Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации Сибирский Государственный Университет
Телекоммуникаций и Информатики СибГУТИ

Кафедра вычислительных систем

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
по дисциплине «Моделирование»

Выполнил:
ст. гр. ИВ-921
Ярошев Р. А.

Проверил:
Старший преподаватель
Петухова Я.В.

Новосибирск 2023

Оглавление

Формулировка задания

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

1. Очередь FIFO,
2. Очередь SF,
3. Очередь SF с сортировкой.

Теоретические сведения

СМО — система, которая производит обслуживание поступающих в неё требований. Обслуживание требований в СМО осуществляется обслуживающими приборами. Существуют:

- ***Системы с потерями***, в которых требование, 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 образую очередь;
- ***Системы с накопителем конечной ёмкости*** (ожидание и ограничениями), в которых длина очереди не может превышать ёмкости накопителя; при этом требование, поступающее в переполненную СМО теряются;
- ***Системы с потерями***, в которых требования находятся в очереди лишь ограниченное время, после чего теряются.

Очередь FIFO — называется такой последовательный список переменной длины, в котором включение элементов выполняется только с одной стороны списка, а исключение с другой.

Очередь SF — называется такой последовательный список, в котором сравниваются количество свободных обработчиков и если имеется свободное место, то алгоритм пытается найти в очереди агента, который может быть обработан имеющимися ресурсами.

Результаты выполнения работы

- ***Source*** — используется в качестве начальной точки потока агентов, а так же создает агентов;
- ***SelectOutput*** - блок направляет входящих агентов в один из двух выходных портов в зависимости от выполнения заданного (детерминистического или заданного с помощью вероятностей) условия.
- ***Queue*** — моделирует очередь агентов, ожидающих приема объектами;
- ***Hold*** — блокирует/освобождает поток агентов на определенном участке диаграммы процесса;
- ***Delay*** — задерживает агентов на заданный период времени. Время задержки вычисляется динамически, может быть случайным, зависеть от текущего агента или от каких-либо других условий;
- ***Sink*** — уничтожает поступивших агентов.
- **timeMeasureStart/End —** задает время нахождения агента в системе.

Задачи генерируются в модулях *sourse: «Малая»* и *«Большая»*. Для каждой задачи заданы параметры: для большой: ширина - 2, высота - 1 и длина - 2, для малой ширина - 1, высота - 1 и длина - 1.

Большие и малые задачи поступают на вход очереди (модуль *queue*) вместимостью 150, выход из которой осуществляется в соответствии с задержкой *uniform(0,1)* секунды, установленной модулем *delay*. Обработанные задачи направляются в модуль *sink*. Задачи не поместившиеся в очередь уничтожаются модулем *sink2*. Задачи, ушедшие по таймауту направляются в *sink1*.

Переменная *sumWidth* суммирует параметры агентов, находящихся в блоке *delay*.

Функция mySort ищет подходящей задачу из очереди для заполнения емкости delay до 3.

1. Очередь FIFO с потерями.

1. Блок *source «Малая»*

- Агенты прибывают согласно: интенсивности
- Интенсивность прибытия: 10 в секунду
- Максимальное кол-во прибытий: 1500
- Длина: 1
- Ширина: 1
- Высота: 1

2. Блок *source «Большая»*

- Агенты прибывают согласно: интенсивности
- Интенсивность прибытия: 5 в секунду
- Максимальное кол-во прибытий: 1500
- Длина: 1
- Ширина: 2
- Высота*:* 2

3. Блок *selectOutput*

- Выход true выбирается по условию: !queue.canEnter();

4. Блок *queue*

- Вместимость: 150
- Очередь: FIFO
- Таймаут: 15 секунд

5. Блок *hold*

- Режим: Условное (свое условие для каждого агента)
- Условие блокировки: *sumWidth + agent.getWidth() > 3 && sumHeight + agent.getHeight() > 3*
- Максимальная ширина: 3
- Максимальная высота: 3

6. Блок *delay*

- Время задержки: *uniform(0,1)* секунды
- Вместимость: 3

- При входе:
  sumWidth += (int)agent.getWidth();
  sumHeight += (int)agent.getHeight();
- При подходе к выходу:
  sumWidth -= (int)agent.getWidth();
  sumHeight += (int)agent.getHeight();
- При выходе:
  hold.recalculateConditions();

7. *sink* — содержит успешно завершенные задачи,
8. *sink1* — содержит задачи ушедшие по таймауту,
9. *sink2* — содержит задачи не поместившиеся в очередь,
10. timeMeasureStart — начало замера времени,
11. timeMeasureEnd:
    timeMeasureStart.

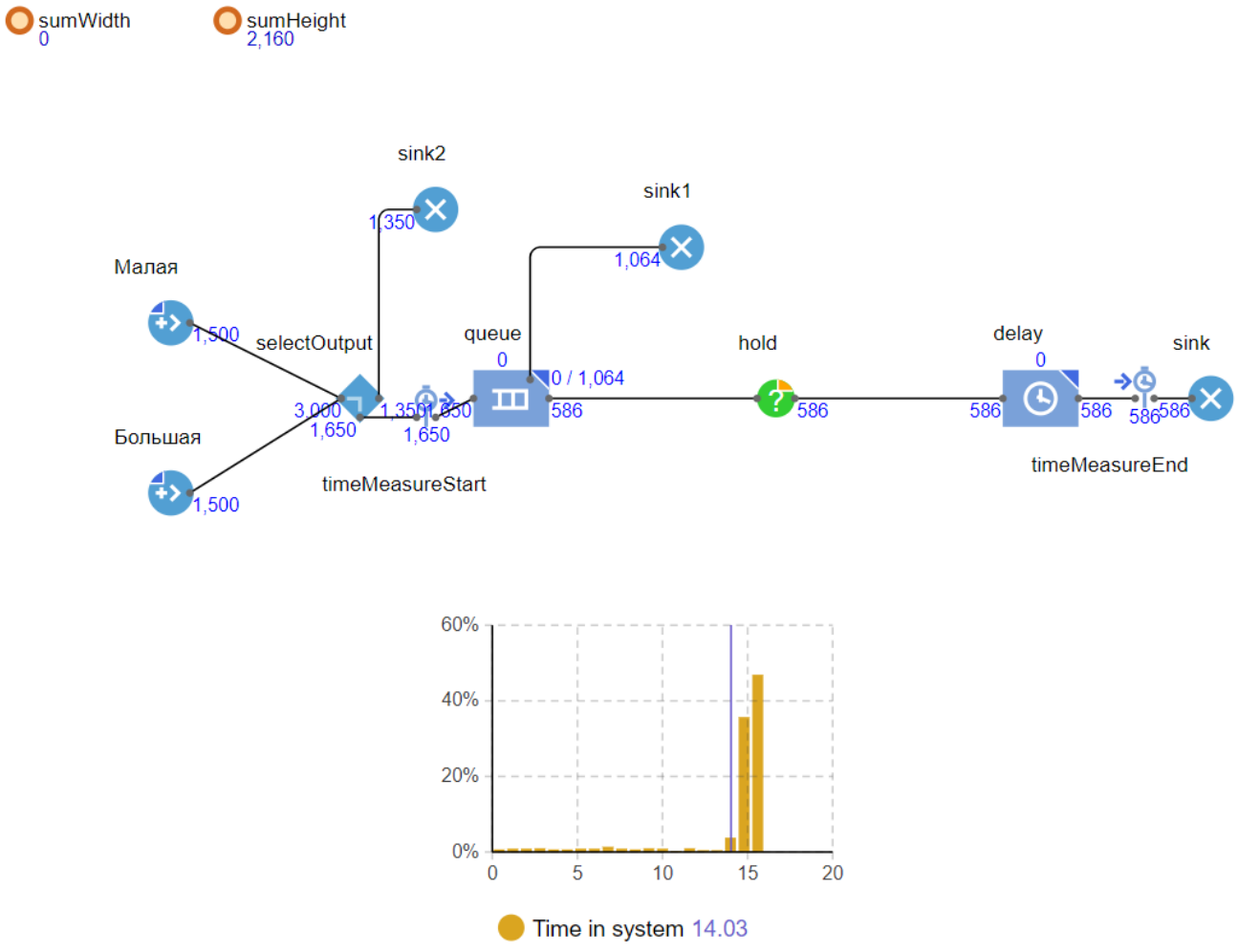

Рисунок 1. Модель FIFO.

2. Очередь SF с потерями.

1. Модуль *source «Малая»:*

- Агенты прибывают согласно: интенсивности
- Интенсивность прибытия: 10 в секунду
- Максимальное кол-во прибытий: 1500
- Длина: 1
- Ширина: 1
- Высота: 1
- Блок source «Большая»
- Агенты прибывают согласно: интенсивности
- Интенсивность прибытия: 5 в секунду
- Максимальное кол-во прибытий: 1500
- Длина: 1
- Ширина: 2
- Высота: 2

2. Блок *selectOutput:*

- Выход true выбирается: при выполнении условия
- *Условие: !queue.canEnter();*

3. Блок queue:

- Вместимость: 150
- Очередь: По приоритету
- Приоритет агента: -agent.getWidth()
- Таймаут: 15 секунд

4. Блок *hold:*

- Режим: Условное (свое условие для каждого агента)
- Условие блокировки: sumWidth + agent.getWidth() > 3 && sumHeight + agent.getHeight() > 3
- Максимальная ширина: 3
- Максимальная высота: 3

5. Блок *delay:*

- Время задержки: *uniform(0,1)* секунды
- Вместимость: 3

- При входе:
  sumWidth += (int)agent.getWidth();
  sumHeight += (int)agent.getHeight();
- При подходе к выходу:
  sumWidth -= (int)agent.getWidth();
  sumHeight += (int)agent.getHeight();
- При выходе:
  hold.recalculateConditions();

6. Блок *sink* — содержит обработанные задачи,
7. Блок *sink1* — содержит задачи ушедшие по таймауту,
8. Блок *sink2* — содержит задачи не поместившиеся в очередь,
9. timeMeasureStart — начало замера времени,
10. timeMeasureEnd:
    timeMeasureStart.

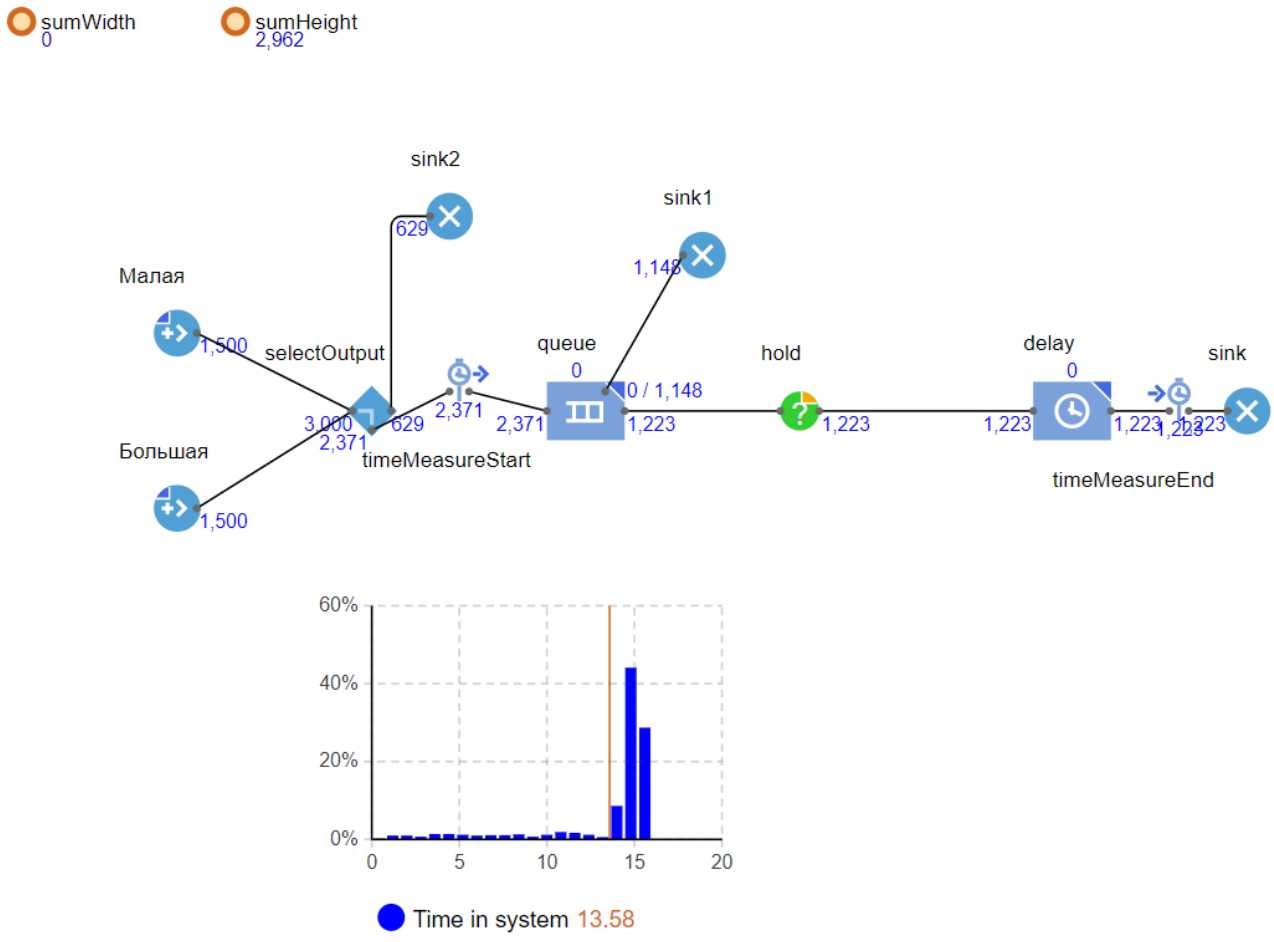

Рисунок 2. Модель SF с потерей задач.

3. Очередь SF с сортировкой.

1. Блок source «Малая»:

- Агенты прибывают согласно: интенсивности,
- Интенсивность прибытия: 10 в секунду,
- Максимальное кол-во прибытий: 1500,
- Длина: 1,
- Ширина: 1,
- Высота: 1,
- Блок source «Большая»,
- Агенты прибывают согласно: интенсивности,
- Интенсивность прибытия: 5 в секунду,
- Максимальное кол-во прибытий: 1000,
- Длина: 1,
- Ширина: 2,
- Высота: 2.

2. Блок selectOutput:

- Выход true выбирается: при выполнении условия
- Условие: !queue.canEnter();

3. Блок queue:

- Вместимость: 150
- Очередь: По приоритету
- Приоритет агента: agent.getWidth()
- Таймаут: 10 секунд

4. Блок hold:

- Режим: Условное (свое условие для каждого агента),
- Условие блокировки: sumWidth + agent.getWidth() > 3 && sumHeight + agent.getHeight() > 3,
- Максимальная ширина: 3,
- Максимальная высота: 3.

5. Блок *delay:*

- Время задержки: *uniform(0,1)* секунды,
- Вместимость: 3,
- При входе: agent.getLength(), agent.getWidth(),
- При подходе к выходу:

```
sumWidth += (int)agent.getWidth();
sumHeight += (int)agent.getHeight();
sortQueue();
```

- При выходе:

```
sumWidth -= (int)agent.getWidth();
sumHeight -= (int)agent.getHeight();
sortQueue();
```

6. Блок *sink* — уничтожает обработанные задачи,
7. Блок *sink1* — уничтожает задачи ушедшие по таймауту,
8. Блок *sink2* — уничтожает задачи не поместившиеся в очередь,
9. timeMeasureStart — начало замера времени,
10. timeMeasureEnd:
    timeMeasureStart.

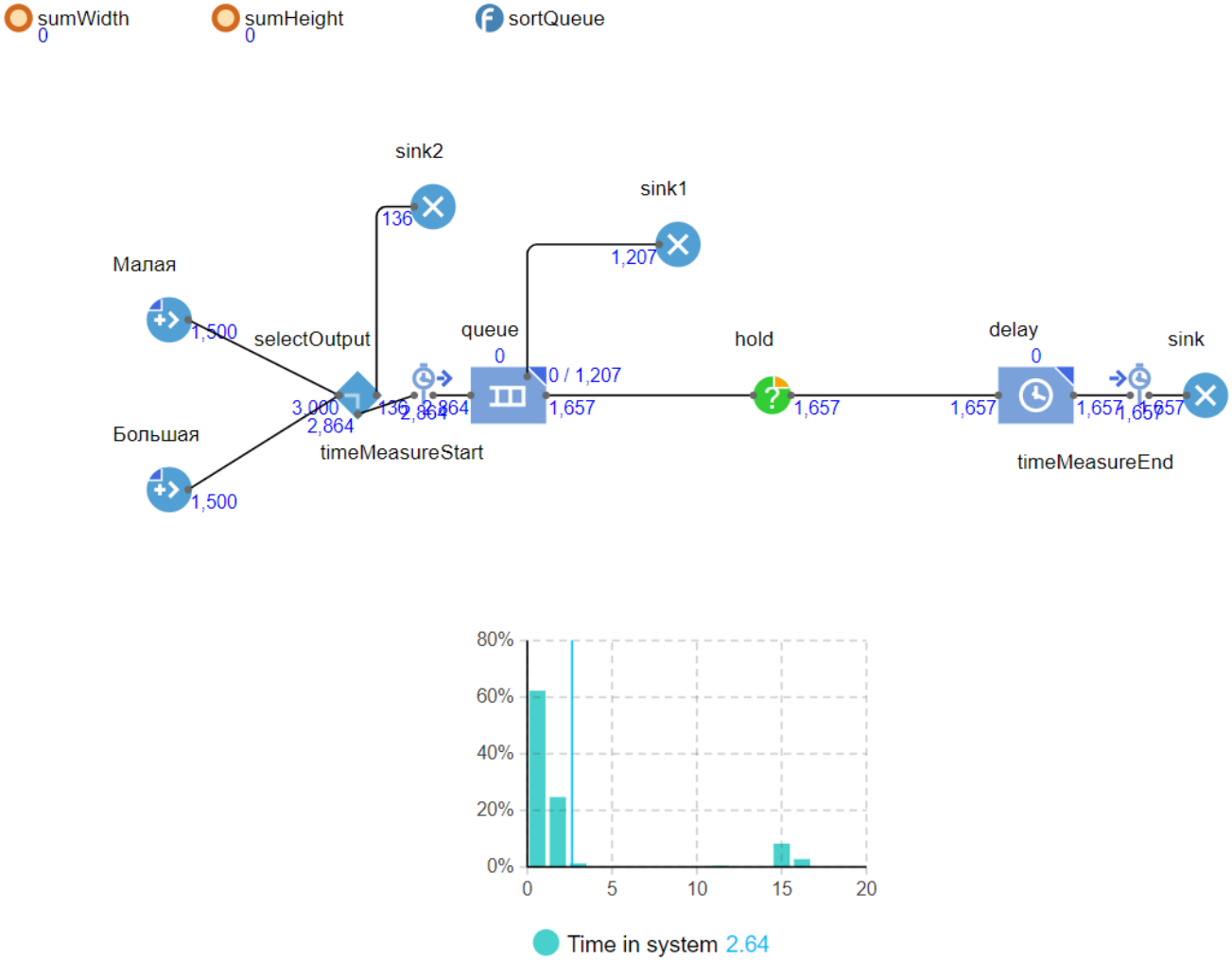

Рисунок 3. Модель SF с сортировкой задач.

Вывод по результатам работы

В ходе данной лабораторной работы были построены 3 модели систем массового обслуживания в системе AnyLogic:

1. Очередь FIFO;
2. Очередь SF;
3. Очередь SF с сортировкой.

В экспериментах было сгенерировано 1500 малых задач с шириной 1 и высотой 1, а также с интенсивностью 10 задач в секунду, и больших задач с шириной 2, высотой 2 и интенсивностью 5 задач в секунду.

Для сравнения эффективности способов обработки задач применялся модуль timeMeasure. Были построены соответствующие графики, отображающие время нахождения задачи в системе.

По данным графиков, видно, что модель с сортировкой задач по параметрам: ширина, требует меньше времени на обработку агента и составляет 2.64 с модельного времени. Хуже всего функционирует модель FIFO с уходом задач от переполнения в модуле queue, где время обработки агента составляет 14.03 с модельного времени.