

МИКРО-ПОТОКИ РАБОТ: СОЧЕТАНИЕ ПОТОКОВ РАБОТ И ПОТОКОВОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ ПОДДЕРЖКИ ЦИФРОВЫХ ДВОЙНИКОВ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

© 2019 А.Б.А. Алаасам¹, Г.И. Радченко¹, А.Н. Черных^{1,2}

¹Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76),

²Научно-исследовательский центр Энсенады

(22860 Энсенада, Мексика, Carretera Ensenada - Tijuana No. 3918)

E-mail: alaasatab@susu.ru, gleb.radchenko@susu.ru, chernykh@cicese.mx

Поступила в редакцию: 12.11.2019

В последнее время наблюдается взрывной рост в развитии концепции цифровой индустрии. Одним из важнейших элементов этой концепции является применение методов математического моделирования и интеллектуального анализа данных для создания моделей производственных процессов и конечной продукции, базирующихся на обработке сигналов, поступающих с интеллектуальных сенсоров. Совокупность таких моделей, представляющих собой виртуальное представление промышленных процессов, систем и оборудования называют цифровыми двойниками. Цифровые двойники используют данные, получаемые от сенсоров, установленных на производственных линиях или на базе конечной продукции, для прогнозирования сбоев в работе оборудования, оптимизации качества продукции и сокращения негативного воздействия производственных процессов на окружающую среду. Комплексы моделей, лежащие в основе цифровых двойников, могут быть описаны в виде вычислительных потоков работ (Workflow), состоящих из набора вычислительных сервисов, каждый из которых представляет собой модель одного из этапов технологического процесса. Для организации гибкой поддержки облачных вычислений для выполнения цифровых двойников, мы предлагаем концепцию микро-потоков работ (Micro-Workflows), которая сочетает в себе мощь концепции научных потоков работ (Scientific Workflows), гибкость контейнерных технологий и устойчивость подхода потоковой обработки данных (Stream Processing) в распределенных вычислительных системах.

Ключевые слова: цифровые двойники, микросервисы, контейнеризация, микро-потоки работ, Kafka, Kessler.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Алаасам А.Б.А., Радченко Г.И., Черных А.Н. Микро-потоки работ: сочетание потоков работ и потоковой обработки данных для поддержки цифровых двойников технологических процессов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 4. С. 100–116. DOI: 10.14529/cmse190407.

Введение

Концепция цифровой индустрии подразумевает организацию производственных процессов с использованием интеллектуальных программных платформ, обеспечивающих сбор, хранение и обработку данных с сенсорных сетей, с использованием методов интеллектуального анализа данных и машинного обучения, для повышения эффективности производства [8]. Среди инициатив по развитию этой концепции можно выделить программу «Индустрия 4.0» [20], разработанную в Германии, «Коалицию лидеров умного производства» [6] в США и концепцию промышленного Интернета вещей, предложенную корпорацией General Electric [22].

Одним из ключевых подходов, объединяющих эти программы, является концепция цифровых двойников (ЦД), которая обеспечивает создание и поддержку виртуальных моделей реального оборудования, промышленных процессов и конечной продукции. Согласно [14], ЦД — это интегрированная мульти-физическая, мульти-масштабная вероятностная симуляция сложного изделия, которая использует наиболее подходящие физические модели, актуальные данные сенсоров и др. для того чтобы получить как можно более достоверное представление соответствующего реального объекта.

ЦД используют различные подходы для моделирования реальных объектов и технологических процессов, включая методы статистического и интеллектуального анализа данных, методы вычислительного моделирования, такие как метод конечных элементов и т.д. [18]. Каждый из этих методов предъявляет особые требования к необходимым вычислительным ресурсам. Например, методы интеллектуального анализа данных требуют хранилищ большого объема с высокой пропускной способностью для сбора и получения доступа к аналитическим данным, а также высокой масштабируемости вычислительной системы для их обработки; для применения методов машинного обучения требуются узлы с установленными ускорителями а для моделей, использующей метод конечных элементов, требуется высокая производительность центрального процессора и большие объемы оперативной памяти [1].

Предоставление комплекса вычислительных систем, обеспечивающего возможности высокого масштабирования, а также соответствующего столь разнообразным требованиям к вычислительным ресурсам может быть реализовано посредством применения технологий облачных вычислений [13]. В настоящее время, облачные вычислительные системы стали повсеместно применяться для выполнения задач компьютерного моделирования, предоставляя доступ к большому разнообразию вычислительных ресурсов, используя, в основном, технологию виртуальных машин. Но применение виртуальных машин связано с большими накладными расходами, которые могут значительно ограничить производительность систем ввода/вывода и эффективность использования вычислительных ресурсов [17, 32]. Технология контейнеризации, получившая значительное развитие в последнее время, основана на ограничении количества ресурсов вычислительного узла, отдаваемых приложению. Контейнер предоставляет среду выполнения приложения на уровне операционной системы, сокращая накладные расходы по сравнению с виртуальной машиной [4].

Для описания технологических процессов часто применяется концепция потоков работ (Workflows) [30]. Потоки работ применяются в промышленности, коммерческих компаниях и науке как способ формализации и структурирования процессов, состоящих из совокупности независимых действий, связанных между собой отношениями зависимости. Вычислительный поток работ предоставляет эффективное решение для моделирования сложных технологических процессов путем управления сервисами и оптимизации порядка их выполнения [26]. В связи с этим, цифровой двойник промышленного процесса или оборудования, может быть представлен в виде вычислительного потока работ, состоящего из совокупности вычислительных сервисов, которые представляют собой модели эталов технологического процесса.

Для того, чтобы цифровой двойник мог обеспечить синхронизацию между состоянием процесса в реальном мире и его виртуальной копией [5], мы должны обеспечить ему воз-

возможность получать, передавать и анализировать поток данных от оборудования промышленного Интернета вещей (IIoT) в режиме реального времени. Для решения этой задачи могут использоваться платформы потоковой обработки данных (Stream Processing), обеспечивающие слабосвязанность программной архитектуры, гибкость и контроль над сложными процессами, завязанными на асинхронной обработке данных [7].

Еще одной важной концепцией, сформировавшейся в последнее время, является *микросервисный подход* проектирования приложений, реализуемых на базе облачных вычислительных платформ. В рамках микросервисного подхода, серверная часть приложения разбивается на отдельные, изолированные компоненты — микросервисы, обеспечивающие прозрачный доступ к своим функциональным возможностям и взаимодействующим друг с другом посредством кроссплатформенного API. Эти компоненты могут быть развернуты и масштабироваться независимо друг от друга в автоматическом режиме [21]. Такой подход к проектированию и реализации распределенных систем, части которых подвергаются разной нагрузке, либо должны обеспечивать различные характеристики предоставляемого качества обслуживания, позволяет обеспечить независимое развертывание, масштабирование и управление временем жизни их компонентов [33].

В данной работе нами будет дано описание концепции цифровых двойников для моделирования промышленных процессов. Также, нами будет представлена модель событийно-ориентированного подхода, совмещающего возможности научных потоков работ, гибкость контейнерных технологий и устойчивость подхода потоковой обработки данных в распределенной вычислительной системе. Нами будет представлен пример реализации данного подхода, посредством разработки акторов, обеспечивающих потребление и отправку потоков данных в платформу Apache Kafka из системы управления научными потоками работ Kepler. Данный подход обеспечивает возможность формирования потока работ как совокупности слабосвязанных сервисов (микро-потоков работ или Micro-Workflows, по аналогии с микросервисами). Такие сервисы могут быть развернуты внутри вычислительных контейнеров и обеспечивать возможность коммуникации через платформу потоковой обработки данных.

Дальнейшая статья имеет следующую структуру. В разделе 2 нами дается обзор существующих подходов к облачным системам, поддерживающим концепцию цифровых двойников. В разделе 3 описывается концепция цифровых двойников, позволяющая осуществлять моделирование реальных производственных процессов. В разделе 4 представлен обзор концепции облачной платформы цифровых двойников. В разделе 5 мы даем описание подхода микро-потоков работ для проектирования цифровых двойников. В разделе 6 представлен процесс разработки микро-потоков работ и реализация эксперимента. В разделе 7 приводятся результаты оценки эффективности микро-потоков работ. В заключении представлены обобщенные результаты работы и описаны возможные направления дальнейших исследований.

1. Обзор смежных работ

На сегодняшний день одной из наиболее распространенных концепций, связанных с платформами цифровых двойников, являются облачные системы Интернета вещей (IIoT). Они обеспечивают сбор данных с различных датчиков и проведение их анализа. Авторы статьи [23] дают описание модели Sensing as a service (Senaas, «Детектирование как сервис»), состоящей из четырех концептуальных слоев: 1) датчики и владельцы датчиков, 2)

публикаторы датчиков, 3) поставщики расширенных сервисов и 4) потребители данных от датчиков. В статье приводится описание нескольких облачных систем для сбора данных от датчиков. Например, Xively [27] — это общедоступная облачная среда для Интернета вещей, которая упрощает и ускоряет создание, развертывание и масштабирование систем датчиков. В рамках проекта OpenIoT [28] разрабатывается платформа с открытым исходным кодом, обеспечивающая динамическое формирование самоуправляемых облачных сред для Интернета вещей. Авторы платформы IoT-Hub [11] предлагают решение для сбора, проверки качества, хранения и визуализации данных от датчиков Интернета вещей на основе платформы Apache Spark¹ в качестве механизма обработки данных.

Цифровой двойник может быть представлен как последовательность заданий по обработке данных, связанных между собой набором ребер, представляющих зависимости по данным между этими заданиями. Такой подход может быть реализован в виде научного потока работ (Scientific Workflow или SWF), который обычно управляется, выполняется и отслеживается системами управления научными потоками работ (Scientific Workflows Management Systems — SWfMS), такими как DiVTB [25], Pegasus [9], Kepler [3], ASKALON [10], или tGSF [15, 16]. SWfMS позволяет ученым сконцентрироваться на решении исследовательских задач, обеспечивая автоматизацию процесса проведения сложных вычислительных экспериментов. Авторы статьи [12] предлагают среду для отображения вычислительных потоков работ, написанных на языке Dispel, на другие платформы такие как Apache Storm² и MPI.

Однако выполнение набора SWF, характеризующихся сильно-связанностью вычислительных сервисов, последовательным выполнением и ограниченной поддержкой потоковой обработки данных не соответствует событийному подходу, необходимому для работы с цифровыми двойниками. В нашей работе мы предлагаем решение для интеграции вычислительных потоков работ с концепцией микросервисов с целью создания слабосвязанной событийно-ориентированной инфраструктуры для разработки и выполнения цифровых двойников.

2. Цифровой двойник

Концепция цифровых двойников (см. рис. 1) была представлена в начале 2000-х годов. Она основывается на подходе синхронизации реально существующих объектов с их виртуальными представлениями в виде вычислительных моделей [29]. Цифровой двойник — это иерархическая система математических моделей, вычислительных методов и сервисов программного обеспечения, которая предоставляет близкую к реальному времени синхронизацию между состоянием реального процесса или системы и его/ее виртуальной копией [5].

Синхронизация в цифровом двойнике математических моделей (мультифизических, многомасштабных и вероятностных) с реальными данными позволяет оценивать эффективность работы оборудования, получать информацию о состоянии процесса или конечной продукции и прогнозировать их характеристики, используя информацию из архивной базы данных.

Можно выделить следующие характеристики цифровых двойников:

¹ <https://spark.apache.org/>

² <https://storm.apache.org/>

- *Отражение в режиме реального времени.* Цифровой двойник отображает физический объект (оборудование, процесс и систему) в режиме реального времени, обеспечивая соответствие виртуальной копии реально существующему объекту.
- *Иерархия.* Цифровой двойник промышленного процесса может состоять из цифровых двойников этапов процесса, которые, в свою очередь, могут содержать цифровые двойники оборудования, на котором осуществляются данные этапы.
- *Саморазвитие.* Так как цифровой двойник является отображением объекта или процесса, то он должен постоянно совершенствовать модель путем сравнения процесса моделирования одновременно с физическим объектом и процессом [31].

Цифровые двойники могут использовать различные типы моделей для моделирования текущего состояния реального объекта. Полное и своевременное отображение состояния можно получить только путем использования моделей с необходимой и достаточной степенью точности для соответствия требованию «отображения в режиме реального времени». Чем большая точность требуется, тем больше необходимо ресурсов и времени для выполнения моделирования. В связи с этим можно выделить следующие типы моделей, используемые в цифровых двойниках:

- *Сценарные модели* применяются для уведомления конечных пользователей о событиях, в случае если параметры технологического процесса превышают установленные пределы. Низкая вычислительная сложность таких моделей позволяет внедрять их непосредственно на оборудовании, установленном на промышленном участке. В то же время необходимо выделять вычислительные ресурсы для фильтрации поступающих сообщений и направления уведомлений заинтересованным исполнителям.
- *Первопринципные модели,* основанные на основных принципах термодинамических, физических и химических моделей, на которых базируется работа конкретного оборудования. Эти модели позволяют прогнозировать ухудшение характеристик технологических процессов путем учета текущей динамики, а также предсказывать дату и время для проведения очередного технического обслуживания оборудования по фактическому состоянию его эффективности [18].
- *Конечно-элементные вычислительные модели* дают возможность моделирования сложных многофакторных процессов путем решения систем дифференциальных

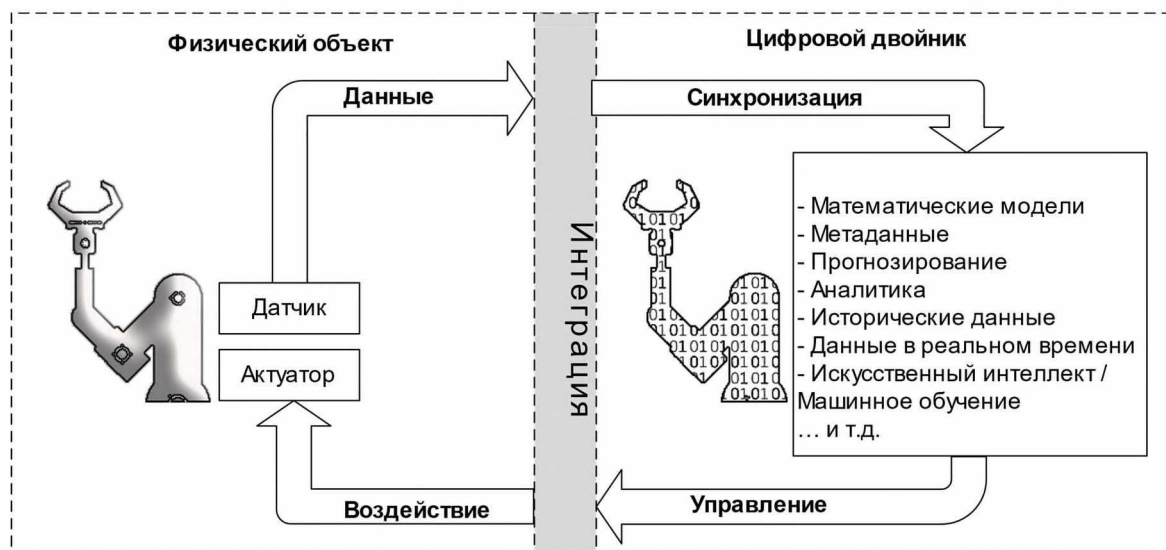


Рис. 1. Концепция цифрового двойника

уравнений с заданными граничными условиями. Они используются для проектирования новых продуктов, моделирования новых технологических циклов и моделирования чрезвычайных ситуаций. Решение задач этого класса требует специального вычислительного оборудования.

- *Методы интеллектуального анализа данных* используются для решения задач, для которых не подходит ни один из вышеперечисленных методов ввиду сложности, многофакторности или вероятностных особенностей происходящих процессов. Эти методы требуют больших объемов данных хорошего качества (предварительно очищенных) различной природы происхождения.

3. Облачная платформа для поддержки цифровых двойников

Для поддержки работы цифровых двойников нами предлагается концепция облачной платформы для поддержки цифровых двойников, которая позволяет осуществить динамическое распределение вычислительных ресурсов и предоставляет прикладной программный интерфейс для представления цифровых двойников на базе микросервисного подхода. Таким образом, облачная платформа цифровых двойников предоставляет облачную модель Digital Twin-as-a-Service (DTaaS, «Цифровой двойник как услуга»). В рамках модели DTaaS, цифровой двойник представляется как совокупность облачных сервисов, обеспечивающих хранение и анализ данных, получаемых от датчиков, для моделирования реально существующих объектов и визуализации их виртуальных представлений.

Облачная платформа цифровых двойников предоставляет единую точку входа для следующих ключевых категорий пользователей (см. рис. 2):

- *Производители промышленного оборудования* могут создавать цифровые двойники поставляемого оборудования для потребителей, осуществлять удаленный мониторинг и анализ использования поставленного оборудования в целях повышения качества продукции.



Рис. 2. Облачная платформа цифровых двойников

- *Промышленные предприятия* могут создавать цифровые двойники промышленных процессов, обеспечивающих анализ и принятие решений для развития предприятия.
- *Исследовательские центры* могут разрабатывать и продвигать свои аналитические модели для цифровых двойников, а также предоставлять услуги по поддержке и повышению эффективности промышленных процессов и систем.

Облачная платформа цифровых двойников обеспечивает функционирование на следующих уровнях:

- *Уровень пользователя цифрового двойника.* На этом уровне пользователь может получить доступ к имеющимся цифровым двойникам в виде облачных приложений на основе модели Software-as-a-Service («Программное обеспечение как услуга»). Пользовательский интерфейс цифрового двойника может быть представлен конечному пользователю в различном виде: от трехмерных моделей и мнемонических схем до табличных данных и текстовых сообщений, описывающих выбранные аспекты моделируемого реального объекта для конкретных типов пользователей. Например, инженер может использовать цифровой двойник газового котла в виде трехмерного объекта в дополненной реальности для обнаружения закупорки в печи. Тот же цифровой двойник может быть представлен в виде графика снижения эффективности работы котла и информации о прогнозируемой оптимальной дате проведения ТО для экономиста.
- *Уровень разработчика цифрового двойника.* На этом уровне облачная платформа предоставляет ресурсы для разработки цифрового двойника на основе модели Platform-as-a-Service («Платформа как услуга»). Цифровой двойник может быть описан как совокупность вычислительных сервисов, осуществляющих связь друг с другом посредством передачи сообщений.
- *Уровень разработчика вычислительного сервиса.* На этом уровне облачная платформа предоставляет прикладной программный интерфейс для разработки вычислительного сервиса на основе модели Backend-as-a-Service («Серверное приложение как услуга»). Вычислительный сервис — это микросервис, отвечающий за конкретные операции обработки данных или за выполнение конкретного набора вычислительных методов, обеспечивающих моделирование определенного аспекта цифрового двойника.
- *Уровень поставщика облачной инфраструктуры.* На этом уровне экземпляры вычислительных сервисов сопоставляются с ресурсами облачной вычислительной платформы, предоставляемыми на основе модели Container-as-a-Service («Контейнер как услуга»).

4. Подход микро-потоков работ для проектирования цифровых двойников

Существующие в настоящий момент системы, поддерживающие выполнение научных потоков работ, не обладают механизмами поддержки масштабирования под-потоков и интеграцию с системами потоковой обработки данных. Такие возможности играют важную роль для поддержки работы цифровых двойников.

Для решения этой проблемы мы предлагаем перепроектировать монолитный подход, распространенный в настоящий момент при проектировании научных потоков работ в меньшие слабо-связанные совокупности микро-потоков работ (Micro-Workflows или

MWF) [2, 24], которые можно было бы реализовать в виде независимых вычислительных сервисов, развернутых в отдельных контейнерах и поддерживающих связь друг с другом посредством системы потоковой обработки данных.

Данный подход позволит обеспечить низкие значения задержки сетевого обмена для обработки данных Интернета вещей и позволит реализовать независимую разработку и развертывание микро-потоков работ, составляющих цифровые двойники.

Он также обеспечивает значительное расширение возможностей горизонтального масштабирования и вертикального распределения микро-потоков работ между публичными и частными облаками. Микро-потоки работ, которые должны обеспечивать ответ с низким значением задержки, могут быть развернуты в частных облаках, расположенных рядом с оборудованием Интернета вещей, в то время как микро-потоки работ, обеспечивающие пакетную обработку больших данных, могут быть расположены в узлах вблизи систем хранения.

Для преобразования изначального научного потока работ в совокупность микро-потоков работ, связанных между собой посредством системы потоковой обработки данных, мы ввели «Узел-потребитель» и «Узел-производитель» в точках разделения изначального потока работ. Эти узлы позволяют осуществлять обмен данными между микро-потоками работ и системой потоковой обработки данных (см. рис. 3).

5. Внедрение микро-потоков работ и развертывание эксперимента

Для проверки нашего подхода мы применили предложенную архитектуру с использованием системы Kepler³ в качестве системы управления научными потоками работ, среды Apache Kafka⁴ в качестве платформы потоковой обработки данных, которая под-

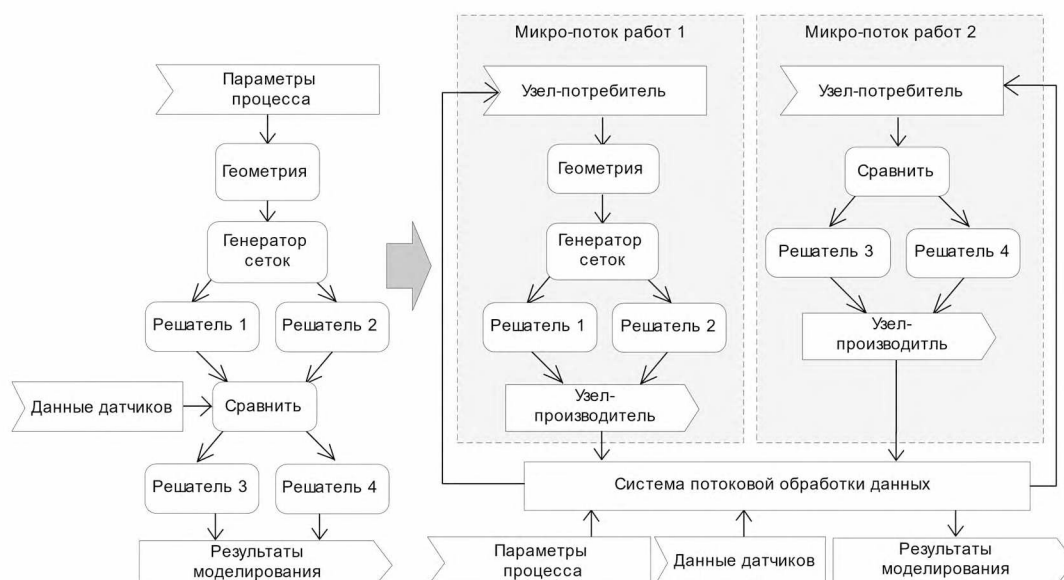


Рис. 3. Подход микро-потоков работ

³ <https://kepler-project.org/>

⁴ <https://kafka.apache.org/>

держивает горизонтальную масштабируемость, и Docker в качестве платформы контейнеризации. Когда гибко связанные сервисы взаимодействуют между собой посредством сообщений, они должны предоставлять универсальный формат сообщений (схему). Платформа Confluent⁵ предоставляет реестр схем на основе системы сериализации данных Avro⁶. Для автоматизации развертывания среды Kafka мы используем образ и реестр схем на базе контейнера, созданного Landoop⁷.

Проведем оценку нашего подхода, используя данные от датчиков, представленных в материалах Конкурса «DEBS⁸ 2012 Grand Challenge: Manufacturing equipment». Представленные данные представляют собой запись реальной информации, передаваемой датчиками системы мониторинга технологического процесса в химической промышленности. Задержка между двумя последовательно расположенными точками в обрабатываемых нами исходных данных составляет около 10 мкс. Каждое обрабатываемое сообщение включает в себя 66 полей. Первый блок операторов регистрирует изменение состояния полей во входных данных и передает их вместе с временными отметками, отмечающими, когда произошло изменение состояния. Второй блок операторов соотносит изменение состояния датчиков и изменение состояния клапанов путем расчета временной разницы между возникновением изменений состояния и передает их далее вместе с временными отметками. Задачей последнего блока операторов является постоянный мониторинг тренда рассчитанных значений от предыдущих выходных данных на период, равный 24 часам.

5.1. Новые акторы для Kepler

Для оценки нашего подхода был разработан набор акторов для системы Kepler. Они расширяют возможности Kepler для работы в среде Apache Kafka и обеспечивают реализацию первого запроса для обработки данных конкурса DEBS 2012, включая:

- **KafkaConsumer** — это актор, обеспечивающий чтение сообщения из исходной темы (topic) Kafka, десериализацию полученных сообщений в формате Avro и пересылку этих сообщений в качестве записей на выходной порт;
- **KafkaProducer** — это актор, обеспечивающий получение записей с входного порта, сериализацию полученных записей путем преобразования в формат Avro и их пересылку в конечную тему Kafka;
- **DetectStateChange** — это актор, регистрирующий изменение в двух последовательных записях данных (с 0 на 1 и наоборот);
- **CorrelateStateChange** — это актор, который выполняет корреляционный анализ состояния изменения во временных последовательностях «0» и «1» между парами датчиков. Он рассчитывает временную разницу между повышенным (1) и пониженным (0) состояниями каждой из входных пар датчиков;
- **XXYY State** — это актор, который выделяет связанные величины временной разницы и подготавливает их для отображения в виде графического представления данных.

⁵ <https://www.confluent.io/>

⁶ <https://avro.apache.org/>

⁷ <https://www.landoop.com/>

⁸ <http://debs.org/grand-challenges/>

5.2. Развертывание

Развертывание нашего решения осуществлено с использованием суперкомпьютера «Торнадо» на базе ресурсов суперкомпьютерного центра Южно-Уральского государственного университета [19] (см. рис. 4). Виртуальная машина VM1 («Эмулятор датчиков») моделирует процесс генерации данных датчиками Интернета вещей. Она потребляет данные из базы данных DEBS 2012 и последовательно публикует их в среде Kafka, развернутой в контейнере Landoor (VM2). Мы разделяем поток работ по первому запросу DEBS 2012 на два микро-потока работ и упаковываем каждый микро-поток работ в отдельный контейнер. Контейнеры разворачиваются на виртуальных машинах VM2 и VM3. Прямое соединения между виртуальными машинами или узлами нет. Все взаимодействие осуществляется через сетевой сервер. Виртуальные машины VM1 и VM3 работают на одном и том же физическом узле и снабжены 4ГБ ОЗУ и четырьмя узлами процессора Intel Xeon X5680. Машина VM2 работает на отдельном физическом узле и снабжена 12 ГБ ОЗУ и восьмиядерным процессором Intel Xeon X5680. Соединение между виртуальными машинами организовано посредством внешнего физического узла, действующего как виртуальный маршрутизатор, подсоединенный через сеть Gigabit Ethernet.

Микро-поток работ MWF-1 потребляет изначальные данные от датчиков из исходной темы в системе Kafka и обрабатывает их, используя актор DetectStateChange (см. рис. 5а). Результаты публикуются в промежуточной теме в системе Kafka. Микро-поток работ MWF-2 потребляет данные из промежуточной темы Kafka и обрабатывает все оставшиеся блоки операторов, включая построение графика конечных результатов (см. рис. 5б).

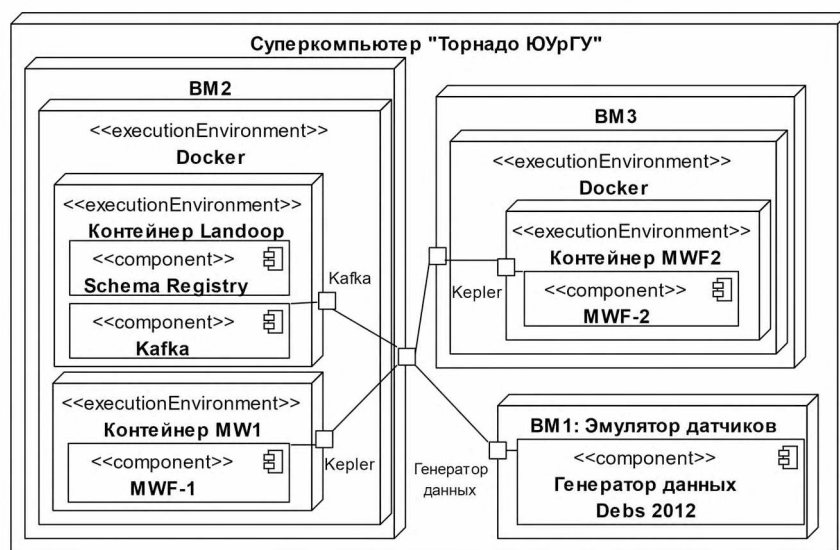


Рис. 4. Развертывание эксперимента на узлах суперкомпьютера «Торнадо ЮУрГУ»

6. Оценка эффективности микро-потоков работ

Для оценки эффективности предлагаемой системы мы добавили следующие дополнительные поля в создаваемые сообщения:

1. mX_{orgts} : время, когда датчик отправляет исходное сообщение номер X.
2. mX_{rcvts} : время, когда Kepler получает исходное сообщение X из исходной темы Kafka.
3. K_{sentts} : время, когда актор KafkaProducer отправляет итоговое сообщение в конечную тему Kafka.

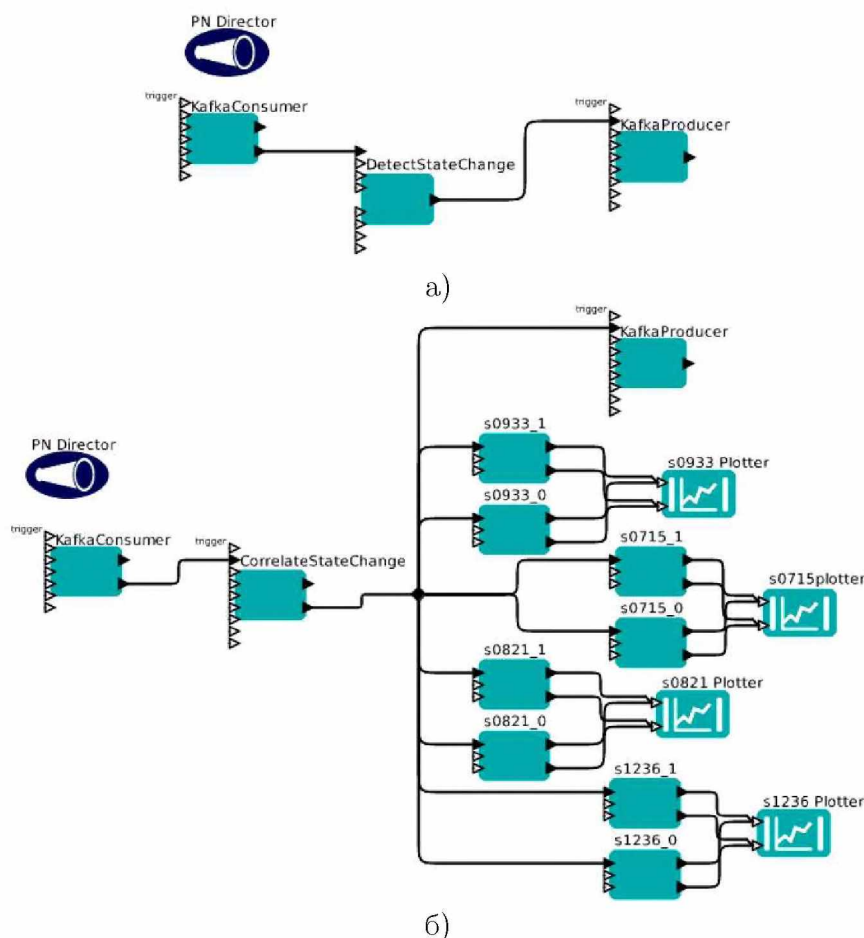


Рис. 5. Развертываемые микро-потоки работ

Для проверки эффективности обмена данными и их обработки, предлагаются следующие критерии оценивания:

1. Av_{SM} (средний интервал между исходными сообщениями): средняя задержка по времени между двумя последовательными исходными сообщениями с данными, переданными датчиком в Kafka.
2. Av_{TAT} (среднее время обработки): средний временной интервал, требуемый для создания одного итогового сообщения микро-поток работ. При этом интервал отсчитывается, начиная от времени получения второго требуемого исходного сообщения ($m2_{rcvts}$) из исходной темы Kafka, и заканчивая временем передачи итогового сообщения (K_{sentts}) в Kafka.
3. Av_{DV} (среднее время доставки): средний временной интервал между mX_{orgts} и mX_{rcvts} , включая сериализацию данных, передачу сообщения от VM1 к VM2, подтверждение получения сообщения от Kafka.
4. Av_{L12} (средняя задержка): средняя сетевая задержка между машиной VM1 и машиной VM2.

Результаты наших экспериментов представлены в таблице. За 24 часа тестирования наша система обработала более 7,3 миллионов сообщений со средним временем выполнения задания около 3,2 мс. Среднее ресурсопотребление нашей системы с использованием потока данных межплатформного программного обеспечения среды Kafka можно оценить следующим образом:

$$Av_{DV} - Av_{L12} = 4,4 - 3,5 = 0,9 \text{ (мс)}.$$

Таблица

Результаты оценки эффективности

Время теста	24 часа
Количество сообщений	7 328 844
Av_{SM}	11,8 мс
Av_{TAT}	3,2 мс
Av_{DV}	4,4 мс
Av_{L12}	3,5 мс

Заключение

В данной работе была представлена концепция облачной платформы цифровых двойников и подход микро-потоков работ, обеспечивающий поддержку цифровых двойников. Наше решение расширяет возможности существующих подходов благодаря сочетанию мощности научных потоков работ, гибкости контейнерных технологий и устойчивости подхода потоковой передачи данных. Оно поддерживает возможность обработки событий от различных источников (таких как датчики Интернета вещей) внутри вычислительных потоков работ. Внедрение и тестирование данной архитектуры на базе системы управления научными потоками работ Kepler и платформы потоковой обработки данных Apache Kafka показывают, что предлагаемая архитектура успешно решает задачи обработки данных интернета вещей, обеспечивая достаточно небольшие размеры сетевой задержки. Основным направлением для будущих работ является оценка масштабируемости и применимости данного подхода для поддержки значительных объемов данных Интернета вещей в режиме реального времени.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта No 18-07-01224 а.

Литература

1. Al-Jarrah O.Y. et al. Efficient Machine Learning for Big Data: A Review // Big Data Research. 2015. Vol. 2, no. 3. P. 87–93. DOI: 10.1016/j.bdr.2015.04.001.
2. Alaasam A.B.A. et al. Scientific Micro-Workflows: Where Event-Driven Approach Meets Workflows to Support Digital Twins // Proceedings of the international conference Russian Supercomputing Days, RuSCDays'18 (Moscow, Russia, September, 24–25, 2018). MSU. 2018. Vol. 1. P. 489–495.
3. Altintas I. et al. Kepler: an extensible system for design and execution of scientific workflows // Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 2004. Vol. I. P. 423–424. DOI: 10.1109/SSDM.2004.1311241.
4. Boettiger C. An introduction to Docker for reproducible research // ACM SIGOPS Operating Systems Review. 2015. Vol. 49, no. 1. P. 71–79. DOI: 10.1145/2723872.2723882.
5. Borodulin K. et al. Towards Digital Twins Cloud Platform: Microservices and Computational Workflows to Rule a Smart Factory // Proceedings of the 10th International Conference on Utility and Cloud Computing, UCC '17 (Austin, Texas, USA, December, 5–8, 2017). P. 209–210. DOI: 10.1145/3147213.3149234.
6. Bryner M. Smart manufacturing: The next revolution // Chemical Engineering Progress. 2012. Vol. 108, no. 10. P. 4–12.

7. Carvalho O., Roloff E., Navaux P.O.A. A Distributed Stream Processing based Architecture for IoT Smart Grids Monitoring // Companion Proceedings of the 10th International Conference on Utility and Cloud Computing, 2017. New York, NY, USA. ACM Press, 2017. P. 9–14. DOI: 10.1145/3147234.3148105.
8. Davis J. et al. Smart manufacturing, manufacturing intelligence and demand-dynamic performance // Computers and Chemical Engineering. 2012. Vol. 47. P. 145–156. DOI: 10.1016/j.compchemeng.2012.06.037.
9. Deelman E. et al. Pegasus, a workflow management system for science automation // Future Generation Computer Systems. 2015. Vol. 46. P. 17–35. DOI: 10.1016/j.future.2014.10.008.
10. Fahringer T. et al. ASKALON: a Grid application development and computing environment // The 6th IEEE/ACM International Workshop on Grid Computing, 2005. IEEE, 2005. P. 10. DOI: 10.1109/GRID.2005.1542733.
11. Filgueira R. et al. IoT-Hub: New IoT data-platform for Virtual Research Environments // 10th International Workshop on Science Gateways, IWSG 2018, 2018. P. 13–15.
12. Filguiera R. et al. Dispel4py: A Python framework for data-intensive scientific computing // International Journal of High Performance Computing Applications. 2017. Vol. 31, no. 4. P. 316–334. DOI: 10.1177/1094342016649766.
13. Foster I.T., Gannon D.B. Cloud Computing for Science and Engineering / Ed. by I.T. Foster, D.B. Gannon. The MIT Press, 2017. 392 p.
14. Glaessgen E., Stargel D. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles // 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (Reston, Virginia, 2012). American Institute of Aeronautics and Astronautics, 2012. P. 1–14. DOI: 10.2514/6.2012-1818.
15. Hiraes-Carbajal A. et al. A Grid simulation framework to study advance scheduling strategies for complex workflow applications // 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, IPDPSW, 2010. IEEE, 2010. P. 1–8. DOI: 10.1109/IPDPSW.2010.5470918.
16. Hiraes-Carbajal A. et al. Multiple Workflow Scheduling Strategies with User Run Time Estimates on a Grid // Journal of Grid Computing. 2012. Vol. 10, no. 2. P. 325–346. DOI: 10.1007/s10723-012-9215-6.
17. Huber N. et al. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments // Proceedings of the 1st International Conference on Cloud Computing and Services Science (Noordwijkerhout, The Netherlands, May, 7–9, 2011). SciTePress - Science and and Technology Publications, 2011. P. 563–573.
18. Korambath P. et al. A smart manufacturing use case: Furnace temperature balancing in steam methane reforming process via kepler workflows // Procedia Computer Science. 2016. Vol. 80. P. 680–689. DOI: 10.1016/j.procs.2016.05.357.
19. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources // Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies, PCT 2016 (Arkhangelsk, Russia, March, 29–31, 2016). CEUR Workshop Proceedings, 2016. P. 561–573.
20. Lee J., Bagheri B., Kao H.-A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems // Manufacturing Letters. 2015. Vol. 3. P. 18–23. DOI: 10.1016/j.mfglet.2014.12.001.

21. Lewis J., Fowler M. Microservices. 2014. URL: <http://martinfowler.com/articles/microservices.html> (дата обращения: 28.01.2016).
22. Li S., Xu L. Da, Zhao S. The internet of things: a survey // Information Systems Frontiers. 2015. Vol. 17, no. 2. P. 243–259. DOI: 10.1007/s10796-014-9492-7.
23. Perera C. et al. Sensing as a service model for smart cities supported by Internet of Things // Transactions on Emerging Telecommunications Technologies. 2014. Vol. 25, no. 1. P. 81–93. DOI: 10.1002/ett.2704.
24. Radchenko G., Alaasam A., Tcherynykh A. Micro-Workflows: Kafka and Kepler Fusion to Support Digital Twins of Industrial Processes // 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion, 2018. P. 83–88. DOI: 10.1109/UCC-Companion.2018.00039.
25. Radchenko G., Hudyakova E. A service-oriented approach of integration of computer-aided engineering systems in distributed computing environments // UNICORE Summit 2012, Proceedings, 2012. P. 57–66.
26. Rahman M. et al. Adaptive workflow scheduling for dynamic grid and cloud computing environment // Concurrency Computation Practice and Experience. 2013. P. 1816–1842. DOI: 10.1002/cpe.3003.
27. Sinha N., Pujitha K.E., Alex J.S.R. Xively based sensing and monitoring system for IoT // 2015 International Conference on Computer Communication and Informatics, ICCCI, 2015. IEEE, 2015. P. 1–6. DOI: 10.1109/ICCCI.2015.7218144.
28. Soldatos J. et al. OpenIoT: Open Source Internet-of-Things in the Cloud // Lecture Notes in Computer Science. 2015. Vol. 9001. P. 13–25. DOI: 10.1007/978-3-319-16546-2_3.
29. Tao F. et al. Digital twin-driven product design, manufacturing and service with big data // International Journal of Advanced Manufacturing Technology. 2017. Vol. 94, no. 9–12, P. 3563–3576. DOI: 10.1007/s00170-017-0233-1.
30. Taylor I. et al. Workflows for e-Science. Springer London, 2007. 523 p.
31. Tuegel E.J. et al. Reengineering Aircraft Structural Life Prediction Using a Digital Twin // International Journal of Aerospace Engineering. 2011. Vol. 2011. P. 1–14. DOI: 10.1155/2011/154798.
32. Xavier M.G. et al. Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments // 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (Belfast, UK, February, 27–March, 1, 2013). IEEE, 2013. P. 233–240. DOI: 10.1109/PDP.2013.41.
33. Савченко Д.И., Радченко Г.И. Методология тестирования микросервисных облачных приложений // Суперкомпьютерные дни в России: Труды международной конференции. 2015. P. 245–256.

Алаасам Амир Басим Абдуламир, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Радченко Глеб Игоревич, к.ф.-м.н., доцент, директор Высшей школы электроники и компьютерных наук, заведующий кафедрой Электронных вычислительных машин, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Черных Андрей Николаевич, к.т.н., доцент, заведующий лабораторией «НИЛ Проблемно-ориентированных облачных сред», Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация), профессор, научно-исследовательский центр Энсенады (Энсенада, Мексика)

DOI: 10.14529/cmse190407

MICRO-WORKFLOWS: A COMBINATION OF WORKFLOWS AND DATA STREAMING TO SUPPORT DIGITAL TWINS OF PRODUCTION PROCESSES

© 2019 A.B.A. Alaasam¹, G.I. Radchenko¹, A.N. Tchernykh^{1,2}

¹South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia),

²Ensenada Research Center

(Carretera Ensenada - Tijuana No. 3918, Ensenada, 22860 Mexico),

E-mail: alaasamab@susu.ru, gleb.radchenko@susu.ru, chernykh@cicese.mx

Received by: 12.11.2019

Recently, there has been an explosive growth in the development of the concept of “Digital Industry”. One of the most important elements of this concept is the application of methods of mathematical modeling and intelligent data analysis to create models of production processes and final products based on the processing of signals coming from intelligent sensors. A set of such models, which represent a virtual representation of industrial processes, systems and equipment, is called “Digital twin”. Digital twins use data from sensors installed on production lines or on the basis of final products to predict equipment failures, optimize product quality and reduce the environmental impact of production processes. The model sets, underlying digital twins, can be described as workflows consisting of a set of computational services, each of which is a model of a process step. To provide flexible support for cloud computing to perform digital twins, we offer the concept of “Micro-Workflows”, which combines the power of the Scientific Workflows concept, the flexibility of container technology and the sustainability of the Stream Processing approach in distributed computing systems.

Keywords: digital twins, microservices, containerization, micro-workflow, Kafka, Kepler.

FOR CITATION

Alaasam A.B.A., Radchenko G.I., Tchernykh A.N. Micro-Workflows: A Combination of Workflows and Data Streaming to Support Digital Twins of Production Processes. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. vol. 8, no. 4. pp. 100–116. (in Russian) DOI: 10.14529/cmse190407.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Al-Jarrah O.Y. et al. Efficient Machine Learning for Big Data: A Review. *Big Data Res.* 2015. vol. 2, no. 3. pp. 87–93. DOI: 10.1016/j.bdr.2015.04.001.
2. Alaasam A.B.A. et al. Scientific Micro-Workflows: Where Event-Driven Approach Meets Workflows to Support Digital Twins. *Proc. Int. Conf. Russ. Supercomput. Days, RuSCDays'18 (Moscow, Russia, September, 24–25, 2018)*. MSU. 2018. vol. 1. pp. 489–495.

3. Altintas I. et al. Kepler: an extensible system for design and execution of scientific workflows. Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 2004. vol. I. pp. 423–424. DOI: 10.1109/SSDM.2004.1311241.
4. Boettiger C. An introduction to Docker for reproducible research. ACM SIGOPS Oper. Syst. Rev. 2015. vol. 49, no. 1. pp. 71–79. DOI: 10.1145/2723872.2723882.
5. Borodulin K. et al. Towards Digital Twins Cloud Platform: Microservices and Computational Workflows to Rule a Smart Factory. Proc. The 10th Int. Conf. Util. Cloud Comput., UCC '17 (Austin, Texas, USA, December, 5–8, 2017). pp. 209–210. DOI: 10.1145/3147213.3149234.
6. Bryner M. Smart manufacturing: The next revolution. Chem. Eng. Prog. 2012. vol. 108, no. 10. pp. 4–12.
7. Carvalho O., Roloff E., Navaux P.O.A. A Distributed Stream Processing based Architecture for IoT Smart Grids Monitoring. Companion Proc. 10th Int. Conf. Util. Cloud Comput. (Austin, Texas, USA, December, 5–8, 2017). New York, NY, USA: ACM Press, 2017. pp. 9–14. DOI: 10.1145/3147234.3148105.
8. Davis J. et al. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. Comput. Chem. Eng. 2012. vol. 47. pp. 145–156. DOI: 10.1016/j.compchemeng.2012.06.037.
9. Deelman E. et al. Pegasus, a workflow management system for science automation. Futur. Gener. Comput. Syst. 2015. vol. 46. pp. 17–35. DOI: 10.1016/j.future.2014.10.008.
10. Fahringer T. et al. ASKALON: a Grid application development and computing environment. 6th IEEE/ACM Int. Work. Grid Comput. 2005. pp. 10. DOI: 10.1109/GRID.2005.1542733.
11. Filgueira R. et al. IoT-Hub: New IoT data-platform for Virtual Research Environments. 10th Int. Work. Sci. Gateways, IWSG 2018. 2018. pp. 13–15.
12. Filguiera R. et al. Dispel4py: A Python framework for data-intensive scientific computing. Int. J. High Perform. Comput. Appl. 2017. vol. 31, no. 4. pp. 316–334. DOI: 10.1177/1094342016649766.
13. Foster I.T., Gannon D.B. Cloud Computing for Science and Engineering. MIT Press, 2017. 392 p.
14. Glaessgen E.H., Stargel D.S. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf. 14th AIAA. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2012. pp. 1–14. DOI: 10.2514/6.2012-1818.
15. Hiraes-Carbajal A. et al. A Grid simulation framework to study advance scheduling strategies for complex workflow applications. 2010 IEEE Int. Symp. Parallel Distrib. Process. Work. Phd Forum. IEEE, 2010. pp. 1–8. DOI: 10.1109/IPDPSW.2010.5470918.
16. Hiraes-Carbajal A. et al. Multiple Workflow Scheduling Strategies with User Run Time Estimates on a Grid. J. Grid Comput. 2012. vol. 10, no. 2. pp. 325–346. DOI: 10.1007/s10723-012-9215-6.
17. Huber N. et al. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. Proc. 1st Int. Conf. Cloud Comput. Serv. Sci. Noordwijkerhout, The Netherlands: SciTePress - Science and and Technology Publications, 2011. pp. 563–573.

18. Korambath P. et al. A smart manufacturing use case: Furnace temperature balancing in steam methane reforming process via kepler workflows. *Procedia Comput. Sci.* Elsevier Masson SAS, 2016. vol. 80. pp. 680–689. DOI: 10.1016/j.procs.2016.05.357.
19. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources. *Proc. 10th Annu. Int. Sci. Conf. Parallel Comput. Technol., PCT 2016 (Arkhangelsk, Russia, March, 29–31, 2016)*. CEUR Workshop Proceedings, 2016. vol. 1576. pp. 561–573.
20. Lee J., Bagheri B., Kao H.A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manuf. Lett.* 2015. vol. 3. pp. 18–23. DOI: 10.1016/j.mfglet.2014.12.001.
21. Lewis J., Fowler M. Microservices. Available at: <http://martinfoowler.com/articles/microservices.html> (accessed: 08.11.2019).
22. Li S., Xu L. Da, Zhao S. The internet of things: a survey . *Inf. Syst. Front.* 2015. vol. 17, no. 2. pp. 243–259. DOI: 10.1007/s10796-014-9492-7.
23. Perera C. et al. Sensing as a service model for smart cities supported by Internet of Things. *Trans. Emerg. Telecommun. Technol.* 2014. vol. 25, no. 1. pp. 81–93. DOI: 10.1002/ett.2704.
24. Radchenko G., Alaasam A., Tchernykh A. Micro-Workflows: Kafka and Kepler Fusion to Support Digital Twins of Industrial Processes. 2018 IEEE/ACM Int. Conf. Util. Cloud Comput. Companion, UCC Companion. 2018. pp. 83–88.
25. Radchenko G., Hudyakova E. A service-oriented approach of integration of computer-aided engineering systems in distributed computing environments. UNICORE Summit 2012, Proc. 2012. vol. 15. pp. 57–66. DOI: 10.1109/UCC-Companion.2018.00039.
26. Rahman M. et al. Adaptive workflow scheduling for dynamic grid and cloud computing environment. *Concurr. Comput. Pract. Exp.* 2013. vol. 25, no. 13. pp. 1816–1842. DOI: 10.1002/cpe.3003.
27. Sinha N., Pujitha K.E., Alex J.S.R. Xively based sensing and monitoring system for IoT. 2015 Int. Conf. Comput. Commun. Informatics, ICCCI 2015. 2015. pp. 1–6. DOI: 10.1109/ICCCI.2015.7218144.
28. Soldatos J. et al. OpenIoT: Open source internet-of-things in the cloud. *Lect. Notes Comput. Sci.* 2015. vol. 9001. pp. 13–25. DOI: 10.1007/978-3-319-16546-2_3.
29. Tao F. et al. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* 2017. vol. 94, no. 9–12. pp. 3563–3576. DOI: 10.1007/s00170-017-0233-1.
30. Taylor I. et al. *Workflows for e-Science*. Springer London, 2007. 523 p.
31. Tuegel E.J. et al. Reengineering aircraft structural life prediction using a digital twin. *Int. J. Aerosp. Eng.* 2011. pp. 1–14. DOI: 10.1155/2011/154798.
32. Xavier M.G. et al. Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments. 2013 21st Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. Belfast, UK: IEEE, 2013. pp. 233–240. DOI: 10.1109/PDP.2013.41.
33. Savchenko D.I., Radchenko G. Testing methodology for microservice cloud applications. *Supercomputer Days in Russia Proceedings of the International Conference*. Moscow: Moscow State University, 2015. pp. 245–256. (in Russian)