

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский Государственный Университет Телекоммуникаций и Информатики»

Кафедра вычислительных систем

Лабораторная работа по дисциплине «Моделирование»

Выполнили: студенты 4 курса группы ИС-941

Патрушев Н.В.

Сизов М.А.

Проверил: старший преподаватель кафедры ВС

Петухова Я. В.

Новосибирск

2023 г.

Оглавление

Постановка задачи	3
Теория	4
Ход работы	6
Вывод	10

Постановка задачи

Реализовать в системе AnyLogic несколько систем массового обслуживания, с различными подходами к обработке входящих заявок:

- FIFO (First In - First Out)
- SF (Shortest First)
- FIFO с потерями
- SF с потерями

Теория

AnyLogic – программное обеспечение для имитационного моделирования, разработанное российской компанией The AnyLogic Company. Инструмент обладает современным графическим интерфейсом и позволяет использовать язык Java для разработки моделей.

Системой массового обслуживания (СМО) называется система, в которой возникает массовые требования на выполнения каких-либо видов услуг, а также происходит удовлетворение этих требований – обслуживание.

На вход n -канальной СМО с m -очередью поступает простейший поток заявок. Если прибор свободен, то он берет из очереди заявку или несколько заявок. Заявка принимается на обслуживание и обслуживается одним из n каналов. После окончания обслуживания один канал освобождается. Если был принят «пакет» заявок, то ожидается завершения обслуживания всех заявок в приборе. Если заявка застаёт все каналы занятыми, то она становится в очередь и «терпеливо» ждёт своего обслуживания. Дисциплина очереди естественная: кто раньше пришёл, тот раньше и обслуживается. Состояние рассмотренной системы будем связывать с числом заявок, находящихся в системе.

В зависимости от наличия возможности ожидания поступающими требованиями начала обслуживания СМО подразделяются на:

- системы с потерями, в которых требования, не нашедшие в момент поступления ни одного свободного прибора, теряются;
- системы с ожиданием, в которых имеется накопитель бесконечной ёмкости для буферизации поступивших требований, при этом ожидающие требования образуют очередь;

- системы с накопителем конечной ёмкости (ожиданием и ограничениями), в которых длина очереди не может превышать ёмкости накопителя; при этом требование, поступающее в переполненную СМО (отсутствуют свободные места для ожидания), теряется.

Задача анализа системы массового обслуживания заключается в определении ряда показателей ее эффективности, которые можно разделить на следующие группы:

- *показатели, характеризующие систему в целом:* число n занятых каналов обслуживания, число обслуженных, ожидающих обслуживания или получивших отказ заявок в единицу времени и т.д.;
- *вероятностные характеристики:* вероятность того, что заявка будет обслужена или получит отказ в обслуживании, что все приборы свободны или определенное число их занято, вероятность наличия очереди и т.д.;
- *экономические показатели:* стоимость потерь, связанных с уходом не обслуженной по тем или иным причинам заявки из системы, экономический эффект, полученный в результате обслуживания заявки, и т.д.

Ход работы

Для создания системы использовались следующие элементы AnyLogic:

- Source – генерирует заявки определенного типа. используется в качестве начальной точки потока агентов, а также создает агентов;
- Queue – моделирует очередь агентов, ожидающих приема объектами;
- Delay – задерживает агентов на заданный период времени. Время задержки вычисляется динамически, может быть случайным, зависеть от текущего агента или от каких-либо других условий; Время задержки в работе - triangular(0.5, 1, 1.5) секунд.
- Sink – уничтожает поступившие агентов. Используется в качестве конечной точки потока заявок.
- ResourcePull – задает набор ресурсов которые могут освобождаются и захватываться агентами.
- Seize - Захватывает для агента заданное количество ресурсов из указанного блока ResourcePool.

Задачи для модели генерировались с временем между прибытиями 1.

Вместимость очереди 10 задач, время обработки задачи обрабатывающим устройством определяется по функции triangular(0.5, 1, 1.5).



Рисунок 1. FIFO

Вместимость очереди 10 задач. Разрешен уход по таймауту, если задача находится в очереди 10 секунд. Время обработки задачи обрабатывающим устройством определяется по функции $\text{triangular}(0.5, 1, 1.5)$.



Рисунок 2. FIFO с потерями

На вход модели поступают агенты, которые попадают в одну из двух очередей, в которых занимают 2 или 1 ресурс из доступного пула ресурсов. Всего доступно 3 ресурса. Таким образом реализуется разделение задач на маленькие и большие. После обработки агенты высвобождают занятые собой ресурсы и очередь двигается дальше. Таким образом обрабатываются либо 3 маленьких задач одновременно, либо 1 большая и 1 маленькая. Вместимость очередей 1000.

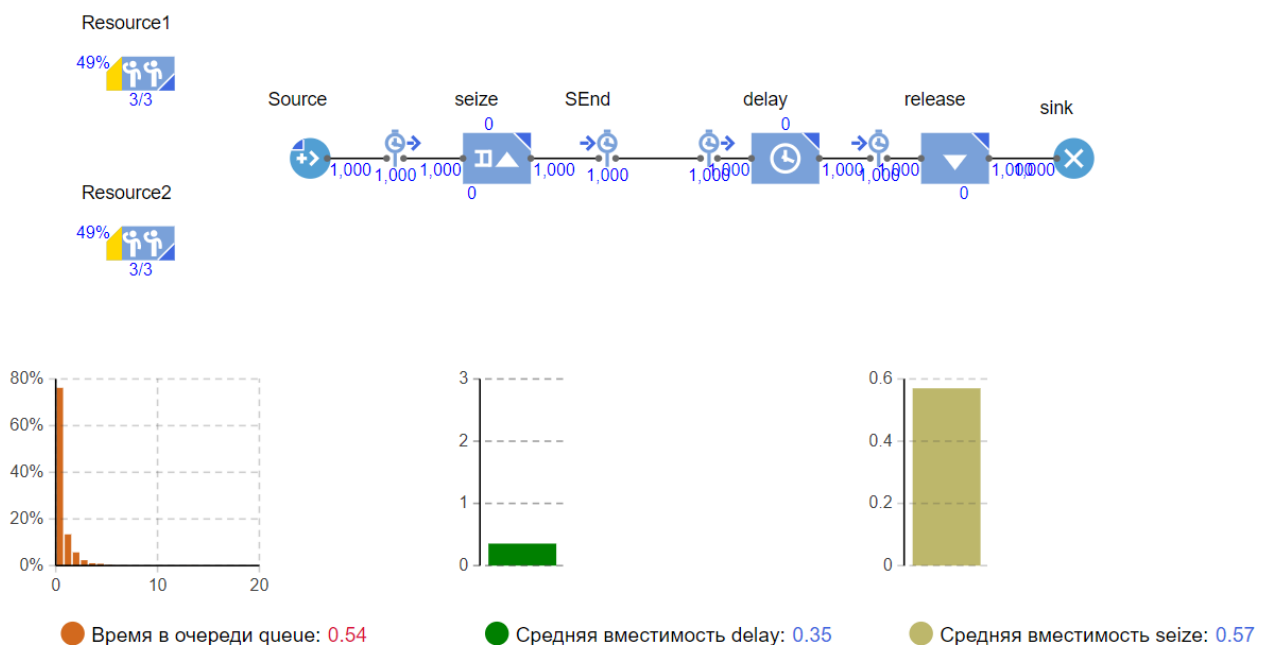


Рисунок 3. SF

Такая же модель очереди SF, только с предусмотренной потерей агентов. Вместимость каждой очереди 5 агентов, время до ухода по таймауту 10 секунд.

Вывод

Сравнив две созданные модели FIFO и SF можно увидеть, что длина очереди FIFO больше, чем длина очереди SF. При этом у FIFO среднее время во всех случаях больше, чем у SF. Из этого следует, что система SF может обрабатывать задачи, приходящие с большей интенсивностью, чем аналогичная система FIFO.

Однако стоит отметить, что, хоть система SF и показывает наилучшие результаты, в такой системе маленькие задачи простаивают в очереди дольше.

Используя систему с потерями, среднее время обработки было минимально из-за отсутствия времени простоя в очереди. При FIFO потери достигли 47 агентов, а в SF лишь 6. Модель показывает, что большинство агентов ушли из-за заполненной очереди. По таймауту уходили единицы, поскольку средняя длина очереди была меньше порогового значения.