УДК 004.032.24

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ

Мещеряков Роман Валерьевич,
д-р техн. наук, профессор кафедры комплексной информационной безопасности электронно-вычислительных систем ФГБОУ ВПО «Томский государственный университет систем управления и радиоэлектроники», Россия, 634050, г. Томск, пр. Ленина, 40. E-mail: mrv@security.tomsk.ru

Моисеев Александр Николаевич,
канд. техн. наук, доцент каф. программной инженерии Факультета информатики ФГБОУ ВПО «Национальный исследовательский Томский государственный университет», Россия, 634050, г. Томск, пр. Ленина, 36. E-mail: moiseev.tsu@gmail.com

Демин Антон Юрьевич,
канд. техн. наук, доцент каф. информатики и проектирования систем Института кибернетики ФГБОУ ВПО «Национальный исследовательский Томский политехнический университет», Россия, 634050, г. Томск, пр. Ленина, 30. E-mail: ad@tpu.ru

Дорофеев Вадим Анатольевич,
ст. преподаватель каф. информатики и проектирования систем Института кибернетики ФГБОУ ВПО «Национальный исследовательский Томский политехнический университет», Россия, 634050, г. Томск, пр. Ленина, 30. E-mail: dva@tpu.ru

Матвеев Сергей Александрович,
ведущ. сотр. ООО «ИНКОМ», Россия, 634009, г. Томск, ул. Р. Люксембург, 14а. E-mail: incom@cc.tpu.edu.ru

Модели сетей массового обслуживания являются одним из популярных инструментов математического моделирования различных реальных систем – телекоммуникационных сетей, систем распределенной обработки данных, транспортных сетей, сетевых моделей финансовых потоков и т. д. К сожалению, аналитические результаты исследования таких моделей могут быть получены лишь в некоторых, достаточно частных случаях, поэтому задачи анализа сетей массового обслуживания сложных конфигураций обычно решаются с помощью механизмов имитационного моделирования. Однако, в отличие от простых систем массового обслуживания, сети предполагают множество блоков обслуживания и их взаимодействие между собой. Таким образом, при моделировании сетей массового обслуживания увеличивается размерность задач, исполняемых на одном вычислительном узле, и настольные компьютеры уже не справляются с необходимым объемом моделирования за адекватное время. Отсюда возникает актуальная задача применения механизмов параллельных вычислений и выполнения имитационного моделирования с использованием суперкомпьютерных кластеров.

***Цель исследования**: разработка и программная реализация объектной модели системы имитационного моделирования сетей массового обслуживания, а также реализация в рамках данной программы возможности параллельных вычислений и статистической обработки с целью выполнения моделирования сетей массового обслуживания на суперкомпьютерных кластерах.*

***Методы исследования:** имитационное моделирование на основе дискретно-событийного подхода, математические модели потоков событий: пуассоновский поток, рекуррентный, MAP, полумарковский поток; статистическая обработка данных; методы объектно-ориентированного анализа, проектирования и программирования, технология MPI.*

***Результаты.** Представлена объектная модель системы имитационного моделирования сетей массового обслуживания. Разработанное на ее основе приложение позволяет моделировать сети достаточно произвольной конфигурации. Выполнена реализация параллельных вычислений и последующей статистической обработки данных. Проведены вычислительные эксперименты исполнения приложения на суперкомпьютерном кластере ТПУ для различных размерностей задачи, которые показали высокую эффективность применения параллельных вычислений для задач моделирования сетей массового обслуживания.*

Ключевые слова:
Имитационное моделирование, сети массового обслуживания, объектно-ориентированный подход, параллельные вычисления, технология MPI.

Введение

Сети массового обслуживания [1] являются математическими моделями, применяемыми для решения различных задач: анализа и синтеза сетей передачи и обработки информации [2, 3], моделирования автомобильных потоков [4], решения экономических задач [5] и т. д. Таким образом, исследование различных математических моделей сетей массового обслуживания является актуальной задачей. Для таких исследований применяются различные математические методы [1, 6]. Однако, к сожалению, получить аналитические результаты удается лишь для некоторых частных случаев или определенных классов моделей. Поэтому значительный интерес вызывает применение методов имитационного моделирования [7] для решения задач анализа сетей массового обслуживания различной конфигурации [8].

В настоящей работе представлено описание математической модели сети массового обслуживания и имитационный подход к моделированию процесса функционирования сети общего вида, а также объектно-ориентированная реализация этого подхода.

Поскольку конечным результатом, представляющим интерес для исследователя, являются вероятностные характеристики функционирования сети, требуется, чтобы имитационное моделирование могло предоставить достаточно достоверные эмпирические оценки этих характеристик. Очевидно, что для этого потребуется получить достаточно большие выборки относительно смены состояний сети во время ее функционирования, которые должны быть тем больше, чем больше размерность задачи – количество узлов сети. Таким образом, потребуется либо достаточно длительное моделирование одной реализации процесса, либо, что более предпочтительно, моделирование множества реализаций, пусть на более коротких временных интервалах. Отсюда возникает задача распараллеливания процесса моделирования различных реализаций для одной и той же сети, его выполнение на отдельных вычислительных устройствах и дальнейшее объединение полученных результатов с целью вычисления эмпирических оценок вероятностных характеристик функционирования. Для достижения этой цели предлагается применить технологию MPI для распараллеливания процесса моделирования, выполняемого с использованием разработанного программного обеспечения [9].

Математическая модель сети массового обслуживания

Рассмотрим разомкнутую (открытую) [1] сеть массового обслуживания общего вида. Сеть имеет K узлов, каждый из которых представляет блок обслуживающих приборов. На вход сети поступает поток заявок, который описывается некоторой математической моделью потоков событий, например, пуассоновский поток [10], рекуррентный, MAP- или SM-поток [11–13]. Заявки входящего потока делятся между узлами сети согласно вероятностному распределению v_k, $k=\overline{1,K}$ (рис. 1), где

$$\sum_{k=1}^{K} v_k = 1.$$

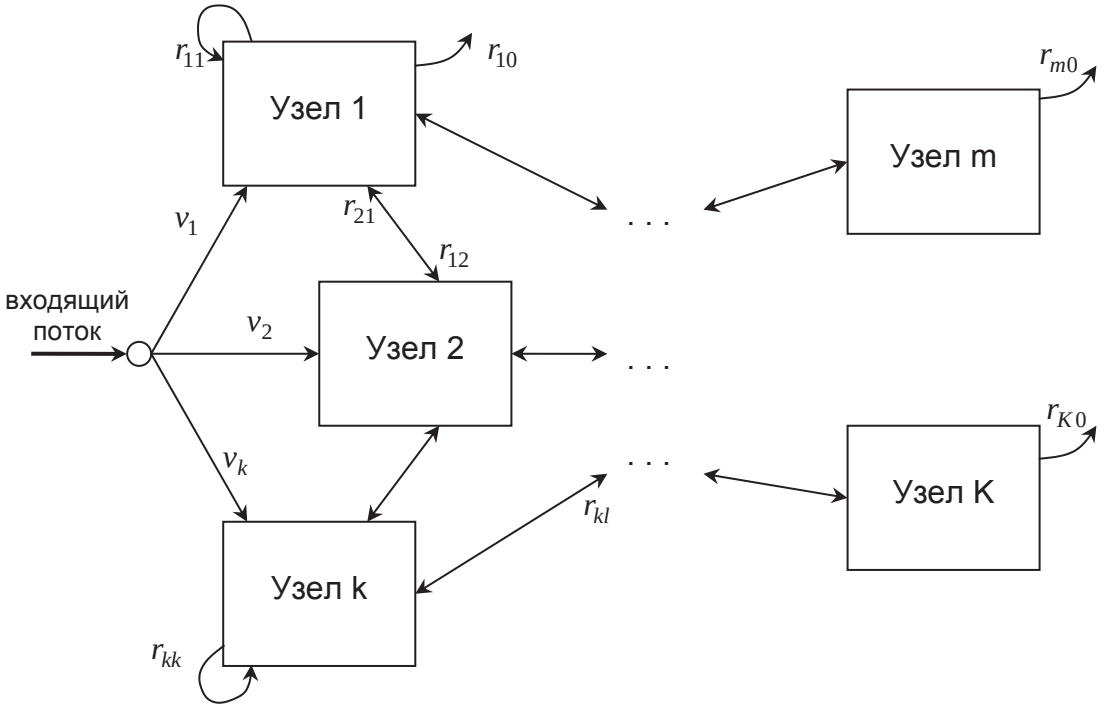

Рис. 1. *Общая схема сети массового обслуживания*

Fig. 1. *General pattern of a queuing system network*

Каждый узел сети может содержать любое конечное число обслуживающих приборов, либо число приборов может быть неограниченным. Дисциплина обслуживания задается в виде функции распределения длительности обслуживания. Обычно этот закон распределения одинаков для всех приборов одного узла. Если число приборов конечно, узел сети может иметь буфер для накопления заявок, ожидающих обслуживания. Буфер может представлять собой классическую очередь с некоторой дисциплиной упорядочения заявок, либо являться так называемым источником повторных вызовов, из которого заявки самостоятельно требуют обслуживания после определенной задержки, длина которой распределена согласно некоторому вероятностному закону (так называемые RQ-системы [14]). Буфер также может быть ограниченным по объему, в этом случае заявки, поступившие в узел в момент, когда все приборы узла заняты, а буфер полностью заполнен, считаются необработанными и удаляются из системы. Кроме указанных могут задаваться и различные другие параметры работы блоков обслуживания – конфликты заявок, начальные распределения и т. д.

По окончании обслуживания в k-м узле заявка с вероятностью r_{kl} переходит для дальнейшего обслуживания в узел l, ($l=\overline{1,K}$), причем с вероятностью r_{kk} она переходит на тот же узел (для повторного обслуживания). С вероятностью $r_{k0} = 1 - \sum_{l=1}^{K} r_{kl}$ заявка считается успешно обработанной и покидает систему.

Относительно описанной модели ставятся задачи получения различных характеристик функционирования сети массового обслуживания – среднего числа занятых приборов, вероятности потери, среднее время пребывания заявки в сети и т. д. Но наиболее важным, конечно, является многомерный закон распределения состояния сети – числа заявок, находящихся в каждом узле. К сожалению, аналитически указанные задачи решаются лишь в некоторых частных случаях или для некоторых классов сетей, например для так называемых экспоненциальных сетей [15] или для сетей с неограниченным числом приборов в узлах [6]. В остальных же случаях существенную помощь в исследовании математических моделей может оказать метод имитационного моделирования.

Имитационное моделирование. Дискретно-событийный поход

Суть метода имитационного моделирования [7] заключается в том, что на компьютере воссоздается достаточно точная копия (модель) исследуемой системы и выполняется моделирование ее функционирования. При этом в отличие от аналитических исследований имитационная модель может учитывать все известные свойства и особенности моделируемого объекта. Основным ограничением для имитационного моделирования являются лишь ресурсы вычислительной техники (оперативная память, производительность процессора) и время проведения вычислений.

В настоящее время одним из наиболее популярных подходов к имитационному моделированию стохастических систем является дискретно-событийный подход [7]. Моделирование, выполняемое с помощью данного подхода, является, с одной стороны, математически корректным и обоснованным, а с другой – достаточно эффективным для реализации на ЭВМ.

Отличительной чертой объектов имитационной модели систем массового обслуживания является то, что они мгновенно изменяют свое состояние в определенные (чаще всего – случайные) моменты времени. Такие моменты времени и действия, которые должны быть выполнены в это время, будем называть событием. При правильном учете всех возможных событий система может изменять свое состояние лишь в эти дискретные моменты времени, и, таким образом, нет необходимости производить моделирование системы на непрерывных интервалах времени между событиями. Каждый следующий момент наступления события и действия, которые необходимо выполнить, полностью определяются действиями, совершенными во время предыдущих событий.

Можно выделить следующие основные понятия и механизмы дискретно-событийного подхода применительно к моделированию систем массового обслуживания [9, 16], дополненные объектами для моделирования сетей:

1) заявка (сообщение, пакет данных) – некий объект, который поступает в систему и передается между ее элементами, пока не покинет систему;
2) источник входящих заявок – гипотетический объект, порождающий входящий поток заявок;
3) блок обслуживающих приборов (узел) – некоторое количество (от 1 до ∞) собранных вместе устройств, занимающихся обработкой (обслуживанием) заявок;
4) буфер – встроенный в блок обслуживающих приборов накопитель заявок, которые в настоящий момент времени не могут быть обслужены по каким-либо причинам;
5) маршрутизатор – гипотетический объект, управляющий разделением входящего потока по узлам сети и потоками заявок внутри нее.

В связи с тем, что в моделях массового обслуживания используются накопители различных типов, будем классифицировать их следующим образом [16]:

- пассивные – это такие накопители, заявки из которых могут быть извлечены только самим блоком обслуживающих приборов в момент изменения его состояния – окончания обслуживания очередной заявки (например, буфер в виде очереди);
- активные – это накопители, которые самостоятельно отправляют находящиеся в них заявки

на обслуживание независимо от текущего состояния блока приборов (например, источник повторных вызовов).

Основными типами событий для сетей массового обслуживания являются:

1) поступление заявки в систему;
2) завершение обработки заявки на устройстве (в узле) и передача ее на другое устройство (узел);
3) при наличии источников повторных вызовов (ИПВ) – обращение заявки из источника повторных вызовов;
4) завершение моделирования.

Процесс поступления заявки предполагает создание заявки и ее направление на один из узлов сети. Затем происходит проверка занятости прибора (в случае ограниченного числа приборов в узле). Если прибор свободен, то заявка встает на обслуживание, в противном случае – помещается в буфер.

В процессе моделирования системы таймер модельного времени постоянно корректируется в соответствии с теми основными событиями, которые возникают в моделируемой системе. После обработки очередного события значение таймера модельного времени $T_{мод}$ сдвигается к моменту следующего события:

$$T_{мод} = \min(T_{пост}, T_{ИПВ}, T_{заверш.обслуж.}, T_{заверш.модел.}),$$

где $T_{пост}$ – момент времени поступления заявки в систему; $T_{ИПВ}$ – момент времени обращения заявки из ИПВ; $T_{заверш.обслуж.}$ – момент времени завершения обслуживания заявки на обслуживающем устройстве; $T_{заверш.модел.}$ – момент времени завершения моделирования. Кроме того, после каждого события соответствующий ему таймер, а также все зависимые таймеры обновляются в соответствии с текущим состоянием системы. Как только достигнут момент завершения моделирования, процесс моделирования останавливается.

Таким образом, дискретно-событийное имитационное моделирование сети массового обслуживания выполняется путем генерации событий на временной оси и последовательным сдвигом таймера модельного времени по событиям на этой оси. В зависимости от того, к какому событию произошел переход, выполняются соответствующие действия:

1. Поступление заявки в систему – маршрутизатор определяет узел, на который поступает заявка. Если блок обслуживающих приборов может ее обработать, то он генерирует на временной оси в будущем событие завершения обслуживания, иначе – помещает заявку в буфер. Если буфер является активным, то он генерирует на временной оси в будущем событие обращения заявки из ИПВ.
2. Завершение обслуживания – маршрутизатор – определяет, должна ли заявка покинуть систему, если нет – то на какой узел она должна перейти для дальнейшего обслуживания. В последнем случае для узла-приемника выполняются действия аналогичные п. 1. Если обслуживший узел имеет пассивный буфер, то он извлекает из него очередную заявку и ставит ее на обслуживание.
3. Обращение заявки из ИПВ – для узла, к которому относится ИПВ, выполняются действия аналогичные п. 1.
4. Завершение моделирования – имитационная модель прекращает все вычисления.

Программа имитационного моделирования ODIS

На основе вышеизложенных принципов с использованием объектно-ориентированного подхода разработан программный комплекс ODIS (Object Distributed Simulation) [9], предназначенный для имитационного моделирования сетей массового обслуживания. Ниже приводится краткое описание архитектуры системы с учетом элементов, необходимых для моделирования сети массового обслуживания.

Основной алгоритм имитационного моделирования выполняется специальным объектом Модель (SimulationModel), который имеет следующий интерфейс (рис. 2).

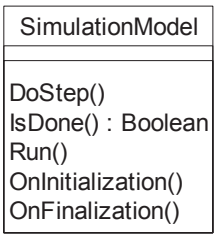

Рис. 2. *Интерфейс базового управляющего класса имитационного моделирования*

Fig. 2. *Interface of basic control class of simulation modeling*

Данный класс реализует Шаблонный Метод [17] Run (). Алгоритм этого метода представлен на рис. 3. Он начинается с выполнения операции OnInitialization (), предназначенной для инициализирующих действий. Затем в цикле производится выполнения метода DoStep (), который отвечает за один шаг моделирования. Выполнение шагов продолжается до тех пор, пока функция IsDone () не вернет значение true, что будет означать, что процесс моделирования окончен. По завершении моделирования вызывается операция OnFinalization (), предназначенная для выполнения завершающих действий.

Класс SimulationModel может служить базовым для любых систем имитационного моделирования, которые используют пошаговый процесс моделирования. В частности, в системе моделирования сетей массового обслуживания реализован наследник этого класса NetworkQueueSimulationModel, который переопределяет операции:

- OnInitialization () – для начального заполнения журнала событий (см. ниже) событиями входящих заявок;
- IsDone () – для индикации конкретного условия останова (по времени моделирования либо по числу событий входящего потока);

- DoStep () – для непосредственной реализации шага моделирования (перехода между событиями).

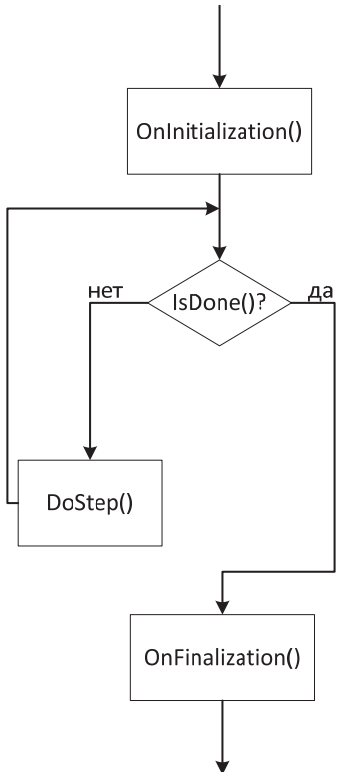

Рис. 3. *Общий алгоритм имитационного моделирования – метод Run () класса SimulationModel*

Fig. 3. *General algorithm of simulation model is the method Run () of the class SimulationModel*

Для реализации дискретно-событийного механизма моделирования введен специальный объект Событие (Event), инкапсулирующий всю информацию, необходимую для корректной регистрации и обработки потока событий внутри модели. Все события записываются в специальный список – журнал событий, который сортирован по времени и обеспечивает дискретно-событийное управление модельным временем. Его связи с другими объектами программы показаны на рис. 4.

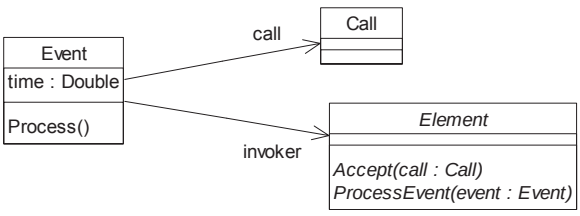

Рис. 4. *Модель событий*

Fig. 4. *Event model*

Абстракция Call (Заявка) введена в систему как сущность переноса данных, связанных с конкретным входящим событием, а также для протоколирования информации его обработки. Объект Element – это любой элемент системы (источник заявок, блок обслуживающих приборов, буфер, маршрутизатор), способный принимать заявки (операция Accept) и/или генерировать и обрабатывать связанные с заявками события (операция ProcessEvent (...)). Объект-событие сохраняет в ссылке invoker указатель на элемент, который создал это событие или должен обработать его. Для элементов модели событиями будут являться такие моменты времени в будущем, когда элемент должен выполнить определенное действие. Например, для источника заявок это будет поступление заявки на обслуживание, для блока обслуживающих приборов – окончание обслуживания, для источника повторных вызовов – попытка заявки снова обратиться за обслуживанием.

При такой организации системы весь процесс, происходящий на одном шаге моделирования (операция DoStep () класса NetworkQueueSimulationModel), описывается следующим простым алгоритмом (рис. 5). Модель извлекает из журнала ближайшее событие, вызывает его операцию Process (), которая просто переадресует вызов обрабатывающему элементу. Конечный элемент (invoker) выполняет необходимые действия.

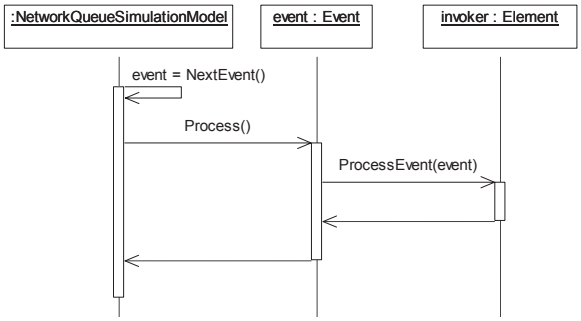

Рис. 5. *Реализация метода DoStep () для управляющего класса имитационного моделирования сети массового обслуживания*

Fig. 5. *Implementation of DoStep () method for control class of simulation modeling of a queuing system network*

Различные элементы модели являются потомками базового класса Element (рис. 6). Каждый из них замещает операции Accept () и ProcessEvent () в соответствии со своими обязанностями. В частности, источник заявок (Source) не может принимать заявки – его метод Accept () генерирует исключение. А вот операция ProcessEvent () (вызывается, когда возникает событие поступления заявки в систему) реализует пересылку заявки на прикрепленный элемент (указатель NextElement), для сети массового обслуживания это маршрутизатор Router. Пересылка заключается в вызове операции Accept (...) этого элемента с соответствующими параметрами.

Операция Accept (...) объекта ServerBlock (блок обслуживающих приборов, узел) проверяет, свободен ли блок (операция IsFree ()), если это так, то заявка поступает на обслуживание – вызывается операция EnforceAccept (...), иначе – она передается буферу (указатель buffer) при его наличии или

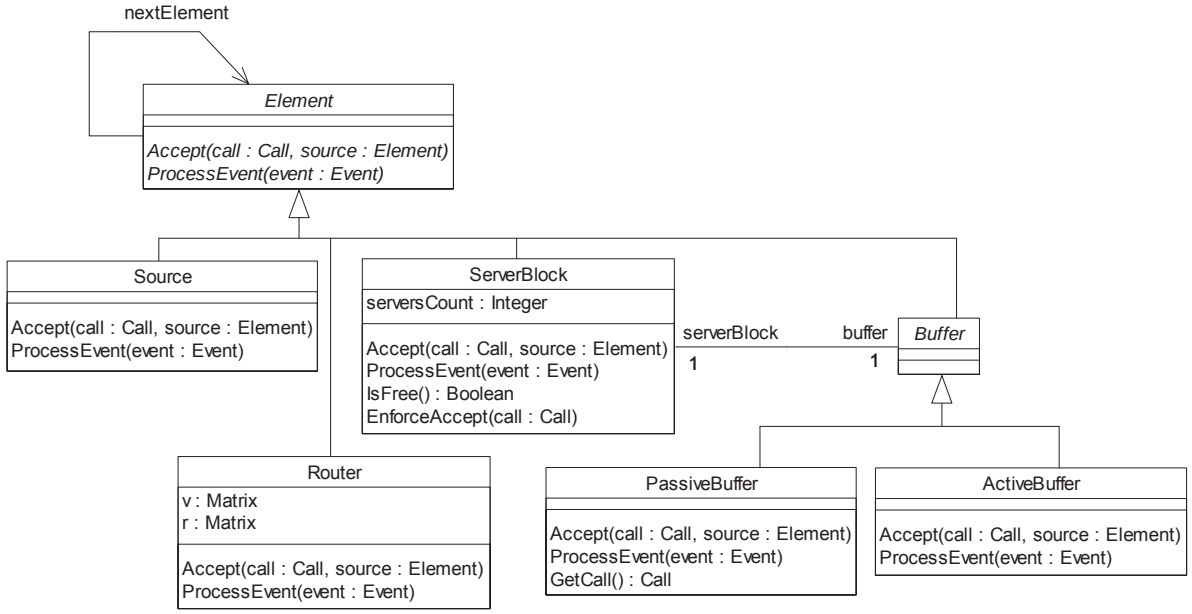

Рис. 6. *Иерархия элементов модели*

Fig. 6. *Model elements hierarchy*

удаляется из системы. Операция ProcessEvent (...) блока (обработка события окончания обслуживания) просто удаляет заявку из списка обслуживаемых данным блоком, и в случае наличия пассивного буфера принудительно забирает из него очередную заявку (если буфер не пуст). Последнее действие не использует механизм журнала событий, т. к. оно производится немедленно после события окончания обслуживания.

Объект PassiveBuffer реализует пассивный накопитель (например, очередь FIFO). Операция Accept (...) объекта PassiveBuffer проверяет, возможно ли поместить заявку в буфер (не достиг ли он предельного объема), и если это возможно – помещает заявку в буфер. Операция ProcessEvent (...) вызывает исключение, так как PassiveBuffer не является активным элементом – таким, который самостоятельно может перемещать заявки в системе. Для извлечения заявки из пассивного буфера блок обслуживающих приборов вызывает его операцию GetCall (), которая извлекает заявку из буфера в соответствии с его дисциплиной (FIFO, LIFO и др.).

В отличие от пассивных активные накопители ActiveBuffer являются активными объектами системы – они самостоятельно пытаются вернуть находящиеся в них заявки на обслуживающие приборы. Операция Accept (...) этих накопителей также вносит заявки во внутренний буфер, однако при этом каждый раз генерируется и вносится в общий журнал будущее событие попытки вернуть заявку на обслуживание. Указатель invoker этого события ссылается на активный буфер, поэтому при его наступлении вызывается ActiveBuffer.ProcessEvent (...), который просто извлекает соответствующую заявку из буфера (указатель на нее имеется в обрабатываемом объекте Event) и отправляет ее на вход обслуживающего блока – так, как будто эта заявка только что вошла в систему. Таким образом обеспечивается единообразная обработка и упрощение программного кода.

Маршрутизатор Router сам не может обрабатывать заявки, поэтому его операция ProcessEvent (...) генерирует ошибку. Операция Accept (...) этого элемента переадресует заявки узлам сети в зависимости от параметра source этой операции (заявка только поступила в сеть или уже была обработана на одном из узлов), а также в соответствии с вектором разделения входящего потока **v** или матрицей маршрутизации **r**.

Накопление и обработка статистической информации

Результатом имитационного моделирования является реализация случайного процесса изменения состояния (количества заявок в узлах) сети массового обслуживания. Однако для исследователя наибольший интерес представляют статистические показатели функционирования. В частности, это – эмпирические оценки математического ожидания и ковариаций числа заявок в узлах сети.

Контур накопления и обработки статистической информации в программном комплексе ODIS реализован с помощью классов иерархии StatisticsAccumulator (рис. 7).

Абстрактный класс StatisticsAccumulator содержит два основных атрибута, используемых для обработки статистики:

- Totaltime – общее время моделирования;
- GenerationVolume – объем выборки.

Также этот класс объявляет интерфейс в виде основных операций, которые обязательно должны

быть реализованы в потомках (все они в этом классе объявлены как абстрактные):

- AddInterval (timeInterval, state) – добавляет состояние системы state (массив целых чисел, соответствующих числу заявок в каждом узле) к общей статистике, при этом timeInterval – интервал времени, в течение которого сохранялось это состояние;
- GetMeans () – вычисляет и возвращает матрицу размера 1×K, содержащую математические ожидания числа занятых приборов в узлах;
- GetCovariance () – вычисляет и возвращает матрицу ковариации числа занятых приборов в узлах размера K×K;
- Save (filename) – сохраняет статистику в файле с именем filename для последующего анализа;
- Load (filename) – создает новый объект StatisticsAccumulator и загружает в него статистические данные из файла filename, сохраненные командой Save (...);
- Merge (accumulator) – добавляет статистические данные из другого объекта StatisticsAccumulator, заданного в качестве параметра accumulator.

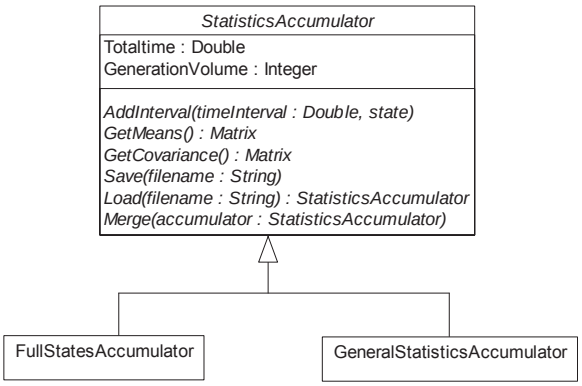

Рис. 7. *Классы накопления и обработки статистической информации (подразумевается, что все операции класса StatisticsAccumulator перекрыты в потомках)*

Fig. 7. *Classes of statistic information accumulation and processing (all operations of StatisticsAccumulator class are supposed to be overrode in descendents)*

Класс FullStatesAccumulator накапливает полную статистику о многомерных состояниях системы. Данные записываются в формате <состояние, время>. Здесь состояние – массив целых чисел, содержащий количество заявок в каждом из узлов; время – продолжительность пребывания системы в этом состоянии. Вычисление средних и ковариаций производится путем усреднения по времени всего массива данных. Основной недостаток – при больших размерностях задачи (количестве узлов сети) и больших значениях состояний требует значительного объема оперативной памяти.

Класс GeneralStatisticsAccumulator при каждом изменении состояния пересчитывает средние и ковариации. Сохраняемые данные содержат матожидания и интегральные по времени корреляции числа занятых приборов в узлах сети, а также маргинальные (одномерные) распределения времени пребывания каждого узла в определенном состоянии. Основной недостаток – немного замедляет процесс моделирования, заранее требуется знать, какого рода статистику необходимо получить.

Параллельные вычисления

Основной особенностью применения описанного приложения для имитационного моделирования сетей массового обслуживания является возможность многократного запуска процесса моделирования для одной и той же сети. Таким образом, выполняя моделирование на различных вычислительных устройствах можно получить и сохранить в файлах статистическую информацию гораздо большего объема, чем при моделировании на одном компьютере за то же время. По окончании моделирования все эти файлы могут быть загружены и объединены с помощью специальной утилиты (см. выше операции Load () и Merge () класса StatisticsAccumulator). Таким образом, за малое время может быть получен статистический материал большего объема.

Параллельные вычисления проводились на суперкомпьютерном кластере Томского политехнического университета. Для запуска приложения должны были быть решены следующие задачи:

1) декомпозиция (распараллеливание) приложения;
2) выбор технологии, обеспечивающей параллельное выполнение и синхронизацию;
3) проведение вычислительных экспериментов и анализ временных данных;
4) выявление «узких мест».

Поскольку характеристики вычислительной системы, количество узлов, связей и другие параметры, а также количество обрабатываемых заявок являются входными данными приложения, то согласно классификации Флинна [18] возможно использовать вычислительные комплексы с архитектурами SIMD и MIMD, позволяющими параллельно обрабатывать несколько потоков данных.

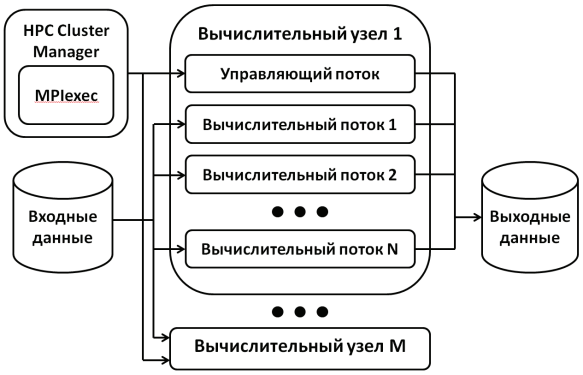

Рис. 8. *Архитектура распараллеленного приложения для моделирования*

Fig. 8. *Architecture of parallelized application for simulation*

При множественных информационных потоках (multiple data) целесообразно проводить де-

композицию по данным: одинаковые по размеру фрагменты данных можно назначить вычислительным потокам до начала параллельной обработки, что соответствует статической декомпозиции и значительно упрощает распараллеливание и синхронизацию (рис. 8).

Суперкомпьютерный кластер ТПУ состоит из двух частей с разделяемой памятью. Характеристики каждой части приведены в таблице.

Таблица. *Основные характеристики суперкомпьютерного кластера ТПУ*

Table. *Principle characteristics of supercomputing cluster at Tomsk Polytechnic University*

Характеристика Characteristic	СКИФ1	СКИФ2
Количество вычислительных узлов Quantity of computation nodes	24	39
Количество процессоров Quantity of processor units	48	78
Количество вычислительных ядер Quantity of compute kernel	96	320
Тактовая частота, ГГц Clock speed, GHz	2,66	2,9
Общий объем оперативной памяти, Гб Total core memory, Gb	192	479

В качестве технологии распараллеливания была выбрана технология MPI. Поскольку приложение соответствует архитектуре SIMD, то возможно было использование графических ускорителей (технологии CUDA, AMD APP, OpenCL и т. д.), однако это потребовало бы существенной переработки исходного кода приложения, поскольку приложение было разработано на языке C#, а для графических ускорителей распараллеливаемая часть алгоритма должна быть написана на языке C. Поддержка языка C# изначально заложена в библиотеку MPI.NET, что существенно облегчило задачу выбора технологии [19].

Технология MPI подразумевает наличие одного управляющего и нескольких рабочих процессов. Управляющий процесс имеет номер 0 и выдаёт задания рабочим процессам и обрабатывает результаты их работы. Входные данные для каждой задачи моделирования записаны в файлах N.odis, где N – это размерность задачи. Каждый такой файл содержит параметры моделируемой системы, такие как число заявок, а размерность задачи показывает число узлов в моделируемой системе. В результате проведения экспериментов были получены данные о производительности моделирования на разном количестве процессорных ядер (рис. 9).

На выходе каждого процесса организовывался файл размером от нескольких килобайт до сотен мегабайт, содержащий результаты обработки заявок. Поскольку для хранения выходной информации используется общее высокоскоростное хранилище, возникло опасение, что одновременная запись таких больших объемов данных может негативно сказаться при замере производительности. Вычислительные эксперименты показали, что время выполнения моделирования с записью на диск и без записи на диск отличается, но при небольшой размерности задачи (до 50) это почти не влияет на общее время работы. При большей размерности задачи вклад этапа записи результатов на диск ока-

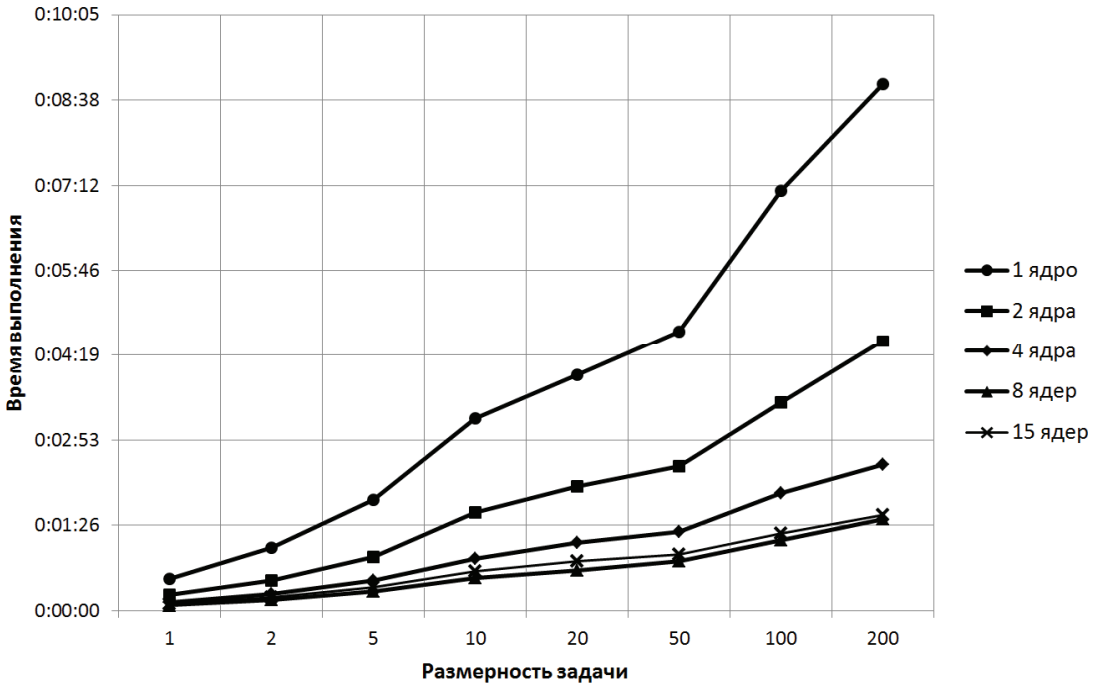

Рис. 9. *Зависимость времени выполнения приложения от размерности задачи*

Fig. 9. *Dependence of application execution time on problem dimension*

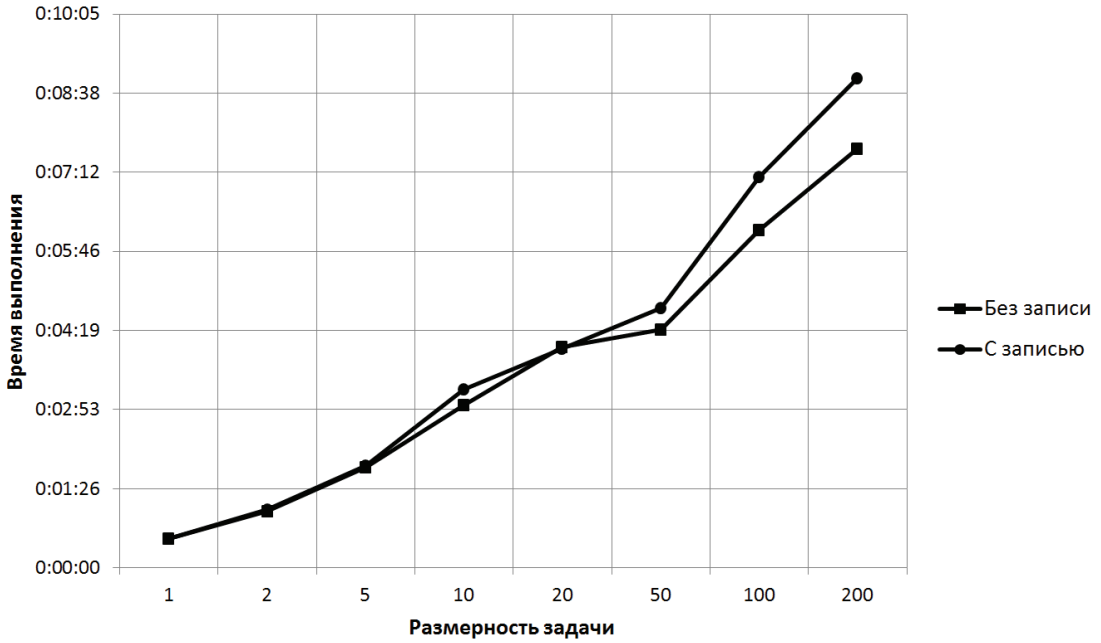

Рис. 10. *Влияние процесса записи результирующего файла на время выполнения приложения*

Fig. 10. *Influence of target file recording on application execution time*

зывается уже более существенным, что снижает эффективности применения большого количества параллельных потоков (рис. 10). Решением здесь может быть использование нескольких раздельных хранилищ информации либо разнесение операций записи различных потоков во времени.

В ходе моделирования было отмечено экспоненциальное увеличение требований к объему оперативной памяти при увеличении размерности задачи. Так, при моделировании системы из 500 узлов на 8 и 16 ядрах приложению не хватало 32 Гб памяти. Для решения этой проблемы нужно уменьшать либо количество параллельных процессов, или же размерность задачи.

Выводы

В работе представлена объектная модель системы имитационного моделирования сетей массового обслуживания. Разработанное на ее основе приложение позволяет моделировать сети достаточно произвольной конфигурации.

Основной проблемой использования приложения для моделирования сетей массового обслуживания с большим количеством узлов является необходимость моделирования большого числа сущностей, а также получения статистических выборок больших объемов. Это неминуемо приводит к увеличению затрат времени процессора и объемов используемой оперативной памяти. С целью оптимизации указанных параметров в ходе практической реализации задачи приложение было адаптировано для запуска на суперкомпьютерном кластере. В результате экспериментальных запусков была установлена принципиальная возможность использования суперкомпьютера для моделирования системы, а также получены сведения о влиянии вспомогательных операций, таких как запись массивов полученной информации на диск, на производительность приложения.

Разработанное приложение может быть использовано для научных исследований в области прикладного вероятностного анализа, а также для расчета характеристик функционирования реальных технических систем, модели которых представлены в виде сетей и систем массового обслуживания [20].

СПИСОК ЛИТЕРАТУРЫ

1. Ивницкий В.А. Теория сетей массового обслуживания. – М.: Изд-во «Физматлит», 2004. – 772 с.
2. Многофазная модель массового обслуживания системы распределенной обработки данных / В.В. Грачев, А.Н. Моисеев, А.А. Назаров, В.З. Ямпольский // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2012. – № 2 (26). Ч. 2. – С. 248–251.
3. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 512 с.
4. Федоткин М.А. Модели в теории вероятностей. – М.: Изд-во «Физматлит», 2012. – 614 с.
5. Маталыцкий М.А., Статкевич С.Э. НМ-сети как новые стохастические модели прогнозирования доходов различных объектов // Вестник ГрГУ. Серия 5. Экономика. – 2009. – № 1. – С. 107–115.
6. Назаров А.А., Моисеев А.Н. Исследование открытой немарковской сети массового обслуживания $GI–(GI|\infty)^K$ с высокоинтенсивным рекуррентным входящим потоком // Проблемы передачи информации. – 2013. – Т. 49. – Вып. 2. – С. 78–91.

7. Лоу А., Кельтон В. Имитационное моделирование. 3-е изд. – СПб.: Питер, 2004. – 848 с.
8. Задорожный В.Н. Оптимизация однородных немарковских сетей массового обслуживания // Проблемы управления. – 2009. – № 6. – С. 68–75.
9. Моисеев А.Н., Синяков М.В. Разработка объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2010. – № 1. – С. 89–93.
10. Бочаров П.П., Печинкин А.В. Теория массового обслуживания. – М.: Изд-во РУДН, 1995. – 529 с.
11. Moiseev A., Nazarov A. Investigation of High Intensive General Flow // Problems of Cybernetics and Informatics: Proc. of the IV International Conference (PCI'2012). – Baku, Azerbaijan, September 12–14, 2012. – Baku: IEEE, 2012. – P. 161–163.
12. Моисеев А.Н., Назаров А.А. Исследование высокоинтенсивного MAP-потока // Известия Томского политехнического университета. – 2013. – Т. 322. – № 2. – С. 16–18.
13. Моисеев А.Н., Назаров А.А. Асимптотический анализ высокоинтенсивного полумарковского потока событий // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2013. – № 3 (29). – С. 109–115.
14. Artalejo J.R., Gomez-Corall A. Retrial Queueing Systems. A Computational Approach. – Berlin: Sprnger. 2008. – 318 p.
15. Jackson J.R. Networks of waiting lines // Operat. Res. – 1957. – V. 5. – № 4. – P. 131–142.
16. Моисеев А., Моисеева С., Синяков М. Базовая объектная модель слоя предметной области системы имитационного моделирования процессов массового обслуживания // Application of Information and Communication Technology in Economy and Education (ICAICTEE-2011): Proc. of the Int. Conf. – Sofia, December 2–3, 2011. – Sofia: University of National and World Economy, 2011. – P. 230–236.
17. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влессидес. – СПб.: Питер, 2010. – 368 с.
18. Михайлов Б.М., Халабия Р.Ф. Классификация и организация вычислительных систем. – М.: МГУПИ, 2010. – 144 с.
19. Дёмин А.Ю., Дорофеев В.А. Распараллеливание алгоритма выделения границ объектов на основе структурно-графического представления // Известия Томского политехнического университета. – 2013. – Т. 323. – № 5. – С. 159–164.
20. Мещеряков Р.В. Информационные иерархические системы // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 151–154.

Поступила 20.05.2014 г.

UDC 004.032.24

USING PARALLEL COMPUTING IN QUEUEING NETWORK SIMULATION

Roman V. Meshcheryakov,
Dr. Sc., Tomsk State University of Control System and Radioelectronics,
40, Lenin Avenue, 634050, Tomsk, Russia. E-mail: mrv@security.tomsk.ru

Alexander N. Moiseev,
Cand. Sc., Tomsk State University, 36, Lenin Avenue,
Tomsk, 634050, Russia. E-mail: moiseev.tsu@gmail.com

Anton Yu. Demin,
Cand. Sc., Tomsk Polytechnic University, 30, Lenin Avenue,
Tomsk, 634050, Russia. E-mail: ad@tpu.ru

Vadim A. Dorofeev,
Tomsk Polytechnic University, 30, Lenin Avenue,
Tomsk, 634050, Russia. E-mail: dva@tpu.ru

Sergey A. Matveev,
LLC «INCOM», 14a, Rosa Luxemburg street,
Tomsk, 634009, Russia. E-mail: incom@cc.tpu.edu.ru

Queueing networks models are one of the most popular tools of mathematical modeling of various physical systems: telecommunication networks, distributed data processing systems, transportation networks, network models of cash flows, etc. Unfortunately, analytical results of the study of such models can be obtained only in some rather special cases. Therefore, the objectives of the analysis of queueing networks with complex configurations are usually resolved through mechanism of simulation. However, the main difference of the queueing networks from simple queueing models is that each network can contain many service nodes and these nodes interact with each other. Thus, the simulation of the queueing networks increases the dimension of the tasks executed on one computing device. So, desktop computers cannot perform the required simulation in adequate time. Hence, we have the urgent task of applying the mechanisms of parallel computing and performing simulations using supercomputer clusters.

The main aim of the study *is to develop and to implement the object model of the simulation system of the queueing networks and to implement as well the capabilities of parallel computing and statistical processing in order to perform simulation of queueing networks on supercomputer clusters.*

The methods used in the study: *simulation based on the discrete/event approach; mathematical models of the event flows, such as Poisson, renewal, Markovian Arrival Process, and semi-Markov processes; statistics data processing; object-oriented methods of analysis, software design and programming, MPI technology.*

The results. *The paper introduces the object model of the software for simulating queueing networks. The application developed on its basis allows simulating queueing networks with rather arbitrary configuration. The parallel computing was implemented and the data were processed. The authors have carried out the real numerical experiments of application execution on the supercomputer cluster of TPU for different dimensions of the task which demonstrated high efficiency of applying parallel computing for simulation of the queueing networks.*

Key words:
Simulation modelling, queueing networks, object-oriented approach, parallel computing, MPI technology.

REFERENCES

1. Ivnicky V.A. *Teoriya setey massovogo obsluzhivaniya* [Queueing networks theory]. Moscow, Fizmatlit Publ., 2004. 772 p.
2. Grachev V.V., Moiseev A.N., Nazarov A.A., Yampolsky V.Z. Mnogofaznaya model massovogo obsluzhivaniya sistemy raspredelennoy obrabotki dannykh [Multiphase queueing networks model for distributed data processing]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki*, 2012, no. 2 (26), P. 2, pp. 248–251.
3. Vishnevsky V.M. *Teoreticheskie osnovy proektirovaniya kompyuternykh setey* [Theoretical basics of computer networks]. Moscow, Tekhnosfera Publ., 2003. 512 p.
4. Fedotkin M.A. *Modeli v teorii veroyatnostey* [Models in the probability theories].Moscow, Fizmatlit Publ., 2012. 614 p.
5. Matalytsky M.A., Statkevich S.E. NM-seti kak novye stokhasticheskie modeli prognozirovaniya dokhodov razlichnykh obektov [NM-networks as new stochastic model for different objects incoms forecasts]. *Vestnik GrGU, Seriya 5. Ekonomika*, 2009, no. 1, pp. 107–115.
6. Nazarov A.A., Moiseev A.N. Analysis of an open non-Markovian GI-$(GI|\infty)^K$ queueing network with high-rate renewal arrival process. *Problems of Information Transmission*, 2013, vol. 49, no. 2, pp. 167–178. DOI: 10.1134/S0032946013020063.
7. Lou A., Kelton V. *Imitatsionnoe modelirovanie* [Simulation modelling]. St-Petersburg, Piter Publ., 2004. 3rd ed., 848 p.
8. Zadorozhny V.N. Optimizatsiya odnorodnykh nemarkovskikh setey massovogo obsluzhivaniya [Optimization of homogeneous Markov queuing networks]. *Problemy upravleniya*, 2009, no. 6, pp. 68–75.
9. Moiseev A.N., Sinyakov M.V. Razrabotka obektno-orientirovannoy modeli sistemy imitatsionnogo modelirovaniya protsessov massovogo obsluzhivaniya [Development of an object-oriented system model simulation of queuing processes]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaya texnika i informatika*, 2010, no. 1, pp. 89–93.
10. Bocharov P.P., Pechinkin A.V. *Teoriya massovogo obsluzhivaniya* [Queueing Theory]. Moscow, RUDN Publ., 1995. 529 p.
11. Moiseev A., Nazarov A. Investigation of High Intensive General Flow. *Proceedings of the IV International Conference. Problems of Cybernetics and Informatics (PCI2012)*. Baku, September 12–14, 2012. Baku, Azerbaijan: IEEE Press, 2012. pp. 161–163.
12. Moiseev A.N., Nazarov A.A. Issledovanie vysokointensivnogo MAP-potoka [Study of high-MAP-stream]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 322, no. 2, pp. 16–18.
13. Moiseev A.N., Nazarov A.A. Asimptotichesky analiz vysokointensivnogo polumarkovskogo potoka sobyty [Asymptotic analysis of Semimarkov high-flow events]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki*, 2013, no. 3 (29), pp. 109–115.
14. Artalejo J.R., Gomez-Corall A. *Retrial Queueing Systems. A Computational Approach*. Berlin, Sprnger. 2008. 318 p.
15. Jackson J.R. Networks of waiting lines. *Operat. Res.*, 1957, vol. 5, no. 4, pp. 131–142.
16. Moiseev A., Moiseeva S., Sinyakov M. Bazovaya obektnaya model sloya predmetnoy oblasti sistemy imitatsionnogo modelirovaniya protsessov massovogo obsluzhivaniya [Basic object model layer domain system simulation of queuing processes]. *Application of Information and Communication Technology in Economy And Education (ICAICTEE-2011). Proc. of the Int. Conf.* Sofia, Bulgaria, December 2–3, 2011. Sofia, University of National and World Economy, 2011. pp. 230–236.
17. Gamma E., Xelm R., Dzhonson R., Vlessides Dzh. *Priemy obektno-orientirovannogo proektirovaniya. Patterny proektirovaniya* [Elements of reusable Object-oriented design. Design patterns]. St-Petersburg, Piter publ., 2010. 368 p.
18. Mikhaylov B.M., Khalabiya R.F. *Klassifikatsiya i organizatsiya vychislitelnykh sistem* [Classification and organization of computer systems]. Moscow, MGUPI Press, 2010. 144 p.
19. Dyomin A.Yu., Dorofeev V.A. Rasparallelivanie algoritma vydeleniya granits obektov na osnove strukturno-graficheskogo predstavleniya [Parallelization of the boundaries of objects extraction algorithm based on structural and graphical representation]. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. 159–164.
20. Meshcheryakov R.V. Informatsionnye ierarkhicheskie sistemyi [Information hierarchical systems]. *Bulletin of the Tomsk Polytechnic University*, 2009, vol. 314, no. 5, pp. 151–154.

Received: 20 May 2014.