

УДК 378+004.42

DOI: 10.25559/SITITO.14.201804.859-864

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ «ПЛАНИРОВАНИЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ» И ЕЕ МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Ю.А. Крыжановская
Воронежский государственный университет, г. Воронеж, Россия

WORKING PROGRAM FOR STUDENT COURSE «DISTRIBUTED COMPUTING SCHEDULING» AND METHODOICAL SUPPORT DESCRIPTION

Yuliana A. Kryzhanovskaya
Voronezh State University, Voronezh, Russia

© Крыжановская Ю.А., 2018

Ключевые слова

Учебная дисциплина; рабочая программа; компетенции; распределенные вычисления; планирование распределенных вычислений; алгоритм; иллюстрирующие примеры; многопоточное приложение; программная реализация.

Аннотация

В данной статье обсуждается составленная рабочая программа учебной дисциплины Б1.В.04 «Планирование распределенных вычислений», изучаемой при обучении в магистратуре направления 01.04.02 «Прикладная математика и информатика» по профилю «Математическое моделирование» во втором семестре. Рассмотрение включает описание программы учебной дисциплины и программной реализации некоторых иллюстрирующих курс примеров.

Описание рабочей программы учебной дисциплины содержит тематический план, перечень знаний, умений и навыков, которыми должен обладать обучающийся после завершения изучения дисциплины «Планирование распределенных вычислений», а также критерии оценивания результатов обучения.

Методическое обеспечение рассматриваемой дисциплины включает лабораторные работы и комплект демонстрационных программ, иллюстрирующих изложение курса. Обсуждаемые материалы были апробированы на практике в течение двух лет при проведении занятий по рассматриваемому курсу. В данной статье приводится краткое описание программной реализации некоторых иллюстрирующих примеров (скалярных часов и алгоритма взаимного исключения Лэмпорта [1], а также широковеб-ательного алгоритма Сузуки-Касами [2]), позволяющих улучшить качество усвоения изучаемого материала. Реализация выполнена на платформе .NET Framework в среде программирования Microsoft Visual Studio 2017 на языке программирования C#. Для наглядности демонстрации приложения разработаны как многопоточные.

В рамках учебной дисциплины «Планирование распределенных вычислений» отводится достаточно большое время на самостоятельную работу. Предполагается, что во время проведения аудиторных занятий обучающиеся знакомятся с базовыми понятиями и алгоритмами, работают с демонстрационными программами, выполняют лабораторные работы, а затем самостоятельно расширяют свои знания, руководствуясь планом и тематикой занятий.

Keywords

Student course; working program; competence; distributed computing; distributed computing scheduling; algorithm; illustrative examples; multithreaded application; software implementation.

Abstract

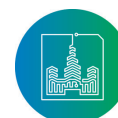
The article describes working program for student course B1.V.04 «Distributed Computing Scheduling» which is taught during second semester of master course 01.04.02 «Applied Mathematics and Informatics» of «Math modeling» specialty. The narration includes description of student course and fragments of study guide to be edited.

The description of working program for student course consists of editorial calendar, list of expertise, knowledge and skills which student should obtain after the course «Distributed Computing Scheduling» and criteria for educational outcome results.

Methodical support includes laboratory works and demonstration programs. The considered materials were time tried during lectures and laboratory works within 2 years. The article briefly describes software implementation of several illustrative examples (scalar clock and Lamport's mutual exclusion algorithm [1], and Suzuki-Kasami [2] broadcasting algorithm). The implementation is made on platform .NET

Об авторе:

Крыжановская Юлиана Александровна, старший преподаватель, кафедра ERP-систем и бизнес процессов, факультет прикладной математики, информатики и механики, Воронежский государственный университет (394018, Россия, г. Воронеж, Университетская площадь, д. 1), ORCID: <http://orcid.org/0000-0002-7420-1900>, kryzhanovskaya_ya@amm.vsu.ru



Framework in programming environment Microsoft Visual Studio 2017 by software language C#. The applications are developed as multithreaded ones for illustration purposes.

Within the context of student course «Distributed Computing Scheduling» quite a lot of time is dedicated to independent work. It is assumed that students learn basic concepts and algorithms during lectures, work with demo programs, do laboratory works and then deepen their knowledge following the lecture plan and theme.

Введение

Распределенные вычисления [3] получили достаточно широкое распространение в самых различных областях человеческой деятельности, так или иначе подразумевающих применение средств вычислительной техники и информационных технологий: в экономике, криптографии, научных вычислениях и многих других. Для успешной работы, в частности, в сфере разработки и сопровождения сложных вычислительных комплексов и систем управления ими, выпускнику учреждения высшего образования необходимо иметь представление о распределенных вычислениях, решаемых с их помощью задачах, используемых алгоритмах осуществления и планирования таких вычислений.

Поэтому одним из курсов, изучаемых в рамках профиля «Математическое моделирование» направления 01.04.02 «Прикладная математика и информатика», является курс «Планирование распределенных вычислений». Данный курс имеет большое значение для специалистов в сфере ИТ, а вопросы, рассматриваемые в рамках данной учебной дисциплины, важны для понимания того, какие приемы, алгоритмы и подходы могут быть применены в практической деятельности.

Характеристики программы учебной дисциплины

Программа учебной дисциплины представлена на ресурсе URL: <http://www.moodle.vsu.ru>. Предусмотрено проведение контрольной работы и зачета по итогам освоения дисциплины.

Целями преподавания дисциплины «Планирование распределенных вычислений» являются формирование у студентов целостной системы знаний о возможностях и основных направлениях использования распределенных вычислительных систем, методах и технологиях организации и планирования распределенных вычислений, средств и методов построения моделей распределенных вычислений при проведении фундаментальных и прикладных научных исследований.

Задачами преподавания дисциплины «Планирование распределенных вычислений» являются: приобретение базового набора знаний из области распределенных вычислений, изучение основных понятий и методов построения вычислительных моделей, изучение методов планирования распределенных вычислений, практическое освоение методов и программных средств организации и применения моделей распределенных вычислительных систем.

В результате освоения дисциплины обучающийся должен:

- знать основные понятия распределенных вычислительных систем;
- усвоить ключевые понятия и принципы организации распределенных вычислений;
- знать основные понятия и методы построения вычислительных моделей;

- уметь применять методы планирования распределенных вычислений;
- владеть практическими навыками построения и применения моделей распределенных вычислений;
- владеть общей методикой разработки распределенных приложений и планирования распределенных вычислений.

Общая трудоемкость дисциплины составляет 2 зачетных единицы (72 часа), из которых 10 академических часов отведено на лекционные занятия, 10 академических часов – на лабораторные занятия, а оставшиеся – на самостоятельную работу обучающихся. Предполагается, что во время проведения аудиторных занятий обучающиеся знакомятся с базовыми понятиями и алгоритмами, а затем самостоятельно расширяют свои знания, руководствуясь планом и тематикой занятий.

При изучении дисциплины «Планирование распределенных вычислений» рассматриваются следующие разделы:

- Введение в распределенные вычисления. Вычислительные задачи [3, 4, 5, 6, 7, 8, 9, 10].
- Распределенные алгоритмы [11].
- Распределенные системы. Модели [12]. Модельно-ориентированное программирование [13].
- Отказоустойчивость распределенных вычислительных систем [14, 15].
- Вычислительные кластеры [16]. Вычислительные сети Grid [7, 17, 18]. Распределенные вычисления в Интернет [19, 20]. Программирование распределенных систем [21].
- Планирование вычислений [4, 22].
- Планирование распределения и загрузки ресурсов [7, 23].

Дисциплина относится к вариативной части учебного плана, изучается во втором семестре и является предшествующей для производственной практики по получению профессиональных умений и опыта проектной и производственно-технологической деятельности.

Основными видами образовательных технологий дисциплины являются лекции, организованные с использованием современных мультимедийных технологий, лабораторные работы, а также самостоятельная работа обучающихся.

По итогам освоения дисциплины предусмотрено проведение зачета по дисциплине.

Пример контрольно-измерительного материала, используемого при проведении зачета:

1. Свойства распределенных алгоритмов.
2. События.
3. Использование локальных контрольных точек.
4. Требования к распределенным системам.
5. Не блокирующие операции отправки/получения.
6. Примеры поиска оптимальной стратегии вычислений.
7. Синхронный обмен сообщениями.



При оценивании используются следующие критерии:

- Зачтено – демонстрируется повышенный или базовый уровень сформированности компетенций; обучающийся владеет материалом, отвечает на вопросы; умеет рассуждать; в случае незнания небольшой части материала способен выстроить собственную логическую цепочку рассуждений и получить ответ; может допускать незначительные неточности.
- Не зачтено – демонстрируется минимальное знание или незнание материала, нежелание рассуждать.

Методическое обеспечение

Предполагается, что методическое обеспечение учебной дисциплины в данном случае будет включать в себя перечень лабораторных работ с рекомендациями по их выполнению и комплект программ, реализующих иллюстрирующие примеры. И рассмотрение примеров, и выполнение лабораторных работ имеют своей целью закрепление усвоения изучаемого материала. В настоящее время ведется подготовка учебного пособия по рассматриваемой дисциплине, содержащего базовые сведения, а и лабораторные работы. Ниже будет приведено описание реализации двух иллюстрирующих примеров. Реализация выполнена на платформе .NET Framework в среде программирования Microsoft Visual Studio 2017 на языке программирования C# [24, 25].

1. Первый из них посвящен алгоритму Сузуки-Касами [2]. Для понимания его работы предполагается рассмотрение программной реализации в виде оконного многопоточного приложения на языке C#. Программа должна моделировать работу распределенной системы, организованной с помощью алгоритма Сузуки-Касами, на сгенерированных потоках. В качестве компьютерной сети (КС) используем буферы потоков. Каждый буфер моделирует работу одного компьютера в сети. Роль REQUEST запросов от компьютера к серверу будут играть множество потоков внутри буфера. Приложение, соответственно, является сервером, обрабатывающим потоки.

Поток содержит в себе информацию о своём приоритете, аналогично запросам в КС. Однако, поскольку все происходит в рамках одного компьютера, маркер не обязан содержать в себе массив приоритетов запросов, потому, что он уже содержится в буфере. Поэтому маркером, который передают потоки, будет служить стандартная булевская переменная.

В рамках программной реализации созданы несколько классов, которые отвечают за хранение нужной информации, генерацию потоков, а также выполнение действий по реализации алгоритма, таких, как регулирование количества потоков, получающих доступ к одним и тем же ресурсам, чтение потока, владеющего в данный момент маркером, если это невозможно, то считывание случайного потока из буфера (что не противоречит алгоритму Сузуки-Касами), реализацию команд контроллера, поступающих от интерфейса приложения.

В левом окне “Генерация потоков” создан первый (компьютер) буфер, который начал отправлять (запросы) потоки данному (серверу) приложению. Поскольку потоков еще немного, быстродействие приложения позволяет обрабатывать их в порядке появления, приоритет не учитывается. Потоки, которые были обработаны и завершены, выведены на правое окно “Распределение потоков”.

На Рис.1. и Рис. 2. приведены примеры работы приложения.

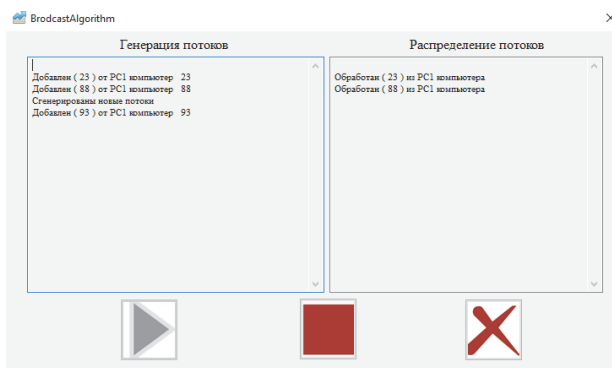


Рис. 1. Начало работы

Fig. 1. Getting started

Однако, при увеличении количества буферов и, соответственно, количества потоков, простого быстродействия недостаточно. В этом месте поможет распределенная система с алгоритмом Сузуки-Касами. Рассмотрим подробнее на Рис.2. ниже.

На Рис. 2. видно, что количество компьютеров в сети (буферов) отправляющих запросы (потоки) увеличилось. Ресурсов сервера (приложения) недостаточно, что успевать обрабатывать все запросы в порядке их поступления.

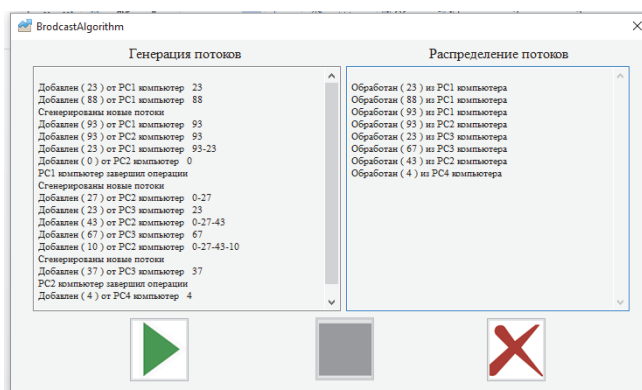


Рис.2. Работа приложения

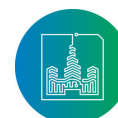
Fig.2. Application operation

После обработки запросов первого буфера (PC1), алгоритм передает маркер самому приоритетному запросу от PC2. После его завершения, программа передала маркер случайному запросу так как, в момент передачи не было непосредственно произведен запрос. Затем приложения обрабатывает два запроса в порядке их приоритета, передавая им маркер. Далее при передаче маркера поступил запрос от PC4, который был сразу обработан. Следующая передача будет снова осуществляться по приоритету.

2. Следующее приложение иллюстрирует работу скалярных часов и алгоритма Лэмпорта [1].

При реализации данного примера были разработаны классы, предназначенные для реализации скалярного времени, создания нового потока, вызов функций, имитирующих КС, имитации компьютера, в котором каждые полторы секунды появляются новые процессы, чтения имеющихся процессов, проверки состояния КС, вывода результатов.

На Рис. 3. приведен пример работы программы. В левом



окне добавляются процессы в список, где P1, P2..., PN – это имена компьютеров, а числа от 1 до 100 это разные процессы внутри одного потока. В правом окне записаны те процессы, которые уже завершены.

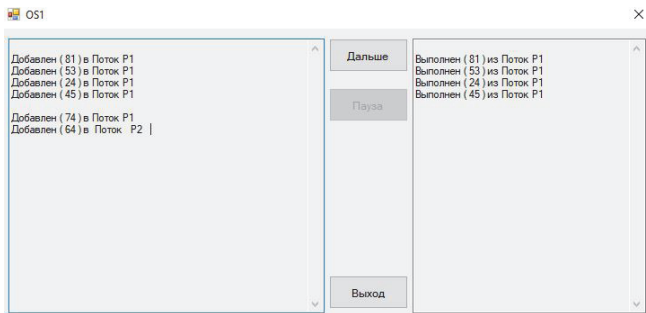


Рис. 3. Пример работы
Fig. 3. An example of work

На Рис. 4. приведен пример той же работы что и на Рис. 3., где все процессы выполнены в порядке их появления в системе (использовались часы Лэмпорта).

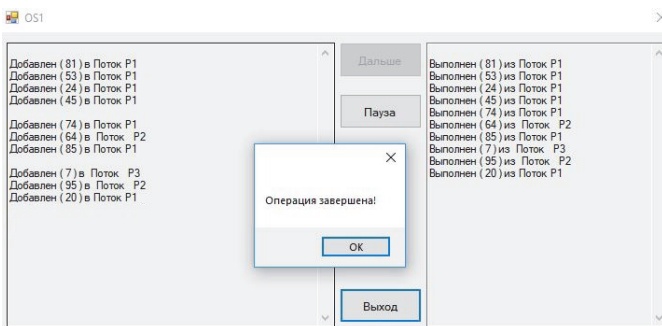


Рис. 4. Пример с использованием часов Лэмпорта
Fig. 4. An example with the Lamport clock

Заключение

В результате проделанной работы была составлена (переработана в соответствии с изменением учебного плана) рабочая программа учебной дисциплины «Планирование распределенных вычислений» и представлены описания программной реализации иллюстрирующих примеров, которые планируется включить в пособие к дисциплине. Пособие будет содержать материалы, необходимые для закрепления усвоения материалов учебной дисциплины, включая лабораторные работы и иллюстрирующие примеры с описанием комплекта программ.

Список использованных источников

- [1] Chandy K.M., Lamport L. Distributed snapshots: determining global states of distributed systems // ACM Transactions on Computer Systems. 1985. Vol. 3, issue 1. Pp. 63-75. DOI: 10.1145/214451.214456
- [2] Suzuki I, Kasami T. A distributed mutual exclusion algorithm // ACM Transactions on Computer Systems. 1985. Vol. 3, issue 4. Pp. 344-349. DOI: 10.1145/6110.214406
- [3] Таненбаум Э., ван Стеен М. Распределенные системы.

- Принципы и парадигмы. СПб: Питер, 2003. 876 с. URL: <https://elibrary.ru/item.asp?id=19580374> (дата обращения: 05.09.2018).
- [4] Топорков В.В. Модели распределенных вычислений. М.: Физматлит, 2004. 320 с.
- [5] Оганов А.Р., Посыпкин М.А., Ройзен В.В., Самцевич А.И., Сухомлин В.А., Храпов Н.П. Система добровольных вычислений для компьютерного дизайна материалов // Современные информационные технологии и ИТ-образование. 2016. Т. 12, № 1. С. 91-96. URL: <https://elibrary.ru/item.asp?id=27539222> (дата обращения: 05.09.2018).
- [6] Посыпкин М.А. и др. Решение задач криптоанализа потоковых шифров в распределенных вычислительных средах // Труды Института системного анализа Российской академии наук. 2009. Т. 46. С. 119-137. URL: <https://elibrary.ru/item.asp?id=15323443> (дата обращения: 05.09.2018).
- [7] Шмакина А.В. Обзор технологий распределенных вычислений // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2014. Т. 3, № 3. С. 51-85. URL: <https://elibrary.ru/item.asp?id=21938877> (дата обращения: 05.09.2018).
- [8] Рассел Дж. Распределённые вычисления / Дж. Рассел, Рон. Кон. М.: VSD, 2013. 103 с.
- [9] Радченко Г.И. Распределённые вычислительные системы. Челябинск: Фотохудожник, 2012. 182 с. URL: <https://elibrary.ru/item.asp?id=19599387> (дата обращения: 05.09.2018).
- [10] Coulouris G. et al. Distributed Systems: Concepts and Design. Fifth edition / G. Coulouris, J. Dollimore, T. Kindberg, G. Blair. Addison – Wesley, 2011. 1047 p.
- [11] Фоккинк У. Распределенные алгоритмы. Интуитивный подход. СПб.: Питер, 2017. 272 с.
- [12] Horalek J., Soběslav V. Analysis and Solution Model of Distributed Computing in Scientific Calculations / N. Nguyen, S. Tojo, L. Nguyen, B. Trawiński (eds) // Intelligent Information and Database Systems. ACIIDS 2017. Lecture Notes in Computer Science. Vol. 10191. Springer, Cham, 2017. Pp. 367-376. DOI: 10.1007/978-3-319-54472-4_35
- [13] Brodsky Yu.I. Model synthesis and model-oriented programming – the technology of design and implementation of simulation models of complex multicomponent systems // In the World of Scientific Discoveries, Series B. 2014. Vol. 2, no. 1. Pp. 12-31. URL: <https://elibrary.ru/item.asp?id=21942274> (дата обращения: 05.09.2018).
- [14] Бондаренко А.А., Якововский М.В. Обеспечение отказоустойчивости высокопроизводительных вычислений с помощью локальных контрольных точек // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2014. Т. 3, № 3. С. 20-36. URL: <https://elibrary.ru/item.asp?id=21938875> (дата обращения: 05.09.2018).
- [15] Грушо А.А., Забежайло М.И., Зацаринный А.А., Николаев А.В., Писковский В.О., Тимонина Е.Е. Классификация ошибочных состояний в распределенных вычислительных системах и источники их возникновения // Системы и средства информатики. 2017. Т. 27, № 2. С. 29-40. DOI: 10.14357/08696527170203
- [16] Bychkov I.V., Manzyuk M.O., Semenov A.A., Zaikin O.S., Afanasiev A.P., Posypkin M.A. Technology for integrating IDLE computing cluster resources into volunteer computing projects //



- Proceedings of the 5th International Workshop on Computer Science and Engineering: Information Processing and Control Engineering. WCSE 2015-IPCE. Moscow, 2015. Pp. 109-114. URL: <https://elibrary.ru/item.asp?id=24528373> (дата обращения: 05.09.2018).
- [17] Magoules F. Fundamentals of Grid Computing: Theory, Algorithms and Technologies. CRC Press, 2010. 322 p.
- [18] Ивашко Е. Распределённые вычисления. Часть 1. Высокопроизводительные вычисления в каждый дом // IBM – United States. 2009. [Электронный ресурс]. URL: <http://www.ibm.com/developerworks/ru/library/l-grid> (дата обращения: 05.09.2018).
- [19] Таненбаум Э., Остин Т. Архитектура компьютера (издание 6). СПб.: Питер, 2017. 816 с.
- [20] Гофф М.К. Сетевые распределённые вычисления: достижения и проблемы. М.: КУДИТС-ОБАР, 2005. 320 с.
- [21] Varela C.A. Programming Distributed Computing Systems: A Foundational Approach. The MIT Press, 2013. 296 p.
- [22] Gritsenko A., Demurchev N., Kopytov V., Shulgin A. Decomposition Analysis and Machine Learning in a Workflow-Forecast Approach to the Task Scheduling Problem for High-Loaded Distributed Systems // Modern Applied Science. 2015. Vol. 9, issue 5. Pp. 38-49. DOI: 10.5539/mas.v9n5p38
- [23] Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D. Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates / W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk (eds) // New Results in Dependability and Computer Systems. Advances in Intelligent Systems and Computing. Vol. 224. Springer, Heidelberg, 2013. Pp. 459-468. DOI: 10.1007/978-3-319-00945-2_42
- [24] Рухтер Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. СПб.: Питер, 2013. 896 с.
- [25] Шарп Дж. Microsoft Visual C#. Подробное руководство. СПб.: Питер, 2017. 848 с.
- stitute for Systems Analysis of the Russian Academy of Science. 2009; 46:119-137. Available at: <https://elibrary.ru/item.asp?id=15323443> (accessed 05.09.2018). (In Russian)
- [7] Shmakina A. Survey on distributed computing technologies. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2014; 3(3):51-85. Available at: <https://elibrary.ru/item.asp?id=21938877> (accessed 05.09.2018). (In Russian)
- [8] Russel J., Kon R. Distributed computations. M.: VSD, 2013. 103 p. (In Russian)
- [9] Radchenko G.I. Distributed computations systems. Chelyabinsk: Photoartist, 2012. 184 p. Available at: <https://elibrary.ru/item.asp?id=19599387> (accessed 05.09.2018). (In Russian)
- [10] Coulouris G., Dollimore J., Kindberg T., Blair G. Distributed Systems: Concepts and Design. Fifth edition. Addison – Wesley, 2011. 1047 p.
- [11] Fokkink W. Distributed Algorithms. Intuitive approach. The MIT Press, 2013. 242 p.
- [12] Horalek J., Soběslav V. Analysis and Solution Model of Distributed Computing in Scientific Calculations. N. Nguyen, S. Tojo, L. Nguyen, B. Trawiński (eds) *Intelligent Information and Database Systems. ACIIDS 2017*. Lecture Notes in Computer Science. Vol. 10191. Springer, Cham, 2017, pp. 367-376. DOI: 10.1007/978-3-319-54472-4_35
- [13] Brodsky Yu.I. Model synthesis and model-oriented programming – the technology of design and implementation of simulation models of complex multicomponent systems. *In the World of Scientific Discoveries, Series B*. 2014; 2(1):12-31. Available at: <https://elibrary.ru/item.asp?id=21942274> (accessed 05.09.2018).
- [14] Bondarenko A.A., Yakobovskiy M.V. Fault Tolerance for HPC by Using Local Checkpoints. *Bulletin of the South Ural State University, Series: Computational Mathematics and Software Engineering*. 2014; 3(3):20-36. Available at: <https://elibrary.ru/item.asp?id=21938875> (accessed 05.09.2018). (In Russian)
- [15] Grusho A.A. Erroneous states classification in distributed computing systems and sources of their occurrence. *Systems and Means of Informatics*. 2017; 27(2):29-40. (In Russian) DOI: 10.14357/08696527170203
- [16] Bychkov I.V., Manzyuk M.O., Semenov A.A., Zaikin O.S., Afanasiev A.P., Posypkin M.A. Technology for integrating IDLE computing cluster resources into volunteer computing projects. *Proceedings of the 5th International Workshop on Computer Science and Engineering: Information Processing and Control Engineering. WCSE 2015-IPCE*. Moscow, 2015, pp. 109-114. Available at: <https://elibrary.ru/item.asp?id=24528373> (accessed 05.09.2018).
- [17] Magoules F. Fundamentals of Grid Computing: Theory, Algorithms and Technologies. CRC Press, 2010. 322 p.
- [18] Ivashko E. Distributed computations. GRID-systems. IBM – United States. 2009. Available at: <http://www.ibm.com/developerworks/ru/library/l-grid> (accessed 05.09.2018). (In Russian)
- [19] Tanenbaum A.S., Austin T. Structured Computer Organization (6th Edition). Pearson, 2012. 808 p.
- [20] Goff M.K. Network Distributed Computing: Fitscapes and Fallacies. Prentice Hall, 2004. 302 p.
- [21] Varela C.A. Programming Distributed Computing Systems: A Foundational Approach. The MIT Press, 2013. 296 p.

Поступила 05.09.2018; принята в печать 10.10.2018;
опубликована онлайн 10.12.2018.

References

- [1] Chandy K.M., Lamport L. Distributed snapshots: determining global states of distributed systems. *ACM Transactions on Computer Systems*. 1985; 3(1):63-75. DOI: 10.1145/214451.214456
- [2] Suzuki I., Kasami T. A distributed mutual exclusion algorithm. *ACM Transactions on Computer Systems*. 1985; 3(4):344-349. DOI: 10.1145/6110.214406
- [3] Tanenbaum A.S., van Steen M. Distributed Systems. Principles and Paradigms. Prentice Hall, 2001. 803 p.
- [4] Toporkov M. Distributed systems models. M.: Fizmatlit, 2004. 320 p. (In Russian)
- [5] Oganov A., Posypkin M., Roisen V., Samtsevich A., Sukhomlin V., Hrapov N. Volunteer Computing for Computational Materials Design. *Modern Information Technologies and IT-Education*. 2016; 12(1):91-96. Available at: <https://elibrary.ru/item.asp?id=27539222> (accessed 05.09.2018). (In Russian)
- [6] Posypkin M. et al. Solving stream ciphers cryptanalysis in distributed computing environments. *Proceeding of the In-*



- [22] Gritsenko A., Demurchev N., Kopytov V., Shulgin A. Decomposition Analysis and Machine Learning in a Workflow-Forecast Approach to the Task Scheduling Problem for High-Loaded Distributed Systems. *Modern Applied Science*. 2015; 9(5):38-49. DOI: 10.5539/mas.v9n5p38
- [23] Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D. Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates. W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk (eds) *New Results in Dependability and Computer Systems*. Advances in Intelligent Systems and Computing. Vol. 224. Springer, Heidelberg, 2013, pp. 459-468. DOI: 10.1007/978-3-319-00945-2_42
- [24] Richter J. CLR via C#. (Developer Reference). 4th Edition. Microsoft Press, 2012. 896 p.
- [25] Sharp J. Microsoft Visual C# Step by Step. (Developer Reference). 8th Edition. Microsoft Press, 2015. 816 p.

Submitted 05.09.2018; revised 10.10.2018;
published online 10.12.2018.

About the author:

Yuliana A. Kryzhanovskaya, Senior Lecturer, ERP-systems and business processes chair, Applied mathematics, informatics and mechanics faculty, Voronezh State University (1 Universitetskaya pl., Voronezh 394018, Russia), ORCID: <http://orcid.org/0000-0002-7420-1900>, kryzhanovskaya_ya@amm.vsu.ru



BY

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted reuse, distribution, and reproduction in any medium provided the original work is properly cited.

