

Моделирование систем. Практика 6.

Вариант 1

Р. Абдуллин, В. Верхотуров

БСБО-05-20

РТУ МИРЭА

1 декабря 2022 г.

1 Модель

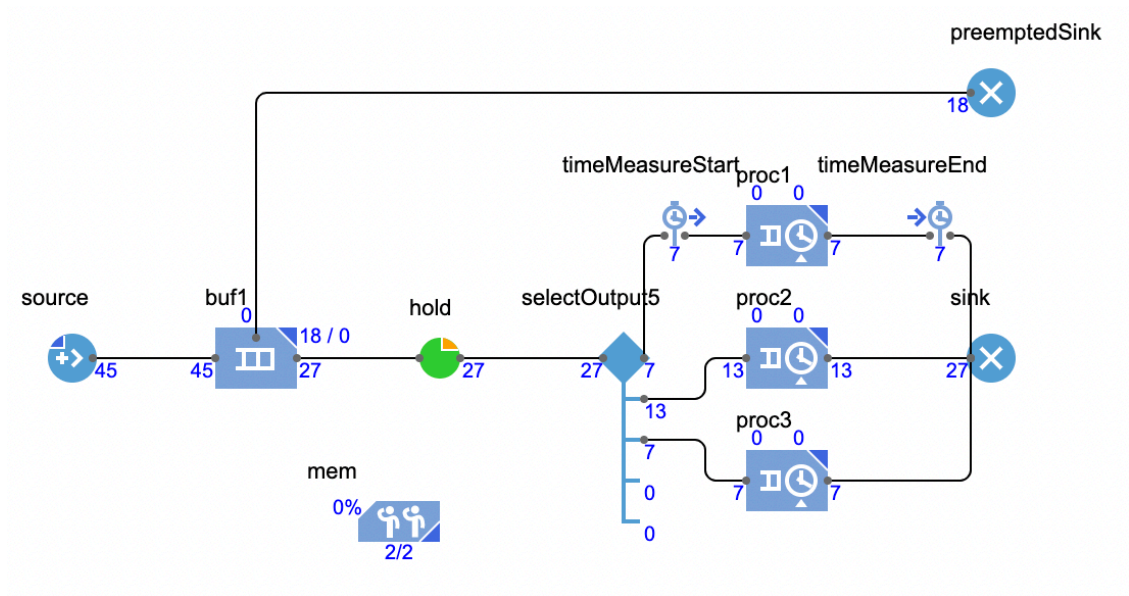


Рис. 1: Параллельная вычислительная система с использованием общих ресурсов

source - Source

Name: ☒ Show

Arrivals defined by: ▼

Arrival rate:

Set agent parameters from DB: ☐

Multiple agents per arrival: ☐

Limited number of arrivals: ☒

Maximum number of arrivals:

Рис. 2: Source

buf1 - Queue

Name:

buf1

Capacity:

24

Maximum capacity:

Agent location:

node

Advanced

Queuing:

FIFO

Enable exit on timeout:

Enable preemption:

Restore agent location on exit:

Force statistics collection:

Actions

On enter:


On at exit:

refreshHold();

Рис. 3: Buf1

hold - Hold

Name:

Mode:  Manual (use block(). unblock())




Initially blocked:  ☐


Рис. 4: Hold


selectOutput5 - SelectOutput5

Name:

Use:  ☐ Probabilities
☒ Conditions
☐ Exit number

Condition 1:  ☐ `proc1.size() == 0`

Condition 2:  ☐ `proc2.size() == 0`

Condition 3:  ☐ `proc3.size() == 0`




Condition 4:  ☐ `false`

Рис. 5: selectOutput5

 **sink - Sink**


Name: ☒ Show name ☐

▼ **Actions**

On enter: 


```
refreshHold();  
  
countSunkedAgents += 1;  
getProc1CoefficientWork();  
  
timeIntervalsData.add(time() - agent.enterTime);
```

Рис. 6: Sink

 **preemptedSink - Sink**

Name:

▼ **Actions**

On enter: 

```
countSunkedAgents += 1;  
getProc1CoefficientWork();
```

Рис. 7: PreemptedSink

mem - ResourcePool

Name: ☒ Show n

Resource type: ▼

Capacity defined: ▼

Capacity:

When capacity decreases: ▼

New resource unit: ▼

Home location (nodes):

Show default animation: ☒

Рис. 8: Memory

refreshHold - Function

Name: ☒ Show name ☐ Ignore

Visible: ☒ yes

☒ Just action (returns nothing)
☐ Returns value

► Arguments

▼ Function body

```

boolean isBusyProc1 = proc1.size() > 0;
boolean isBusyProc2 = proc2.size() > 0;
boolean isBusyProc3 = proc3.size() > 0;

hold.setBlocked(isBusyProc1 && isBusyProc2 && isBusyProc3);

```

Рис. 9: RefreshHold

V countSinkedAgents - Variable

Name: ☒ Show

Visible: ☒ yes

Type: ▼


Initial value: 

Рис. 10: CountSinkedAgents

F getProc1CoefficientWork - Function

Name: ☒ Show name ☐ Ignore

Visible: ☒ yes

☒ Just action (returns nothing)
☐ Returns value

► Arguments

▼ Function body

```


if (countSinkedAgents == 45) {
    double tatalProc1Time = 0;

    int size = timeMeasureEnd.dataset.size();

    for (int i = 0; i < size; i++) {
        tatalProc1Time += timeMeasureEnd.dataset.getY(i);
    }
    proc1CoefficientWork = tatalProc1Time / time();
}

```

Рис. 11: GetProc1CoefficientWork

 **proc1CoefficientWork - Variable**

Name:

proc1CoefficientWork

Visible:

☒ yes

Type:

double ▼

Рис. 12: proc1CoefficientWork

2 Описание модели

Правило прибытия (сек.) Интенсивность, 2;

Максимальное кол-во сообщений 45;

Размер буфера 24;

Время обработки:

процессора № 1 28;

процессора № 2 25;

процессора № 3 28;

Контролируемый параметр резидентное время, коэффициент загрузки процессора (использовать timeMeasureStart и timeMeasureEnd);

Блок процессор 1.

3 Резидентное время

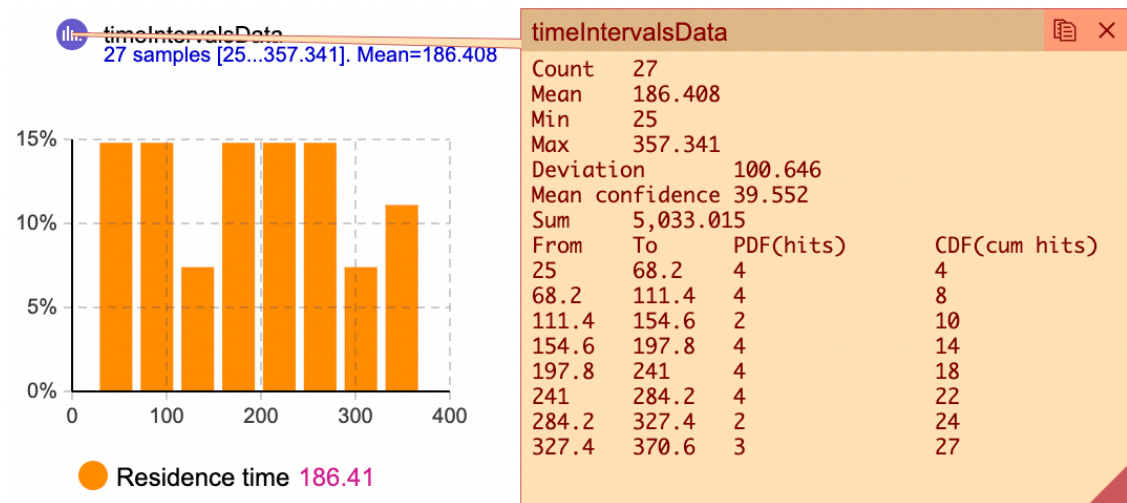


Рис. 13: Коэффициент загрузки процессора

4 Резидентное время

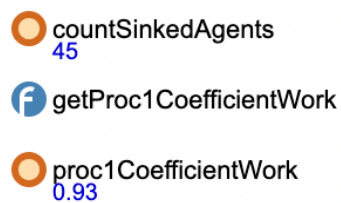


Рис. 14: Коэффициент загрузки процессора

5 Визуализация

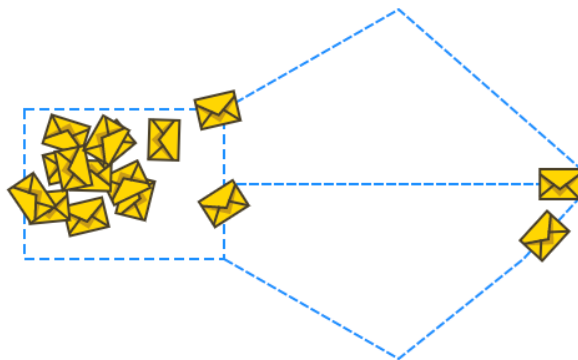


Рис. 15: Визуализация модели

6 Анализ системы

18 сообщений из 45 были утеряны в буфере. При памяти, вместительностью 2, невозможна работа одновременно трех процессоров.

7 Вывод

Для улучшения модели можно увеличить очередь, объем разделяемой памяти.