Práctica 3 SS: Modelos de Simulación Dinámicos y Discretos

José Manuel Pérez Lendínez, 26051613-l

December 24, 2019

Contents

1 Método de incremento fijo de tiempo.

En este caso vamos a realizar un programa que simule la carga de trabajo de un servidor. Los clientes llegan al servidor y esperan en una cola fifo hasta que este quede libre para ser utilizado. Tanto el momento en el que llegan los clientes como el tiempo que necesitan utilizar el servidor vienen dado por dos generadores de variables aleatorias independientes e idénticamente distribuidas. Estos generadores nos devolverán un valor que determinara un momento de tiempo que pasara hasta la llegada de un nuevo cliente o el tiempo de utilización del servidor. De esta manera le sumaremos este valor el tiempo actual de nuestro simulador para obtener los datos que nos interesa.

El programa recibe 4 valores como entrada con el siguiente formado.

./simulador TiempoDeLlegada N°Clientes TiempoDeServicio N°DeRepeticiones

Las variables indicaran lo siguiente:

1. **Tiempo de llegada (tlleg)**: Sera el valor utilizado para generar los tiempos de llegadas al servidor. De forma que si le pasamos un 1 equivaldría a 60 minutos y en caso de pasarle por ejemplo 0.15 serian 9 minutos.

2. **N° de clientes**: El programa para la ejecucion cuando atienda a un numero exacto de clientes, que indicaremos con este parámetro.

3. **Tiempo de servicio(tserv)**: Utiliza los mismos tipos de valores que la variable para tiempos de llegada. Pero en este caso indicaremos el tiempo que un cliente utiliza el servidor.

4. **N° de repeticiones**: La simulación la repetiremos un cierto numero de veces para obtener la media de los resultados obtenidos. Ese numero de simulaciones sera indicado por este parámetro.

El simulador nos devolverá una media de tiempo ocioso del servidor y la media de clientes en cola para utilizar el servidor.

El tiempo en la simulación incrementara de forma fija en una unidad. Vamos a realizar realizar varias mediciones y mostrarlas en la siguiente tabla para ver como se comporta el programa para valores distintos en los parámetros de tiempo de servicio y llegada.

Para estos datos he utilizada 10000 clientes y lo he repetido 50 veces para sacar las medias. Para la tabla vamos a utilizar las medidas de las medidas nombradas en la practica horas, medias horas, minutos, segundos,decimas de segundos y milisegundos.

Para trabar con horas tlleg = 0.15 y tserv = 0.1, si se quiera por ejemplo trabajar con medias horas solo tendríamos que multiplicar estos por 2, tlleg = 0.30 y tserv = 0.2. Otro ejemplo seria trabajar con segundos que tendríamos que multiplicar 0,15 y 0.1 por 3600.

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	0.0233562	0.135782	0.00116736
0.30	0.2	0.215516	2.98828	0.00159546
9	6	1.26906	31.3571	0.00200088
540	360	1.34331	33.246	0.0173753
5400	3600	1.30638	33.5554	0.150701
54000	36000	1.35409	33.237	1.37355

En la tabla se ve claramente como los valores para las media de cliente en cola y tiempo ocioso del servidor llega un momento en que prácticamente y pasan de ser muy bajas a ser siempre parecida. Esto se debe a que cunado se trabajan con unidades de tiempo altos como pueden ser las horas (tlleg = 0.15 y tserver = 0.1) o medias horas (tlleg = 0.3 y tserver = 0.2) los generadores obtienen valores muy bajos, siendo casi 0. Esto hace devuelva 1 casi siempre para el tiempo de servicio y llegada, los clientes estén llegando continuamente y el servidor este ocupado prácticamente siempre. En cambio al empezar a utilizar tiempos mas pequeños, a partir de minutos (tlleg = 9 y tserver = 6), si se tiene mas en cuenta la aleatoriedad y deja de devolver solo valores próximos a 0 consiguiendo ya resultados mas realistas.

Una vez visto esto y sabiendo que con medidas de tiempo mas pequeñas(tlleg y tserv mas grandes) analizaremos en estos el tiempo medio de cola y ocioso del servidor. Con esto vemos claramente como el servidor se mantiene ocioso prácticamente un 33% del tiempo y en cola tenemos una media de 1.3 clientes.

A la hora de analizar el tiempo se observa como va creciendo, esto se debe a que con unidades de tiempo mas pequeñas hace falta incrementar mas veces el reloj del simulador para llegar a un evento.

2 Método de incremento variable de tiempo.

El método de incremento variable se basa en el condigo anterior añadiendo dos cambios sencillos. Por esto la forma del llamar al pograma es la misma que en el caso anterior.

Los cambios principales se basa en que dejamos de trabajar con el tiempo en como entero y utilizaremos un float. Nuestro generadores anteriores tenían que redondear el float para devolver un numero entero, en este método esto no es necesario por lo que el generador devolverla el float sin la necesidad de redondear ni comprobar si el redondeo lo convirtió en un 0. El ultimo cambio

es que el reloj no aumenta de unidad en unidad, el reloj en este caso avanza hasta el evento mas cercano en el tiempo. En nuestro caso sera o la llegada de un nuevo cliente al servidor o que un cliente deje de utilizar el servido.

Vamos a ejecutar con los mismos datos que en el ejemplo anterior para comparar.

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	1.32145	33.396	0.0009926
0.30	0.2	1.34731	33.1729	0.00103374
9	6	1.33189	33.3837	0.00105938
540	360	1.36657	33.4256	0.00105644
5400	3600	1.33417	33.0666	0.0010442
54000	36000	1.32711	33.2303	0.00104676

Empecemos analizando el tiempo. En este caso el tiempo no aumenta solo una unidad, sino que salta al evento mas cercano. Esto hace que el tiempo de ejecucion no dependa de la distancia entre un evento y el estado actual, si no del numero de eventos que se den durante la ejecucion del simulador. Para este ejemplo he ejecutado para 10000 cliente, y cada cliente utiliza el servidor una vez. Solo tendremos 20000 eventos y seran lo que necesitemos procesar. Da igual la unidad que utilicemos, solo se tiene el cuenta la cantidad de evento que tendremos que ejecutar. Por tanto el tiempo medio de ejecucion como se ve en la tabla es el mismo para todos los ejemplos quitando las pequeñas sobrecargas que pudiera tener el procesador al ejecutarlos.

Si nos centramos en las medias de clientes en cola y tiempo ocioso del servidor, vemos que no ocurre lo mismo que con el incremento fijo. En este caso no pasamos de unas medias muy bajas a unas mas altas, sino que se mantienen las medias durante todas las ejecuciones en los mismos valores. Esto se debe a que en este caso como explique anteriormente al estar trabajando con float no tenemos el problema con unidades de tiempo grandes, como por ejemplo horas (tlleg = 0.15 y tserv = 0.1). Aunque obtengamos valores próximos a 0 con los generadores, nuestro simulador si puede trabajar con estos y no los redondea a 1 como en el apartado anterior. Con esto podemos asegurar que con el incremento variables, la unidad de tiempo que elijamos no nos afectara a la hora de obtener las medias.

Vamos a comparar nuestros resultados con los resultados obtenidos teóricamente. Cuando $n \to +\infty$, sabes teóricamente que:

$$\rho = \frac{tserv}{tIIeg},\ Q(n) \to \frac{\rho^2}{1-\rho},\ PTO(n) \to 100 * (1-\rho)$$

Vamos a sustituir y resolver con los valores para horas:

$$\rho = \frac{0.1}{0.15} = 0.667$$

$$Q(n) \to \frac{0.667^2}{1-0.667} = 1.334$$

$$PTO(n) \to 100 * (1-0.667) = 33,334$$

Con esto vemos como teóricamente tenemos resultados prácticamente iguales a los conseguidos con el incremento variable. En cambio si miramos los resultados que conseguimos con el incremento fijo:

T. de llegada (tlleg)	**T. servicio(tserv)**	**Media de clientes en cola**	**Media % de tiempo ocioso Servidor**	**T. medio de ejecución (seg)**
0.15	0.1	0.0233562	0.135782	0.00116736

Vemos que no se acercan ni lo mas mínimo a los resultados teóricos. Esto ya nos asegura todo los indicios anteriores de que el método fijo tiene problemas para conseguir resultados realistas con valores altos de tiempo.

3 Estructura de un programa de simulación dinámico y discreto.

3.1 Simulación con un único valor

En este apartado trabajaremos con un modelo mas avanzado del programa anterior. En este caso tendremos la opción de elegir el numero de servidores(parámetro m) que tendremos trabajando en paralelo para atender clientes. Para comprobar que nuestro modelo trabaja correctamente, utilizaremos el caso en m=1, un tlleg = 0.15 y tserv = 0.1. Para este modelo tendremos en cuenta las siguientes medidas:

1. **Tiempo medio de espera en cola:** $\frac{tserv^2}{tlleg-tserv} = 0.2$
2. **Tiempo medio de estancia en el sistema:** $\frac{tserv*tlleg}{tlleg-tserv} = 0.3$

3. **Número medio de clientes en cola:** $\frac{tserv^2}{tlleg*(tlleg-tserv)} = 1.333$

4. **Número medio de clientes en el sistema:** $\frac{tserv}{tlleg-tserv} = 2$

5. **Longitud media de colas no vacías:** $\frac{tlleg}{tlleg-tserv} = 3$

6. **Porcentaje de tiempo de ocio del servidor:** $(1-\frac{tserv}{tlleg})*100 = 33.333$

Vamos a mostrar ahora los datos obtenidos por el simulador para estos tlleg y tserv. El formato de la llamada es el siguiente:

$$./colammk < numeroServidores >< tiempoParada >< tlleg >< tserv >$$

Para los ejemplos elegiremos los siguientes parámetros utilizando varios tiempos de parada(100,1000,10000 y 100000).

$$./colammk\ 1 < tiempoParada > 0.15\ 0.1$$

T. Parada	100	1000	10000	100000
T.medio de espera en cola	0.286	0.206	0.196	0.200
T. medio de estancia en el sistema	0.386	0.306	0.296	0.300
Num. medio clientes en cola	1.939	1.353	1.305	1.335
Num. medio clientes en sistema	2.659	2.002	1.969	2.000
Long. media de colas no vaciás	3.750	3.175	2.976	3.018
% de tiempo de ocio del servidor	27.983	35.171	33.569	33.518

Si comparamos los valores obtenidos y los valores teóricos para nuestras salidas, vemos claramente como van acercándose todos los para metros a estos valores conforme subimos el tiempo de parada. Llegado a un tiempo de parada igual a 10^5, vemos como ya obtenemos parámetros prácticamente iguales a los teóricos. Esto nos dice que partir de 10^5 obtendremos valores muy aproximados a los teóricos y que nuestras simulaciones tendrían que acercarse a este numero para obtener resultados que podamos dar como buenos.

3.2 Simulación con varios servidores.

En este apartado vamos a analizar como afecta a nuestros resultados tener un mayor numero de servidores, aunque aumentaremos el tserv para los servidores. De esta forma intentaremos igualar las condiciones de un único servidor mas rápido y varios servidores mas lentos para ver como afecta el aumento en numero de servidores bajo condiciones parecidas. Para igualar el tiempo de servidor, si para un único servidor usamos tserv, para m servidores utilizamos un tiempo igual m*tserv. Realizaremos las pruebas con tiempo de parada 10^5 y aumentaremos el numero de servidores de 2 en 2.

Num. Servidores	1	3	5	7	9	11
tserv	0.1	0.3	0.5	0.7	0.9	1.1
T.medio de espera en cola	0.200	0.132	0.099	0.076	0.057	0.045
T. medio de estancia en el sistema	0.300	0.432	0.599	0.776	0.957	1.145
Num. medio clientes en cola	1.335	0.881	0.661	0.506	0.382	0.302
Num. medio clientes en sistema	2.000	2.881	3.991	5.182	6.382	7.644
Long. media de colas no vaciás	3.018	2.987	3.027	3.013	2.943	2.918
% de tiempo de ocio del servidor	33.518	33.341	33.392	33.206	33.334	33.261

Vamos a analizar estos datos detenidamente, para determinar el comportamiento del simulador y que el programa funciona correctamente.

1. **T. medio de espera en cola:** En este caso se ve claramente una mejora. A mas servidores menos esperaran los clientes en cola. Esto se debe a que aunque se suba el tiempo que tarde un servidor en ejecutar la tarea(tserv), tendremos mas servidores a los que dar tareas y al final en vez de estar en cola esperando, estarán en un servidor ejecutando.

2. **T. medio de estancia en el sistema:** Este parámetro es todo lo contrario al anterior. Si los clientes en vez de en cola están el el servidor y el tserv aumenta también, se tendrá un aumento en este parámetro cada vez que aumente el numero de servidor y el tserv.

3. **Num. medio de clientes en cola:**Este parámetro también cae conforme mas servidores tenemos. Esto se da por lo explicado anteriormente también. Tenemos mas servidores y los clientes que llegan es mas probable que se asignen directamente a un servidor o tengan pocos clientes en la cola antes que el.

4. **Numero medio de clientes en sistema:** En este caso al tener un aumento en el tserv este parámetro crecerá, estarán mas tiempo en el sistema al necesitar mas tiempo del servidor para ejecutar su tarea.

5. **Porcentaje de tiempo de ocio del servidor:** El porcentaje de ocio no varia aunque se aumente los servidores. Esto se da porque aunque aumentemos los servidores también se aumenta el tiempo de servicio, por tanto se mantiene el tiempo de ocio igual.

Como conclusión de aumentar lo servidores, tendremos una mejora para los clientes a la hora de esperar para poder realizar su servicio, aunque también tendremos un mayor tiempo de uso de los servidores y el gasto que esto puede conllevar, aunque esto se da porque se aumenta también el tserv. No obtendremos cambios en el tiempo de ocio del servidor, por lo que tendremos un 33% de tiempo que los servidores no estarán realizando tarea.

3.3 Cambios en el software.

3.3.1 Añadir medias y derivación típica.

Vamos a realizar un cambio en el simulador para realizar varias ejecuciones y obtener las medias y desviación típica para estas medias. Para analizar este cambio vamos a ver como afectan el numero de simulaciones probando 10,100 y 1000 simulaciones, para el programa con 1 servidor, tserv = 9, tlleg = 6 y un tiempo de parada de 10000. Cambio este tsev y tlleg a estos valores porque al trabajar con estos números, el tiempo de ejecución es mas rápidos y puedo realizar mas simulaciones.

Para la ejecucion solo tenemos que añadir un nuevo parámetro que nos da el numero de simulaciones:

$./colammk < numeroServidores >< tiempoParada >< tlleg >< tserv >< numRepeticiones >$

Vamos a mostrar los datos medidos.

Num. Simulaciones	100	1000	2000
T.medio de espera en cola	12.002 ± 2.473	11.862 ± 2.390	11.854 ± 2.475
T. medio de estancia en el sistema	1.002 ± 2.473	17.862 ± 2.390	17.854 ± 2.475
Num. medio clientes en cola	1.338 ± 0.286	1.325 ± 0.287	1.320 ± 0.295
Num. medio clientes en sistema	2.002 ± 1.529	1.991 ± 1.519	1.985 ± 1.516
Long. media de colas no vaciás	3.006 ± 0.486	2.962 ± 0.445	2.965 ± 0.467
% de tiempo de ocio del servidor	33.537 ± 2.686	33.321 ± 2.687	33.534 ± 2.801
Longitud de la máxima cola	12.705 ± 3.155	12.200 ± 2.486	12.991 ± 3.210

C