



Смягчение неопределенности при разработке научных приложений в интегрированной среде

^{1,2,3} А.Н. Черных, ORCID: 0000-0001-5029-5212 <chernykh@cicese.mx>

⁴ И.В. Бычков, ORCID: 0000-0002-1765-0769 <bychkov@icc.ru>

⁴ А.Г. Феоктистов, ORCID: 0000-0002-9127-6162 <agf65@icc.ru>

⁴ С.А. Горский, ORCID: 0000-0003-0177-9741 <gorsky@icc.ru>

⁴ И.А. Сидоров, ORCID: 0000-0002-2398-5426 <ivan.sidorov@icc.ru>

⁴ Р.О. Костромин, ORCID: 0000-0001-8406-8106 <kostromin@icc.ru>

⁵ А.В. Еделева, ORCID: 0000-0003-2219-9754 <flower@isem.irk.ru>

⁶ В.И. Зоркальцев, ORCID: 0000-0003-1001-1400 <zork@isem.irk.ru>

^{3,7,8,9} А.И. Аветисян, ORCID: 0000-0002-0470-9944 <arut@ispras.ru>

¹ Центр научных исследований и высшего образования
Мексика, 22860, Нижняя Калифорния, Энсенада, ш. Тихуана-Энсенада, 3918

² Южно-Уральский государственный университет,
454080, Россия, Челябинск, проспект Ленина, 76, Россия

³ Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, Москва, ул. А. Солженицына, 25, Россия

⁴ Институт динамики систем и теории управления им. В.М. Матросова СО РАН,
664033, Россия, Иркутск, ул. Лермонтова, 134, а/я 29, Россия

⁵ Институт систем энергетики им. Л.А. Мелентьева СО РАН,
664033, Россия, Иркутск, ул. Лермонтова, 130, Россия

⁶ Лимнологический институт СО РАН,

664033, Россия, Иркутск, ул. Улан-Баторская, 3, а/я 278, Россия

⁷ Московский Государственный университет имени М. В. Ломоносова,
Москва, 119991, ГСП-1, Ленинские горы, д. 1

⁸ Московский физико-технический институт,
141700, Московская область, г. Долгопрудный, Институтский пер., 9

⁹ НИУ "Высшая школа экономики",
101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. В статье представлены новые средства инструментария Orlando Tools. Он используется в качестве основы интегрированного программного окружения для разработки распределенных пакетов прикладных программ. Дополнительные средства Orlando Tools ориентированы на смягчение различных типов неопределенностей, возникающих при распределении заданий в интегрированной вычислительной среде. Они обеспечивают непрерывную интеграцию, доставку и развертывание прикладного и системного программного обеспечения пакетов. Это помогает существенно снизить негативное влияние неопределенности на время решения проблем, надежность вычислений и эффективность использования ресурсов. Экспериментальный анализ результатов решения практических задач наглядно демонстрирует преимущества применения инструментария.

Ключевые слова: распределенная среда; пакеты программ; программное обеспечение; непрерывная интеграция

Для цитирования: Черных А.Н., Бычков И.В., Феоктистов А.Г., Горский С.А., Сидоров И.А., Костромин Р.О., Еделева А.В., Зоркальцев В.И., Аветисян А.И. Смягчение неопределенности при разработке научных приложений в интегрированной среде. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 151-172. DOI: 10.15514/ISPRAS-2021-33(1)-11

Благодарности. Исследование выполнено при поддержке РФФИ, проекты № 19-07-00097-а и № 18-07-01224-а. Разработка агентов метамониторинга и распределения ресурсов выполнена при частичной поддержке научного проекта IV.38.1.1 программы фундаментальных исследований СО РАН.

Mitigating Uncertainty in Developing Scientific Applications in Integrated Environment

^{1,2,3} A. Tchernykh, ORCID: 0000-0001-5029-5212 <chernykh@cicese.mx>

⁴ I.V. Bychkov, ORCID: 0000-0002-1765-0769 <bychkov@icc.ru>

⁴ A.G. Feoktistov, ORCID: 0000-0002-9127-6162 <agf@icc.ru>

⁴ S.A. Gorsky, ORCID: 0000-0003-0177-9741 <gorsky@icc.ru>

⁴ I.A. Sidorov, ORCID: 0000-0002-2398-5426 <ivan.sidorov@icc.ru>

⁴ R.O. Kostromin, ORCID: 0000-0001-8406-8106 <kostromin@icc.ru>

⁵ A.V. Edelev, ORCID: 0000-0003-2219-9754 <flower@isem.irk.ru>

⁶ V.I. Zorkalzev, ORCID: 0000-0003-1001-1400 <zork@isem.irk.ru>

^{3,7,8,9} A.I. Avetisyan, ORCID: 0000-0002-0470-9944 <arut@ispras.ru>

¹ Centro de Investigación Científica y de Educación Superior,
3918, Ensenada-Tijuana Highway, Ensenada, 22860, Mexico

² South Ural State University, Chelyabinsk,
76, Lenin prospekt, Chelyabinsk, 454080, Russia

³ Ivannikov Institute for System Programming of the RAS,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

⁴ Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS,
134, Lermontov st., Irkutsk, postbox 292, 664033, Russia

⁵ Melentiev Energy Systems Institute of the Siberian Branch of the RAS,
130, Lermontov st., Irkutsk, 664033, Russia

⁶ Limnological Institute of the Siberian Branch of the RAS,
3, Ulan-Batorskaya st., Irkutsk, postbox 278, 664033, Russia

⁷ Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia

⁸ Moscow Institute of Physics and Technology (State University)

⁹ Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia

⁹ National Research University Higher School of Economics (HSE)
11 Myasnitskaya Ulitsa, Moscow, 101000, Russia

Abstract. The paper represents new means of the Orlando Tools framework. This framework is used as the basis of an integrated software environment for developing distributed applied software packages. The additional means are focused on mitigating various types of uncertainties arising from the job distribution in an integrated computing environment. They provide continuous integration, delivery, and deployment of applied and system software of packages. This helps to significantly reduce the negative impact of uncertainty on problem-solving time, computing reliability, and resource efficiency. An experimental analysis of the results of solving practical problems clearly demonstrates the advantages of applying these means.

Keywords: distributed environment; software packages; software; continuous integration

For citation: Tchernykh A., Bychkov I.V., Feoktistov A.G., Gorsky S.A., Sidorov I.A., Kostromin R.O., Edelev A.V., Zorkalzev V.I., Avetisyan A.I. Mitigating Uncertainty in Developing Scientific Applications in Integrated Environment. *Trudy ISP RAN/Proc. ISP RAS*, vol. 33, issue 1, 2021, pp. 151-172 (in Russian). DOI: 10.15514/ISPRAS-2021-33(1)-11

Acknowledgements. The study is supported by the Russian Foundation of Basic Research, projects no. 19-07-00097 and no. 18-07-01224. The development of meta-monitoring and resource allocation agents was supported in part by the Basic Research Program of SB RAS, project no. IV.38.1.1.

1. Введение

Анализ современных тенденций развития высокопроизводительных вычислений применительно к математическому моделированию сложных систем позволяет сделать ряд важных выводов. Как правило, такое математическое моделирование связано с решением NP-трудных задач, требующих применения высокопроизводительных вычислительных систем. Однако такие системы значительно различаются по своим вычислительным возможностям, аппаратным и программным платформам, системным архитектурам, интерконнектам, пользовательским интерфейсам и другим компонентам. Зачастую конечные пользователи вычислительных систем сталкиваются со следующими разнообразными трудностями их использования (особенно, при переходе пользователей на новую систему): учетом специфики предметной области, включая использование слабоструктурированных данных, наличие различных неопределенностей, изменение размерности решаемых задач и т.п.; проведением экспериментов в гетерогенной системе, например, адаптацией алгоритмов решения задачи на различные архитектуры узлов системы; прогнозированием производительности узлов и времени выполнения научных приложений.

Необходимо создание универсального информационно-вычислительного окружения, обеспечивающего инструментальную поддержку подготовки и проведения крупномасштабного эксперимента в гетерогенной среде и его специализацию под конкретную предметную область [1]. Такая среда позволит расширить набор используемых моделей, алгоритмов, приложений и баз данных. Кроме того, она обеспечит участие различных научных групп в экспериментах с использованием нужных ресурсов.

Облачные платформы, грид-системы и суперкомпьютерные центры коллективного пользования (ЦКП) отождествляют три организационно-функциональные формы параллельных и распределенных вычислений, которые востребованы научным сообществом.

Грид-системы поддерживают стандартное программное обеспечение (ПО) и сервисы для совместного использования и управления федеративными и географически распределенными ресурсами для вычислений и обработки данных [2].

Облачные платформы быстро и динамично предоставляют необходимые вычислительные ресурсы и хранилища данных, а также обеспечивают пропускную способность сети с требуемым качеством [3]. Тем не менее, конечные пользователи вынуждены самостоятельно доставлять и развертывать свое ПО и данные поверх выделенной инфраструктуры, а также управлять ими. Кроме того, использование облачных ресурсов и сервисов зачастую обходится дороже, чем проведение экспериментов в грид-системах.

Отличительными преимуществами суперкомпьютерных ЦКП является предоставление платных и бесплатных ресурсов и услуг в соответствии с принятыми административными политиками, предоставление базового системного и прикладного ПО, обеспечение унифицированного и жестко регламентированного доступа к вычислительным ресурсам, оценка эффективности вычислений и использования ресурсов, техническая и методическая поддержка конечных пользователей.

В общем случае ресурсы ЦКП являются невыделенными и разделяемыми всеми пользователями центра. Однако по желанию ресурсы могут быть выделены пользователям для решения сложных задач с использованием облачных или грид-вычислений [4]. Различия в способах доступа к вышеперечисленным ресурсам и их использования существенно ограничивают сферу применения высокопроизводительных вычислений исследователями. Таким образом, интеграция облачных и грид-вычислений, включая ресурсы ЦКП, для

поддержки решения трудно решаемых задач обеспечит широкие возможности для проведения массовых масштабных научных экспериментов в такой среде [5].

В статье представлены новые возможности инструментального комплекса Orlando Tools (OT) [6]. Он является основой интегрированной вычислительной среды для разработки и применения распределенных пакетов прикладных программ (РППП). Предполагается, что среда создается на основе трех вышеперечисленных форм организации вычислений.

РППП представляют специальный класс научных приложений [7]. ПО таких пакетов имеет модульную структуру. Модули пакета реализуют алгоритмические знания о процессе решения задач. Набор модулей, используемых для решения задачи, и порядок их выполнения описываются схемой решения задачи. Направленный ациклический граф применяется для отражения отношений и потоков данных между модулями. Понятие схемы решения задачи соответствует понятию абстрактного научного рабочего процесса (workflow) [8]. Спецификация схемы решения задачи (задание) для мета-планировщиков среды описывает информацию об используемых модулях пакета, их входных и выходных данных, требованиях к ресурсам, системам хранения данных и т.п. В отличие от систем поддержки научных рабочих процессов применение РППП связано с решением класса задач в некоторой предметной области с учетом интенсивного изменения алгоритмических знаний.

Применение подсистемы, реализующей новые возможности OT, обеспечивает обработку слабоструктурированных предметных данных, автоматизацию сборки, отладки и совместного тестирования модулей РППП, прогнозирование времени выполнения этих модулей, их классификацию относительно разнородных ресурсов среды и, как следствие, повышение качества распределения ресурсов. Преимущества применения подсистемы показаны на примере разработки двух версий РППП для исследования направлений устойчивого развития топливно-энергетического комплекса (ТЭК) Вьетнама с применением подсистемы [9] и без нее [10]. В первом случае, экспериментальные результаты показали существенное сокращение общего времени подготовки и проведения экспериментов.

2. Связанные исследования

Вопросы создания интегрированного программного окружения для поддержки крупномасштабных научных экспериментов с использованием различных форм организации высокопроизводительных вычислений не очень широко рассматриваются в научной литературе. Тем не менее, можно выделить ряд интересных проектов, имеющих большую практическую значимость организации совместных вычислений.

Модульный инструментальный Distributed and Unified Numerics Environment (DUNE) для решения уравнений в частных производных представлен в [11]. Он обеспечивает использование высокопроизводительных вычислений научными приложениями.

Интеграция приложений, например, популярных систем Computer Aided Engineering, в распределенную вычислительную среду может быть выполнена на основе сервис-ориентированного подхода, предложенного в [12]. Разработан новый алгоритм управления вычислениями в такой среде [13].

Проект SPREXA направлен на поддержку исследований в области интегрированной разработки вычислительных алгоритмов, системного и прикладного ПО, исследования данных и управления ими, среды программирования и инструментальных средств [14].

В [15] описывается веб-ориентированная платформа Galaxy для научного анализа больших биомедицинских наборов данных в области геномики, протеомики, метаболомики и визуализации. Большое число ученых активно использует ее по всему миру в своих крупномасштабных экспериментах.

Гибкая облачная платформа Globus [16] может быть использована для реализации, развертывания и эксплуатации сервисов при поддержке подобных экспериментов. Аналогичные средства, реализованные с использованием агентов, предложены в [17].

Зачастую исследователям необходимо выполнение композиции доступных сервисов с определенными критериями качества. Как правило, задача построения такого композитного сочетания сервисов является NP-трудной. Подход к ее решению представлен в [18].

Концепция и принципы работы интегрированного программного окружения предложены в [1]. Они предполагают разработку инструментальных средств, входящих в такое окружение, для реализации основных этапов крупномасштабных экспериментов, включая поддержку длительного жизненного цикла прикладного и системного ПО. Однако вышеперечисленные и другие существующие проекты оставляют многие проблемы нерешенными в полной мере. В их числе можно выделить следующие проблемы [1, 14]: исследование процессов работы приложений в разнородных ресурсах среды; проектирование приложений с учетом архитектуры среды; применение связующего ПО для поддержки взаимодействия между приложениями и операционными системами, используемыми в узлах среды; интеграция связующего ПО и приложений по управлению программно-аппаратными ресурсами среды; интеллектуализация управления заданиями; автоматизация контроля, сборки и тестирования системного и прикладного ПО; извлечение, анализ и преобразование предметных данных.

Эти проблемы также актуальны для научных рабочих процессов [19]. Их решение способствует смягчению неопределенности в распределении рабочих процессов по ресурсам. Системы поддержки научных рабочих процессов и РППП имеют ряд отличий. Как правило, научные рабочие процессы, такие как LIGO, Montage, CyberShake, SIPHT или Epigenomics, ориентированы на решение одной задачи в некоторой предметной области. Пакет предназначен для решения класса задач. Схемы решения задач отражают многообразие используемых в пакете алгоритмов. Научные рабочие процессы не претерпевают существенных изменений при их использовании [20]. Пакеты, напротив, активно развиваются. Их алгоритмические знания интенсивно изменяются. В пакетах могут разрабатываться новые схемы решения задач или модифицироваться существующие схемы. Анализ состояния исследований в данной области показывает, что в настоящее время нет средств для реализации непрерывной интеграции ПО применительно к РППП в полной мере [21]. Требуется разработка новых средств для решения данной проблемы. Такие средства должны обеспечивать выполнение специальных функций непрерывной интеграции, обусловленных спецификой пакетов, и взаимодействие с существующими средствами непрерывной интеграции, реализующими традиционные функции. Проведенный сравнительный анализ возможностей известных средств непрерывной интеграции показывает, что для реализации подобного взаимодействия наиболее подходит GitLab [22].

В отличие от большинства научных рабочих процессов [23], пакеты создаются для выполнения в гетерогенной распределенной вычислительной среде. Ресурсы, интегрируемые в рамках среды, могут быть выделенными и невыделенными. Возможностей традиционных средств виртуализации, обычно используемых системами поддержки научных рабочих процессов [24], недостаточно для интеграции вышеупомянутых ресурсов в единую среду.

Сравнительный анализ систем виртуализации представлен в [9]. Он показал, что в динамической среде необходимо применять связующее ПО для поддержки эффективного взаимодействия с различными гипервизорами и системами управления контейнерами. Такой системой, которая может использоваться в выделенных ресурсах, является платформа OpenStack [25]. Однако для невыделенных ресурсов необходима разработка специального средства запуска виртуальных машин (VM).

Вышеупомянутые отличия приводят к различным типам неопределенностей, которые возникают при планировании вычислений и распределении ресурсов. В их числе

неопределенность в возможностях ресурсов, свойствах заданий, предпочтениях владельцев и пользователей ресурсов, качестве услуг и др. С точки зрения неопределенности, основными проблемами в масштабируемых научных приложениях являются планирование вычислений и распределение ресурсов. Активно развиваются исследования по улучшению планирования и распределения за счет уменьшения перечисленных выше неопределенностей.

Методы анализа структур рабочего процесса и их декомпозиции рассмотрены в [26]. Анализ рабочего процесса позволяет уточнить свойства его компонентов. Декомпозиция проводится с целью рационального распределения компонентов по ресурсам. Кроме того, автоматизация планирования рабочих процессов (схем решения задач) для непроцедурных формулировок задач позволяет значительно упростить выбор приложений, необходимых для решения задач, при наличии неопределенности в использовании программного обеспечения [27].

Алгоритм уменьшения неопределенности распределения заданий проблемно-ориентированных приложений, учитывающий специфику решаемых задач, предложен в [28]. Адаптивное распределение ресурсов в условиях неопределенности исследуется в [29]. Предложены модель задержки выполнения заданий с целью прогнозирования их времени выполнения и стратегия адаптивного стохастического распределения заданий. Предложен подход к моделированию облачных вычислений с неопределенностью [30], учитывающий представление ресурсов в гибридной среде и обеспечивающий повышение эффективности предоставления ресурсов с учетом их конфиденциальности, целостности и доступности.

Неопределенность качества услуг, связанная с координацией предпочтений владельцев и пользователей ресурсов, может быть смягчена путем применения агентных технологий [31].

3. Orlando Tools: основа для интегрированного программного окружения

Архитектура интегрированной вычислительной среды показана на рис. 1. Подсистемы ОТ представляют ее основу. Внешние информационно-вычислительные системы и ресурсы используются в процессе разработки и применения РППП.

Выделяются следующие основные подсистемы ОТ: *пользовательский интерфейс*, обеспечивающий доступ различных категорий пользователей РППП к их компонентам, подсистемам ОТ и ресурсам среды; *конструктор концептуальной модели среды*, реализующий спецификацию знаний пользователей РППП о предметной области решаемых задач; *конструктор библиотек модулей*, поддерживающий разработку и модификацию прикладного ПО РППП; *менеджер ПО*, осуществляющий автоматизацию контроля версий, тестирование, доставку и развертывание как прикладного и системного ПО РППП среды; *менеджер вычислений*, реализующий управление вычислениями на разнородных ресурсах среды; *менеджер состояния среды*, выполняющий мониторинг ее ресурсов; *API* для доступа к информационно-вычислительным системам и ресурсам.

Знания о предметных областях РППП, информация о ресурсах среды и сведения об информационно-вычислительных системах и ресурсах сохраняются в соответствующих разделах базы знаний ОТ. Для извлечения таких знаний могут быть использованы различные базы предметных данных. Исходные данные и результаты вычислений хранятся в базах расчетных данных инструментального комплекса. В процессе вычислений могут использоваться дополнительные системы хранения данных.

Пользовательский интерфейс. Компоненты подсистем ОТ представлены сервисами. Для пользователей реализован веб-ориентированный доступ к их операциям. Сервисы, представляющие компоненты, к которым могут обращаться другие программные сущности (другие компоненты, модули пакетов или внешние программные системы), имеют дополнительное описание на языке WSDL.

Конструктор концептуальной модели. Концептуальная модель среды, формируемая в рамках конструктора, состоит из трех частей. Первая часть описывает алгоритмические знания о решении задач предметной области РППП. Вторая часть содержит информацию об инфраструктурных объектах среды. Знания о процессах непрерывной интеграции, доставки и развертывания ПО на ресурсах среды формируют третью часть модели.

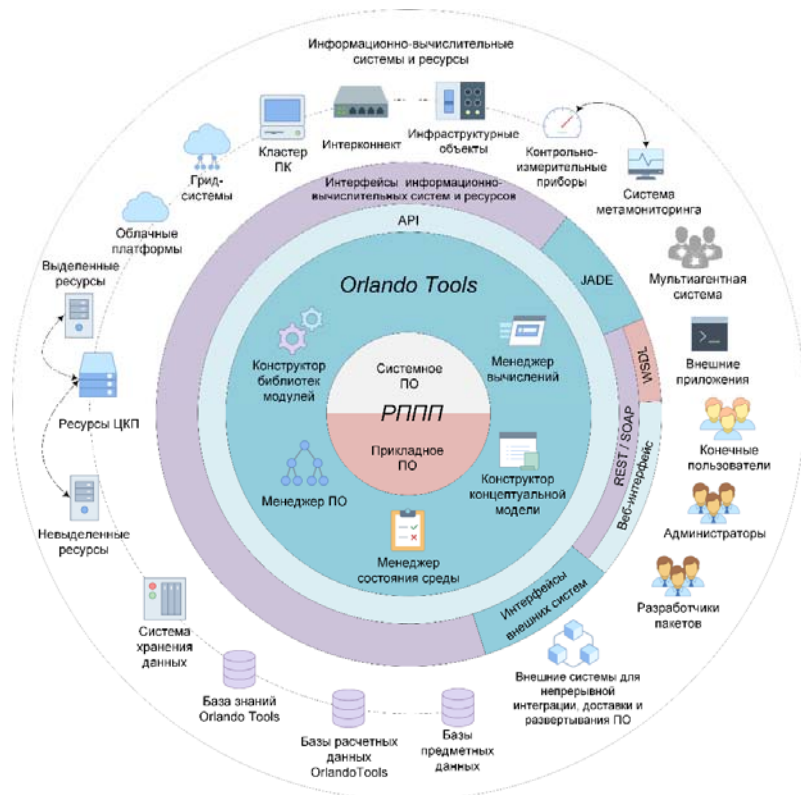


Рис. 1. Архитектура интегрированной вычислительной среды
Fig. 1. Architecture of integrated computing environment

Алгоритмические знания являются результатом структурного анализа предметной области. В ОТ она описывается с помощью специального набора концептуальных понятий. В их числе такие концепты, как параметры, операции, модули и схемы решения задач.

Параметры представляют характеристики предметной области, необходимые в процессе решения задач. Над множеством параметров определяются операции. Они ставят в соответствие подмножеству входных параметров, значения которых должны быть известны, другое подмножество выходных параметров, значения которых нужно вычислить. Связи между операциями определяются на основе их входных и выходных параметров. Продукции определяют логические условия применения операций в процессе решения задач в зависимости от промежуточных результатов вычислений и состояния ресурсов среды.

Схемы решения задач создаются разработчиками РППП и их конечными пользователями по процедурной или не процедурной постановкам задач на основе информационных связей

между операциями. Процедурная постановка задачи представляет собой список операций, требуемых для решения задачи и частично-упорядоченных в рамках ярусно-параллельной формы направленного ациклического графа. Непроцедурная постановка задачи формулируется в следующем виде: «По исходному подмножеству параметров, значения которых известны, вычислить значения параметров целевого подмножества». В случае такой постановки задачи, нахождение операций, требуемых для решения задачи, и их упорядочение выполняются автоматически. Модули ассоциируются с программными реализациями операций. Каждая схема решения задачи через свои операции определяет набор модулей, необходимых для ее выполнения. Описание модуля может включать текст модуля, язык программирования, тип и назначение входных, выходных и транзитных параметров, методы передачи параметров и обработки нестандартных ситуаций, требуемый компилятор и его опции, формат вызова, и другая необходимая информация.

В рамках данного исследования представлены новые возможности ОТ по автоматизации контроля версий и тестирования ПО РППП. В связи с этим описание модулей расширено дополнительными сведениями о репозиториях с их исходными и бинарными файлами, сценариях их сборки и тестирования в различных узлах среды как в отдельности, так и в составе схем решения задач, тестовых данных, способах доставки и развертывания модулей на ресурсах среды. В частности, конструктор концептуальной модели обеспечивает разработку необходимых скриптов для внешних систем. Эти скрипты реализуют сценарии непрерывной интеграции, доставки и развертывания ПО.

Информация об инфраструктурных объектах среды включает характеристики узлов, каналов связи, сетевых устройств, топологии сети и других объектах. Кроме того, эти характеристики включают в себя статистику сбоев и отказов программного обеспечения и оборудования. Дополнительно учитываются сведения об административных политиках, определенных для ресурсов среды. Такие сведения включают данные о пользователях и их заданиях, правах доступа к ресурсам и квотах на их использование, а также системах управления ресурсами, их характеристиках и дисциплинах диспетчеризации вычислительных работ.

Информация об инфраструктуре среды и административных политиках извлекается с помощью различных информационно-вычислительных и управляющих систем, контрольно-измерительных приборов и систем мониторинга. Она отображается в концептуальной модели с помощью системных параметров, операций и модулей. Описание концептуальной модели хранится в базе знаний ОТ. Ее разделы распределены между ОТ, системой метамониторинга и внешними системами непрерывной интеграции, доставки и развертывания ПО.

Конструктор библиотек модулей. Прикладное ПО РППП может формироваться на основе существующих программных библиотек или создаваться разработчиками пакета. В обоих случаях описание модулей включается в концептуальную модель среды. ОТ предоставляет редактор исходного кода модулей и средства работы с внешними системами их сборки, отладки и тестирования. Системное ПО формируется на основе системных библиотек ОТ.

Менеджер ПО является новой подсистемой ОТ. Он предназначен для выявления и частичного устранения потенциальных проблем при непрерывной интеграции, доставки и развертывания новых версий как прикладного, так и системного ПО РППП.

Его основные функции включают: работу с репозиториями исходных и бинарных файлов ПО, включая поддержку контроля их версий; автоматизацию сборки и тестирования модулей в отдельности; создание образов ВМ для выполнения модулей; автоматизацию тестирования модулей в составе схем решения задач; прогнозирование времени выполнения модулей в узлах среды; автоматизацию доставки и развертывания модулей в среде.

Новые средства для мета-описания процессов выполнения всех перечисленных выше операций впервые реализованы в конструкторе концептуальной модели среды. Общая схема непрерывной интеграции, доставки и развертывания прикладного и системного ПО представлена на рис. 2.

Сведения о производительности узлов, полученные при тестовом выполнении модулей, позволяют установить степень соответствия производительности ресурсов критериям эффективности выполнения модулей. Такое соответствие устанавливается администраторами среды заблаговременно, до начала назначения ресурсов в процессе проведения вычислительных экспериментов. Тем самым, обеспечивается существенное смягчение неопределенности в последующем процессе распределения заданий по ресурсам.

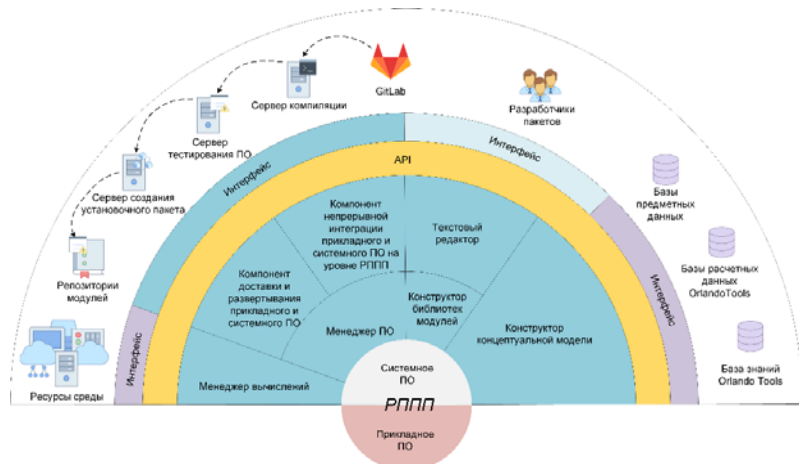


Рис. 2. Схема непрерывной интеграции, доставки и развертывания ПО
Fig. 2. Scheme of continuous integration, delivery, and deployment of software

Менеджер вычислений включает компоненты для формулировки постановок задач и построения схем их решения, генератор заданий для выполнения построенных схем, планировщик и интерпретатор. Планировщик производит декомпозицию исходных схем решения задач, представленных в заданиях, на подсхемы с целью оптимизации распределения вычислительной и коммуникационной нагрузки на ресурсы среды. Интерпретатор осуществляет выполнение схем или подсхем решения задач.

Дополнительно менеджер вычислений может передать задание на обработку мультиагентной системе (MAC) [32]. Она реализована на основе Java Agent Development Framework (JADE) [33]. MAC относится к классу метапланировщиков, таких как GridWay [34] или Condor DAGMan [35]. В то же время, в отличие от подобных средств, она позволяет управлять потоками заданий в среде, интегрирующей ресурсы облачных платформ, грид-систем и ЦКП. MAC включает агентов классификации заданий, мониторинга и диспетчеризации ресурсов среды, а также агентов, непосредственно представляющих интегрированные ресурсы (ресурсных агентов). Ресурсные агенты объединяются в новое виртуальное сообщество для выполнения каждого задания. Условием включения агента в сообщество для выполнения конкретного задания является наличие у него ресурсов, обладающих необходимыми возможностями для выполнения модулей данного задания.

Схема распределения ресурсов агентами представлена на рис. 3. Агент-классификатор определяет класс задания и формирует виртуальное сообщество ресурсных агентов. Система классификации детально рассмотрена в [36]. Классифицированное задание передается в очередь агенту-диспетчеру. При обработке очереди заданий агент-диспетчер производит их декомпозицию на подзадания, которые затем распределяются по сегментам интегрированной вычислительной среды, представляющим облачные платформы, грид-системы и ЦКП.

Для выполнения подзаданий запускаются ВМ, образы которых предварительно созданы менеджером ПО. ВМ объединяются в единую виртуальную среду выполнения РППП.

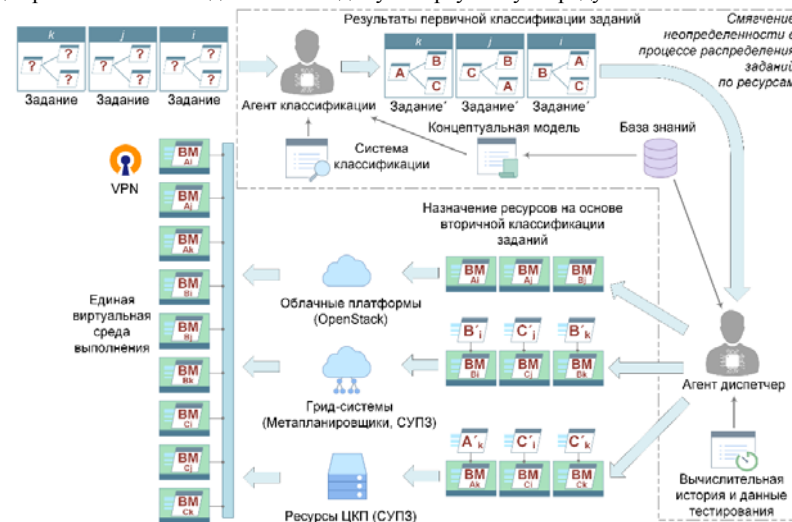


Рис. 3. Назначение ресурсов агентами
Fig. 3. Resource allocation by agents

Управление облачными ресурсами осуществляется с помощью OpenStack. При запуске ВМ на ресурсах грид-систем и ЦКП формируются задания для систем управления прохождением заданий (СУПЗ), установленных на ресурсах. Эти задания отправляются в очереди СУПЗ. Для ускорения прохождения заданий выполняется поиск узлов со свободными слотами в их расписаниях обслуживания заданий. Каждый слот определяет число ядер, которые могут быть использованы для выполнения задания в течение некоторого интервала времени.

В целом агенты обеспечивают смягчение неопределенности в распределении заданий по ресурсам. Оно осуществляется автоматически в четыре этапа.

На первом этапе осуществляется первичная классификация задания путем анализа его спецификации с помощью специальной характеристической функции. Проверяется формальное соответствие характеристик задания и их значений характеристикам предопределенных классов и допустимым областям их значений. В результате первичной классификации задание может принадлежать нескольким классам. Выбор ресурсов, соответствующих классу задания, производится на втором этапе. В рамках третьего этапа выполняется вторичная классификация заданий на основе анализа статистической информации о вычислительной истории выполнений задания и их соответствия имеющимся классам. Результаты анализа применяются для корректировки данных первичной классификации. На последнем этапе результаты вторичной классификации уточняются на основе полученных ранее данных тестирования прикладного ПО соответствующим менеджером или сведениям о прогнозируемом времени выполнения модулей.

Первые два этапа выполняются агентом классификации заданий. Последние два этапа осуществляются агентом-диспетчером. Предварительно администраторы определяют необходимые классы заданий в системе классификации и затем устанавливают соответствие ресурсов классам на основе своего экспертного опыта. Окончательное распределение заданий по ресурсам выполняется ресурсными агентами с помощью тендера вычислительных работ (заданий) [37]. В рамках тендера торги реализованы на основе закрытого

однораундового аукциона Викри второй цены [38]. В процессе торгов учитываются дополнительные условия выполнения заданий. В их числе предпочтения владельцев ресурсов и критерии эффективности решения задач, определяемые пользователями среды.

Менеджер состояния среды получает информацию о состоянии среды от системы метамониторинга, реализованной на основе агентного подхода [32]. Эта система осуществляет сбор сведений о состоянии инфраструктурных объектов среды с помощью контрольно-измерительных приборов (Sensors, APC PowerChute, IPMI, SMART и др.) и локальных систем мониторинга (Ganglia, Nagios, Zabbix, ZenOSS и др.). Дополнительно она осуществляет контроль показателей функционирования систем управления этими объектами. На основе собранной информации система метамониторинга выполняет оценку текущей вычислительной ситуации, прогнозирует ее развитие и формирует управляющие воздействия на инфраструктурные объекты и системы управления с целью предупреждения или частичного устранения сбоев и отказов программно-аппаратного обеспечения. В случае критической ситуации, когда такие действия невозможно выполнить в автоматическом режиме, система метамониторинга отправляет уведомление администратору среды.

Доступ к информационно-вычислительным системам и ресурсам. Веб-ориентированный доступ пользователей к информационно-вычислительным системам и ресурсам среды обеспечивается средствами ОТ. Данный комплекс взаимодействует с системами и ресурсами по протоколам, поддерживаемыми ими.

4. Смягчение неопределенности

В статье неопределенность рассматривается с точки зрения недостаточности имеющихся знаний относительно знаний, необходимых для принятия субоптимального решения. В нашем случае такими знаниями являются время выполнения модулей в разнородных узлах среды, свойства заданий и возможность выполнения заданий на тех или иных ресурсах.

Оценка времени выполнения модуля. Разработан алгоритм оценки времени выполнения модулей. Модуль запускается из среды профилирования программ на выбранном эталонном узле. В процессе его выполнения определяется время его работы с различными компонентами узла. Общее время выполнения модуля на целевом узле прогнозируется на основе анализа различий характеристик узлов, влияющих на ускорение (замедление) вычислений. Алгоритм включает следующие этапы:

- определение эталонного и целевого узлов;
- определение множества $CR = \{cr_1, cr_2, \dots, cr_m\}$ характеристик эталонного узла, используя знания о ресурсах концептуальной модели среды;
- определение множества $CT = \{ct_1, ct_2, \dots, ct_m\}$ характеристик целевого узла на основе тех же знаний;
- подготовка данных d в расчетной базе знаний ОТ;
- размещение профилировщика, модуля и подготовленных данных на эталонном узле с использованием менеджера ПО;
- запуск профилировщика на эталонном узле;
- запуск модуля из среды профилировщика;
- выполнение модуля на эталонном узле;
- получение значений счетчиков профилировщика;
- определение реального времени $T_r(d)$ выполнения модуля;
- вычисление значений $p_1(d), p_2(d), \dots, p_n(d)$ из множества P , которые отражают вычислительную нагрузку узла (число выполненных целочисленных операций, число

операций с плавающей запятой, число обращений к основной памяти и кэш-памяти разных уровней, число пропущенных обращений и др.) при выполнении модуля в зависимости от данных d ;

- оценка времени $\hat{T}_r(d) = \sum_{i=1}^k f_i(CR, P, g(d))$ выполнения модуля на эталонном узле;
- оценка времени $\hat{T}_t(d) = \sum_{i=1}^k f_i(CT, P, g(d))$ выполнения модуля на целевом узле;
- определение погрешности $\varepsilon = T_r(d) - \hat{T}_r(d)$;
- оценка времени $\hat{T}_t(x)\hat{T}_t(d) + \varepsilon \frac{\hat{T}_r(d)}{\hat{T}_t(d)}$ выполнения модуля на целевом узле.

В алгоритме характеристики $cr_1, cr_2, \dots, cr_m, ct_1, ct_2, \dots, ct_m$, компоненты данных d , время T_r , оценки \hat{T}_r и \hat{T}_t , а также погрешность ε являются системными параметрами. Функции $p_1, p_2, \dots, f_1, f_2, \dots, f_k$ и g , а также выражения для вычисления \hat{T}_r, \hat{T}_t , и ε реализуются системными модулями. Системные объекты описываются в концептуальной модели среды.

Смягчение неопределенности в распределении заданий по ресурсам. Для оценки степени смягчения неопределенности в процессе распределения заданий по ресурсам в статье используется информационная двоичная энтропия. Данный показатель часто используется для оценки степени неопределенности сложной системы. Предполагается, что уменьшение энтропии ведет к смягчению неопределенности.

Пусть по результатам первичной классификации найдены m классов, характеристикам которых удовлетворяет задание. Среда включает n разнородных ресурсов. Одному классу может соответствовать k ресурсов, $k \in \overline{1, n}$. Информация о соответствии между классами и ресурсами представлена булевой матрицей Y размерности $n \times m$. Элемент матрицы $y_{ji} = 1$ ($y_{ji} = 0$) показывает, что j -й ресурс соответствует (не соответствует) i -му классу. Определим энтропию $E^c(x_c) = \sum_{i=1}^m p_i(x_c)E_i^c(x_c)$, отражающую степень неопределенности принадлежности задания одному из классов, где x_c – случайное событие, которое обуславливает принадлежность задания одному из m классов, $p_i(x_c)$ – вероятность такого события относительно i -го класса, $\sum_{i=1}^m p_i(x_c) = 1$, $E_i^c(x_c)$ – энтропия, соответствующая данному событию. Значение $E_i^c(x_c)$ вычисляется следующим образом:

$$E_i^c(x_c) = \begin{cases} 0, & \text{если } p_i(x_c) = 0, \\ -\log_2 p_i(x_c) & \text{в противном случае.} \end{cases}$$

Тем же способом определим энтропию $E^r(x_r) = \sum_{j=1}^n p_j(x_r)E_j^r(x_r)$, отражающую степень неопределенности принадлежности задания одному из классов, где x_r – случайное событие, которое обуславливает возможность назначения заданию одного из n ресурсов, $p_j(x_r)$ – вероятность такого события относительно j -го ресурса, $E_j^r(x_r)$ – энтропия, соответствующая данному событию. Значения $p_j(x_r)$ и $E_j^r(x_r)$ вычисляются следующим образом:

$$p_j(x_r) = \sum_{l: y_{jl}=1} p_l(x_c) / \sum_{l: y_{jl}=1} p_l(x_c), \quad l \in \overline{1, m}, \quad \sum_{j=1}^n p_j(x_r) = 1,$$

$$E_j^r(x_r) = \begin{cases} 0, & \text{если } p_j(x_r) = 0, \\ -\log_2 p_j(x_r) & \text{в противном случае.} \end{cases}$$

Рассмотрим иллюстративный пример смягчения неопределенности в процессе распределения заданий по ресурсам. Пусть по результатам первичной классификации найдены 8 классов, характеристикам которых удовлетворяет задание, поступившее в среду для выполнения указанного в задании модуля. Эти классы различаются между собой областью допустимых значений одной характеристики, определяющей допустимое время выполнения заданий. Значение данной характеристики не указано в спецификации задания. Среда включает 3 разнородных ресурса (3 кластера, узлы которых различаются по вычислительным характеристикам). Соответствие ресурсов классам представлено матрицей

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Имеется вычислительная история выполнения такого задания на ресурсах среды.

В табл. 1 приведены значения $p_1(x_c)$, $p_2(x_c)$, ..., $p_8(x_c)$, E^c , $p_1(x_r)$, ..., $p_3(x_r)$ и E^r для каждого этапа смягчения неопределенности в распределение заданий по ресурсам.

Табл. 1. Вероятности событий и показатели энтропии

Table 1. Event probabilities and entropy parameters

№	$p_1(x_c)$	$p_2(x_c)$	$p_3(x_c)$	$p_4(x_c)$	$p_5(x_c)$	$p_6(x_c)$	$p_7(x_c)$	$p_8(x_c)$	E^c	$p_1(x_r)$	$p_2(x_r)$	$p_3(x_r)$	E^r
1	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	3.00	0.333	0.333	0.333	1.06
2	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	3.00	0.444	0.556	0	0.99
3	0.115	0.741	0.132	0.012	0	0	0	0	1.14	0.988	0.012	0	0.09
4	0	0.936	0.064	0	0	0	0	0	0.34	1.000	0	0	0

На первом этапе вероятности принадлежности задания одному из 8 классов равны между собой, так же как равны и вероятности назначения заданию одного из 3 ресурсов. В этом случае $E^c = 3.00$ и $E^r = 1.06$. На следующем этапе выбираются два ресурса, наиболее подходящие для выполнения задания с точки зрения экспертного опыта администраторов ресурсов. При этом степень неопределенности назначения ресурсов заданию понижается до $E^r = 0.99$. Вторичная классификация задания на основе анализа статистической информации о вычислительной истории конкретизирует вероятности принадлежности задания классам, полученные на третьем этапе. Значение E^c становится равным 1.14. В соответствие с этим изменяются вероятности назначения ресурсов заданию. Степень неопределенности назначения ресурсов заданию уменьшается до $E^r = 0.09$. На последнем этапе результаты вторичной классификации уточняются на основе прогнозного времени выполнения модуля или полученных ранее данных его тестирования. В результате этого $E^c = 0.34$ и $E^r = 0.09$. Таким образом, ресурс 1 назначается для выполнения задания.

5. Экспериментальный анализ

РППП для решения задач поиска направлений устойчивого развития топливно-энергетического комплекса. Поиск направлений устойчивого развития как ТЭК в целом, так и отдельных систем энергетики является типовой задачей. Необходимость ее решения многократно возникает в процессе их эксплуатации и развития. Для решения подобной задачи разработан РППП. Развитие ТЭК исследуется с учетом всех комбинаций крупных возмущений, которые могут привести к недостаточному обеспечению конечных потребителей энергоресурсами. Такое исследование приводит к возникновению большого числа возможных состояний комплекса и затрудняет их анализ классическими методами.

В пакете используется новый подход к исследованию развития ТЭК с помощью комбинаторного моделирования. Он основан на рассмотрении различных комбинаций возможных состояний комплекса и переходов между ними для заданного периода T . Поскольку множество возможных состояний является достаточно большим, рассматривается только подмножество допустимых состояний. Множество состояний представлено ориентированным графом $G = \langle V, E \rangle$, где V – множество вершин (возможных состояний ТЭК), а E – множество ребер (переходов между состояниями). Дуги направлены из состояния s_0 в начальный момент времени t_0 в направлении состояний в следующий момент времени t_1 и далее в состояния в следующие моменты времени t_2, t_3, \dots, t_n (рис. 4). Каждая вершина отражает состояние взаимосвязанных объектов комплекса. Такое состояние включает эксплуатацию, реконструкцию, модернизацию и создание объектов. Каждый объект выполняет одну или несколько технологических операций (хранение, производство, обработку, преобразование и транспортировку энергоресурсов). Комбинаторный анализ

проводится путем изучения различных комбинаций состояний объектов и переходов между состояниями за период T . Цель состоит в том, чтобы найти подграф графа G , который включает пути с минимальными затратами на развитие ТЭК.

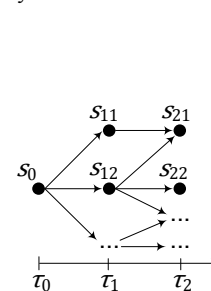


Рис. 4. Фрагмент графа G
Fig. 4. Fragment of the graph G

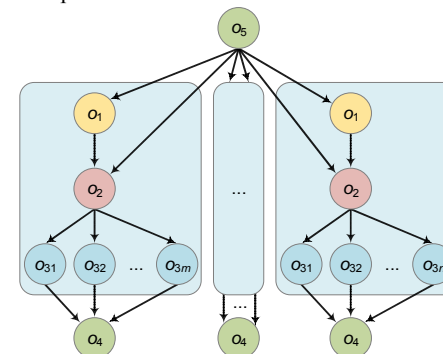


Рис. 5. Схема решения задачи
Fig. 5. Problem-solving scheme

Общая схема решения задачи приведена на рис. 5. Операция o_5 осуществляет подготовку данных сценариев развития ТЭК Вьетнама. Она формирует критерии оценки функционирования комплекса, список мероприятий по его развитию, множество организационно-функциональных ограничений, продолжительность моделирования и др.

Затем на основе сформированных данных операция o_1 выбирает сценарий и строит соответствующий ему вариант модели развития ТЭК. Операция o_2 производит декомпозицию модели на подмодели с учетом логических и балансовых ограничений. Подмодели параллельно исследуются с помощью экземпляров операции o_3 . При этом решается следующая задача линейного программирования: $f(S) = CX + P(R - Y) \rightarrow \min$, $S = \langle A, X, Y, C, P, D, R, U, W \rangle$, $AX - Y \geq 0$, $X \leq D$, $Y \leq R$, где S – обобщенное состояние ТЭК, C – вектор удельных затрат по технологическим способам работы действующих, реконструируемых или модернизируемых, а также вновь сооружаемых объектов ТЭК, A – матрица технологических коэффициентов производства (добычи, переработки, преобразования) и транспорта ресурсов, $a_{ij} = q_{ij}(U, W)$, D – вектор, определяющий технически возможные интенсивности применения технологических способов работы объектов (производства и транспорта ресурсов), $d_i = h_i(U)$, R – вектор, задающий требуемые объемы потребления ресурсов, P – вектор удельных ущербов из-за дефицита ресурсов, U – вектор мероприятий по развитию ТЭК, W – вектор возмущений. Выражения CX и $P(R - Y)$ в целевой функции соответственно отражают издержки, связанные с работой комплекса, и ущерб от недопоставки ресурсов потребителям вследствие воздействия возмущения. В качестве возмущений рассматриваются такие чрезвычайные ситуации, как крупномасштабные технологические аварии, стихийные бедствия и террористические акты. Операция o_4 агрегирует результаты моделирования, выполняет их многокритериальный анализ и формирует множество рациональных путей развития ТЭК для каждого сценария. Модули $m_1 - m_5$ реализуют операции $o_1 - o_5$. Их частая модификация обусловлена развитием алгоритмических знаний РППП. В частности, она обуславливается применением новых критериев оценки работы комплекса, расширением списка мероприятий по его развитию и использованием дополнительных организационно-функциональных ограничений. Кроме того, развитие программно-аппаратного обеспечения среды зачастую приводит к необходимости перекомпиляции, сборки и тестирования модулей с учетом этих

обновлений. В связи с этим рассматриваются две базовые версии v_1 [9] и v_2 [10] пакета для решения задачи поиска направлений устойчивого развития ТЭК Вьетнама. Вторая версия разработана с использованием нового менеджера ПО ОТ.

Вычислительные ресурсы среды. Решение задач с помощью обеих вышеупомянутых версий пакета выполнялось в интегрированной вычислительной среде, включающей следующие ресурсы: облачные виртуализированные ресурсы под управлением OpenStack – 5 выделенных узлов (2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 64 GB RAM) высокопроизводительного кластера ЦКП «Иркутский суперкомпьютерный центр СО РАН» (ИСКЦ); невыделенные ресурсы – узлы (2x18 cores CPU Intel Xeon X5670, 2.1 GHz, 128 GB RAM) высокопроизводительного кластера ЦКП ИСКЦ; грид-ресурсы – удаленные ПК. Выделенные ресурсы предназначены только для решения рассматриваемой задачи. Невыделенные ресурсы являются общими для всех пользователей ЦКП. Они используются в соответствии с установленными для них административными политиками и квотами. Пользователи, которым уже выделены ресурсы ЦКП, могут использовать только те невыделенные ресурсы, на которых есть свободные слоты в расписании обслуживания СУПЗ. В процессе решения задачи в интегрированной вычислительной среде ее грид-ресурсы могут использоваться их владельцами.

Подготовка эксперимента. Модификация ПО включает следующие этапы:

- 1) модификацию исходного кода прикладного и системного ПО, включая поиск его нужной версии в репозиториях на сервере GitLab, клонирование репозитория на сервер ОТ или сервер разработчика, запуск среды программирования, редактирование кода и сохранение модифицированного кода на сервер GitLab;
- 2) сборку ПО в соответствии со сценариями, предопределенными в конструкторе концептуальной модели для внешних систем непрерывной интеграции;
- 3) модификацию тестовых данных, включая их поиск на сервере GitLab или сервере тестирования, запуск системы обработки данных и их редактирование;
- 4) модификацию сценариев непрерывной интеграции, доставки и развертывания ПО с помощью конструктора концептуальной модели;
- 5) тестирование ПО в соответствии с предопределенными сценариями;
- 6) построение установочных пакетов ПО в соответствии со сценариями его непрерывной интеграции, доставки и развертывания;
- 7) модификацию концептуальной модели (параметров, операций, модулей, продуктов, схем решения задач и связей между ними) среды, трансляцию полученной модификации в XML и проверку его полноты и корректности;
- 8) размещение ПО на ресурсах с помощью установочных пакетов;
- 9) тестирование ПО на ресурсах, включая оценку их производительности;
- 10) тестирование схем решения задач РППП по предопределенным сценариям.

Как правило, данные этапы многократно повторяются в процессе модификации ПО. При разработке версии v_1 все они выполнялись в неавтоматизированном режиме. Оценки средних временных затрат в минутах на модификацию ПО пакета показаны на рис. 6. Они включают оценки затрат общего времени (рис. 6 а и 6 в) и разработчика (рис. 6 б и 6 г) на этапах модификации одного модуля с использованием менеджера ПО (версия v_2) и без него (версия v_1). Рис. 6 а и 6 в отражают временные затраты на этапах, связанных с модификацией алгоритмических знаний. Их выполнение базируется на совместном использовании средств ОТ и внешних систем непрерывной интеграции. Остальные оценки, относящиеся к модификации пакета только средствами ОТ, показаны на рис. 6 б и 6 г.

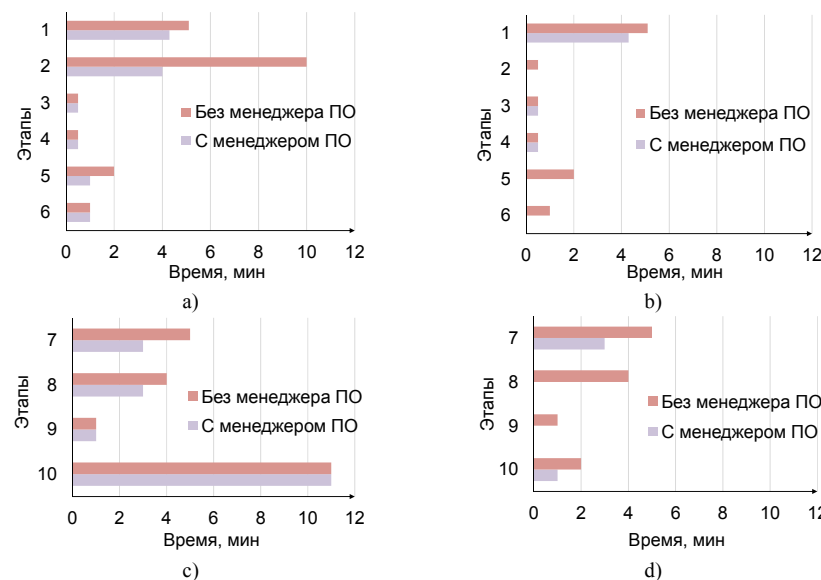


Рис. 6. Оценки затрат общего времени (а, б) и разработчика (с, д) на этапах модификации ПО
Fig. 6. Evaluations of the total time overheads (a, b) and developer (c, d) at the stages of software modification

Результаты на рис. 6, показывают, что применение менеджера ПО обеспечивает существенное сокращение как временных затрат разработчика, так и общих временных затрат на подготовку эксперимента. Они обусловлены следующими факторами: сокращением времени поиска исходного кода ПО в репозиториях; автоматизацией сборки, тестирования, построения установочных пакетов, размещения и тестирования ПО на ресурсах, тестирования схем решения задач; сокращением времени сборки, тестирования и построения установочных пакетов за счет переноса выполнения этих этапов на более производительные ресурсы; повышением надежности вычислений в процессе решения задачи путем предварительного тестирования ПО на разнородных ресурсах; автоматизацией процесса модификации концептуальной модели среды при изменении ПО в репозиториях.

На рис. 7 показаны оценки временных затрат разработчика ПО и общих временных затрат в минутах на подготовку и проведение экспериментов для версии v_2 с менеджером ПО и без него. Здесь 11-й этап – это проведение эксперимента. Оценки времени даны нарастающим итогом в соответствии с порядком выполнения этапов. Очевидно, что применение менеджера ПО обеспечивает сокращение времени подготовки и проведения экспериментов.

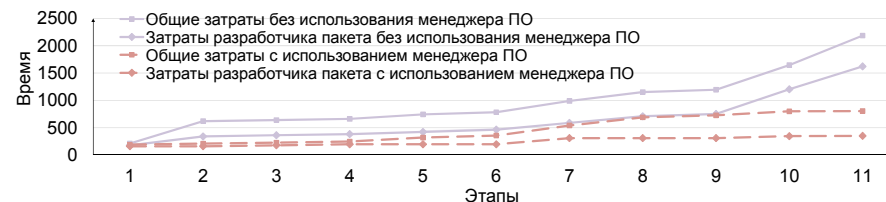


Рис. 7. Оценки временных затрат
Fig. 7. Time overheads evaluations

Вычислительный эксперимент. Мы сравниваем показатели выполнения модулей двух версий v_1 и v_2 при решении типовых задач с одинаковой вычислительной сложностью. При этом использованы данные одной и той же структуры. Они получены на основе статистической информации о работе ТЭК Вьетнама в разные периоды. В табл. 2 приведены следующие показатели решения задачи для одного сценария развития ТЭК: n_s – число запусков модуля, t_{avg} – среднее время выполнения модуля в секундах, n_f – число отказов модуля, n_r – число рестартов модуля, k – коэффициент полезного использования компонентов узла.

Табл. 2. Показатели выполнения модулей

Table 2. Module execution parameters

Модуль / Module	n_s		t_{avg}			n_f		n_r		k
	v_1	v_2	v_1	v_2	v_1	v_2	v_1	v_2	v_1	v_2
m_1	1	1	10.22	10.17	0	0	0	0	90.01	90.70
m_2	1	1	172.64	166.42	0	0	0	0	94.35	94.74
m_3	531442	531497	10.17	9.74	1371	0	1678	0	88.27	98.46
m_4	1	1	130.04	129.23	0	0	0	0	30.00	30.02
m_5	1	1	11.62	10.55	0	0	0	0	27.27	30.63

Время параллельной обработки подмоделей развития ТЭК экземплярами модуля m_3 составляет основную долю общего времени решения задачи. Результаты, приведенные в табл. 2, показывают, что число отказов экземпляров модуля m_3 версии v_2 , сократилось. Соответственно, число их рестартов уменьшилось. Оптимизация работы модуля m_3 версии v_2 и устранение отказов в его работе привело к улучшению коэффициента полезного использования ресурсов данным модулем на 10.19%. Коэффициенты полезного использования ресурсов при выполнении остальных модулей версии v_2 так же улучшены.

На рис. 8 приведено время t_2 и t_3 вычислений в минутах при использовании модулей версий v_1 и v_2 . В обоих случаях совместно использовались выделенные и невыделенные ресурсы. Распределение заданий выполнялось под управлением MAC.

Дополнительно приведено оценочное время t_1 решения задачи только на выделенных ресурсах при использовании модулей версии v_1 . В этом случае распределение заданий осуществлялось под управлением OpenStack. В случае версии v_1 использование дополнительных невыделенных ресурсов позволило сократить время решения задачи на 18% по сравнению с оценочным временем ее решения только на невыделенных ресурсах. Применение версии v_2 обеспечило сокращение этого времени еще на 5%.

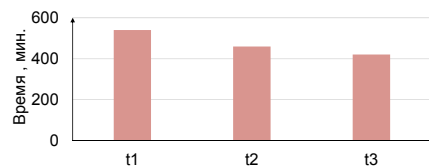


Рис. 8. Время решения задачи

Fig. 8. Problem-solving time

7. Заключение

ОТ предоставляет возможность организовывать унифицированную среду, интегрирующую облачные платформы, грид-системы и суперкомпьютерные ЦКП. Он включает средства для

разработки и использования РППП с использованием современных методов и средств параллельных и распределенных вычислений, а также их виртуализации.

В статье представлены новые возможности инструментального комплекса ОТ, который является основой интегрированной вычислительной среды для разработки и применения РППП. Новая подсистема непрерывной интеграции, доставки и развертывания прикладного и системного программного обеспечения нацелена на сокращение временных затрат на подготовку эксперимента, повышение надежности распределенных вычислений и ускорение процесса решения задачи с учетом неопределенности.

Эти цели достигаются путем смягчения неопределенности в распределении заданий по ресурсам за счет извлечения дополнительных необходимых знаний о времени выполнения модуля в разных узлах гетерогенной среды, классификации заданий в соответствии с их характеристиками, использования знаний администраторов среды о соответствии ресурсов классам заданий, а также сведений, получаемых в процессе непрерывной интеграции, доставки и развертывания программного обеспечения. Преимущества ее применения показаны на примере разработки двух версий РППП для решения важных практических задачи исследования направлений устойчивого развития ТЭК Вьетнама.

Список литературы / References

- [1]. Il'in V. Artificial Intelligence Problems in Mathematical Modeling. Communications in Computer and Information Science, vol. 1129, 2019, pp. 505-516.
- [2]. Wang L., Jie W., Chen J. Grid computing: infrastructure, service, and applications. CRC Press, 2018, 528 p.
- [3]. Varshney S., Sandhu R., Gupta P.K. QoS Based Resource Provisioning in Cloud Computing Environment: A Technical Survey. Communications in Computer and Information Science, vol. 1046, 2019, pp. 711-723.
- [4]. Б.М. Шабанов, О.И. Самоваров. Принципы построения межведомственного центра коллективного пользования общего назначения в модели программно-определяемого ЦОД. Труды ИСП РАН, том 30, вып. 6, 2018 г., стр. 7-24. DOI: 10.15514/ISPRAS-2018-30(6)-1 / B.M. Shabanov and O.I. Samovarov. Building the Software-Defined Data Center. Programming and Computer Software, vol. 45, no. 8, 2019, pp. 458-466.
- [5]. Mateescu G., Gentzsch W., Ribben C.J. Hybrid computing – where HPC meets grid and cloud computing // Future Generation Computer Systems, 2011, vol. 27, no. 5, pp. 440-453. DOI: 10.1016/j.future.2010.11.003.
- [6]. Feoktistov A., Gorsky S., Sidorov I. et al. Orlando Tools: Energy Research Application Development through Convergence of Grid and Cloud Computing. Communications in Computer and Information Science, vol. 965, 2019, pp. 289-300.
- [7]. Feoktistov A., Kostromin R., Sidorov I., Gorsky S. Development of Distributed Subject-Oriented Applications for Cloud Computing through the Integration of Conceptual and Modular Programming // In Proc. of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, 2018, pp. 256-261.
- [8]. Yu J., Buyya R. A taxonomy of workflow management systems for grid computing. Journal of Grid Computing, vol. 3, no. 3-4, 2005, pp. 171-200.
- [9]. Feoktistov A., Sidorov I., Tchernykh A. et al. Multi-Agent Approach for Dynamic Elasticity of Virtual Machines Provisioning in Heterogeneous Distributed Computing Environment In Proc. of the International Conference on High Performance Computing and Simulation (HPCS-2018), 2018, pp. 909-916.
- [10]. Bychkov I., Oparin G., Feoktistov A. et al. Subject-oriented computing environment for solving large-scale problems of energy security research. Journal of Physics: Conference Series, vol. 1368, 2019, pp. 052030-1-052030-12.
- [11]. Burri A., Dedner A., Kloforn R., Ohlberger M. An efficient implementation of an adaptive and parallel grid in DUNE. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol. 91, 2006, pp. 67-82.

- [12]. Radchenko G., Hudyakova E. A service-oriented approach of integration of computer-aided engineering systems in distributed computing environments. In Proc. of the UNICORE Summit, 2012, pp. 57-66.
- [13]. Shamakina A. Brokering service for supporting problem-oriented grid environments. In Proc. of the UNICORE Summit, 2012, pp. 67-75.
- [14]. Bungartz H.J., Neumann P., Nagel W.E. Software for Exascale Computing-SPPEXA 2013-2015. Lecture Notes in Computational Science and Engineering, vol. 113, 2016, 565 p.
- [15]. Afgan E., Baker D., Batut B. et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research*, vol. 46, no. W1, 2018, pp. W537-W544.
- [16]. Ananthkrishnan R., Blaiszik B., Chard K., Chard R. Globus platform services for data publication. In Proc. of the Practice and Experience on Advanced Research Computing, 2018, pp. 1-7.
- [17]. Sukhoroslov O. Supporting Efficient Execution of Workflows on Everest Platform. *Communications in Computer and Information*, vol. 1129, 2019, pp. 713-724.
- [18]. Gavvala S.K., Chandrasheka J., Gangadharan G.R., Buyya R. QoS-aware cloud service composition using eagle strategy. *Future Generation Computer Systems*, vol. 90, 2019, pp. 273-290.
- [19]. Deelman E., Peterka T., Altintas I., Carothers C.D. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, 2018, pp. 159-175.
- [20]. Wangsom P., Lavangananda K., Bouvry P. Multi-Objective Scientific-Workflow Scheduling with Data Movement Awareness in Cloud. *IEEE Access*, vol. 7, 2019, pp. 177063-177081.
- [21]. Feoktistov A., Gorsky S., Sidorov I., Tchernykh A. Continuous Integration in Distributed Applied Software Packages. In Proc. of the 42st International Convention on Information and Communication Technology, Electronics and Microelectronics, 2019, pp. 1775-1780.
- [22]. Gruver G. Start and Scaling Devops in the Enterprise. BookBaby, 2016. 100 p.
- [23]. Talia D. Workflow Systems for Science: Concepts and Tools. *ISRN Software Engineering*, 2013, vol. 2013, Article ID 404525.
- [24]. Deelman E., Vahi K., Juve G et al. Pegasus, a workflow management system for science automation // *Future Generation Computer Systems*, vol. 46, 2015, pp. 17-35.
- [25]. Bumgardner V.K. *OpenStack in Action*. Manning Publications, 2016. 384 p.
- [26]. Hiraes-Carbajal A., González-García J. L., Tchernykh A. Workload Generation for Trace Based Grid Simulations. In Proc. of the 1st International Supercomputer Conference in Mexico (ISUM-2010), 2010, pp. 1-10.
- [27]. Bychkov I., Oparin G., Tchernykh A. et al. Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management. *Procedia Computer Science*, vol. 103, 2017, pp. 162-167.
- [28]. Соколинский Л.Б., Шамакина А.В. Методы управления ресурсами в проблемно-ориентированных вычислительных средах. *Программирование*, том 42, no. 1, 2016 г., стр. 26-38 / Sokolinsky L.B., Shamakina A.V. Methods of resource management in problem-oriented computing environment. *Programming and Computer Software*, vol. 42, no. 1, 2016, pp. 17-26.
- [29]. Ramirez-Velarde R., Tchernykh A., Barba-Jimenez C. et al. Adaptive Resource Allocation with Job Runtime Uncertainty *Journal of Grid Computing*, vol. 15, no. 4, 2017, pp. 415-434.
- [30]. Tchernykh A., Schwiigelshohn U., Talbi E.-G., Babenko M. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*, vol. 36, 2019, 100581. DOI: 10.1016/j.jocs.2016.11.011.
- [31]. Kalyaev A.I., Kalyaev I.A. Method of multiagent scheduling of resources in cloud computing environments. *Journal of Computer and Systems Sciences International*, vol. 55, no. 2, 2016, pp. 211-221.
- [32]. Bychkov I.V., Oparin G.A., Feoktistov A.G. et al. Multiagent control of computational systems on the basis of meta-monitoring and imitational simulation. *Optoelectronics, Instrumentation and Data Processing*, vol. 52, no. 2, 2016, pp. 107-112.
- [33]. Java Agent Development Framework, available at: <https://jade.tilab.com>, accessed 30.05.2020.
- [34]. Herrera J., Huedo E., Montero R., Llorente I. Porting of Scientific Applications to Grid Computing on GridWay. *Scientific Programming*, vol. 13, no. 4, 2005, pp. 317-331.
- [35]. Tannenbaum T., Wright D., Miller K., Livny M. Condor – A Distributed Job Scheduler. In *Beowulf Cluster Computing with Linux*. The MIT Press, 2002, pp. 307-350.
- [36]. Feoktistov A., Tchernykh A., Kostromin R., Gorsky S. Knowledge Elicitation in Multi-Agent System for Distributed Computing Management. In Proc. of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, 2017, pp. 1350-1355.

- [37]. Feoktistov A., Kostromin R., Sidorov I. et al. Multi-Agent Algorithm for Re-Allocating Grid-Resources and Improving Fault-Tolerance of Problem-Solving Processes. *Procedia Computer Science*, vol. 150, 2019, pp. 171-178.
- [38]. Vickrey W. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, vol. 16, no. 1, 1961, pp. 8-37.

Информация об авторах / Information about authors

Андрей Николаевич ЧЕРНЫХ получил степень кандидата наук в Институте точной механики и вычислительной техники РАН. Он является профессором CICESE. В научном плане его интересуют многоцелевая оптимизация распределения ресурсов в облачной среде, проблемы безопасности, планирования, эвристики и метаэвристики, интернет вещей и т.д.

Andrei Nikolaevitch TCHERNYKH received his PhD degree at the Institute of Precision Mechanics and Computer Engineering of the Russian Academy of Sciences. He is holding a full professor position in computer science at CICESE. He is interesting in grid and cloud research addressing multiobjective resource optimization, both, theoretical and experimental, security, uncertainty, scheduling, heuristics and meta-heuristics, adaptive resource allocation, and Internet of Things.

Игорь Вячеславович БЫЧКОВ – академик РАН, доктор технических наук, профессор, директор Института динамики систем и теории управления им. В.М. Матросова СО РАН (ИДСТУ СО РАН). Области исследования: искусственный интеллект, геоинформационные системы, веб-технологии, математическое моделирование и облачные вычисления.

Igor Vyacheslavovich BYCHKOV – Academician of RAS, Ph.D., Professor, Director of the Matrosov Institute for System Dynamics and Control Theory of SB RAS (ISDCT SB RAS). Fields of research: artificial intelligence, geoinformation systems, web-technologies, mathematical modeling, and cloud computing.

Александр Геннадьевич ФЕОКТИСТОВ – кандидат технических наук, доцент, заведующий лабораторией параллельных и распределенных вычислительных систем ИДСТУ СО РАН. Области исследования: распределенные пакеты программ, мультиагентные технологии.

Alexander Gennadevich FEOKTISTOV – Ph.D., Associate Professor, Head of the Laboratory of Parallel and Distributed Computing Systems of ISDCT SB RAS. Fields of research: distributed software packages and multi-agent technologies.

Сергей Алексеевич ГОРСКИЙ – кандидат технических наук, научный сотрудник ИДСТУ СО РАН. Сфера научных интересов: параллельные и распределенные вычисления, математическое моделирование и сервис-ориентированное программирование.

Sergei Alexeevich GORSKY – Ph.D., Research Officer of ISDCT SB RAS. Research interests: parallel and distributed computing, mathematical modeling, and service-oriented programming.

Иван Александрович СИДОРОВ – кандидат технических наук, научный сотрудник ИДСТУ СО РАН. Его научные интересы включают системное администрирование, параллельные и распределенные вычисления, агенты и мониторинг.

Ivan Alexandrovich SIDOROV – Ph.D., Research Officer of ISDCT SB RAS. His research interests include system administration, parallel and distributed computing, agents, and monitoring.

Роман Олегович КОСТРОМИН – младший научный сотрудник ИДСТУ СО РАН. Область исследования: мультиагентные средства управления распределенными вычислениями.

Roman Olegovich KOSTROMIN – Junior Researcher ISDCT SB RAS. Field of research: multi-agent tools for distributed computing management.

Алексей Владимирович ЕДЕЛЕВ – кандидат технических наук, старший научный сотрудник Института систем энергетики им. Л.А. Мелентьева СО РАН. Его научные интересы включают моделирование развития энергетики и системы поддержки принятия решений.

Aleksey Vladimirovich EDELEV – Ph.D. of Engineering Sciences, Senior Researcher of the Melentiev Energy Systems Institute of SB RAS. His research interests include energy development modeling and decision-making support systems.

Валерий Иванович ЗОРКАЛЬЦЕВ – доктор технических наук, профессор, главный научный сотрудник Лимнологического института СО РАН. Области исследования: теория оптимизации, математическая экономика и методика моделирования.

Valeriy Ivanovich ZORKALTSEV – Doctor of Technical Sciences, Professor, Chief Researcher of the Limnological Institute of SB RAS. Field of research: optimization theory, mathematical economics, and simulation technique.

Арутюн Ишханович АВЕТИСЯН – академик РАН, доктор физико-математических наук, профессор РАН, директор ИСП РАН, заведующий кафедрами системного программирования ВМК МГУ, МФТИ и факультета компьютерных наук ВШЭ. Сфера научных интересов: анализ и трансформация программ, безопасность программного обеспечения, технологии параллельных и распределенных вычислений.

Arutyun I. AVETISYAN – Academician of RAS, Doctor of Physics and Mathematics, Professor of RAS, Director of the ISP RAS, Head of the Departments of System Programming, the faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University, Moscow Institute of Physics and Technology, and the Faculty of Computer Science at the Higher School of Economics. Research interests: program analysis and transformation, software security, parallel and distributed computing.