

Informe de Simulación

Kevin Majim Ortega Alvarez Lidier Robaina Caraballo

February 28, 2024

1 Introducción

1.1 Breve descripción del proyecto

Se desea crear una simulación que simule un proceso de atención a varios clientes donde los servidores se encuentran en serie, o sea un cliente para salir del sistema debe ser atendido por los n servidores. Cada servidor puede atender a un cliente o puede enviar a dicho cliente hacia otro servidor

1.2 Objetivos y Metas

Con dicha simulación nos hemos propuesto responder las siguientes interrogantes:

1. ¿Cuánto tiempo le toma a un cliente completar su recorrido por los n servidores? ¿Qué pasa si añadimos más servidores o aumentamos el tiempo que demora en cerrar el servicio?
2. ¿Cuál es la media de salto para cada servidor? ¿Qué servidor envía (No completa la solicitud del cliente si no que lo envía a otro servidor) más personas a otros? ¿Si modificamos parámetros distintos a la probabilidad de salto aumenta o disminuye?
3. ¿Cuál es el tiempo promedio que demora un sistema en atender a todos los clientes? ¿Como se refleja el cambio si alteramos los parámetros?

Para dar respuestas a las anteriores preguntas tenemos como metas:

1. Crear la simulación
2. Realizar el análisis estadístico de los resultados obtenidos de la simulación
3. Dar respuestas a nuestras interrogantes
4. Plantearnos nuevas interrogantes

1.3 El sistema

Los clientes llegan a un sistema que tiene n servidores, y las llegadas distribuye M . Cada cliente que llega debe ser atendido primero por el servidor 1 y, al completar el servicio en el servidor 1, el cliente pasa al servidor 2.

Cuando un cliente llega, entra en servicio con el servidor 1 si ese servidor está libre, o se une a la cola del servidor 1 en caso contrario. De manera similar, cuando el cliente completa el servicio en el servidor 1, entra en servicio con el servidor 2 si ese servidor está libre, o se une a su cola y así sucesivamente. Después de ser atendido en el servidor n , el cliente abandona el sistema.

Los tiempos de servicio en el servidor i tienen la distribución G_i

Además del servidor i se tiene una probabilidad p de saltar al servidor j

1.4 Variables que describen el sistema

1. Variable de tiempo: **time**
2. Variables contadoras:
 - a **arrival number**: Cantidad de arribos en el tiempo
 - b **out number**: Cantidad de salidas en el tiempo
 - c **jump forward**: Cantidad de saltos hacia servidores más adelantados
 - d **jump back**: Cantidad de saltos hacia servidores más atrasados
3. Variables de estado del sistema
 - a **serve's queue**: Lista que indica la cantidad de clientes en el servidor i
4. Variables de salida
 - a **arrival**: Lista con los diccionarios que representan el tiempo del cliente n en el servidor i (donde i es el índice del servidor en la lista)
 - b **wait time**: Tiempo que le tomo al cliente n salir del sistema (ser atendido por cada servidor)
 - c **jump number**: Cantidad de saltos que se han producido desde el servidor i

2 Detalles de Implementacion

El código está compuesto por dos archivos .py (Sim.py y main.py) donde en main se crean los parametros que se le introducirán a Sim. El archivo Extract.Samples.py se encarga de la extracción de los datos mas relevantes de la simulación y de guardarlos cada uno en documentos de excel. Además tenemos un archivo Statistic.py, el cual realiza los calculos estadísticos de los resultados que se encuentran en los excel.

La función **Simulation** recibe como entrada:

1. **n**: Cantidad de servidores $\rightarrow \text{Int}$
2. **T**: Tiempo que se demora en cerrar la tienda $\rightarrow \text{Int}$
3. **p**: Probabilidad con la que se desea que un servidor salte $\rightarrow \text{Float}$

La probabilidad con la que un cliente sea enviado a otro servidor se genera a partir de una variable (change_variable) $CH \sim \mathcal{U}(0, 1)$, donde el parametro p representa que: para un $change_variable \leq p$ el servidor atiende a su cliente y no lo envia hacia otro servidor en caso contrario, a continuacion tenemos el codigo:

```
if(change_variable>p):
    jump_number[serve]+=1
while True:
    change_variable=np.random.uniform(0,n)
    chan_serve=int(change_variable)
    if chan_serve==serve: continue
```

Donde se vuelve a generar la variable aleatoria CH , lo que ahora $CH \sim \mathcal{U}(0, n)$ para de esta forma convertirlo a entero y ese sera el servidor al que se envíe el cliente. Si se envia a el mismo entonces se genera nuevamente.

3 Resultados y Experimentos

3.1 Hallazgos de la simulación

En la simulación se recogieron algunas variables de especial interés para dar respuestas a nuestras objetivos. Variables tales como:

1. El tiempo que total que demora el modelo
2. Los tiempos que demoran cada cliente en recorrer los n servidores
3. La cantidad de saltos realizados por cada servidor
4. El tiempo en que un cliente es atendido por un servidor

También se extrajeron otras variables como: La cantidad de envíos a servidores más adelante y a servidores que se encuentran más atrasados dado que podemos preguntarnos tambien hacia donde van dirigidos los saltos con mayor frecuencia, y dado este caso podemos alterar el flujo para modelar situaciones más específicas. Una de las nuevas interrogantes sería: ¿Cuánto en promedio demora un servidor en atender a un cliente? Para responder está pregunta se extrajo también el tiempo en que cada cliente llega a un nuevo servidor, dado que, de esta manera se puede observar cual es el tiempo promedio que demora cada servidor en atender un cliente. Los que nos lleva a plantearnos preguntas

como, ¿Qué servidor es más rápido?. Una interrogante que surge, casi al mismo tiempo con nuestro objetivo dos, es ¿hacia que servidores se envían más personas?.

Los datos recogidos en el modelo se presentan en tablas de trabajo de excel, a partir de las cuales se realizan los análisis estadísticos (tales como calculo de la media y de la desviacion muestral) los cuales se guardan en bloc de notas con el mismo nombre.

Tabla de medias para $n = 15, T = 20, p = 0.7$	
Tiempo	105.2270507064013
Tiempo de salida de un cliente	47.20446870368254
Salto para el servidor n_1	8.786666666666667
Salto para el servidor n_2	8.653333333333334
Salto para el servidor n_3	8.68
Salto para el servidor n_4	8.653333333333334
Salto para el servidor n_5	8.706666666666667
Salto para el servidor n_6	8.966666666666667
Salto para el servidor n_7	9.046666666666667
Salto para el servidor n_8	8.826666666666666
Salto para el servidor n_9	8.813333333333333
Salto para el servidor n_{10}	8.673333333333334
Salto para el servidor n_{11}	8.933333333333334
Salto para el servidor n_{12}	8.646666666666667
Salto para el servidor n_{13}	8.573333333333334
Salto para el servidor n_{14}	8.626666666666667
Salto para el servidor n_{15}	8.613333333333333
Llegada al servidor n_1	12.942044945185726
Llegada al servidor n_2	20.27143219195456
Llegada al servidor n_3	25.2337404347859
Llegada al servidor n_4	29.421793535758418
Llegada al servidor n_5	32.97898342645674
Llegada al servidor n_6	35.50139337805129
Llegada al servidor n_7	38.21646857562005
Llegada al servidor n_8	39.733362030241985
Llegada al servidor n_9	41.25855021149807
Llegada al servidor n_{10}	42.23457026993232
Llegada al servidor n_{11}	42.78279911894122
Llegada al servidor n_{12}	43.219789484895315
Llegada al servidor n_{13}	44.39793086863655
Llegada al servidor n_{14}	44.69844261271796
Llegada al servidor n_{15}	45.51452072558415

Table 1: Tabla 1

Tabla de medias para $n = 3, T = 20, p = 0.7$	
Tiempo	43.80582345727979
Tiempo de salida de un cliente	25.482363177827487
Saltos para el servidor n_1	8.746666666666666
Saltos para el servidor n_2	8.793333333333333
Saltos para el servidor n_3	8.626666666666667
Llegada al servidor n_1	10.97679634360336
Llegada al servidor n_2	17.74840758330131
Llegada al servidor n_3	21.748768873390606

Table 2: Tabla 2

Tabla de medias para $n = 3, T = 20, p = 0.8$	
Tiempo	38.82202390072639
Tiempo de salida de un cliente	22.842904577624978
Saltos para el servidor n_1	4.9
Saltos para el servidor n_2	5.1
Saltos para el servidor n_3	5.373333333333333
Llegada al servidor n_1	11.012556104725936
Llegada al servidor n_2	16.203202188477857
Llegada al servidor n_3	19.73297639825384

Table 3:

3.2 Interpretación de los resultados

Como se evidencian en los datos presentados en las tablas en la sección anterior, podemos responder nuestras interrogantes:

1. Con esos datos se deduce que al alterar los parámetros de la simulación como el tiempo o la cantidad de servidores puede aumentar o disminuir el tiempo promedio que le toma a un cliente completar su recorrido. Lo cual tiene mucho sentido dado que un cliente debe pasar por cada vez mas o menos servidores, además con el aumento del tiempo de cerrada del servicio, aumentan la cantidad de clientes en el sistema.
2. En principio podemos decir que la media de salto para cada servidor es muy similar, dado que todos distribuyen uniforme, luego si variamos la el tiempo de cierre o la probabilidad del salto, esta media puede modificar su valor, permitiendo que a valores de p más altos, la media disminuya por la manera en que se encuentra implementado el modelo. En cambio si aumentamos el tiempo de cierre (T) aumenta el valor de la media, dado que aumentan la cantidad de clientes en el sistema, por lo tanto los servidores tienen mas clientes a los cuales enviar a otros. ¿Que pasa si aumentamos la cantidad de servidores y mantenemos invariante la probabilidad y el tiempo? En principio una variación en el número de servidores no afecta la cantidad de saltos que estos puedan tener.

Tabla de medias para $n = 3, T = 50, p = 0.8$	
Tiempo	81.49546116108615
Tiempo de salida de un cliente	45.00335056529862
Salto para el servidor n_1	12.466666666666667
Salto para el servidor n_2	12.293333333333333
Salto para el servidor n_3	12.493333333333334
Llegada al servidor n_1	25.46947111591416
Llegada al servidor n_2	34.70330388589315
Llegada al servidor n_3	40.38601898698339

Table 4:

3. Para responder esta pregunta podemos empezar por analizar que sucede cuando se varían los tres parámetros de entrada. Como vemos en la tabla 1 y la tabla 4 comparando luego esta última con la 3, el tiempo que dura la simulación con 15 servidores es bastante elevado, mas elevado incluso que uno de 3 con menos de la mitad de tiempo de cierre y un valor mayor de p , lo cual tiene sentido, aunque no es sea un valor esperado a simple vista, y no para todo n mayor que otro esto se cumple. ¿Qué sucede entonces cuando tenemos la misma cantidad de servidores y varían el resto de parámetros? en el caso de las tablas 4 y 3 como se evidencia, al crecer el tiempo en que se puede recibir clientes, aumenta el tiempo que demora en terminar la simulación. ¿que pasa con p ? bueno al analizar los resultados de las tablas 3 y 2, se puede observar que el tiempo de la simulación disminuye en el caso que aumenta este parámetro, dado que mientras mayor sea este menos probabilidad hay de que algún cliente salte a otros servidores, dado que con el salto se pueden atrasar en su carrera por pasar por cada servidor

Al momento de responder esta última pregunta surge otra: ¿Acaso si todos los servidores solo envían clientes hacia servidores mas adelantados, un parametro p más pequeño, puede disminuir el tiempo que demora el modelo? En un inicio la respuesta sería si, pero habría que probarlo en la simulación.

3.3 Análisis estadístico

Para tener una perspectiva general de los datos que hemos obtenido, buscamos estimar la media poblacional de las variables descritas anteriormente, para ello, dado que contamos con las variables de las muestras, decidimos estimar la media poblacional de cada una de ellas a través de la media muestral. Pero para lograr esto necesitamos calcular la desviación muestral, lo cual es fácil conociendo la media muestral, dado que esta se escribe como $S = \sum_{i=1}^n \frac{(\bar{x} - x)^2}{n-1}$ luego conociendo esta desviación podemos estimar la desviación poblacional.

3.4 Aálisis de parada

El caso de parada ideal sería cuando $d > \frac{S}{\sqrt{k}}$, donde k debe ser mínimo 100 y un valor no tan elevado, dado que se puede correr riesgo de sobreajuste lo cual significa que los modelos estadísticos pueden capturar ruido en lugar de relaciones verdaderas de los datos lo que puede llevar a conclusiones erróneas, también se puede interpretar mal los datos. en nuestro caso se generan 150 muestras (o simulaciones) las cuales son mas que 100 y no las suficientes coomo para que se corra el riesgo de caer en los problemas mencionados anteriormente