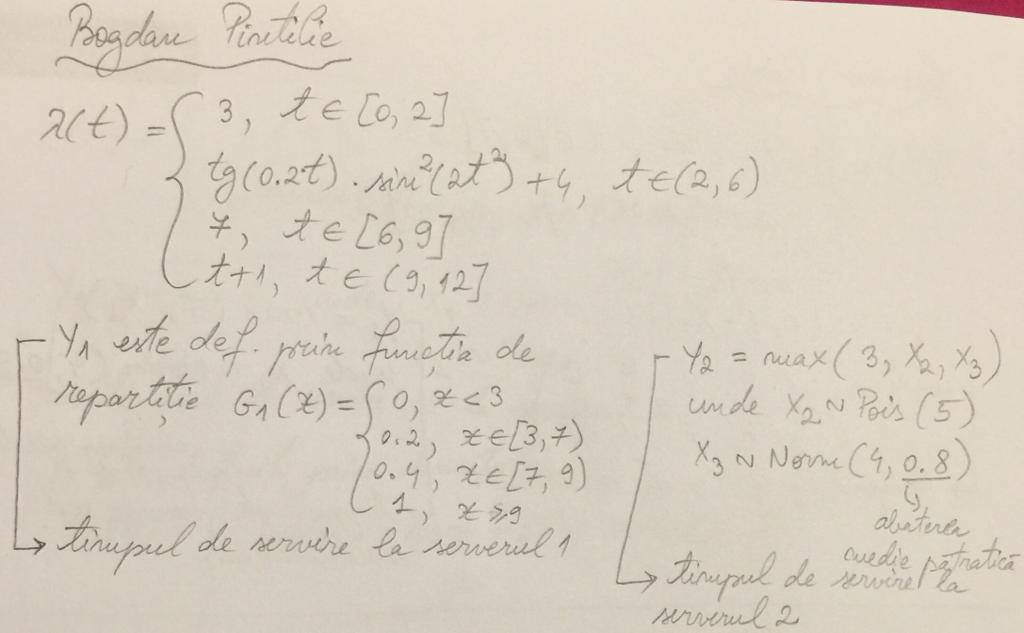
Documentatie proiect

-Tehnici de Simulare-

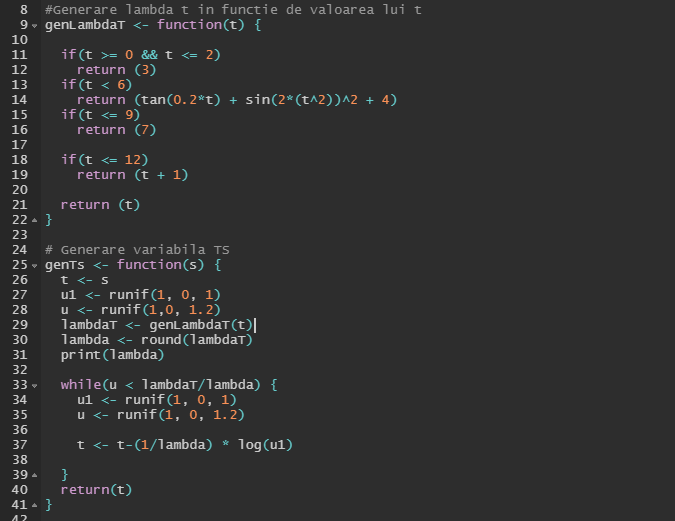
Proiect realizate de: Pintilie Bogdan-Ioan

Problema aleasa pentru a fi simulata este activitatea unei spalatorii cu auto-servire pentru automobile. Spalatoria va fi alcatuita din doua cabine, fapt pentru care aceasta spalatorie poate fi vazuta ca un sistem de tip coada cu doua servere legate in parallel. In continuare, vom prezenta scenariul cu care vom lucra, datele aferente si anumite situatii particulare.

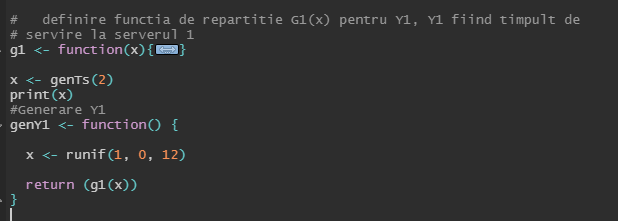
Scenariul pe care se va lucre este urmatorul: spalatoria dispune de doua servere (n.r. cabine) legate in paralel, dar si diferite in sensul ca una dintre cele doua cabine, mai exact prima, este o cabina moderna care are un timp de executie mai mic decat cea de a doua. Mai departe vom prezenta datele de lucru.



Programul de lucru ales este non-stop, iar coada poate fi oricat de mare. Acum urmeaza prezentarea codului si modul de implementare si simulare.



In imaginea de mai sus, este relizata generarea variabilei de timp cu ajutorul functiei lambda.



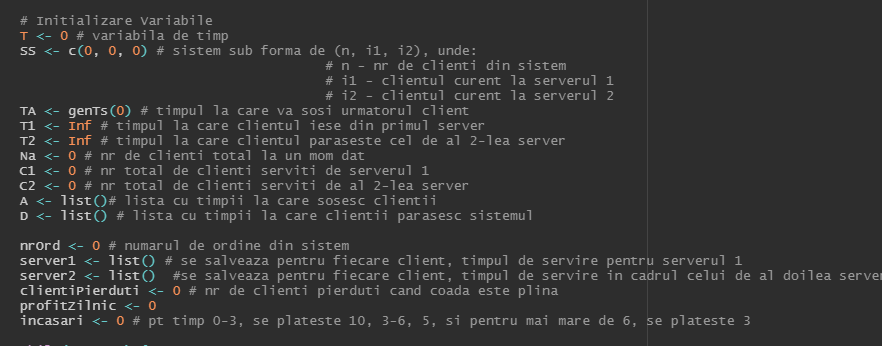
Functia de repartitie pentru timpul de servire la primul server



Definirea timpului de servire pentru cel de-al doilea server.

Codul prinicipal, unde are loc simularea propriu-zisa:

* Initializarile variabilelor folosite pentru simulare



* Simualrea pe un anumit interval de timp. In acest caz, pana cand variabila T ajunge la 400

while(T < 400) {

# Primul caz: cand soseste clientul verificam daca poate fi servit imediat

# sau daca intra in coada de asteptare

if(TA == min(TA, T1, T2)) {

T = TA

Na = Na + 1

TA = genTs(T)

A[Na] = T

if(SS[1] <= 1) { # un singur client in sistem

# verific dc sistemul este gol, i.e. nici un client in sistem

if(SS[1] == 0 & SS[2] == 0 & SS[3] == 0 ){

# adaugam clientul direct in sistem, pe prima pozitie

SS[1] = 1

SS[2] = Na

Y1 = genY1()

T1 = T + Y1

}else if(SS[1]== 1 & SS[2] != 0 & SS[3] == 0) { # serverul 1 ocupat

# intra clinetul in serverul 2, cel modern, care are un timp putin mai scurt,

# cu 20%

SS[1] = 2

SS[3] = Na

Y2 = genY2()

T2 = (T + Y2)

}else if(SS[1] == 1 & SS[2] == 0 && SS[3] != 0){ # serverul 2 ocupat

SS[1] = 2

SS[2] = Na

Y1 = genY1()

T1 = T + Y1

}

} else # Mai multi clienti in sistem

{

SS[1] = SS[1] + 1

}

}

# Caz 2: Serverul 1 se elibereaza inainte de sosirea unui nou client

if(T1 < TA & T1 <= T2){

T = T1

C1 = C1 + 1

D[SS[2]] = T

server1 = c(server1, T-unlist(A[SS[2]])) # adaugam timpul in care clientul

# n a stat in serverul 1

if(SS[1] == 1) # avem doar un client in sistem

{

SS[1] = 0

SS[2] = 0

SS[3] = 0

T1 = Inf

} else if(SS[1] == 2){ #In sistem sunt 2 clienti

SS[1] = 1

SS[2] = SS[2] - 1

T1 = Inf

} else if(SS[1] > 2){# In sistem sunt cel putin 3 clienti

nrOrd = max(SS[2], SS[3]) # nrOrd + 1 reprezinta numarul de ordine pt client nou

SS[1] = SS[1] - 1

SS[2] = nrOrd + 1

Y1 = genY1()

T1 = T + Y1

}

}

# Caz 3. Server 2 liber inaintea serverului 1

if(T2 < TA && T2 < T1) {

T = T2

C2 = C2 + T

D[SS[3]] = T

server2 = c(server2, T-unlist(A[SS[3]])) # adaugam timpul in care clientul

# n a stat in serverul 2

if(SS[1] == 1) # Doar un client in sistem

{

SS[1] = 0

SS[2] = 0

SS[3] = 0

T2 = Inf

}else if(SS[1] == 2){ # In sistem sunt 2 clienti

SS[1] = 1

SS[3] = 0

T2 = Inf

}else if(SS[1] > 2) {# Daca in sistem exista mai mult de 2 clienti

nrOrd = max(SS[2], SS[3])

SS[1] = SS[1] - 1

SS[3] = nrOrd + 1

Y2 = genY2()

T2 = T + Y2

}

}

} # sfarsit while

Dupa ce simularea s-a terminat vom afisa datele obtinue in sub forma de histograme penrtru a vedea evolutia activitatii, dar si numere/date care vor arata timpul max/min sau mediu, profit, total client pe acel interval de timp etc.

totalC <- min(length(A), length(D))

timpPetrecut <- list()

# se calculeaza timpul prntru fiecare cleint in sistem si salvam in lista definita mai sus

for(i in seq(1, 400)) {

timpPetrecut <- c(timpPetrecut,unlist(D[i]) - unlist(A[i]))

if(timpPetrecut[length(timpPetrecut)] <= 3){

incasari = incasari + 10

}else if(timpPetrecut[length(timpPetrecut)] <= 6){

incasari = incasari + 5

}else {

incasari = incasari + 3

}

}

cheltuieli = totalC \* 0.6

cat("\nIncasari: ", incasari, "\n")

cat("Total clienti: ", totalC, "\n")

hist(unlist(timpPetrecut), breaks = 100, main="Timp petrecut")

cat("timp Maxim petrecut in sistem=", max(unlist(timpPetrecut)), "\n")

cat("timp minim petrecut in sistem=", min(unlist(timpPetrecut)), "\n")

cat("media timpului petrecut in sistem=", mean(unlist(timpPetrecut)), "\n")

hist(unlist(server1), main="Timpul petrecut in primul server")

hist(unlist(server2), main="Timpul petrecut in al doilea server")

cat("Numarul mediu de clienti serviti de primul server: ",

length(server1)/totalC, "\n")

cat("NUmarul mediu de cleinti serviti de al doilea server: ",

length(server2)/totalC, "\n")

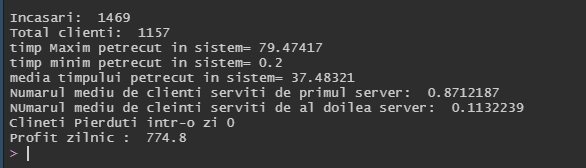
cat("Clineti Pierduti intr-o zi", clientiPierduti, "\n")

profitZilnic= incasari - cheltuieli

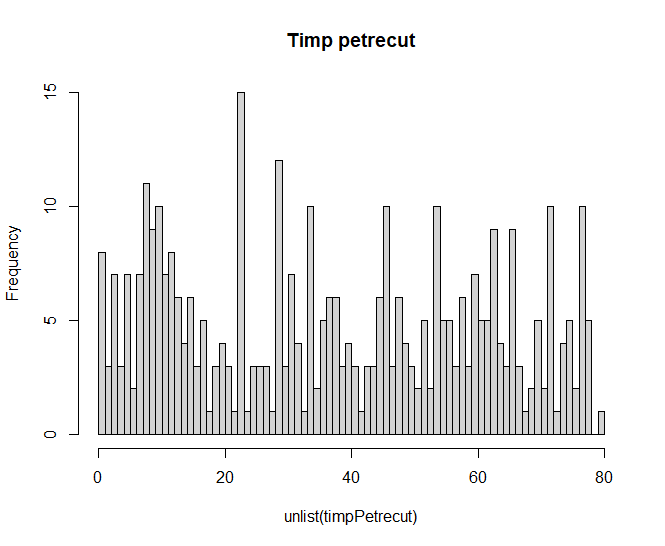
cat("Profit zilnic : ", profitZilnic, "\n")

}

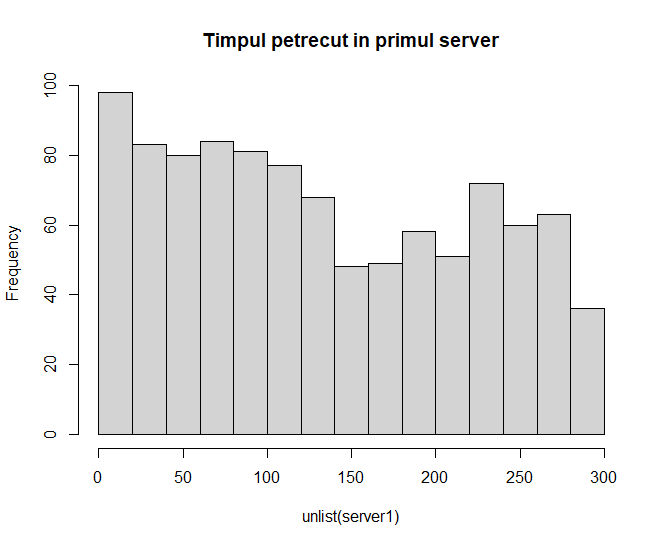
Datele obtinute sunt urmatoarele:



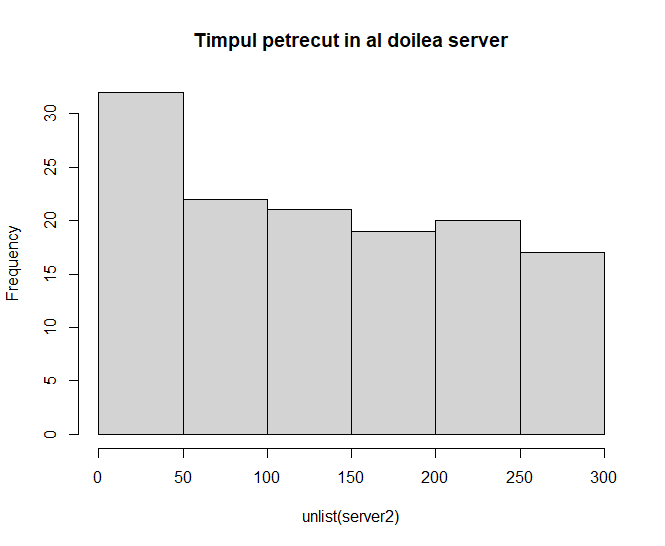
Timpul petrecut de fiecare client la spalatorie.



Timpul pentru serverul 1.



Timpul din cadrul celui de al doilea server.



In concluzie, clientii sunt serviti in majoritate de primul server, nici un client nu a parasite coada.