

2. Шолле Ф. Глубокое обучение на Python / Ф. Шолле. — Санкт-Петербург : Питер, 2019. — 400 с. — Текст : непосредственный.
3. TensorFlow Object Detection API / GitHub. — Текст : электронный // GitHub: Where the world builds software. — 2022 : [сайт]. — URL: [https://github.com/tensorflow/models/tree/master/research/object\\_detection](https://github.com/tensorflow/models/tree/master/research/object_detection) (дата обращения: 05.05.2022).
4. mAP (mean Average Precision) / GitHub. — Текст : электронный // GitHub: Where the world builds software. — 2022 : [сайт]. — URL: <https://github.com/Cartucho/mAP> (дата обращения: 05.05.2022).

*А. О. КОЗЛОВА, А. М. ВОРОБЬЕВ, О. А. ИВАНОВ*

*Тюменский государственный университет, г. Тюмень*

**УДК 004.052.3**

## **МОДЕЛЬ СЕРВИСА РАСПРЕДЕЛЕННЫХ EDGE-ВЫЧИСЛЕНИЙ НА ОСНОВЕ ПРОТОКОЛА AMQP**

***Аннотация.** В данной статье рассмотрена модель сервиса распределенных Edge-вычислений на основе протокола AMQP, который активно применяется в различных решениях управления в области IoT. Описано создание механизма «выгрузки» заданий на ресурсы, предоставленные акторами “shared ecomoty”, рассмотрено их взаимодействие на основе стандартизированных шин передачи сообщений.*

***Ключевые слова:** экономика общественного потребления, AMQP, Edge-вычисления, импортозамещение.*

**Введение.** Отечественная отрасль информационных технологий в 2022 г. столкнулась с внезапным и резким недостатком вычислительных мощностей, что поставило под угрозу экстенсивные компоненты планов развития профильных компаний и государственных учреждений. Недостаток серверного оборудования можно проиллюстрировать письмом руководства социальной сети “VK” в адрес Министерства цифрового развития, связи и массовых коммуникаций, в котором авторы просили оказать содействие в поиске поставщика, готового поставить «десятки тысяч серверов» для плановой замены и расширения возможностей социальной сети [1].

Тем временем многие компании и проекты, оснащенные вычислительными мощностями до произошедших событий, не испытали изменения в характере нагрузки в связи с особенностями функционирования информационной инфраструктуры — обслуживание бизнес-процесса, который не испытал существенных изменений за последнее время.

В исследованиях [3-5] предлагаются различные решения по использованию вычислительных ресурсов. Автор С. П. Воробьев в своей статье [6] рассматривает математическую модель сетевой инфраструктуры, объединяющей комплексы облачных, туманных и граничных вычислений, с учетом преимуществ технологий IoT.

Решение, предложенное в данной работе, заключается в формировании механизма создания гибридных вычислительных сред с выполнением отдельных специфических нагрузок на внешних ресурсах, введенных в систему на принципах экономики совместного потребления. Следует отметить, что принципиально допустимо сравнивать предложенную модель с концепциями “Edge Computing” [7], отличие будет лежать лишь в проблематике — скорость обработки данных в классических Edge-решениях и сам факт применения дополнительной мощности в условиях дефицита.

Основные принципы и механизмы построения подобного экспериментального решения изложены авторами данного проекта в работе [2] с описанием основных особенностей применения принципов экономики общественного потребления на предметную область.

**Постановка задачи.** Данная работа концентрируется на другом способе использования пула ресурсов экономики общественного потребления — это создание механизма «выгрузки» задания на ресурсы, предоставленные акторами “shared economy” с последующим получением результатов (рис. 1).

Аналогией подобному решению может служить использование вычислительных ресурсов видеокарты для решения задач, не связанных с отображением графики, как показано в [3] — расчет физических моделей, майнинг, обработка видео и т. п.



*Рис. 1.* Общий вид гибридного вычислительного решения

Данный подход имеет следующие отличия:

- привлеченные внешние вычислительные ресурсы ориентированы строго на исполнение поставленных клиентом задач;
- привлеченный ресурс работает исключительно в качестве отдельного модуля расширения существующей ИС;
- привлеченные ресурсы используют статически развернутую инфраструктуру передачи сообщений (данных и управления);
- использование внешней платформы для вычислений подразумевает формирование «нагрузки» — вычислительного кода, который передается на внешнюю платформу и ожидает получение данных для вычислений;
- использование внешней платформы обязывает осуществлять взаимодействие через специальный интерфейс;
- внешняя платформа имеет свои особенности функционирования, связанные с природой ее построения — далеко не все задачи могут быть эффективно исполнены средствами такой платформы.

Для функционирования данной модели применения внешних ресурсов требуется решить следующие задачи:

1. Определить возможности и ограничения, сформировать облик задачи, способной быть эффективно выполненной с применением предложенной модели.

2. Разработать метод формирования «вычислительной группы» на основе пула всех привлеченных ресурсов.

3. Разработать способ передачи и получения данных в исполнительный модуль: кодовая нагрузка для выполнения, входные данные, результаты работы.

Характеристики исполнительской среды, построенной на принципах совместного потребления, накладывают следующие ограничения на эффективность исполнения кода:

- исполняемый код имеет быстрый и эффективный доступ только к локальному хранилищу контейнера и ОЗУ;
- вычислительный узел принадлежит третьему лицу, обработка конфиденциальной информации может привести к утечкам;
- отсутствие гарантий работоспособности узла.

Исходя из изложенных ограничений, можно сформировать следующие черты задачи, эффективно решаемой в рамках предложенной модели:

- экземпляры нагрузки выполняют задачу без взаимодействия между собой;
- данные, собираемые или обрабатываемые нагрузкой на узле, не представляют особой ценности в момент обработки вне собственной ИС;
- формат обработки данных близок к пакетному — четко сформулированные задачи, минимизация взаимодействия в процессе решения задачи, выдача конечного результата в основную ИС.

Примерами подобных задач можно считать алгоритмы накопления и обработки больших данных, боты исследования СМИ и социальных сетей, программы-кодировщики или расчетное ПО.

Для реализации предлагается модель взаимодействия на основе стандартизированных шин передачи сообщений (рис. 2).

### **Функциональные элементы**

В предложенной схеме выделяются следующие функциональные элементы:

- клиентская информационная система — ИС-потребитель привлеченных вычислительных мощностей;
- вычислительный узел — ресурс, предоставленный внешним актором на условиях экономики совместного потребления;

- вычислительная нагрузка — клиентский программный код в формате docker-образа, выполняющий полезную работу на ресурсах вычислительной группы;
- менеджер — программный комплекс менеджмента привлеченных ресурсов, обрабатывает запросы на ввод и вывод ресурсов из эксплуатации, формирует вычислительные группы по запросам клиентов;
- шлюз — программный компонент, обеспечивающий взаимодействия клиента с Менеджером и созданными в процессе вычислительными группами;
- шина передачи сообщений — программный комплекс передачи сообщений, используется Шлюзом, Менеджером и Вычислительными ресурсами для команд управления и передачи данных.

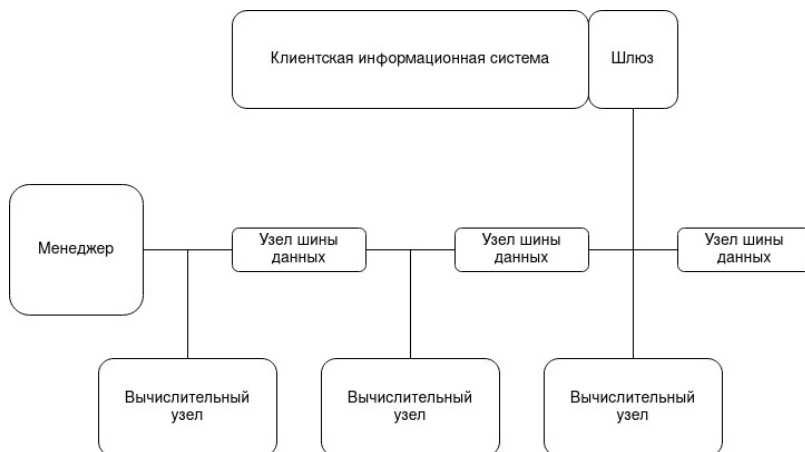


Рис. 2. Модель взаимодействия на основе шин передачи сообщений

Для организации обмена в реализации RabbitMQ используется протокол AMQP, который активно применяется в различных решениях управления в области IoT [4], где решает принципиально схожую задачу обеспечения обмена данными среди некоего пула ресурсов [5].

Ввод ресурсов актора экономики общественного потребления в указанную схему не претерпел существенных изменений с поправкой на подключение к используемой Шине передачи сообщений.

Для эффективного обслуживания запросов клиентов динамически (под конкретную задачу) формируется Вычислительная группа — группа Вычислительных узлов, объединенных в единый контур управления и направленных на обслуживание запросов конкретного клиента. Формирование Вычислительной группы инициируется со стороны потребителя путем обращения к API Шлюза:

1. В запросе клиент указывает следующие параметры:
  - HTTPS-ссылка на docker-образ вычислительной нагрузки;
  - желаемое количество вычислительных узлов в группе;
  - минимальное количество процессорных ядер и ОЗУ одного узла.
2. Шлюз обращается к Менеджеру с использованием шины обмена сообщениями GLOBAL:
  - менеджер определяет список хостов, подходящих под запрос;
  - менеджер обращается к RabbitMQ с целью формирования отдельного виртуального хоста обмена сообщениями.
3. Шлюз и Вычислительные узлы осуществляют подключение к созданному изолированному виртуальному хосту передачи сообщений согласно созданным учетным данным.
4. Шлюз осуществляет рассылку адреса для скачивания docker-образа нагрузки на вычислительные узлы.
5. Вычислительные узлы запускают образы вычислительной нагрузки, система готова к применению.

Для решения последней поставленной задачи — механизмы передачи пользовательских сообщений от клиентской ИС до клиентской нагрузки, развернутой на вычислительных узлах — предлагается следующая модель взаимодействия (рис. 3):

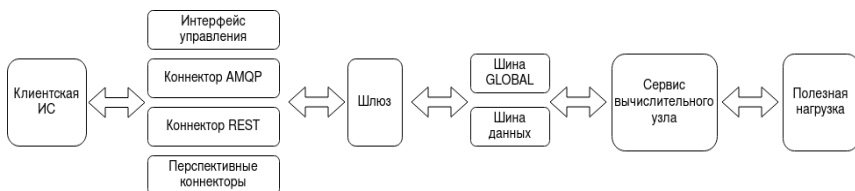


Рис. 3. Конвейер передачи команд

Ключевые компоненты модели характеризуются следующими элементами:

- API управления — классический REST интерфейс управления — отвечает за подключение к системе, создание Вычислительных Групп и т. п.
- Коннектор — способ взаимодействия клиентской ИС и полезной нагрузки, развернутой на Вычислительных узлах.
- Шлюз — выполняет задачу преобразования данных, подаваемых через Коннекторы различного типа к стандарту сообщений AMQP.
- Сервис вычислительного узла — осуществляет получение сообщений из выделенного виртуального хоста и передачу их на обработку полезной нагрузке, реализованной в виде docker-контейнера на Вычислительном узле.
- Транслятор — программная конструкция, осуществляющая конвертацию данных, полученных по шине AMQP в выбранный разработчиком полезной нагрузки интерфейс передачи данных.

Таким образом, двунаправленная коммуникация между клиентской ИС и полезной нагрузкой на Вычислительных узлах осуществляется с применением шины данных, развернутой в рамках изолированного виртуального хоста RabbitMQ. Выбранное решение позволяет проявить гибкость используемых средств взаимодействия за счет заменяемых коннекторов и трансляторов, применение которых превращает Шлюзы и Сервисы вычислительного узла в прокси-узлы для передаваемых данных.

**Заключение.** В результате исследования была предложена модель организации гибридной обработки данных с применением внешних вычислительных ресурсов, предоставленных на условиях экономики совместного потребления. Разработан алгоритм формирования вычислительных групп под требование заказчика, проработан вопрос организации связи между постановщиком задач и исполнительными модулями системы.

В дальнейшем планируется уточнить вопросы управления группами вычислительных узлов для реализации наиболее популярных

паттернов построения информационных систем и наименее затратной интеграции предложенного решения в существующие ИС.

### **Благодарности**

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-47-720006 «Исследование подходов и реализация технологий сбора, передачи, обработки и хранения данных для масштабируемой вычислительной облачной платформы на принципах “Экономики совместного потребления”».

### **СПИСОК ЛИТЕРАТУРЫ**

1. VK попросила Минцифры найти серверы на фоне дефицита импортной техники. — Текст : электронный // Ведомости : еженедельное интернет-издание 2022. 25 апреля : [сайт]. — URL: <https://www.vedomosti.ru/technology/news/2022/04/25/919548-problemah-it-kompanii-serverov> (дата обращения: 12.05.2022).
2. Воробьев А. М. Экспериментальная реализация гибридной модели облачного сервиса на принципах экономики совместного потребления / А. М. Воробьев [и др.]. — Текст : непосредственный // Инженерный вестник Дона. — 2021. — № 9 (81). — С. 135-152.
3. Yamato Y. Automatic GPU offloading technology for open IoT environment / Y. Yamato [et al.]. — Text : direct // IEEE Internet of Things Journal. — 2018. — Vol. 6, № 2. — P. 2669-2678.
4. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP / N. Naik. — Text : direct // IEEE international systems engineering symposium (ISSE). — 2017. — P. 1-7.
5. Bhimani P. Message delivery guarantee and status update of clients based on IOT-AMQP / P. Bhimani, G. Panchal. — Text : direct // Intelligent Communication and Computational Technologies. — Springer, 2018. — P. 15-22.
6. Воробьев С. П. Математическая модель оптимизации сетевой инфраструктуры распределенной корпоративной системы на базе облачных, туманных и граничных технологий / С. П. Воробьев. — Текст : непосредственный // Моделирование, оптимизация и информационные технологии. — 2019. — Т. 7, № 3. — С. 4-14.
7. Shi W. Edge computing: Vision and challenges / W. Shi [et al.]. — Text : direct // IEEE internet of things journal. — 2016. — Vol. 3, № 5. — P. 637-646.