

## Организация вычислительной среды разработки и применения научных рабочих процессов на основе контейнеризации

А. Г. Феоктистов\*, Р. О. Костромин, М. Л. Воскобойников, Д. И. Ли-Дэ  
Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 664033,  
Иркутск, Россия

\*Контактный автор: Феоктистов Александр Геннадьевич, e-mail: agf@icc.ru

Поступила 10 октября 2023 г., принята в печать 17 октября 2023 г.

Статья посвящена разработке инструментальных средств построения гетерогенной вычислительной среды для создания и эксплуатации распределенных пакетов прикладных программ. Реализация научных рабочих процессов (схем решения задач) в рамках пакета основывается на контейнеризации прикладного и системного программного обеспечения. Суть и новизна представленного подхода заключаются в совместном, согласованном использовании стандартизированного языка Business Process Execution Language, веб-сервисов обработки данных и контейнеризации научных рабочих процессов. Такая интеграция позволяет: описать на вышеупомянутом языке схемы решения задач, объединяющие сервисы для доступа к базам данных и приложениям предметных специалистов из разных организаций; обеспечить за счет контейнеризации масштабирование как процесса выполнения вычислительных заданий, так и процесса функционирования систем управления этими заданиями в среде; осуществить взаимодействие научных рабочих процессов с геоинформационными системами с помощью веб-сервисов обработки данных. Результаты исследования успешно апробированы для пакета анализа уязвимости энергетических комплексов и отдельных систем энергетики на территории Байкальской природной территории.

*Ключевые слова:* распределенная вычислительная среда, научные рабочие процессы, контейнеризация, пакет прикладных программ, анализ живучести энергетических систем.

*Цитирование:* Феоктистов А.Г., Костромин Р.О., Воскобойников М.Л., Ли-Дэ Д.И. Организация вычислительной среды разработки и применения научных рабочих процессов на основе контейнеризации. Вычислительные технологии. 2023; 28(6):151–164. DOI:10.25743/ICT.2023.28.6.013.

### Введение

С начала 90-х гг. прошлого столетия научные рабочие процессы (НРП) широко применяются исследователями для подготовки и проведения вычислительных экспериментов [1]. При этом важной проблемой является интеграция и применение используемого программного обеспечения (ПО). Как правило, построение НРП осуществляется на основе модульного подхода к программированию, который поддерживает простоту и гибкость в разработке, модификации и выполнении прикладного программного обеспечения (ППО).

В связи с этим активно развиваются системы управления (СУ) научными рабочими процессами. В их числе можно выделить такие известные традиционные системы, как Uniform Interface to Computing Resources (UNICORE) [2], Directed Acyclic Graph Manager (DAGMan) [3], Pegasus [4], и другие программные комплексы [5]. Наличие широкого спектра существующих СУ НРП обусловлено различием требований разных исследовательских групп к описанию их алгоритмических знаний, спецификой подходов к решению задач в конкретных предметных областях, разнообразием характеристик компьютерных ресурсов и средств управления ими, а также многими другими факторами.

Необходимость обеспечения разнообразия способов выполнения ППО, масштабирования вычислений и эффективности использования ресурсов требует наличия развитого системного программного обеспечения (СПО) для реализации, доставки, размещения и запуска НРП. Сегодня актуальным подходом к решению перечисленных выше задач является применение контейнеризации программ [6]. С точки зрения пользователей системы управления НРП, инструменты контейнеризации в достаточной степени низкоуровневые. Поэтому предоставляемые ими возможности должны включаться в функционал системы управления НРП.

Современные СУ НРП активно развиваются в направлении построения сервис-ориентированных научных процессов [7–9]. Кроме того, в связи с высокой актуальностью вопросов, решаемых в рамках экологического мониторинга, в их числе активно создаются специализированные системы управления сервис-ориентированными НРП в области геоинформатики. Тем самым актуализируется включение веб-сервисов обработки данных (web processing services — WPS) в схему решения задачи. Известными примерами подобных систем являются Business Process Execution Language (BPEL) Designer Project [10] и GeoJModelBuilder [11]. Рассматриваемый в [10] язык BPEL поддерживает стандартизацию описания структуры НРП и позволяет включать в рамках такого описания вызовы сервисов, разрабатываемых различными группами исследователей.

Однако, несмотря на наличие широкого спектра современных систем управления НРП, проблема интеграции функциональных возможностей традиционных систем с новыми возможностями построения сервис-ориентированных процессов и обеспечения дополнительного взаимодействия с геоинформационными системами не решена в полной мере. В этой связи статья направлена на решение вышеупомянутой проблемы. Особое внимание уделено реализации и выполнению НРП на основе контейнеризации.

## 1. Диспетчер НРП

Диспетчер НРП развивает функциональные возможности исполнительной подсистемы инструментального комплекса Orlando Tools [12], предназначенного для создания распределенных научных приложений на основе НРП в гетерогенной среде. Комплекс включает следующие основные подсистемы: пользовательский интерфейс, конструктор концептуальной модели среды, конструктор библиотек модулей, менеджер ПО, менеджер вычислений, менеджер состояния среды, API для доступа к информационно-вычислительным ресурсам и системам. База знаний Orlando Tools содержит описание предметных областей разрабатываемых приложений, вычислительные модели, а также сведения о ресурсах. Исходная информация и результаты выполнения НРП хранятся в расчетных базах данных (РБД) Orlando Tools.

Архитектура диспетчера НРП представлена на рис. 1. Диспетчер реализован на программной платформе Node.js [13]. Он включает системные модули для конвертирования описания предметной области пакета в вычислительную модель на языке BPEL, модули публикации и вызова WPS-сервисов, формулировки задач конечными пользователями пакета на вычислительной модели, планирования вычислений и построения НРП, подготовки вычислительной среды, генерации вычислительных заданий, взаимодействия с локальными менеджерами ресурсов и метапланировщиками среды, работы с базами данных, визуализации НРП и расчетных данных.

Модуль конвертирования производит преобразование описания предметной области пакета, представленное на входном языке конечного пользователя (в данной работе подразумевается язык CMDL [14]), в вычислительную модель на язык BPEL. Конечный пользователь с помощью модуля формулировки задачи определяет ее условия и выбирает доступные ему вычислительные ресурсы. После планирования вычислений и построения НРП осуществляет создание схемы решения задачи, на основе которой модуль генерации производит формирование множества вычислительных заданий для локальных менеджеров ресурсов и метапланировщиков среды в соответствии с НРП. Далее модуль подготовки вычислительной среды выполняет сборку, настройку, запуск и контроль контейнера вычислительной среды. После завершения выполнения НРП имеется возможность визуализировать результаты расчетов.

В рамках вычислительной среды специальный модуль осуществляет взаимодействие с ее локальными менеджерами ресурсов и метапланировщиками в процессе выполнения НРП, а также с модулем работы с базами данных. В рамках исследования используются HTCondor [15] и DAGMan. Дополнительный модуль отслеживает состояние выполнения НРП и осуществляет их визуализацию в пользовательском интерфейсе диспетчера.

Системные модули генерации вычислительных заданий, взаимодействия с локальными менеджерами ресурсов и метапланировщиками среды, работы с базами данных и визуализации НРП включаются в контейнер вычислительной среды.

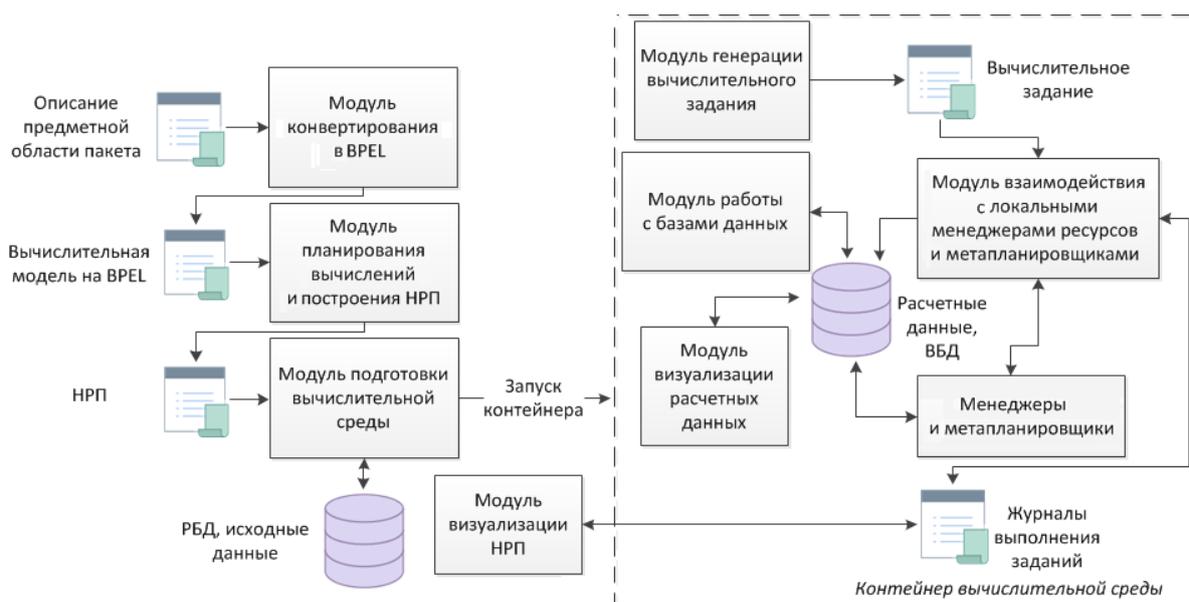


Рис. 1. Архитектура диспетчера НРП  
 Fig. 1. Architecture of the workflow manager

## 2. Методы и средства контейнеризации

Как правило, средой запуска НРП являются суперкомпьютерные центры коллективного пользования (ЦКП) для высокопроизводительных вычислений (high performance computing — НРС). В качестве суперкомпьютеров часто используются НРС-кластеры.

Одними из ключевых свойств любого ЦКП являются высокая стабильность и доступность, что неизбежно влияет на сложность процессов обновления системного и прикладного обеспечения его аппаратных средств. На большей части НРС-кластеров зачастую установлены фиксированные версии библиотек, служб и параметров операционной системы (ОС). Еще один важный фактор — безопасность ЦКП. Поэтому режимы доступа к кластерам ограничены.

Современные НРП состоят из модулей смешанной структуры. Среди них могут быть не только классические параллельные приложения, но и веб-сервисы [16]. Приложения из различных предметных областей могут иметь зависимости от различных библиотек и параметров операционной системы, а также использовать специализированные сопроцессоры и ускорители. Таким образом, в последние годы актуализируется проблема организации смешанной распределенной НРС-среды для выполнения НРП [17].

Популярным и хорошо зарекомендовавшим себя подходом к построению специализированных сред для НРС-вычислений является контейнеризация ПО, в частности на базе Docker [18, 19]. Контейнеризация, в отличие от виртуализации, сокращает накладные расходы на подготовку, хранение и распространение настроенных образов ПО. Все требования для запуска модулей НРП представляются в декларативном виде в текстовых файлах, что позволяет хранить описание инфраструктуры для запуска НРП в виде кода [20]. В ряде случаев вполне достаточно задействовать готовые образы из репозитория Docker. Однако чаще требуется создание Docker-файлов для каждого класса модулей НРП. Такой подход позволяет обеспечить гибкость функционирования среды и проведения экспериментов в ней [21]. В результате сборки Docker-файлов создается настроенный образ, содержащий требуемые библиотеки, модули и исполняемые файлы модулей. Запуск настроенных образов на доступных ресурсах возможен в Docker Compose и с помощью системы оркестрации Kubernetes [22]. В исключительных случаях возможно применение средств развертывания и настройки виртуальных машин с помощью OpenStack [23] и Terraform [24], а также систем автоматизации конфигурирования, например Ansible [25].

Несмотря на то, что Docker упрощает задачи настройки образов и их запуск, он не является лучшим решением для НРС. Как отмечено в работе [26], Docker имеет проблемы с запуском не от привилегированного пользователя ЦКП, а повышение привилегий создает риски информационной безопасности. С другой стороны, Docker включает дополнительный слой абстракции для сети, что создает дополнительные накладные расходы и ограничивает поддержку высокопроизводительных линий связи, таких как InfiniBand. Это обусловило разработку специализированного программного средства Singularity, активно применяемого как в России, так и за рубежом для запуска Docker-контейнеров в высокопроизводительной среде [21, 27]. Singularity обеспечивает полную поддержку образов Docker и нивелирует недостатки Docker по запуску контейнеров. В рамках статьи мы рассматриваем его в качестве базового инструмента запуска Docker-образов. Предполагается, что на доступных конечным пользователям ресурсах Docker и Singularity установлены заранее. Кроме того, образ операционной системы Ubuntu 22.04 используется в качестве базового образа (БО) вычислительной среды.

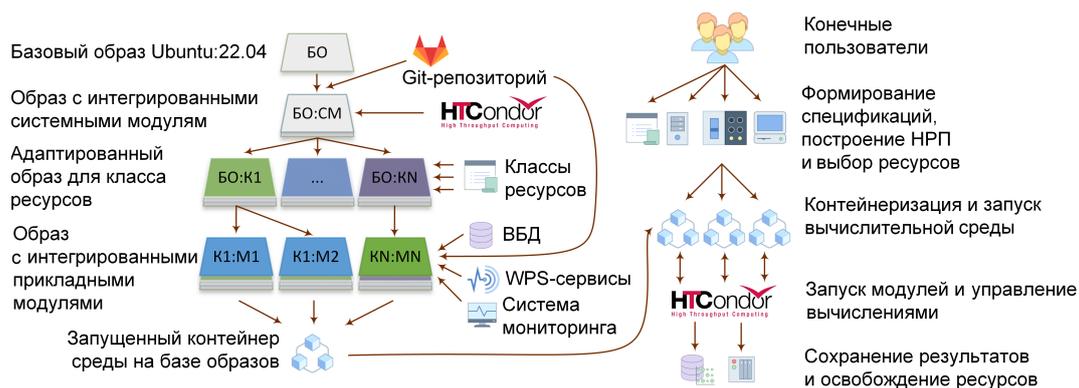


Рис. 2. Схема запуска и выполнения НРП в контейнеризированной вычислительной среде  
 Fig. 2. Scheme for launching and executing workflows in a containerized computing environment

Ниже приведены основные этапы подготовки и запуска вычислительной среды (рис. 2):

- интеграция СПО локальных менеджеров ресурсов, метапланировщиков и диспетчера из Git-репозитория в базовый образ БО:СПО модулем подготовки вычислительной среды;
- построение базовых образов БО:К<sub>1</sub>, ..., БО:К<sub>n</sub> путем адаптации образа БО:СПО к различным классам ресурсов;
- интеграция в адаптированные образы ППО и внешних библиотек из Git-репозитория, а также подключение временной базы данных и системы мониторинга;
- сборка, размещение, запуск и контроль состояния образов на ресурсах;
- выполнение НРП в среде;
- сохранение результатов и освобождение ресурсов.

В целом процесс выполнения НРП состоит в следующем. Специалист предметной области (конечный пользователь) формулирует постановку задачи, подготавливает исходную информацию и исполняемые модули с помощью диспетчера НРП. Затем выбирает необходимые ресурсы (ресурсы суперкомпьютерных центров коллективного пользования, облачных платформ, собственных высокопроизводительных серверов и др.) с учетом квот на их использование и способы доступа к ним. Диспетчер автоматически строит НРП и генерирует спецификацию вычислительного задания. Подсистема контейнеризации осуществляет выделение требуемых ресурсов, подготовку образов в соответствии с классами ресурсов и прикладными модулями НРП с последующим запуском контейнеров на данных ресурсах. По готовности ресурсов для запуска НРП передается сообщение подсистеме управления вычислениями на базе системы HTCCondor, которая также интегрирована в образы. После завершения выполнения НРП метапланировщик отправляет полученные результаты в РБД долгосрочного хранения. Подсистема контейнеризации стирает временную базу данных, освобождает ресурсы и расформирует среду.

### 3. Практическое применение

Диспетчер НРП успешно апробирован в процессе подготовки и проведения экспериментов по решению задач анализа уязвимости энергетических комплексов и отдельных систем энергетики [28]. В рамках распределенного пакета прикладных программ, разработанного для решения этих задач, моделируются отказы элементов энергетичес-

кой системы и их комбинаций для ее различных конфигураций с последующей оценкой влияния отказов на снижение производительности исследуемой системы. С целью ускорения обработки данных расчет суммарных показателей негативного влияния отказов на системную производительность применяется технология In-Memory Data Grid (IMDG) [29]. В качестве средства поддержки технологии IMDG используется инструментарий Apache Ignite [30].

Скриншоты НРП 1 и 2 для глобального анализа уязвимости приведены соответственно на рис. 3 и 4. Глобальный анализ уязвимости позволяет оценить падение производительности энергетической системы относительно числа ее отказавших элементов. Элементы Inputs и Outputs на скриншотах представляют соответственно входные и выходные данные НРП.

Научный рабочий процесс 1 реализует алгоритм нахождения минимально необходимого числа сценариев возмущений с целью получения устойчивой оценки снижения производительности энергетической системы из-за массовых отказов ее элементов.

Операция  $o_1$  извлекает из расчетной базы данных список элементов энергетической системы. Затем операция  $o_2$  определяет необходимое число узлов для формирования IMDG-кластера Apache Ignite и выполняет его настройку. Далее операция  $o_3$  запускает экземпляр IMDG-кластера, на который операция  $o_4$  записывает структуру энергетической системы. Операция  $o_5$  моделирует серию возмущений. Результаты ее выполнения агрегируются операцией  $o_6$ . Узлы IMDG-кластера останавливаются и освобождаются операцией  $o_7$ . В заключение операция  $o_8$  осуществляет проверку достижения мини-

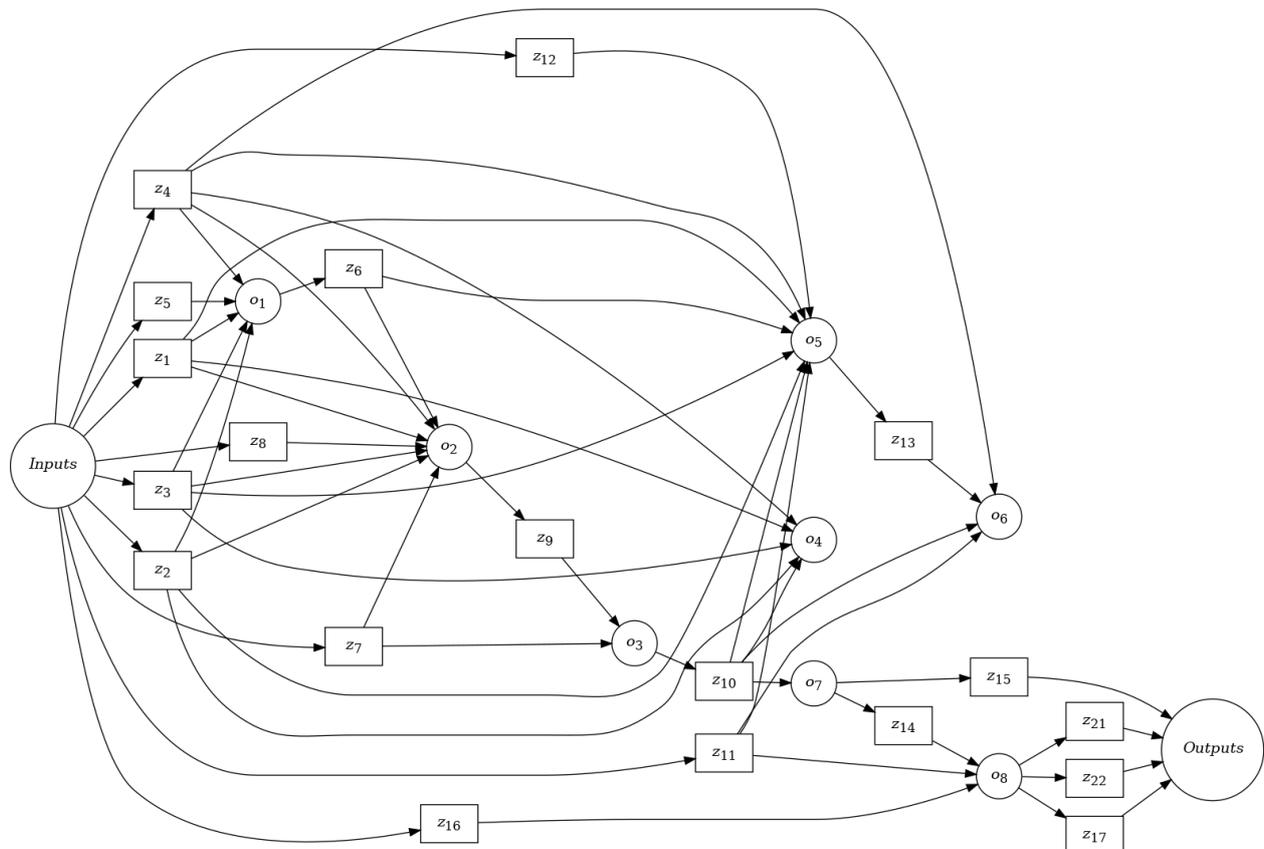


Рис. 3. Научный рабочий процесс 1 для глобального анализа уязвимости  
Fig. 3. Workflow 1 for the global vulnerability analysis

мально необходимого числа сценариев возмущений. В НРП 2 операция  $o_9$  извлекает из РБД множество конфигураций исследуемой энергетической системы. Операция  $o_{10}$  разделяет общую задачу на отдельные подзадачи, в рамках каждой из которых отдельная конфигурация обрабатывается экземпляром операции  $o_{11}$ . Операция  $o_{11}$  отражает НРП 1.  $z_1-z_{22}$  — входные и выходные параметры операций.

В качестве иллюстрации результатов глобального анализа уязвимости на рис. 5 приведен график падения производительности энергетической системы в зависимости от

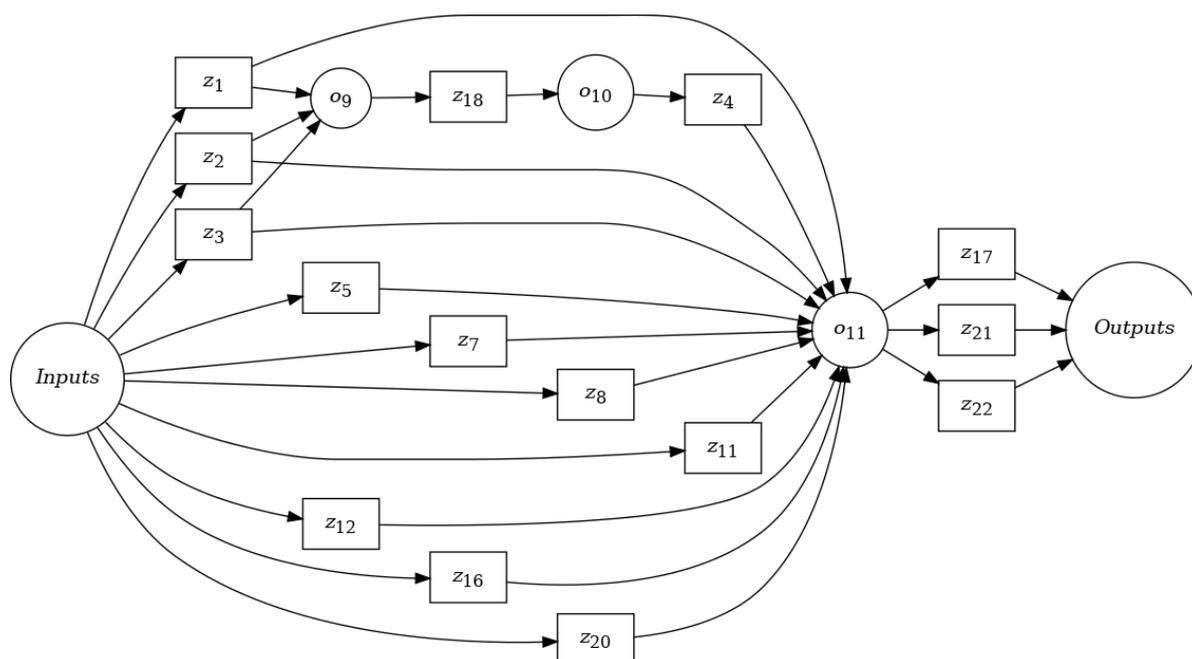


Рис. 4. Научный рабочий процесс 2 для глобального анализа уязвимости  
 Fig. 4. Workflow 2 for the global vulnerability analysis

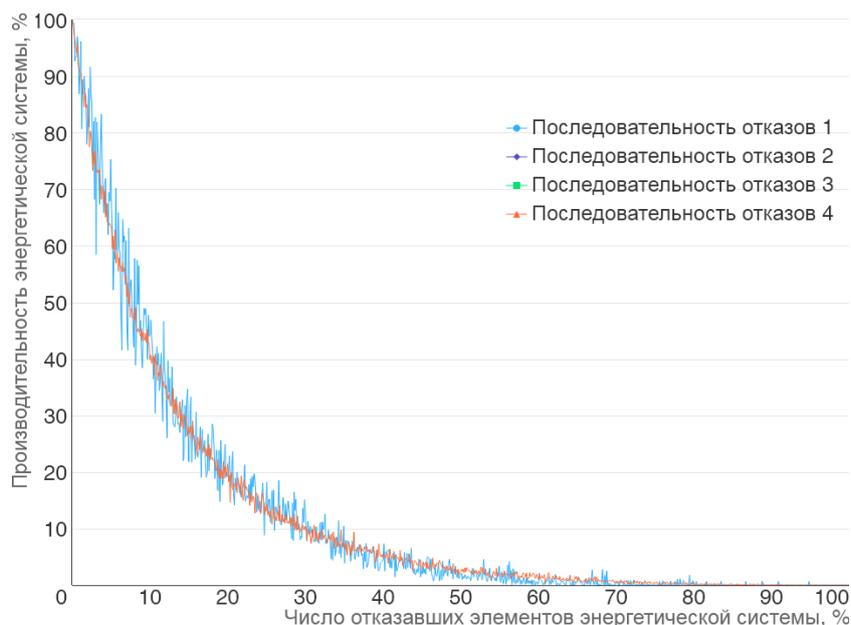


Рис. 5. График падения производительности энергетической системы  
 Fig. 5. Graph of decreasing performance for the energy system

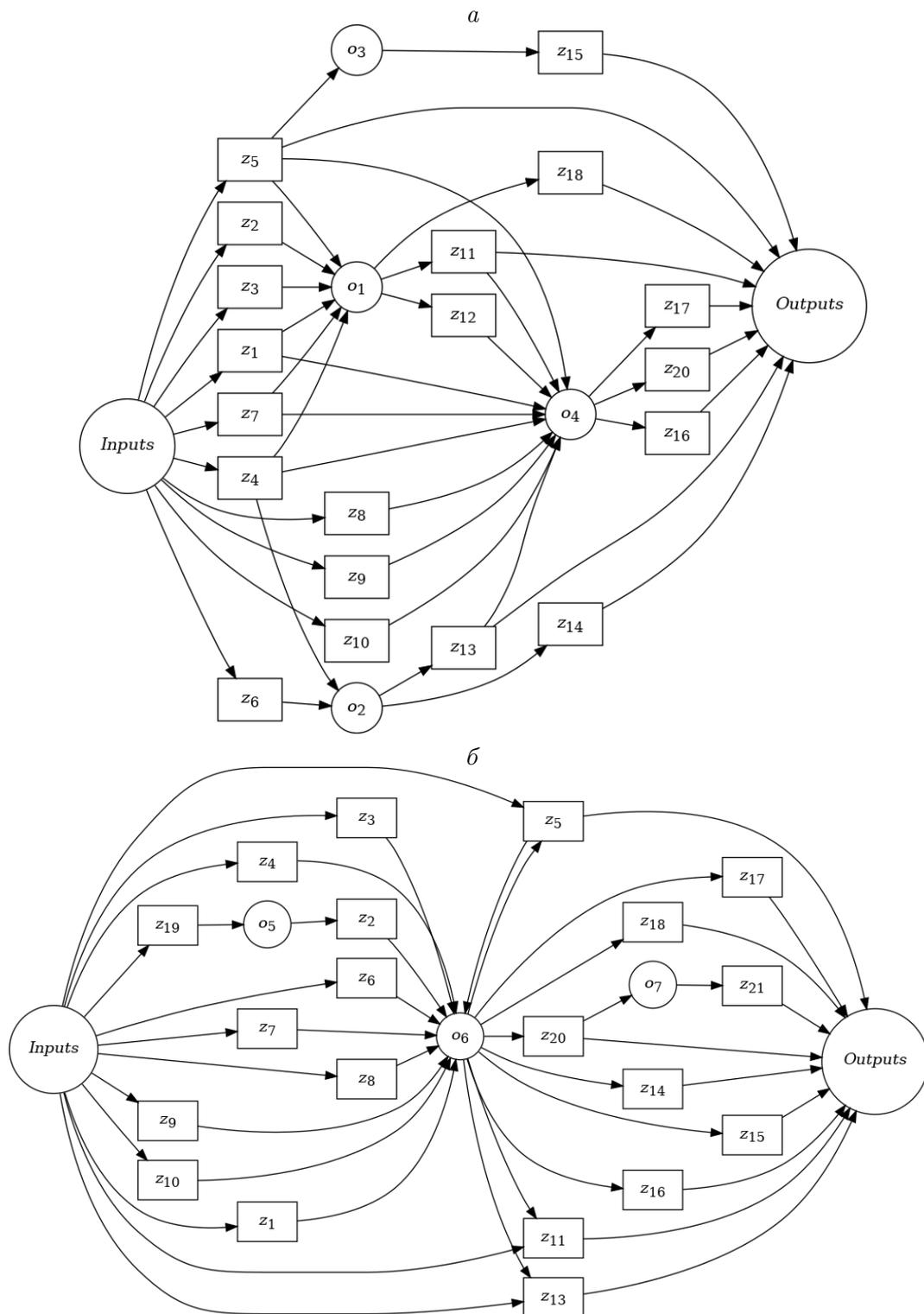


Рис. 6. Научные рабочие процессы 3 (а) и 4 (б) для поиска критических элементов  
 Fig. 6. Workflows 3 (a) and 4 (b) for searching of critical elements

количества ее отказавших элементов для различных последовательностей отказов. Визуализация расчетных данных (выходных параметров) в диспетчере НРП выполняется с помощью библиотеки Highcharts [31]. Визуализация НРП выполняется с помощью пакета утилит Graphviz [32].



Рис. 7. Обеспечение электрической нагрузки инфраструктурного объекта  
Fig. 7. Providing the electrical load of an infrastructure object

Скриншоты НРП 3 и 4 для поиска критических элементов приведены на рис. 6. Здесь под критическим элементом системы понимают такой ее элемент, отказ которого приводит к существенному дефициту в обеспечении электроэнергией. Научный рабочий процесс 3 формирует множество отказов элементов системы, моделирует эти отказы и оценивает их последствия. В НРП 3 операция  $o_1$  строит множество возмущений заданного размера. Операция  $o_2$  устанавливает размер модели. Оценка последствий отказов осуществляется параллельно экземплярами операции  $o_4$ . Операция  $o_3$  определяет новый размер множества возмущений. Научный рабочий процесс 4 находит критические элементы, меняя в цикле максимальное число одновременных отказов элементов. В НРП 4 операция  $o_6$  отражает НРП 3. Операции  $o_5$  и  $o_7$  реализуются системным модулем. Эти операции выполняются для настройки и инициализации кластера Apache Ignite, реализующего технологию IMDG.  $z_1$ – $z_{18}$ ,  $z_{20}$ – $z_{22}$  — входные и выходные параметры операций.

В качестве примера решения задачи поиска критических элементов на рис. 7 показана диаграмма обеспечения электрической нагрузки инфраструктурного объекта Байкальской природной территории в течение суток с помощью фотоэлектрических преобразователей и аккумуляторных батарей при отказе дизель-генераторной установки.

## Заключение

Представлен подход к организации вычислительной среды создания и применения НРП. В рамках подхода предложено интегрированное использование языка VPEL для описания сервис-ориентированных НРП и веб-сервисов обработки данных для взаимодействия с геоинформационными системами, а также методов и средств контейнеризации НРП. Разработан функционирующий прототип диспетчера НРП, реализующий основные аспекты предложенного подхода. Диспетчер НРП осуществляет гибкий доступ к распределенным научным приложениям, в том числе к приложениям, поддерживающим технологию IMDG для обеспечения эффективного доступа к большим данным,

и графическое отображение результатов расчетов на основе конфигурирования и комплексования инструментов автоматизации построения графиков и диаграмм.

Дальнейшее направление исследований предполагает развитие системных модулей диспетчера, связанных с контейнеризацией прикладного и системного ПО.

**Благодарности.** Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 “Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах” (рег. № 121032400051-9).

## Список литературы

- [1] **Barker A., van Hemert J.** Scientific workflow: a survey and research directions. Lecture Notes in Computer Science. Parallel Processing and Applied Mathematic (PPAM 2007), Lecture Notes in Computer Science. 2007; (4967):746–753. DOI:10.1007/978-3-540-68111-3\_78.
- [2] UNICORE. The AnyLogic Company; 2022. Available at: <https://www.unicore.eu> (accessed 22.09.2023).
- [3] DAGMan. The AnyLogic Company; 2022. Available at: <https://htcondor.org/dagman/dagman.html> (accessed 22.09.2023).
- [4] **Deelman E., Singh G., Su M.H., Blythe J., Gil Y., Kesselman C., Mehta G., Vahi K., Berriman G.B., Good J., Laity A., Jacob J.C.** Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Scientific Programming. 2005; 13(3):219–237. DOI:10.1155/2005/128026.
- [5] **Yu J., Buyya R.** A taxonomy of workflow management systems for grid computing. Journal of Grid Computing. 2005; 3(3–4):171–200. DOI:10.1007/s10723-005-9010-8.
- [6] **Bentaleb O., Belloum A.S.Z., Sebaa A.** Containerization technologies: taxonomies, applications and challenges. The Journal of Supercomputing. 2022; (78):1144–1181. DOI:10.1007/s11227-021-03914-1.
- [7] **Balis B.** HyperFlow: a model of computation, programming approach and enactment engine for complex distributed workflows. Future Generation Computer Systems. 2016; (55):147–162. DOI:10.1016/j.future.2015.08.015.
- [8] **Sukhoroslov O.** Supporting efficient execution of workflows on everest platform. Communications in Computer and Information Science. 2019; (1129):713–724. DOI:10.1007/978-3-030-36592-9\_58.
- [9] **Alaasam A.B.A., Radchenko G.I., Tchernykh A.N.** Micro-workflows data stream processing model for industrial internet of things. Supercomputing Frontiers and Innovations. 2021; 8(1):82–98. DOI:10.14529/jsfi210106.
- [10] BPEL designer project. Available at: <https://projects.eclipse.org/projects/soa.bpel> (accessed 22.09.2023).
- [11] GeoJModelBuilder. Available at: <https://github.com/geoprocessing/GeoJModelBuilder> (accessed 22.09.2023).
- [12] **Tchernykh A., Bychkov I., Feoktistov A., Gorsky S., Sidorov I., Kostromin R., Edelev A., Zorkalzev V., Avetisyan A.** Mitigating uncertainty in developing and applying scientific applications in an integrated computing environment. Programming and Computer Software. 2020; 46(8):483–502. DOI:10.1134/S036176882008023X.
- [13] Download Node.js®. Available at: <https://nodejs.org> (accessed 22.09.2023).

- 
- [14] **Edelev A., Beresneva N., Gorsky S., Sidorov I., Feoktistov A.** Representation of subject knowledge from the field of vulnerability analysis of energy systems in distributed applied software packages. Proceedings of the VIth International Workshop “Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security” (IWCI 2019). Atlantis Press; 2019: 184–188. DOI:10.2991/iwci-19.2019.32.
- [15] HTCondor. Available at: <https://htcondor.org> (accessed 22.09.2023).
- [16] **Бычков И.В., Феоктистов А.Г., Горский С.А., Костромин Р.О., Федоров Р.К.** Автоматизация интеграции сервисов веб-обработки данных экологического мониторинга с распределенными научными приложениями. Автометрия. 2022; 58(4):67–75. DOI:10.15372/AUT20220407.
- [17] **Смагин С.И., Сорокин А.А., Мальковский С.И., Королёв С.П., Лукьянова О.А., Никитин О.Ю., Кондрашев В.А., Черных В.Ю.** Организация эффективной многопользовательской работы гибридных вычислительных систем. Вычислительные технологии. 2019; 24(5):49–60 DOI:10.25743/ICT.2019.24.5.005.
- [18] **Баранов А.В., Долгов Б.В., Федотов А.В.** Контейнеризация пользовательских заданий в суперкомпьютерной системе коллективного пользования. Труды научно-исследовательского института системных исследований Российской академии наук. 2019; 9(6):123–131. DOI:10.25682/NIISI.2019.6.0016.
- [19] **Ровнягин М.М., Чугунков И.В., Савченко Н.А.** Технология гибридных контейнеризированных вычислений для организации высокопроизводительной обработки данных в кластерных системах. Вопросы кибербезопасности. 2019; 3(31):39–43.
- [20] **Горский С.А., Костромин Р.О., Сидоров И.А.** Интеграция средств контейнеризации для автоматизации разработки и выполнения пакетов прикладных программ. Материалы IV Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2022). ИДСТУ СО РАН; 2022: 121–162.
- [21] **Sochat V.V., Prybol C.J., Kurtzer G.M.** Enhancing reproducibility in scientific computing: metrics and registry for singularity containers. PloSone. 2017; 12(11):e0188511. DOI:10.1371/journal.pone.0188511.
- [22] Kubernetes. Available at: <https://kubernetes.io> (accessed 22.09.2023).
- [23] OpenStack. Available at: <https://www.openstack.org> (accessed 22.09.2023).
- [24] Terraform. Available at: <https://www.terraform.io> (accessed 22.09.2023).
- [25] Ansible. Available at: <https://www.ansible.com> (accessed 22.09.2023).
- [26] **Azab A.** Enabling Docker containers for high-performance and many-task computing. IEEE International Conference on Cloud Engineering (IC2E). IEEE; 2017: 279–285. DOI:10.1109/IC2E.2017.52.
- [27] **Тахтеев И.А.** Обзор систем контейнеризации Docker и Singularity в рамках кластеров суперкомпьютеров. Молодой ученый. 2022; 21(416):217–229.
- [28] **Beresneva N., Edelev A., Gorsky S., Marchenko M.** Comprehensive analysis of multi-energy system vulnerability. Computational Technologies. 2022; 27(5):79–88. DOI:10.25743/ICT.2022.27.5.008.
- [29] **Gorsky S., Edelev A., Feoktistov A.** Data processing in problem-solving of energy system vulnerability based on in-memory data grid. Mathematics and Its Applications in New Computer Systems (MANCS 2021). Lecture Notes in Networks and Systems. 2022; (424):271–279. DOI:10.1007/978-3-030-97020-8\_25. Available at: [https://link.springer.com/chapter/10.1007/978-3-030-97020-8\\_25](https://link.springer.com/chapter/10.1007/978-3-030-97020-8_25).
- [30] Apache Ignite. Available at: <https://ignite.apache.org> (accessed 22.09.2023).
- [31] Highcharts. Available at: <https://www.highcharts.com> (accessed 22.09.2023).
- [32] Graphviz. Available at: <https://graphviz.org> (accessed 22.09.2023).
-

## INFORMATION TECHNOLOGIES

DOI:10.25743/ICT.2023.28.6.013

**Implementation of computing environment for developing and applying scientific workflows based on containerization**

A. G. FEOKTISTOV\*, R. O. KOSTROMIN, M. L. VOSKOBOINIKOV, D. I. LI-DE

Matrosov Institute for System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Russia

\*Corresponding author: Alexander G. Feoktistov, e-mail: [agf@icc.ru](mailto:agf@icc.ru)*Received October 10, 2023, accepted October 17, 2023.***Abstract**

The paper addresses the design of tools for constructing a heterogeneous computing environment for the development and use of distributed applied software packages. The implementation and execution of scientific workflows (problem-solving schemes) within the package is based on containerization of applied and system software. The essence and novelty of the represented approach lies in the integrated use of the standardized Business Process Execution Language, web processing services and containerization in implementing and executing scientific workflows. This integrated use allows describing the aforementioned language problem-solving schemes that combine services for accessing databases and applied software, which are developed by subject matter experts from different organizations. To ensure scaling of both the process for executing computational jobs and the functioning of systems for managing these jobs in a heterogeneous computing environment through containerization the proposed approach was applied. Finally, the described method was used to integrate scientific workflows with geographic information systems using web processing services. The results of the study were successfully applied for the development and use of a package for analyzing the vulnerability of energy complexes and separated energy systems in the Baikal Natural Territory.

*Keywords:* distributed computing environment, scientific workflows, containerization, applied software package, analysis of energy system resilience.

*Citation:* Feoktistov A.G., Kostromin R.O., Voskoboinikov M.L., Li-De D.I. Implementation of computing environment for developing and applying scientific workflows based on containerization. Computational Technologies. 2023; 28(6):151–164. DOI:10.25743/ICT.2023.28.6.013. (In Russ.)

**Acknowledgements.** The study was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. FWEW-2021-0005 “Technologies for the development and analysis of subject-oriented intelligent group control systems in non-deterministic distributed environments” (reg. No. 121032400051-9).

**References**

1. **Barker A., van Hemert J.** Scientific workflow: a survey and research directions. Lecture Notes in Computer Science. Parallel Processing and Applied Mathematic (PPAM 2007), Lecture Notes in Computer Science. 2007; (4967):746–753. DOI:10.1007/978-3-540-68111-3\_78.
2. UNICORE. The AnyLogic Company; 2022. Available at: <https://www.unicore.eu> (accessed 22.09.2023).
3. DAGMan. The AnyLogic Company; 2022. Available at: <https://htcondor.org/dagman/dagman.html> (accessed 22.09.2023).
4. **Deelman E., Singh G., Su M.H., Blythe J., Gil Y., Kesselman C., Mehta G., Vahi K., Berriman G.B., Good J., Laity A., Jacob J.C.** Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Scientific Programming. 2005; 13(3):219–237. DOI:10.1155/2005/128026.

5. **Yu J., Buyya R.** A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*. 2005; 3(3–4):171–200. DOI:10.1007/s10723-005-9010-8.
6. **Bentaleb O., Belloum A.S.Z., Sebaa A.** Containerization technologies: taxonomies, applications and challenges. *The Journal of Supercomputing*. 2022; (78):1144–1181. DOI:10.1007/s11227-021-03914-1.
7. **Balis B.** HyperFlow: a model of computation, programming approach and enactment engine for complex distributed workflows. *Future Generation Computer Systems*. 2016; (55):147–162. DOI:10.1016/j.future.2015.08.015.
8. **Sukhoroslov O.** Supporting efficient execution of workflows on everest platform. *Communications in Computer and Information Science*. 2019; (1129):713–724. DOI:10.1007/978-3-030-36592-9\_58.
9. **Alaasam A.B.A., Radchenko G.I., Tchernykh A.N.** Micro-workflows data stream processing model for industrial internet of things. *Supercomputing Frontiers and Innovations*. 2021; 8(1):82–98. DOI:10.14529/jsfi210106.
10. BPEL designer project. Available at: <https://projects.eclipse.org/projects/soa.bpel> (accessed 22.09.2023).
11. GeoJModelBuilder. Available at: <https://github.com/geoprocessing/GeoJModelBuilder> (accessed 22.09.2023).
12. **Tchernykh A., Bychkov I., Feoktistov A., Gorsky S., Sidorov I., Kostromin R., Edelev A., Zorkalzev V., Avetisyan A.** Mitigating uncertainty in developing and applying scientific applications in an integrated computing environment. *Programming and Computer Software*. 2020; 46(8):483–502. DOI:10.1134/S036176882008023X.
13. Download Node.js®. Available at: <https://nodejs.org> (accessed 22.09.2023).
14. **Edelev A., Beresneva N., Gorsky S., Sidorov I., Feoktistov A.** Representation of subject knowledge from the field of vulnerability analysis of energy systems in distributed applied software packages. *Proceedings of the VIth International Workshop “Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security” (IWCI 2019)*. Atlantis Press; 2019: 184–188. DOI:10.2991/iwci-19.2019.32.
15. HTCondor. Available at: <https://htcondor.org> (accessed 22.09.2023).
16. **Bychkov I.V., Feoktistov A.G., Gorsky S.A., Kostromin R.O., Fedorov R.K.** Avtomatizatsiya integratsii servisov veb-obrabotki dannykh ekologicheskogo monitoringa s raspredelennymi nauchnymi prilozheniyami [Automation in integrating web processing services of environmental monitoring data with distributed scientific applications]. *Avtometriya*. 2022; 58(4):67–75. DOI:10.15372/AUT20220407. (In Russ.)
17. **Smagin S.I., Sorokin A.A., Malkovsky S.I., Korolev S.P., Lukyanova O.A., Nikitin O.Yu., Kondrashev V.A., Chernykh V.Yu.** Organizatsiya effektivnoy mnogopolzovatel'skoy raboty gibridnykh vychislitelnykh sistem [The organization of effective multi-user operation of hybrid computing systems]. *Computational Technologies*. 2019; 24(5):49–60 DOI:10.25743/ICT.2019.24.5.005. (In Russ.)
18. **Baranov A.V., Dolgov B.V., Fedotov A.V.** Konteynerizatsiya polzovatel'skikh zadaniy v superkompyuternoy sisteme kollektivnogo polzovaniya [Job containerization in a supercomputer job management system]. *Proceedings of the Scientific Research Institute for System Research of the Russian Academy of Sciences*. 2019; 9(6):123–131. DOI:10.25682/NIISI.2019.6.0016. (In Russ.)
19. **Rovnyagin M.M., Chugunkov I.V., Savchenko N.A.** Tekhnologiya gibridnykh konteynerizirovannykh vychisleniy dlya organizatsii vysokoproizvoditel'noy obrabotki dannykh v klasternykh sistemakh [Hybrid containerized computing technology for organizing highperformance data processing in cluster systems]. *Cybersecurity Issues*. 2019; 3(31):39–43. (In Russ.)
20. **Gorsky S.A., Kostromin R.O., Sidorov I.A.** Integratsiya sredstv konteynerizatsii dlya avtomatizatsii razrabotki i vypolneniya paketov prikladnykh programm [Integration of containerization tools to automate the development and execution of application packages]. *Proceedings of the IV International Workshop on Information, Computation and Control Systems for Distributed Environments (ICCS-DE 2022)*. ISDCT SB RAS; 2022: 121–162. (In Russ.)
21. **Sochat V.V., Prybol C.J., Kurtzer G.M.** Enhancing reproducibility in scientific computing: metrics and registry for singularity containers. *PloSone*. 2017; 12(11):e0188511. DOI:10.1371/journal.pone.0188511.
22. Kubernetes. Available at: <https://kubernetes.io> (accessed 22.09.2023).
23. OpenStack. Available at: <https://www.openstack.org> (accessed 22.09.2023).
24. Terraform. Available at: <https://www.terraform.io> (accessed 22.09.2023).

25. Ansible. Available at: <https://www.ansible.com> (accessed 22.09.2023).
26. **Azab A.** Enabling Docker containers for high-performance and many-task computing. IEEE International Conference on Cloud Engineering (IC2E). IEEE; 2017: 279–285. DOI:10.1109/IC2E.2017.52.
27. **Tahteev I.A.** Obzor sistem konteynerizatsii Docker i Singularity v ramkakh klasterov superkomp'yuterov [Overview of Docker and Singularity containerization systems within supercomputer clusters]. Young Scientist. 2022; 21(416):217–229. (In Russ.)
28. **Beresneva N., Edelev A., Gorsky S., Marchenko M.** Comprehensive analysis of multi-energy system vulnerability. Computational Technologies. 2022; 27(5):79–88. DOI:10.25743/ICT.2022.27.5.008.
29. **Gorsky S., Edelev A., Feoktistov A.** Data processing in problem-solving of energy system vulnerability based on in-memory data grid. Mathematics and Its Applications in New Computer Systems (MANCS 2021). Lecture Notes in Networks and Systems. 2022; (424):271–279. DOI:10.1007/978-3-030-97020-8\_25. Available at: [https://link.springer.com/chapter/10.1007/978-3-030-97020-8\\_25](https://link.springer.com/chapter/10.1007/978-3-030-97020-8_25).
30. Apache Ignite. Available at: <https://ignite.apache.org> (accessed 22.09.2023).
31. Highcharts. Available at: <https://www.highcharts.com> (accessed 22.09.2023).
32. Graphviz. Available at: <https://graphviz.org> (accessed 22.09.2023).