

Цифровой проект и платформа для работы с ним*

Е. В. Биряльцев^{1,2}, М. Р. Галимов², Д. Е. Демидов^{3,4}, А. М. Елизаров^{3,4}

¹Институт прикладных проблем Академии наук Республики Татарстан

²ООО «Градиент технолоджи», г. Казань

³Казанское отделение Межведомственного суперкомпьютерного центра РАН – филиал ФГУ ФНЦ «НИИ системных исследований» РАН, г. Казань

⁴Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

Обоснована актуальность создания интернет-платформы для обеспечения промышленных предприятий, научно-исследовательских организаций, государственных органов и физических лиц эффективными инструментами и методами разработки и использования прикладного программного обеспечения (в том числе, в области проектных разработок и исследований) на основе единой цифровой платформы, базирующейся на технологиях больших данных, высокопроизводительных вычислений и многосторонних интернет-платформ. Представлен работающий прототип названной платформы, функционирующий с использованием суперкомпьютерных технологий и системы виртуализации рабочих столов.

Ключевые слова: интернет-платформа, программная платформа, облачные сервисы, прикладное программное обеспечение, разработка программного обеспечения, автоматизация бизнес-процессов.

1. Введение

В современном обществе проектная деятельность приобретает все большее значение. Этому способствуют: тенденция к гиперсегментации рынков, требующая разработки уникальных предложений для уникальной группы пользователей; возрастающая роль инновационной экономики, создающей принципиально новые товары и услуги; освоение междисциплинарных ниш, требующих комбинации технологий, ранее совместно не используемых; бизнес-подход к управлению данными (data driven approach), включающий в производственный цикл стадию изучения и восстановления закономерностей в исследуемых данных, а также другие тенденции, характеризующие отличия информационного общества от постиндустриального.

Основная доля себестоимости уникальных продуктов и услуг приходится на проектную часть, в отличие от массового продукта, где стоимость проектной деятельности распределяется на большую серию однотипных объектов, а себестоимость определяется эффективностью производства. Поэтому дальнейшее снижение себестоимости в настоящее время требует снижения себестоимости именно проектной части.

Проектная деятельность, в отличие от производственной, является преимущественно цифровой и потенциально хорошо поддается автоматизации. Вместе с тем проектная деятельность имеет существенные отличия от массового производства, и ее автоматизация должна производиться с учетом этих отличий.

Считая любой проект частным, наукоемким случаем инновации вообще, мы можем воспользоваться методологией, предложенной Чесбро в [1] и последующих работах, и констатировать, что построение эффективной проектной деятельности, замкнутой в рамках одной иерархической структуры, невозможно: для эффективной реализации проекта необходимы привлечение различных участников и организация между ними эффективного рыночного взаимодействия.

*Работа выполнена при частичной финансовой поддержке РФФИ в рамках научных проектов №№ 18-07-00964 и 18-47-160010.

Проектная деятельность, с точки зрения экономической науки, является многосторонним рынком с асимметричной информацией, что порождает высокие транзакционные издержки, барьеры и риски. Автоматизация деятельности такого рода должна быть направлена в первую очередь на уменьшение транзакционных издержек. Многосторонние рынки и рынки с асимметричной информацией являются в последнее время приоритетным предметом изучения экономической науки. Для эффективной работы на таких рынках предложены концепция многосторонней платформы и концепция обеспечения доверия на основе распределенного реестра. Важным фактором для успешного превращения объекта в рыночный товар является его коммодитизация, т. е. представление в объективном измеримом виде.

2. Используемые подходы

Отметим, что автоматизация проектной деятельности активно развивается в настоящее время в следующих направлениях:

1. *Графовые вычисления.* Представление вычислительного процесса в виде направленного ациклического графа и предоставление пользователям возможности самостоятельно, без помощи программиста, конструировать и видоизменять этот граф, в настоящее время широко применяется в таких областях, как DataMining; здесь можно отметить, например, такие системы, как RapidMiner Studio, Knime Analytics Platform и ExtractTransformLoad, а также Mule и Apache NIFI. Графовое представление обработки геофизических данных также давно применяется в нефтегазовом секторе в продуктах таких компаний, как Schlumberger, Paradigm, Roxar и ряда других. Например, компания SAP представила недавно продукт DataHub, позволяющий строить графы извлечения информации из различных источников и построения аналитических обрабатывающих графов в рамках одной оболочки. Перечисленные программные продукты можно рассматривать как средства быстрой разработки и изменения инструментов извлечения знаний и интеграции информационных систем.

2. *Средства коллективной разработки программного обеспечения.* Исторически первыми такими программами были платформы, обеспечивающие версионное хранение исходных кодов проектов (Github, Gitlab и Bitbucket). В настоящий момент они представляют собой комплексные сервисы по разработке и управлению программными проектами и, кроме хранения кода, обеспечивают командную работу, документирование проектов, интеграцию со средствами непрерывной сборки и тестирования приложений. Сегодня компания Амазон представляет аналогичный сервис AWS Step Functions для организации координации компонентов распределенных приложений и микросервисов в виде графических схем рабочих потоков. Крайне интересно выглядят два новых проекта Pallium.network и SingularityNET, которые ставят своей целью построение распределенной в интернете коммерческой сети для решения задач, связанных с машинным обучением. Предполагается, что сторонние пользователи смогут подключать свои программные решения в качестве отдельных узлов такой сети и затем предлагать их для коммерческого использования остальным участникам.

3. *Автоматизация учета интеллектуальной собственности и биллинга ее использования.* В настоящее время технологии распределенного реестра, первоначально использующиеся в криптовалютах, применяются также для учетных операций с токенизированными активами вообще и операциями с интеллектуальной собственностью в частности. Например, в упомянутых выше Pallium.network и SingularityNET для фиксации авторства используется блокчейн. Платформа Ethereum позволяет строить на ее основе умные контракты, исполняющиеся по наступлению определенных событий, что позволяет осуществлять взаимные микроплатежи за использование ресурсов в автоматическом режиме.

Предлагаемый ниже подход является интеграцией перечисленных выше подходов.

3. Предлагаемый подход

3.1 Представление цифрового проекта

Для автоматизации проектной деятельности необходимо формализовать понятие и представление проекта. Мы будем рассматривать проект как последовательность действий, приведших к некоторому результату, с целевой функцией, определяющей соответствие результата цели проекта. Хорошо известны нотации IDEF0, DFD и аналогичные, применяемые для описания после-

довательности действий вида, приведенного на Рис. 1а. В этой диаграмме под действием понимается далее любое не специфицируемое действие с материальными или цифровыми объектами, осуществляемое управляющим механизмом, под которым может пониматься автомат или человек.

Для цифрового проекта, все действия которого выполняются с цифровыми объектами, мы можем использовать аналогичный формализм (Рис. 1б), конкретизируя понятия действия как абстрактного вычислительного алгоритма, формализованного в виде λ -функции, ограничений как глобальных переменных, входящих в заданную целевую функцию, а также управляющего механизма как совокупность тела функции и ее параметров, задаваемых в процессе выполнения.

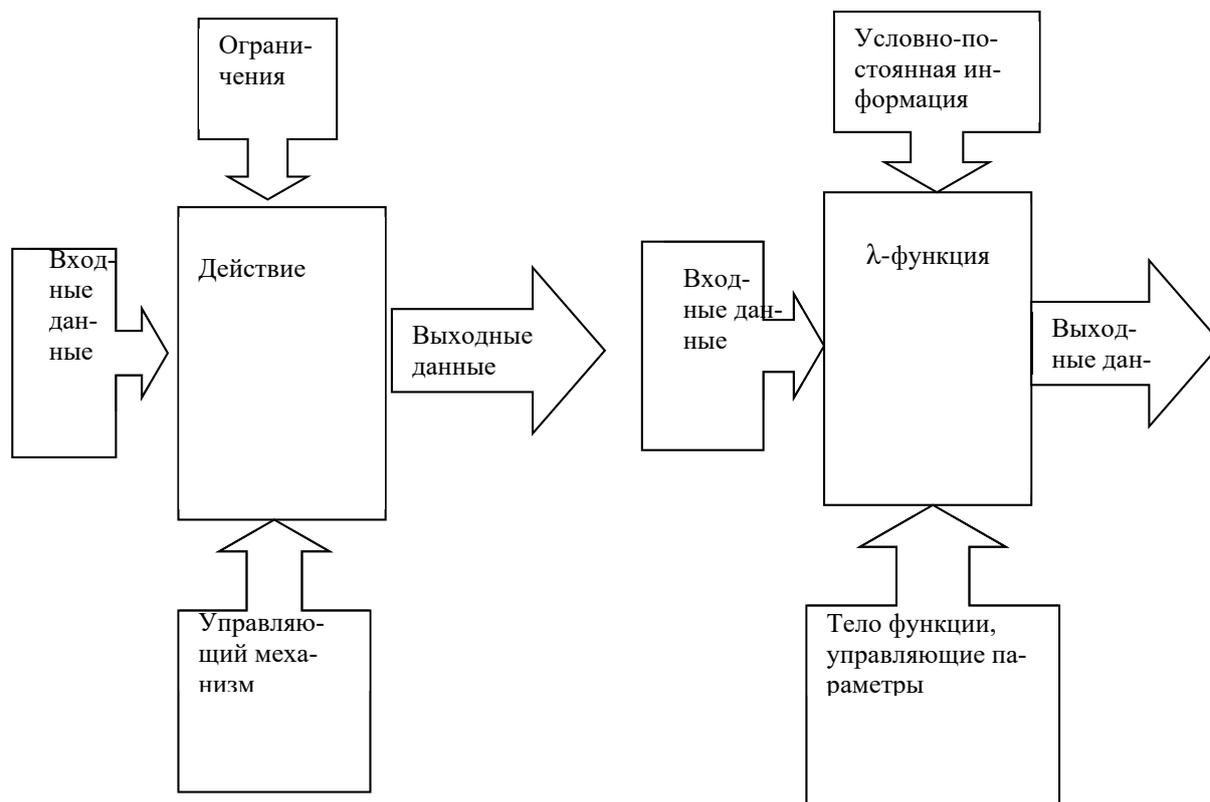


Рис. 1. IDEF0-нотация и представление проекта как λ -функции.

Аналогично методологии SADT сложные функции иерархически декомпозируются на более простые, вплоть до элементарных действий, а выходные данные предыдущих узлов могут являться для последующих узлов входными данными, ограничениями или управляющими параметрами.

Выполненный проект представляет собой конкретный набор выходных параметров, полученный в результате применения алгоритма, выбранного пользователем, в условиях имеющихся ограничений и параметров, им заданных. Ключевым отличием предлагаемого подхода от нотации IDEF0 является физическое представление проекта в виде графа, узлами которого являются исполняемый программный модуль со специфицированными управляющими параметрами, а дугами – наборы данных. Объект «Проект» физически сохраняется в таком виде вместе с данными и исполняемыми модулями и может быть повторно, полностью или частично, исполнен оболочкой, поддерживающей такое представление.

Предлагаемое представление проекта изменяет понятие «Прикладное программное обеспечение» (ППО). Традиционно ППО было организовано в виде предметно-ориентированных автоматизированных рабочих мест (АРМ), которые агрегировали известный функционал. Выполнение проекта предусматривало перемещение информации между АРМ. В предлагаемом нами подходе АРМ отсутствуют, функции могут выбраны из библиотек или предыдущих проектов или написаны пользователем и включены в граф проекта по мере необходимости. В результате любой проект является исполняемым в относительно простой оболочке, не требует наличия у поль-

зователя никаких дополнительных программных средств, что существенно упрощает его коллективное выполнение. Для повторяющихся проектов или их составных частей могут быть выделены типовые проекты, снабженные интерфейсом пользователя и соответствующей документацией.

3.2 Особенности платформы для операций с цифровыми проектами

В настоящее время признанным механизмом эффективной реализации многосторонних сделок являются платформы [2]. Предложенное выше унифицированное представление проекта частично коммодитизирует проект как предмет сделки, однако сама сделка по выполнению проекта отличается от обычных актов купли-продажи. Понимая проект как наукоемкую инновацию, можно применить саарбрюккенскую модель [3] инновации как многостадийного процесса (Рис. 2), включающего научную, инженерную, предпринимательскую стадии и бизнес-стадию.

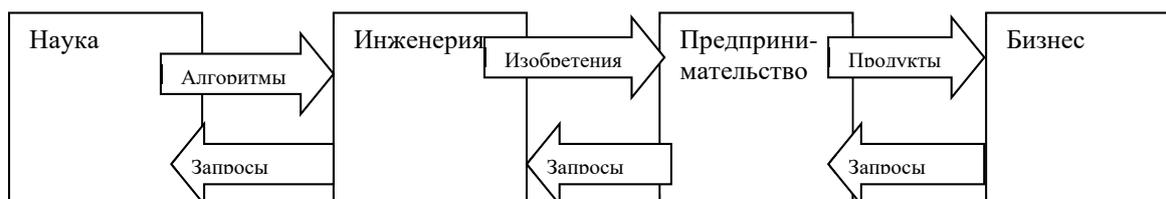


Рис. 2. Саарбрюккенская модель инновационного процесса.

Из этой диаграммы видно, что процесс выполнения проекта состоит как минимум из 3-х типов сделок (если не считать сделки внутри стадий между однотипными исполнителями), а характер этих сделок различен. Таким образом, чтобы существенно снизить транзакционные издержки, платформа должна поддерживать все три типа внутрипроектных сделок, а также финальную сделку между бизнесом и конечным потребителем проекта. Для цифрового проекта мы можем специфицировать объект сделки по стадиям следующим образом.

1. *На научной стадии* создаются цифровые модели объектов и процессов, выраженные в виде систем уравнений, а также методы решения этих систем для различных случаев. Все эти объекты выражаются в виде алгоритмических модулей, имеющих входные, выходные и управляющие параметры, а также условно-постоянную информацию. От инженерной стадии поступает запрос на модификацию модели или метода решения, в обратном направлении поступают варианты модулей. Возможна также ситуация, когда модуль создается в инициативном порядке и предлагается инженерной стадии.

2. *На инженерной стадии* создаются цепочки модулей как метод решения некоторой группы инженерных задач. Цепочка собирается из готовых или заказных модулей и может быть привязана к фронт-энду для демонстрации возможностей. Инициатива в создании изобретения может принадлежать инженерной стадии либо формироваться в виде запроса от предпринимательской стадии.

3. *На предпринимательской стадии* изобретение адаптируется к конкретному потребителю (группе потребителей) и представляет собой технологию, доступную для применения линейным специалистам. На этой стадии специфицируются источники исходной информации, при необходимости реализуются адаптеры-преобразователи этой информации из исходного вида в тот вид, который требует применяемый метод, а также определяются характеристики метода для данного потребителя и т. д. Цепочка модулей привязывается к промышленному фронт-энду для повышения юзабилити.

4. *Бизнес-стадия.* Для индустриального общества на этой стадии производится масштабирование производства и доставки товара/услуги потребителю. Для постиндустриальной стадии производство может ограничиться единственным внедрением, и созданный продукт будет уникальным. Вместе с тем созданный продукт может использоваться в более крупных проектах в виде подпроекта или являться прототипом для похожих проектов. В первом случае бизнес-стадия представлена компанией-интегратором, использующей частные цепочки для создания более крупной, во втором случае распространение может идти через партнеров в различных предметных и территориальных секторах рынка. В обоих этих случаях для повторного использования продукта может потребоваться его модификация на стадиях: предпринимательской (адаптация к

новой структуре информации, терминологии, квалификации и структуре персонала заказчика), изобретательской (изменение графа для учета качественно новой исходной информации, формирования дополнительных выходных данных) или научной (для адаптации моделей или алгоритмов).

Фронт-энд к цепочкам может быть реализован по порталной технологии. Для каждого узла разрабатывается собственное диалоговое окно, включающее визуализацию всех данных и заданные параметры обработки. Интерфейс к цепочке в целом собирается из отдельных окон с заданием правил перехода между ними.

Помимо многосторонности проектная деятельность обладает асимметрией информации на всех стадиях, представители соседних стадий плохо понимают специфические проблемы друг друга, а через стадию – не понимают вообще. В связи с этим возникает недоверие к любым результатам, представляемым одной стадией другой. Мы можем разделить эту проблему на две оставляющие: недоверие на основе опасения фальсификации и недоверие на основе сомнения в компетентности.

В последнее время проблема невозможности подделки получила решение в виде методов распределенного реестра [4]. Нетрудно заметить, что графовое представление проектов легко адаптируется к применению таких методов. Мы можем проиндексировать каждый узел его хеш-функцией, включающей хеши кода узлов, параметров, входных и выходных данных. Объемные данные при этом могут выстраиваться в иерархическую структуру хешей, известную как дерево Меркла. Так как входные данные для любого узла получаются из выходных данных предыдущих узлов, узлы получаются связанными по данным, и вся цепочка однозначно зафиксирована цепочкой хешей. Для окончательной фиксации цепочки последний хеш можно опубликовать в публичном реестре или связать со следующим случайно выбранным проектом, исключая, таким образом, возможность искажения результатов, полученных впоследствии.

Для проверки компетенции исполнителей возможно проведение внешней или внутренней экспертиз, заключающихся в частичном или полном пересчете графа. Во-первых, таким образом можно пересчитать некоторые или все результаты и убедиться, что они получены декларируемой цепочкой. Во-вторых, возможен пересчет результатов с другими параметрами обработки, для получения уверенности в субоптимальности полученного результата. Таким же образом возможно сравнение структурно различных вариантов проекта, например, для демонстрации преимущества новой структуры графа или реализации отдельных модулей.

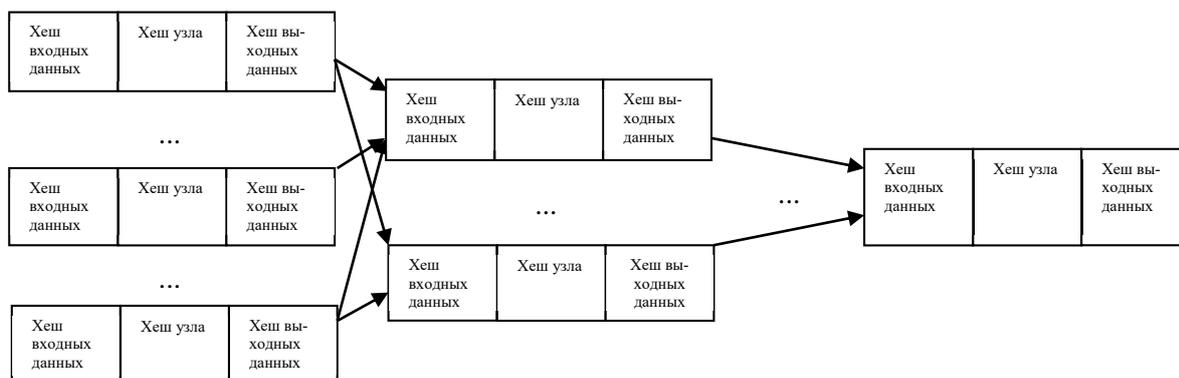


Рис. 3. Представление проекта цепочкой хеш-ссылок.

Выполненный таким образом проект может фиксировать не только финальную цепочку расчетов, но и проверенные, а также забракованные результаты, для чего необходимо зафиксировать концы всех забракованных вариантов в отдельном блоке. Промежуточные данные забракованных веток можно не хранить, а ограничиться только их хешами.

3.3. Функциональный состав платформы

Платформа как комплекс программных продуктов должна выполнять определенный обязательный набор функций (ядро) и поддерживать применение сервисов, упрощающих взаимодействие пользователей.

Узел на данном уровне абстракции представляет собой одновременно структуру данных, хранящихся на платформе, и исполняемый алгоритм в виде скрипта на интерпретируемом языке или контейнере. Узел должен быть выполнен в REST-архитектуре и быть исполняемым на любом кластере платформы.

К ядру относятся следующие функции.

1. *Управление выполнением цепочек* – осуществляется аналогом планировщика задач, который получает готовые к выполнению модули, связанные с входными и условно-постоянными данными и параметрами, и распределяет их по исполняющим узлам. Планировщик может использовать различные политики назначения исполнителя, вплоть до биржевых механизмов (отдает тому, кто меньше запросит денег). Планировщик также ведет журналирование хода выполнения работ и учет затраченных ресурсов.

2. *Управление хранением данных* – выполняет подсистема хранения, осуществляющая хранение узлов, графов, их данных в едином формате. Система поддерживает связность графов на основе частного блокчейна, обеспечивает ведение версий выполняемых и выполненных проектов, резервирование и страйпинг данных, служебные функции.

3. *Коммуникации между пользователями*. Подсистема поддерживает ведение форумов, чатов, архивов, маркетплейсов и других средств коммуникации пользователей, включая аудио-видео конференции.

4. *Администрирование пользователей и ресурсов* – управление созданием, модификацией, удалением пользователей и ресурсов, назначением им прав и лимитов.

5. *Взаимодействие системы с пользователем* (общий фронт-энд). Все предыдущие функции доступны через API и едины на платформе. Для взаимодействия с пользователями платформы предоставляется базовый интерфейс пользователя, возможно в нескольких вариантах.

Сервис представляет собой систему взаимодействующих публичных проектов, доступ к которым регламентируется правилами присоединения к сервису. Состав сервисов может наращиваться пользователями платформы, однако следующие сервисы представляются наиболее важными:

1) Поисково-рекомендательный сервис – производит поиск модулей, графов, пользователей, ресурсов, а также их агрегацию и ранжирование по релевантности запросу с учетом запросного контекста. Впоследствии может быть развит в диалоговую систему типа личного голосового помощника.

2) Билинговый сервис – на основе журналирования использования модулей и графов формирует сводные отчеты по использованным ресурсам (возможно, с тарификацией) для потребителей и владельцев ресурсов.

3) Мастера построения и отладки:

3.1) Графов и модулей. Включает в себя интерактивный билдер и тестировщик графов, редактор-отладчик текстов модулей на скриптовых языках (по типу МАТЛАБа) к процедурным узлам, конструктор SQL-запросов для логических узлов. Поддерживает ведение версий на стадии разработки-отладки;

3.2) Фронт-энда. Визуальный конструктор и тестировщик интерфейсных модулей к графам. Интерфейс к модулю является экранным плагином, предоставляющим пользователю возможность графически и текстово просматривать входные, выходные и условно-постоянные данные, а также задавать параметры исполняемого модуля. Фронт-энд может охватывать более одного узла, в этом случае пользователю предоставляются доступ ко всем данным узлов для просмотра и возможность задания параметров всех узлов. Отдельные фронт-энды могут группироваться в сценарии работы с графом. Сценариев также может быть больше одного.

4) Управления проектами. Осуществляются планирование и отслеживание хода проектов на платформе менеджерами, данные о ходе работ берутся непосредственно с платформы и не требуют ручного ввода пользователями. Может строить прогноз окончания выполнения работ и моделировать его при изменении ресурсов проекта.

В дальнейшем очевидными направлениями развития сервисов являются:

- Оптимизационный сервис: граф, для которого определена функция качества результата, можно рассматривать как функцию, требующую оптимизации; возможно построение сервиса, который будет заниматься перебором вариантов управляющих параметров с использованием алгоритмов оптимизации;

- Обучающий сервис: возможно проигрывание графа в специальном режиме, направленном на усвоение пользователем особенностей работы с этим графом;
- Сервис отчетов и публикаций, оформляющий результаты работы в человеко-читаемый вид;
- Сервис извлечения знаний: массив однотипных выполненных проектов (или подпроектов), особенно вместе с забракованными вариантами, является обучающей выборкой для извлечения знаний о том, как надо решать подобные задачи; извлеченные знания могут использоваться для оптимизационного сервиса, контроля действий пользователя или самостоятельного выполнения подпроектов искусственным интеллектом;
- Сервис взаиморасчетов – билингвоя система ограничивается сбором информации о затраченных и представленных ресурсах и формировании комплексных счетов, расчеты при этом производятся через внешние системы; учитывая надежную фиксацию всех выполненных операций, мы можем ввести расчеты во внутренней криптовалюте, в том числе, с использованием смарт-контрактов.

3.4. Прототип платформы

Согласно описанной выше концепции был разработан прототип платформы, содержащий следующие основные компоненты:

- регистрирующий узел (centralregnode);
- управляющий кластер (controlnode);
- вычислительный кластер (computecluster);
- клиентские приложения (cloudview).

Регистрационный узел – это точка регистрации различных управляющих и вычислительных кластеров для обеспечения интеграционных процедур и создания единой точки доступа. Кроме того, регистрационный узел обеспечивает учет пользователей для всех кластеров, подключенных к данной платформе, а также размещение для всех проектов общедоступных объектов.

Управляющий кластер – это программно-аппаратная инфраструктура, реализующая основные сервисы платформы и предоставляющая к ним удаленный доступ в интернет-среде. Исполнение ресурсоемких программ производится на подключаемых вычислительных кластерах.

Вычислительный кластер – это высокопроизводительная программно-аппаратная инфраструктура, предназначенная для выполнения ресурсоемких пользовательских программ (скриптов). Вычислительные кластеры могут быть как интегрированными в управляющие кластеры, так и являться кластерами общего пользования, входящими в общедоступные вычислительные сети.

Клиентские приложения – это набор программного обеспечения, который обеспечивает взаимодействие пользователя с сервисами управляющего кластера платформы. Клиентские приложения могут подключаться к любым управляющим кластерам при наличии прав у соответствующих пользователей. Для поиска доступных управляющих кластеров может использоваться регистрационный узел.

Архитектура платформы позволяет проводить гибкую настройку в зависимости от потребностей пользователей и создавать на ее основе как закрытые площадки внутри компаний, так и открытые – в интернете. Кроме того, имеется потенциальная техническая возможность интеграции различных разрозненных площадок в одну, что часто требуется при проведении сложных комплексных работ с множеством исполнителей.

Рассмотрим подробнее архитектуру управляющего кластера – основного элемента платформы. Он состоит из:

- центрального сервера приложения;
- реляционного хранилища;
- объектного хранилища;
- сервисов.

Центральный сервер приложений является единой точкой доступа к сервисам, предоставляемым данным управляющим кластером, а также обеспечивает:

- API&CLI для работы сервисов;
- среду размещения/регистрации сервисов платформы;

- организационное взаимодействие пользователей с платформой.

Реляционное хранилище предназначено для хранения служебной информации платформы, структуры и метаданных графов обработки, хранения информации сервисов, а также непосредственного хранения данных небольших объемов.

Объектное хранилище предназначено для хранения пакетов данных больших размеров. Непосредственное хранение осуществляется в специализированном key-value архиве, в качестве которого выступает база данных PostgreSQL 10. Данные хранятся в виде байтовых массивов (bytea) в записях таблиц. Для снижения нагрузки на единичные таблицы данные распределяются по группе таблиц согласно первым символам хеша (ключа) хранения. Предусмотрена возможность разбиения данных на блоки определенного размера и хранения в виде цепочки таких блоков.

Сервисы – это программные модули, развернутые на центральном сервере приложений и реализующие различные функциональные возможности управляющего кластера платформы. Сервисы взаимодействуют с основными компонентами управляющих и вычислительных кластеров платформы программным способом и предоставляют пользователю возможность оперировать функционалом платформы.

Таким образом, управляющий кластер позволяет хранить и обрабатывать данные различных типов и размеров, а также обеспечивать работу группы пользователей.

Важной особенностью платформы является возможность проведения вычислительно нагруженных и ресурсоемких работ по моделированию и обработке данных различных исследований. Для реализации этого принципа управляющий кластер прототипа был интегрирован с высокопроизводительной GPU-системой и системой удаленного VDI-доступа, программно-аппаратная архитектура которых описана в [5]. Для проведения распределенных вычислений поддерживается система управления общими ресурсами и выполнения групп задач SLURM, а для выполнения локальных задачи используется система управления NQ.

В рамках работ по формированию прототипа был реализован ряд клиентских приложений, которые обеспечивают:

- администрирование платформы;
- функционирование личного кабинета пользователя;
- управление проектом (редактор и менеджер проекта);
- коллективную работу на основе проведения дискуссий и обмена графами обработки и данными на форумах и галереях платформы.

Все клиентские приложения разработаны в виде desktop-клиента с использованием языка программирования Python 3 и технологии PyQt 5. В связи с этим для обеспечения возможности работы с данными больших размеров основная деятельность пользователей происходит удаленно на площадке VDI-кластера. Начаты исследования по разработке веб-клиента для организации взаимодействия с сервисами платформы. Для возможности расширения функциональности клиентских приложений сторонними пользователями предусмотрена система плагинов. Плагин позволяет изменять как визуальное представление приложения пользователю, так и расширять его производственные возможности.

Разработанный прототип использован для реализации прикладных пилот-проектов в различных предметных областях, некоторые из которых описаны ниже.

4. Пилотные проекты, выполненные на основе прототипа платформы

В настоящее время параллельно с разработкой прототипа платформы реализуется несколько прикладных пилот-проектов с ее использованием, что позволяет отладить функционал и юзабилити платформы во взаимодействии с реальными конечными пользователями. Ниже дано краткое описание трех проектов из различных предметных областей, использующих технологии высокопроизводительных вычислений.

4.1. Локализация микросейсмической эмиссии

Нефтегазовый сектор экономики переживает сегодня существенную трансформацию как по объектам разработки, так и методам обработки геофизической и геологической информации. По-

скольку значительная часть традиционных запасов углеводородов исчерпана, начинается активное освоение так называемых трудноизвлекаемых запасов, в число которых входят сланцевые углеводороды. Последние отличаются, в частности, отсутствием привязки залежей к куполообразным погребенным структурам-ловушкам, в связи с чем традиционные методы сейсморазведки не работоспособны для разведки сланцевых залежей. Поэтому сейчас активно разрабатываются новые методы, основанные, в частности, на локациях повышенной микросейсмической эмиссии, характерной для трещиноватых зон с повышенным скоплением углеводородов.

Технологии локализации зон повышенной микросейсмической эмиссии являются быстро развивающейся областью прикладной науки и технологии, базирующихся на высокопроизводительных вычислениях для численного моделирования распространения сейсмических волн в сложных средах и применения статистических методов большой размерности для обработки полных сигналов.

В качестве пилот-проекта на прототипе платформы был реализован алгоритм локализации микросейсмической эмиссии [6], в котором реализованы такие вычислительно нагруженные задачи, как решение динамической задачи для вязкоупругой среды методом конечных элементов с размерностью порядка сотен миллионов ячеек и методы дисперсионного анализа с размерностью ковариационных матриц порядка десятков тысяч (см. также [7, 8]). Отличительной особенностью метода является использование априорной информации в качестве регуляризатора, что требует перестройки алгоритма под различные виды априорной информации, не позволяет создать универсальный АРМ для решения данной задачи и требует корректировки программного обеспечения. Для решения этой задачи был разработан базовый граф, включающий основные алгоритмы и используемый в качестве типового решения. Платформенная реализация работы с графовыми представлениями проектов позволяет геофизику, реализующему конкретный проект, произвести реконфигурацию графа, добавляя или исключая отдельные модули, а также написать собственный модуль, учитывающий особенности проекта, и включить его в проект, не прибегая к услугам профессиональных программистов.

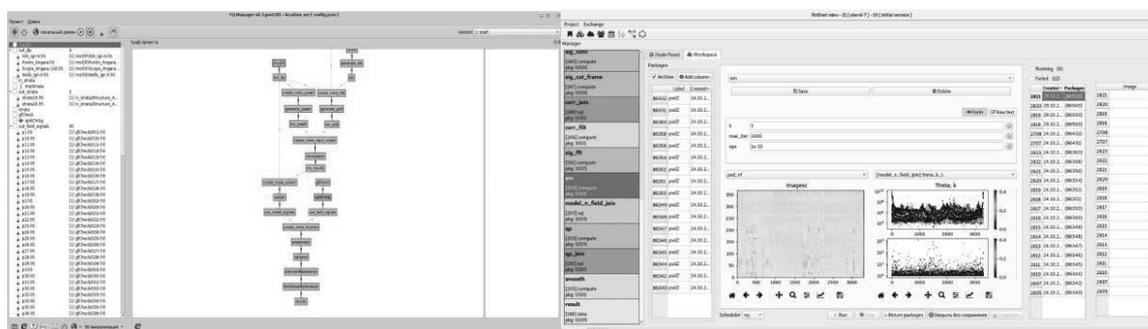


Рис. 4. Вычислительный граф и интерфейс пользователя проекта

Взаимодействие геофизика с графом производится через пользовательский интерфейс, набранный из типовых экранных элементов, позволяющих осуществлять ввод управляющих параметров и просмотр входных и выходных данных каждого узла в виде графиков, диаграмм, карт и трехмерных динамических объектов. Граф проекта и интерфейс одного из модулей представлены на Рис. 4.

4.2. Доступ к источникам информации

Развиваемый нами подход предложен в качестве основы для решения нестандартных аналитических задач Ситуационному центру(СЦ) Республики Татарстан [9]. Такие аналитические задачи отличаются необходимостью анализа заранее неизвестного набора информации в сжатые сроки программирующими пользователями – аналитиками СЦ. В качестве пилотного проекта был реализован доступ к данным сайта «Открытый Татарстан» [10], содержащего разнообразные отчеты ведомств. Этот сайт имеет API для доступа к данным, однако его формат ориентирован на выдачу отдельных файлов-отчетов по периодам в формате JSON, анализируемый период при этом выносился API в название файла в неунифицированном формате. Доступ к отчету осуществляется по его идентификатору, не несущему семантической информации о содержимом.

Таким образом, для анализа временного ряда какого-либо параметра необходимо установить идентификаторы отчетов, извлечь по идентификатору, а затем провести разбор и склейку нескольких файлов с использованием библиотек парсинга сплошного текста и формата JSON, что затруднительно для программирующих пользователей.



Рис. 5. Каталог скриптов доступа к данным и интерфейс нечеткого поиска

Для доступа к данным сайта «Открытый Татарстан» разработан скрипт на языке Python, параметризованный по идентификатору отчета, извлекающий набор отчетов и преобразующий его в реляционную таблицу. Для набора отчетов установлено соответствие идентификатора и названия отчета, скрипт доступа с соответствующим идентификатором был представлен как источник данных и помещен в галерею доступа к данным. Организован нечеткий поиск с элементами семантического анализа по названию источника данных и названиям выходных переменных, что позволяет пользователю-аналитику искать наборы данных по их смыслу. Данные возможности представлены на Рис. 5.

Найденные источники данных могут быть перенесены аналитиком в рабочую область платформы. При необходимости совместного анализа источники данных соединяются в смысле оператора JOIN по общим ключам в единую таблицу. Анализ данных статистическими методами из библиотеки статистических модулей, например, методом главных компонент, позволяет определить характер распределения объектов, выделить основные кластеры и отклоняющиеся объекты и провести другие аналитические исследования выбранных данных (Рис. 6).

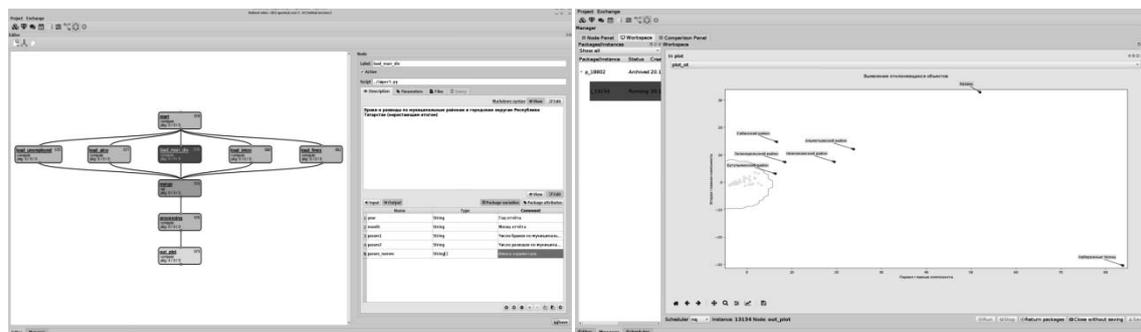


Рис. 6. Граф интеграции и обработки данных и пример интерфейса аналитика

Механизм доступа к источникам данных, реализованный на платформе, позволяет скрыть от аналитика реализацию хранения данных в конкретных источниках, разгрузив его от преобразования данных. Вместе с тем графовое представление проекта позволяет аналитику самостоятельно набрать источники данных для анализа, а также выбрать аналитический метод или их комбинацию, не прибегая к помощи прикладного программиста.

4.3. Сервис семантического поиска

Как отмечалось выше, одним из важных сервисов платформы является семантический поиск, в том числе, экспертов в узких предметных областях. В ситуации отсутствия большого количества пользователей на прототипе платформы был реализован эмулятор такого сервиса, позволяющий искать экспертов, используя аннотации их публикаций в научно-технической печати

и различные заявки на проекты. Реализованная система базируется на внешних источниках данных о научно-технической активности ученых и специалистов и состоит из двух процессов, описанных ниже. При анализе физико-математических документов применялись технологии, развитие, например, в [11–13].

Первый процесс (Рис. 7 слева) извлекает сведения о проведенных или планируемых научно-технических разработках из различных источников данных. В модельном примере в качестве источников были использованы заявки на конкурс 50 лучших идей Татарстана (около 1000 заявок) и данные о публикациях сотрудников Казанского федерального университета (около 56 тыс. публикаций). Информация извлекалась из источников, унифицировалась по формату, фильтровалась по информативности (отбрасывались неинформативные аннотации и некорректные заявки) и преобразовывалась в единую базу знаний о научно-технической активности. Такой процесс выполняется периодически администратором системы по мере изменения данных в старых источниках таких данных или добавления новых источников.

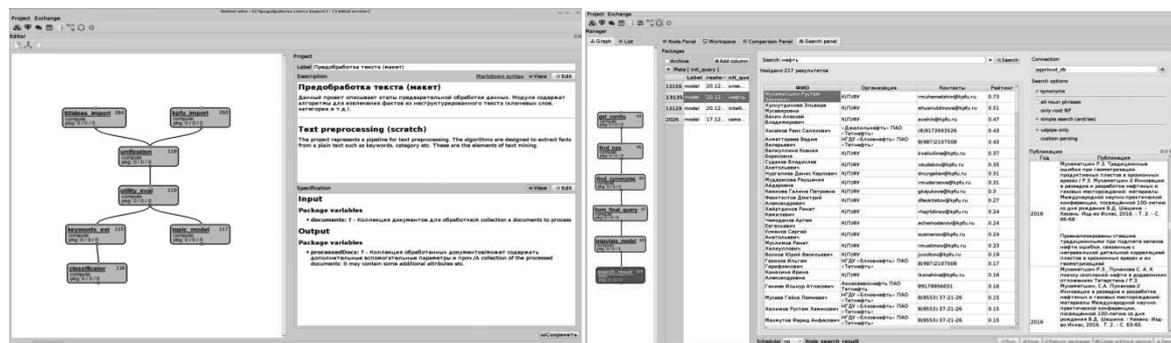


Рис. 7. Граф преобразования источников информации в базу знаний и интерфейс поиска

Второй процесс связан с обработкой пользовательских запросов. Пользователь вводит описание интересующей его предметной области в строку запроса (Рис. 7 справа), после чего система разворачивает его в набор синонимов и по всем получившимся вариантам выполняет поиск в базе знаний по аннотациям выполненных работ. Каждой найденной записи ставится в соответствие ее вес, при этом учитывается давность публикации и близость найденной формулировки к первоначально введенной пользователем. Найденные аннотации группируются по авторам, веса аннотаций суммируются и автору присваивается суммарный индекс, отражающий степень его экспертного опыта, соответствующего запросу, введенному пользователем.

Для группы экспертов, найденных описанным способом, может быть выполнен просмотр аннотаций их научно-технических проектов и статей, а также сформирован адрес электронной почты для установления контактов и дальнейшего уточнения информации.

Описанная реализация системы поиска экспертов может быть развита в двух направлениях – как пользовательская система с подключением более насыщенных источников информации, в частности, данных таких общероссийских научных информационных систем, как eLibrary, баз данных научных фондов (РФФИ, РФНФ), Роспатента и других организаций, а также как сервис внутри самой платформы, позволяющий искать авторов прикладных проектов, выполненных непосредственно на ней.

5. Заключение

Предложенная форма представления цифрового проекта и реализация прототипа интернет-платформы для работы с ним интегрирует некоторые хорошо известные тенденции, в том числе:

- изменения подходов к архитектуре программных средств, таких, как графовые представления вычислений, микросервисная архитектура, концепция DevOps;
- развития методов работы с большими данными, таких, как DataDrivenApproach и DataMining;
- концентрация усилий на минимизации транзакционных издержек за счет применения интернет-платформ и методов распределенного реестра.

При этом предложенный подход адаптирует эти концепции для любого проекта, который может быть выполнен в цифровом виде.

Опыт реализации прототипа интернет-платформы, описанной выше, и прикладных проектов с его использованием в трех различных предметных областях показал принципиальную адекватность и эффективность предлагаемого подхода.

Литература

1. Chesbrough H.W. Open Innovation: The New Imperative for Creating and Profiting from Technology. Cambridge, MA: Harvard Business School Publishing, 2003.
2. Geoffrey G. Parker, Marshall W. Van Alstyne, Sangeet Paul Choudary. Platform Revolution: How Networked Markets are Transforming the Economy and How to Make Them Work for You /W. W. Norton & Company, 2016.
3. Scheer A.-W. Start-Ups are Easy, But... [текст] Springer Science and Business Media, 2001. 220 p.
4. S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. URL: <https://bitcoin.org/bitcoin.pdf>
5. Беляева А.А., Биряльцев Е.В., Галимов М.Р., Демидов Д.Е., Елизаров А.М., Жибрик О.Н. Кластерная архитектура программно-технических средств организации высокопроизводительных систем для нефтегазовой промышленности// Программные системы: теория и приложения. 2017. Т. 8. № 1(32). С. 151–171. URL: http://psta.psiras.ru/read/psta2017_1_151-171.pdf
6. Birialtsev E.V., Demidov D.E., Mokshin E.V. Determination of moment tensor and location of microseismic events under conditions of highly correlated noise based on the maximum likelihood method// Geophysical Prospecting. 2017. V. 65, Issue 6. P. 1510–1526.
7. Biryal'tsev E.V., Galimov M.R., Elizarov A.M. Software Platform for Mass Supercomputing // Doklady Mathematics. 2017. V. 95, No. 2. P. 1–5.
8. Biryal'tsev E.V., Galimov M.R., Elizarov A.M. Workflow-Based Internet Platform for Mass Supercomputing// Lobachevskii Journal of Mathematics. 2018. V. 39, No. 5. P. 647–654.
9. Биряльцев Е.В., Минниханов Р.Н. Ситуационный центр главы региона Российской Федерации в парадигме цифровой экономики. Современные проблемы безопасности жизнедеятельности: интеллектуальные транспортные системы и ситуационные центры: материалы V Межд. науч.-практ. конф. / Под общей ред. член-корр. Академии наук Республики Татарстан, д-ра техн. наук, проф. Р.Н. Минниханова. – Казань: Центр инновационных технологий, 2018. Ч. II. С. 3–11.
10. Открытый Татарстан. Отчеты ведомств: <https://open.tatarstan.ru/reports/categories>
11. Elizarov A.M., Lipachev E.K., Nevzorova O.A., Solov'ev V.D. Methods and Means for Semantic Structuring of Electronic Mathematical Documents // Doklady Mathematics. 2014. V. 90, No 1. P. 521–524.
12. Elizarov A.M., Zhizhchenko A.B., Zhil'tsov N.G., Kirillovich A.V., Lipachev E.K. Mathematical knowledge ontologies and recommender systems for collections of documents in physics and mathematics // Doklady Mathematics. 2016. V. 93, No 2. P. 231–233.
13. Elizarov A., Kirillovich A., Lipachev E., Nevzorova O. Digital Ecosystem OntoMath: Mathematical Knowledge Analytics and Management // Communications in Computer and Information Science, vol 706. Springer, Cham, 2017. P. 33–46.