

# Минимизация среднего времени пребывания в ненаблюдаемых системах с параллельным обслуживанием и дисциплиной справедливого разделения процессора в серверах<sup>1</sup>

М.Г. Коновалов, Р.В. Разумчик

Федеральный исследовательский центр “Информатика и управление” Российской академии наук,  
Москва, Россия, Российский университет дружбы народов, Москва, Россия

Поступила в редколлегию 29.07.2019

**Аннотация**—Рассматривается задача эффективного распределения единственного потока однородных заданий в “почти” ненаблюдаемых системах, состоящих из параллельно работающих серверов с очередями неограниченной емкости и одного диспетчера, немедленно распределяющего поступающие по одному задания по серверам. Эффективность понимается в смысле минимума стационарного среднего времени пребывания задания в системе. Для выбора сервера диспетчеру доступна лишь следующая информация: производительность серверов, распределение времени между поступлениями заданий, распределение их размеров, полная история совершенных действий и промежутки между ними. Динамическая информация о состоянии системы (как, например, длина очередей) диспетчеру неизвестна. Предполагается, что в серверах реализована дисциплина справедливого разделения процессора. Из классические диспетчеризаций для таких систем применимы лишь: случайный выбор и циклический выбор. В работе предлагается семейство новых стратегий, которые используют, помимо априорной информации о системе, оценки состояния серверов по результатам имеющихся наблюдений. На численных примерах показано, что, по сравнению с известными стратегиями, новые стратегии позволяют существенно уменьшить среднее время пребывания.

**КЛЮЧЕВЫЕ СЛОВА:** системы с параллельным обслуживанием; дисциплина справедливого разделения процессора; диспетчеризация; стратегии размещения заданий; управление при неполном наблюдении.

## 1. ВВЕДЕНИЕ

Статья посвящена проблеме поиска оптимальных с точки зрения минимального стационарного среднего времени пребывания стратегий (диспетчеризаций, алгоритмов) в “почти” ненаблюдаемых<sup>1</sup> системах параллельной обработки заданий. Такие системы представляют собой совокупность параллельно и независимо друг от друга работающих обслуживающих ресурсов (серверов), которые выполняют задания, направляемые на них диспетчером. При этом диспетчер, осуществляя выбор сервера для выполнения очередного задания, не имеет возможности отложить<sup>2</sup> решение и может руководствоваться только

- априорной информацией о системе, т.е. распределениями времен между поступлениями заданий, распределениями размеров заданий, производительностями серверов, информацией о дисциплине обслуживания заданий в каждом сервере;

<sup>1</sup> Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 18-07-00692).

<sup>1</sup> С практической точки зрения данные системы находят применение при моделировании систем облачных вычислений, grid-систем и систем добровольных вычислений (см., например, [1, 2]).

<sup>2</sup> Т.е. у диспетчера отсутствует очередь для хранения поступающих заданий, позволяющая осуществлять отложенный выбор.

– информацией о моментах времени поступления предыдущих заданий и о принятых решениях.

В таких условиях, в отсутствии динамической информации о состоянии системы (например, о числе заданий на серверах, об остаточных временах обслуживания, о размерах заданий и др.), возникает проблема построения оптимальной процедуры диспетчеризации. Оптимальность понимается<sup>3</sup> в смысле минимума стационарного среднего времени пребывания задания в системе — пожалуй, наиболее популярной целевой функции для распределенных компьютерных систем передачи, обработки и хранения данных. Дальнейшие содержательные рассуждения о построении правил диспетчеризации невозможны без уточнения дисциплины обслуживания заданий внутри серверов. Предполагается, что выполнение заданий в каждом сервере происходит в соответствии с дисциплиной EPS (Egalitarian Processor Sharing), т.е. дисциплиной равномерного (справедливого) деления процессора (см., например, [3, С. 55]). Отметим, что случай, когда в серверах используется дисциплина FIFO был подробно изучен в [4, 14]. Однако предлагаемое там решение перестает “работать” при переходе от дисциплины FIFO к EPS, что связано с тем, что при дисциплине EPS не удается выписать рекуррентные соотношения для времени пребывания в системе поступающего задания.

Из всех известных из научной периодической печати классических диспетчеризаций (см. [5]) в указанных условиях диспетчеру доступны лишь две: случайный выбор и циклический выбор. Подробнее об этих стратегиях будет сказано в следующем разделе (см. (1) и (2)). Отметим, что эти стратегии являются оптимальными в узком диапазоне значений исходных параметров модели. Кроме того, для их реализации не используются все доступные наблюдения за системой — предыдущие совершенные действия и моменты их выбора. В результате, с одной стороны, эти стратегии оказываются исключительно легко реализуемыми на практике, что в некоторой степени оправдывает их неоптимальность. С другой стороны, такое положение дел может приводить к существенному проигрышу в целевой функции по сравнению со стратегиями, использующими максимальную текущую информацию. Решение задачи построения именно таких стратегий и численное исследование их качества и является предметом исследования этой статьи.

Прежде чем переходить к формальному описанию задачи необходимо сделать несколько замечаний. Обычно проблемы диспетчеризации являются сложными, и относящиеся к ним задачи чаще всего не удается решить исключительно математическими методами. Поэтому нередким явлением является применение методов статистического моделирования, эвристических идей и инженерного подхода для получения удовлетворительных решений. Рассматриваемая задача не является исключением, и все предлагаемые здесь решения опираются на метод статистических испытаний. Идея построения новых алгоритмов для диспетчеризации в “почти” ненаблюдаемых системах перекликается с приемом, используемым в теории адаптивного управления и известным под названием идентификационный подход. Суть последнего в следующем. За основу берется алгоритм, успешно действующий по отношению к конкретному объекту, если “правильно” выбрать параметры алгоритма. Однако выбор параметров зависит от свойств объекта, которые могут быть априори неизвестны. В этом случае иногда удается осуществить настройку параметров непосредственно в процессе управления, основываясь на “идентификации” по результатам наблюдений. В рассматриваемой ситуации идентифицируют-

<sup>3</sup> Предлагаемые в статье решения являются достаточно универсальными и позволяют строить процедуры и для других критериев оптимальности (например, дисперсии времени пребывания, квантиля времени пребывания заданного уровня).

ся не параметры “основного” алгоритма<sup>4</sup>, а необходимые для его реализации, но недоступные для наблюдения, динамические характеристики серверов.

## 2. ПОСТАНОВКА ЗАДАЧИ

В систему из  $M$  серверов поступает рекуррентный входной поток заданий<sup>5</sup>, так что интервалы времени между поступлениями заданий образуют последовательность независимых случайных величин с одинаковым распределением. Задания имеют случайную длину (объем), которая определяется одним и тем же распределением. Каждое поступившее задание должно быть немедленно направлено на один из серверов, причем диспетчеру, осуществляющему этот выбор (в автоматическом или ручном режиме), недоступна какая-либо информация о текущем (прошлом или будущем) состоянии системы (например, размеры очередей, остаточная работа на каждом сервере и т.п.); известно лишь распределение входного потока, распределение длины поступающих заданий, скорости серверов и полная предыстория принятых решений (с указанием моментов времени, в которые решения принимались). Количество заданий, которые могут одновременно выполняться на сервере, не ограничено. Выполнение заданий в каждом сервере происходит в соответствии с дисциплиной EPS, т.е. дисциплиной равномерного (справедливого) разделения процессора. С учетом последнего предположения без ограничения общности можно считать, что в каждом сервере имеется один процессор для обработки заданий. Серверы работают независимо, без обмена заданиями.

Описанная схема приводит к следующей математической модели. Обозначим через  $F$  функцию распределения длины интервала между последовательными поступлениями заданий, а через  $G$  — функцию распределения длины задания. Производительность (интенсивность обслуживания) сервера  $i$  обозначим через  $v^{(i)}$ . Система функционирует в непрерывном времени  $t \geq 0$ . Пусть  $0 \leq t_1 < t_2 < \dots$  — последовательность моментов поступления заданий в систему, а  $\Delta_n = t_{n+1} - t_n$  — промежутки между этими моментами. Решение (действие), принимаемое в момент  $t_n$  относительно вновь поступившего задания, обозначим через  $y_n$ . Полагаем, что  $y_n = i$ , если задание направлено на сервер  $i$ . Пусть задание, поступившее в момент  $t_n$  и обслуженное согласно правилу  $y_n$ , проводит в системе (неизвестное) время, равное  $w_n$ . Ставится задача отыскания такой стратегии  $y = \{y_n, n = 1, 2, \dots\}$ , которая минимизировала бы предельное среднее время пребывания заданий в системе, определяемое как

$$W = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E_y(w_n),$$

где  $E_y(\cdot)$  — интегрирование по мере, порождаемой последовательностью  $y$ , а существование предела обусловлено структурой этой последовательности, которая обычно выбирается из “разумных соображений” достаточно регулярной. Стратегию  $y$  будем иногда называть диспетчеризацией, а ее элементы  $y_n$  — правилами диспетчеризации.

Для решения поставленной задачи необходимо указать множество допустимых диспетчеризаций, на котором осуществляется минимизация. Удобно интерпретировать это множество с точки зрения возможностей наблюдения за траекторией процесса (подробнее см. [4, Раздел 2]). Если обозначить через  $h^n$  совокупность наблюдаемых параметров системы до момента принятия решения  $t_n$ , то допустимое правило диспетчеризации имеет вид  $y_n = f(h^n)$ , где  $f$  — рандомизированная или детерминированная функция со значениями  $\{1, 2, \dots, M\}$ , а  $h^n$

<sup>4</sup> В качестве которого рассматриваются различные алгоритмы, хорошо зарекомендовавшие себя в наблюдаемых системах параллельной обработки заданий.

<sup>5</sup> Для простоты изложения предполагается, что в систему поступает один поток однородных заданий. Однако нет никакой принципиальной сложности в применении полученных результатов для системы с несколькими входящими потоками разнородных заданий.

принимает значения из некоторого множества наблюдений  $H^n$ . Из известных “классических” диспетчеризаций в системах с параллельным обслуживанием (см. [5]), для рассматриваемой ненаблюдаемой системы с дисциплиной EPS в серверах применимы лишь только следующие два правила:

$$y_n = f(p_1, \dots, p_M) \quad (1)$$

и

$$y_n = f(y_{n-M+1}, \dots, y_{n-1}). \quad (2)$$

Правило (1) подразумевает отсутствие каких-либо наблюдений, т.е.  $H^n = \emptyset$ ; допустимая стратегия<sup>6</sup> описывается набором  $(p_1, \dots, p_M)$  из  $M$  чисел – вероятностей  $p_i$  выбора для очередного задания сервера  $i$ . Известен ряд результатов, касающихся ее оптимальности, используя которые можно численно находить значения  $p_i$  (см., например, [6–10]). Наиболее полно задача решается для полностью марковских систем и систем с входящим пуассоновским потоком заданий и серверов типа  $M/G/1$ . В общих случаях, если и доступны, то лишь приближенные и эвристические решения.

Согласно правилу (2) предыстория к моменту  $t_n$ ,  $n \geq M$ , определяется значением из множества  $H^n = \{1, 2, \dots, M\}^{M-1}$ . Здесь интерес представляют детерминированные стратегии<sup>7</sup>, т.е. стратегии, параметризуемые бесконечными последовательностями  $\{a_1, a_2, \dots, a_{n-1}, a_n, \dots\}$ , в которых  $a_i$  означает, что  $i$ -е задание направляется на сервер с номером  $a_i$ . Судя по публикациям в научной периодической печати, правил нахождения оптимальной детерминированной стратегии для ненаблюдаемых систем с дисциплиной EPS в серверах до сих пор не разработано<sup>8</sup>. Однако, циклическая стратегия<sup>9</sup> – частный случай детерминированной стратегии – является разумным<sup>10</sup> выбором, хотя обычно и не является оптимальной.

### 3. ДИСПЕТЧЕРИЗАЦИЯ ПО ПРЕДЫСТОРИИ ДЕЙСТВИЙ, МОМЕНТАМ ПОСТУПЛЕНИЯ И ОСНОВАННЫХ НА НИХ СТАТИСТИКАХ

Отметим, что правила (1) и (2) из всей доступной информации для выбора управления в момент поступления  $n$ -го задания (а это последовательности  $\{y_1, \dots, y_{n-1}\}$  и  $\{t_1, \dots, t_n\}$ ) используется разве что информацию о последних  $(M - 1)$  решениях –  $\{y_{n-M+1}, \dots, y_{n-1}\}$ . Однако поскольку диспетчеру известно не только “что было сделано”, но и “когда было сделано”, то хотелось бы иметь в распоряжении и стратегии, которые учитывают всю доступную информацию<sup>11</sup>. О таких новых стратегиях речь пойдет дальше.

<sup>6</sup> В зарубежной литературе – PAP (Probabilistic Allocation Policy) или RND (Random), или BS (Bernoulli Splitting).

<sup>7</sup> Такое положение вещей связано с интуитивным представлением о том, что входящий поток на каждый сервер при детерминированной стратегии является более регулярным (т.е. “менее” случайным), чем при вероятностной, что может приводить к уменьшению значения среднего времени пребывания в системе.

<sup>8</sup> Стоит отметить, что для систем из серверов с дисциплиной FIFO, удается иногда построить такие детерминированные последовательности, результаты которых не удается улучшить, не привлекая при диспетчеризации дополнительную информацию о системе. В подобных случаях они, по-видимому, близки к оптимальным (см., подробнее в [11–14]).

<sup>9</sup> Пожалуй, наиболее известный вариант такой стратегии – Round Robin (далее – RR):  $n$ -е поступающее задание направляется на сервер с номером  $(n \bmod M) + 1$ .

<sup>10</sup> Ввиду того, что входящий поток на каждый сервер при детерминированной стратегии является более регулярным (“менее” случайным). Однако даже в слабо загруженных системах, циклическая стратегия может приводить к неограниченным значениям  $W$  (см., например, табл. 2). Это связано с тем, что загрузка одного или нескольких серверов при такой стратегии становится больше единицы.

<sup>11</sup> Сразу не вполне ясно, позволит ли учет дополнительной информации уменьшить значение  $W$ . Этому есть интуитивные предпосылки, см. [4, С. 61]. Численные эксперименты, некоторые результаты которых представлены в разделе 4, оправдывают их.

Предположим, что рассматриваемая система полностью наблюдаема, т.е. диспетчеру доступна любая информация о текущем состоянии системы и, значит, доступны следующие правила диспетчеризации:

- $y_n = \operatorname{argmin}_{1 \leq i \leq M} \left( l_n^{(i)} \right)$ , где  $l_n^{(i)}$  — размер очереди на сервере  $i$  в момент принятия решения  $t_n$ . Это известная диспетчеризация JSQ, направляющая задание на сервер с минимальной очередью;
- $y_n = \operatorname{argmin}_{1 \leq i \leq M} \left( u_n^{(i)} \right)$ , где  $u_n^{(i)}$  — время, необходимое для полного выполнения всех заданий, которые в момент принятия решения  $t_n$  находятся на сервере  $i$ , при условии, что новые задания не поступают. Если в момент  $t_n$  на сервере  $i$  находится  $k$  заданий, причем их длины равны  $l_1, \dots, l_k$ , то  $u_n^{(i)} = \sum_{j=1}^k l_j / v^{(i)}$ . Эту диспетчеризацию назовем MRT (Minimal Remaining Time);
- $y_n = \operatorname{argmin}_{1 \leq i \leq M} \left( \omega_n^{(i)} \right)$ , где  $\omega_n^{(i)}$  — числовая характеристика сервера  $i$  в момент принятия решения  $t_n$ , вычисляемая следующим образом. Если в момент  $t_n$  на сервере  $i$  находится  $k$  заданий, причем их упорядоченные по возрастанию длины равны  $l_1, \dots, l_k$ , то

$$\omega_n^{(i)} = \sum_{i=1}^k \tau_i C^{k-i}, \quad \tau_i = \frac{k-i+1}{v^{(i)}} \left( l_i - \sum_{j=1}^{i-1} \frac{k-j+1}{v^{(i)}} l_j \right), \quad (3)$$

где  $C$  — фиксированное положительное число. Эту эвристическую диспетчеризацию назовем MHV(C) (Minimal Heuristic Value).

Заметим, что если функция  $\operatorname{argmin}$  возвращает несколько значений, то поступившее задание направляется на самый быстрый сервер; если таких несколько, то из них сервер выбирается наугад.

Поскольку в описанных выше трех диспетчеризациях речь идет о моменте поступления задания, то оценки  $l_n^{(i)}$ ,  $u_n^{(i)}$  и  $\omega_n^{(i)}$  состояния сервера  $i$  возможны, вообще говоря, как без учета нового задания, так и в предположении, что оно отправлено именно на данный сервер. В последнем варианте будем снабжать обозначения дополнительной чертой сверху. Например, запись  $\bar{l}_n^{(i)}$  означает, что в момент поступления задания в систему оценка состояния сервера  $i$  трактуется как текущий размер очереди на сервере  $i$  плюс новое задание. Тогда, если в правиле диспетчеризации JSQ заменить  $l_n^{(i)}$  на  $\bar{l}_n^{(i)} - l_n^{(i)}$ , получим новое правило, которое, допуская некоторую вольность речи, можно назвать диспетчеризацией по минимальному увеличению размера очереди. Аналогично, заменяя в правиле диспетчеризации MRT  $u_n^{(i)}$  на  $\bar{u}_n^{(i)} - u_n^{(i)}$ , получим правило минимального увеличения оставшегося времени (далее — MRTI, Minimal Remaining Time Increase). Наконец, заменяя в правиле диспетчеризации MHV(C)  $\omega_n^{(i)}$  на  $\bar{\omega}_n^{(i)} - \omega_n^{(i)}$ , получаем правило минимального увеличения эвристической оценки (далее — MHVI(C), Minimum Heuristic Value Increase).

Вернемся к основной постановке задачи, согласно которой диспетчеру недоступна динамическая информация о состоянии системы: ни размеры очереди, ни объемы заданий и т.п. Идея заключается в том, чтобы, взяв за основу любой из алгоритмов, описанных в этом разделе, заменять отсутствующие данные статистическими оценками, которые получаются на основе доступных наблюдений. Опишем общее правило, на основе которого конструируются новые диспетчеризации.

Пусть в некоторый момент  $t_n$  в систему поступило новое задание, которое необходимо направить на какой-либо сервер  $y_n$ . Пусть  $h_{n-d}^n = (y_{n-d}, \Delta_{t-d}, y_{n-d+1}, \Delta_{n-d+1}, \dots, y_{n-1}, \Delta_{n-1})$  — предыстория к моменту  $t_n$  глубины  $d$ , составленная из имеющихся наблюдений, то есть из

управлений  $y_k$  и промежутков  $\Delta_k$  между моментами совершения действий  $y_k$  и  $y_{k-1}$ ,  $n-d \leq k \leq n-1$ . Пусть также  $G_1, G_2, \dots, G_d$  —  $d$  независимых реализаций случайной величины, имеющей (известную) функцию распределения объема заданий  $G$ .

С помощью вектора наблюдений  $\tilde{h}^d = h_{n-d}^n = (\tilde{y}_1, \tilde{\Delta}_1, \dots, \tilde{y}_d, \tilde{\Delta}_d)$ , где  $\tilde{y}_l = y_{l+n-d-1}$ ,  $\tilde{\Delta}_l = \Delta_{l+n-d-1}$ ,  $1 \leq l \leq d$ , а также с помощью чисел  $G_1, G_2, \dots, G_d$  симулируем независимый от основного процесса отрезок траектории следующим образом. В начальный момент в пустую (вспомогательную) систему поступает задание объемом  $G_1$ , которая направляется на сервер  $\tilde{y}_1$ . Спустя время  $\tilde{\Delta}_1$  поступает задание объемом  $G_2$ , которая направляется на сервер  $\tilde{y}_2$ , и так далее, вплоть до поступления задания объемом  $G_d$ . Задания обрабатываются точно так же как в основной системе. Управление  $\tilde{y}_{d+1}$  выбирается на основе правила диспетчеризации, которое заложено в алгоритме JSQ, MRT, MHV(C), MRTI или MHVI(C). Обозначим это управление  $y^{(1)}$ .

Повторим процедуру из предыдущего абзаца точно таким же образом еще  $(r-1)$  раз, где  $r$  — параметр алгоритма. В результате получим набор управлений  $y = (y^{(1)}, \dots, y^{(r)})$ . В качестве искомого управления  $y_n$  в основной системе, примем то управление  $y^{(i)}$ , которое встречается в наборе  $y$  наиболее часто. Если индексов  $i$  окажется несколько, то применяется равновероятный выбор.

Таким образом, каждому алгоритму JSQ, MRT, MHV(C), MRTI, MHVI(C), которые доступны диспетчеру при полном наблюдении системы, соответствует “приближенный” алгоритм, обозначаемый далее soJSQ<sup>12</sup>, soMRT, soMHV(C), soMRTI, soMHVI(C) соответственно, действующий в условиях неполного наблюдения. Например, алгоритм soJSQ направляет задания на тот сервер, который чаще других был выбран в описанной выше серии из  $r$  имитаций.

#### 4. РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ РАСЧЕТОВ

Проверка большого числа примеров позволила установить следующий экспериментальный факт: новые стратегии всегда показывают такой выигрыш в целевой функции по сравнению со стратегиями (1) и (2), который может оправдать их применение на практике. Ниже приводятся результаты трех серий экспериментов для стратегий soJSQ, soMRT, soMHVI(C), которые дают общее представление об их “поведении” при всевозможных конфигурациях системы.

##### 4.1. Первая серия экспериментов

В этой серии экспериментов фиксировались функции распределения входящего потока и длины задания, а число серверов  $M$  и загрузка системы варьировались. Входящий в систему поток предполагался пуассоновским. В качестве распределения длины поступающих заданий рассматривались следующие варианты (среднее значение всюду равно 1):

- вырожденное распределение — длина всех заданий равна 1;
- равномерное распределение на отрезке  $[0, 5; 1, 5]$  с дисперсией 0,08(3);
- нормальное распределение с дисперсией 0,01;
- экспоненциальное распределение с параметром 1;
- гиперэкспоненциальное распределение с дисперсией 1,5 (функция распределения  $G(x) = 1 - q_1 e^{-\mu_1 x} - (1 - q_1) e^{-\mu_2 x}$ , значения  $(q_1, \mu_1, \mu_2)$  приводятся в таблицах);
- распределение Парето с дисперсией 1/3 (функция распределения  $G(x) = 1 - (x_m/x)^k$ ,  $x \geq x_m$ , значения  $(k, x_m)$  приводятся в таблицах);
- распределение Вейбулла с дисперсией 6 (функция распределения  $G(x) = 1 - e^{-ax^b}$ ; значения  $(a, b)$  приводятся в таблицах).

<sup>12</sup> Приставка so- от англ. scant observation — “скудные наблюдения”.

В таблицах 1–5 приводятся значения целевой функции  $W$  для девяти диспетчеризаций (RND, RR, JSQ, soJSQ, MRT, soMRT, MHV, MHVI, soMHVI) при указанных распределениях длин заданий и при разных значениях загрузки системы.

Напомним, что под стратегией RND понимается описанный в разделе 2 алгоритм, в котором серверы выбираются с фиксированной вероятностью. В этой серии экспериментов предполагается, что этот параметр выбран оптимальным образом. Стратегия RR представляет собой упомянутую в разделе 2 дисциплину циклического выбора. Диспетчеризация JSQ направляет задания каждый раз в наиболее короткую очередь. При стратегии MRT (MHVI) очередное задание направляется тот сервер, для которого время (приращение времени) до освобождения минимально, при условии, что новые задания в систему больше не принимаются. Новые правила диспетчеризации soJSQ, soMRT и soMHVI реализуются в соответствии с описанием в разделе 3. Отметим, что для использования новых алгоритмов soJSQ, soMRT и soMHVI необходимо задать значение параметров  $d$  (глубину предыстории),  $r$  (количество имитируемых траекторий) и  $C$  (константу из формулы (3)). Отыскание оптимальных значений для параметров  $d$ ,  $r$  и  $C$ , по существу, представляет собой отдельную, пока нерешенную проблему. Значения, которые далее указываются в таблицах – это наилучшие значения, которые удалось найти в экспериментах путем подбора.

Роль точки отсчета во всех экспериментах играют стратегии RND и RR (см. (1) и (2)): они являются основными объектами сравнения для новых диспетчеризаций soJSQ, soMRT и soMHVI. Другие стратегии (JSQ, MRT, MHV и MHVI), которые, вообще говоря, неприменимы в ненаблюдаемой системе, служат методологическим целям. Анализируя численные результаты (см., например, первую колонку любой из таблиц 1-9) можно убедиться, что “приближенные” алгоритмы могут не уступать или вовсе превосходить в эффективности точные, основанные на наблюдениях алгоритмы. Интуитивно можно было предположить, что наблюдения должны давать существенные преимущества точным алгоритмам. Это интуитивное предположение, однако, подтверждается только для больших значений дисперсии длины заданий.

Во всех случаях новые диспетчеризации позволяют улучшить значение показателя  $W$  по сравнению со стратегиями RND и RR. Разброс достигаемого выигрыша очень большой: от 1% до 70%. При этом во всех случаях наблюдается заметное уменьшение выигрыша с увеличением дисперсии длины задания.

#### 4.2. Вторая серия экспериментов

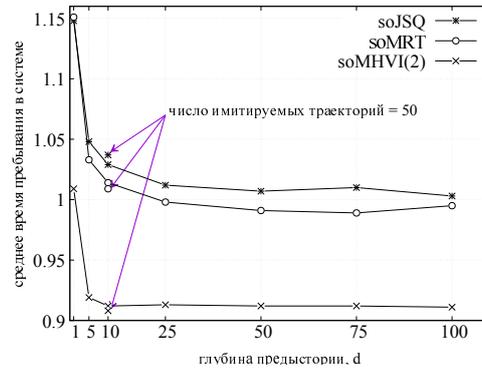
В этой серии экспериментов, как и в предыдущей, фиксировались функции распределения входящего потока и длины задания, а число серверов  $M$  и загрузка системы варьировались. Вместо пуассоновского рассматривался гиперэкспоненциальный входящий поток, допускающий больший разброс во временах между поступлениями заданий. В качестве распределения длин поступающих заданий рассматривались те же варианты, что и в разделе 4.1. Значения целевой функции  $W$  для девяти диспетчеризаций (RND<sup>13</sup>, RR, JSQ, soJSQ, MRT, soMRT, MHV, MHVI, soMHVI) представлены в таблицах 6-9.

С изменением характера входного потока качественная картина “поведения” целевой функции  $W$  в сравнении с первой серией экспериментов не поменялась. Во всех случаях новые диспетчеризации позволяют улучшить значение показателя  $W$  по сравнению со стратегиями RND и RR. При этом сохраняется тенденция уменьшения выигрыша с увеличением дисперсии длины задания.

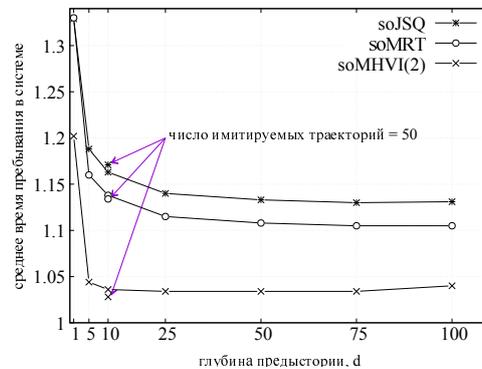
<sup>13</sup> При произвольном рекуррентном входящем потоке для стратегии RND не существует аналитического способа расчета оптимальных (с точки зрения стационарного среднего времени пребывания) значений  $(p_1, \dots, p_M)$  – вероятностей  $p_i$  выбора сервера  $i$  для очередного задания. В такой ситуации вероятность  $p_i$  принималась равной  $v^{(i)} / \sum_{j=1}^M v^{(j)}$ .

## 4.3. Третья серия экспериментов

Целью этой серии экспериментов было выявление влияния значения параметров, определяющих новые диспетчеризации soJSQ, soMRT и soMHVI на значение целевой функции  $W$ . Рассматривалась система из двух серверов,  $v^{(1)} = 2$ ,  $v^{(2)} = 1$ . Входящий в систему поток гиперэкспоненциальный со средним 1, а длины заданий имеют распределение Вейбулла со средним 1. Численные значения среднего времени пребывания  $W$  для стратегий soJSQ, soMRT и soMHVI в зависимости от глубины предыстории  $d$  и значения дисперсии входящего потока представлены на рисунках 1–4.

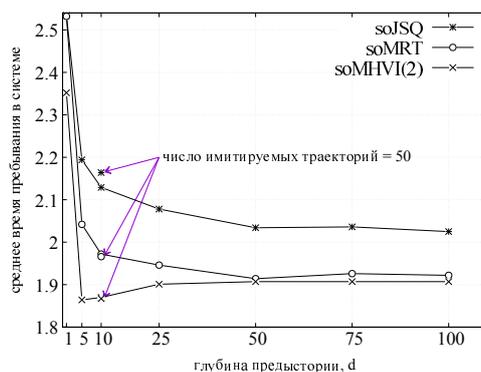


**Рис 1.** Зависимость значения стационарного среднего время пребывания задания в системе от глубины предыстории  $d$  для трех новых диспетчеризаций. Случай двух серверов ( $M = 2$ ). Производительности серверов  $v^{(1)} = 2$ ,  $v^{(2)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 1$  и дисперсией  $\frac{5}{3}$ ;  $F(x) = 1 - 0,75e^{-1,5x} - 0,25e^{-0,5x}$ ,  $x \geq 0$ . Длина заданий имеет распределение Вейбулла с параметрами  $(\frac{140}{99}; 0,5)$ , средним 1 и дисперсией 6. Загрузка системы равна  $\lambda/(v^{(1)} + v^{(2)}) \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

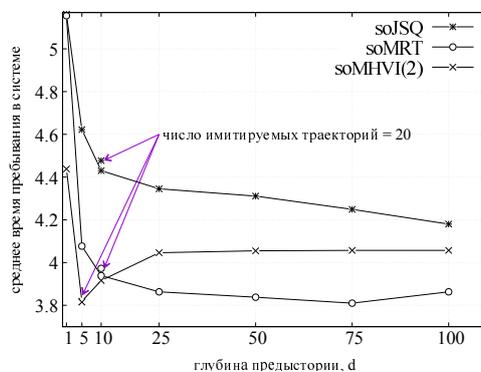


**Рис 2.** Зависимость значения стационарного среднего время пребывания задания в системе от глубины предыстории  $d$  для трех новых диспетчеризаций. Случай двух серверов ( $M = 2$ ). Производительности серверов  $v^{(1)} = 2$ ,  $v^{(2)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 1$  и дисперсией 4;  $F(x) = 1 - 0,8571e^{-2x} - 0,1429e^{-0,25x}$ ,  $x \geq 0$ . Длина заданий имеет распределение Вейбулла с параметрами  $(\frac{140}{99}; 0,5)$ , средним 1 и дисперсией 6. Загрузка системы равна  $\lambda/(v^{(1)} + v^{(2)}) \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

С количественной точки зрения влияние дисперсии входящего потока на значение целевой функции  $W$  существенно: чем больше дисперсия, тем больше и среднее время пребывания



**Рис 2.** Зависимость значения стационарного среднего время пребывания задания в системе от глубины предьстории  $d$  для трех новых диспетчеризаций. Случай двух серверов ( $M = 2$ ). Производительности серверов  $v^{(1)} = 2, v^{(2)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 1$  и дисперсией 15,4;  $F(x) = 1 - 0,918e^{-5x} - 0,082e^{-0,1x}, x \geq 0$ . Длина заданий имеет распределение Вейбулла с параметрами  $(\frac{140}{99}; 0,5)$ , средним 1 и дисперсией 6. Загрузка системы равна  $\lambda/(v^{(1)} + v^{(2)}) \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .



**Рис 2.** Зависимость значения стационарного среднего время пребывания задания в системе от глубины предьстории  $d$  для трех новых диспетчеризаций. Случай двух серверов ( $M = 2$ ). Производительности серверов  $v^{(1)} = 2, v^{(2)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 1$  и дисперсией 35,1;  $F(x) = 1 - 0,9548e^{-10x} - 0,0452e^{-0,05x}, x \geq 0$ . Длина заданий имеет распределение Вейбулла с параметрами  $(\frac{140}{99}; 0,5)$ , средним 1 и дисперсией 6. Загрузка системы равна  $\lambda/(v^{(1)} + v^{(2)}) \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

задания в системе. Кроме того, с ростом дисперсии изменяется и эффективность диспетчеризаций. Так наилучшая при малых значениях дисперсии диспетчеризация soMHVI уступает место диспетчеризации soMRT при больших значениях дисперсии. Однако с качественной точки зрения наблюдается своеобразная нечувствительность новых диспетчеризаций к дисперсии входящего потока: после определенного значения  $d$  (в приведенных выше примерах  $d \approx 25$ ) значение целевой функции  $W$  практически перестает зависеть от глубины предьстории  $d$ .

### 5. ЗАКЛЮЧЕНИЕ

Результаты проведенного анализа указывают на принципиальную возможность уменьшить среднее время пребывания в ненаблюдаемых системах с параллельным обслуживанием и дисциплиной EPS. Для этого можно воспользоваться стратегиями диспетчеризации, учитывающими предьсторию решений и моментов их принятия. Предложен достаточно универсальный метод порождения новых диспетчеризаций, идея которого заключается в том, чтобы, взяв за

основу любой из алгоритмов, применимых в полностью наблюдаемых системах, заменять отсутствующие данные статистическими оценками, которые получаются на основе доступных наблюдений. Во всех экспериментах с разными распределениями входного потока и разными распределениями длины заданий новые стратегии показали выигрыш (до 50%) по сравнению со стратегиями случайного выбора (RND) и циклического выбора (RR) – единственными классическими стратегиями, применимыми в ненаблюдаемых системах с параллельным обслуживанием.

В связи с новыми диспетчеризациями остаются и открытые вопросы, например: какова зависимость новых диспетчеризаций от глубины предыстории и числа имитируемых траекторий? можно ли предложить простое правило для выбора значения этих параметров? Ответы на эти вопросы, а также аналитическое обоснование эффективности новых диспетчеризаций представляется интересным предметом дальнейших исследований.

Наконец укажем, что в [15] намечен отличный от предложенного в этой статье подход к построению алгоритмов распределения заданий, который не основывается на методе статистических испытаний, однако применим только если распределение размера является распределением фазового типа. Обозначая через  $N_n^{(i)}$  – среднее число заданий на сервере  $i$  в момент поступления задания с номером  $n$  (но до прибавления задания к какому-либо серверу), новое правило формулируется так: отправить задание с номером  $n$  на сервер с номером  $\operatorname{argmin}_{1 \leq i \leq M} (E(N_n^{(i)}))$ , где  $E(N_n^{(i)})$  – среднее значение  $N_n^{(i)}$ , вычислять которое, вообще говоря, можно рекуррентным образом вне зависимости от вида функции распределения  $F$ . Исследование плодотворности этого подхода также представляется интересным направлением исследований.

#### СПИСОК ЛИТЕРАТУРЫ

1. Javadi B., Kondo D., Vincent J.-M., Anderson D.P. Discovering statistical models of availability in large distributed systems: An empirical study of seti@home // *IEEE Transactions on Parallel and Distributed Systems*, 2011, vol. 22, no. 11, pp. 1896–1903.
2. Javadi B., Thulasiraman P., Buyya. R. “Cloud resource provisioning to extend the capacity of local resources in the presence of failures”. *14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, 2012, pp. 311–319.
3. Яшков С.Ф. Анализ очередей в ЭВМ. М.: Радио и связь, 1989.
4. Коновалов М.Г., Разумчик Р.В. О размещении заданий на двух серверах при неполном наблюдении. *Информатика и её применения*, 2016, Т. 10, Вып. 4, стр. 57–67.
5. Коновалов М.Г., Разумчик Р.В. Обзор моделей и алгоритмов размещения заданий в системах с параллельным обслуживанием. *Информатика и её применения*, 2015, Т. 9, Вып. 4, стр. 56–67.
6. Combe M.B., Voxma O.J. “Optimization of static traffic allocation policies” // *Theoretical Computer Science*, 1994, vol. 125, no. 1, pp. 17–43.
7. Sethuraman J., Squillante M.S. “Optimal stochastic scheduling in multiclass parallel queues”. *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '99)*, 1999. ACM, New York, NY, USA, pp. 93–102.
8. Tang Ch.S., van Vliet M. “Traffic allocation for manufacturing systems” // *European Journal of Operational Research*, 1994, vol. 75, no. 1, pp. 171–185.
9. Neely M.J., Modiano E. “Convexity in queues with general inputs” // *IEEE Transactions on Information Theory*, 2005, vol. 51, no. 2, pp. 706–714.
10. Bell C.H., Stidham S. “Individual versus social optimization in the allocation of customers to alternative servers” // *Management Science*, 1983, vol. 29, no. 7, pp. 831–839.

11. Altman E., Gaujal B., Hordijk A. “Balanced sequences and optimal routing” // *Journal of the ACM*, 2000, vol. 47, no. 4, pp. 752–775.
12. Anselmi J., Gaujal B., Nesti T. “Control of parallel non-observable queues: Asymptotic equivalence and optimality of periodic policies” // *Stochastic Systems*, 2015, vol. 5, pp. 120–145.
13. Hordijk A., van der Laan D.A. “Periodic routing to parallel queues and billiard sequences” // *Mathematical Methods of Operations Research*, 2004, vol. 59, no. 2, pp. 173–192.
14. Konovalov M., Razumchik R. “Improving routing decisions in parallel non-observable queues” // *Computing*, 2018, vol. 100, no. 10, pp. 1–21.
15. Konovalov M., Razumchik R. “Minimizing mean response time in non-observable distributed systems with processor sharing nodes” // *Communications of the ECMS*, 2019, vol. 33, no. 1, pp. 456–461.

## Minimizing mean response time in non-observable distributed processing systems with nodes, operating under egalitarian processor sharing policy.

**Konovalov M.G., Razumchik R.V.**

**Аннотация**—Consideration is given to the dispatching problem in an non-observable distributed processing system with  $M$ ,  $M \geq 2$ , single server queues operating in parallel, each under the processor sharing discipline. Jobs, having the same job size distribution, arrive one by one to the dispatcher, which immediately routes it to one of the queues. When making a routing decision the dispatcher does not have any online information about the system (like current queues’ sizes, size of the arriving job etc.). The only information available to the dispatcher is: job size distribution, job’s inter-arrival time distribution, server’s speeds, time instants of previously arrived jobs and previous routing decisions. Under these conditions, one is interested in the routing policies which minimize the job’s long-run mean response time. New class of dispatching policies is proposed which, according to the numerical experiments, may significantly outperform all classic dispatching policies available for such a system: (optimal) probabilistic policy and the round robin policy.

**KEYWORDS:** dispatching; server farm; processor sharing; customer assignment; non-observable queues

**Таблица 1.** Стационарное среднее время  $W$  пребывания задания в системе из двух серверов ( $M = 2$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 2, v^{(2)} = 1$ . Входящий поток пуассоновский со средним  $\lambda^{-1} = 1; F(x) = 1 - e^{-x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^2 v^{(i)} \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>H<sub>2</sub></b> (0,5; $\frac{2}{3}, 2$ ) $Var = 1,5$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,915	0,913	0,914	0,914	0,914	0,914	0,914
<b>RR</b>	0,946	0,971	0,974	1,096	1,133	1,002	1,221
<b>JSQ</b>	0,721	0,724	0,724	0,732	0,734	0,727	0,737
<b>soJSQ</b>	0,711 ( $d=2$ )	0,754 ( $d=2$ )	0,763 ( $d=2$ )	0,894 ( $d=5$ )	0,923 ( $d=5$ )	0,790 ( $d=2$ )	0,980 ( $d=12$ )
	0,722 ( $d=5$ )	0,769 ( $d=5$ )	0,877 ( $d=5$ )		0,907 ( $d=10$ )	0,811 ( $d=5$ )	0,987 ( $d=15$ )
<b>MRT</b>	0,711	0,714	0,715	0,729	0,734	0,724	0,747
<b>soMRT</b>	0,711 ( $d=3$ )	0,750 ( $d=2$ )	0,756 ( $d=2$ )	0,872 ( $d=5$ )	0,897 ( $d=10$ )	0,791 ( $d=2$ )	0,973 ( $d=10$ )
<b>MHV</b>	0,698 ( $C=2$ )	0,699 ( $C=3$ )	0,699 ( $C=2$ )	0,701 ( $C=2$ )	0,703 ( $C=2$ )	0,703 ( $C=2$ )	0,707 ( $C=2$ )
<b>MHVI</b>	0,698 ( $C=2,25$ )	0,699 ( $C=3$ )	0,700 ( $C=2,2$ )	0,699 ( $C=2,25$ )	0,698 ( $C=2,25$ )	0,702 ( $C=2,25$ )	0,696 ( $C=2,25$ )
<b>soMHVI</b>	0,698 ( $d=2, C=2$ )	0,733 ( $d=2, C=2,75$ )	0,721 ( $d=2, C=2,75$ )	0,808 ( $d=5, C=2,5$ )	0,825 ( $d=5, C=2,75$ )	0,739 ( $d=2, C=2,75$ )	0,875 ( $d=5, C=2,75$ )

**Таблица 2.** Стационарное среднее время  $W$  пребывания задания в системе из четырех серверов ( $M = 4$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 4, v^{(2)} = 3, v^{(3)} = 2, v^{(4)} = 1$ . Входящий поток пуассоновский со средним  $\lambda^{-1} = 0,2; F(x) = 1 - e^{-5x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^4 v^{(i)} \approx 0,5$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>H<sub>2</sub></b> (0,5; $\frac{2}{3}, 2$ ) $Var = 1,5$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,711	0,711	0,711	0,711	0,711	0,711	0,711
<b>RR</b>	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
<b>JSQ</b>	0,411	0,414	0,414	0,420	0,422	0,415	0,424
<b>soJSQ</b>	0,421 ( $d=5$ )	0,508 ( $d=5$ )	0,525 ( $d=5$ )	0,737 ( $d=10$ )	0,852 ( $d=10$ )	0,555 ( $d=10$ )	0,917 ( $d=20$ )
		0,487 ( $d=10$ )	0,496 ( $d=10$ )	0,684 ( $d=15$ )	0,741 ( $d=20$ )	0,528 ( $d=20$ )	
<b>MRT</b>	0,402	0,406	0,405	0,417	0,420	0,410	0,431
<b>soMRT</b>	0,402 ( $d=8$ )	0,457 ( $d=10$ )	0,464 ( $d=10$ )	0,665 ( $d=15$ )	0,708 ( $d=20$ )	0,517 ( $d=20$ )	0,905 ( $d=20$ )
<b>MHV</b>	0,380 ( $C=2,25$ )	0,382 ( $C=2,25$ )	0,382 ( $C=2,25$ )	0,385 ( $C=2,25$ )	0,388 ( $C=2,25$ )	0,384 ( $C=2,25$ )	0,392 ( $C=2,25$ )
<b>MHVI</b>	0,381 ( $C=2,5$ )	0,383 ( $C=2,5$ )	0,383 ( $C=2,5$ )	0,383 ( $C=2,5$ )	0,384 ( $C=2,5$ )	0,385 ( $C=2,5$ )	0,384 ( $C=2,5$ )
<b>soMHVI</b>	0,382 ( $d=8, C=2,5$ )	0,414 ( $d=10, C=2,75$ )	0,416 ( $d=10, C=2,75$ )	0,555 ( $d=15, C=2,75$ )	0,583 ( $d=20, C=3,5$ )	0,449 ( $d=20, C=3,5$ )	0,707 ( $d=20, C=4$ )

**Таблица 3.** Стационарное среднее время  $W$  пребывания задания в системе из восьми серверов ( $M = 8$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 5, v^{(2)} = v^{(3)} = v^{(4)} = 2, v^{(5)} = \dots = v^{(8)} = 1$ . Входящий поток пуассоновский со средним  $\lambda^{-1} = 0,2; F(x) = 1 - e^{-5x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^8 v^{(i)} \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>H<sub>2</sub></b> (0,5; $\frac{2}{3}, 2$ ) $Var = 1,5$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,596	0,596	0,596	0,596	0,596	0,596	0,596
<b>RR</b>	0,745	0,779	0,786	1,074	1,160	0,865	1,378
<b>JSQ</b>	0,383	0,383	0,383	0,385	0,385	0,384	0,387
<b>soJSQ</b>	0,384 ( $d=5$ )	0,446 ( $d=5$ )	0,456 ( $d=5$ )	0,603 ( $d=2$ )	0,629 ( $d=2$ )	0,5 ( $d=5$ )	0,688 ( $d=2$ )
				0,629 ( $d=5$ )	0,666 ( $d=5$ )		0,742 ( $d=5$ )
<b>MRT</b>	0,383	0,383	0,383	0,385	0,386	0,384	0,387
<b>soMRT</b>	0,383 ( $d=12$ )	0,447 ( $d=5$ )	0,456 ( $d=5$ )	0,602 ( $d=2$ )	0,629 ( $d=2$ )	0,490 ( $d=3$ )	0,692 ( $d=2$ )
				0,629 ( $d=5$ )	0,662 ( $d=5$ )	0,499 ( $d=5$ )	0,737 ( $d=5$ )
<b>MHV</b>	0,350 ( $C=2,5$ )	0,349 ( $C=2,5$ )	0,349 ( $C=2,25$ )	0,345 ( $C=2,25$ )	0,343 ( $C=2,25$ )	0,349 ( $C=2,25$ )	0,342 ( $C=2,25$ )
<b>MHVI</b>	0,352 ( $C=2,5$ )	0,351 ( $C=2,5$ )	0,35 ( $C=2,5$ )	0,346 ( $C=2,5$ )	0,343 ( $C=2,5$ )	0,35 ( $C=2,5$ )	0,341 ( $C=2,5$ )
<b>soMHVI</b>	0,350 ( $d=12, C=2,5$ )	0,370 ( $d=12, C=2,75$ )	0,373 ( $d=12, C=2,75$ )	0,465 ( $d=12, C=2,75$ )	0,484 ( $d=12, C=2,75$ )	0,398 ( $d=12, C=2,75$ )	0,538 ( $d=20, C=4$ )

**Таблица 4.** Стационарное среднее время  $W$  пребывания задания в системе из шестнадцати серверов ( $M = 16$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = v^{(2)} = v^{(3)} = 3, v^{(4)} = \dots = v^{(8)} = 2, v^{(9)} = \dots = v^{(16)} = 1$ . Входящий поток пуассоновский со средним  $\lambda^{-1} = 0,1; F(x) = 1 - e^{-10x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^{16} v^{(i)} \approx 0,37$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>H<sub>2</sub></b> (0,5; $\frac{2}{3}, 2$ ) $Var = 1,5$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,816	0,816	0,816	0,816	0,816	0,816	0,816
<b>RR</b>	0,725	0,747	0,752	1,042	1,130	0,839	1,372
<b>JSQ</b>	0,413	0,412	0,413	0,413	0,413	0,413	0,414
<b>soJSQ</b>	0,427 ( $d=5$ )	0,461 ( $d=5$ )	0,468 ( $d=5$ )	0,616 ( $d=5$ )	0,658 ( $d=5$ )	0,509 ( $d=5$ )	0,76 ( $d=5$ )
	0,412 ( $d=10$ )	0,455 ( $d=8$ )	0,461 ( $d=8$ )			0,495 ( $d=7$ )	0,747 ( $d=8$ )
<b>MRT</b>	0,413	0,413	0,413	0,413	0,413	0,413	0,414
<b>soMRT</b>	0,413 ( $d=10$ )	0,446 ( $d=7$ )	0,452 ( $d=7$ )	0,617 ( $d=5$ )	0,658 ( $d=5$ )	0,510 ( $d=5$ )	0,748 ( $d=8$ )
<b>MHV</b>	0,403 ( $C=2,5$ )	0,403 ( $C=2,25$ )	0,403 ( $C=2,25$ )	0,401 ( $C=2,25$ )	0,400 ( $C=2,25$ )	0,403 ( $C=2,25$ )	0,399 ( $C=2,25$ )
<b>MHVI</b>	0,404 ( $C=2$ )	0,404 ( $C=2,5$ )	0,404 ( $C=2,5$ )	0,403 ( $C=2,5$ )	0,402 ( $C=2,5$ )	0,404 ( $C=2,5$ )	0,399 ( $C=2,5$ )
<b>soMHVI</b>	0,403 ( $d=20, C=4$ )	0,441 ( $d=16, C=10$ )	0,440 ( $d=20, C=10$ )	0,703 ( $d=64, C=15$ )	0,750 ( $d=50, C=7$ )	0,522 ( $d=48, C=10$ )	1,169 ( $d=72, C=15$ )

**Таблица 5.** Стационарное среднее время  $W$  пребывания задания в системе из шестидесяти трех серверов ( $M = 63$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 20, v^{(2)} = v^{(3)} = 10, v^{(4)} = \dots = v^{(7)} = 5, v^{(8)} = \dots = v^{(15)} = 3, v^{(16)} = \dots = v^{(31)} = 2, v^{(32)} = \dots = v^{(63)} = 1$ . Входящий поток пуассоновский со средним  $\lambda^{-1} = 0,01; F(x) = 1 - e^{-100x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^{63} v^{(i)} \approx 0,556$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Pareto</b> (5; 0,8) $Var = \frac{1}{15}$	<b>Pareto</b> (2,4; $\frac{7}{12}$ ) $Var = \frac{25}{24}$	<b>Pareto</b> (2,2; $\frac{6}{11}$ ) $Var = \frac{25}{11}$
<b>RND</b>	0,992	0,992	0,992	0,992
<b>RR</b>	$\infty$	$\infty$	$\infty$	$\infty$
<b>JSQ</b>	0,297	0,293	0,294	0,294
<b>soJSQ</b>	$\infty$ ( $d=10$ )	0,494 ( $d=50$ )	$\infty$ ( $d=50$ )	0,979 ( $d=70$ )
	0,298 ( $d=50$ )		0,780 ( $d=70$ )	
<b>MRT</b>	0,298	0,294	0,294	0,294
<b>soMRT</b>	0,300 ( $d=50$ )	0,463 ( $d=60$ )	0,811 ( $d=70$ )	0,906 ( $d=70$ )
<b>MHV</b>	0,297 ( $C=2,5$ )	0,293 ( $C=2,25$ )	0,294 ( $C=2,25$ )	0,294 ( $C=2,25$ )

**Таблица 6.** Стационарное среднее время  $W$  пребывания задания в системе из двух серверов ( $M = 2$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 2, v^{(2)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 1$  и дисперсией 1,5;  $F(x) = 1 - 0,5e^{-\frac{2}{3}x} - 0,5e^{-2x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^2 v^{(i)} \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	1,112	1,105	1,103	1,072	1,098	1,037
<b>RR</b>	1,065	1,091	1,094	1,191	1,113	1,266
<b>JSQ</b>	0,794	0,796	0,796	0,789	0,795	0,773
<b>soJSQ</b>	0,777 ( $d=2$ )	0,821 ( $d=2$ )	0,833 ( $d=2$ )	0,959 ( $d=2$ )	0,851 ( $d=2$ )	1,047 ( $d=5$ )
	0,796 ( $d=5$ )	0,852 ( $d=5$ )	0,855 ( $d=5$ )	0,964 ( $d=5$ )	0,885 ( $d=5$ )	
<b>MRT</b>	0,777	0,780	0,781	0,784	0,790	0,783
<b>soMRT</b>	0,777 ( $d=5$ )	0,817 ( $d=2$ )	0,822 ( $d=2$ )	0,924 ( $d=10$ )	0,854 ( $d=2$ )	1,012 ( $d=10$ )
<b>MHV</b>	0,769 ( $C=2$ )	0,768 ( $C=2$ )	0,768 ( $C=2$ )	0,755 ( $C=2$ )	0,769 ( $C=2$ )	0,740 ( $C=2$ )
<b>MHVI</b>	0,770 ( $C=2,25$ )	0,772 ( $C=2$ )	0,768 ( $C=2,25$ )	0,751 ( $C=2,25$ )	0,767 ( $C=2,25$ )	0,728 ( $C=2,25$ )
<b>soMHVI</b>	0,770 ( $d=2,$ $C=2,25$ )	0,791 ( $d=2,$ $C=2,75$ )	0,794 ( $d=2,$ $C=2,75$ )	0,874 ( $d=10,$ $C=2,75$ )	0,807 ( $d=2,$ $C=2,75$ )	0,904 ( $d=5,$ $C=2,75$ )

**Таблица 7.** Стационарное среднее время  $W$  пребывания задания в системе из четырех серверов ( $M = 4$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 4, v^{(2)} = 3, v^{(3)} = 2, v^{(4)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 0,2$  и дисперсией 0,045;  $F(x) = 1 - 0,5e^{-4x} - 0,5e^{-\frac{20}{3}x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^4 v^{(i)} \approx 0,5$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,818	0,816	0,816	0,812	0,814	0,805
<b>RR</b>	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
<b>JSQ</b>	0,421	0,423	0,424	0,428	0,424	0,429
<b>soJSQ</b>	0,435 ( $d=5$ )	0,522 ( $d=5$ )	0,540 ( $d=5$ )	0,971 ( $d=5$ )	0,607 ( $d=5$ )	1,550 ( $d=5$ )
		0,478 ( $d=20$ )	0,491 ( $d=15$ )	0,682 ( $d=20$ )	0,546 ( $d=15$ )	0,894 ( $d=25$ )
<b>MRT</b>	0,410	0,414	0,414	0,424	0,419	0,435
<b>soMRT</b>	0,411 ( $d=10$ )	0,465 ( $d=15$ )	0,470 ( $d=15$ )	0,655 ( $d=20$ )	0,526 ( $d=15$ )	0,905 ( $d=25$ )
<b>MHV</b>	0,389 ( $C=2$ )	0,391 ( $C=2,25$ )	0,391 ( $C=2,25$ )	0,392 ( $C=2,25$ )	0,393 ( $C=2,25$ )	0,396 ( $C=2,25$ )
<b>MHVI</b>	0,390 ( $C=2,5$ )	0,392 ( $C=2,5$ )	0,398 ( $C=2$ )	0,391 ( $C=2,5$ )	0,393 ( $C=2,5$ )	0,388 ( $C=2,5$ )
<b>soMHVI</b>	0,390 ( $d=10,$ $C=2,75$ )	0,423 ( $d=15,$ $C=3$ )	0,426 ( $d=15,$ $C=3$ )	0,557 ( $d=20,$ $C=4$ )	0,457 ( $d=15,$ $C=3,5$ )	0,689 ( $d=25,$ $C=4$ )

**Таблица 8.** Стационарное среднее время  $W$  пребывания задания в системе из восьми серверов ( $M = 8$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = 5, v^{(2)} = v^{(3)} = v^{(4)} = 2, v^{(5)} = \dots = v^{(8)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 0,2$  и дисперсией 0,045;  $F(x) = 1 - 0,5e^{-4x} - 0,5e^{-\frac{20}{3}x}, x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^8 v^{(i)} \approx 0,33$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}; 0,5$ ) $Var = 6$
<b>RND</b>	0,807	0,809	0,808	0,807	0,807	0,800
<b>RR</b>	0,752	0,788	0,794	1,087	0,873	1,383
<b>JSQ</b>	0,390	0,389	0,389	0,391	0,390	0,389
<b>soJSQ</b>	0,390 ( $d=5$ )	0,455 ( $d=5$ )	0,464 ( $d=5$ )	0,636 ( $d=5$ )	0,508 ( $d=5$ )	0,695 ( $d=2$ )
						0,743 ( $d=5$ )
<b>MRT</b>	0,390	0,389	0,389	0,390	0,390	0,391
<b>soMRT</b>	0,390 ( $d=5$ )	0,454 ( $d=5$ )	0,465 ( $d=5$ )	0,636 ( $d=5$ )	0,507 ( $d=5$ )	0,741 ( $d=5$ )
<b>MHV</b>	0,356 ( $C=2,25$ )	0,354 ( $C=2,25$ )	0,389 ( $C=2,25$ )	0,349 ( $C=2,25$ )	0,354 ( $C=2,25$ )	0,344 ( $C=2,25$ )
<b>MHVI</b>	0,358 ( $C=2,5$ )	0,356 ( $C=2,5$ )	0,356 ( $C=2,5$ )	0,349 ( $C=2,5$ )	0,355 ( $C=2,5$ )	0,343 ( $C=2,5$ )
<b>soMHVI</b>	0,356 ( $d=12,$ $C=2,75$ )	0,377 ( $d=12,$ $C=2,75$ )	0,380 ( $d=12,$ $C=2,75$ )	0,468 ( $d=12,$ $C=2,75$ )	0,403 ( $d=12,$ $C=2,75$ )	0,537 ( $d=20,$ $C=4$ )

**Таблица 9.** Стационарное среднее время  $W$  пребывания задания в системе из шестнадцати серверов ( $M = 16$ ) при различных стратегиях диспетчеризации и различных распределениях длин заданий с фиксированным средним значением 1. Производительности серверов  $v^{(1)} = v^{(2)} = v^{(3)} = 3$ ,  $v^{(4)} = \dots = v^{(8)} = 2$ ,  $v^{(9)} = \dots = v^{(16)} = 1$ . Входящий поток гиперэкспоненциальный со средним  $\lambda^{-1} = 0,1$  и дисперсией 0,012;  $F(x) = 1 - 0,5e^{-7,5x} - 0,5e^{-15x}$ ,  $x \geq 0$ . Загрузка системы  $\lambda / \sum_{i=1}^{16} v^{(i)} \approx 0,37$ . Число имитируемых траекторий  $r = 10$ .

	<b>DET</b> (1) $Var = 0$	<b>Uniform</b> [0,5; 1,5] $Var = \frac{1}{12}$	<b>Normal</b> (1; 0,1) $Var = 0,1$	<b>Exp</b> (1) $Var = 1$	<b>Pareto</b> (3, $\frac{2}{3}$ ) $Var = \frac{1}{3}$	<b>Weibull</b> ( $\frac{140}{99}$ ; 0,5) $Var = 6$
<b>RND</b>	0,952	0,951	0,952	0,946	0,949	0,945
<b>RR</b>	0,728	0,753	0,759	1,047	0,843	1,376
<b>JSQ</b>	0,421	0,419	0,419	0,419	0,420	0,418
<b>soJSQ</b>	0,439 ( $d=5$ )	0,473 ( $d=5$ )	0,478 ( $d=5$ )	0,624 ( $d=5$ )	0,519 ( $d=5$ )	0,765 ( $d=5$ )
	0,421 ( $d=10$ )	0,456 ( $d=7$ )	0,461 ( $d=7$ )		0,505 ( $d=7$ )	0,741 ( $d=10$ )
<b>MRT</b>	0,421	0,420	0,420	0,418	0,420	0,418
<b>soMRT</b>	0,420 ( $d=10$ )	0,462 ( $d=8$ )	0,468 ( $d=8$ )	0,626 ( $d=5$ )	0,519 ( $d=5$ )	0,739 ( $d=10$ )
<b>MHV</b>	0,409 ( $C=2,25$ )	0,408 ( $C=2,25$ )	0,419 ( $C=2,25$ )	0,405 ( $C=2,5$ )	0,408 ( $C=2,25$ )	0,4 ( $C=2,25$ )
<b>MHVI</b>	0,411 ( $C=2,5$ )	0,410 ( $C=2,5$ )	0,409 ( $C=2,5$ )	0,406 ( $C=2,5$ )	0,409 ( $C=2,5$ )	0,401 ( $C=2,5$ )
<b>soMHVI</b>	0,408 ( $d=20,$ $C=4$ )	0,447 ( $d=16,$ $C=10$ )	0,449 ( $d=20,$ $C=15$ )	0,670 ( $d=80,$ $C=15$ )	0,515 ( $d=64,$ $C=15$ )	1,055 ( $d=80,$ $C=20$ )