

SIMULACIÓN DE EVENTOS DISCRETOS N SERVIDORES EN PARALELO

CHAVELY GONZÁLEZ ACOSTA C412

Contents

1. INTRODUCCIÓN

(1) Breve descripción del proyecto

El sistema a simular en el presente proyecto consiste en: m clientes, los cuales llegan a un sistema que tiene n servidores, con tiempo entre los arribos que distribuye M . Cuando un cliente llega, se une a la cola si ambos servidores están ocupados. El caso de solo tener dos servidores se describe a continuación: Si el servidor 1 está libre, el cliente entra en servicio con el servidor 1. Si el servidor 1 está ocupado pero el servidor 2 está libre, el cliente entra en servicio con el servidor 2. Cuando un cliente completa el servicio con un servidor (no importa cuál), ese cliente luego abandona el sistema. El cliente que ha estado en la cola durante más tiempo (si hay clientes en la cola) entra en servicio. La distribución de servicio en el servidor i es G_i .

(2) Objetivos y metas

El objetivo fundamental de este proyecto es simular correctamente el sistema antes descrito y arribar a conclusiones.

(3) Variables que describen el problema

Las variables de interés para este proyecto incluyen el tiempo entre los arribos de los clientes, la distribución de los servidores ocupados y la duración del servicio en cada servidor.

2. DETALLES DE IMPLEMENTACIÓN

La implementación de la simulación de eventos discretos se basa en los siguientes pasos:

(1) Modelado del sistema

El sistema se compone de:

- Un conjunto de m clientes que arriban con una diferencia de tiempo que distribulle M y requieren algún servicio
- n servidores: $s_1, s_2, s_3, \dots, s_{n-1}, s_n$, los cuales trabajan en paralelo y el tiempo que tardan en completar un servicio distribulle G_i
- Una cola(FIFO) de clientes, inicialmente vacía, para mantener a los clientes en espera mientras están llenos todos los servidores hasta que se vacíe algún servidor
- En todo momento el servidor al cual se le asigna un servicio nuevo es, de los que no estén ocupados en ese momento, el de menor índice.

(2) Generación de eventos

Para generar las variables aleatorias se utilizan los métodos del módulo de python *numpy.random*. Las distribuciones propuestas son las siguientes: poisson, geométrica, uniforme, binomial, exponencial, normal, beta y ganma, con sus respectivos parámetros predefinidos por el propio programa. Al momento de relizar la simulación se decide cual(es) utilizar.

- Variables aleatorias: distribución de llegada de los clientes (M) y las distribuciones de servicio en los servidores (G_i)

(3) Procesamiento de eventos

Para simular los eventos: llegada de un cliente, agregar un cliente a la cola, sacar a un cliente de la cola, asignar un servidor para un servicio, ejecutar un servicio en un servidor y liberar un servidor al terminar de realizar un servicio; se utilizaron las siguientes clases:

- heap: una cola con prioridad, que permite eliminar($O(\log(n))$) e insertar($O(\log(n))$) elementos manteniendo siempre en el tope el elemento de menor valor($O(1)$). Esta estructura se utilizo para mantener el conjunto de servidores disponibles de forma tal que siempre que se pide(elimina) uno devuelve el de menor índice, y para mantener el conjunto de servidores ocupados, de forma tal que siempre que se pide(elimina) uno devuelve el que termina primero(cuyo tiempo de finalización es menor).
- queue: una cola que cumple con la invariante: el primer elemento que entra es el primero que sale(FIFO). Esta estructura se utiliza para mantener a los clientes en espera ordenados mientras no hay servidores vacíos.
- action: un objeto con 4 propiedades: momento, nombre, servidor y cliente, el cual permite almacenar la información sobre un evento dado.
- report: un objeto que permite almacenar toda la información que ocurre durante una simulación mediante una lista de accionesy luego procesarla.
- simulation: se encarga de realizar la simulación, recibe como parámetros las distribuciones que siguen los servidores y el tiempo entre la aparición de clientes. Para esto utiliza la siguiente lógica: se agregan todos los clientes de uno en uno, al momento de agregarlos primero se genera el tiempo que demora en aparecer el cliente, y se actualiza el curen time de la simulación, luego se ejecutan todos los servicios posibles hasta ese momento, es decir, todos los servidores que estuviesen ocupados en algún servicio cuyo tiempo de finalizacion fuese menor que el momento actual se eliminan del heap de los servidores ocupados y se agregan al heap de los servidores activos, a la vez si habían clientes en la cola se van procesando al momento en que se vacía algún servidor, finalmente cuando ya no hay más servidores que terminen antes del momento actual se procesa el nuevo cliente, es decir, se escoge el servidor con menor índice que se encuentre desocupado, se elimina de los servidores activos, se genera un tiempo

aleatorio que se demorará en realizar el servicio y se agrega a los servidores ocupados con un tiempo de finalización igual al momento actual más el tiempo generado, de no haber servidores activos se agrega el cliente a la cola. En todo momento se agrega la acción que se está realizando al *simulation.report*. Finalmente se procesan todos los clientes que no han terminado de ejecutar sus respectivos servicios o están en cola, se imprimen los resultados y se reinicia el estado de la simulación.

(4) **Simulación**

Ejecutar la simulación para un número determinado de eventos o hasta que se cumpla una condición de parada.

(5) **Análisis de resultados**

Recopilar y analizar los datos de la simulación para entender el comportamiento del sistema y extraer insights.

3. RESULTADOS Y EXPERIMENTOS

Los resultados de la simulación proporcionarán información valiosa sobre el comportamiento del sistema bajo condiciones específicas. Esto incluirá:

Hallazgos de la simulación: Observaciones sobre la utilización de los servidores, la duración de las colas y los tiempos de servicio.

Interpretación de los resultados: Análisis de cómo los resultados se relacionan con las variables de interés y las expectativas del modelo.

Análisis estadístico: Evaluación de las variables de interés mediante análisis estadístico para entender su comportamiento y variabilidad.

Análisis de parada de la simulación: Identificación de las condiciones que se utilizaron para detener la simulación y su impacto en los resultados.

4. MODELO MATEMÁTICO

(1) Descripción del modelo de simulación

La llegada de un cliente se simula tomando el momento de llegada del último cliente, se genera un entero que distribuya M , se actualiza el estado de la simulación hasta el momento de llegada del último cliente más el tiempo que tardó el cliente actual en llegar y si no hay servidores activos se agrega el nuevo cliente a la cola, si en cambio hay algún servidor activo se le asigna al cliente actual.

La asignación de un servidor a un cliente consiste en eliminar el servidor de los servidores activos, y agregarlo junto con el momento en el cual terminará de realizar el servicio al conjunto de los servidores ocupados. El momento en el cual terminará de realizar el servicio es: momento actual más número aleatorio generado según la distribución del servidor.

La salida de los clientes se realiza cuando se actualiza el estado de la simulación tras actualizar el momento actual de la misma. Se comprueba si el momento de terminar el servicio ya pasó y se elimina el servidor del conjunto de servidores ocupados y se agrega al conjunto de servidores activos.

(2) Supuestos y restricciones

Definición de los supuestos del modelo, como la independencia de los eventos de llegada y la asincronicidad de los servicios.