

Simulación

Proyecto 1

Daniel Reynel Domínguez Ceballos
C411

Orden del Problema

La Cocina de Kojo (Kojo's Kitchen) La cocina de Kojo es uno de los puestos de comida rápida en un centro comercial. El centro comercial está abierto entre las 10:00 am y las 9:00 pm cada día. En este lugar se sirven dos tipos de productos: sándwiches y sushi. Para los objetivos de este proyecto se asumirá que existen solo dos tipos de consumidores: unos consumen solo sándwiches y los otros consumen solo productos de la gama del sushi. En Kojo hay dos períodos de hora pico durante un día de trabajo; uno entre las 11:30 am y la 1:30 pm, y el otro entre las 5:00 pm y las 7:00 pm. El intervalo de tiempo entre el arribo de un consumidor y el de otro no es homogéneo pero, por conveniencia, se asumirá que es homogéneo. El intervalo de tiempo de los segmentos homogéneos, distribuye de forma exponencial. Actualmente dos empleados trabajan todo el día preparando sándwiches y sushi para los consumidores. El tiempo de preparación depende del producto en cuestión. Estos distribuyen de forma uniforme, en un rango de 3 a 5 minutos para la preparación de sándwiches y entre 5 y 8 minutos para la preparación de sushi. El administrador de Kojo está muy feliz con el negocio, pero ha estado recibiendo quejas de los consumidores por la demora de sus peticiones. Él está interesado en explorar algunas opciones de distribución del personal para reducir el número de quejas. Su interés está centrado en comparar la situación actual con una opción alternativa donde se emplea un tercer empleado durante los períodos más ocupados. La medida del desempeño de estas opciones estará dada por el porcentaje de consumidores que espera más de 5 minutos por un servicio durante el curso de un día de trabajo. Se desea obtener el porcentaje de consumidores que esperan más de 5 minutos cuando solo dos empleados están trabajando y este mismo dato agregando un empleado en las horas pico.

Principales Ideas

Para resolver el problema la idea principal fue simular como funcionaría la cocina con dos y tres servidores en paralelo. Para ello se empezaría en $t = 0$, que representa la hora de apertura del puesto de comida, y se terminaría en $t = 660$ que representa la hora del cierre (660 min después de la hora de apertura). Para la entrada de clientes se utilizó una variable aleatoria $X \sim E(\lambda)$. pero este λ no sería el mismo durante toda la simulación, ya que durante los horarios pico la cantidad de clientes que entran al sitio debería aumentar y en el caso de la simulación con 3 servidores, en este horario debería entrar un 3er servidor, para resolver esto se llevó la cuenta del valor de t , para cuando este alcanzase los puntos del intervalo donde empieza el horario pico: $90 \leq t \leq 210$ y $420 \leq t \leq 540$, se cambiase el valor de λ por otro que aumentase la cantidad de clientes que entran al sitio durante la simulación y en el caso de los 3 servidores, incorporar al 3er servidor en ese intervalo.

Para modelar si un cliente pediría sushi o sandwich se utilizó una variable

aleatoria que toma los valores $\{0,1\}$ y según lo que pidiese, el servidor se demoraría una cantidad de tiempo según la variable aleatoria uniforme correspondiente ($X \sim U(5,8)$ o $X \sim U(3,5)$).

Para llevar los estados de cuantos servidores tengo libre y a que servidor le toca atender al cliente utilicé dos maquinas de estado, la 1ra con 4 estados para 2 servidores y la 2da con 9 estados para 3 servidores. Como solo se utilizan 3 servidores en los horarios pico, lo que se hacía era una función que a partir de una máquina y el estado actual, saber para que estado debería estar en la otra. Los estados de las maquinas son los siguientes:

E_i : Estado i-ésimo

S_i : Servidor número i

n: Cantidad de clientes pendientes(un cliente que se esta atendiendo también se cuenta como pendiente)

Máquina1:

$$E_0 \rightarrow S_1 = 0, S_2 = 0, n = 0$$

$$E_1 \rightarrow S_1 = 1, S_2 = 0, n = 1$$

$$E_2 \rightarrow S_1 = 0, S_2 = 1, n = 1$$

$$E_3 \rightarrow S_1 = 1, S_2 = 1, n = 2$$

$$E_4 \rightarrow S_1 = 1, S_2 = 1, n > 2$$

Máquina2:

$$E_5 \rightarrow S_1 = 0, S_2 = 0, S_3 = 0, n = 0$$

$$E_6 \rightarrow S_1 = 1, E_2 = 0, S_3 = 0, n = 1$$

$$E_7 \rightarrow S_1 = 0, E_2 = 1, S_3 = 0, n = 1$$

$$E_8 \rightarrow S_1 = 0, E_2 = 0, S_3 = 1, n = 1$$

$$E_9 \rightarrow S_1 = 1, E_2 = 1, S_3 = 0, n = 2$$

$$E_{10} \rightarrow S_1 = 0, S_2 = 1, S_3 = 1, n = 2$$

$$E_{11} \rightarrow S_1 = 1, S_2 = 0, S_3 = 1, n = 2$$

$$E_{12} \rightarrow S_1 = 1, S_2 = 1, S_3 = 1, n = 3$$

$$E_{13} \rightarrow S_1 = 1, S_2 = 1, S_3 = 1, n > 3$$

Una maquina en el estado E_i se mueve al estado E_j si al agregar un elemento(producirse un arribo) a E_i obtengo como resultado E_j (los elementos se agregan de izquierda a derecha, esto quiere decir que si tenemos $E_7 \rightarrow S_1 = 0, S_2 = 1, S_3 = 0, n = 1$ al agregar un elemento se activa S_1 y paso al estado E_9)

Si todos los Servidores están activos y se agrega un elemento, si estoy en E_12 o E_13 paso a E_13 , si estoy en E_3 o E_4 paso a E_4

Moverse de Máquina 1 a Máquina 2 en horario pico:

$E_0 \rightarrow E_5$
 $E_1 \rightarrow E_6$
 $E_2 \rightarrow E_7$
 $E_3 \rightarrow E_9$
 $E_4 \rightarrow E_{12}$: si la cantidad de clientes pendientes es 3
 $E_4 \rightarrow E_{13}$: si la cantidad de clientes pendientes es mayor que 3

Moverme de Máquina 2 a Máquina 1 al terminar horario pico:

$E_5 \rightarrow E_0$
 $E_6 \rightarrow E_1$
 $E_7 \rightarrow E_2$
 $E_8 \rightarrow E_0$
 $E_9 \rightarrow E_3$
 $E_{10} \rightarrow E_2$
 $E_{11} \rightarrow E_1$
 $E_{12} \rightarrow E_3$: si la cantidad de clientes en cola es 2(sin contar al que es atendido por el 3er servidor)
 $E_{13} \rightarrow E_4$: si la cantidad de clientes en cola es mayor que 2 (sin contar al que es atendido por el 3er servidor)

Al terminar la simulación tengo tres listas, una con el tiempo de llegada de cada uno de los clientes(si un cliente llega a las 11:00am se dice que llegó al tiempo $t = 60$), una con el tiempo de salida del local(de la misma forma que el tiempo de llegada) y una con el tiempo que se demoró en que se le preparase el pedido(que va desde 3 a 5 en sandwich y de 5 a 8 en sushi). El tiempo que deseo conocer para saber la eficiencia de 3 servidores respecto a 2 es el tiempo de espera de cada cliente antes de recibir el pedido, para ello se calculó este tiempo de la forma:

$\text{tiempo_de_espera} = \text{tiempo_Salida} - \text{tiempo_llegada} - \text{tiempo_preparación}$
 Luego se calculó el % de los tiempos de espera mayores a 5 minutos.

Para saber cual era más eficiente se corrió cada simulación 1000 veces y se tomó el valor del % promedio para tener una idea de que tanto mejora un caso respecto al otro.

Modelo desarrollado

Variables Estáticas de Tiempo

$T = 660$ (Tiempo total 11h = 660min)

$TPStartDia = 90$ (Empieza hora pico 90min despues de abrir)

$TPEndDia = 210$ (Termina hora pico 120min despues de empezar)

$TPStartTarde = 420$ (Empieza hora pico 420min despues de abrir)

$TPEndTarde = 540$ (Termina hora pico 540min despues de empezar)

Variables de tiempo

t : tiempo actual

tA : tiempo de arribo del proximo cliente

$tD1$: tiempo de salida del S1(servidor 1)

$tD2$: tiempo de salida de S2

$tD3$: tiempo de salida de S3

$NextEvent$: tiempo de entrada/salida al horario Pico

Variables contadoras

Na : Cantidad de arribos

Nd : Cantidad de idas

n : Cantidad de clientes en la cola

Variables de estado

$clientType$: Orden que pedirá el cliente(sushi o sandwich)

$clientQueue$ Cola con los clientes

SE : Proximo Evento(Para saber cuando empieza y termina el horario pico)

SS : Estado de la maquina de estados

Variables de salida

$Arrive$: Tiempos de llegada de los clientes a partir de $t = 0$

$Departure$: Tiempo de salida de los clientes a partir de $t = 0$

$PreparationTime$: Tiempo que demora hacer el sandwich o el sushi de un cliente

Inicialización

$tD1 = tD2 = tD3 = \infty$

$nextEvent = TPStartDia$

$t = tA = Na = Nd = TD1 = TD2 = TD3 = n = 0$

$NextEvent = 90$ Queue = []

$Arrive = Departure = PreparationTime =$

generar tA

$SS = 0$

Eventos de arribo

$tA \leq TDi \wedge tA < nextEvent \wedge tA < T$

- t = tA (se mueve linea del tiempo)
- Na = Na + 1 (Aparece un cliente)
- n = n + 1 (cantidad de clientes en la cola)
- Generar tA_t, tA = t + tA_t (Próximo arribo)
- Mover SS según la maquina de estados y generar el TDi correspondiente en caso de que haga falta.
- Si se generó TD1, TD2 o TD3: generar clientType
- Generar ptSand o ptSush a partir de clientType
- PreparationTime[Np] = ptSand o ptSush (según el que se generó)
- Cambiar Estado de SS según corresponda
- Arrive[Na] = t (marcar tiempo de llegada)

Eventos de partida

$TDi \leq tA \wedge TDi < nextEvent \wedge TDi < T$

- t = TDi (mover linea del tiempo)
- Nd = Nd + 1
- n = n - 1 (sale un cliente)
- Si en el estado correspondiente a SS, $S_i = 0$ entonces TDi = ∞
- Si SS es el estado 4 o 13, generar TDt y TDi = t + TDi
- Departure[Nd] = t (contar partida)

Entrada a horario Pico

$NextEvent \leq tA \wedge NextEvent < TDi$

- t = NextEvent
- NextEvent = Proximo Cierre Horario Pico
- Si hay cliente pendiente en cola generar ptSand o ptSush
- Np = 3er cliente de la cola
- preparationTime[Np] = ptSand o ptSush (según corresponda)
- tD3 = t + ptSand o ptSush
- SS = SS' (Aplicar cambio de máquinas de estado)

Salida de horario Pico

$NextEvent \leq tA \wedge NextEvent < TDi$

- t = NextEvent
- NextEvent = Proxima Apertura Horario Pico
- SS' = SS (Aplicar cambio de máquinas de estado)

Resultados

Al ejecutar cada simulación 1000 veces se obtuvo una media del 0.22% de clientes que esperan más de 5 minutos, al analizar los datos podemos notar que se debe a que la mayor cantidad de personas insatisfechas por la demora ocurren durante y después de los horarios pico, donde se forma una cola enorme de clientes que sigue afectando después de terminados los horarios pico.

Sin embargo al añadir un 3er servidor al sitio se aligera la carga de estos horarios condicionando a que no ocurra esta aglomeración. La media de las simulaciones con un 3er servidor en los horarios pico fue de 0.09%, un resultado mucho mejor. Por tanto es muy aconsejable tener un 3er servidor para los horarios.

Enlace Repositorio de Github

<https://github.com/zondekallix/Proyecto1Simulacion>