

М. Г. Коновалов, Р. В. Разумчик (Москва, ФИЦ ИУ РАН).
О диспетчеризации в одном классе полностью наблюдаемых систем с параллельным обслуживанием.

УДК 519.87+519.24

DOI https://doi.org/10.52513/08698325_2024_31_1_1

Резюме: Для систем с параллельным обслуживанием без потерь, в которых решение о постановке задачи в очередь выполняется со случайной задержкой, известные алгоритмы диспетчеризации могут быть скомбинированы для получения рекордных выигрышей в классических целевых функциях.

Ключевые слова: Системы с параллельным обслуживанием, диспетчеризация, управление, случайная задержка.

Рассматривается одна из модификаций известной проблемы (см. [1, Раздел 5]) распределения ресурсов (маршрутизации, распределению заданий, диспетчеризации) в системах с параллельным обслуживанием. Система состоит из N независимых серверов одинаковой производительности, каждый из которых может выполнять одновременно не более одного задания. Задания поступают в систему по одному в виде рекуррентного потока; интервалы между поступлениями определяются одной и той же функцией распределения (ф.р.) F , которая считается известной. По каждому поступившему заданию диспетчером немедленно принимается решение о том, на какой сервер оно должно быть направлено. На исполнение этого решения всегда затрачивается случайное время с известной ф.р. D , после чего задание начинает выполняться без прерывания, если соответствующий сервер свободен, либо ждет начала обслуживания в его очереди, емкость которой предполагается неограниченной. Размер заданий — случаен с функцией распределения G . Переходы заданий между очередями невозможны. Диспетчер, осуществляя выбор сервера для выполнения очередного задания, имеет полную информацию о состоянии серверов, а также обо всей предыстории принятых им решений.

Классическая для подобных систем проблема — это проблема построения оптимальной процедуры диспетчеризации, и классической целевой функцией является (стационарное) среднее время отклика. В случае вырожденного распределения D для решения этой проблемы к настоящему времени разработано огромное число статических и динамических алгоритмов¹, покрывающих многие варианты распределений F и G (а также различные дисциплины обслуживания в серверах). Большинство из них добивается выигрыша путем сложного или искусного использования доступной точной информации о состояниях системы. Однако с появлением задержки и по мере «увеличения случайности» распределения D , качество динамических алгоритмов падает²: принятые дис-

© Редакция журнала «ОПиПМ», 2024 г.

¹С некоторым обзором можно познакомиться по [2, 3].

²При этом эффективность статических алгоритмов, обычно уступающих динами-

петчером решения оказываются неадекватными тем состояниям системы, с которыми «встречаются» задержанные задания.

В докладе речь пойдет о новых семействах алгоритмов диспетчеризации, которые позволяют нивелировать (при большом числе серверов) описанный выше эффект. В их основе — идея о комбинировании статических и динамических алгоритмов по некоторым (эвристическим) правилам. Вопрос их оптимальности является открытым.

СПИСОК ЛИТЕРАТУРЫ

1. *Litvak N., Yechiali U.* Routing in queues with delayed information. — Queueing Systems, 2003, v. 43, p. 147–165.
2. *Коновалов М. Г., Разумчик Р. В.* Диспетчеризация в системе с параллельным обслуживанием с помощью распределенного градиентного управления марковской цепью. — Информ. и ее примен., 2021, т. 15, № 3, с. 41–50.
3. *Коновалов М. Г., Разумчик Р. В.* Обзор моделей и алгоритмов размещения заданий в системах с параллельным обслуживанием. — Информ. и ее примен., 2015, т. 9, № 4, с. 56–67.

Поступила в редакцию
29.VII.2024

UDC 519.87+519.24

DOI https://doi.org/10.52513/08698325_2024_31_1_1

Konovalev M. G., Razumchik R. V.* (Moscow, FRC CSC RAS). **A dispatching problem in one class of fully observable parallel queues.*

Abstract: For parallel queues with feedback, in which the dispatching decisions are executed after i.i.d. random intervals, conventional dispatching policies can be combined to significantly reduce the stationary mean response time.

Keywords: Parallel service, dispatching, control, random delay.