

Simulación de Servidores en Serie

Ovidio Navarro and Juan José Muñoz

Abstract—

Keywords—

1. Introduction

Este proyecto tiene como objetivo desarrollar una simulación de eventos discretos para analizar y entender mejor ciertos fenómenos. A través de este trabajo, buscamos aplicar los principios de la simulación de eventos discretos para modelar y experimentar con estos fenómenos, y obtener resultados que nos ayuden a tomar decisiones informadas. En este proyecto se propone realizar una simulación de n servidores en serie. Este sistema modela una situación donde los clientes deben pasar secuencialmente a través de múltiples servidores para completar un proceso o recibir un servicio. En este informe, exploraremos un sistema con n servidores en serie, donde los clientes llegan al sistema y son atendidos por cada servidor en orden. Con la peculiaridad de que cuando un cliente es atendido, tiene una cierta probabilidad de saltar a la cola de otro servidor y no precisamente al siguiente.

1.1. Objetivos y Metas

Recopilar datos importantes como son:

- El tiempo extra que demora en terminar la simulación luego que termina el tiempo asignado
- Cantidad de clientes en la simulación
- Saltos de cola y saltos por cada cliente

Preguntarnos que pasaria si cambiamos los tiempos y los parámetros de la simulación Entre otros muchos datos que serían de interés investigar

1.2. Variables que describen el problema

Existen 3 tipos de variables en la simulación de Eventos Discretos, variable de Tiempo, Estado y Contadoras

- Variable de Tiempo: *waitingtime* representa el tiempo de espera que le toma al servidor atender al cliente actual; *time* representa el tiempo real transcurrido
- Variable de Estado: *clientqueue* representa la cola de cada servidor; *arrivals* representa el tiempo en el cual llegó cada cliente
- Variable Contadora: *clientout* cuenta la cantidad de clientes que terminan el proceso; *clientjumps* que determina la cantidad de saltos que dio un cliente

2. Detalles de Implementación

Este simulador recibe la cantidad de servidores, el tiempo que se desea que dure el servicio y un umbral que es un número entre 0-1 que dicta la probabilidad de que ocurra un salto entre servidores.

Luego todas las variables contadoras se inicializan en 0 y se crean otras estructuras de datos para guardar la información necesaria.

Toda la simulación ocurre dentro de un *whiletrue* donde existen 4 casos

- Caso 1: $if(arrival_{time} < min_{wait} and arrival_{time} < time_{sim})$: precisa si el tiempo de llegada de un cliente es menor que el minimo tiempo de espera de los servidores y ademas esta todavia en tiempo la simulacion, indicando que puede procesarse a un cliente nuevo
- Caso 2 $if(min_{wait} < arrival_{time} and min_{wait} < time_{sim})$ verificar si la espera minima es menor que el tiempo de llegada y que el tiempo de la simulacion para agregar al cliente en la cola
- Caso 3 $if(arrival_{num} - client_{out} == 0 and min(min_{wait}, arrival_{time}) > time_{sim})$ Comprueba todos los clientes que llagaron hallan salido del proceso
- Caso 4 $if(arrival_{num} - client_{out} == 0 and min(min_{wait}, arrival_{time}) > time_{sim})$ Este caso comprueba que se hallan atendido a todos los clientes y termina el ciclo *while*. De esta forma se garantiza que todos sean atendidos incluso luego que termine el tiempo de vida de los servidores.

Para verificar si luego de ser atendido un cliente este pueda hacer un salto de servidor se utiliza una variable aleatoria con distribucion uniforme entre [0-1] y que si esta es mayor que el umbral el cliente salta a otro servidor generado de igual forma con una variable aleatoria uniforme entre [0-n] con lo cual el cliente tambien tendria la posibilidad de saltar a un servidor mas atrasado respecto al actual.

3. Resultados y Experimentos

3.1. Hallazgos de la simulación

Se puede observar que a medida que se agrega mas tiempo de vida a los servidores aumenta drasticamente la cantidad de clientes que pasan por el sistema. Lo mismo sucede con el tiempo extra y con las variables de salto, en este ultimo caso dependiendo bastante del valor que tome el umbral.

3.2. Análisis de parada de la simulación

La simulacion se detiene una vez halla terminado el tiempo de vida de los servidores y ademas se hallan procesado todos los clientes.