

SIMULACIÓN DE EVENTOS DISCRETOS N SERVIDORES EN PARALELO

CHAVELY GONZÁLEZ ACOSTA C412

CONTENTS

1. Introducción	2
2. Detalles de implementación	3
3. Resultados y experimentos	5
4. Modelo matemático	12

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 como el tiempo promedio que pasa un cliente en la cola, el porcentaje de clientes que tienen que esperar para ser atendidos y el porcentaje de ocupación de cada servidor.

(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 e insertar elementos manteniendo siempre en el tope el elemento de menor valor. 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 indice, 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.
- simulation_report: un objeto que permite almacenar toda la información que ocurre durante una simulación mediante una lista de acciones y luego procesarla.
- report: objeto q permite almacenar la información sobre un conjunto de simulaciones y arribar a conclusiones al respecto.
- 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 curent 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 (comenzar, terminar, encolar, desencolar) 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 y se reinicia el estado de la simulación.

(4) **Simulación**

Ejecución de la simulación para distintas cantidades de clientes y servidores, se ejecutó 100 veces por cada par n, m (n = cantidad de clientes, m = cantidad de servidores), por cada uno se presenta un solo reporte con el contenido general de las 100 simulaciones:

3. RESULTADOS Y EXPERIMENTOS

La simulación se realizó solo con la distribución uniforme para las variables aleatorias.

Los resultados de la simulación proporcionaron la siguiente información:

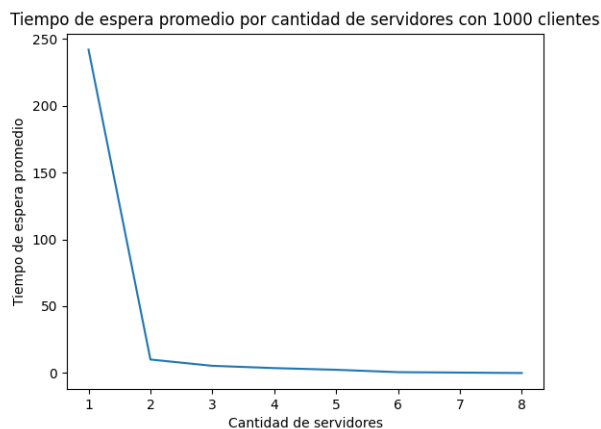
- **Utilización de los servidores**

A partir del sexto o séptimo servidor no se utiliza ningún otro servidor aún cuando la cantidad de clientes aumenta, por tanto tampoco se encola ningún cliente cuando se tienen más de seis servidores. Por tanto podemos concluir que son necesarios solo 6 o 7 servidores para minimizar la cantidad de clientes en cola y la cantidad de servidores disponibles.

- **Duración de las colas**

Con uno o dos servidores la mayoría de los clientes tienden a entrar a la cola, por tanto no es buena opción tener tan pocos servidores, a partir de los tres servidores el tiempo que pasan los clientes en la cola empieza a disminuir drásticamente aún para grandes cantidades de clientes. Finalmente para seis o siete servidores prácticamente ningún cliente presenta necesidad de entrar en la cola, y ya de ocho servidores en adelante no solo no se entra ningún cliente a la cola sino que tampoco se usan los últimos servidores.

– Cantidad de clientes: 1000



```
chavely@chavely-X550LD: ~/Documents/start up/cs/simulacion/simulation-project$ python
```

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 1

Porcentaje de ocupacion de los servidores:
[100.0]

Promedio de tiempo pasado por los clientes en cola: 241.99

Promedio de clientes ecolados por simulacion: 963.8

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 2

Porcentaje de ocupacion de los servidores:
[59.16, 40.84]

Promedio de tiempo pasado por los clientes en cola: 10.1

Promedio de clientes ecolados por simulacion: 316.98

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 3

Porcentaje de ocupacion de los servidores:
[52.53, 32.02, 15.45]

Promedio de tiempo pasado por los clientes en cola: 5.38

Promedio de clientes ecolados por simulacion: 83.65

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 4

Porcentaje de ocupacion de los servidores:
[51.5, 30.08, 13.62, 4.8]

Promedio de tiempo pasado por los clientes en cola: 3.68

Promedio de clientes ecolados por simulacion: 19.1

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 5

Porcentaje de ocupacion de los servidores:
[51.44, 29.88, 13.0, 4.47, 1.21]

Promedio de tiempo pasado por los clientes en cola: 2.42

Promedio de clientes ecolados por simulacion: 3.49

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 6

Porcentaje de ocupacion de los servidores:
[51.0, 30.04, 13.34, 4.35, 1.07, 0.21]

Promedio de tiempo pasado por los clientes en cola: 0.64

Promedio de clientes ecolados por simulacion: 0.38

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 7

Porcentaje de ocupacion de los servidores:
[51.18, 29.77, 13.28, 4.35, 1.14, 0.26, 0.03]

Promedio de tiempo pasado por los clientes en cola: 0.32

Promedio de clientes ecolados por simulacion: 0.15

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 8

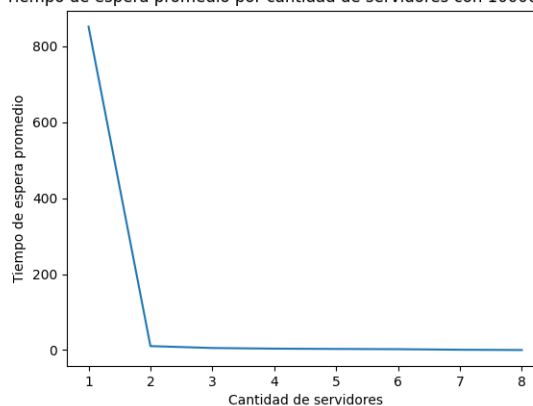
Porcentaje de ocupacion de los servidores:
[51.32, 29.92, 13.08, 4.28, 1.15, 0.22, 0.03, 0.0]

Promedio de tiempo pasado por los clientes en cola: 0.0

Promedio de clientes ecolados por simulacion: 0.0

– Cantidad de clientes: 10000

Tiempo de espera promedio por cantidad de servidores con 10000 clientes



Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 1

Porcentaje de ocupacion de los servidores:
[100.0]

Promedio de tiempo pasado por los clientes en cola: 851.67

Promedio de clientes ecolados por simulacion: 9889.86

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 2

Porcentaje de ocupacion de los servidores:
[58.96, 41.04]

Promedio de tiempo pasado por los clientes en cola: 10.52

Promedio de clientes ecolados por simulacion: 3179.67

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 3

Porcentaje de ocupacion de los servidores:
[52.57, 31.86, 15.57]

Promedio de tiempo pasado por los clientes en cola: 5.51

Promedio de clientes ecolados por simulacion: 826.97

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 4

Porcentaje de ocupacion de los servidores:
[51.42, 30.23, 13.59, 4.76]

Promedio de tiempo pasado por los clientes en cola: 3.81

Promedio de clientes ecolados por simulacion: 179.57

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 5

Porcentaje de ocupacion de los servidores:
[51.31, 29.91, 13.19, 4.42, 1.17]

Promedio de tiempo pasado por los clientes en cola: 3.03

Promedio de clientes ecolados por simulacion: 33.97

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 6

Porcentaje de ocupacion de los servidores:
[51.2, 29.93, 13.2, 4.33, 1.11, 0.23]

Promedio de tiempo pasado por los clientes en cola: 2.41

Promedio de clientes ecolados por simulacion: 5.16

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 7

Porcentaje de ocupacion de los servidores:
[51.21, 29.91, 13.2, 4.34, 1.08, 0.22, 0.04]

Promedio de tiempo pasado por los clientes en cola: 0.92

Promedio de clientes ecolados por simulacion: 0.79

Reporte de las simulaciones:

Cantidad de simulaciones: 100. Cantidad de servidores: 8

Porcentaje de ocupacion de los servidores:

[51.18, 29.97, 13.16, 4.33, 1.09, 0.21, 0.04, 0.01]

Promedio de tiempo pasado por los clientes en cola: 0.2

Promedio de clientes ecolados por simulacion: 0.15

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

Los eventos de llegada se simulan como una variable aleatoria completamente independiente para cada cliente que arriba.

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

La simulación de eventos discretos se refiere a la técnica de modelar y analizar sistemas complejos mediante la simulación de eventos individuales que ocurren en el tiempo. En el contexto de un sistema con (n) servidores en paralelo y una cola para los clientes, esta técnica permite estudiar el flujo de trabajo, la eficiencia del sistema y el comportamiento bajo diferentes condiciones.

(3) Componentes del Modelo

- **Servidores:** (n) servidores operan en paralelo. Cada servidor puede atender a un cliente a la vez. Los servidores pueden tener diferentes tiempos de servicio.
- **Cola de Clientes:** Los clientes llegan al sistema y esperan en una cola para ser atendidos. La cola tiene una capacidad máxima, y si la cola está llena, los clientes adicionales se rechazan o deben esperar hasta que haya espacio disponible.

(4) Modelado Matemático

El modelo matemático para este sistema se basa en la simulación de eventos discretos, donde cada evento (llegada de un cliente, servicio de un cliente, liberación de un servidor) se modela como un evento discreto en el tiempo.

Eventos de Llegada: Los clientes llegan al sistema con una distribución de probabilidad exponencial, indicando el tiempo entre llegadas sucesivas.

Eventos de Servicio: Cuando un servidor está libre, puede comenzar a atender a un cliente. El tiempo de servicio se modela de manera similar a los tiempos de llegada, utilizando una distribución exponencial.

Eventos de Liberación: Cuando un servidor ha terminado de atender a un cliente, se libera para atender al siguiente cliente en la cola o para esperar a que llegue un nuevo cliente.

(5) Algoritmo de Simulación

La simulación se realiza mediante un algoritmo que mantiene un registro de los eventos futuros y actualiza el estado del sistema (número de clientes en la cola, servidores ocupados, etc.) según ocurran estos eventos. Aquí hay un esquema básico del algoritmo:

Inicialización: Se establece el estado inicial del sistema, incluyendo el tiempo actual(0) y el número de servidores libres(inicialmente todos).

Llegada de un nuevo cliente: Se genera el momento de llegada del cliente y se actualiza el estado de la simulación, luego se simula la atención a ese cliente. Actualización del estado de la simulación: Dado un tiempo se actualiza el estado de la simulación sacando a todos los clientes que terminaban de recibir su servicio antes del momento actual y a la vez se van procesando los clientes que estuviesen en la cola. Repetición: Se repite el proceso para el siguiente cliente hasta que se terminen de procesar todos.

(6) **Ventajas y Aplicaciones**

Este modelo matemático es útil para analizar y optimizar sistemas de colas y servidores, como líneas de montaje, centros de llamadas, y sistemas de procesamiento de transacciones. Permite evaluar el rendimiento del sistema bajo diferentes condiciones y entender cómo los cambios en el sistema (por ejemplo, aumentar el número de servidores o modificar los tiempos de servicio) afectan la eficiencia y el tiempo de espera de los clientes.

La simulación de eventos discretos es una herramienta poderosa para el análisis de sistemas complejos, proporcionando una visión detallada del comportamiento del sistema en tiempo real y permitiendo experimentar con diferentes políticas y configuraciones sin el costo y riesgo asociado con la implementación en el mundo real.