

DOI: 10.15514/ISPRAS-2024-36(6)-11



Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs

¹ А.Г. Феоктистов, ORCID: 0000-0002-9127-6162 <agf65@icc.ru>

¹ М.Л. Воскобойников, ORCID: 0000-0003-3034-4907 <mikev1988@mail.ru>

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

¹ Институт динамики систем и теории управления им. В.М. Матросова СО РАН,
664033, Иркутск, ул. Лермонтова, 134, а/я 29, Россия.

² Центр научных исследований и высшего образования Энсенада, Нижняя Калифорния,
Мексика, 22860, Энсенада, шоссе Энсенада-Тихуана, 3918, почтовый ящик 360.

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

Аннотация. Рассматриваются актуальные аспекты организации сервис-ориентированных вычислений в вычислительной среде с гетерогенными ресурсами. Обсуждается развитие технологий разработки и применения сервис-ориентированных научных приложений, в которых схемы решения задач формируются в виде рабочих процессов. Приводятся существующие стандарты описания рабочих процессов. Предлагается новый инструментальный комплекс для создания сервис-ориентированных научных приложений, развивающий и дополняющий возможности систем подобного назначения.

Ключевые слова: вычислительная среда; гетерогенные ресурсы; сервис-ориентированные научные приложения; рабочие процессы.

Для цитирования: Феоктистов А.Г., Воскобойников М.Л., Черных А.Н. Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs. Труды ИСП РАН, том 36, вып. 6, 2024 г., стр. 195–214. DOI: 10.15514/ISPRAS-2024-36(6)-11

Благодарности: Исследование проведено в рамках проекта № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9) при поддержке Министерства науки и высшего образования Российской Федерации.

Development and Use of Service-Oriented Scientific Applications in the FDE-SWFs Toolkit

¹ A.G. Feoktistov ORCID: 0000-0002-9127-6162 <agf@icc.ru>

¹ M.L. Voskoboinikov ORCID: 0000-0003-3034-4907 <mikev1988@mail.ru>

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

¹ Matrosov Institute for System Dynamics and Control Theory
of the Siberian Branch of the Russian Academy of Sciences,
134, Lermontov st., Irkutsk, postbox 292, 664033, Russia.

² Centro de investigación Científica y de educación Superior de Ensenada, Baja California,
22860, México, Ensenada, Ensenada-Tijuana Highway, 3918, postbox 360.

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

Abstract. The automation of workflow-based computing for solving large resource-intensive problems has undoubtedly had an impact on increasing the productivity of scientific research. In recent years, workflows have become the basis for abstractions covering data processing and high-performance computing using distributed applications. Workflow management systems are powerful tools for the collaborative development and use of distributed scientific applications. Nowadays, as part of the development of such systems, particular attention is currently being paid to supporting service-oriented scientific applications. Within this field of research, there is a large spectrum of problems related to the support of modular scientific applications, the standardization of their components and interfaces, the use of heterogeneous information and computing resources, and organization interdisciplinary research. Unfortunately, the solution to the above listed problems has not been fully implemented in known workflow management systems that support the development and use of service-oriented scientific applications. In this context, the paper discusses relevant aspects of organizing service-oriented computing in a computing environment with heterogeneous resources. The development of technologies for the development and use of service-oriented scientific applications, in which problem-solving schemes are formed in the form of workflows, is discussed. Existing standards for describing workflows are represented. A new framework for creating service-oriented scientific applications is proposed. It extends and complements the capabilities of systems for such purposes.

Keywords: heterogeneous distributed computing environment; heterogeneous resources; service-oriented scientific applications; workflows.

For citation: Feoktistov A.G., Voskoboinikov M.L., Tchernykh A.N. Development and use of service-oriented scientific applications in the FDE-SWFs toolkit. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 6, 2024. pp. 195-214 (in Russian). DOI: 10.15514/ISPRAS-2024-6(6)-11.

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».

1. Введение

Автоматизация расчетов на основе рабочих процессов (РП) в рамках решения больших ресурсоемких задач оказывает несомненное влияние на повышение продуктивности научных исследований. За последние годы РП стали основой абстракций, охватывающих обработку данных и проведение высокопроизводительных вычислений с помощью распределенных приложений. При этом использование специализированной системы управления рабочими процессами (СУРП, англ., Workflow Management System – WMS) зачастую освобождают конечных пользователей от необходимости вникать в детали выполнения РП, а также управления ими в вычислительной среде с гетерогенными ресурсами (ВСТР).

СУРП, такие как UNICORE [1], HTCondor [2], Pegasus [3] и другие программные комплексы [4-5], являются мощными инструментами командной разработки и применения

распределенных научных приложений. Они предназначены для интеграции программного обеспечения, планирования схем решения с его помощью научных и прикладных задач, назначения ресурсов для выполнения вычислений, запуска вычислительных процессов и управления ими, обработки данных и реализации других системных операций в распределенной программно-аппаратной среде.

В рамках развития СУРП особое внимание в настоящее время уделяется поддержке сервис-ориентированных научных приложений (СОНП) [5]. Сервис-ориентированное программирование направлено на разработку программных систем, поддерживающих взаимодействие приложений и сервисов различных типов на основе обмена сообщениями с использованием опубликованных и обнаруживаемых интерфейсов [6]. Зачастую сервисы предоставляют хороший способ реализовать дискретные операции приложений для вычислений и обработки данных в рамках бизнес-процессов в различных предметных областях исследований. Поэтому развитие сервис-ориентированных вычислений во многом обусловлено возможностью эффективного решения целого ряда проблем, связанных с поддержкой модульных научных приложений, стандартизации их компонентов и интерфейсов, использования разнородных информационных и вычислительных ресурсов, а также организации междисциплинарных исследований.

К сожалению, решение вышеперечисленных проблем в известных СУРП, поддерживающих СОНП (см., например, [7-9]), осуществлено не в полной мере. В этой связи в статье рассмотрены важные аспекты разработка и применение СОНП, а также представлен новый инструментальный комплекс, компоненты которого развивают и дополняют функционал известных в этом направлении СУРП.

2. Развитие сервис-ориентированных приложений

Парадигма разработки и применения СОНП представляет собой логическую эволюцию от объектно-ориентированных систем к системам сервисов. Как и в объектно-ориентированных системах, некоторыми фундаментальными концепциями веб-служб являются инкапсуляция, передача сообщений и динамическое связывание. Однако парадигма, основанная на сервисах, выходит за рамки сигнатур методов. Информация о функциях сервиса, его местоположении, способах доступа и др. также могут быть представлены в интерфейсе сервиса. Разработку СОНП также можно рассматривать как эволюцию модульного подхода к программированию, так как веб-сервисы представляют собой легкие, слабосвязанные, независимые от платформы и языка компоненты.

В настоящее время в основе рассматриваемой парадигмы доминирует сервис-ориентированная архитектура (COA, англ., Service-Oriented Architecture – SOA). COA основывается на использовании множества независимых веб-сервисов, выполняющих предопределенные операции, связанных с выполнением системных или прикладных приложений. Под веб-сервисом (англ., Web service) понимается программная система со стандартизированными интерфейсами, идентифицируемая уникальным веб-адресом (URL-адресом) [10]. При этом веб-сервисы не обладают знаниями о выполняемых приложениях, а приложения не нуждаются в информации о способах их выполнения веб-сервисами. Веб-технологии на основе COA активно поддерживаются крупными компаниями-разработчиками, что обеспечивает их широкое распространение и использование.

Применительно к вычислительной среде COA обуславливает ряд следующих важных преимуществ в процессах ее организации и применении [11]:

- многократное использование компонентов среды для построения сложных распределенных программных комплексов;
- модульный подход к разработке программного обеспечения (ПО);
- поддержку сетевого доступа к компонентам среды их разработчикам и пользователям, а также их взаимодействия между собой;

- обеспечение открытости среды за счет использования стандартов протоколов передачи данных и представления сервисных операций над этими данными;
- кроссплатформенность, позволяющую смягчить зависимость вычислительного процесса от используемых программно-аппаратных платформ и языков программирования;
- возможность безболезненной интеграции ПО разных разработчиков.

ПО, разработанное на основе COA, как правило, реализуются в виде набора веб-служб, взаимодействующих по протоколу Simple Object Access Protocol (SOAP). Веб-служба является единицей модульности в рамках COA ПО. В то же время COA может быть реализована с использованием широкого спектра дополнительных технологий, таких как REpresentational State Transfer (REST), Remote Procedure Call (RPC), Distributed Component Object Model (DCOM) и Common Object Request Broker Architecture (CORBA).

Основными форматами представления структурированных данных являются eXtensible Markup Language (XML) и JavaScript Object Notation (JSON), для которых поддерживается проверка корректности данных с помощью XML Schema и JSON Schema соответственно. Неструктурированные данные, как правило, представляются в виде текстовых файлов или файлов других форматов. Передача данных между веб-сервисами осуществляется посредством их включения в тело сообщения (в случае небольшого размера передаваемой информации) или путем передачи адреса источника данных (англ., Uniform Resource Locator – URL), откуда их можно извлекать (в случае большого размера данных).

Существуют разные способы описания сетевых служб или веб-сервисов [12]. В их числе можно выделить Web Service Description Language (WSDL) для описания веб-сервисов на основе SOAP и Web Application Description Language (WADL) для описания веб-приложений на основе HyperText Transfer Protocol (HTTP), в том числе веб-сервисов в стиле REST. В обоих случаях в качестве базового языка описания используется XML.

WSDL предназначен для описания веб-сервисов, доступа к ним и передаваемых между ними сообщений. Описание веб-сервиса на WSDL включает следующие основные разделы:

- определение типов данных, указывающих вид отправляемых и получаемых сервисом XML-сообщений, проверка которых осуществляется с помощью средств XML Schema;
- описание элементов данных – списка сообщений, используемых сервисом;
- задание абстрактных операций (портов) – списка методов, которые могут быть выполнены применительно к сообщениям;
- связывание сервисов – определение способов доставки сообщений;
- адрес вызова сервиса.

Последняя официальная спецификация языка WSDL Version 2.0 позволяет описывать, как вызовы различных специализированных веб-сервисов на основе SOAP, например, WPS-сервисов, так и сервисов на основе других протоколов, например, REST-сервисов.

Разработка и применение СОНП характеризуется рядом преимуществ по сравнению с другими типами приложений. Наличие в СОНП набора сервисов позволяет разработчикам приложения создавать, отлаживать, тестировать, разворачивать и модифицировать свои сервисы независимо от других разработчиков, что упрощает распределенную разработку приложения.

Каждый сервис может быть разработан и развернут на разных ресурсах с различными характеристиками их производительности, объемов оперативной и дисковой памяти, пропускной способностью интерконнекта и т. п. в Grid-системах, на ресурсах суперкомпьютеров или облачных платформ. С использованием контейнеризации сервисы можно запустить на нескольких параллельно работающих узлах без необходимости развертывания на новом узле всего приложения в целом.

Важным преимуществом СОНП является их отказоустойчивость. Отказ одного сервиса, как правило не приводит к отказу всего приложения в целом. При этом отказавший сервис может быть легко перезапущен или его операции могут быть выполнены другим сервисом при наличии вычислительной избыточности в СОНП.

В табл. 1 приведены сведения о разработках в области сервис-ориентированных вычислений. Рассматриваются следующие важные аспекты поддержки сервис-ориентированных вычислений на уровнях приложения и/или вычислительной среды в целом: разные типы сервисов, способы их спецификации, архитектура среды выполнения, сервис-ориентированные модели представления различных сущностей вычислительной среды, средства разработки, системы управления, СОНП и др.

Табл. 1. Разработки в области сервис-ориентированных вычислений.

Table 1. Developments in the field of service-oriented computing.

Источник / Source	Аспекты поддержки / Support aspects	Уровень поддержки / Support level
[13-17]	Grid и облачные вычисления, SOAP-сервисы, Grid-сервисы, SaaS, Globus Toolkit.	Среда
[9]	Облачные вычисления, SOAP-сервисы, СУПИ, WaaS Cloud Platform	Приложение
[8]	Облачные вычисления, SOAP-сервисы, СУПИ, HyperFlow	Приложение
[18]	Методика спецификации SOAP-сервисов, методика спецификации РП, WSDL, BPEL	Приложение
[19]	Микросервисы, композиции сервисов, синтез программ	Приложение
[20]	REST	Среда
[1]	Grid и облачные вычисления, Grid-сервисы, SOAP, REST, СУПИ, UNICORE	Среда
[7]	Облачные вычисления, SOAP, REST, СУПИ, Galaxy	Среда
[21]	Кооперативные вычисления, управление данными, микросервисы, iRODS	Среда, приложение
[22]	Облачные вычисления, SOAP, REST, MathCloud, Everest	Среда, приложение
[23]	Grid и облачные вычисления, SOAP-сервисы, Grid-сервисы, CAEBeans, испытательные стенды	Среда
[24, 25]	Облачные вычисления, РП, веб-сервисы, интеллектуализация управления вычислениями, IaaS, SaaS, PaaS, iPSE	Среда
[26]	СОНП	Приложение
[27]	HPC, Amazon Web Services, Google Compute Engine, OpenStack, Cloud Stack, IaaS, PaaS, SaaS	Среда
[28]	СОНП, обработка данных	Среда, приложение
[29]	СОНП, обработка данных	Среда, приложение
[30]	СОНП, обработка данных	Среда, приложение
[31]	Исследования в энергетике, СОНП, обработка данных	Среда
[32]	Геоинформатика, REST-сервисы, SOAP-сервисы, WPS-сервисы, композиции сервисов	Среда
[33]	REST-сервисы, SOAP-сервисы, микросервисы, композиции сервисов	Приложение
[34]	SOAP-сервисы, композиции сервисов, WSDL	Среда, приложение
[35]	Мультиагентное управление ресурсами, микросервисы, шаблоны сервисов	Среда

3. Стандарты представления РП

РП обеспечивают возможность проведения крупномасштабных научных экспериментов с использованием больших наборов данных. При этом процессы обработки данных распределяются на различные вычислительные ресурсы. РП могут включать операции

обнаружения и связывания ресурсов, а также сбора, обработки, анализа и визуализации данных. РП должны быть логичными, структурированными и надежными.

Операции РП выполняются в соответствии со схемой решения задачи в логической последовательности, определяемой его структурой. Применение стандартов для описания и выполнения РП позволяет распространять их среди научного сообщества и облегчает их многократное использование. РП могут быть размещены в общедоступных репозиториях.

Стандарты представления РП берут свое начало в области моделирования бизнес-процессов. Соответствующие решения были разработаны рядом коммерческих организаций, таких как IBM и Microsoft. Открытые стандарты разрабатываются независимыми консорциумами, в том числе The World Wide Web Consortium (W3C), Organization for the Advancement of Structured Information Standards (OASIS), Workflow Management Coalition (WFMC), Business Process Management Initiative (BPMI), United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) и Object Management Group (OMG) [36]. Некоторые консорциумы концентрируют усилия на разработке комплектов дополнительных стандартов, тогда как другие разрабатывают отдельные многоцелевые стандарты. Пока еще нет единого мнения относительно того, какие стандарты наиболее подходят для СОНП, а также не существует установленной структуры стандартов для СОА.

СУРП часто характеризуются описанием процессов с точки зрения потока данных, а не ориентации потока управления бизнес-процессом. В рамках ряда проектов проводятся исследования, сравнивающие применимость различных стандартов описания РП. Важным направлением является разработка стандартов, позволяющих учитывать требования к вычислениям и передаче данных для больших наборов данных, а также обеспечивающих разделение абстрактного уровня представления РП и уровня его выполнения на конкретных программно-аппаратных ресурсах. В целом успешная реализация РП зависит от использования системы стандартов, каждый из которых обеспечивает эффективное планирование и выполнение вычислительных операций и операций обработки данных.

Различными коммерческими организациями и консорциумами предложен ряд стандартов описания РП [37]. В том числе разработаны языки XML Process Definition Language (XPDL), XLANG, Web Services Flow Language (WSFL), Business Process Modeling Language (BPML), Business Process Specification Schema (BPSS), Web Services Conversation Language (WSCL), Web Services Choreography Interface (WSCI), Yet Another Workflow Language (YAWL), Business Process Execution Language for Web Services (BPEL4WS или BPEL) 1.0, BPEL4WS 1.1, Web Services Choreography Description Language (WS-CDL) и Web Services Business Process Execution Language (WS-BPEL или BPEL) 2.0.

XPDL, разработанный Workflow Management Coalition (WFMC), предназначен для обмена определениями процессов между различными информационными системами, как в графическом, так и в семантическом виде. XPDL неоднократно пересматривался. Последняя ревизия состоялась в 2012 г.

XLANG от Microsoft является расширением WSDL. Его основное назначение заключается в определении бизнес-процессов и организации обмена сообщениями между веб-сервисами.

WSFL, разработанный компанией IBM, представляет собой XML-язык, описывающий бизнес-процесс в виде композиции веб-сервисов, в которой описывается последовательность вызовов операций сервисов. Порядок выполнения операций определяется на основе потоков управления и данных между сервисами. Бизнес-процесс определяет операции по получению, обработке и отмену данными в заданной последовательности.

BPML, предложенный BPMI, представляет собой язык описания бизнес-процессов на основе XML, представленный BPMI. Он предоставляет средства выполнения последовательных и параллельных операций, поддерживает ветвления и циклы, обеспечивает стандартные функции вызова сервисов, отправки и получения сообщений, позволяет разработчику РП

планировать выполнение задач в соответствии с заданным расписанием. В BPML предусмотрено управление РП с длительным сроком из выполнения.

BPSS представляет собой стандартную структуру, описывающую процесс обмена информацией. BPSS основана на метамодели UN/CEFACT. Эта схема позволяет предприятиям определять бизнес-транзакции и организовывать их сотрудничество между партнерами для электронного обмена документами и сигналами в коммерческих целях. BPSS входит в состав инструментария Electronic Business using XML (ebXML) от OASIS и UN/CEFACT.

WSCL от Hewlett-Packard предназначен для определения диалогов бизнес-уровня в виде общедоступных процессов, поддерживаемых веб-сервисами. WSCL определяет последовательность обмена XML-документами. Определения диалогов WSCL также являются XML-документами и поэтому могут интерпретироваться веб-сервисами.

YAWL представляет собой расширение XML и предназначен для формализованного описания бизнес-процессов. YAWL разработан в Техническом университете г. Эйндховен. Там также разработана специализированная программная платформа, поддерживающая текстовый и графический режимы построения бизнес-процессов и средства их выполнения. Исходный код программного обеспечения распространяется под лицензией GNU Lesser General Public License (LGPL).

WS-CDL создан W3C на основе XML для спецификации однорангового взаимодействия веб-сервисов на основе хореографии – упорядоченного обмена сообщениями между внешними сущностями. Спецификации сервисов определяют связи между гетерогенными вычислительными средами, используемыми для разработки и размещения веб-приложений. В целом обеспечение хореографии веб-сервисов позволяет организовать функционально совместимое одноранговое взаимодействие между любыми сервисами, независимо от поддерживающей платформы или модели программирования, используемой при их реализации.

К 2003 г. назрела необходимость перехода от разнородных стандартов отдельных консорциумов к некоторому единому стандарту. Усилия IBM, Microsoft, BEA, OASIS и др. консорциумов началась разработка и развитие BPEL. С развитием веб-сервисов произошло слияние WSFL и XLANG, что привело к появлению нового поколения языка спецификаций BPEL4WS 1.1. Язык BPEL4WS 1.1 позволил расширить модель взаимодействия веб-сервисов и сделать ее применимой для отображения бизнес-транзакций.

С появлением BPEL список стандартов, широко востребованных на практике, сокращается. Формально считается, что BPEL и XPDЛ обеспечивают оркестрацию взаимодействий между внутренними и внешними сущностями процессов, а WS-CDL и ebXML – хореографию. Однако функциональные возможности BPEL и XPDЛ позволяют описывать и хореографию. В целом BPEL 2.0 [38] определяет модель для описания поведения РП в терминах взаимодействий (совокупности сообщений) между процессами и их партнерами (внешними сервисами). Важными дополнительными преимуществами BPEL являются следующие:

- РП могут не только вызывать веб-сервисы, но и сами быть представленными в виде сервисов;
- широкий спектр элементов управления и работы с данными, включающий элементы определения сложные структуры данных и параллельных процессов их обработки, циклы, ветвления, подпроцессы, элементы реализации асинхронного взаимодействия веб-сервисов и др.;
- использование WSDL для описания интерфейсов веб-сервисов обеспечивает гибкую интеграцию с другими РП и веб-приложениями;
- детальное описание РП реализует оркестровку внутренних и внешних сущностей процесса, а спецификация процесса обмена сообщениями отражает хореографию внешних сущностей (вызываемых веб-сервисов).

В дополнение к перечисленным выше стандартам неформальная рабочая группа, объединяющая представителей различных организаций и частных лиц, которые заинтересованы в переносимости рабочих процессов, разрабатывает язык Common Workflow Language (CWL) [39]. Цель работы группы заключается в создании спецификаций, позволяющих представителям научного сообщества описывать мощные, простые в использовании, портативные и поддерживающие воспроизводимость рабочие процессы. При представлении ряда конструкций рабочих процессов (например, конвейеров) CWL использует возможности языков YAML Ain't Markup Language (YAML) и JSON. Для контейнеризации прикладного ПО в портативных средах выполнения применяется система Docker. Он предназначен для описания РП с интенсивным использованием данных областях исследований, таких как биоинформатика, медицина, химия, физика и астрономия. Версия 1.0 языка CWL выпущена 8 июля 2016 г.

Разработка и развитие стандартов описания веб-сервисов представлено в ретроспективе на рис. 1. Представленное на данном рисунке развитие языков спецификации веб-сервисов обобщает, уточняет и дополняет диаграммы из [18].

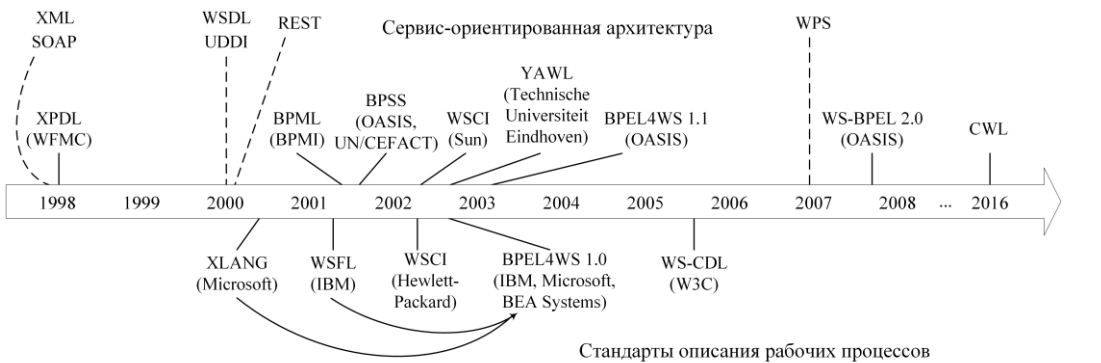


Рис. 1. Развитие языков спецификации рабочих процессов.
Fig. 1. Evolution of workflow specification languages.

4. Инструментальный комплекс

Существующие СУРП [40]) в том числе системы, обеспечивающие поддержку сервис-ориентированной парадигмы разработки прикладного ПО [7-9], широко используются для создания и применения научных приложений и управления ими в вычислительной среде. К сожалению, специализированные средства непрерывной интеграции разрабатываемого ПО, организации распределенных баз данных и работы с ними в памяти вычислительных узлов, тестирования и испытания компонентов СОНП в этих системах слабо развиты.

В этой связи в статье представлен новый инструментальный комплекс Framework for Development and Execution of Scientific WorkFlows (FDE-SWFs). FDE-SWFs относится к классу СУРП. Он базируется на подходе к разработке распределенных пакетов прикладных программ в ВСГР, поддерживаемом фреймворком Orlando Tools (OT) [41], и в тоже время существенно развивает и дополняет его функциональные возможности.

FDE-SWFs включает следующие основные подсистемы: пользовательский интерфейс, системные компоненты для конструирования и спецификации вычислительных моделей, а также управления вычислительными процессами и средой их выполнения. Для доступа пользователей к внешним информационно-вычислительным ресурсам и системам, с которыми им необходимо взаимодействовать при подготовке и проведении вычислительных экспериментов, он предоставляет набор специализированных API. База знаний инструментального комплекса содержит спецификацию вычислительных моделей

разрабатываемых приложений, РП и сведения о вычислительных ресурсах. Исходная информация и результаты выполнения РП хранятся в расчетных базах данных.

Вычислительная модель СОНП описывается в виде структуры

$$M = \langle Z, T, M, O, S, P, SW, N, R \rangle,$$

включающей следующие основные элементы:

- Z – множество значимых параметров предметной области приложения;
- T – множество их допустимых типов;
- M – множество программных модулей, представляющих алгоритмические знания предметной области;
- O – совокупность абстрактных вычислительных операций и операций обработки данных, отражающих семантику алгоритмических знаний в модели;
- S – множество сервисов, реализующих абстрактные операции;
- P – множество постановок задач, формулируемых в процедурной или непроедурной формах на вычислительной модели;
- SW – множество РП, построенных на основе процедурных или непроедурных постановок задач;
- N – множество ресурсов ВСГР, на которых размещаются и выполняются модули и сервисы приложения;
- R – связи между вышеперечисленными множествами объектов модели.

В общем случае множества Z , O , M , S и SW включают подмножества прикладных и системных объектов. Прикладные объекты создаются разработчиком приложения и дополняются предопределенными системными объектами, предназначенными для поддержки взаимодействия модулей, сервисов и РП с компонентами FDE-SWFs и внешними информационно-вычислительными ресурсами и системами при подготовке и проведении экспериментов. Конструктор вычислительной модели поддерживает возможность создания новых операций в O на основе РП, а также генерацию программ на языке Python для РП с последующим включением сгенерированных программ в множество M .

Вычислительная модель представляет концептуальный уровень вычислительной среды СОНП, на котором определяются понятия и связи между объектами программно-аппаратного, программно-алгоритмического и сервис-ориентированного уровней (рис. 2). Она позволяет разработчикам и пользователям приложений взаимодействовать с ВСГР и управлять ее компонентами на абстрактном концептуальном уровне, который во многом скрывает детали организации вычислительных процессов на других уровнях.

Конструктор РП обеспечивает их построение на основе процедурной или непроедурной постановок задач. В первом случае, разработчик приложения самостоятельно строит рабочий процесс в интерактивном режиме с использованием предопределенного набора операторов выполнения операции, ветвления, различного вида циклов, обработки параллельных списков данных и других конструкций. В случае непроедурной постановки задачи, разработчик определяет множества исходных и целевых параметров. Затем встроенный планировщик автоматически строит последовательность выполнения операций РП с целью расчета значений целевых параметров.

После завершения построения РП данный конструктор обеспечивает возможность автоматического формирования их спецификаций на языке BPEL, а также автоматическую генерацию программного кода выполнения этих рабочих процессов на языке Python с целью обеспечения возможности запуска и интерпретации РП автономно от среды FDE-SWFs.

Конструктор РП поддерживает визуализацию РП в виде двудольного направленного ациклического графа. Граф включает только два типа вершин (параметры и операции) и два типа дуг между этими вершинами: входная дуга соединяет вершину-входной параметр с вершиной-операцией, выходная дуга соединяет вершину-операцию с вершиной-выходным параметром. Примеры визуализации РП, построенных на основе процедурной и непроцедурной постановок задач, приведены на рис. 3.



Рис. 2. Уровни архитектуры вычислительной среды сервис-ориентированных приложений.
Fig. 2. Architecture layers of computing environment for service-oriented applications.

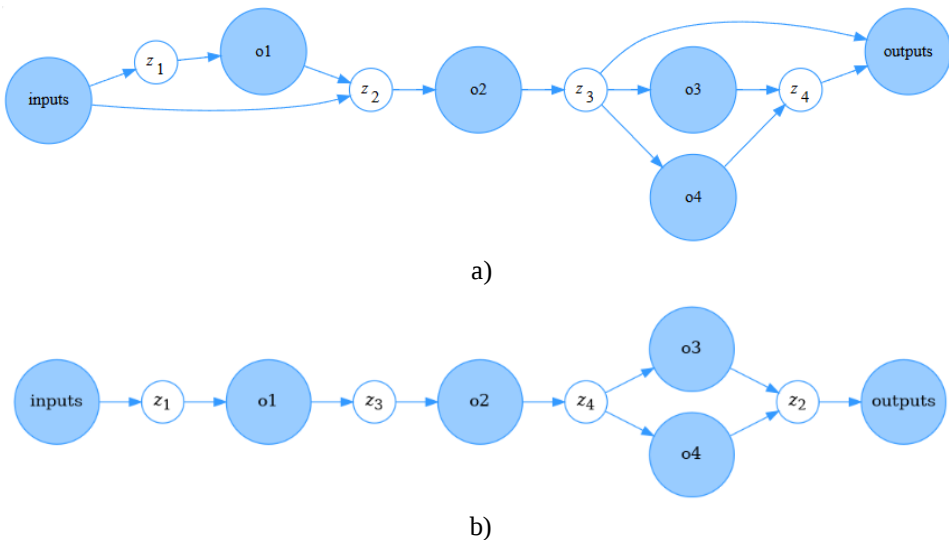


Рис. 3. Визуализация РП, построенного на основе процедурной(а) и непроцедурной (b) постановок задач.
Fig. 3. Visualization of the scientific workflow based on procedural (a) and non-procedural (b) problem formulations.

5. Система управления РП

На рис. 4 представлена схема работы ВСГР. FDE-SWFs реализован на программной платформе Node.js. Разработчики приложений с помощью веб-интерфейса комплекса описывают вычислительную модель СОИП. На вычислительной модели разработчик строит РП по процедурной или не процедурной постановкам задач. РП, может быть реализован следующими способами: на основе композиции WSDL-сервисов, представленной на языке программирования Python; в виде отдельного WPS-сервиса, который реализуется традиционный РП на основе исполняемых модулей и заданий для внешних метапланировщиков и систем управления заданиями, таких как Condor DAGMan и HTCondor; с помощью стандартизированного декларативного языка BPEL.

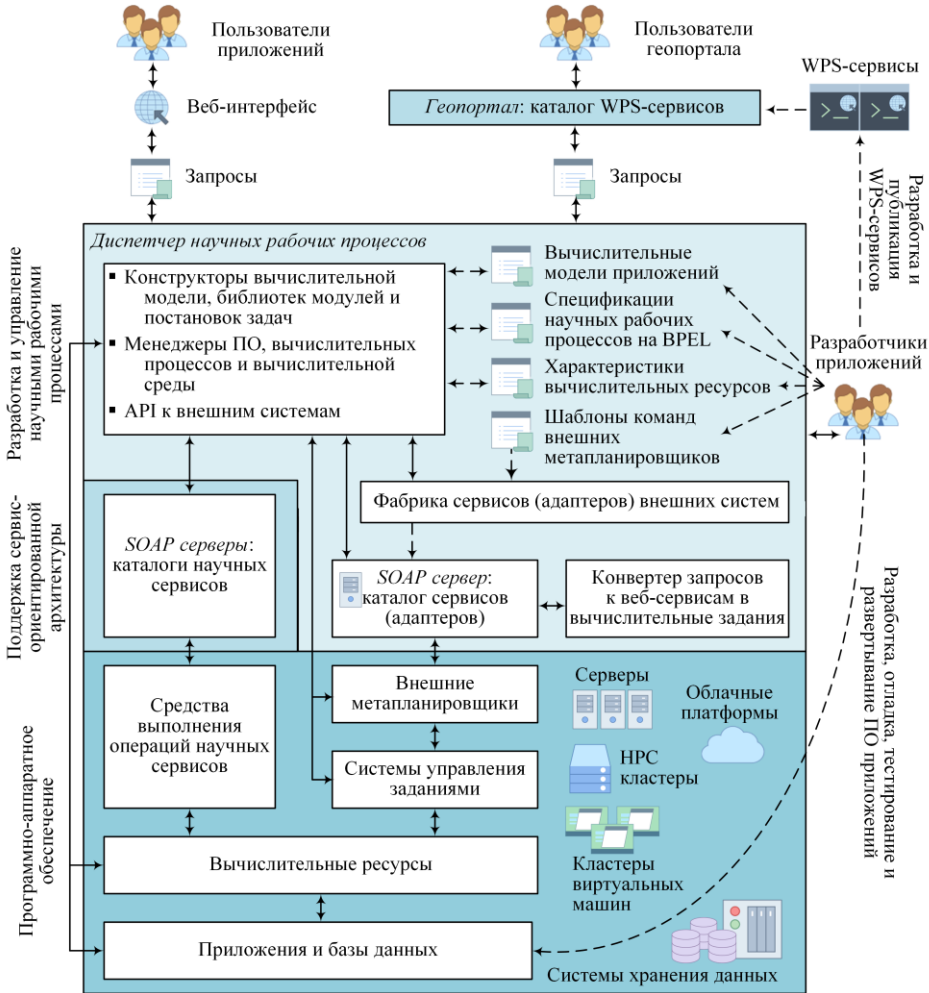


Рис. 4. Схема разработки и применения приложений.

Fig. 4. Scheme of application development and use.

В последнем случае, РП может использоваться любыми внешними СУРП, поддерживающими BPEL. На рис. 5 представлены схемы выполнения РП под управлением FDE-SWFs.

В рамках Схемы 1 пользователь с помощью интерфейса FDE-SWFs формулирует постановку задачи на вычислительную модель предметной области. По сформулированной постановке задачи модуль планирования вычислений выполняет построение плана решения задачи в виде РП во внутреннем представлении системы. Модуль преобразования производит конвертацию РП на BPEL для использования во внешних СУРП. Далее модуль интерпретации осуществляет асинхронно параллельное выполнение последовательности операций РП, представленных в виде композиции WSDL-сервисов. Полученные в результате выполнения композиции WSDL-сервисов расчетные данные сохраняются в хранилище данных. По завершению выполнения РП пользователь может осуществить визуализацию расчетных данных с целью их дальнейшего анализа.

Схема 2 в отличие от Схемы 1 предполагает, что операции РП, вместо представленные программными модулями. Процесс выполнения РП состоит в следующем. Специалист предметной области (конечный пользователь) формулирует постановку задачи, подготавливает исходную информацию и исполняемые модули с помощью менеджера вычислений. Затем выбирает необходимые ресурсы (ресурсы суперкомпьютерных ЦКП, облачных платформ, собственных высокопроизводительных серверов и др.) с учетом квот на их использование и способы доступа к ним. Менеджер автоматически строит РП и генерирует спецификацию вычислительного задания. Подсистема контейнеризации осуществляет выделение требуемых ресурсов, подготовку образов в соответствии с классами ресурсов и прикладными модулями РП с последующим запуском контейнеров на данных ресурсах (рис. 6). По готовности ресурсов для запуска РП передается сообщение СУПЗ (например, HTCondor или PBS Torque), которая также интегрирована в образы. После завершения выполнения РП метапланировщик отправляет полученные результаты в расчетную базу данных. Пользователь также может произвести визуализацию расчетных данных.

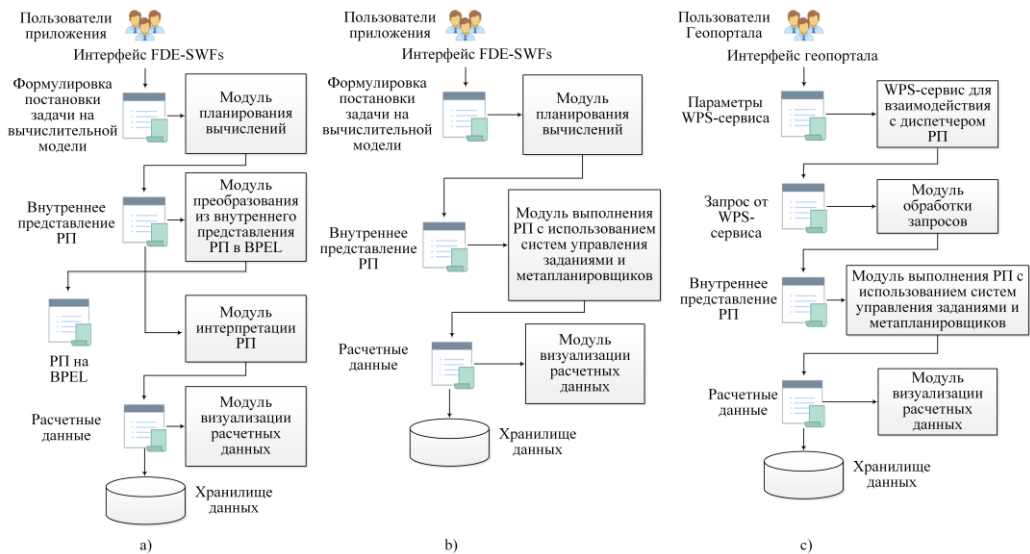


Рис. 5. Схемы выполнения РП: Схема 1 (a), Схема 2 (b) и Схема 3 (c).

Fig. 5. Schemes of the scientific workflow execution: Scheme 1 (a), Scheme 2 (b) and Scheme 3 (c).

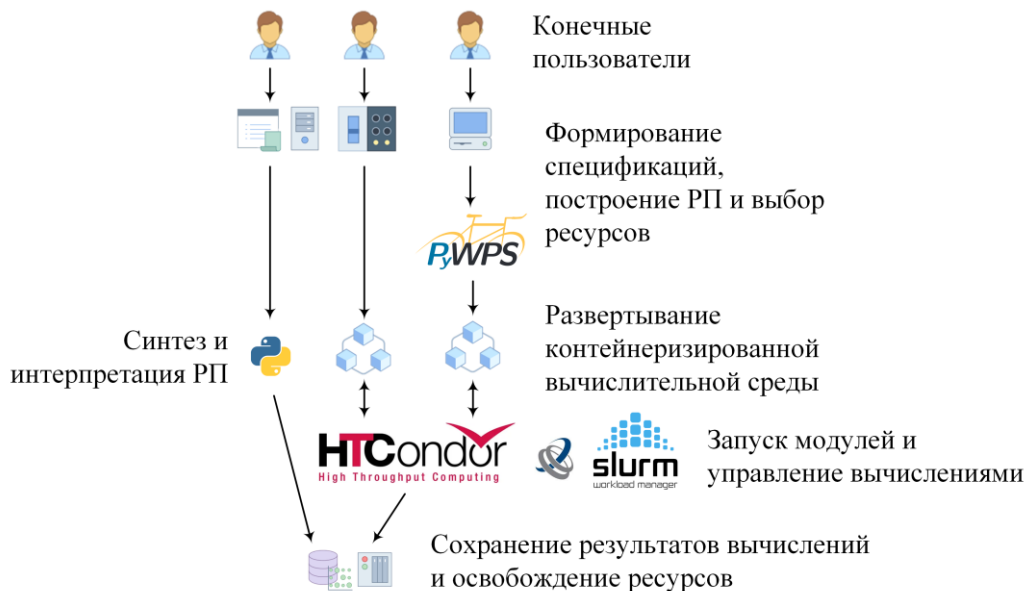


Рис. 6. Выполнение РП в контейнеризированной вычислительной среде.
Fig. 6. Scientific workflow execution in a containerized computing environment.

Возможность выполнения РП с помощью WPS-сервисов реализуется Схемой 3. В этом случае РП регистрируются как WPS-сервисы. FDE-SWFs автоматизирует создание, регистрацию и применение WPS-сервисов. В частности, FDE-SWFs автоматически регистрирует программные модули и РП в виде асинхронных WPS-сервисов в соответствующих каталогах на геопортале. РП могут включать вызовы других сервисов WPS, что позволяет работать с наборами сервисов. В FDE-SWFs поддерживается возможность обмена файлами между WPS-сервисами в качестве их параметров, в том числе обмен данными с системой хранения данных геопортала.

На рис. 7 приведен график времени разработки и выполнения тестового приложения с двумя РП в ОТ и FDE-SWFs для всех трех схем, представленных выше, с накоплением итогового результата. Были учтены следующие этапы разработки и применения приложения для одной из задач исследования процессов функционирования и свойств модели энергетической инфраструктуры:

- 1) описание вычислительной модели;
- 2) конструирование РП по процедурной постановке задачи;
- 3) конструирование РП по непроцедурной постановке задачи;
- 4) конфигурирование ресурсов ВСГР;
- 5) ввод исходных данных;
- 6) запуск и выполнение РП;
- 7) получение результатов вычислений;
- 8) визуализация результатов вычислений.

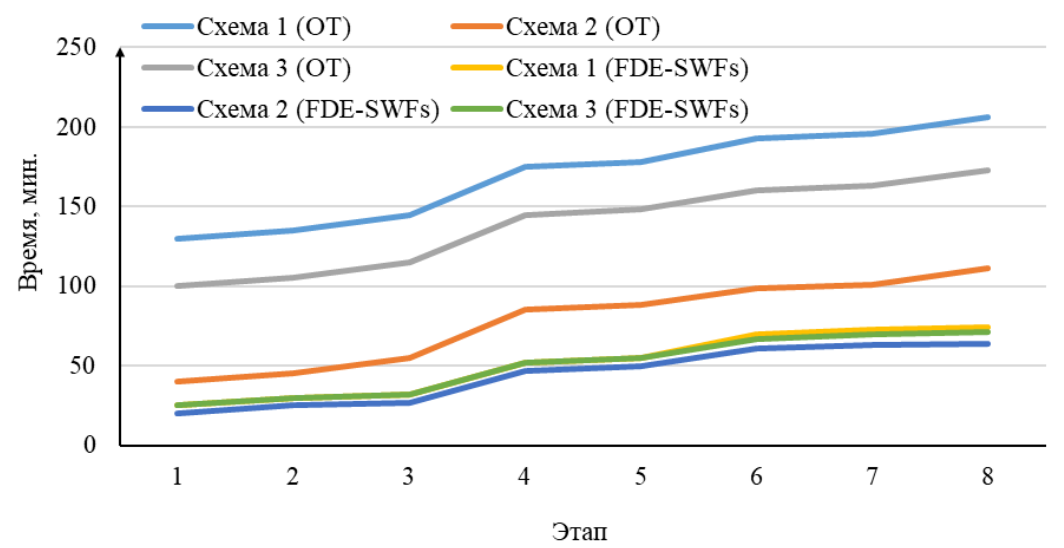


Рис. 7. Время разработки и выполнения тестового приложения.
Fig. 7. Test application development and execution makespan.

При работе с сервисами в ОТ дополнительно учитывались временные затраты на их создание в ручном режиме. В FDE-SWFs сервисы для выполнения модулей прикладного ПО создаются автоматически. Результаты сравнительного анализа времени выполнения показывают, что развитие функциональных возможностей в FDE-SWFs позволило существенно сократить временные затраты при выполнении всех трех схем.

6. Сравнительный анализ

В рамках сравнительного анализа рассматриваются следующие функциональные возможности FDE-SWFs:

- поддержка РП в виде направленного графа с циклами и ветвлениями (c_1);
- поддержка сервис-ориентированных РП (c_2);
- поддержка стандарта спецификации РП (c_3);
- поддержка взаимодействия с WPS-сервисами (c_4);
- поддержка контейнеризации ПО (c_5);
- генерация автономных программ на базовом языке программирования по выполнению РП независимо от среды СУРП (c_6).

Степень реализации функциональной возможности в СУРП определяется следующими показателями: реализовано (1), частично реализовано (0.5), не реализовано (0). Веса w_1-w_6 востребованности функциональной возможности заданы на основе агрегированных субъективных оценок пользователей. Результаты сравнительного анализа приведены в табл. 2. E – это суммарная оценка рассматриваемого набора функциональных возможностей СУРП, определяемая соотношением

$$E = \sum_{i=1}^n w_i c_i,$$

где n – число функциональных возможностей.

Табл. 2. Функциональные возможности СУРП.

Table 2. WMSs capabilities.

СУРП / WMS	c_1	c_2	c_3	c_4	c_5	c_6	w_1	w_2	w_3	w_4	w_5	w_6	E
OT	0.5	0	0	0.5	0.5	0	0.50	0.75	0.50	0.30	0.90	0.30	0.85
Pegasus	0	0	0	0	1.0	0.5						1.05	
Apache Airflow	0	1.0	0	0	1.0	0.5						1.80	
Galaxy	1.0	1.0	1.0	0	1.0	0						2.65	
FDE- SWFs	1.0	1.0	1.0	0.5	0.5	1.0						2.65	

Для сравнения выбраны предшественник FDE-SWFs – фреймворк OT, система Pegasus, одна из лидеров среди традиционных СУРП, и системы Apache Airflow [42] и Galaxy, представляющие активно развивающееся направление систем поддержки СОИП. В контексте рассматриваемого набора возможностей лидерами являются Galaxy и FDE-SWFs. Это во многом обусловлено поддержкой сервис-ориентированных РП, стандартизацией спецификаций рабочих процессов и особенностями специализации данных систем. Так, например, характеристики c_4 и c_6 имеют невысокие веса w_4 и w_6 среди пользователей традиционных СУРП. Однако с точки зрения пользователей FDE-SWFs в области исследования систем энергетики эти характеристики приобретают большую важность.

7. Практическое использование

В настоящее время FDE-SWFs успешно используется при разработке и применении ряда приложений для решения задач в области исследования свойств и процессов функционирования критически важных энергетических инфраструктур. В частности, с его помощью осуществляется подготовка и проведение крупномасштабных экспериментов для решения следующих задач:

- глобальный анализ степени уязвимости энергетических инфраструктур [43];
- оценка падения производительности систем энергетики из-за неординарного потока отказов их элементов при возникновении крупных внешних возмущений [43];
- профилирование и последующая оценка эффективности работы алгоритмов определения надежности функционирования энергетических систем различной конфигурации [44];
- определение наиболее подходящих алгоритмов структурно-параметрической оптимизации моделей энергетических комплексов на разных уровнях их территориально-отраслевой иерархии с помощью тестирования и многокритериального выбора исследуемых алгоритмов [45].

Отличительными особенностями FDE-SWFs при разработке СОИП на основе РП в сравнении с известными СУРП являются:

- использование технологии In-Memory Data Grid для размещения распределенных баз данных в оперативной памяти узлов среды с целью существенного ускорения обработки и анализа этих данных [43];
- создание испытательных стендов, предоставляющих разработчикам средства проведения экспериментов по оценке качества обрабатываемых данных, функционирования прикладного ПО, анализа результатов расчетов и других особенностей разрабатываемых приложений [44, 45];
- конструирование РП с использованием специальных системных операторов, в том числе операторов агрегирования и дезагрегирования данных, проведения многометодных расчетов, динамического планирования вычислений и др. [44, 45].

FDE-SWFs активно используется в учебном процессе в рамках дисциплин по изучению параллельных и распределенных вычислений для студентов и аспирантов образовательных организаций.

8. Заключение

С развитием научных приложений, основанных на рабочих процессах, возникает обоснованная потребность в их развертывании с использованием сервисов. Сервис-ориентированные СУПП зачастую существенно расширяют возможности традиционных систем подобного назначения. Они реализуют новую архитектуру, отвечающую современным бизнес-парадигмам проведения крупномасштабных междисциплинарных исследований на основе РП и в полной мере использующей преимущества информационных и коммуникационных технологий.

В этой связи в статье рассмотрены важные аспекты разработки и применения СОНП и реализации вышеупомянутой архитектуры в ВСГР средствами FDE-SWFs. Отличительная особенность FDE-SWFs заключается в обеспечении разнообразных способов построения и выполнения РП. РП строится по процедурной или непроцедурной постановкам задач. Они могут быть реализованы на основе композиции WSDL-сервисов, в виде отдельного WPS-сервиса, который реализуется традиционный РП на основе исполняемых модулей и заданий для внешних метапланировщиков и СУПЗ, а также с помощью стандартизированного декларативного языка BPEL. Соответственно в FDE-SWFs реализовано три схемы разработки и выполнения РП. Кроме того, показано, что в сравнении с известными сервис-ориентированными СУПП FDE-SWFs предоставляет ряд важных дополнительных возможностей. В их числе использование технологии In-Memory Data Grid, создание испытательных стендов и конструирование РП с использованием специальных системных операторов. В настоящее время FDE-SWFs успешно используется при разработке ряда приложений для решения задач анализа производительности и уязвимости энергетических инфраструктур и исследования эффективности работы алгоритмов структурно-параметрической оптимизации таких инфраструктур.

В рамках будущих исследований пристальное внимание будет вопросам повышения степени автоматизации интеграции и контейнеризации системного и прикладного ПО с целью дальнейшего улучшения эффективности развертывания и выполнения приложений в ВСГР, а также оптимизации использования вычислительных ресурсов. Кроме того, планируется разработка методов и средств обеспечения воспроизводимости вычислительных экспериментов по выполнению РП.

Список литературы / References

- [1]. Erwin D.W., Snelling D.F. UNICORE: A Grid Computing Environment. *Lecture Notes in Computer Science*, 2001, vol. 2150, pp. 825-834. DOI: 10.1007/3-540-44681-8_116.
- [2]. Litzkow M.J., Livny M., Mutka M.W. Condor – A hunter of idle workstations. In *Proc. of the 8th International Conference on Distributed Computing Systems*, Institute of Electrical and Electronics Engineers, 1988, pp. 104-111. DOI: 10.1109/DCS.1988.12507.
- [3]. Deelman E., Vahi K., Juve G., Rynge M., Callaghan S., Maechling P. J., Mayani R., Chen W., da Silva R.F., Livny M., Wenger K. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 2015, vol. 46, pp. 17-35. DOI: 10.1016/j.future.2014.10.008.
- [4]. Talia D. Workflow systems for science: Concepts and tools. *International Scholarly Research Notices*, 2013, vol. 2013, pp. 404525. DOI: 10.1155/2013/404525.
- [5]. Da Silva R.F., Filgueira R., Pietri I., Jiang M., Sakellariou R., Deelman E. A characterization of workflow management systems for extreme-scale applications. *Future Generation Computer Systems*, 2017, vol. 75, pp. 228-238. DOI: 10.1016/j.future.2017.02.026.
- [6]. Brown A., Johnston S., Kelly K. Using service-oriented architecture and component-based development to build web service applications. *Rational Software Corporation*, 2002, vol. 6. 16 p.

- [7]. Afgan E., Baker D., Coraor N., Chapman B., Nekrutenko A., Taylor J. Galaxy CloudMan: delivering cloud compute clusters. BMC bioinformatics, 2010, vol. 11, no. 12, pp. 1-6. DOI: 10.1186/1471-2105-11-S12-S4.
- [8]. Balis B. HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows. Future Generation Computer Systems, 2016, vol. 55, pp. 147-162. DOI: 10.1016/j.future.2015.08.015.
- [9]. Hilman M.H., Rodriguez M.A., Buyya R. Workflow-as-a-service cloud platform and deployment of bioinformatics workflow applications. In Knowledge Management in the Development of Data-Intensive Systems (In Mistrik I., Galster M., Maxim B., Tekiner-dogan B. editors). Boca Raton, FL, USA, CRC Press, 2021, pp. 205-226.
- [10]. Papazoglou M. Web Services: Principles and Technology. New York, Pearson Education, 2008. 752 p.
- [11]. Welke R., Hirschheim R., Schwarz A. Service-oriented architecture maturity. Computer, 2011, vol. 44, no. 2, pp. 61-67. DOI: 10.1109/MC.2011.56.
- [12]. Tsalgatidou A., Pilioura T. An overview of standards and related technology in web services. Distributed and Parallel Databases, 2002, vol. 12, pp. 135-162. DOI: 10.1023/A:1016599017660.
- [13]. Ananthakrishnan R., Chard K., Foster I., Tuecke S. Globus platform-as-a-service for collaborative science applications. Concurrency and Computation: Practice and Experience, 2015, vol. 27, no. 2, pp. 290-305. DOI: 10.1002/cpe.3262.
- [14]. Foster I. Globus Online: Accelerating and democratizing science through cloud-based services. IEEE Internet Computing, 2011, vol. 15, no. 3, pp. 70-73. DOI: 10.1109/MIC.2011.64.
- [15]. Foster I. Globus toolkit version 4: Software for service-oriented systems. Journal of Computer Science and Technology, 2006, vol. 21, pp. 513-520. DOI: 10.1007/s11390-006-0513-y.
- [16]. Foster I., Kesselman C. The Grid: Blueprint for a new computing infrastructure. Morgan-Kaufmann, 2002. 667 p.
- [17]. Foster I., Kesselman C. Globus: A metacomputing infrastructure toolkit. International Journal of Supercomputer Applications and High Performance Computing, 1997, vol. 11, no. 2, pp. 115-128. DOI: 10.1177/109434209701100205.
- [18]. Juric M.B., Chandrasekaran S., Frece A., Hertis M., Srdic G. WS-BPEL 2.0 for SOA Composite applications with oracle SOA Suite 11g. Packt Publishing Ltd, 2010. 644 p.
- [19]. Kim S., Bastani F.B., Yen I.L., Chen I.-R. High-assurance synthesis of security services from basic microservices. In Proc. of the 14th Intern. Symposium on Software Reliability Engineering (ISSRE 2003), IEEE, 2003, pp. 154-165. DOI: 10.1109/ISSRE.2003.1251039.
- [20]. Fielding R.T. Architectural styles and the design of network-based software architectures. University of California, Irvine, 2000. 180 p.
- [21]. Rajasekar A. iRODS primer: integrated rule-oriented data system. Morgan & Claypool Publishers, 2010. 143 p.
- [22]. Sukhoroslov O. Building web-based services for practical exercises in parallel and distributed computing. Journal of Parallel and Distributed Computing, 2018, vol. 118, pp. 177-188. DOI: <https://doi.org/10.1016/j.jpdc.2018.02.024>.
- [23]. Savchenko D.I., Radchenko G.I., Taipale O. Microservices validation: Mjolinrr platform case study. In Proc. of the 38th International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, 2015, pp. 235-240. DOI: 10.1109/MIPRO.2015.7160271.
- [24]. Бухановский А.В., Ковальчук С.В., Марьин С.В. Интеллектуальные программные комплексы компьютерного моделирования сложных систем: концепция, архитектура и примеры реализации. Известия высших учебных заведений. Приборостроение, 2009 г., том 52, № 10, стр. 5-24. (in Russian).
- [25]. Васильев В.Н., Князьков К.В., Чуров Т.Н., Насонов Д.А., Марьин С.В., Ковальчук С.В., Бухановский А.В. CLAVIRE: облачная платформа для обработки данных больших объемов. Информационно-измерительные и управляющие системы, 2012, том 10, вып. 11, стр. 7-16. (in Russian).
- [26]. Поляков С.В., Выродов А.В., Пузырьков Д.В., Якобовский М.В. Облачный сервис для решения многомасштабных задач нанотехнологии на суперкомпьютерных системах. Труды ИСП РАН, 2015, том 27, вып. 6, стр. 409-420. (in Russian). DOI 10.18522/2311-3103-2016-12-103114.
- [27]. Кудрявцев А.О., Кошелев В.К., Избышев А.О., Дудина И.А., Курмангалеев Ш.Ф., Аветисян А.И., Иванников В.П., Велихов В.Е., Рябинкин Е.А. Разработка и реализация облачной системы для решения высокопроизводительных задач. Труды ИСП РАН, 2013, том 24, стр. 13-34.

- [28]. Ханчук А.И., Сорокин А.А., Смагин С.И., Королев С.П., Макогонов С.В., Тарасов А.Г., Шестаков Н.В. Развитие информационно-телекоммуникационных систем в ДВО РАН. Информационные технологии и вычислительные системы, 2013, № 4, стр. 45-57.
- [29]. Шокин Ю.И., Федотов А.М., Жижимов О.Л. Технологии создания распределенных информационных систем для поддержки научных исследований. Вычислительные технологии, 2015, том. 20, № 5, стр. 251-274.
- [30]. Брагинская Л.П., Григорюк А.П., Ковалевский В.В., Загорулько Г.Б. Разработка научной среды для комплексных исследований в активной сейсмологии. Информационные технологии в науке, образовании и управлении: Материалы XLIV междунар. конф. и XIV междунар. конф. молодых учёных IT + S&E`16 (22 мая – 01 июня 2016 г., г. Гурзуф, Россия), М., Институт новых информационных технологий, 2016, стр. 3-10.
- [31]. Массель Л.В., Болдырев Е.А., Макагонова Н.Н., Копайгородский А.Н., Черноусов А.В. ИТ-инфраструктура научных исследований: методический подход и реализация. Вычислительные технологии, 2006, том 11, № S8, стр. 59-68.
- [32]. Бычков И.В., Ружников Г.М., Фёдоров Р.К., Шумилов А.С. Система планирования и выполнения композиций веб-сервисов в гетерогенной динамической среде. Вычислительные технологии, 2016, том 21, № 6, стр. 18-35.
- [33]. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А. Сервис-ориентированное управление распределёнными вычислениями на основе мультиагентных технологий. Известия Южного федерального университета. Технические науки, 2014, № 12, стр. 17-27.
- [34]. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Корсуков А.С. Сервис-ориентированный подход к организации распределённых вычислений с помощью инструментального комплекса DISCENT. Информационные технологии и вычислительные системы, 2014, № 2, стр. 7–15.
- [35]. Kostromin R., Basharina O., Feoktistov A., Sidorov I. Microservice-Based Approach to Simulating Environmentally-Friendly Equipment of Infrastructure Objects Taking into Account Meteorological Data. *Atmosphere*, 2021, vol. 12, no. 9, pp. 1217. DOI: 10.3390/atmos12091217.
- [36]. Михеев А., Орлов М. Война стандартов в мире workflow. Режим доступа: <https://ecm-journal.ru/material/Vojna-standartov-v-mire-workflow> (дата обращения: 03.05.2024).
- [37]. Артамонов И.В. Описание бизнес-процессов: вопросы стандартизации. Прикладная информатика, 2011, № 3 (33), стр. 20-28.
- [38]. Web Services Business Process Execution Language Version 2.0. Available at: <https://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, accessed 03.05.2024.
- [39]. Common Workflow Language (CWL). Available at: <https://www.commonwl.org/>, accessed 03.05.2024.
- [40]. Feoktistov A., Edelev A., Tchernykh A., Gorsky S., Basharina O., Fereferov E. An Approach to Implementing High-Performance Computing for Problem Solving in Workflow-based Energy Infrastructure Resilience Studies. *Computation*, 2023, vol. 11, no. 12, pp. 243. DOI: 10.3390/computation11120243.
- [41]. Feoktistov A., Gorsky S., Sidorov I., Bychkov I., Tchernykh A., Edelev A. Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment. *Communications in Computer and Information Science*, 2020, vol. 1087, pp. 18-32. DOI: 10.1007/978-3-030-41005-6_2.
- [42]. Apache Airflow. Available at: <https://airflow.apache.org/>, accessed 03.05.2024.
- [43]. 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, том 28, № 6, стр. 151-164. DOI: 10.25743/ICT.2023.28.6.013.
- [44]. Danilov G., Voskoboinikov M. Testbed-based approach to testing a library for evaluating network reliability algorithms. In *Proc. of the International Workshop on Critical Infrastructures in the Digital World (IWCi-2024)*, March 14-20, 2024, Bolshoe Goloustno, Russia, Irkutsk, ESI SB RAS, 2024, pp. 3-4.
- [45]. Феоктистов А.Г., Воскобойников М.Л., Еделев А.В. Методы и средства разработки и применения испытательного стенда сервис-ориентированных приложений. Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределённых сред (ICCS-DE 2024, 1-5 июля 2024 г., г. Иркутск, Россия), Иркутск, Изд-во ИДСТУ СО РАН, 2024, стр. 164-168.

Информация об авторах / Information about authors

Александр Геннадьевич ФЕОКТИСТОВ – доктор технических наук, доцент, главный научный сотрудник, заместитель директора ИДСТУ СО РАН по научной работе. Области исследования: параллельные и распределенные вычисления, научные приложения, мультиагентные технологии, имитационное моделирование и испытательные стенды.

Alexander Gennadevich FEOKTISTOV – Dr. Sci. (Tech.), Associate Professor, Chief Researcher, Deputy Director of ISDCT SB RAS for scientific work. Fields of research: parallel and distributed computing, scientific applications, multi-agent technologies, simulation modeling, and testbeds.

Михаил Леонтьевич ВОСКОБОЙНИКОВ – младший научный сотрудник ИДСТУ СО РАН. Сфера научных интересов: параллельные и распределенные вычисления, сервис-ориентированное программирование и испытательные стенды.

Mikhail Leontevich VOSKOBOINIKOV – Junior researcher of ISDCT SB RAS. Research interests: parallel and distributed computing, service-oriented programming, and testbeds.

Андрей Николаевич ЧЕРНЫХ – доктор физико-математических наук, главный научный сотрудник ИСП РАН, профессор Центра научных исследований и высшего образования в Энсенате, Нижняя Калифорния, Мексика, директор лаборатории параллельных вычислений. Области исследований: теоретическая и экспериментальная многоцелевая оптимизация распределения ресурсов в облачной среде, безопасность, теория расписаний в условиях неопределенности, эвристики и метаэвристики, адаптивное распределение ресурсов, а также машинное обучение с сохранением конфиденциальности и полностью гомоморфным шифрованием.

Andrei Nikolaevitch TCHERNYKH – Dr. Sci (Phys.-Math.), Adjunct Chief Researcher at the ISP RAS, Full Professor in Computer Science at CICESE Research Center, Ensenada, Baja California, Mexico, Director of Parallel Computing Laboratory. Fields of research: multiobjective resource optimization in a cloud environment, both theoretical and experimental, security, scheduling with uncertainty, heuristics and meta-heuristics, adaptive resource allocation, and privacy-preserving machine learning with fully homomorphic encryption.

