ИНТЕГРАЦИЯ УНАСЛЕДОВАННЫХ ПРИЛОЖЕНИЙ В ЭКОСИСТЕМУ НАУЧНЫХ ИССЛЕДОВАНИЙ "ЭНИ"

DOI: 10.36724/2072-8735-2020-14-8-33-41

Трунов Артем Сергеевич,
Московский Технический Университет Связи и Информатики, Москва, Россия, **greek17@yandex.ru**

Воронов Вячеслав Игоревич,
Московский Технический Университет Связи и Информатики, Москва, Россия, **Vorvi@mail.ru**

Воронова Лилия Ивановна,
Московский Технический Университет Связи и Информатики, Москва, Россия, **Voronova.lilia@yandex.ru**

Manuscript received 12 May 2020;
Revised 22 June 2020;
Accepted 03 August 2020

Ключевые слова: *экосистема, высокопроизводительные вычисления, распределенные вычисления, микросервисная архитектура*

В статье описывается разработка цифровой платформы для экосистемы научных исследований "Эни", основанной на микросервисном подходе, обеспечивающем автоматическое масштабирование вычислительных ресурсов системы при работе с большими данными. Приводится краткое описание структуры цифровой платформы, ее основных компонентов и функциональность базовых подсистем. Представлены два типа потребителей цифровой платформы – унаследованные и платформенные приложения. Показаны их возможности и ограничения. Описывается подсистема распределенных вычислений, обеспечивающая непрерывное управление и мониторинг микросервисной архитектуры платформы, в частности отвечает за: балансировку нагрузки, обнаружение сервисов, восстановление системы после сбоев, сквозную аутентификацию, "канареечные выкаты", контроль доступа. Представлена подсистема высокопроизводительных вычислений включает в себя модели и методы для организации параллельного расчета на различных аппаратных устройствах, таких как многопроцессорная система, кластер вычислительных устройств, объединенных локальной сетью, графический процессор. В качестве программных решений для организации параллельных вычислений используются технологии многопоточной обработки данных, интерфейс передачи сообщений MPI, технология CUDA. Также описывается подсистема интеллектуального анализа данных предназначена для развертывания разных типов нейронных сетей с различной архитектурой, в том числе: прямого распространения, сверточные, рекурентные и генеративные нейронные сети. Приведена интеграция унаследованного приложения MD-SLAG-MELT v13.0 в экосистему научных исследований с использованием платформенных интеграционных адаптеров. Представлены архитектуры и основные компоненты исходного программного комплекса и интегрированного программного комплекса MD-SLAG-MELT v14.0. Представлены результаты нагрузочного тестирования с анализом метрик производительности и времени отклика унаследованного приложения, при обработке больших данных.

Информация об авторах:
Трунов Артем Сергеевич, *Московский Технический Университет Связи и Информатики, старший преподаватель каф. ИСУиА, Руководитель направления ДИТ Сеть продаж ПАО Сбербанк, Москва, Россия*
Воронов Вячеслав Игоревич., *Московский Технический Университет Связи и Информатики, доцент каф. ИСУиА, к.т.н., Москва, Россия*
Воронова Лилия Ивановна, *Московский Технический Университет Связи и Информатики, зав. каф. ИСУиА, профессор, д.ф.-м.н., Москва, Россия*

Для цитирования:
Трунов А.С., Воронов В.И., Воронова Л.И. Интеграция унаследованных приложений в экосистему научных исследований "ЭНИ" // T-Comm: Телекоммуникации и транспорт. 2020. Том 14. №8. С. 33-41.

For citation:
Trunov A.S., Voronov V.I., Voronova L.I. (2020) Integration of legacy applications into "ENI" scientific research ecosystem. *T-Comm*, vol. 14, no. 8, pp. 33-41. (*in Russian*)

Введение

В ходе эволюции информационных технологий изменяются и процессы разработки программного обеспечения от унаследованных приложений до платформ. В последние десять лет приоритетной в разработке программных продуктов была концепция развития платформ. Основная идея платформы – предоставление инфраструктуры, технологий и интерфейсов, которые позволяют снизить затраты на разработку и сконцентрироваться непосредственно на бизнес ценности выпускаемого продукта.

Но в этой концепции есть ограничения, для конечного потребителя, когда тот сталкивается с невозможностью получить конкретную услугу или реализовать идею на выбранной платформе. Ему приходится делать выбор между платформами или в худшем случае дорабатывать их под свои требования, увеличивая временные и финансовые затраты на выпуск продукта для конечного потребителя.

Решая эту проблему, появилось новое направление - построение экосистемы. Под экосистемой понимают набор взаимосвязанных сервисов, которые развиваются, помогая друг другу выжить, как в реальной биологической системе.

Сегодня крупные технологические мировые компании, такие как Google, Apple, Microsoft, а также отечественные, такие как Яндекс и Сбербанк, строят свои собственные экосистемы [1, 2]. Крупные IT компании понимают, что без экосистемы, в перспективе ближайших десяти лет, сложно будет или даже невозможно конкурировать с компаниями, развивающими свои собственные экосистемы. Наряду с крупными компаниями и государство понимает необходимость выстраивания собственных экосистем. Ключевой темой выступления президента РФ на Петербургском международном экономическом форуме 2017 стала цифровая экономика, в основе которой должны лежать отраслевые экосистемы[3]. Также в 2017 году были утверждены два важнейших документа: Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы и Программы «Цифровая экономика Российской Федерации» [4, 5]. Оба стратегических документа уделяют большое внимание формированию экосистемы цифровой экономики.

В 2019 году департаментом информационных технологий города Москвы проведено исследование, целью которого было составить максимально полную карту работающих в городе цифровых экосистем. По данных исследования самыми популярными отраслями, в которых развиваются цифровые экосистемы являются: покупка товаров, финансы и банки, транспорт. Основной вектор развития отечественных цифровых экосистем нацелен на оказании услуг и продаже товаров конечному потребителю[6].

Однако, стратегический путь для социально-экономических преобразований – трансформация науки и, в частности, развитие систем научных исследований. Системы научных исследований как отдельный класс программного обеспечения появились еще до персональных компьютеров, но их развитие остановилось на уровне унаследованных приложений, разрабатываемых отдельно каждым научным сообществом. В редких случаях, некоторые университеты используют общие платформы, например BOINC, открытая программная платформа для распределенных вычислений [7].

Создание экосистемы в области научных исследований обеспечит взаимовыгодные отношения участников, позволит объединить научное сообщество, предоставит ему информационно-технологическую инфраструктуру, что будет способствовать росту разработки и выпуска инновационных продуктов для нужд цифровой экономики. В этой связи, тема работы является актуальной и имеет научную значимость.

Создание экосистемы – это сложный наукоемкий процесс, который включает в себя проектирование технической инфраструктуры, организацию процесса взаимодействия научных сообществ, а также цифровую программную платформу, как площадку для реализации научных исследований.

В статье описывается разработка авторами цифровой платформы для экосистемы научных исследований, под названием «Эни», основанной на сервисном подходе, обеспечивающем автоматическое масштабирование вычислительных ресурсов системы при работе с большими данными. В качестве примера реализации приведена интеграция унаследованного приложения MD-SLAG-MELT v13.0

Научные исследования, связанные с интеграцией разнородных приложений с высоким интеллектуальным потенциалом для научных исследований уже несколько лет ведутся на кафедре Интеллектуальных систем в управлении и автоматизации(ИСУиА) Московского технического университета связи и информатики(МТУСИ).

В рамках экосистемы «Эни» предполагается интегрировать архитектурные решения на основе современных информационных технологий [8-10], программные решения использующие модели и методы распределённых и параллельных высокопроизводительных вычислений [11-14], методы интеллектуального анализа данных, в том числе нейросетевые [15- 17], а также программные решения, связанные с биометрической идентификацией [18, 19], распознаванием жестового языка в режиме реального времени [20, 21], медицинской диагностикой [22- 24].

Структура и описание основных компонентов цифровой платформы экосистемы научных исследований «Эни»

В настоящее время современное программное обеспечение разрабатывается на основе независимых друг от друга и слабосвязанных между собой компонентов (сервисов). Такой подход к разработке существенно упрощает дальнейшее масштабирование разрабатываемого программного обеспечения. В связи с этим возникает необходимость учитывать взаимодействие между этими сервисами, разрабатывая специальные интерфейсы. Также существует возможность, не разрабатывая новые сервисы, использовать уже готовые апробированные решения, так называемые унаследованные приложения, дающие практический результат. Особенно актуально это в системах цифровой поддержки научных исследований, где legacy application имеют высокий исследовательский потенциал. Как правило, такие решения не поддерживают современные интерфейсы взаимодействия.

Еще одной отличительной особенностью современного программного обеспечения является реализация программной платформы, на основе которой, разрабатываются отдельные сервисы, что дает возможность унифицировать взаимодействие между сервисами, тем самым уменьшая количество ошибок, которые могут возникать при их взаимодействии. Платформа, как правило, состоит из внутренней части – бизнес хаба и внешней – презентационного слоя.

Также программное обеспечение для поддержки научных исследований является достаточно ресурсоемким и требовательным к вычислительным ресурсам. В этой области использование классических алгоритмов, которые выполняются последовательно в одном потоке, становится затруднительно, так как на получение результата может уходить большое количество времени, что серьезно влияет на производительность системы в целом, с учетом многопользовательского одновременного доступа. Для решения этой проблемы применяются различные методы высокопроизводительных вычислений, такие как параллельные и распределенные вычисления.

Таким образом, разработана модель цифровой платформы экосистемы научных исследований, интегрирующая методы и модели интеллектуального анализа данных, параллельных и распределенных вычислений, интеграционные адаптеры синхронного и асинхронного взаимодействия между сервисами и инструментальные средства разработки.

Структура цифровой платформы для экосистемы научных исследований реализована с учетом результатов анализа современных средств и подходов в разработке программного обеспечения. Компоненты платформенных подсистем спроектированы как отдельные микросервисы, с возможностью горизонтального и вертикального масштабирования. Горизонтальная масштабируемость достигается за счет применения технологии service mash [25], которая обеспечивает инфраструктурные нужды платформы, такие как: управление трафиком (таймауты, повторные попытки подключения, балансировка нагрузки); авторизация и аутентификация; аудит, мониторинг и логирование.

При структурно-функциональной декомпозиции цифровой платформы «Эни» выделяются четыре функциональных подсистемы:

- Подсистема высокопроизводительных вычислений
- Подсистема распределенных вычислений
- Подсистема интеллектуального анализа данных
- Подсистема платформенных сервисов

Структура цифровой программной платформы, как базисная часть архитектуры экосистемы научных исследований «Эни» представлена на рис. 1.

Подсистема распределенных вычислений обеспечивает непрерывное управление и мониторинг микросервисной архитектуры платформы, в частности, балансировка нагрузки, обнаружение сервисов, восстановление после сбоев, метрики, мониторинг, сквозная аутентификация, «канареечные выкаты», контроль доступа и д.р.

Подсистема высокопроизводительных вычислений включает в себя модели и методы для организации параллельного расчета на различных аппаратных устройствах, таких как многопроцессорная система, кластер вычислительных устройств, объединенных локальной сетью, графический процессор. В качестве программных решений для организации параллельных вычислений используются технологии многопоточной обработки данных, интерфейс передачи сообщений MPI, технология CUDA.

Подсистема интеллектуального анализа данных предназначена для развертывания разных типов нейронных сетей с различной архитектурой, в том числе: прямого распространения, сверточные, рекурентные и генеративные нейронные сети.

Подсистема платформенных сервисов – это фреймворк для разработки современных микросервисных приложений. Включает в себя библиотеки для авторизации, аутентификации, REST API, распределенное кэширование, журналирование, мониторинг, систему параметров и др.

Рис. 1. Архитектура экосистемы научных исследований «Эни»: структура цифровой платформы и ее взаимодействие с унаследованными приложениями

В рамках экосистемы «Эни» выделяются два типа «потребителей»: платформенные и унаследованные приложения. Потребители, для реализации своих бизнес-задач могут использовать функционал разработанных платформенных подсистем.

Потребители первого типа (платформенные приложения) для взаимодействия с платформенными подсистемами используют внутреннее Java API, входящее в состав подсистемы платформенных сервисов.

Потребители второго типа (унаследованные приложения) не входят в состав цифровой платформы, но могут взаимодействовать с ней через специальные адаптеры. Адаптеры работают по распространённым протоколам REST API и SOAP, что упрощает интеграцию с платформой. Для унаследованных приложений возможно три вида интеграции: синхронная через передачу REST запросов, асинхронная через Message Queue, и событийная по подписке.

В настоящее время в стадии интеграции в ИЭНИ находятся три *унаследованных* приложения: программный комплекс MD-SLAG-MELT [26], программно-аппаратный комплекс биометрической идентификации [27, 28] и программно-аппаратный комплекс распознавания русского жестового языка [29]. Параллельно ведется разработка платформенных приложений: интеллектуальная система ортопедического протезирования [30, 31], фреймворк для обработки больших данных в потоковой модели в среде HADOOP [32, 33] и ин-

теллектуальная система нейросетевой классификации сердечных ритмов на основе кардио-данных [22].

Программный комплекс MD-SLAG-MELT

Рассмотрим подробнее особенности интеграции цифровой платформы «Эни» с унаследованными приложениями на примере программного комплекса MD-SLAG-MELT v13.0.

«MD-SLAG-MELT v13.0» – это информационно-исследовательская система, которая обеспечивает реализацию комплексных компьютерных экспериментов для моделей многокомпонентных шлаковых расплавов с большим числом частиц (10^6 – 10^8 частиц). В рамках единой интегрированной среды распределенного программного комплекса реализуются компьютерное моделирование квантово-химическим, молекулярно-динамическим и статистико-геометрическим методом на основе комплексной модели многочастичной системы с сильным взаимодействием. Основным методом исследования является метод молекулярной динамики, суть которого состоит в численном интегрировании уравнений движения по фазовым траекториям. Задачи, которые решает данный программный комплекс, являются актуальными в области физической химии и технологии неорганических материалов, так как результаты проводимых в системе экспериментов используются при создании новых металлических материалов с заранее заданными свойствами.

ИИС «MD-SLAG-MELT v13.0» построена на основе классической трехзвенной архитектуры и представлена на рис. 2.

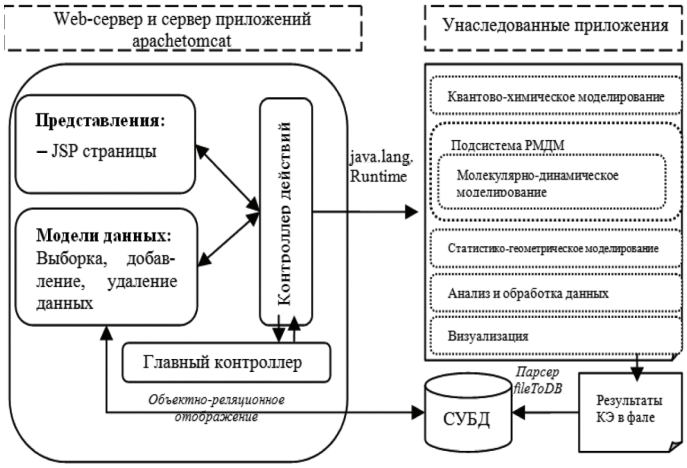

Рис. 2. Архитектура программного комплекса «MD-SLAG-MELT v13.0»

Основой «MD-SLAG-MELT v13.0» является сервер приложений, на котором развернут комплекс унаследованных приложений, реализующий различные виды компьютерного моделирования. Также на данном сервере функционируют приложения, обеспечивающие многомерную обработку результатов компьютерного эксперимента в виде гиперкуба данных и приложения, реализующие исследование структурных особенностей расплава.

На этапе структурно-функционального анализа программного комплекса «MD-SLAG-MELT v13.0» была выявлена проблема сильной связанности программных компонентов. В частности, комплекс из унаследованных приложений, выполняющих непосредственно компьютерное моделирование, представляют собой монолитное приложение. Данная проблема сильно ухудшает масштабируемость системы, а также приводит к жесткой привязке к устаревшим технологиям, что мешает взаимодействию данной ИИС с современными информационными системами.

Интеграция программного комплекса MD-SLAG-MELT v13.0 в экосистему научных исследований «Эни»

Для интеграции программного комплекса MD-SLAG-MELT V13.0 с экосистемой научных исследований используется подсистема распределенных вычислений и интеграционный адаптер. Рассмотрим входные и выходные интерфейсы унаследованных приложений программного комплекса MD-SLAG-MELT v13.0.

Рис. 3. Входные и выходные интерфейсы унаследованного приложения MD-SLAG-MELT 13.0

Унаследованное приложение запускалось исполняемым файлом в операционной системе Windows с входными параметрами в виде символьных потоков. Результат работы записывался в файловую систему Windows.

Для интеграции MD-SLAG-MELT v13.0 в экосистему «Эни» используется событийный и синхронный сегменты подсистемы распределенных вычислений, при этом необходимо через интеграционный адаптер *создать запросы* на создание файлов начальных условий и запуск унаследованного приложения, а после завершения работы приложения, сформировать ответ и передать его в цифровую платформу.

Для интеграционных взаимодействий в подсистеме распределенных вычислений используется протокол JSON-RPC. Этот формат основан на структуре JSON сообщений, в котором передается имя вызываемой функции и параметры. В ответе приходит результат выполнения запроса или код и текст ошибки.

В таблице 1 представлены форматы запросов и ответов на создание файлов начальных условий и запуск унаследованных приложений.

Таблица 1

Форматы запросов к унаследованным приложениям

Запрос и ответ в формате JSON-RPC	Описание
запрос --> {"method": "prepare", "params": [], "id": 1} ответ <-- {"result": “Ok”, "id": 1}	Запрос на передачу начальных параметров для компьютерного эксперимента
запрос --> {"method": "start", "params": [], "id": 2} ответ <-- {"result": “Ok”, "id": 1}	Запуск компьютерного эксперимента
запрос --> {"method": "start", "params": [], "id": 2} <-- "error": {"code": -32700, "message": "Parse error"}, "id": null}	Ошибка передачи параметров
запрос --> {"method": "start", "params": [], "id": 10} <-- "error": {"code": -32601, "message": "Procedure not found."}, "id": 10}	Ошибка вызова удаленной процедуры

Интеграционный адаптер на стороне программной платформы формирует два запроса в формате JSON-RPC. В ответ на два синхронных запроса приходят ответы об успешном выполнении. Если параметры запроса были сформированы некорректно или в запросе имя функции указано с ошибкой, то в ответе будет передана информация об ошибке.

После завершения работы унаследованного приложения, интеграционный адаптер слушает файл результатов и когда тот будет сформирован, вызовет метод на формирование ответа платформе. Так как время завершения работы унаследованного приложения может быть разное, для передачи ответа на платформу используется событийный сегмент. Между интеграционными адаптерами устанавливается специальная *интеграционная шина*, которая выступает в качестве брокера сообщений.

Интеграционная шина реализует метод producer-consumer. Producer устанавливается на стороне унаследованного приложения и передает результат его работы в топик, который является контейнером для временного хранения данных. Данные в топике хранятся по принципу очереди «первый пришел, первый вышел». Consumer подписывается на топик и ждет когда producer отправит сообщение в топик.

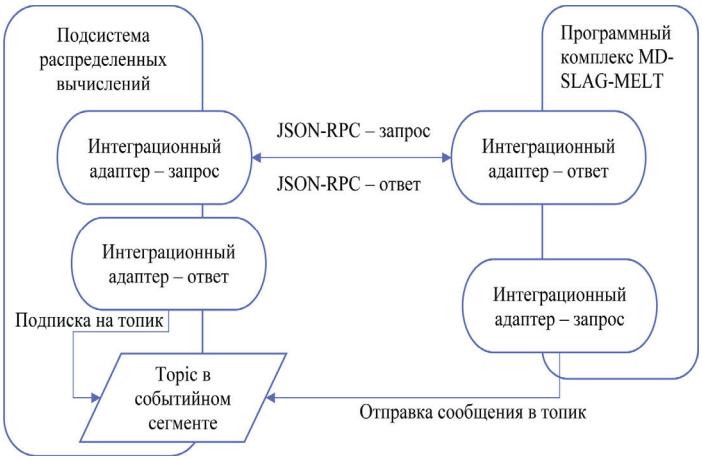

Рис. 4. Взаимодействие между цифровой платформой экосистемы научных исследований «Эни» и программным комплексом MD-SLAG-MELT v14.0

На рисунке 4 представлено взаимодействие между платформой экосистемы научных исследований и программным комплексом MD-SLAG-MELT v14.0.

Для возможности масштабирования унаследованного приложения произведена реорганизация структуры «MD-SLAG-MELT» с использованием компонентов подсистемы распределенной обработки данных. Новая структура программного комплекса представлена на рис. 5.

Рис. 5. Структура программного комплекса MD-SLAG-MELT v.14 в рамках «Эни»

Для обеспечения единого и унифицированного способа обмена сообщениями между новыми сервисами и унаследованным приложением использована технология контейнеризации Docker, которая обеспечит каждому сервису свой легковесный, быстро запускающийся и нетребовательный к ресурсам контейнер, в которой будет входить только само приложение и необходимое окружение (зависимости). Для обеспечения совместной работы сервисов используется набор основных паттернов и практик микросервисной архитектуры.

В классической микросервисной архитектуре сервис самостоятельно решает проблемы, связанные управлением трафиком, такие как таймауты, повторные попытки вызова, балансировка нагрузки, также управление безопасностью, в частности, авторизация и аутентификация пользователей, а также мониторинг, логирование и трассировка.

В структуре программного комплекса MD-SLAG-MELT v14.0 эти проблемы решаются за счет выделения специального прокси контейнера, который включается в состав каждого сервисного контейнера. Таким образом взаимодействие между сервисами MD-SLAG-MELT v14.0 происходят между прокси-контейнерами, а уже прокси-контейнеры взаимодействуют с самими сервисами.

Такой подход позволяет создать отказоустойчивую, масштабируемую систему, которая решает проблемы повторных попыток и circuit breaker.

Анализ результатов тестирования

Для того, чтобы убедиться, что интегрированный в «Эни» программный комплекс «MD-SLAG-MELT v14.0», способен справиться с высокими вычислительными нагруз-

ками, проведено его нагрузочное тестирование. Целью нагрузочного тестирование является сбор метрик, определение производительности и времени отклика унаследованного приложения, под воздействием большего количества запросов. Для проведения нагрузочного тестирования было запущено 4 контейнера с унаследованным приложением и проведены замеры времени отклика программного комплекса «MD-SLAG-MELT v14.0». Каждый контейнер устанавливается на отдельный вычислитель. Характеристики вычислителя: Intel Core i3-3220 3.3ГГц, 2 ядра, 4 ГБ RAM. Также для каждого контейнера было выделено разное количество аппаратных ресурсов (процессорное время и оперативная память) в процентном соотношении от мощности вычислителя.

В течение часа подавалась одинаковая нагрузка в 30 запросов с интервалом от 0 до 1000 миллисекунд. Полученные результаты замера нагрузки на контейнер за час времени представлены в табл. 2.

Таблица 2

Данные нагрузки

№	Процент ресурсов вычислителя в %	Среднее время ответа, сек	Пропускная способность (запросов в секунду)
1	25	1,5	0,67
2	50	0,4	2,23
3	75	0,13	8,84
4	100	0,074	13,54

Анализ полученных результатов показывает, что при средней нагрузке в 30 запросов в секунду с использование 25% мощности процессора время отклика достигло полторы секунды. При таком времени отклика конечному пользователю будет казаться, что приложение зависает. Контейнер с 50% использования ЦП испытывает ощутимую нагрузку, а остальные контейнеры обрабатывают данные в пределах приемлемого времени.

Увеличив нагрузку на унаследованное приложение, обеспечим в течение часа 100 запросов в секунду на контейнеры. Полученные результаты замера нагрузки на контейнер за час времени представлены в табл. 3.

Таблица 3

Данные нагрузки

№	Процент ресурсов вычислителя в %	Среднее время ответа, сек	Пропускная способность (запросов в секунду)
1	25	6,1	0,16
2	50	2,757	0,36
3	75	1,738	0,58
4	100	1,367	0,73

Таким образом, при средней нагрузке в 100 запросов в секунду, с использованием 25% мощности вычислителя, время отклика достигло шести секунда, из чего можно сделать вывод, что работать с таким временем отклика пользователю системы будет невозможно. Пользователю контейнера с 50% использования мощности вычислителя, также будет казаться, что система «зависла». Контейнеры с выделенными ресурсами на 75% и 100% также не справляются с данной нагрузкой, поэтому требуется повышение мощности и выделения дополнительных контейнеров.

Для автоматического запуска контейнеров в подсистеме распределенных вычислений предусмотрен специальный сервис прогнозирования нагрузки. Когда время отклика на запрос превышает 1 секунду, происходит запуск дополнительного контейнера с унаследованным приложением и часть пользовательских запросов перенаправляется на него.

На рисунке 6 представлен график зависимости времени отклика системы от количества запросов.

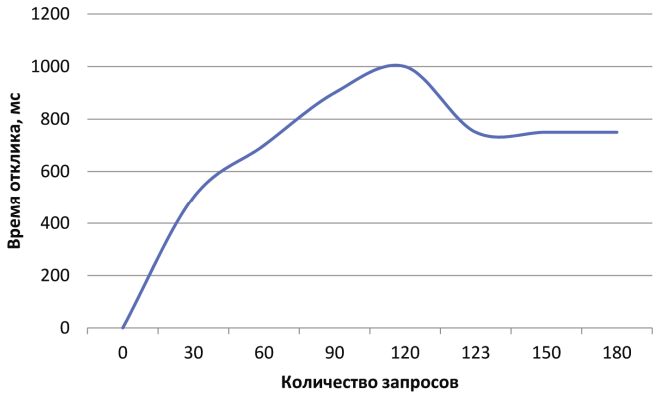

Рис. 6. График зависимости времени отклика системы от количества запросов

Согласно графику, при достижении 120 запросов в секунду, время отклика превысило 1 секунду. После этого автоматически был запущен дополнительный контейнер и время отклика снизилось на 25% и приняло приемлемое значение с точки зрения пользователя.

Заключение

Разработанная авторами цифровая платформа для экосистемы научных исследований «Эни» позволяет практически бесшовно интегрировать унаследованные приложения, предоставляя унифицированный интерфейс взаимодействия между ее участниками. При этом для обеспечения единого и унифицированного обмена сообщениями между платформенными сервисами и унаследованными приложениями использована технология контейнеризации Docker с набором основных паттернов и практик микросервисной архитектуры.

На примере интегрированного унаследованного приложения – программного комплекса MD-SLAG-MELT v14.0 исследована возможность автоматического управлением трафиком, балансировкой нагрузки, управлением безопасностью, мониторингом и логированием за счет прокси-контейнеризации сервисов, что позволяет создать отказоустойчивую, масштабируемую систему, обеспечивающую возможность работы с большими данными.

Литература

1. Azure API для FHIR и наша расширяющаяся экосистема партнеров - https://azure.microsoft.com/ru-ru/updates/azure-api-for-fhir-and-our-growing-ecosystem-of-partners.

2. Годовой отчет сбербанка 2019 https://www.sberbank.com/common/img/uploaded/files/pdf/yrep/sberbank-ar19-rus.pdf.

3. Доклад президента на ПМЭФ 2017 http://kremlin.ru/events/president/news/54667.

4. Указ Президента Российской Федерации от 09.05.2017 г. № 203 О Стратегии развития информационного общества в Российской Федерации на 2017 – 2030 годы.

5. Программа «Цифровая экономика Российской Федерации» http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf.

6. X5RetailGroup создает свою экосистему. https://retail-life.ru/x5-sozdast-cifrovuju-jekosistemu-ritejla-v-odnoj-knopke-za-15-let.

7. *David P. Anderson, Carl Christensen, Bruce Allen*. Designing a Runtime System for Volunteer Computing.

8. *Trunov A.S., Voronova L.I., Voronov V.I., Sukhachev D.I., Strelnikov V.G.* Legacy applications model integration to support scientific experiment // В сборнике: 2018 Systems of Signals Generating and Processing in the Field of on Board Communications 2018. С. 8350641.

9. *Воронова Л.И., Григорьева М.А., Воронов В.И., Трунов А.С.* Программный комплекс "MD SLAG-MELT" Информационно-исследовательской Системы "Шлаковые Расплавы" версии 10.0. Депонированная рукопись № 29-В2012 26.01.2012

10. *Безумнов Д.Н., Воронова Л.И., Варганова Е.Э.* Программа для автоматизации процесса прототипирования распределённых систем сбора и обработки информации. Свидетельство о регистрации программы для ЭВМ RU 2019610043, 09.01.2019. Заявка № 2018664390 от 13.12.2018.

11. *Пилипчак П.Е., Воронова Л.И., Трунов А.С.* Модель балансировки нагрузки для параллельного расчета системы n-частиц на гетерогенной кластерной системе // Современные наукоемкие технологии. 2014. № 5-2. С. 218-219.

12. *Трунов А.С., Сухачев Д.И., Воронова Л.И., Стрельников В.Г.* Программа интеграции унаследованных приложений для поддержки высокопроизводительных вычислений. Свидетельство о регистрации программы для ЭВМ RU 2019610042, 09.01.2019. Заявка № 2018664391 от 13.12.2018.

13. *Трунов А.С., Филатов А.М., Воронова Л.И., Усачев В.А.* Программа интеллектуального автоматического распределения вычислительной нагрузки в программно-аппаратном комплексе для высокопроизводительных вычислений. Свидетельство о регистрации программы для ЭВМ RU 2018666858, 21.12.2018. Заявка № 2018664407 от 13.12.2018.

14. *Воронова Л.И., Трунов А.С.* Оптимизация параллельного алгоритма подсистемы распределенного молекулярно-динамического моделирования // Межотраслевая информационная служба. 2011. № 3. С. 3-11.

15. *Воронова Л.И., Трунов А.С., Воронов В.И.* Разработка методов параллельного расчета коррелированной многочастичной системы на графическом процессоре // Вестник РГГУ. Серия: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. 2013. № 14 (115). С. 236-247.

16. *Воронов В.И., Воронова Л.И., Генчель К.В.* Применение параллельных алгоритмов в нейронной сети для распознавания жестового языка // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2018) VII Международная научно-техническая и научно-методическая конференция. Сборник научных статей. В 4-х томах. Под редакцией С.В. Бачевского. 2018. С. 207-212.

17. *Usachev V.A., Voronova L.I., Voronov V.I., Zharov I.A., Strelnikov V.G.* Neural network using to analyze the results of environmental monitoring of water // В сборнике: 2019 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2019 2019. С. 8706733.

18. *Voronov V.I., Zharov I.A., Bykov A.D., Trunov A.S., Voronova L.I.* «Designing a neural network identification subsystem in the hardware-software complex of face recognition» // T-Comm, v.14. №5-2020, ISSN 2072-8743

19. *Жаров И.А., Воронов В.И., Воронова Л.И., Быков А.Д., Вовик А.Г., Усачев В.А.* Программа для идентификации изображения в программном комплексе нейросетевого распознавания лица. Свидетельство о регистрации программы для ЭВМ RU 2020612466, 25.02.2020. Заявка № 2020611422 от 11.02.2020.

20. *Ежов А.А., Воронова Л.И., Воронов В.И., Гончаренко А.А., Артемов М.Д.* Программа для поддержки невербального общения, использующая машинное обучение на основе сверточной нейронной сети. Свидетельство о регистрации программы для ЭВМ RU 2019610179, 09.01.2019. Заявка № 2018664377 от 13.12.2018.

21. *Goncharenko A., Voronova L., Artemov M., Voronov V., Bezumnov D.* Sign language recognition information system development using wireless technologies for people with hearing impairments. Conference of Open Innovations Association, FRUCT. 2019. № 24. С. 104-109.

22. *Brus V.R., Voronova L.I., Voronov V.I.* Neural network classification of cardiac activity based on cardiogram data // 2020 Systems of signals generating and processing in the field of on board communications (IEEE Conference # 48371) 19-20 March 2020.

23. *Ван Юе, Воронова Л.И.* Классификация вида болезни позвоночника с использованием нейросети. В сборнике: Технологии информационного общества Материалы XIII Международной отраслевой научно-технической конференции. 2019. С. 407-410.

24. *Баширов А.Н., Воронов В.И.* Прогнозирование вероятности возникновения бронхиальной астмы у детей с применением алгоритма случайного леса // Современные наукоемкие технологии. 2019. № 12-2. С. 249-255.

25. Introducing Istio Service Mesh for Microservices by Christian Posta and Burr Sutter

26. Воронова Л.И., Григорьева М.А., Воронов В.И., Трунов А.С. Программный комплекс "md-slag-melt" для моделирования наноструктуры и свойств многокомпонентных расплавов. Расплавы. 2013. № 4. С. 36-49.

27. Быков А.Д., Воронов В.И., Воронова Л.И., Жаров И.А., Фадеев М.А., Усачев В.А. Программа обнаружения и трекинга лиц в видеопотоке в программно-аппаратном комплексе биометрической идентификации на основе нейросетевого распознавания лиц. Свидетельство о регистрации программы для ЭВМ RU 2020612484, 25.02.2020. Заявка № 2020610985 от 03.02.2020.

28. *Воронов В.И., Быков А.Д., Воронова Л.И.* Проектирование подсистемы детектирования лиц и интерфейса работы с базой данных в программно-аппаратном комплексе биометрической идентификации на основе нейросетевого распознавания лиц // T-Comm: Телекоммуникации и транспорт, т.14. №4-2020, ISSN 2072-8743

29. *Гончаренко А.А., Воронова Л.И., Воронов В.И., Ежов А.А., Артемов М.Д.* Программный комплекс для управления данными в информационно-коммуникационной системе социальной доступности для людей с ограниченными возможностями по слуху. Свидетельство о регистрации программы для ЭВМ RU 2019610962, 18.01.2019. Заявка № 2018665275 от 26.12.2018.

30. *Ачкасов Е.Е., Полукаров Н.В., Лазарева И.А., Жукова Е.В.* Персонифицированный подход к комплексной реабилитации комбинированной деформации стоп у близнецов с детским церебральным параличом с использованием стелек с 3D-компьютерным моделированием // Вопросы практической педиатрии, 2019. Т. 14. №5. С. 119-124.

31. *Полукаров Н.В.* Стелька и способ ее получения, Патент RU (11) 2 718 262 (13) C1

32. *Rathee, S., Kashyap A.* Adaptive Miner: an efficient distributed association rule mining algorithm on Spark. Journal of Big Data (2018) 5. P. 6.

33. *Vinutha, D.C., Raju, G.T.* (2018). An Accurate and Efficient Scheduler for Hadoop MapReduce Framework. Indonesian Journal of Electrical Engineering and Computer Science. 12. pp. 1132-1142.

INTEGRATION OF LEGACY APPLICATIONS INTO "ENI" SCIENTIFIC RESEARCH ECOSYSTEM

Artem S. Trunov, *Moscow Technical University of Communication and Informatics, Moscow, Russia,* **greek17@yandex.ru**
Vyacheslav I. Voronov, *Moscow Technical University of Communication and Informatics, Moscow, Russia,* **Vorvi@mail.ru**
Lilia I. Voronova, *Moscow Technical University of Communication and Informatics, Moscow, Russia,* **Voronova.lilia@ya.ru**

Abstract
The article describes the development of a digital platform for the Eni research ecosystem, based on a microservice approach that automatically scales the computing resources of the system when working with big data. A brief description of the structure of the digital platform, its main components and the functionality of the basic subsystems is provided. Two types of digital platform consumers are presented - legacy and platform applications. Their capabilities and limitations are shown. The subsystem of distributed computing is described, which provides continuous management and monitoring of the microservice architecture of the platform, in particular, it is responsible for: load balancing, service discovery, system recovery after failures, end-to-end authentication, "canary rollouts", access control. The high-performance computing subsystem is presented. It includes models and methods for organizing parallel calculations on various hardware devices, such as a multiprocessor system, a cluster of computing devices connected by a local network, and a graphics processor. As software solutions for organizing parallel computing, multithreaded data processing technologies, MPI messaging interface, and CUDA technology are used. Also described is a data mining subsystem designed to deploy different types of neural networks with different architectures, including direct distribution, convolutional, recurrent and generative neural networks. The integration of the legacy application MD-SLAG-MELT v13.0 into the ecosystem of scientific research using platform integration adapters is presented. The architectures and the main components of the original software package and the integrated software package MD-SLAG-MELT v14.0 are presented. The results of load testing with the analysis of performance metrics and response time of an inherited application, when processing big data, are presented.

Keywords: *ecosystem, high performance computing, distributed computing, microservice architecture...*

References

1. Azure API for FHIR and our expanding ecosystem of partners - https://azure.microsoft.com/ru-ru/updates/azure-api-for-fhir-and-our-growing-ecosystem-of-partners.
2. Sberbank Annual Report 2019 https://www.sberbank.com/common/img/uploaded/files/pdf/yrep/sberbank-ar19-rus.pdf.
3. Report of the President of Russia at the SPIEF 2017 http://kremlin.ru/events/president/news/54667.
4. Decree of the President of the Russian Federation of 05.09.2017 No. 203 On the Strategy for the Development of the Information Society in the Russian Federation for 2017-2030.
5. Program "Digital Economy of the Russian Federation" http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf.
6. X5RetailGroup creates its ecosystem. https://retail-life.ru/x5-sozdast-cifrovuju-jekosistemu-ritejla-v-odnoj-knopke-za-15-let.
7. David P. Anderson, Carl Christensen, Bruce Allen. Designing a Runtime System for Volunteer Computing.
8. Trunov A.S., Voronova L.I., Voronov V.I., Sukhachev D.I., Strelnikov V.G. (2018). Legacy applications model integration to support scientific experiment. *2018 Systems of Signals Generating and Processing in the Field of on Board Communications*. P. 8350641.
9. Voronova L.I., Grigoryeva M.A., Voronov V.I., Trunov A.S. The software package "MD SLAG-MELT" Information Research System "Slag Melts" version 10.0. Deposited manuscript No. 29-B2012 01/26/2012.
10. Bezumnov D.N., Voronova L.I., Varganova E.E. A program for automating the process of prototyping distributed systems for collecting and processing information. Certificate of registration of a computer program RU 2019610043, 09/09/2019. Application No. 2018664390 dated 12/13/2018.
11. Pilipchak P.E., Voronova L.I., Trunov A.S. (2014). Load balancing model for parallel calculation of an n-particle system on a heterogeneous cluster system. *Modern high technology*. No. 5-2. P. 218-219.
12. Trunov A.S., Sukhachev D.I., Voronova L.I., Strelnikov V.G. Legacy application integration program to support high-performance computing. Certificate of registration of a computer program RU 2019610042, 09/09/2019. Application No. 2018664391 dated 12/13/2018.
13. Trunov A.S., Filatov A.M., Voronova L.I., Usachev V.A. The program of intelligent automatic distribution of computing load in the hardware-software complex for high-performance computing. Certificate of registration of a computer program RU 2018666858, 12/21/2018. Application No. 2018664407 dated 12/13/2018.
14. Voronova L.I., Trunov A.S. (2011). Optimization of the parallel algorithm of the subsystem of distributed molecular dynamics modeling. *Intersectoral information service*. No. 3. P. 3-11.
15. Voronova L.I., Trunov A.S., Voronov V.I. (2013). Development of parallel calculation methods for a correlated multiparticle system on a graphics processor. Bulletin of the Russian State Humanitarian University. Series: Documentation and Archival Studies. *Computer science. Information protection and information security*. No. 14 (115). P. 236-247.

16. Voronov V.I., Voronova L.I., Genchel K.V. (2018). The use of parallel algorithms in a neural network for recognition of sign language. In the collection: Actual problems of information and telecommunications in science and education *(APINO 2018) VII International Scientific-Technical and Scientific-Methodological Conference*. Collection of scientific articles. In 4 volumes. Edited by S.V. Bachevsky. P. 207-212.

17. Usachev V.A., Voronova L.I., Voronov V.I., Zharov I.A., Strelnikov V.G. (2019). Neural network using to analyze the results of environmental monitoring of water. *2019 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2019*. P. 8706733.

18. Voronov V.I., Zharov I.A., Bykov A.D., Trunov A.S., Voronova L.I. (2020). Designing a neural network identification subsystem in the hardware-software complex of face recognition. *T-Comm*. Vol.14. No. 5, ISSN 2072-8743

19. Zharov I.A., Voronov V.I., Voronova L.I., Bykov A.D., Vovik A.G., Usachev V.A. A program for identifying images in the neural network face recognition software package. Certificate of registration of a computer program RU 2020612466, 02.25.2020. Application No. 2020611422 of 02/11/2020.

20. Ezhov A.A., Voronova L.I., Voronov V.I., Goncharenko A.A., Artemov M.D. A program for supporting non-verbal communication using machine learning based on a convolutional neural network. Certificate of registration of a computer program RU 2019610179, 01/09/2019. Application No. 2018664377 dated 12/13/2018.

21. Goncharenko A., Voronova L., Artemov M., Voronov V., Bezumnov D. (2019). Sign language recognition information system development using wireless technologies for people with hearing impairments. *Conference of Open Innovations Association, FRUCT*. No. 24. P. 104-109.

22. Brus V.R., Voronova L.I., Voronov V.I. (2020). Neural network classification of cardiac activity based on cardiogram data. *2020 Systems of signals generating and processing in the field of on board communications*" (IEEE Conference # 48371) 19-20 March 2020.

23. Van Yue., Voronova L.I. (2019). Classification of the type of spinal disease using a neural network. *TECHNOLOGIES OF THE INFORMATION SOCIETY Materials of the XIII International Industrial Scientific and Technical Conference*. P. 407-410.

24. Bashirov A.N., Voronov V.I. (2019). Predicting the likelihood of bronchial asthma in children using the random forest algorithm. *Modern high technology*. No 12-2. P. 249-255.

25. Introducing Istio Service Mesh for Microservices by Christian Posta and Burr Sutter.

26. Voronova L.I., Grigoryeva M.A., Voronov V.I., Trunov A.S. (2013). The md-slag-melt software package for modeling the nanostructure and properties of multicomponent melts. *Melts*. No. 4. P. 36-49.

27. Bykov A.D., Voronov V.I., Voronova L.I., Zharov I.A., Fadeev M.A., Usachev V.A. A program for detecting and tracking faces in a video stream in a hardware-software complex of biometric identification based on neural network recognition of faces. Certificate of registration of a computer program RU 2020612484, 02.25.2020. Application No. 2020610985 dated 03.02.2020.

28. Voronov V.I., Bykov A.D., Voronova L.I. (2020). Designing a face detection subsystem and a database interface in a hardware-software complex of biometric identification based on neural network face recognition. *T-Comm*. Vol. 14. No. 4, ISSN 2072-8743

29. Goncharenko A.A., Voronova L.I., Voronov V.I., Ezhov A.A., Artemov M.D. Software package for data management in the information and communication system of social accessibility for people with hearing impairments. Certificate of registration of a computer program RU 2019610962, 01/18/2019. Application No. 2018665275 dated 12/26/2018

30. Achkasov E.E., Polukarov N.V., Lazareva I.A., Zhukova E.V. (2019). A personalized approach to the comprehensive rehabilitation of the combined deformity of the feet of twins with cerebral palsy using insoles with 3D computer modeling. *Issues of Practical Pediatrics*. Vol. 14, No. 5. P. 119-124.

31. Polukarov N.V. "Insole and method for its production", Patent RU (11) 2 718 262 (13) C1.

32. Rathee, S., Kashyap A. (2018). Adaptive Miner: an efficient distributed association rule mining algorithm on Spark. *Journal of Big Data* 5. P. 6.

33. Vinutha, D.C., Raju, G.T. (2018). An Accurate and Efficient Scheduler for Hadoop MapReduce Framework. *Indonesian Journal of Electrical Engineering and Computer Science*. No. 12. P. 1132-1142.

Information about authors:

Artem S. Trunov, *Moscow Technical University of Communication and Informatics, senior teacher, Moscow, Russia*

Vyacheslav I. Voronov, *Moscow Technical University of Communication and Informatics, Associate Professor of the department "Intelligent systems in control and automation", PhD in engineering, Moscow, Russia*

Lilia I. Voronova, *Moscow Technical University of Communication and Informatics, head of the department "Intelligent systems in control and automation", D.Sc. in Physical and Mathematical Sciences, Moscow, Russia*