

Una biblioteca en C++ para la simulación rápida de colas

Palabras clave: simulación de colas, simulación rápida, simulación de procesos

Abstract

En este artículo reportamos los componentes de una biblioteca en C++ (y algunos resultados experimentales) para una simulación rápida de una red acíclica de colas. Esta biblioteca se desarrolló inicialmente para simular el Programa de Resultados Electorales Preliminares para las Elecciones Federales de México de 2018 y se ha aplicado recientemente para probar el desempeño de diferentes reglas de eliminación para el problema transitorio inicial en simulación de estado estacionario. Para ilustrar la validación del código que admite esta biblioteca, presentamos algunos resultados experimentales de réplicas de simulaciones de estado estable de una red de colas (Jackson) en serie. Estos resultados concuerdan con los resultados teóricos esperados de una red Jackson de colas. Se planea extender el código para una red con ciclos y crear una librería para R.

Introducción

La librería para simulación rápida de colas que reportamos en este artículo fue desarrollada inicialmente en C++ para simular el Programa de Resultados Electorales Preliminares (PREP) para las elecciones de 2018 organizadas por el Instituto Nacional Electoral de México. Un PREP para elecciones federales en México involucra varios procesos con una gran cantidad de entidades y recursos, por lo que su simulación en un software comercial puede requerir tiempos de ejecución inaceptables [1]. Para superar este inconveniente, se desarrolló una librería C++ para simulación rápida de colas para la simulación de un PREP. El código de esta librería tiene una clase (llamada `simqueue`) para la simulación de una cola usando la idea de la ecuación de Lindley [2]:

$$W_{j+1} \Rightarrow [W_j + S_j - T_{j+1}]^+, \quad (1)$$

donde $[x]^+$ es igual a x para $x \geq 0$, y 0 de otra forma, y, para $j = 0, 1, \dots$, W_j y S_j son los tiempos de espera (en cola) y de atención, para el cliente j , respectivamente, y T_{j+1} es el tiempo entre la llegada del cliente j y la llegada del cliente $(j + 1)$. La ecuación (1) es válida para una cola primer llegado primer servido con un servidor ($s = 1$), pero puede extenderse naturalmente para $s \geq 2$.

Para una corrida de n clientes (dadas las secuencias de tiempos entre llegadas y tiempos de servicio para los primeros n clientes), la Ecuación (1) nos permite simular los tiempos de espera en la cola mucho más rápido que usando el enfoque de eventos discretos (ver, e.g., [3] para una definición formal), que generalmente se implementa en un software comercial para simular modelos de simulación de eventos discretos más complejos (ver, por ejemplo, [4]). Para la simulación del PREP mexicano reportado en [1] se exportaron rutinas para la simulación de una sola cola con llegadas programadas de la biblioteca C++ a una biblioteca de enlace dinámico (dll), y las salidas y llegadas entre los diferentes nodos (procesos) de la red de colas se coordinaron utilizando Visual Basic for Applications (VBA) desde una interfaz de Excel. Para una aplicación posterior (reportada en [5] y [6]) la biblioteca de C++ se amplió con otra clase (llamada `simqueuenet`) que nos permite simular una red acíclica de colas. Se requiere que la red simulada sea acíclica para procesar la simulación de nodos en un cierto orden usando los procedimientos de la clase `simqueue`. La rutina para simular una red acíclica de colas se exportó a una dll y, usando una interfaz de Excel para recopilar los datos y reportar los resultados, se usó en [5] y [6] para probar el desempeño de diferentes métodos para eliminar la inicial. transitorio para la estimación

de medidas de rendimiento en estado estacionario. Para estas aplicaciones, la clase `simqueue` se amplió para considerar un número inicial de clientes en la cola, además de las llegadas programadas.

Algunos resultados experimentales

Para validar el código de nuestra biblioteca C++, exportamos una rutina (en una dll) para simular una red de colas, y se usó para ejecutar algunos experimentos usando una interfaz de Excel para ingresar y reportar datos en la hoja de cálculo. Los resultados que reportamos corresponden a simulaciones de una red de cuatro colas en serie, con llegadas al primer nodo solamente, y tiempos entre llegadas y de servicio distribuidos exponencialmente. Los resultados corresponden a 1000 experimentos para la estimación del tiempo de espera esperado en estado estacionario en cola en cada uno de los cuatro nodos (utilizando el método de medias por lotes) con longitudes de ejecución de 1000, 4000, 16000 y 64000 observaciones. Consideramos una intensidad de tráfico de $\rho = 0,8$, todas las colas inicialmente vacías y 10 lotes para el método de medias por lotes. Como se trata de una red de Jackson [7], podemos calcular el valor real del tiempo de espera esperado en estado estable en la cola, que es $E[W] = 4$ para cada nodo. Para el promedio (sobre 1000 experimentos) de los estimadores puntuales observamos que se acercan al valor real de 4 (a medida que aumenta la longitud de la ejecución). Además, el error cuadrático medio (MSE) tiende a cero.. El MSE (estimado) se calculó como el promedio (sobre 1000 experimentos) del cuadrado de las diferencias entre la media estimada del estado estacionario y el valor real. Como era de esperar, observamos que los anchos medios de los intervalos de confianza (IC) para la media disminuyen (a medida que aumenta la longitud de ejecución). Asimismo, sobre la cobertura empírica del valor real proporcionado por los IC del 90% correspondiente, observamos cierta sub-cobertura para una longitud de ejecución pequeña (1000) y una sobre-cobertura (aunque no significativa) en el primer nodo para una longitud de ejecución de 64000. Para longitudes de ejecución de 16000 y 64000, las coberturas son muy buenas.

Conclusiones

Nuestros resultados experimentales muestran que, usando la biblioteca reportada en este artículo, las estimaciones de la media de estado estacionario en cada nodo son consistentes, dando evidencia para la verificación del código de esta biblioteca para la simulación rápida de una red acíclica de colas.

Bibliografía

1. D. F. Muñoz, H. Gardida, H. Velázquez, and J. D. Ayala, “Simulation models to support the preliminary electoral results program for the Mexican Electoral Institute,” *Annals of Operations Research*, in press (2020).
2. D. V. Lindley, “The theory of queues with a single server,” *Mathematical Proceedings of the Cambridge Philosophical Society* 48, 277–289 (1952).
3. P. W. Glynn, “A GSMP Formalism for Discrete Event Systems,” *Proceedings of the IEEE* 77, 14–23 (1989).
4. J. S. Smith, D. T. Sturrock, and W. D. Kelton, *Simio and Simulation: Modeling, Analysis, Applications* (Simio LLC, Sewickley, PA, 2018).
5. D. F. Muñoz, “Initial transient deletion rules for steady-state simulation,” to appear in *Proceedings of the 19th International Conference on Numerical Analysis and Applied Mathematics*, edited by T. E. Simos and C. Tsitouras (AIP, College Park, MD, 2021).
6. D. F. Muñoz, “Empirical evaluation of initial transient deletion rules for the steady-state mean estimation problem,” *Computational Statistics*, in press (2022).
7. J. R. Jackson, “Networks of waiting lines,” *Operations Research* 5, 518–521 (1957).