

Практическое занятие № 8

Тема: Расчет СМО с дисциплинами Split-Join и Fork-Join.

Цель: Приобретение практических навыков расчета показателей оперативности обработки данных с процессами расщепления и слияния заявок с помощью системы массового обслуживания Split-Join и Fork-Join.

Язык программирования, ПО и библиотеки: python 3.x, установленный пакет библиотек Anaconda или отдельно следующие: pandas, numpy, matplotlib. Среда разработки – PyCharm, Spyder 3 или Jupyter.

В эпоху больших данных все больше основных вычислительных инфраструктур развертываются в распределенном режиме и неизбежно задействуют очереди Split-Join и Fork-Join для облегчения хранения и обработки крупномасштабных наборов данных. Например, Cassandra [1] и Dynamo [2], два популярных хранилища данных «ключ-значение», используют очереди Fork-Join для одновременного выполнения операций чтения и записи.

В системе очередей Fork-Join поступающая в систему заявка (в дальнейшем – задача) разветвляется на n подзадач, и каждая подзадача отправляется на один из свободных узлов. Результаты завершенных подзадач объединяются на центральном узле соединения. Когда интенсивность прибытия заявок высока, подзадаче, возможно, придется ждать обслуживания в подочереди своего узла в порядке «первым пришел – первым обслужен». Считается, что задание выполнено после того, как все результаты были получены в узле соединения.

На рисунке 1 приведен пример системы массового обслуживания (СМО) с дисциплиной Fork-Join и тремя каналами обслуживания. В данном примере подзадачи A_2 и A_3 задачи A закончили обслуживание и ожидают подзадачу A_1 , а освободившиеся каналы обслуживания заняты подзадачами B_2 и B_3 задачи B . Как только подзадача A_1 завершит обслуживание, произойдет объединение результатов, и задача A покинет систему. В системе Split-Join (рисунок 2), в отличие от Fork-Join, очередная задача ждет освобождения всех подзадач предыдущей.

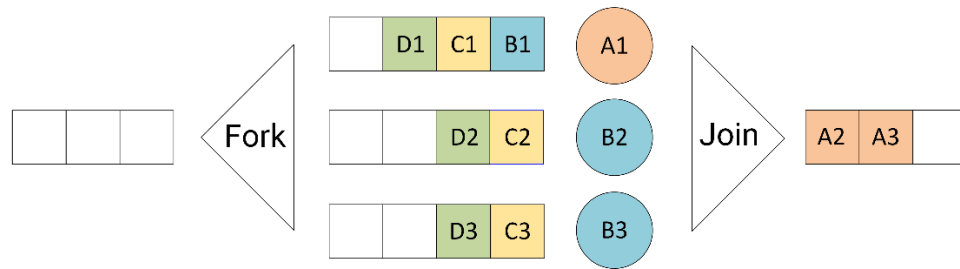


Рисунок 1. Процесс обслуживания в СМО с дисциплиной Fork-Join

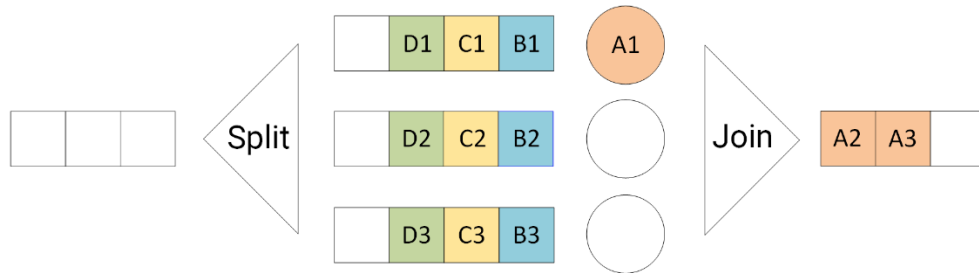


Рисунок 2. Процесс обслуживания в СМО с дисциплиной Split-Join

Процессам Fork-Join посвящено достаточно большое количество работ [3-16]. Для систем Fork-Join точное выражение для среднего времени пребывания в системе с произвольным распределением обслуживания было получено только для системы с двумя каналами [5, 6]. Для числа каналов $n > 2$ и экспоненциального распределения времени обслуживания с помощью различных методов были получены аппроксимации среднего времени пребывания [8, 10, 11].

Приведем кратко полученные результаты. В [6] приведена точная формула для среднего время пребывания в системе *при $n=2$ и экспоненциального распределения времени обслуживания*

$$v_1 = (H_2 - \rho/8) \cdot v_1^{(M)}, \quad (1)$$

где $H_n = \sum_{i=1}^n 1/i$ и для $n=2$ $H_2 = 1.5$,

$\rho = \lambda / \mu$ – коэффициент загрузки системы,

$v_1^{(M)} = \frac{1}{(\mu - \lambda)}$ – среднее время пребывания в СМО М/М/1.

В [8] Nelson R. и Tantawi A.N. предложена аппроксимация для среднего времени пребывания в СМО Fork-Join с n каналами обслуживания

$$v_1 \approx \left[\frac{H_n}{H_2} + \frac{4}{11} \left(1 - \frac{H_n}{H_2} \right) \rho \right] \frac{12 - \rho}{8} \frac{1}{\mu - \lambda}, n \geq 2. \quad (2)$$

В [10] Varki E., Merchant A. и Chen H. предложили формулу

$$v_1 \approx \frac{1}{\mu} \left[H_n + \frac{\rho}{2(1 - \rho)} \left(\sum_{i=1}^n \frac{1}{i - \rho} + (1 - 2\rho) \sum_{i=1}^n \frac{1}{i(i - \rho)} \right) \right], n \geq 2. \quad (3)$$

В [11] Varma S. и Makowski A.M. предложен метод аппроксимации среднего времени пребывания на основе комбинации методов интерполяции высоких и низких значений коэффициентов загрузки системы:

$$v_1 \approx [H_n + (V_n - H_n)\rho] \frac{1}{\mu - \lambda}, \quad (4)$$

$$\text{где } V_n = \sum_{i=1}^n \binom{n}{i} (-1)^{i-1} \sum_{m=1}^i \binom{i}{m} \frac{(m-1)!}{i^{m+1}}.$$

Для СМО Split-Join начальные моменты времени обслуживания заявок $g_j, j = \overline{1, m}$ представляют собой начальные моменты распределения максимума из n случайных величин времен обслуживания каждой из подзадач в канале и может быть рассчитано по формулам:

$$g_j = j \sum_{k=1}^r t_k^{j-1} A_k \overline{F}^*(t_k) e^{t_k}, \quad (5)$$

где для $\{x_k\}$ и $\{A_k\}$ существуют справочные таблицы при различных значениях r , а $\overline{F}^*(t)$ представляет собой ДФР распределения максимума времени обслуживания в n каналах, функция которого определяется как:

$$F^*(t) = \prod_{i=1}^n F_i(t),$$

где $F_i(t)$ – функция распределения времени обслуживания в i -ом канале.

После получения начальных моментов распределения максимума становится возможным представить весь процесс обслуживания в виде одноканальной СМО типа $M/G/1$ с начальными моментами времени обслуживания в канале $g_j, j = \overline{1, m}$.

Порядок выполнения практического занятия:

1. Определите свой вариант выполнения ПЗ по журналу согласно таблице 1. Вариант определяет метод аппроксимации СМО Fork-Join, который необходимо будет реализовать в ходе ПЗ, и начальное значение коэффициента загрузки СМО.

Таблица 1. Определение варианта выполнения ПЗ

Номер по журналу	Аппроксимация для СМО Fork-Join	Значение коэффициента загрузки для одноканальной СМО
1	Nelson R. и Tantawi A.N.	0.9
2	Varki E., Merchant A. и Chen H.	0.9
3	Varma S. и Makowski A.M.	0.9
4	Nelson R. и Tantawi A.N.	0.85
5	Varki E., Merchant A. и Chen H.	0.85
6	Varma S. и Makowski A.M.	0.85
7	Nelson R. и Tantawi A.N.	0.8
8	Varki E., Merchant A. и Chen H.	0.8
9	Varma S. и Makowski A.M.	0.8
10	Nelson R. и Tantawi A.N.	0.75
11	Varki E., Merchant A. и Chen H.	0.75
12	Varma S. и Makowski A.M.	0.75
13	Nelson R. и Tantawi A.N.	0.7
14	Varki E., Merchant A. и Chen H.	0.7
15	Varma S. и Makowski A.M.	0.7
16	Nelson R. и Tantawi A.N.	0.95
17	Varki E., Merchant A. и Chen H.	0.95
18	Varma S. и Makowski A.M.	0.95

2. Откройте файл *pz8.py*. В ходе ПЗ мы будем исследовать зависимость среднего времени пребывания заявок в СМО Split-Join и Fork-Join от числа каналов. Для этого мы сперва задаем диапазон изменения числа каналов исследуемых СМО от 2 до 10 включительно

```
ns = [n for n in range(2, 11)]
```

а также интенсивность входного потока и число заявок, требуемых к обслуживанию в СМО Split-Join и Fork-Join

```
# интенсивность вх потока заявок  
l = 1.0  
  
# Число заявок, требуемых к обслуживанию в СМО Split-Join и Fork-Join  
n_jobs = 3000
```

Далее задаем коэффициент загрузки одноканальной СМО. Измените его на ваше значение, согласно варианту ПЗ

```
# Коэффициент загрузки при n=1.  
ro_start = 0.9
```

3. В ходе ПЗ мы будем увеличивать число каналов с одновременным кратным увеличением интенсивности обслуживания в канале. Предполагается, что разбиение исходной задачи на части приводит к кратному уменьшению времени обработки в канале. Это можно проиллюстрировать на примере обработки изображений, получаемых с космических аппаратов дистанционного зондирования Земли (рисунок 3). В данном случае изображение в процессе обработки разделяется на фрагменты, которые обрабатываются параллельно. После окончания обработки последнего из фрагментов происходит объединение результатов.

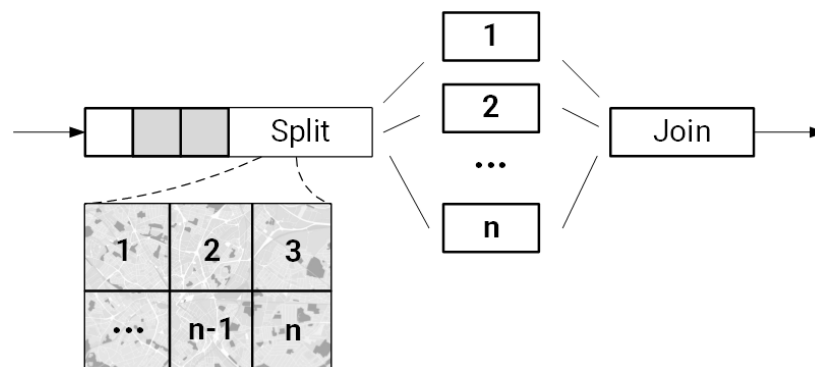


Рисунок 3. Процесс обработки изображения с КА ДЗЗ

4. После задания списков для накопления результатов расчетов и имитационного моделирования в тексте программы можно заметить следующие строки

```
# Вычисляем время обработки без ускорений  
# ro = 1*b1/n  
# -> b1 = ro*n/1 = ro -> mu = 1.0/b1 = 1.0/ro  
mu_one = 1 / ro_start  
b_one_channel = rd.Exp_dist.calc_theory_moments(mu_one, 3)  
v_base = mgl_calc.get_v(1, b_one_channel)[0]
```

Здесь происходит расчет среднего времени пребывания заявок без дробления задачи на части, т.е. в том случае, если у нас только один канал обслуживания. С этим значением будем сравнивать рассчитанные времена пребывания с учетом разбиения задачи на части и вычислять во сколько раз произошло ускорение обработки по сравнению с исходным вариантом.

5. Далее запускается цикл расчета средних времен пребывания заявок в СМО Split-Join и Fork-Join. Обратите внимание, что интенсивность обслуживания в канале кратно увеличивается

```
# Пересчитываем интенсивность обслуживания заявок
mu = mu_one * n
```

6. Численный расчет СМО Split-Join уже реализован в тексте программы и осуществляется здесь

```
# 2. Численный расчет СМО Split-Join через распределение максимума СВ
b_one_channel = rd.Exp_dist.calc_theory_moments(mu, 3)
b_max = getMaxMoments(n, b_one_channel)
v_ch = mg1_calc.get_v(1, b_max)[0]
v1_sj_ch.append(v_ch)
```

В первой строчке происходит расчет начальных моментов времени обслуживания в канале. В итоге в *b_one_channel* будет список из трех начальных моментов времени обслуживания, рассчитанных по формуле

$$f_k = \frac{k!}{\mu^k},$$

где *k* – номер начального момента.

Далее происходит вычисление начальных моментов распределения максимума СВ по формуле (5) и расчет одноканальной СМО M/G/1, у которой распределение обслуживания задано начальными моментами распределения максимума СВ.

7. Если запустить сейчас программу на выполнение, то она завершится с ошибками, поскольку не реализована функция расчета среднего времени пребывания заявок в СМО Fork-Join

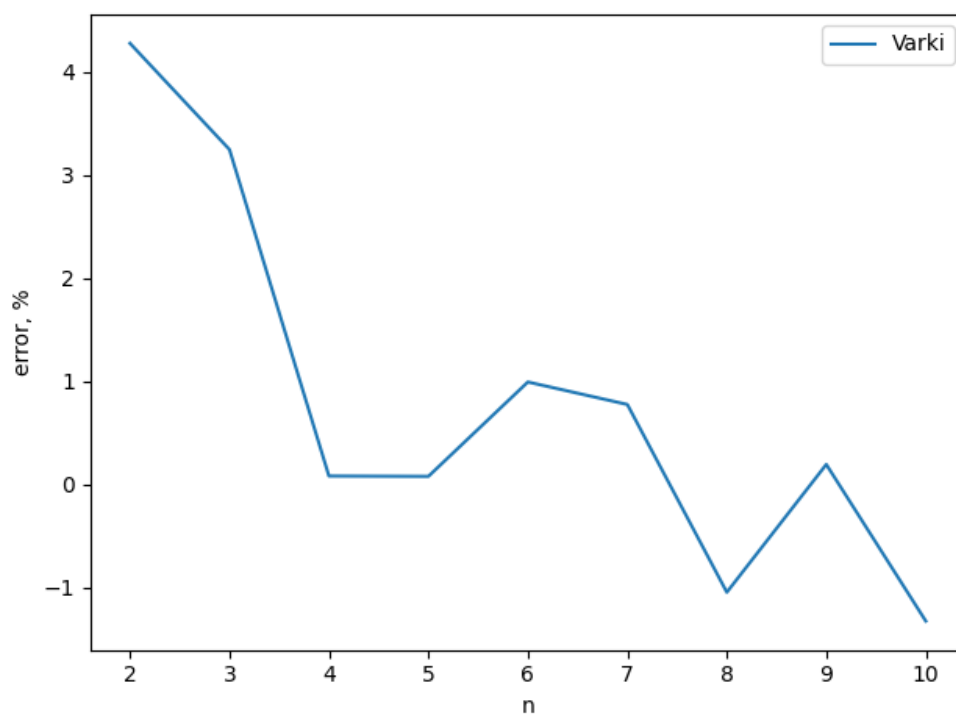
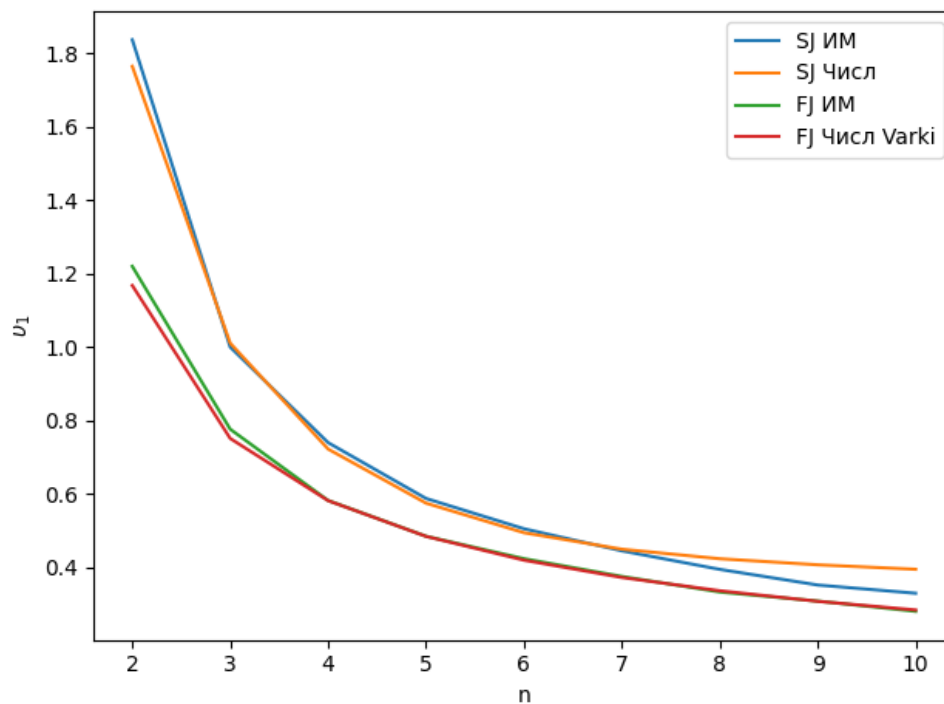
```
# !!! Рассчитываем среднее время пребывания в Fork-Join
# на основе аппроксимации согласно вашего варианта
v_ch = calc_approximation(1, mu, n)
```

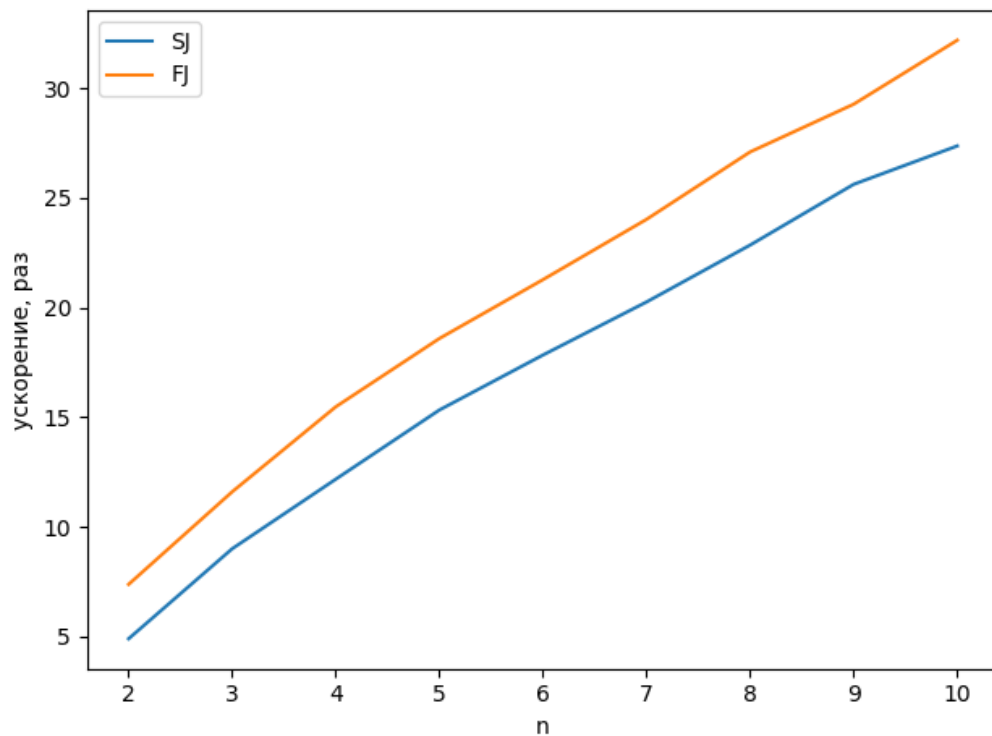
Реализуйте соответствующую формулу для аппроксимации вместо строчки *pass*.

```
def calc_approximation(l, mu, n):
    """
    Напишите функцию расчета среднего времени пребывания заявок
    в СМО Fork-Join с помощью аппроксимации, в зависимости от вашего
    варианта:
    - Nelson R. и Tantawi A.N.
    - Varki E., Merchant A. и Chen H.
    - Varma S. и Makowski A.M.
    """
    pass
```

Варианту с аппроксимацией Nelson R. и Tantawi A.N. соответствует формула (2), варианту с аппроксимацией Varki E., Merchant A. и Chen H. – (3), варианту с аппроксимацией Varma S. и Makowski A.M. – (4).

После реализации формулы запустите программу на выполнение. Должны получиться графики, подобные представленным ниже





Контрольные вопросы

1. Чем отличаются дисциплина обслуживания Split-Join от дисциплины Fork-Join?
2. Какое распределение времени обслуживания задано у моделей, исследуемых в ходе ПЗ? Какое распределение входного потока?
3. Почему относительная ошибка оценки среднего времени пребывания заявок в системе при увеличении числа каналов уменьшается?
4. Существуют ли на сегодняшний день методы, позволяющие точно рассчитать временные характеристики СМО с дисциплинами Split-Join и Fork-Join?
5. Что вы можете сказать о характере зависимости среднего времени пребывания заявок в системе при увеличении числа каналов? Какое число каналов имеет смысл использовать, учитывая, что добавление новых каналов приводит к дополнительным издержкам (например, если канал представляет собой отдельный вычислительный узел, то издержки связаны с затратами на его приобретение или аренду, плату за потребляемую электроэнергию, обслуживание и т.д.)?

Литература

1. *Lakshman A., Malik P.* Cassandra: A decentralized structured storage system // *Operating Systems Review*. – 2010. – Vol. 44(2). – P. 35-40. – DOI:10.1145/1773912.1773922.
2. *DeCandia G., Hastorun D., Jampani M., Kakulapati G., Lakshman A., Pilchin A., Sivasubramanian S., Vosshall P. and Vogels W.* Dynamo:Amazon's highly available key-value store. // *ACM SIGOPS symposium*. – 2007. – P. 205–220. – DOI: 10.1145/1294261.1294281.
3. *Alomari F, Menasce D.A.* Efficient Response Time Approximation for Multiclass Fork and Join Queues in Open and Closed Queueing Networks // *IEEE Transaction on Parallel and Distributed Systems*. – 2014. – Vol. 25. – P.1437-1446. – DOI: 10.1109/TPDS.2013.70.
4. *Baccelli F., Makowski A.M., Shwartz A.* The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds // *Advanced in Applied Probability*. – 1989. – Vol. 21. – P. 629-660. – DOI: 10.2307/1427640
5. *Baccelli F.* Two parallel queues created by arrivals with two demands. The M/G/2 symmetrical case // *Technical report INRIA-Rocquencourt*. – 1985. – Vol. 426.
6. *Flatto L. Hahn S.* Two parallel queues created by arrivals with two demands // *SIAM Journal on Applied Mathematics*. – 1979. – Vol. 44. – P.1041-1053. – DOI: 10.1137/0144074.
7. *Khabarov R. S., Lokhvitckii V. A., Dudkin A. S.* Relationship invariants based sojourn time approximation for the fork-join queueing system / // *CEUR Workshop Proceedings: Proceedings of Models and Methods of Information Systems Research Workshop 2019*. – 2020. – P. 63-68.
8. *Nelson R., Tantawi A.N.* Approximate analysis of fork/join synchronization in parallel queues / // *IEEE Transactions on Computers*. – 1988. – Vol. 37. – P. 739-743. – DOI: 10.1109/12.2213.
9. *Qiu Z., Perez J.G., Harrison P.G.* Beyond the mean in fork-join queues: Efficient approximation for response-time tails // *Performance evaluations*. – 2015. – Vol. 91. – P 99-106. – DOI: 10.1016/j.peva.2015.06.007.

10. *Varki E., Merchant A., Chen H.* The M/M/1 Fork-Join Queue with Variable Subtasks. [electronic resource] <http://www.cs.inh.edu/~varki/publication/2002-nov-open.pdf>
11. *Varma S., Makowski A.M.* Interpolation Approximations for Symmetric Fork-Join Queues // Performance Evaluation. –1994. – Vol. 20. – P. 245-265. – DOI: 10.1016/0166-5316(94)90016-7.
12. *Wright P.E.* Two parallel processors with coupled inputs // Advances in Applied Probability. – 1992. – Vol. 24. – P. 986-1007. – DOI: 10.2307/1427722.
13. *Рыжиков Ю.И., Лохвицкий В.А., Хабаров Р.С.* Метод расчета длительности обработки задач в системе массового обслуживания с учетом процессов Split-Join // Известия высших учебных заведений. Приборостроение. – 2019. – № 62(5). – С. 419-423.
14. *Хабаров Р. С., Лохвицкий В. А., Дудкин А. С.* Аппроксимация времени пребывания для системы массового обслуживания Fork-Join на основе инвариантов отношения // Интеллектуальные технологии на транспорте. – 2020. – № 2(22). – С. 46-50.
15. *Хабаров Р.С., Лохвицкий В.А.* Модель оценивания оперативности многопоточной обработки задач в распределенной вычислительной среде с учетом процессов Split-Join // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2019. – № 1. – С. 26-34. – DOI: 10.25586/RNU.V9187.19.01.P.026.
16. *Хабаров Р. С., Лохвицкий В. А., Корчагин П. В.* Расчет временных характеристик системы массового обслуживания с процессами расщепления и слияния заявок и разогревом / // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2021. – № 2. – С. 10-19. – DOI 10.25586/RNU.V9187.21.02.P.010.