

Problema de sistema de colas

Sistema de atención de clientes en un banco

Franco Valencia
Moisés Montaña

Descripción

El programa realiza una simulación de la atención de clientes en un banco, donde llegan clientes en un tiempo aleatorio, y si no hay cajas disponibles tienen que esperar a ser atendidos. El programa toma en cuenta las siguientes características:

Se considera una sola fila de espera y s cajas (servidores) dentro del sistema. El número de cajas lo recibe como entrada el programa.

Los tiempos de llegada y de atención siguen la siguiente fórmula, donde R es un número aleatorio entre 0 y 1, y se piden dos factores de λ diferentes, uno para el tiempo de llegada y otro para el de atención. Estos valores de λ no son λ y μ dentro de las fórmulas de desempeño, simplemente son factores para los tiempos. El valor de λ y μ para las fórmulas de desempeño se va a calcular después de haber terminado la simulación, para tener los datos reales y calcular las medidas de desempeño de una forma más exacta.

Fórmula de distribución exponencial negativa:

$$x = -\lambda \ln(R)$$

Metodología

El programa fue implementado en python 3, con un enfoque de programación orientada a objetos, donde se tienen las clases Cliente, Caja y Banco.

Primero se obtiene, por input del usuario, el número total de clientes que van a llegar al banco, el número de cajas (o servidores), el tiempo de llegada aproximado por cliente y el tiempo aproximado que tarda una caja en atender a un cliente (tiempo de servicio).

El tiempo está en segundos, sin embargo, para fines de la simulación, se interpreta que cada segundo es en realidad un minuto. Así se pueden calcular las tasas de llegada y servicio (λ y μ) por hora con mayor facilidad.

Posteriormente comienza la simulación, donde se va imprimiendo el tiempo, el número de clientes en la fila, el número de clientes que ya fueron atendidos o que están siendo atendidos, el número total de cajas y el número de cajas disponibles. Esto se repite hasta que se atendió el número de clientes que el usuario ingresó.

Finalmente, se calculan las medidas de desempeño con el λ y μ obtenidos después de la simulación.

Resultados

Ejemplo 1:

```
→ quantitative-methods git:(master) X python3 bank.py
Enter the number of clients: >> 10
Enter the number of servers: >> 2
Enter aproximated arrival time: >> 5
Enter aproximated service time: >> 10
```

```
Time: 103
Clients waiting: 0
Clients served: 10
Number of counters: 2
Free counters: 2
-- Events --
Obtained WQ: 4.725801277160644
Obtained WS: 5.726913905143737
Lambda 7.792207792207792
Mu 15.379990689182387
Rho 0.25332290342959995
LQ 0.034742219552850416
WQ 0.004458584842615803
WS 0.06947813005621312
LS 0.5413880264120503
→ quantitative-methods git:(master) X
```

Ejemplo 2:

```
→ quantitative-methods git:(master) X python3 bank.py
Enter the number of clients: >> 15
Enter the number of servers: >> 3
Enter aproximated arrival time: >> 10
Enter aproximated service time: >> 8
```

```
Time: 140
Clients waiting: 0
Clients served: 15
Number of counters: 3
Free counters: 3
-- Events --
Obtained WQ: 0.00014766057332356772
Obtained WS: 1.0012272198994954
Lambda 6.7164179104477615
Mu 60.285401623237334
Rho 0.03713678452606168
LQ 8.258757404447609e-06
WQ 1.2296372135510884e-06
WS 0.016588993392187768
LS 0.11141861233558949
→ quantitative-methods git:(master) X
```