul primei sosiri după momentul s.

Generarea lui T_s

1. $t = s$ (ne plasăm în timp la momentul s)
2. Generăm U_1, U_2
3. $t = t - \frac{1}{\lambda} \cdot \log U_1$ (actualizăm timpul curent)
4. Dacă $U_2 \le \frac{\lambda(t)}{\lambda}$ atunci $T_s = t$ și STOP.
5. Mergi la 2.

UN MODEL DE ASIGURARE DE RISC

Variabila de timp : $\boxed{t}$

Variabile de stare a sistemului (SS) : $\boxed{(n, a)}$

nr. de clienți

capitalul curent al firmei

Evenimente :
1) Apariția unui client nou $P = \dfrac{\nu}{\nu + n\mu + n\lambda}$
2) Pierderea unui client $P = \dfrac{n\mu}{\nu + n\mu + n\lambda}$
3) Înregistrarea unei cereri de despăgubire $P = \dfrac{n\lambda}{\nu + n\mu + n\lambda}$

Lista de evenimente : $\boxed{t_E}$ → momentul de timp la care va apărea următorul eveniment (E)

$\boxed{\text{OBS}}$: Lista de evenimente conține doar valorile ce reprezintă t_E !

Comentariu: Dacă (n, a) reprezintă starea sistemului la momentul t, cum minimul unor v.a. independente repartizate exponențial este tot o v.a. repartizată exponențial, atunci rezultă că momentul de timp la care se produce următorul eveniment va fi egal cu $t + X$, unde $X \sim Exp(\nu + n\mu + n\lambda)$

$\boxed{\text{Obs}}$: În dreptul celor 3 tipuri de evenimente am notat probabilitățile cu care se produc respectivele evenimente !

Variabila output: $I = \begin{cases} 1, & \text{dacă firma are capital pozitiv în intervalul } [0, t] \\ 0, & \text{altfel} \end{cases}$

Schema de simulare

① **Inițializare**

$t = 0$

$a = a_0$

$n = n_0$

Generăm X și $t_E = X$

Pentru a actualiza sistemul ne deplasăm de-a lungul axei temporale până la apariția următorului eveniment, verificând în prealabil dacă acesta are loc înaintea momentul T.

② **Cazul 1** : $\boxed{t_E > T}$

$I = 1$ și STOP

(firma nu se ruinează)

③ **Cazul 2** : $\boxed{t_E \leq T}$

$a = a + n \cdot c(t_E - t)$

$t = t_E$

– continuare ③ –

avem 3 tipuri de evenimente

Generăm J :

$$\boxed{\begin{array}{l} \text{Dacă } J = 1 \\ \text{Dacă } J = 2 \\ \text{Dacă } J = 3 : \end{array}} \quad \begin{array}{l} n = n+1 \\ n = n-1 \\ \end{array}$$

folosim metoda inversă pentru v.a. discrete

Generăm Y

Dacă $Y > a$ $\quad I = 0$ STOP

Altfel $a = a - Y$

Generăm X și $t_E = t + X$

Obs. Actualizarea sistemului se reia până la finalul intervalului de timp pentru care vrem să facem simularea

Comentariu: După determinarea momentului la care are loc următorul eveniment generăm o valoare aleatoare (J) pentru a vedea tipul de eveniment ce va avea loc.

$$J : \begin{pmatrix} 1 & 2 & 3 \\ \frac{\nu}{\nu + n\mu + n\lambda} & \frac{n\mu}{\nu + n\mu + n\lambda} & \frac{n\lambda}{\nu + n\mu + n\lambda} \end{pmatrix}$$

$Y \to$ v.a. cu funcția de repartiție F ce reprezintă valoarea despăgubirii solicitate.

UN MODEL DE INVENTAR

Variabila de timp: $\boxed{t}$

Variabilele de stare a sistemului (SS): (x, y)

x → cantitatea din stoc pentru care nu există comandă

y → cantitatea ~~valoarea~~ din stoc pentru care există o comandă

Variabilele contor: $\boxed{C}$ → totalul costurilor cu comenzile până la momentul t

$\boxed{H}$ → totalul costurilor cu păstrarea produsului în inventar până la t

$\boxed{R}$ → veniturile totale până la momentul t

Lista de evenimente: $\boxed{t_0}$ → momentul la care apare un client nou

$\boxed{t_1}$ → momentul la care o comandă va fi livrată

CONVENȚIE: Dacă nu există nicio comandă curentă $t_1 = \infty$.

Schema de simulare

① Inițializare

$t = 0$

$SS = (x, y)$

Generare t_0.

② $\boxed{\text{Cazul 1: } t_0 < t_1}$ → sosește un client nou

$H = H + (t_0 - t) \cdot x \cdot h$ → deoarece între t și t_0 avem un cost de $(t_0 - t) \cdot h$ pentru fiecare din cele x unități în inventar

$t = t_0$

Generare D → v.a. ce reprezintă cererea clientului apărut la momentul t_0 (are f. de rep. G)

$w = \min(D, x)$ → partea din comandă ce poate fi onorată

$R = R + w \cdot r$

$x = x - w$

Dacă $x < s$ și $y = 0$ atunci $y = S - x$

$t_1 = t + L$

Generare U și $t_0 = t - \frac{1}{\lambda} \cdot \log(U)$

③ $\boxed{\text{Cazul 2: } t_1 \leq t_0}$ → comanda curentă a fost onorată

$H = H + (t_1 - t) \cdot x \cdot h$

$t = t_1$

$C = C + c(y)$

$x = x + y$

$y = 0, \quad t_1 = \infty$

Comentarii:

Putem rula prezenta schemă de simulare până la apariția primului eveniment după momentul T, unde T este prestabilit.

Cantitatea $\dfrac{R - C - H}{T}$ poate fi folosită ca estimare a profitului per unitate de timp

$\boxed{\text{Obs.}}$ Rulând simularea pentru diferite valori pentru s și respectiv S poate ajuta în alegerea "politicii" de inventar.