

Федеральное Государственное бюджетное учреждение науки
Институт проблем управления имени В.А. Трапезникова
Российской Академии наук

На правах рукописи



Сулейкин Александр Сергеевич

**МЕТОДЫ АНАЛИЗА И СИНТЕЗ АРХИТЕКТУРЫ ЦИФРОВЫХ
ПРОИЗВОДСТВЕННЫХ ЭКОСИСТЕМ**

Специальность 2.3.3 – Автоматизация и управление
технологическими процессами и производствами (технические науки)

Диссертация на соискание ученой степени
кандидата технических наук

Москва, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. АНАЛИЗ ТЕНДЕНЦИЙ РАЗВИТИЯ ЦИФРОВЫХ ЭКОСИСТЕМ	11
1.1. Определения и примеры цифровых экосистем	11
1.2. ЦЭС как взаимодействие цифровых консортов	19
1.3. Использование цифровых двойников в консорт-сервисах управления	21
1.3.1. Обзор литературы по разработке цифровых двойников	21
1.3.2. Анализ проблем внедрения цифровых двойников	24
1.3.3. Цифровые двойники в консорт-сервисах управления и моделирования производственных ситуаций	24
1.4. Модели, формируемые идентификатором	26
1.5. Анализ систем класса больших данных	29
1.6. Управление вычислительной инфраструктурой на основе виртуализации и контейнеризации	34
1.7. Анализ возможностей слияния разнородных данных	38
1.8. Выводы по Главе 1	47
ГЛАВА 2. МЕТОДЫ АНАЛИЗА И ИДЕНТИФИКАЦИИ ЦИФРОВЫХ ПРОИЗВОДСТВЕННЫХ ЭКОСИСТЕМ КАК МУЛЬТИКОНСОРТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ	52
2.1. Условия устойчивого функционирования ЦЭС	52
2.2. Методы анализа и прогнозирования ЦЭС на основе цифровых алгоритмов идентификации	53
2.3. Идентификаторы в цепи обратной связи как цифровые двойники	57
2.4. Условия устойчивости для кратномасштабных кейвлет-разложений для мультимодальных систем	58
2.5. Алгоритмы прогнозирования производственных ситуаций	62
2.6. Выводы по Главе 2	65
ГЛАВА 3. РАЗРАБОТКА МОДЕЛИ АРХИТЕКТУРЫ ЦЭС	67
3.1. Концептуальная архитектура системы управления ЦЭС предприятия	67
3.2. Функциональная архитектура консорт-сервисного слоя	71
3.3. Методы анализа стабильного функционирования консорт-сервисного слоя и инфраструктуры ЦЭС	75

3.3.1. Стабильность функционирования консорт-сервисов	75
3.3.2. Стабильность функционирования консортов инфраструктуры	78
3.4. Выводы по Главе 3	85
ГЛАВА 4. РАЗРАБОТКА ПРОТОТИПА СИСТЕМЫ ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ ЗАРУЗКИ СЕРВЕРНОГО КОМПЛЕКСА ДЛЯ УПРАВЛЕНИЯ РЕСУРСАМИ В ЦИФРОВОЙ ПРОИЗВОДСТВЕННОЙ ЭКОСИСТЕМЕ ПАО «КАМАЗ»	87
4.1. Описание технологических компонентов прототипа системы управления ресурсами в цифровой производственной экосистеме ПАО «КАМАЗ»	87
4.2. Реализация задачи прогнозирования состояния серверного комплекса ресурсов.....	91
4.3. Разворачивание сервиса горячего хранения данных	96
4.4. Разворачивание сервиса интеграций и слияния данных	98
4.5. Разворачивание сервиса визуализаций и результаты прогнозирования	103
4.6. Стабильность полученного прототипа системы управления	105
4.7. Выводы по Главе 4	106
ЗАКЛЮЧЕНИЕ	109
ЛИТЕРАТУРА	110
ПРИЛОЖЕНИЕ 1	128

ВВЕДЕНИЕ

Выраженной тенденцией в мировой экономике становится то, что отраслевые и меж-секторальные цифровые платформы становятся основой для разработки цифровых экосистем, что позволяет создавать новые бизнес-модели, инновационные решения и повышать конкурентоспособность. Наиболее сложными, объемными и проблемными в этом аспекте являются промышленное производство, транспорт и электроэнергетика.

Под *цифровой экосистемой* (ЦЭС) будем понимать распределенную социотехническую систему, обладающую свойствами адаптивности, самоорганизации и устойчивости, функционирующую в условиях конкуренции и сотрудничества между различными субъектами данной системы (автоматизированными системами и экономическими субъектами), в которой осуществляется обмен знаниями для эволюционного развития системы. Цифровая экосистема осуществляет функционирование на основе компьютерной сетевой инфраструктуры с использованием цифровых технологий управления. Примером могут служить, например, мультиагентные технологии. Как правило, она представляет собой сеть компаний и организаций в регионе, которые используют попутно образующиеся продукты, отходы и энергию по одному из следующих направлений: уменьшение объемов потребляемого первичного сырья; снижение степени загрязнения окружающей среды; повышение эффективности использования энергии, а, следовательно, уменьшение объемов потребления первичных энергетических ресурсов; уменьшение количества отходов и затрат на их захоронение. По аналогии с природными экосистемами взаимодействующие элементы цифровых экосистем будем называть консортами. Цифровой консорт представляет собой элемент ЦЭС, решающий определенную целевую задачу/набор задач, либо выполняющий сервисные функции.

Конкурентоспособность промышленных производителей, транспортных и электроэнергетических компаний сегодня во многом определяется гибкостью и

эффективностью их деятельности за счет применения передовых информационных технологий, обеспечивающих быстрое и эффективное использование цифровой информации, а также современных технологий управления для решения определенных производственных задач.

Возрастающая информационная насыщенность и сложность решаемых в реальном времени разнообразных производственных задач на разных уровнях производственного управления (управление технологическими процессами, оперативное управление производством и производственная логистика, управление ресурсами и взаимоотношениями с поставщиками и потребителями) делают естественным переход от жестких иерархических «вертикалей» управления к распределенному формату. Особенную значимость играют комплексное обеспечение устойчивости систем управления производственными процессами в условиях динамически нестабильной внешней среды, а также устойчивости вычислительной инфраструктуры в условиях обработки больших производственных данных.

В последние годы наиболее востребованным подходом к управлению сложными технологическими процессами как объектами управления (нелинейными, объектами со значительным транспортным запаздыванием, со взаимосвязанными регулируемыми переменными, со множеством одновременно соблюдаемых ограничений) стала технология управления на основе прогнозирующей модели объекта. Идея такого управления – введение заранее построенной математической модели объекта в контур автоматического управления. В мировую практику вошел термин Model-based predictive control (или сокращенно Model predictive control (MPC) - «Прогнозирующее управление на основе модели» или «предикт-контроллер». Системы управления с использованием прогнозирующей модели получили название систем усовершенствованного управления (APC - Advanced Process Control).

Представляется перспективным использовать управление на основе прогнозирующих моделей как основной принцип управления не только технологическими процессами, но и другими процессами управления в цифровой экосистеме предприятия. Для разработки систем такого типа представляется целесообразным использовать идентификационные модели.

Наиболее известны в области теории и методов идентификации классические работы П. Эйкхоффа, Д. Гроппа, Э.П. Сэйджа и Дж.Л. Мелсы, Я.З. Цыпкина, Н.С. Райбмана, Л. Льюнга, , А.В. Назина, А.Г. Александрова, А.Л. Бунича и других. Для разработки производственных ЦЭС представляется целесообразным использовать цифровые идентификационные модели, основанные на индуктивных знаниях о производственных процессах.

Для разработки и адаптивной настройки систем управления могут эффективно применяться цифровые двойники. Цифровые двойники функционируют на основе данных (архивных и актуальных) о состоянии производственных процессов и состоянии комплекса производственных ресурсов (технических и технологических, энергетических, вычислительных, человеческих). В качестве цифровых двойников могут быть использованы не только киберфизические системы на основе интернета вещей и имитационных моделей, но и идентификаторы в цепи обратной связи системы управления.

Для создания системы управления ЦЭС промышленного предприятия требуется знание методов, технологий и инструментов больших данных (Big Data), на которых базируются современные цифровые платформы. Применение современных технологий хранения и обработки больших данных, их интеграции, виртуализации, контейнеризации, наряду с использованием методов управления цифровыми экосистемами на основе цифровых предиктивных моделей предоставляет возможность разработать интегрированную систему управления производством нового поколения.

Актуальность темы диссертационного исследования обусловлена технологической, экономической и экологической целесообразностью построения системы управления ЦЭС промышленного предприятия как интегрированной системы управления производственным комплексом с использованием предиктивных моделей реального времени производственных процессов на основе обработки больших разнородных производственных данных в реальном времени.

Цель диссертационной работы состоит в разработке методов анализа и синтеза архитектуры цифровых производственных экосистем на основе цифровых предиктивных идентификационных моделей.

Для достижения поставленной цели должны быть решены следующие задачи:

- проанализировать известные результаты в области разработки и развития ЦЭС;
- разработать методы анализа и прогнозирования показателей производственных процессов и ситуаций на основе цифровых алгоритмов идентификации;
- разработать методы идентификации ЦЭС как мультиконсортных динамических систем;
- исследовать и проанализировать: системы класса больших данных для обработки разнородных данных в режиме реального времени в оперативной памяти (Big Data In-Memory Systems), системы обработки сообщений в потоковом режиме, технологии виртуализации и контейнеризации;
- разработать концептуальную архитектуру и функциональную архитектуру сервисного слоя системы управления ЦЭС промышленного предприятия;
- разработать подход к обеспечению стабильного функционирования ЦЭС, как учитывающий влияние внешних эко-систем, так и обеспечивающий: надежность вычислительной инфраструктуры, стабильное функционирование обслуживающих консорт-сервисов, устойчивость систем управления;

- разработать прототип системы прогнозирования состояния загрузки серверного комплекса для управления ресурсами в цифровой производственной экосистеме ПАО «КАМАЗ» - одного из консортов ЦЭС.

Объектом диссертационного исследования является совокупность производственных процессов промышленного предприятия.

Предмет диссертационного исследования – ЦЭС промышленного предприятия как интегрированная система управления производством.

Методы исследования. Методы идентификации систем управления, методы интеллектуального анализа данных, методы машинного обучения, разработки систем больших данных, методы разработки реляционных и нереляционных баз данных, методы визуализации данных.

Связь диссертации с планом научных работ. Исследования выполнялись в соответствии с планом научных работ ИПУ РАН.

Соответствие шифру специальности. Работа соответствует шифру специальности 2.3.3 и охватывает следующие области исследований:

- п. 6. Научные основы, модели и методы идентификации производственных процессов, комплексов и интегрированных систем управления;

- п. 9. Методы эффективной организации и ведения специализированного информационного и программного обеспечения АСУТП, АСУП, АСТПП и др., включая базы и банки данных и методы их оптимизации;

- п. 11. Методы планирования и оптимизации отладки, сопровождения, модификации и эксплуатации задач функциональных и обеспечивающих подсистем АСУТП, АСУП, АСТПП и др., включающие задачи управления качеством, финансами и персоналом;

- п. 13. Теоретические основы и прикладные методы анализа и повышения эффективности, надежности и живучести АСУ на этапах их разработки, внедрения и эксплуатации;

- п. 19. Разработка методов обеспечения совместимости и интеграции АСУ, АСУТП, АСУП, АСТПП и других систем и средств управления.

Научная новизна работы заключается в разработке:

- архитектуры системы управления ЦЭС промышленного предприятия на основе цифровых предиктивных моделей;
- цифровых ассоциативных методов идентификации ЦЭС как мультиконсортных систем.
- метода ассоциативного прогнозирования производственных ситуаций;
- цифровых двойников производственных процессов как идентификаторов в цепи обратной связи на основе цифровых идентификационных моделей;
- условий устойчивости мультиконсортных производственных систем.

Практическая значимость диссертационной работы заключается в разработке прототипа системы управления производственными ресурсами в ЦЭС промышленного предприятия. Разработанная реальная система управления позволяет:

- в режиме реального времени строить прогнозирующие модели изменения состояния ресурсов производства;
- в режиме реального времени выявлять прогнозируемые критичные отклонения исследуемых показателей системы для поддержки принятия превентивных управленческих решений;
- устойчиво функционировать и самостоятельно возобновлять работу в случае отказов оборудования;
- взаимодействовать с другими внешними экосистемами, обмениваясь данными через определенные консорт-сервисы в рамках системы управления.

Внедрение результатов исследований

Результаты теоретического исследования ЦЭС, методов хранения и обработки больших данных, методов слияния данных, алгоритмов создания прогнозирующих

моделей производственных ситуаций на основе пополняемой базы знаний, моделей архитектуры, а также реализованные консорт-сервисы прогнозирования состояния загрузки серверного комплекса ресурсов используются в ПАО «КАМАЗ», что подтверждено Актом о внедрении.

Апробация работы. Результаты диссертационной работы докладывались на международных конференциях – MIM – Нант, Франция, 2022 (2 доклада); Берлин, Германия, 2019; CoDIT – Стамбул, Турция, 2022; IMS – 2022; DAAAM International – Задар, Хорватия, 2017, 2019, 2020, 2021; FRUCT – Москва, Россия, 2019; MACSPro – Вена, Австрия, 2019; IEEE BigData – Лос-Анджелес, США, 2019, 2020; IntelliSys – Ам-стердам, Голландия, 2020; MEDES – 2020, 2021; FICC – Ванкувер, Канада, 2021; INCOM – 2021.

По теме диссертации опубликованы 22 печатные работы, все работы индексируются в Web of Science и/или SCOPUS и содержатся в перечне ВАК.

Диссертация состоит из введения, четырех глав, заключения, списка литературы и приложения. Диссертация содержит 128 стр. основного текста, 23 иллюстрации, 1 таблицу; список литературы включает 170 наименований.

ГЛАВА 1. АНАЛИЗ ТЕНДЕНЦИЙ РАЗВИТИЯ ЦИФРОВЫХ ЭКОСИСТЕМ

1.1. ОПРЕДЕЛЕНИЯ И ПРИМЕРЫ ЦИФРОВЫХ ЭКОСИСТЕМ

Основные определения

Можно констатировать стремительное развитие цифровых экосистем в течение последних лет, что подтверждается большим количеством исследований в этой области, а также активным развитием ИТ-технологий. Цифровая экономика способствует развитию цифровых экосистем, порождает их взаимосвязи, новые виды взаимодействия и новые бизнес-модели.

Существуют определенные разночтения в определении цифровой экосистемы. Наиболее часто в литературе выделяются следующие аспекты.

- F. Nachira, P. Dini, A.A. Nicolai [8]: цифровая экосистема (ЦЭС) формируется посредством объединения ИТ-сетей (ИТ – информационные технологии), социальных и обмена знаниями. Понятия *e-learning ecosystem* и *digital ecosystem* (экосистемы с машинным обучением и цифровые экосистемы) отождествляются. ЦЭС обеспечивает: доступ к знаниям, глобальным цепочкам добавленной стоимости, специфическим сервисам; адаптацию новых технологий; принятие новых бизнес-моделей. «Экономика больше не рассматривается как полностью управляемая система, для которой составляется план функционирования» [9].

- H. Dong, F.K. Hussain, E. Chang под цифровой экосистемой понимают совокупность цифровых двойников и инфраструктуры передачи данных, их хранения и обработки, а также пользователей систем [6].

- В цифровой экосистеме «партнеры и конкуренты взаимодействуют как единая команда, объединяя ресурсы, знания для совместной работы над проектами в режиме взаимной полноты информации и созидания» [10].

- При этом «Информация становится ресурсом, который можно использовать, производить и трансформировать так же, как материальные ресурсы» [11].

- В [12] экосистема определяется как домен кластерной среды, в которой все участники «слабо связаны, соблюдают собственную выгоду и берегают окружающую среду».

- В цифровой экосистеме экономика спонтанно трансформируется в сетевую, т. е. в «непрерывно текущее пространство потоков», получая способность непрерывных обновлений [13].

При этом многие исследователи полагают, что ЦЭС представляет собой сеть компаний и организаций, которые используют попутно образующиеся продукты, отходы и энергию по одному из следующих направлений: уменьшение объемов потребляемого первичного сырья; снижение степени загрязнения окружающей среды; повышение эффективности использования энергии, а, следовательно, уменьшение объемов потребления первичных энергетических ресурсов; уменьшение количества отходов и затрат на их захоронение.

Таким образом, под цифровой экосистемой будем понимать распределенную социо-техническую систему, обладающую свойствами адаптивности, самоорганизации и устойчивости, функционирующую в условиях конкуренции и сотрудничества между различными субъектами данной системы (автоматизированными системами и экономическими субъектами) для обмена знаниями в условиях эволюционного развития системы.

В перечисленных и других работах в области разработки цифровых экосистем, в основном, рассматриваются экономические преимущества цифровых экосистем, особенности их применения в различных областях, а также используемые инновационные технологии. Значительное внимание в работах уделяется задачам построения мультиагентного управления экосистемами, налаживанию взаимосвязей участников и обеспечению устойчивости. В данной диссертационной работе исследована проблема построения систем управления

цифровыми экосистемами на основе прогнозирующих моделей в режиме реального времени, что определяет научную новизну данного исследования.

Цифровые экосистемы управления цепочками поставок

Для обеспечения устойчивого развития производственной системы и достижения конкурентного преимущества необходимо не только гарантировать устойчивость и оптимальность управления ключевыми производственными процессами определенного предприятия. Необходимо исследование всей цепочки создания добавленной стоимости, включая места сопряжения различных этапов производственного процесса, включая предшествующие стадии. Поэтому технология управления цепочками поставок (Supply Chain Management, SCM) играет определяющую роль в интеграции всех логистических процессов – как непосредственно на самом предприятии, производящем продукцию, так и логистики поставщиков сырья, материалов, комплектующих, информационных, транспортных, складских услуг.

Из литературы известно множество моделей и методов управления цепочками поставок [14]. Понятие экосистемы применительно к цепочкам поставок формулируется различным образом [15]. Так, в работе [16] термин «экосистема» используется в качестве единицы анализа при описании группировок поставщиков и распределительных цепочек, которые понимаются как свободные группы организаций, занимающихся созданием и доставкой продуктов и услуг. Тот же термин используют Iansit и Levien [17] в описании «стратегии как экологии». Концепция создания стоимости была определена Портером [18] как «вертикальная цепочка, продолжающаяся от поставщика ресурсов фирмам и покупателям товаров и услуг у этих фирм».

Эта структура была далее расширена [19] в понятие «цепочки поставок» – объединения всех поставщиков, посредников и клиентов, которые вместе представляют цепочку создания стоимости организации и рынка, на котором она работает. В [20] цепочка поставок определяется как «сеть организаций, которые

вовлечены через восходящие и нисходящие связи в различные процессы и виды деятельности, которые ценят продукт в форме продуктов и услуг в руке конечного потребителя».

Появление цифровых экосистем продуцирует реконфигурацию создания стоимости. В [21] представлены три типа «созвездий контрольных точек, которые представляют три модели топологии участников цепочки: закрытая вертикально-интегрированная модель, модель слабосвязанной коалиции и модель многосторонних платформ».

Возникает «новая топологическая модель, возникающая вокруг крупных игроков, которые стремятся доминировать на мировых рынках и нарушить границы отрасли. Эта модель сформирована перекрывающимися бизнес-сообществами, которые включают в себя крупные организации и их соответствующие экосистемы» [22]. Автор описывает «глобальную эволюционную тенденцию – к цепочке поставок, управляемой цифровыми экосистемами», описывая общий концепт развития цепочек поставок и места в нем цифровым экосистемам.

Цифровые экосистемы могут быть описаны [23] как цифровые аналоги биологических экосистем. Отмечены самоорганизующиеся свойства биологических экосистем, которые могут включать надежную, самоорганизующуюся и масштабируемую архитектуру для автоматического решения сложных и динамических проблем. Принципы и семантика, используемая в цифровых экосистемах, сформулированы в [24]. Продолжаются исследования свойств цифровых экосистем путем расширения их сфер применения в таких областях, как транспорт, образование и здравоохранение [25]. В [26] авторы представляют технологию разработки цифровой экосистемы для транспорта и складской логистики. Виртуальный Совместный Консорциум, осуществляющий цифровую бизнес экосистем в Австралии, является примером среды сотрудничества для всех, кто участвует в создании продукта распределительной

цепочки. Последнее предполагает создание цепочки поставок, которая способствует интеграции и сотрудничества малых и средних предприятий, в частности, стимулирует сотрудничество и повышает эффективность бизнеса [27].

В [28] предложена модель распределенной цепочки поставок, основанная на мультиагентной технологии. В [29] задача цепочки доставки сформулирована в терминах «сети зависимостей задач»; на основе математической модели изучены вопросы равновесия и сходимости, и предложено приложение для формирования цепочки доставки в автомобильном производстве. В [30] уделяется внимание исследованию в области создания ценностей в цифровых цепочках поставок для организаций в исследовательской программе, где объединено теоретическое исследование, и реализована полностью автономная цепочка поставок, начиная с завода и заканчивая клиентской компанией.

Из литературы известно, что уже успешно функционируют цифровые платформы, которые, по сути, являются экосистемами в области цепочек поставок. Примером [31] может служить платформа Oracle Transportation Management (OTM) для управления всей транспортной деятельностью в цепочках поставок. Продукт помогает сократить затраты на фрахт, оптимизировать уровни обслуживания и автоматизировать процессы таким образом, чтобы компании могли более эффективно выполнять логистические операции.

Еще один пример: Австралийское подразделение аудиторской компании PwC, Торгово-промышленная палата (ACCI) и порт Брисбена (Австралия) осуществляют разработку решения для повышения эффективности цепей поставок на основе технологии блокчейн (blockchain). Решение получило название Trade Community System [32].

Анализ цифровых экосистем в области цепочек поставок показывает, что это одна из первых сфер применения данной концепции. Сложность цепочек поставок, большое количество участников рынка, различные цели субъектов взаимодействия, различные типы данных, большие объемы данных – все это

формирует глобальную эволюционную тенденцию в переходе от централизованных систем управления цепочками поставок к управлению на основе цифровых экосистем.

Цифровые экосистемы в электроэнергетике

Можно констатировать, что наличие большой сложности взаимодействия субъектов, большого количества заинтересованных лиц, разрозненной информации, устойчивость, самоорганизованность, эффективная утилизация используемых вычислительных ресурсов и необходимость быстрого анализа данных порождают развитие цифровых экосистем и в других сферах, таких как электроэнергетика и производство.

В децентрализованном производстве электрической энергии с использованием возобновляемых источников энергии и сложной сети новых и действующих участников, бизнес-моделей и процессов управлять всей системой на основе прежних централизованных принципов и подходов становится практически невозможно. Многие работы посвящены созданию фреймворков для управления цифровыми электроэнергетическими экосистемами на основе технологии Smart Grid [33], а также фреймворков и архитектуры по управлению такой сложной децентрализованной системой на основе идентификационных моделей [2].

В работе [34] рассматриваются цифровые энергетические платформы, вопросы их безопасности и их влияние на общественную жизнь. Показано, что развитие цифровых платформ также оказывает положительный эффект на общество и окружающую среду, экономит ресурсы и повышает прозрачность. Цифровизация является одним из ключевых драйверов повышения эффективности всей энергетической системы, постепенно образуя цифровые экосистемы уровня города, государства, а также мировые цифровые экосистемы, образуя единые информационные пространства взаимодействия.

Цифровые экосистемы производства

Цифровизация и появление цифровых экосистем также активизируются в сфере производства.

В работе [35] рассмотрены современные тенденции и перспективы развития интегрированных производственных систем в условиях цифровой трансформации. Показано, что ключевые тренды – это создание цифровых двойников производственных процессов, развитие платформенных решений и производственных экосистем.

В работе [2] рассматривались вопросы моделирования архитектуры и интеллектуальных алгоритмов для задач управления цифровыми экосистемами электросетей, а также подход к архитектуре системы управления цифровой производственной экосистемой [36]. Показано, что цифровизация и управление цифровой экосистемой играют важную роль в современном производстве.

Все более актуальными становятся вопросы сбора и анализа большого количества гетерогенных данных, создания интеллектуальных моделей на каждом этапе жизненного цикла производства, а также на разных уровнях производственного управления – цеха, завода, региона и холдинга. Данные задачи требуют создания децентрализованной системы управления цифровой производственной экосистемой.

В научной литературе многочисленные работы последних лет по решению проблем интегрированного управления и производственного планирования ориентированы на разработку кибер-физических производственных систем и цифровых двойников для оптимизации производственного процесса. В работе [37] представлено базовое решение для децентрализованного интегрированного принятия решений по изменению расписания киберфизической производственной системы. В [38] авторы представили структуру синхронизированного цифрового двойника для планирования добычи в условиях неопределенности, а жизнеспособность этой структуры была продемонстрирована в модели производственной линии в лаборатории. В статье [39] авторы рассмотрели текущее

состояние интегрированной системы производства и пришли к выводу, что решения в реальном времени и на основе данных производства и цепочек поставок находятся в тренде, и их популярность будет только продолжать расти.

Другой подход к управлению динамикой производственных процессов описан в [40, 41] – управление посредством изменения конфигурации продуктов. Однако в статье не рассматриваются факторы устойчивости на уровне вычислительной инфраструктуры.

В условиях постоянно растущего объема данных и их разрозненности (дискретность, форматы, протоколы, схема данных, тип загрузки и т. д.) построение системы управления ЦЭС предприятия является сложной задачей, решение которой включает в себя как интеграцию гетерогенных данных с разных уровней производственного управления, построение современной отказоустойчивой масштабируемой системы хранения и управление этими данными, так и построение интеллектуальных сервисов для прогнозирования динамики изменения состояния всех ресурсов, что является ядром интегрированной системы управления.

Повышение требований по быстродействию, точности и эффективности управления в условиях неопределённости при наличии различного рода возмущений в системах управления производственными процессами продемонстрировали ограниченность возможностей традиционных подходов к синтезу систем автоматического управления для широкого класса объектов. К этому классу можно отнести: многие нелинейные и нестационарные объекты, многомерные многосвязные системы, объекты с плохо формализуемыми факторами внешнего воздействия и др.

Постиндустриальная экономика охарактеризовалась созданием методов адекватного представления в машинной среде знаний о свойствах и функционировании реальных объектов и процессов. И ученые, и технологи пришли к выводу, что полная автоматизация предприятия как социотехническая система

должна создаваться на принципиально иной методологической основе. Потребовался новый методологический подход, использующий в управлении производством генерирование в реальном времени индуктивных знаний на основе интеллектуального анализа данных функционирования объекта/процесса. Такой подход позволяет, в частности, эффективно прогнозировать различные нештатные ситуации, обеспечивая устойчивость функционирования цифровых производственных экосистем и их безопасность [1-7].

В данной диссертационной работе рассматриваются вопросы создания системы управления ЦЭС современного промышленного предприятия с использованием предиктивных моделей, использующих индуктивные знания, что представляет собой подход к созданию интегрированной системы нового поколения управления производством.

1.2. ЦЭС КАК ВЗАИМОДЕСТВИЕ ЦИФРОВЫХ КОНСОРТОВ

Экосистема в природе представляет собой «результат взаимодействия разнообразных факторов неживой природы (солнечное излучение, водные ресурсы, воздух, особенности почвы и др.), который обеспечивает и поддерживает не только устойчивое существование определенных видов живых организмов, но и их эволюционное развитие» [41]. «Члены» («участники», «элементы») экосистемы, называемые далее «консортами», в процессе адаптации к естественным условиям существования получают дополнительное преимущество, вступая между собой в многочисленные взаимосвязи.

Подобно натуральным экосистемам, в жизнедеятельности людей (бизнесе, производстве, логистике, и т.д.) также возникают и/или формируются экосистемы. Действия всех элементов взаимовыгодны, реализуют взаимную поддержку и в итоге создают для всех «консорт» возможности, которые не осуществимы вне данной системы. В результате приобретаются новые навыки и компетенции, и формируется добавленная стоимость.

Сегодня существование и успешное функционирование таких сложных социотехнических систем прочно ассоциировано с их реализацией в цифровой информационной среде посредством современных информационных технологий [36]. Создаются цифровые экосистемы (ЦЭС) – цифровые динамические портреты таких экосистем. ЦЭС представляют собой сложные человеко-машинные системы с развитой информационной инфраструктурой. Для них характерно иметь общие системы хранения и обработки больших данных; они используют разнообразные сервисы решения общих и собственных задач. Для цифровых экосистем стало реальностью функционирование отдельных «консортов» на основе общих цифровых платформ и облачных технологий [43].

По мере адаптации и укрепления взаимодействий, совершенствуются факторы устойчивости функционирования, инновационного и устойчивого развития – всех элементов и системы в целом [4]. ЦЭС, по сути, представляет собой формализацию системы взаимодействующих элементов – цифровых консортов. Цифровой консорт представляет собой элемент ЦЭС, решающий определенную целевую задачу/набор задач, либо выполняющий сервисные функции. Иногда в рамках определенного проекта консорты объединяются, образуют *цифровые консорции* (подсистемы ЦЭС). Взаимодействие цифровых консортов (и/или консорций) направлено либо на совместное выполнение определенных задач (собственных или общих для системы), либо определенные консорты выполняют какую-нибудь задачу (или ее часть) для других консортов. В общем случае, такая формализация предусматривает возможность функционирования и взаимодействия цифровых консортов единообразно, в режиме потокового обмена сообщениями. Имеется в виду формализация как функциональных сервисов обработки данных, так и сервисов, реализующих построение моделей реального времени и решение задач управления цифровой экосистемой.

В зависимости от особенностей конкретной экосистемы, ее элементами могут быть как технические системы (отдельные агрегаты или комплексы), так и активные элементы со способностью к самоорганизации. В ЦЭС такие элементы

могут быть формализованы как интеллектуальные агенты [36]. Несмотря на то, что автономность является одним из атрибутов агентов в мультиагентных системах, все более востребованным является применение отдельным агентом общих цифровых сервисов, цифровых продуктов, приложений и устройств, которые могут формироваться «на лету», «по требованию», в реальном времени, с учетом соблюдения общих норм и регламентов.

Как показано в некоторых работах, совсем не обязательно сохранять «монолитность» архитектуры мультиагентных систем. Однако применение микро-сервисного подхода должно быть ограничено рамками требований стандартов FIРА к архитектурам мультиагентных систем [44]. К тому же, в условиях необходимости обработки и анализа больших данных (гетерогенных и поступающих в систему асинхронно), мультиагентный подход к формированию архитектуры ЦЭС не всегда является целесообразным. Не для всех задач управления в такой сложной динамической системе, как ЦЭС, решение на основе достижения консенсуса подсистем существует и может быть получено в течение приемлемого времени. В то же время, некоторые элементы ЦЭС, в частности, технические системы, функционируют в соответствии с заданными (неизменными) алгоритмами. Управление цифровой экосистемой как мультиагентной может стать «интеллектуально избыточным», а то и вовсе нереализуемым.

В данной диссертационной работе предлагается подход к анализу ЦЭС, основанный на исследовании условий ее устойчивого функционирования как мультиконсортной системы.

1.3. ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ ДВОЙНИКОВ В КОНСОРТ-СЕРВИСАХ УПРАВЛЕНИЯ

1.3.1. Обзор литературы по разработке цифровых двойников

Технология Digital Twins (DT) стала одной из десяти основных стратегических технологических тенденций, по мнению всемирно известной

консалтинговой компании Gartner Group в 2017 году [32]. В концепции DT представлена «объединенная проекция» физического и виртуального объектов, посредством которой промышленный продукт получает динамическое цифровое представление. На протяжении всего жизненного цикла разработки продукта, начиная с этапа проектирования и заканчивая этапом развертывания, организации (компании и предприятия) могут иметь полный «цифровой отпечаток» своих продуктов.

Для генерации DT для различных сфер экономики используются одни и те же наработки в таких областях, как: большие данные (Big Data), искусственный интеллект (Artificial Intelligence– AI), машинное обучение (Machine Learning – ML), имитационное моделирование, интернет вещей (Internet of Things – IoT). Цифровые двойники играют ключевую роль в Industry 4.0 [33] и в основном используются в управлении технологическими процессами и оптимизации бизнес-пространства предприятия [33, 34].

Количество публикаций по теме цифровых двойников стремительно увеличивается. Значительный процент публикаций посвящен инженерным областям (производство, робототехника, цепочки поставок) и области компьютерных наук (моделирование, проверка архитектуры, связь, аналитика). Согласно [35], первое появление цифровых двойников было в программе «Apollo» NASA (программа «Аполлон» НАСА, США). Они были определены как «Интегрированное мультифизическое, многомасштабное, вероятностное моделирование транспортного средства или системы, которое использует доступные физические модели, обновления датчиков, историю парка и т. д., чтобы отразить жизнь своего летающего двойника» [35].

По мере развития технологий и цифровизации понятие цифровых двойников получило различные интерпретации. В [45] показано, что цифровой двойник состоит из физического продукта, виртуального продукта и связанных данных, которые связывают физический и виртуальный продукт, а в [37] цифровой двойник

определяется как фреймворк, состоящий из физического пространства, цифрового пространства и уровня соединения. Кроме того, в [38] авторы создают высокоточные виртуальные модели для физических объектов, чтобы имитировать их поведение. В [39] предлагаются технические средства моделирования с целью интеграции различных компонентов модели для моделирования почти любого аспекта сложной системы. В [40] авторы обнаружили, что все данные, сгенерированные на основе цифровых двойников, обмениваются между различными системами с помощью нестандартных протоколов связи, а затем предлагают использовать программное обеспечение для реализации возможности соединения различных подсистем цифровых двойников.

Многоаспектный анализ создания, функционирования и применения DT представлен в работах [37, 38, 42, 45]. В основном, авторы исследований придерживаются «кибер-физической концепции DT плюс интернет вещей», зачастую с использованием 3D-моделей или симулятора. Цифровой двойник должен состоять из реального физического объекта (объект моделирования), его «цифровой копии» и интерфейсов взаимодействия для передачи данных. При этом существует множество проблем при внедрении цифровых двойников.

В работе [46] было предложено разграничивать понятие цифрового двойника, «непосредственно связанного с самим изделием» (в этом случае применяется термин «цифровой двойник изделия»), и т.н. «цифровой двойник производственной системы», который включает в себя «инжиниринговую» и «эксплуатационную» цифровые модели. Эти цифровые двойники должны быть функционально связаны между собой (Interoperability) и «обеспечивать эксплуатационные характеристики проектируемого и изготавливаемого изделия в соответствии с его назначением».

Инжиниринговая модель производственной системы включает цифровое описание ресурсов предприятия, технологическую структуру производств,

номенклатуру и технологии изготовления изделий, систему сбора информации о текущем состоянии оборудования, и т.п.

Эксплуатационная модель производственной системы является, по сути, «цифровой платформой для описания логистической архитектуры предприятия, формирования планов-графиков изготовления изделий, межцеховой и внешней кооперацией, включая регламенты технического обслуживания и ремонта оборудования». В эту модель включается также «динамика внутрицеховых материальных потоков, на основе цифровизации которых формируются оптимальные производственные расписания выполняемых работ».

1.3.2. Анализ проблем внедрения цифровых двойников

Цифровые двойники, анализируя данные, полученные от датчиков в режиме реального времени, представляют практически все аспекты свойств и состояний продукта, процесса или услуги. Многие крупные компании уже используют цифровые двойники для выявления проблем и повышения эффективности. Большинство двойников применяются в крупных компаниях, поскольку их разработка является сложной и дорогостоящей.

При внедрении цифровых двойников могут возникнуть проблемы: унификации данных, обмена, отсутствия стандартов взаимодействия и общего физического и виртуального пространств. Также проблему при решении задач управления с использованием DT представляет разрозненное владение данными. При создании высокоточного цифрового двойника в ЦЭС промышленного предприятия может оказаться необходимым взаимодействие технологов, инженеров, разработчиков моделей и систем, программистов и др.

1.3.3. Цифровые двойники в консорт-сервисах управления и моделирования производственных ситуаций

Цифровые двойники могут быть использованы для моделирования производственной «ситуации». Под ситуационным управлением (Situational

Control) ранее понималась методология управления сложными полу-структурированными техническими, организационными и социально-экономическими объектами с использованием методов искусственного интеллекта [47]. Позднее появилось множество методов и алгоритмов ситуационного управления. Все их объединяет то, что «идентификация ситуации» для объекта управления осуществляется путем сравнения ситуации на объекте, представленном в виде набора признаков, с различными эталонными ситуациями (набором эталонных признаков).

Алгоритмы ситуационного управления в соответствующих инженерных (и не только) задачах обычно используют логико-лингвистические, нечеткие, нейро-нечеткие и другие подходы на основе ИИ (искусственного интеллекта), а не традиционные интегро-дифференциальные и конечно-разностные уравнения. Это направление исследований выглядит конструктивным, все более перспективным и наиболее (а иногда и исключительно) приемлемым для большого класса объектов управления. Во многих недавних публикациях обсуждается эта область, называемая синтезом систем с ситуационной осведомленностью [51].

В настоящей работе предлагается использовать идентификатор в цепи обратной связи системы управления в качестве цифрового двойника для моделирования и прогнозирования производственных ситуаций в консорт-сервисах управления. Также предложен подход к формальному описанию и решению задачи ситуационного управления мультиконсортной ЦЭС. На основе предварительного обучения с использованием базы индуктивных знаний (закономерностей, извлекаемых из исторических данных посредством интеллектуального анализа), автоматически анализируется текущая ситуация и находится «сходная» по определенному критерию ситуация на основе анализа архивных данных исследуемой системы. На основе использования ассоциативных идентификационных моделей осуществляется прогнозирование и поддержка принятия решений по управлению в текущей ситуации.

1.4. МОДЕЛИ, ФОРМИРУЕМЫЕ ИДЕНТИФИКАТОРОМ

Принято выделять два типа моделей: identification - индуктивный подход, который представляет из себя обобщение анализа реальных данных в системе управления, и modeling – дедуктивный подход, который предполагает переход от общих законов описания объекта (механических, физических и др.) к конкретной математической модели [48-49].

Идентификация была признана самостоятельной научной теорией в середине прошлого столетия. Значительная роль в этом принадлежит профессору Н.С. Райбману [49-50]. В идентификации нашли отражение методы математической статистики, системные принципы, а также теории оценивания. Затем стремительно развивались методы, которые интерпретируют кибернетический и информационный подходы к построению моделей объектов техники, социума, природы [51, 52].

Наиболее известны в области теории и методов идентификации классические работы П.Эйкхоффа [48], Д. Гроппа [53], Э.П. Сэйджа и Дж.Л. Мелсы [54-55], Я.З. Цыпкина [51], Н.С. Райбмана [51-52] Л. Льюнга [89-90], и других .

Становление структурной идентификации происходило по мере развития статистических методов при решении задач оценки параметров [56]. В рамках этого подхода неопределенность в поведении объектов интерпретировалась как случайность, а структура модели полагалась заранее известной. Вероятностные распределения помех и возмущений считались заранее известными.

Особое место занимают фундаментальные работы Я.З. Цыпкина, создавшего информационную теорию идентификации [51]. В рамках теории разработаны алгоритмы и методы идентификации, которые являются оптимальными на определенных классах объектов, а также внешних возмущений, в зависимости от имеющейся априорной информации об объекте управления. Так были очерчены предельные возможности создания аппроксимационных моделей управления.

Описаны научные подходы к разработке систем управления с идентификатором, по терминологии Я.З. Цыпкина - систем непрямого адаптивного управления. В частности, разработаны эффективные алгоритмы идентификации линейных объектов с дискретным временем и аддитивным возмущением [51, 57-61], а также изучена проблема грубости системы управления по отношению к структурным изменениям (пертурбациям) внутренних моделей, которые используются во время построения расчетных регуляторов.

В [61, 62] авторы реализовали идентификационный подход к синтезу робастно-оптимальных систем управления для определенных классов внешних возмущений, таких как марковские последовательности, дискретный белый шум, мартингалы. В [58-62] авторы исследовали методы синтеза основного контура в линейных системах с неполной информацией о статистических характеристиках внешних возмущений, а также провели исследование вырожденных задач и условий вырожденности линейно-квадратичных задач синтеза для линейных объектов со стационарными возмущениями.

Получила развитие и реальное применение идея управления с прогнозирующей моделью профессора И.И. Перельмана [63]. Прогнозирующая модель может быть интерпретирована как настраиваемый фильтр, который дает прогноз для выходной переменной объекта управления на основе данных наблюдений.

В последние годы технология управления на основе прогнозирующей модели объекта стала наиболее популярным классом систем управления для сложных технологических объектов (объектов с множеством одновременно соблюдаемых ограничений, нелинейных объектов, объектов с взаимосвязанными регулируемымыми переменными, объектов со значительным транспортным запаздыванием) [64]. Данный подход предполагает наличие заранее построенной математической модели объекта в контур автоматического управления. Model-based predictive control (или сокращенно Model predictive control) – термин, буквально переводимый

как «Прогнозирующее управление на основе модели», «предикт-контроллер» и «многопараметрический контроллер». Системы управления с использованием прогнозирующей модели получили название систем усовершенствованного управления (APC – Advanced Process Control). APC реализуют управление на основе виртуальных анализаторов.

В 70-е гг. была разработана концепция систем управления с идентификатором в цепи обратной связи [65-66]. В отличие от APC, в таких системах прогнозирующая модель может адаптивно настраиваться в режиме реального времени во время эксплуатации системы управления. Идентификатор играет роль датчика параметрических возмущений квазистационарного объекта и управляет коэффициентами регулятора [48]. Впервые промышленная система с идентификатором в цепи обратной связи, используемым для настройки компенсатора, была внедрена в системе управления точностью прокатки труб [66]. При построении моделей технологических процессов непрерывных и полунепрерывных производств требуется решение задачи идентификации в условиях малых выборок наблюдений. Для такого случая выбор алгоритмов идентификации должен быть обусловлен скоростью затухания переходных процессов в идентификаторе, а также невысокой чувствительностью к погрешности начального приближения. Данные условия и необходимость выполнения условия квазистационарности затрудняют практическую реализацию таких систем [50].

Иногда отсутствие полноты информации об объекте приводит к необходимости робастного управления, которое гарантирует функционирование объекта в рамках технологических требований с соблюдением заявленного качества продукта в допустимых диапазонах, прописанных в специальных регламентах [67].

Повышение требований по быстродействию и обработке больших данных в режиме, близком к реальному времени, повышение точности прогнозов, а также управление в условиях разрозненных данных и неопределенности при наличии

различного рода возмущений в системах управления производственными процессами, электроэнергетике и логистике показывает недостаточность возможностей традиционных подходов к синтезу систем автоматического управления. Появились многочисленные методы, реализующие участие в управлении производством интеллекта человека [68-73].

В последние годы в лаборатории идентификации систем управления развивается метод идентификации, основанный на построении виртуальных моделей [74-88]. Термин «виртуальный» многие авторы объясняют как проявление принципа «здесь и сейчас», означающий существование такой модели только тогда, когда действует порождающая ее реальность. Метод, получивший название ассоциативного поиска, предполагает не аппроксимацию реального процесса во времени, а построение прогнозирующей модели динамического объекта на каждом такте с использованием наборов архивных данных («ассоциаций»), базы знаний, сформированных на этапе обучения, которая, при этом, постоянно пополняется новыми данными. Такой подход согласуется с мнением одного из ведущих специалистов по идентификации систем Л. Льюнга [88-89] о том, что необходимо использовать дополнительную априорную информацию об объекте в форме обучения, однако при этом обучение становится этапом идентификации.

Совокупность цифровых моделей, созданных к текущему моменту, полностью определяется идентификатором системы, поэтому идентификатор также может квалифицироваться как цифровой двойник.

1.5. АНАЛИЗ СИСТЕМ КЛАССА БОЛЬШИХ ДАННЫХ

Большие данные представляют собой «комбинацию структурированных, полуструктурированных и неструктурированных данных, собранных организациями, которые можно добывать для получения информации и использовать в проектах машинного обучения, прогнозном моделировании и других аналитических приложениях» [91].

Системы, которые обрабатывают и хранят большие данные, стали общим компонентом архитектур управления данными в компаниях и на предприятиях. Определяющими индикаторами больших данных являются характеристики «3V»: большой объем (volume) данных во многих средах, большое разнообразие (variance) типов данных, хранящихся в больших системах данных, и скорость (velocity), с которой данные генерируются, собираются и обрабатываются. Эти характеристики были впервые отмечены Дагом Лейни (Douglas Laney), Gartner продолжил их популяризацию [91]. Совсем недавно, несколько других «V» были добавлены к различным описаниям больших данных, включая достоверность, ценность и визуализацию данных [92]. Хотя большие данные не сопоставляются с каким-то определенным уровнем объема данных, при развертывании больших данных часто используются терабайты (ТБ), петабайты (ПБ) и даже эксабайты (ЭБ) данных, накопленных с течением времени.

Большие данные также включают в себя широкий спектр типов данных, включая следующие:

- структурированные данные в базах данных и хранилищах данных на основе языка структурированных запросов (SQL);
- неструктурированные данные, такие как текстовые файлы и файлы документов, хранящиеся в кластерах Hadoop или в системах баз данных NoSQL;
- полу-структурированные данные, такие как журналы веб-сервера или потоковые данные с датчиков [92].

Таким образом, методы и технологии больших данных, в том числе в производственной сфере, приобретают всю большую актуальность, и их использование для решения различных прикладных производственных задач управления ЦЭС становится важнейшим фактором не только для повышения надежности и устойчивости консортов управления, но и для повышения быстродействия работы всей системы управления ЦЭС.

Анализ систем In-Memory

Системы управления базами данных в оперативной памяти (системы «баз данных в памяти» - In-Memory Data System, IMDS) — это растущий сегмент мирового рынка баз данных. Создание СУБД в памяти явилось ответом на появление новых задач, стоящих перед приложениями, новыми системными требованиями и операционной средой.

СУБД в памяти — это система управления базами данных, которая хранит информацию непосредственно в оперативной памяти [94]. Это резко контрастирует с подходом традиционных СУБД, которые предназначены для хранения данных на стабильных носителях. Поскольку обработка данных в ОЗУ происходит быстрее, чем доступ к файловой системе и чтение информации из нее, СУБД в памяти обеспечивает на порядок более высокую производительность программных приложений.

СУБД в памяти могут прекрасно масштабироваться до размеров, превышающих терабайты. На основе анализа отчета [95] об использовании СУБД в памяти можно сделать вывод, что они показывают почти линейную горизонтальную масштабируемость, и многие компании уже используют такие системы для быстрой обработки данных и вычислений в памяти, где требуется минимальная задержка. Помимо этого, также используются вычисления в памяти для глубинного обучения [96], однако в данной диссертационной работе предлагается постоянно пополнять базу знаний (In-memory хранение) и переобучать модель только тогда, когда это необходимо исходя из предварительного анализа входящих данных.

Таким образом, технологии вычислений в памяти являются основным методом хранения данных для возможности быстрого переобучения интеллектуальных консортов-решателей целевых задач управления. Только на основе быстрого доступа к данным – чтения и записи – возможно не только переобучить модель в режиме, близком к реальному времени, но и делать это на постоянно пополняемой базе знаний. Данный класс систем позволит осуществлять горизонтальное масштабирование при увеличении количества данных.

Анализ систем обмена сообщениями

Системы обмена сообщениями представляют собой распределенную систему, основанную на асинхронном обмене сообщениями между компонентами системы, в то время как промежуточное программное обеспечение, ориентированное на сообщения, представляет собой продукт, на основе которого построена система обмена сообщениями. В отличие от традиционных систем, в системах обмена сообщениями приложения взаимодействуют не напрямую, а через MOM (Message Oriented Middleware) – специализированное связующее программное обеспечение, представляющее собой асинхронные системы взаимодействия сервера и клиента. Если один компонент системы хочет отправить сообщение другому компоненту, то он отправляет это сообщение в MOM, а затем MOM пересылает его получателю [97].

В более ранних исследованиях были произведены сравнения различных систем обмена сообщениями [98], и анализ показывает, что большинство систем являются неограниченно горизонтально масштабируемыми и отказоустойчивыми. При этом, эти системы также обладают значительной производительностью, которую можно увеличивать исходя из потребностей. Данные характеристики систем обмена сообщениями позволяют сделать систему высокопроизводительной, отказоустойчивой и горизонтально масштабируемой.

Существуют две «базовые» модели обмена сообщениями [99]:

- «точка-точка»;
- «публикация-подписка» (pub-sub).

Модель «точка-точка» применяется, когда одному или нескольким компонентам (так называемым отправителям) необходимо отправить сообщение одному компоненту получателя. Модель основана на концепции очереди сообщений. Отправители отправляют сообщения в очередь, а получатель читает сообщения из очереди (рис. 1):

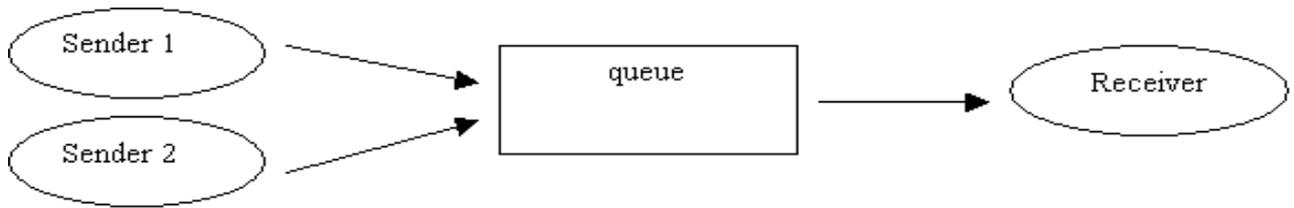


Рисунок 1. Модель «точка-точка» для систем обмена сообщениями (MOM)

Часто предполагается, что в модели «точка-точка» имеется один, и только один приемник. Однако это не совсем так. К одной очереди может быть подключено несколько получателей. Но система доставит сообщение только одному из них, а кому именно - зависит от реализации. Некоторые MOM доставляют сообщение первому зарегистрированному получателю, в то время как есть реализации, которые реализуют циклический перебор.

Издатели отправляют сообщения в определенный топик, и все указанные топика получают эти сообщения (рис. 2):

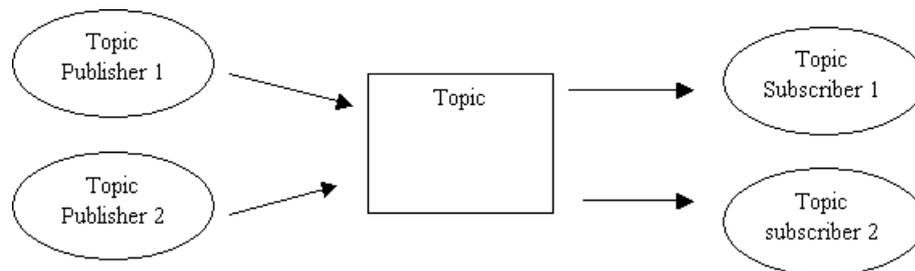


Рисунок 2. Модель публикации-подписки в системах обмена сообщениями (MOM)

Какую модель использовать, зависит от решения конкретной бизнес-задачи. В данной диссертационной работе предлагается универсальная модель, подходящая для решения большинства бизнес-задач – модель публикации-подписки. Данная модель используется для обмена данными между разными подсистемами в режиме реального времени – во-первых, это основная точка входа всех данных в систему для дальнейшей обработки; во-вторых, данная модель применяется для дальнейшей передачи в другие подсистемы только необходимых очищенных, агрегированных и пред-обработанных данных.

Так, системы обмена сообщениями очень важны для высокопроизводительной обработки данных в режиме реального времени. Данный

класс систем позволит выполнять обмен данными между различными консортами системы управления ЦЭС, а также, при необходимости, в рамках одного типа консорта в режиме реального времени, а высокая производительность и горизонтальная масштабируемость таких систем позволят масштабировать систему исходя из необходимых потребностей.

1.6. УПРАВЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ИНФРАСТРУКТУРОЙ НА ОСНОВЕ ВИРТУАЛИЗАЦИИ И КОНТЕЙНЕРИЗАЦИИ

Управление на основе виртуализации

Виртуализация — это технология создания виртуальных экземпляров компьютерных ресурсов для многократного использования одного и того же физического ресурса. Существует несколько технологий виртуализации, которые способны виртуализировать сервер, хранилище, сети и операционные системы. Технология виртуализации известна своей простотой настройки и оптимизацией затрат на настройку среды с более высокой эффективностью обработки. Среда виртуализации разделяла фактические компоненты ресурсов, такие как память (RAM), дисковое пространство и сеть, как отдельную группу ресурсов. Виртуализация помогает организациям масштабировать вычислительные ресурсы. Виртуальные машины (VM) — это популярное использование аппаратной виртуализации, которой управляют через гипервизор [99]. Это одна физическая машина с несколькими виртуальными машинами.

Можно выделить следующие преимущества виртуализации [99, 100]:

- экономия места и эксплуатационных расходов;
- простое управление дата-центрами;
- увеличение производительности ИТ-подразделений;
- повышение надежности и отказоустойчивости системы в случае сбоев;
- уменьшение времени выделения ресурсов при запросе пользователей;
- отсутствие необходимости устанавливать аппаратные компоненты;

- автоматическое обновление аппаратного и программного обеспечения провайдерами;
- снижение потребления энергии, поскольку на физических серверах потребляется меньше энергии;
- эффективная утилизация всех вычислительных ресурсов.
- Также можно выделить следующие недостатки виртуализации:
- не каждое приложение или сервер будут работать в среде виртуализации;
- необходимо инвестировать в обучение существующих сетевых администраторов, у которых нет возможностей управлять виртуальной сетью;
- в виртуализированной системе, когда что-то не так, требуется сложное устранение неполадок;
- сложно заранее оценить необходимые дополнительные ресурсы.

Так, виртуализация помогает утилизировать ресурсы более эффективно, что на масштабе предприятия может помочь существенно снизить затраты. В данной диссертационной работе на базе виртуализации предлагается запускать контейнеры с приложениями, в которых будут работать независимые от внешней среды сервисы (агенты) и цифровые двойники, а совокупность цифровых сервисов и двойников будут определять цифровую экосистему.

Управление на основе контейнеризации

Виртуальные машины поставляются с целым рядом проблем с точки зрения управления выделенными ресурсами. Они упрощают задачу работы в постоянно меняющихся центрах обработки данных, но их сложно масштабировать. Возможность запуска четырех виртуальных машин на одном специализированном оборудовании — эффективное решение в случае дефицита мощностей [101].

Во многих отношениях контейнеризацию лучше всего рассматривать как естественную эволюцию виртуализации. Виртуализация рассматривает каждую виртуальную машину как свой собственный логический (но не физически)

отдельный сервер. Контейнерная обработка рассматривает каждое приложение как свой собственный логически отдельный сервер. Несколько приложений будут использовать одну базовую операционную систему. Эти контейнеры «не знают», что другие контейнеры работают на их выделенном оборудовании. Если они хотят «общаться» с другим сервером, им нужно «общаться» через сетевой интерфейс, как если бы они находились на разных физических устройствах [102].

Таким образом, преимущество контейнеризации заключается в том, что нет необходимости выделять аппаратные ресурсы для избыточных операций. Вместо того, чтобы выделять ядра процессора и память для операционной системы для каждой виртуальной машины, создается сервер с одной базовой виртуальной машиной и выделяются ядра и память только для одного приложения, которое выполняется в каждом контейнере.

Независимо от того, где он создается, на каком оборудовании или в какой операционной системе, контейнер будет работать одинаково. Такая согласованность устраняет целый класс ошибок. Если многие разработчики привыкли делать моментальные снимки виртуальных машин, легко представить себе контейнер таким образом. Каждый раз, когда запускается контейнер, он восстанавливается до исходного снимка независимо от того, какие действия были выполнены в последний раз.

Управление контейнеризацией на основе Docker

Docker - самая популярная контейнерная система в мире. Это не единственная библиотека, поддерживающая контейнеризацию, но она самая большая. Прямо сейчас, на долю Docker приходится более 70% всех производственных контейнеров [103].

На основании [104-106] можно выделить следующие рекомендации по использованию контейнеров:

- каждый контейнер должен иметь только одно приложение. Контейнер базы данных, в котором работает только база данных, может запускать версию 2.2 какой-либо сложной библиотеки, а очередь сообщений, для которой нужна версия 2.5 – будет запущена в другом контейнере. Они работают на одном и том же оборудовании и даже в одной и той же операционной системе, но никогда не используют те же библиотеки.

- Оптимизация сборки для кэша сборки. Когда создается контейнер, контейнерный движок проходит по каждому шагу в определении и создает «слой» контейнера.

- Создание наименьших возможных изображений. Это еще один способ «взлома» процесса контейнера Docker для упрощения этапов сборки. Каждая команда, которая вводится в контейнер Docker, добавляет еще один слой.

- Использование преднастроенных контейнеров для экономии места. Еще один способ оптимизировать размер контейнеров - использовать так называемые преднастроенные изображения. Большинство оперативных команд не устанавливают каждый отдельный контейнер Docker с нуля. Вместо этого они используют предварительно упакованные контейнеры в качестве основы, а затем строят надстройки поверх них.

- Использование преимуществ Docker Hub или реестра. Рекомендуется использовать изображения, которые другие люди создавали ранее.

На основе анализа контейнерных технологий использование Docker является перспективным для управления различными информационными сервисами системы управления. В контейнере возможно разворачивание не только самих консорт-сервисов системы управления, но и также сервисов инфраструктуры - баз данных (statefull-сервисы), фреймворков интеграций (stateless-сервисы) и визуализаций.

1.7. АНАЛИЗ ВОЗМОЖНОСТЕЙ СЛИЯНИЯ РАЗНОРОДНЫХ ДАННЫХ

Проблемы растущего объема данных, поступающих из Интернета вещей (IoT), не теряют актуальности уже более десяти лет. Концепция IoT достигла уровня массового использования в промышленности в начале 2014 года, и ее повсеместное влияние и проблемы внедрения и эксплуатации по-прежнему активно обсуждаются в научной литературе. Методы обработки больших данных стремительно развиваются и совершенствуются, в результате чего уже было получено множество решений различных проблем в области Интернета вещей [107].

Необходимость быстрого, эффективного и точного анализа этих данных все еще остается сложной нерешенной с методологической точки зрения проблемой [108]. Поточковая обработка данных [109] появилась как потенциальный ответ на анализ больших данных в режиме реального времени (близкого к реальному), в том время как инкрементальное обучение затронуло решение некоторых проблем, в том числе скорости переобучения моделей [110, 111]. В течение последнего десятилетия тема инкрементального переобучения моделей также стремительно развивалась, внедрены различные программные библиотеки, однако развитию методологии было уделено мало внимания. В результате анализа исследования авторов в различных сферах, таких как экология, транспорт, управление водными ресурсами, энергоэффективность и моделирование интеллектуальных сетей, были выявлены следующие недостатки:

- наиболее полная библиотека программного обеспечения в работе [112] для потоковой передачи — это академический проект, который требует значительных дополнительных усилий при переносе в промышленную среду;
- многие научные работы проводились по инкрементному обучению, и не уделялось внимания методологии и архитектуре систем управления цифровыми экосистемами на постоянно пополняемой базе знаний.

Отсутствие оперативных методов предварительной обработки данных также снижает вероятность использования гибридных подходов, в которых предварительная обработка данных осуществляется в режиме онлайн, а модели прогнозирования реализуются, используя традиционные подходы машинного обучения (*machine learning* – ML). По оценкам McKinsey, до 40% ценности данных, возникающих из IoT, скрыто в синергетических эффектах различных систем [113]. За исключением инфраструктуры интеграции потоковых данных IoT [114], где решаются вопросы временного выравнивания для интеграции данных, общая методология и архитектура для создания и эффективного функционирования систем управления цифровыми экосистемами с применением IoT в режиме реального времени еще не существует. Предлагаемая в диссертации методология и архитектура, в частности, предусматривают разработку методов объединения разнородных потоковых данных от разных источников – показания датчиков технологических процессов на промышленных предприятиях.

Слияние разнородных данных от разных источников в ряде случаев необходимо для построения эффективных моделей объектов и систем. Слияние данных используется не только для построения производственных процессов в промышленности, но также применимо: для позиционирования и навигации [115–117], для распознавания активностей [118, 119], в системах мониторинга и диагностики различных неисправностей [122–127], на транспорте [132], в здравоохранении [127] и в других отраслях. Например, в здравоохранении слияние данных используется в настройках IoT, таких как удаленный пациент в системе мониторинга. Здесь пациент контролируется с помощью различных датчиков тела и окружающей среды, сигналы обрабатываются и используются для информирования врачей о состоянии пациента.

Слияние также используется для вычисления понимания контекста, который используется для назначения динамических ролей участников процесса. Использование нечеткой логики в контексте понимания обсуждается в [127].

Контроль доступа играет важную роль в управлении данными и является яркой темой в недавних научных публикациях [127–130]. Предлагаемая в данной диссертации методология и архитектура основаны на механизмах, описанных в литературе, и могут использовать преимущества недавних выводов в работах, особенно связанных с потоковыми платформами. В [130] обсуждается обработка отсроченных или непоследовательных измерений. Многие из систем также включают в себя значительные формализованные знания предметной области (модельно-ориентированные подходы) в модели объединения данных [131], при которой модели теряют свой потенциал обобщения. Каркасы и архитектура, которые могут применяться в различных случаях использования, должны быть независимы от предметной области и управляться исключительно данными [132]. Идея получила дальнейшее развитие с помощью машин слияния гетерогенных признаков [133], однако задача создания методологии и архитектуры для системы управления цифровыми экосистемами в режиме реального времени остается за рамками работы.

В больших наборах и потоках данных предварительная обработка данных и их сокращение становятся критическими моментами для извлечения знаний [114, 134]. В данных работах авторы обсуждают важную роль эффективности систем машинного обучения для режима реального времени.

При построении моделей автоматический анализ данных бесполезен, если предварительная обработка данных требует ручного вмешательства [135]. Авторы представляют адаптивную предварительную обработку, которая повышает конечную точность прогноза на основе реальных сенсорных данных. В данной диссертационной работе предлагается использовать ту же стратегию, однако делается акцент на построении масштабируемой архитектуры системы для переобучения прогнозирующих моделей на основе постоянно пополняемой базы знаний в режиме реального времени, в то время как в работе [135] рассматривается адаптация к изменениям на входе потоков данных.

Стоунбрекер и соавторы [136] определили восемь требований к механизму обработки потоков (SPE) уже в 2005 году. Среди них есть требование, что для обработки *недостатков* потока (то есть задержанных или отсутствующих данных) необходимо интегрировать хранимые и потоковые данные. Также сформулированы требования для их перемещения, обработки и реагирования. Данные должны перемещаться мгновенно, что означает: в режиме реального времени. Механизмы потоковой обработки часто демонстрируют свои возможности для моделирования на дополнительных методиках обучения [82, 137]. Самым популярным методом инкрементного обучения по-прежнему является очень быстрый метод дерева решений (VFDT) [91], который был модернизирован много раз за эти годы. Альтернатива, обеспечивающая более быстрое обучение (возможность быстрее достичь лучшей точности) и сводится к пакетной обработке данных — это чрезвычайно быстродействующее дерево решений (EFDT) [139]. Вертикальные деревья Hoeffding (VHT) являются первыми алгоритмами распределенной потоковой передачи для деревьев решений и предлагают значительно улучшенную скорость вычислений по сравнению с VFDT и EFDT [140]. Много усилий было также посвящено инкрементальному обучению в области глубокого обучения [141]. С сетевыми архитектурами, которые включают в себя модули кратковременной памяти (LSTM), проблемы слияния разнородных данных могут быть как минимум частично решены уже в рамках метода обучения. Оценка дополнительных методов обучения обычно достигается с помощью подхода предварительной оценки [142]. Данный подход также может быть реализован на основе предлагаемой в данной диссертации архитектуры.

Некоторые архитектуры пакетной обработки больших данных возникали на базе решений экосистемы Apache Hadoop. В последнее десятилетие возникли ряд популярных высокопроизводительных решений с открытым исходным кодом на базе Apache, таких как Apache Spark [143], Apache Samza [144], Apache Flink [145] и Apache Apex [146]. Кроме того, стремительно развивались системы обмена сообщениями в качестве основы Lambda-архитектуры (архитектура с выделением

двух параллельных уровней обработки данных – пакетного и потокового), на основе нее моделировались фреймворки для мобильных сетей [147], а также предлагались решения для преодоления сложных разнородных данных для тех же мобильных сетей с применением «умного» мониторинга [148]. Помимо этого, активно развивается сфера использования систем больших данных на основе открытого исходного кода. Так, в [149] моделируется архитектура обработки данных ключевых показателей эффективности одной крупной транспортной компании на основе систем открытого исходного кода, а в [150] моделируется рабочий процесс (workflow) с применением распределенных вычислений на основе тех же систем открытого исходного кода.

Ни один из упомянутых выше инструментов, ориентированных на производство, логистику или обработку мобильных данных, не предполагает внедрения унифицированной архитектуры и методологии построения информационной инфраструктуры для переобучения прогнозирующих моделей на основе постоянно пополняемой базы знаний в режиме реального времени. Большинство хорошо известных инструментов для потокового майнинга относятся к классу MOA (Massive Online Analysis) [151] и его клонов, реализованных с помощью других языков программирования (например, stream DM-cpp [151] и scikit-multiflow [152]). Хотя эти инструменты и обеспечивают реализацию современных алгоритмов потокового обучения, они полностью игнорируют необходимость для оперативной предварительной обработки данных и объединения потоковых данных.

QMiner [153] — это механизм обработки потоков (SPE), который предлагает использовать различные операторы для агрегирования потоков данных, а также операторы для объединения и повторной выборки нескольких потоков данных. В [154] авторы реализуют свою методологию поверх инфраструктуры QMiner и расширяют ее функциональность для поддержки гетерогенной потоковой передачи слияния данных. В отличие от данной работы, в данной диссертации предлагается

не инкрементальное переобучение моделей лишь на основе новых векторов входов, а на основе всей базы знаний.

Концептуальная платформа [155] для использования потокового майнинга в области больших данных описывает подход на основе *лямбда-архитектуры* [147, 148, 156]. Авторы перечисляют все соответствующие технологии и методы для сводки потоковых данных, но сама платформа не перечисляет основных свойств ее архитектуры для реализации объединения данных в режиме реального времени.

Вероятностное слияние данных в реальном времени для крупномасштабных приложений IoT [119] демонстрирует использование мультимодальных потоков данных (данных IoT, потоков данных о погоде и открытых данных социальных сетей) для эффективного прогноза заторов на дорогах. Метод реализует двухуровневую архитектуру, где первый уровень – аналитика – извлекает события из потоков данных, а второй уровень представляет собой вероятностный программный комплекс – обработчик данных событий. Комплекс генерирует правила для эффективной пакетной обработки событий. Он также решает проблему использования общей шкалы времени для разнородных источников данных и с помощью внутренних алгоритмов исключает возможности задержки измерений. Аналогичный подход описан в [157], где авторы формализуют понятие «машинное обучение» в представленной архитектуре, где оно используется для поддержки обработки событий, предоставляя правила обработки событий с помощью встроенных аналитических инструментов. Оба подхода, однако, выполняют эту операцию в режиме пакетной обработки данных, что не является потоковой обработкой, и задержки обработки в этом случае являются критическими факторами для многих сфер, где несколько секунд могут стать решающими для принятия управленческого решения на основе полученного прогноза модели.

Слияние нескольких потоков представлено в [158]. Методология использует несколько датчиков для измерения того же свойства объекта для прогнозирования аномалий и не затрагивает проблему слияний разнородных потоков данных.

Парадигма интеграции потоковых данных IoT (ISDI) представлена в [124], и предложенная структура ISDI решает интеграцию данных в реальном времени, используя общий *оконный алгоритм*. В работе рассматривается ключевая проблема синхронизации времени в настройке IoT. В то время, как и ISDI, и предлагаемая в диссертации универсальная архитектура, направлены на решение схожих проблем, однако предлагаемая в диссертации архитектура и подход направлены также на интеллектуальный анализ исторических данных, постоянно накапливая и расширяя базу знаний.

Слияние данных является одной из центральных тем исследований в области IoT, однако до сих пор нет универсальных методологий и архитектур, а также платформ для объединения разнородных потоковых данных с источников и сенсоров и их предобработки, переобучении предиктивных моделей в режиме реального времени на постоянно пополняемой базе знаний.

Предлагаемая методология и архитектура построения системы управления призваны быть применимыми для любых форматов данных, их скорости поступления и объемов. Возможность горизонтального масштабирования и отказоустойчивость консорта слияния данных вычислительной инфраструктуры повысит надежность всей системы управления.

Анализ проблем слияния разнородных данных

Одним из сценариев использования больших данных в сфере IoT является сценарий использования этих данных для построения прогнозирующих моделей. Например, на основе исторических данных «умной» энергосети можно построить модель, способную прогнозировать профили энергопотребления на следующий день, что поможет лучше спланировать распределение энергии между потребителями и, следовательно, обеспечить более дешевую энергию для конечного пользователя.

Для построения высокоточной прогнозирующей модели может оказаться необходимым использование большого количества разрозненных данных – в

разном формате, поступающих с разной периодичностью, разных объемов и типов. Примером такого объекта может быть цифровая экосистема – сложный социотехнический объект управления.

Для переобучения прогнозной модели управления все данные сначала должны быть предобработаны (для соответствия необходимым требованиям разметки данных), агрегированы, объединены и совмещены с историческими данными той же структуры. Возникает проблема определения такой методологии и архитектуры, при которых будет четко определен набор компонент архитектуры, определены основные подсистемы и правила их взаимодействия, смоделирована архитектура и информационные потоки между подсистемами, при которых система будет устойчивой, масштабируемой, надежной, высокопроизводительной и способной с минимальными задержками проводить переобучение модели на основе как новых векторов входов, так и исторических данных с последующим бесшовным переключением рабочей в текущий момент времени модели управления.

Двумя основными причинами того, что такая задача объединения данных не является тривиальной, является неоднородность данных IoT и их временная несогласованность. Согласно [158], неоднородность является свойством больших данных. Многие определения разнородных данных можно найти в литературе, и среди разных исследователей нет единого мнения об определении. Свойства, иллюстрирующие проблему, включают: мультимодальность данных (даже с учетом сочетания непрерывных и категориальных особенностей и структурированных и неструктурированных данных) и технические аспекты (т. е. формат данных), степень независимости одних данных от других, смещение концепции, динамика изменений и безопасность. Временной аспект несогласованности данных проявляется в следующих свойствах:

- частота дискретизации;
- временная задержка;

- доступность данных.

Частота выборки отличается от датчика к датчику. Некоторые датчики обеспечивают постоянную частоту дискретизации. Различные датчики в пределах установки могут реализовывать постоянные, но разные частоты дискретизации. Многие датчики реализуют приблизительно постоянные частоты дискретизации.

На основе анализа текущих систем и литературы можно сделать вывод, что слияние потоковых и пакетных данных является сложной задачей интеграции разнородных данных. Для решения этой задачи предлагается использовать отдельный слой интеграции в архитектуре системы управления, который будет обладать следующими свойствами:

- отказоустойчивостью;
- горизонтально-масштабируемостью;
- возможностью как потоковой обработки данных и подключением к брокерам сообщений для загрузки данных с датчиков систем класса АСУТП и данных с датчиков производственного оборудования, так и пакетной обработкой через REST API для загрузки данных из транзакционных производственных систем и внешних цифровых экосистем;
- возможностью обогащения и фильтрации данных;
- возможностью нормализации входящих потоков;
- поддержкой загрузки во внешние системы хранения данных в памяти;
- возможностью устанавливать расписание загрузок данных для пакетных загрузок;
- возможность делать трансформации «на лету» между различными форматами загружаемых данных;
- поддержка работы в среде контейнеризации.

Таким образом, слой слияния данных (интеграции) в предлагаемой архитектуре системы управления должен обладать функциями выше. Реализация данного слоя интеграции для задачи прогнозирования состояния ресурсов

производства в цифровой экосистеме ПАО «КАМАЗ» описана в главе 4 данной диссертационной работы.

1.8. ВЫВОДЫ ПО ГЛАВЕ 1

В первой главе диссертационной работы приведен аналитический обзор научных исследований в сфере цифровых экосистем. Проанализировано развитие цифровых экосистем в сферах цепочек поставок, энергетики и производства. Отмечено, что цифровые экосистемы (ЦЭС) производства играют важную роль в деятельности современного промышленного предприятия, а успешное функционирование таких сложных социотехнических систем прочно ассоциировано с их реализацией в цифровой информационной среде посредством современных информационных технологий, в особенности технологий Big Data.

Участники экосистемы (называемые «консортиями») в процессе адаптации к естественным условиям существования получают дополнительное преимущество, организуя между собой многочисленные взаимосвязи. Отмечено, что консорты могут быть трех основных типов в системе управления: консорты-решатели целевых задач, интерпретируемых как задачи управления, обслуживающие консорты обработки данных и инфраструктурные консорты. Общая стабильность функционирования ЦЭС как сложной динамической системы должна учитывать состояние каждого отдельного консорта. Обоснована необходимость разработки подхода к анализу стабильности работы интегрированной ЦЭС промышленного предприятия как мультиконсортной системы.

Анализ научных публикаций в сферах цифровых двойников, цифровых экосистем и прогнозирующих моделей управления показывает, что существует ряд исследований непосредственно в рамках данных областей, однако область анализа и синтеза цифровых производственных экосистем еще недостаточно исследована с учетом сложной структуры производственных систем, обладающих свойствами

систем класса Big Data, нестационарности и нелинейности, а также повышенными требованиями к быстродействию и точности идентификационных моделей.

Анализ развития цифровых двойников говорит о большом интересе исследователей в этой сфере в течение последних лет. Уже сейчас существует множество примеров использования данной концепции в разных отраслях экономики, и ее использование для моделирования реальных сложных технологических процессов и их взаимодействия с окружающим миром и другими сложными физическими объектами можно считать перспективным и оправданным. Предложено использовать в качестве цифрового двойника идентификатор в цепи обратной связи системы управления, формирующий цифровые модели. Отмечается актуальность его использования для моделирования производственных ситуаций в рамках основных консорт-сервисов решателей задач управления.

Проведен анализ систем с вычислениями и хранением в памяти. Данный класс систем позволит хранить данные для быстрого доступа к ним, а возможность горизонтального масштабирования таких систем позволит линейно увеличивать пропускную способность системы, исходя из возрастающих входящих производственных потоков данных. При этом переобучение виртуальной модели для консорта-решателя задачи управления осуществляется в режиме, близком к реальному времени.

Проведен обзор систем обмена сообщениями, которые будут являться основной точкой входа всех данных в систему управления ЦЭС. Они будут необходимы для дальнейшей передачи данных – как в другие (внешние) экосистемы (не только производственные), так и для внутреннего взаимодействия самих консортов. Отмечено, что в системе управления ЦЭС промышленного предприятия система обмена сообщениями должна стать центральным хабом данных, вокруг которого будут работать остальные консорты различного типа. Они будут выполнять свои функции и отправлять результат своей работы в систему обмена сообщениями для его потребления другими консортами.

Технологии виртуализации и контейнеризации помогают более эффективно управлять вычислительными ресурсами консорт-сервисов обработки данных и целевых консортов. Существует несколько технологий виртуализации, которые способны виртуализировать сервер, хранилище, сети и операционные системы. Виртуализация существенно снижает эксплуатационные расходы на инфраструктуру, которая является базисом для всех программно-вычислительных комплексов на предприятии. Так, благодаря облачным вычислениям, виртуализации и контейнеризации стало возможным использовать вычислительные мощности и инфраструктуру более эффективно, снижать вероятности ошибок и проблем при использовании одних и тех же программных решениях на разных средах и операционных системах. Данные технологии помогают решить сложные и трудоемкие процессы моделирования и управления, а также всего жизненного цикла внедрения любой модели управления, от прототипирования до внедрения в производство и эксплуатации.

Рассмотрены вопросы слияния (интеграции) разнородных данных. Показано, что в системе управления необходим отдельный консорт-уровень интеграции данных, сформулированы основные свойства инфраструктурного интеграционного консорт-слоя.

Отмечено, что ЦЭС промышленного предприятия представляют собой сложные человеко-машинные системы с развитой информационной инфраструктурой. Такие системы, по существу, являются сложными динамическими системами с большим количеством гетерогенных данных, и для построения системы управления ЦЭС промышленного предприятия необходимо учитывать современные технологии обработки и хранения больших данных. В главе приведен обзор таких технологий, как системы обмена сообщения для потоковой обработки больших данных, системы хранения данных в памяти (In-Memory), а также анализ современных достижений ИКТ в области виртуализации и контейнеризации для более эффективной утилизации ресурсов и управления консорт-сервисами, отказоустойчивостью, развертыванием системы управления и

ее масштабированием. На рисунке 3 изображены основные функциональные аспекты построения вычислительной инфраструктуры системы управления ресурсами производства в ЦЭС предприятия.

Можно констатировать, что мультиконсортный подход к синтезу производственной ЦЭС с использованием цифровых алгоритмов идентификации и прогнозирования является актуальным, поскольку позволяет учесть различные аспекты обеспечения стабильного функционирования всей системы и оптимизации решаемых целевых задач.

В третьей главе представлены архитектура системы управления ЦЭС производства и подход к обеспечению стабильного функционирования совокупности консортов в ЦЭС производства. В четвертой главе описан прототип системы управления ресурсами производства в ЦЭС ПАО «КАМАЗ» для прогнозирования состояния загрузки серверного комплекса ресурсов.

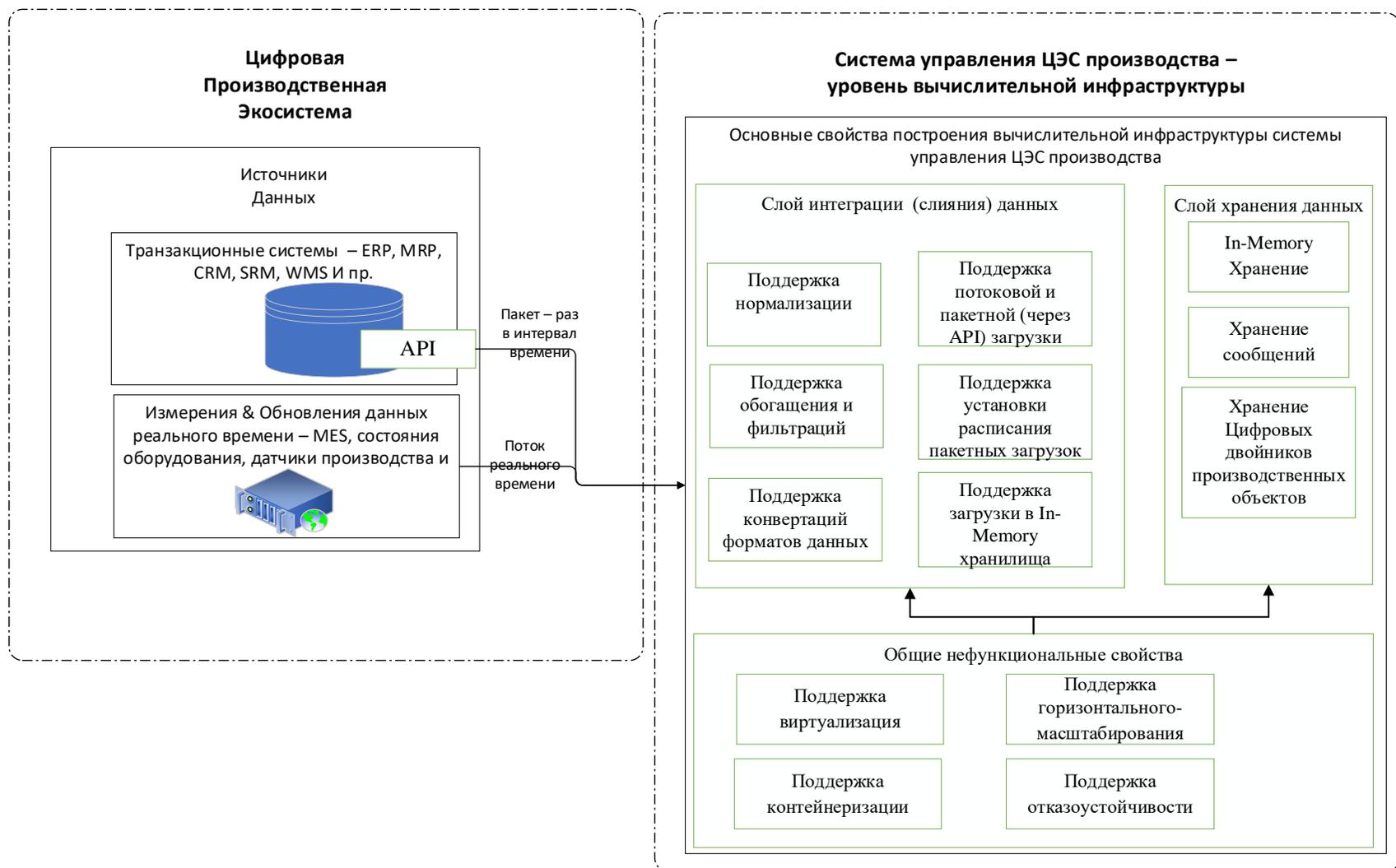


Рисунок 3. Основные функциональные аспекты построения вычислительной инфраструктуры системы управления ресурсами производства в ЦЭС предприятия

ГЛАВА 2. МЕТОДЫ АНАЛИЗА И ИДЕНТИФИКАЦИИ ЦИФРОВЫХ ПРОИЗВОДСТВЕННЫХ ЭКОСИСТЕМ КАК МУЛЬТИКОНСОРТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

2.1. УСЛОВИЯ УСТОЙЧИВОГО ФУНКЦИОНИРОВАНИЯ ЦЭС

Принцип предиктивного управления представляется наиболее перспективным для стабильного и эффективного функционирования цифровых экосистем. Реализуемость управления и работоспособность ЦЭС в общем случае достигается за счет устойчивости динамических подсистем, решающих прикладные задачи, и стабильностью (работоспособностью) вычислительной инфраструктуры системы и программного обеспечения.

ЦЭС формализуется как система взаимодействующих цифровых консортов:

$$DES = \{\langle C_1 \rangle \langle C_2 \rangle \dots \langle C_N \rangle\}, \quad (1)$$

где: $C_i, i = 1, \dots, N$ – все элементы (консорты) исследуемой ЦЭС. Среди этих консортов можно выделить: подсистемы, решающие определенные целевые задачи (это могут быть автоматические либо автоматизированные системы, либо интеллектуальные агенты); инфраструктурные консорты; вспомогательные консорты обработки данных. Таким образом,

$$DES = \{\langle C_1^c, \dots, C_{N_1}^c \rangle \langle C_1^{ic}, \dots, C_{N_2}^{ic} \rangle \langle C_1^{pc}, \dots, C_{N_3}^{pc} \rangle\}, \quad (2)$$

где:

- DES – формальное обозначение ЦЭС (Digital Ecosystem);
- $C_i^c, i = 1, \dots, N_1$ – подсистемы ЦЭС, которые решают определенные целевые задачи, и успешность этих решений интерпретируется как эффективность управления этой подсистемой;
- $C_i^{ic}, i = 1, \dots, N_2$ – консорты вычислительной инфраструктуры, на основе которых функционируют все другие консорты;

- $C_i^{pc}, i = 1, \dots, N_2$ – консорты обработки данных, которые решают задачи подготовки данных для основных консортов, функционирующие на базе вычислительной инфраструктуры.

Консорты выполняют конкретные действия – от транспортировки и обработки данных, до реализации алгоритмов моделирования и управления. Обеспечение решения подсистемой определенной целевой задачи достигается соответствующим управлением этой подсистемы.

Работоспособность функционирования ЦЭС достигается совокупным обеспечением:

1) устойчивости сложной динамической системы

$$DES^C = \langle C_1^c, \dots, C_{N_1}^c \rangle, \quad (3)$$

и, следовательно, – устойчивости консортов-решателей целевых задач как динамических систем;

2) инфраструктурной стабильности и

3) стабильности работы программного обеспечения обработки данных («обеспечивающих консортов»).

2.2. МЕТОДЫ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ЦЭС НА ОСНОВЕ ЦИФРОВЫХ АЛГОРИТМОВ ИДЕНТИФИКАЦИИ

Для систем управления, составляющих первую группу консортов в (2), эффективность и стабильность функционирования в значительной степени обеспечивается точностью идентификационных моделей [48-50].

При реализации предиктивного управления в подсистемах ЦЭС, решающих различные целевые задачи (эти подсистемы могут быть интерпретированы как системы управления), предлагается использовать цифровые предиктивные «data-driven» модели («модели, управляемые данными») реального времени, которые строятся на основе машинного обучения и интеллектуального анализа данных (Data Mining) [43]. Целесообразность использования таких моделей обусловлена

особенностью предлагаемой концепции ЦЭС, в частности, необходимостью потоковой обработки больших данных и слияния гетерогенных асинхронно поступающих данных. Применение традиционных рекуррентных моделей (обычно линейных) при работе с большими данными не представляется адекватным в условиях большого количества гетерогенных данных, нелинейности и нестационарности процессов.

Использование цифровых идентификационных моделей в качестве ядра системы и современной инфраструктуры больших данных, фреймворков и горизонтально-масштабируемых отказоустойчивых систем даст возможность не только строить точные виртуальные линейные модели в конкретный период времени для нелинейных нестационарных производственных процессов, но и повысить быстродействие в обучении таких моделей и формирования управляющего воздействия.

Для идентификации нелинейных процессов эффективное решение может быть получено, например, с помощью метода ассоциативного поиска [69] для построения моделей на основе знаний, извлекаемых из исторических данных функционирования посредством интеллектуального анализа.

Процесс обработки знаний в интеллектуальной системе сводится к обнаружению (ассоциативному поиску) знаний по их фрагменту. При этом знания могут интерпретироваться как ассоциативные связи между образами. В качестве образа используются «наборы индикаторов», в качестве которых рассматриваются компоненты входных векторов. Критерий близости между образами может быть сформулирован самым различным образом. В самом общем случае это может быть логическая функция, предикат. В частном случае, когда наборы показателей являются векторами в n -мерном пространстве, критерием близости может быть расстояние в этом пространстве.

Модель выглядит следующим образом:

$$y(k) = \sum_k [\sum_{i=0}^N a_{ik} x_i(k-j) + \sum_{j=1}^M b_{jk} u_j(k-j)], \quad M \leq N \leq k, \quad (4)$$

где: $x_i(k), i = 1, \dots, N$ – компоненты вектора входов, $u_j(k)$ – компоненты вектора управления, $j = 1, \dots, M, a_{ik}$ – коэффициенты виртуальной модели в момент времени k .

Знания (индуктивные) представляют собой закономерности, извлекаемые из данных с помощью Data Mining. В нашем случае это – цифровые модели, полученные в предыдущие моменты времени k . Они представляют собой совокупность значений переменных: входов, управлений и соответствующих им реальных значений выходов объекта, а также коэффициентов виртуальной модели, построенной ранее для момента k .

В ходе предварительного обучения системы все реализации расширенных входов (измеряемые входы и управления) распределяются на кластеры посредством одного из известных методов кластеризации. Для текущего расширенного вектора входов и соответствующих им выходов можно составить систему линейных уравнений вида (4). Модель напоминает регрессию, однако в ней участвуют только те входные векторы $x(s)$ из истории процесса, которые близки к текущему вектору $x(k)$ в смысле манхэттенского расстояния. Они выбираются из того кластера, куда попал текущий вектор. Такие входы называются ассоциациями $x(k)$.

Оценка коэффициентов a_i , и b_j , полученная методом наименьших квадратов, а также прогнозируемое контуре значение выхода $y(k)$, получаются путем решения этой системы линейных уравнений.

Эти методы получения оценок параметров в каждый момент времени называются методами ассоциативного поиска. Такие «цифровые» алгоритмы идентификации продемонстрировали высокую точность для различных приложений, таких как исследование технологических и логистических процессов

химических производств, нефтепереработки, интеллектуальных сетей, транспортных систем, и т. д. [159].

В данной работе рассматривается задача прогнозирования состояния ресурсов производства, в частности, прогнозирования значений определенных метрик загрузки мощностей производственного оборудования, таких как: процессорные ядра, графическая память процессора, загрузка оперативной памяти процессора, температура процессора, температура графического процессора, температура жестких дисков. Прогноз осуществляется на основе анализа динамики данных телеметрии серверного комплекса ресурсов. Описание решения конкретной задачи для прогнозирования состояния комплекса ресурсов ПАО «КАМАЗ» приводится в Главе 4.

На рисунке 4 изображен график прогнозирования температуры центрального процессора в течение 1 рабочего дня (с 7.00 до 16.00). График показывает высокую точность метода ассоциативного поиска для данной задачи:



Рисунок 4. Пример прогнозирования температуры центрального процессора – прогнозируемые и фактические значения на интервале с 7.00 до 16.00

Следующий график (рисунок 5) показывает прогнозирование температуры графического процессора (GPU) за 1 рабочий день в интервале с 7.00 до 16.00:

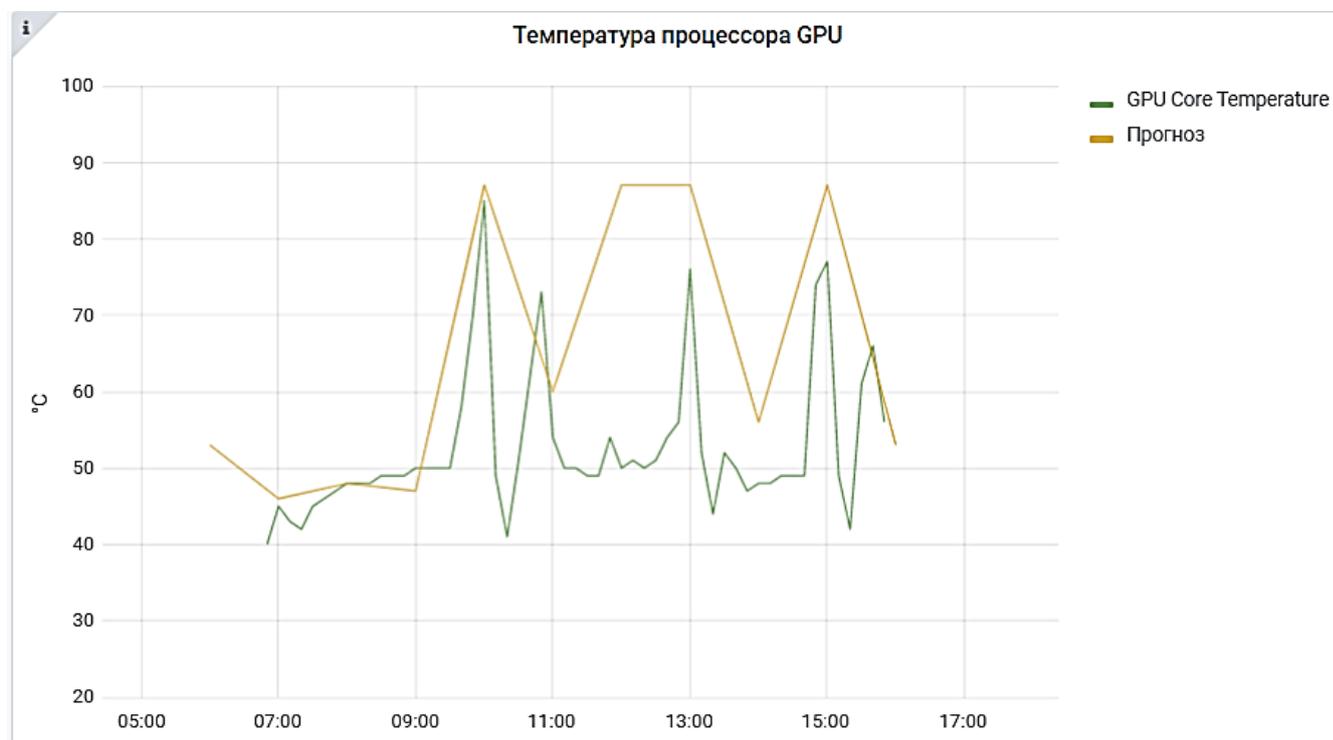


Рисунок 5. Пример прогнозирования температуры графического процессора – прогнозируемые и фактические значения на интервале с 7.00 до 16.00

2.3. ИДЕНТИФИКАТОРЫ В ЦЕПИ ОБРАТНОЙ СВЯЗИ КАК ЦИФРОВЫЕ ДВОЙНИКИ

В данной работе цифровые идентификационные data-driven модели, например, ассоциативные модели типа (4), предлагается использовать для формирования цифровых двойников. В качестве цифрового двойника, формирующего цифровые модели в темпе реального функционирования объекта, используется идентификатор в цепи обратной связи системы управления.

При этом следует учитывать, что в замкнутом контуре переменные в модели (1) в общем случае не являются статистически независимыми.

Кроме того, анализ мультиконсортной системы, также как и анализ мультиагентных и мультимодальных систем, показывает, что может иметь место [159] корреляционная зависимость между состояниями отдельных подсистем, что приводит к вырожденной задаче. В этом случае для идентификации динамических объектов с помощью ассоциативного поиска может быть применена процедура Мура-Пенроуза [160, 161]. Применение этой техники для решения задач идентификации описано в [162]. Также при разработке алгоритма ассоциативного поиска может применяться процедура частичных наименьших квадратов [163].

2.4. УСЛОВИЯ УСТОЙЧИВОСТИ ДЛЯ КРАТНОМАСШТАБНЫХ ВЕЙВЛЕТ-РАЗЛОЖЕНИЙ ДЛЯ МУЛЬТИМОДАЛЬНЫХ СИСТЕМ

Многочисленные технологические процессы в промышленности и электроэнергетике являются нестационарными. Типичным примером является изменчивость свойств сырья в системах управления технологическими процессами в перерабатывающей промышленности из-за того, что оно поступает от разных поставщиков, и т.п. Эффективным результатом для такого типа нелинейных нестационарных процессов явилась разработка предиктивных моделей с помощью метода ассоциативного поиска и вейвлет-анализа [164]. Н.Н. Бахтадзе были сформулированы условия устойчивости нестационарных мультимодальных систем в форме ограничений на коэффициенты кратномасштабных вейвлет-разложений входных и выходных сигналов динамической системы. Формулировка условий устойчивости для мультимодальных систем предоставило возможность разработки предиктивных моделей для сложных динамических систем. В частности, в качестве таких систем могут рассматриваться ЦЭС.

В базе знаний могут храниться не только наборы входов и выходов системы на протяжении всей работы исследуемого объекта, но также соответствующие «динамические слепки». Динамический слепок представляет собой набор масштабирующих и детализирующих коэффициентов кратномасштабного вейвлет-разложения [164] входов и выходов системы для некоторой временной

глубины на каждом временном шаге. Поскольку вейвлет-преобразование позволяет отображать свойства сигналов не только в частотной области, но и отображает изменения частотных характеристик во времени, применение такого преобразования позволяет исследовать объекты с изменяющимися внутренними характеристиками. Для каждого t -того входного или выходного сигнала системы берется «динамический след» системы как совокупность k предыдущих по времени входов и выходов и производится вейвлет-преобразование над этим временным рядом. Полученные коэффициенты вейвлет-преобразования также пополняют базу знаний.

Допустим, что рассматривается дискретное вейвлет-преобразование Хаара:

$$c_{x,t}, d_{x,t} = WT_{Haar}(x[t - k, t]) \quad (5)$$

$$c_{y,t}, d_{y,t} = WT_{Haar}(y[t - k, t])$$

где: $c_{x,t}, d_{x,t}$, - векторы соответственно аппроксимирующих и детализирующих коэффициентов вейвлет-преобразования входа в момент t , $c_{y,t}, d_{y,t}$ - векторы аппроксимирующих и детализирующих коэффициентов вейвлет-преобразования выхода в момент t , WT_{Haar} – дискретное вейвлет-преобразование Хаара.

Совокупность входов и выходов исследуемой системы преобразуется в элементы базы знаний системы для каждого момента времени. Для многомерных систем заполнение базы знаний производится тем же методом для каждой из компонент.

Критерий отбора элементов базы знаний для построения локальной модели определяется в зависимости от свойств исследуемой системы, и может быть как логического, так и метрического типа. Далее показаны два метрических критерия отбора для входов и выходов системы и для динамических следов.

Для отбора M входов и соответствующих выходов системы используется евклидова метрика между векторами входов базы знаний и вектором входов, для которого прогнозируется выход. Среди всех элементов базы знаний отбираются те, для которых евклидова метрика минимальна.

$$\bigcup_{i=1}^M \operatorname{argmin} = \left(\sqrt{\sum_{k=1}^l (x_k^{in} - x_k^{db})^2} \right) \quad (6)$$

где: l - размерность входа. Матрица отобранных векторов проверяется на обусловленность, и в случае необходимости множество отобранных элементов дополняется. Могут быть использованы и другие метрики.

Условия устойчивости объекта для всех коэффициентов кратномасштабного вейвлет-разложения входа и выхода объекта в фиксированный момент времени t представляются в виде [164]:

$$\begin{aligned} & |a_1 d^y(t-1) + \sum_{s=1}^S b_{1s} d^{x_s}(t-1)| < 2|a_0 d^y(t-1)|; \\ & \left| a_2 d^y(t-2) + \sum_{s=1}^S b_{2s} d^{x_s}(t-2) \right| |d^y(t)| < \left| a_1 d^y(t-1) + \sum_{s=1}^S b_{1s} d^{x_s}(t-1) \right| |d^y(t-1)|; \\ & \dots \\ & \left| \sum_{s=1}^S b_{rs} d^{x_s}(t-r) \right| |d^y(t)| < \sum_{s=1}^S b_{r-1,s} d^{x_s}(t-r+1) |d^y(t-1)|. \end{aligned} \quad (7)$$

Системный подход к исследованию функционирования крупномасштабных производственных комплексов в промышленности и энергетике находит эффективную интерпретацию в представлении сложных систем управления такими комплексами, как мультиагентные и *мультимодальные* объекты. Особенностью мультимодальных объектов является возможность прогнозирования их функционирования на основе знаний, сформированных на стадии обучения, и формализующих закономерности изменения совокупности групп факторов, характеризующих определенный режим. Примерами мультимодальных комплексов могут служить: жизненный цикл изготавливаемого продукта (технологические и логистические аспекты); функционирование энергосистемы либо ее фрагментов в нормальном, предаварийном, аварийном и пост-аварийном режимах; технологический процесс перекачивания нефти в трубопроводной системе с учетом изменения ее топологии различной конфигурацией отдельных участков трубопровода и т.п.

Мультимодальность функционирования таких систем может отражать как наличие автономных стадий (этапов) производственного процесса, так и различных режимов функционирования производственного процесса как объекта управления. В обоих случаях для описания объекта используется расширенный вектор входов

$$\hat{x} = x_{t1}^1 \dots x_{tS}^1 \dots x_{t1}^m \dots x_{tS}^m \dots x_{t1}^M \dots x_{tS}^M, \quad (8)$$

где: индекс t означает момент времени функционирования дискретной системы,

m - номер моды (стадии, этапа, режима),

S – номер компоненты вектора входов.

Мода характеризуется попаданием вектора входов в одну из непересекающихся областей в пространстве:

$$R^S = \bigcup_{m=1}^M D_{ts}^m. \quad (9)$$

Мультимодальность может отражать как многоэтапность работы системы (например, перевозки сырья, материалов и комплектующих для производства разными видами транспорта, так и функционирование системы в разных режимах. При этом функционирование объекта в том или ином режиме m , характеризующееся определенными входными параметрами (например, расходы, давления, температуры) предполагает, что все компоненты вектора входов равны нулю, кроме $x_{t1}^m \dots x_{tS}^m$.

Консорты производственной ЦЭС, реализующие выполнение определенных целевых задач, могут быть интерпретированы как элементы «многоэтапной» мультимодальной системы. В качестве таких консортов могут рассматриваться: системы управления ресурсами производства, системы управления складской логистикой, системы управления взаимоотношениями с поставщиками и потребителями, и другие подсистемы технологического и логистического цикла производства.

Для того, чтобы прогнозировать приближение выхода социотехнической системы к границам, опасным с точки зрения возможного нарушения устойчивости, применяется следующая схема:

1. формируется линейная модель нелинейного объекта, на вход которого поступает расширенный вектор \hat{x} , методом ассоциативного поиска;
2. исследуется объект с прогнозируемыми свойствами.

2.5. АЛГОРИТМЫ ПРОГНОЗИРОВАНИЯ ПРОИЗВОДСТВЕННЫХ СИТУАЦИЙ

Вектор состояния сложной динамической системы, состоящий из консортов всех типов, а именно консортов подсистем управления, консортов инфраструктуры и обслуживающих консортов, может быть интерпретирован как *вектор ситуации*. Применяя ассоциативные модели, основанные на знании динамики объекта в замкнутом контуре, мы автоматически анализируем текущую ситуацию и находим «похожие» ситуации «в прошлом» исследуемой системы.

Под *ситуационным управлением* ранее понималась методология управления сложными полу-структурированными техническими, организационными и социально-экономическими объектами с использованием методов искусственного интеллекта [47].

«Идентификация ситуации» для объекта управления осуществляется путем сравнения ситуации на объекте, представленной в виде набора признаков, с различными эталонными ситуациями (набором эталонных признаков) с использованием в большинстве случаев метода «ближайшего соседа».

Алгоритмы управления в соответствующих инженерных (и не только) задачах обычно используют логико-лингвистические, нечеткие, нейро-нечеткие и другие подходы на основе ИИ (искусственного интеллекта), а не традиционные интегро-дифференциальные и конечно-разностные уравнения. В литературе также имеется множество примеров применения этого подхода.

Это направление исследований выглядит конструктивным, все более перспективным и наиболее (а иногда и исключительно) приемлемым для большого класса объектов управления. Во многих публикациях обсуждается эта область, называемая синтезом систем с ситуационной осведомленностью [47].

В данной диссертационной работе предлагается формализовать понятие ситуации и распространить ассоциативные алгоритмы на идентификацию ситуаций. Ситуация для системы управления характеризуется ее состоянием и внешними воздействиями в текущий момент времени. Как правило, последние включают элементы управления, измеренные входные данные и шум. Состояние системы определяется информацией обо всей динамической истории.

Для системы, заданной в пространстве «вход-выход», ситуация в момент времени k будет определяться набором значений:

$$S(k) = \left\langle x(k); u(k-l); y(k-l), a_i^{k-l+1}, b_j^{k-l} \right\rangle \quad (10)$$

$$1 \leq l \leq k-1, l \in L$$

где: $x(k)$ – входы в момент k ; $y(k)$ - выходы; $u(k)$ - управляющие воздействия; L – множество моментов времени, когда входы были близки к текущему с учетом критерия (2) и были выбраны для модели. При этом входы представляют собой формализованные разнообразные факторы, характеризующие ситуацию. Это означает, что ситуация полностью определяется набором признаков, совокупность значений которых на момент времени k будем называть *цифровым двойником ситуации*. Модель (4) для цифрового двойника в момент времени k можно назвать *цифровой идентификационной моделью ситуации*.

Нормализация данных производственных ситуаций

Может случиться так, что некоторые признаки, описывающие ситуацию, носят категориальный характер для всех видов консортов. Примером может служить цветовая характеристика (категории: «красный», «желтый», «синий», ...). В некоторых случаях признаки можно описывать нечеткими переменными, но

возможно и кодирование категориальных признаков другими способами, начиная с примитивной нумерации в пределах допустимого набора значений, а затем используя всевозможные методы проецирования на вещественную ось.

Традиционная нормализация значений данных может быть эффективным способом кодирования. Наиболее известные методы: стандартное масштабирование (нормализация Z-оценки) и масштабирование MinMax:

$$\tilde{x}_{ik} = \frac{x_{ik} - x_{mini}}{x_{maxi} - x_{mini}}, \quad (11)$$

$$y_{jk} = y_{minj} + \tilde{y}_{jk}(y_{maxj} - y_{minj}). \quad (12)$$

Среди нелинейных алгоритмов наиболее популярны алгоритмы с сигмовидной логистической функцией или гиперболическим тангенсом. Переход от традиционных единиц измерения к нормализованным и наоборот при нормализации и де-нормализации в пределах $[0,1]$ осуществляется по формулам:

$$x_{ci} = x_{mini} + x_{maxi} = \frac{1}{e^{-a(x_{ik} - x_{ci})} + 1}; \quad (13)$$

$$y_{jk} = \frac{1}{a} \ln \left[\frac{1}{\tilde{y}_{jk}} - 1 \right],$$

где: x_{ci} , y_{cj} – центры нормированных интервалов изменения входной и выходной переменных:

$$x_{ci} = (x_{mini} + x_{maxi})/2; \quad y_{cj} = (y_{minj} + y_{maxj})/2. \quad (14)$$

Применяя ассоциативные алгоритмы, описанные выше, к нормализованным значениям признаков (включая значения, присвоенные категориальным признакам), мы получаем цифровые прогнозные модели для ситуаций, описываемых как числовыми, так и категориальными признаками для всех типов консортов в системе управления.

2.6. ВЫВОДЫ ПО ГЛАВЕ 2

Во **второй** главе рассматриваются алгоритмы идентификации на основе ассоциативного поиска, формулируются условия устойчивости в терминах кратномасштабных вейвлет разложений для подсистем ЦЭС, решающих разнообразные целевые задачи; их совокупность интерпретируется как мультимодальная система управления. Также представлены ассоциативные алгоритмы прогнозирования производственных ситуаций.

В **разделе 2.1** приведен анализ условий устойчивого функционирования ЦЭС как мультиконсортной системы. Отмечена целесообразность использования цифровых предиктивных моделей.

В **разделе 2.2** предложены методы анализа и прогнозирования ЦЭС на основе цифровых алгоритмов идентификации. Отмечено, что для идентификационного моделирования нелинейных процессов, к которым относится значительное количество производственных процессов, высокую эффективность продемонстрировали методы идентификации, называемые ассоциативными, основанные на принципе построения базы знаний из имеющихся данных, полученных в процессе работы динамической системы, и построении новой линейной модели для каждого искомого входа на основе интеллектуального анализа архивных и текущих данных динамического процесса.

В **разделе 2.3** обосновывается целесообразность формирования цифровых двойников консортов – решателей целевых задач как идентификаторов в цепи обратной связи систем управления.

В **разделе 2.4** исследуется условие устойчивости систем управления как группы консортов ЦЭС в терминах кратномасштабных вейвлет-разложений для нестационарных мультимодальных систем. «Динамический слепок», характеризующий состояние системы, представляет собой кратномасштабное вейвлет-разложение входов и выходов системы для некоторой временной глубины на каждом временном шаге. Поскольку вейвлет-преобразование позволяет

отображать свойства сигналов не только в частотной области, но и отображает изменения частотных характеристик во времени, применение такого преобразования позволяет исследовать объекты с изменяющимися внутренними характеристиками. Для каждого t -того входного или выходного сигнала системы формируется «динамический след» системы как совокупность k предыдущих по времени входов и выходов и производится вейвлет-преобразование.

В разделе 2.4 формируется модель производственной ситуации. Вектор состояния сложной системы, состоящий из консортов всех типов, а именно целевых-консортов, консортов инфраструктуры и обслуживающих консортов, может быть интерпретирован как вектор ситуации. Применяя ассоциативные модели, основанные на знании динамики объекта в замкнутом контуре, мы автоматически анализируем текущую ситуацию и находим «похожие» ситуации «в прошлом» исследуемой системы. В данном разделе формализуется цифровая идентификационная модель ситуации.

ГЛАВА 3. РАЗРАБОТКА МОДЕЛИ АРХИТЕКТУРЫ ЦЭС

Стабильное функционирование системы управления ЦЭС промышленного предприятия определяется не только устойчивостью всех его подсистем управления, но также стабильностью работы инфраструктуры и обеспечивающих консортов. В данной главе представлена архитектура ЦЭС, интерпретируемой как мультиконсортная система и определены условия стабильности ее работы.

3.1. КОНЦЕПТУАЛЬНАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ЦЭС ПРЕДПРИЯТИЯ

В условиях постоянно увеличивающегося количества данных и их разрозненной природы (дискретность, форматы, протоколы, схема данных, тип загрузки и прочее) построение интегрированной системы управления ЦЭС производства является сложной задачей, которая включает как интеграцию гетерогенных, асинхронно поступающих данных с разных уровней промышленного предприятия, построение современной отказоустойчивой горизонтально-масштабируемой системы хранения и управления этим данными, так и построение интеллектуальных сервисов прогнозирования динамики изменения состояния всех ресурсов, что и является ядром интегрированной системы управления. Для эффективного и гибкого управления производственным процессом необходимо интегрировать множество данных от разных источников, таких как датчики производственного оборудования, АСУ ТП, системы производственного управления класса MES, планирования производственных ресурсов MRP, а также учетные данные предприятия с уровня ERP, общие корпоративные справочники предприятия, аналитические отчеты BI-систем, внешние открытые источники данных, данные конкурентов и партнеров из внешних цифровых экосистем и др. Так, в данной диссертационной работе предлагается использовать интегрированную систему управления ЦЭС современного промышленного предприятия, способную решать задачи разных

уровней управления, а слияние данных разных уровней позволит достичь синергетического эффекта и выявлять новые, глубинные зависимости между процессами и объектами разных уровней. При этом консортами могут являться как сервисы, непосредственно направленные на решение задач управления, так и вспомогательные обеспечивающие консорты, цель которых обеспечить надежность, устойчивость, масштабирование и мониторинг интеллектуальных консортов управления.

Верхнеуровневая архитектура системы управления ресурсами в ЦЭС промышленного предприятия представлена на рисунке 6.

На архитектуре системы управления изображены различные уровни управления, от работоспособности которых зависит, будет ли эффективно функционировать система управления ресурсами производства.

Нижний уровень управления образует уровень физической инфраструктуры – диски для хранения данных, оперативная память, процессорные ядра, графические процессоры, сеть. Можно отметить, что рекомендуется использовать наиболее быстрые диски по скорости записи и чтения (класса Optane или NVME), большое количество процессорных современных ядер с возможностью использовать несколько логических потоков с одного физического, большое количество графических процессоров (для определенного класса задач), быструю оптоволоконную сеть, коммутаторы и шины для обеспечения необходимой пропускной способности сети (обычно 100Гб/сек).

Следующий уровень управления представляет собой уровень управления ОС, облаком и виртуализацией. Он должен обеспечить стабильную работу операционной системы, виртуальных машин и функционирования облака – частного или публичного.

Далее идет уровень управления вычислительной инфраструктурой, в котором осуществляется разворачивание основных распределенных масштабируемых систем обработки и хранения больших данных. Все сервисы можно разделить на Statefull (Базы данных с хранением состояния) и Stateless

(сервисы обработки данных без хранения состояния). К данным системам будем относить:

- Системы обмена сообщениями – данный класс систем предназначен для оперативного хранения и последующей обработки данных, полученных от любых внешних систем в потоковом режиме. Если загрузка данных осуществляется в режиме пакетной загрузки, то данный класс систем может не использоваться.

- Фреймворки обработки данных – набор систем, с помощью которых возможно распределенно запускать тяжелые вычислительные задачи. Системы поддерживают горизонтальное масштабирование и отказоустойчивость, что повышает производительность и стабильность работы соответственно.

- Фреймворки моделирования и обучения – системы, с помощью которых возможно решать некоторые вычислительно-тяжелые задачи по обработке изображений, видео и другие в распределенном режиме.

- Системы оркестрации – синхронизация сервисов обработки данных, хранения и моделирования между собой.

- Системы хранения данных:

- In-Memory хранение – хранение данных в памяти позволяет читать данные потребителям без обращения к дискам, что значительно увеличивает скорость вычитки данных. Для задач, где быстродействие системы является ключевым фактором, данный класс систем позволит достичь существенной экономии времени на чтении данных.

- Реляционное хранение – классические реляционные хранилища данных. Данные хранятся на диске, однако использование памяти также предполагается для кеширования запросов клиентов и хранения наиболее часто используемых данных.

- Объектное хранение – хранение неструктурированных данных, когда это необходимо. К таким данным могут относиться фотографии, видео, текст. Отличие таких систем в том, что не нужна схема данных при записи.

Данные системы представляют собой репозитории для хранения любых типов данных.

- «Быстрые» хранилища витрин – класс систем, которые ориентированы на быстрый доступ к данным со стороны клиентов, при этом данные преимущественно хранятся на диске, а не в памяти. Для быстрого доступа к данным используется эффективная организация хранения данных на программном уровне.

- Консорт-сервисный уровень – к данному уровню относятся все сервисы интеграции данных из всех источников, сервисы обработки данных в хранилище, сервисы моделирования прогнозирующих моделей управления – ядра системы управления, сервисы подачи управляющего воздействия на объект управления, сервисы визуализации прогнозов для ЛПР, сервисы оркестрации всех консортов. Все консорт-сервисы работают поверх вычислительной инфраструктуры, и их стабильность работы также зависит от стабильности работы инфраструктуры. При этом стабильность работы инфраструктуры не гарантирует стабильность работы консортных сервисов.

- Уровень принятия решений – уровень, в котором на основе данных прогнозов объектов управления производится принятие решений человеком.

Управление сервисами и вычислительной инфраструктурой является важным звеном во всей архитектуре системы управления. Эффективное управление консорт-сервисами и вычислительной инфраструктурой достигается за счет технологии контейнеризации, когда каждый консорт-сервис или вычислительный сервис хранения или обработки работает из контейнера, что позволяет достигать лучшей управляемости, отказоустойчивости и более легкой поддержки. Управление консортами и вычислительной инфраструктурой включает мониторинг работоспособности, обеспечение безопасности, логирование событий, управление отказоустойчивостью – за счет обеспечения высокой доступности сервисов и автоматического поднятия аналогичных сервисов на других машинах, оркестрации контейнеров между собой, их масштабирование при увеличении

объемов данных, а также вопросы автоматического переноса и развертывания решений между средами.

3.2. ФУНКЦИОНАЛЬНАЯ АРХИТЕКТУРА КОНСОРТ-СЕРВИСНОГО СЛОЯ

На схеме функциональной архитектуры (рисунок 7) консорт-сервисного уровня изображен набор консорт сервисов, которые выполняют основные операции с данными. Консорты работают поверх слоя инфраструктуры и выполняют следующие задачи:

- консорт интеграции и обмена – загрузка данных из различных источников. Источниками могут быть как внутренние транзакционные системы класса ERP, MES и др., внутренние системы производственных процессов и оборудования, передающие данные в потоковом режиме, а также внешние системы. Задача данного консорта – это интеграция необходимых данных в единое хранилище, первичная фильтрация, валидация схемы данных на соответствие заявленным типам.

- Консорт обработки данных – набор консорт-сервисов для обработки данных в хранилище, нормализации, применения различной бизнес логики и т. д. Также для обеспечения собственной системы уникальных первичных ключей необходимо обеспечить переключевание данных, приходящих с источников.

- Консорт моделирования – консорт-сервисы для моделирования интеллектуальных прогнозирующих моделей управления объектом. Данные

Концептуальная Архитектура Мультиконсортной Системы Управления Цифровой Экосистемой Производственного предприятия



Рисунок 7. Концептуальная архитектура построения системы управления ресурсами производства в ЦЭС предприятия

консорты также используются для повышения точности прогнозирования и быстродействия.

- Консорт управляющего воздействия – на основе продуктивных данных в реальном времени непосредственно управление производственным объектом.
- Консорт визуализации моделей прогнозирования – набор консорт-сервисов для визуализации прогнозов, их хранения и управления уведомлениями.
- Консорт оркестрации – синхронизация консофтов между собой для обеспечения синхронности выполнения преобразований над данными – загрузок, валидации схем, фильтраций, нормализаций, переключевания, моделирования, управления и визуализации. Запуск каждого консорт-сервиса должен происходить лишь в определенный период времени, последовательный или параллельный с другими консортами.

Предполагается, что внешние экосистемы будут потребителями данных для системы управления. К таким внешним экосистемам можно отнести экосистемы в области цепочек поставок – обмен прогнозами с поставщиками производства или клиентами, экосистемы государства – прогнозы по выполнению планов производства жизненно важных товаров, их цен и количества, экосистемы партнеров – обмен прогнозами с партнерами для достижения синергетического эффекта. Предполагается также, что количество таких внешних экосистем будет увеличиваться вместе с увеличением количества работающих прогнозных моделей. Осуществление взаимодействия с внешними системами предполагается через API.

Функциональная архитектура консорт-сервисного уровня представлена ниже (рисунок 7):



Рисунок 7. Функциональная архитектура консорт-сервисного уровня системы управления ресурсами производства в ЦЭС предприятия

3.3. МЕТОДЫ АНАЛИЗА СТАБИЛЬНОГО ФУНКЦИОНИРОВАНИЯ КОНСОРТ-СЕРВИСНОГО СЛОЯ И ИНФРАСТРУКТУРЫ ЦЭС

Консорт-сервисы выполняют конкретные действия – от транспортировки и обработки данных, фильтрации, обогащения и нормализации данных, синхронизации и расписания, до реализации алгоритмов моделирования и управления. Обеспечение решения подсистемой определенной целевой задачи достигается соответствующим управлением этой подсистемой.

Рассмотрим вопросы реализации стабильного функционирования ЦЭС на примере подсистемы управления ресурсами производства, которая сама представляет собой мультиконсортную систему. Следует отметить, что любая мультиконсортная система, вне зависимости от ее функционального назначения, может стабильно работать только при условиях эффективной организации консорт-сервисного слоя и слоя вычислительной инфраструктуры предлагаемой архитектуры системы управления. Под эффективной организацией понимается обеспечение необходимого уровня резервирования на «железном» уровне (диски, сеть), а также на уровне управления консорт-сервисами и инфраструктурой (использование реплик данных и технологий высокой доступности на программном уровне для баз данных, использование технологий оркестрации контейнеров для сервисов интеграций, технологий резервирования и обеспечения высокой доступности для фреймворков оркестрации данных).

3.3.1. Стабильность функционирования консорт-сервисов

Стабильность всей системы управления будет состоять из стабильной работы всех уровней архитектуры. Особый интерес представляют уровни вычислительной инфраструктуры и обслуживающих консорт-сервисов, а также системы управления вычислительной инфраструктурой. С помощью них можно повысить устойчивость системы управления за счет организации программных отказоустойчивых компонентов как вычислительной инфраструктуры, так и

консо́ртов-сервисов, сделать консорт-сервисы с автоматическим «бесшовным» переключением между разными вычислительными машинами в случае необходимости.

К уровню вычислительной инфраструктуры будут относиться все программные средства, работающие поверх операционной системы, которые необходимы для реализации какого-либо консорт-сервиса. Например, для реализации консорта интеграции необходим как источник данных, который надо также учитывать при анализе, и база данных для загрузки данных, а также сами программные фреймворки, на основе которых разрабатывается консорт интеграции. Если за основу выбран фреймворк, написанный на Java, то при анализе стабильности необходимо учитывать как стабильность самого фреймворка, так и Java Virtual Machine, которая используется «под капотом» фреймворка.

Стабильность каждого консорт-сервиса и консортов инфраструктуры оценивается показателем, значения которого меняются в интервале от 0 до 1 и означают процент стабильной (без отказов, падений по памяти, остановки работы в силу падения какой-либо машины, и других сбоев работы любого компонента системы управления) работы данного компонента в течение 1 года. Рассчитываться данный показатель должен на 1 год вперед, с учетом планирования: технических средств, их характеристик надежности (представляемые вендорами технических средств), человеческих ресурсов эксплуатации системы управления и экспертной оценке надежности фреймворков и написанных сервисов. К примеру, устойчивость используемых СУБД может фиксировать специалист отдела эксплуатации инфраструктуры, а стабильность работы фреймворков обработки данных – главный архитектор системы управления, и т. д.

Таким образом, устойчивость консорт-сервисного уровня будет рассчитываться следующим образом:

$$S_{cs} = \frac{\sum_1^{is} SI_i}{is} \frac{\sum_1^{os} SO_{os}}{os} \frac{\sum_1^{fs} SF_{fs}}{fs} \frac{\sum_1^{as} SA_{as}}{as} \frac{\sum_1^{hf} SH_{hf}}{hf}, \quad (15)$$

где:

S_{CS} - стабильность совокупности консорт-сервисов ($S_{CS} \in [0; 1]$),

iS – количество всех входящих потоков данных (загружаемых сущностей),

SI_{iS} – Inbound Stability i -ого консорт-сервиса – стабильность входящего потока данных из системы источника ($SI_{iS} \in [0; 1]$),

oS – количество всех исходящих потоков данных (результатирующих сущностей после обработки),

SO_{oS} – Outbound Stability i -ого консорт-сервиса стабильность исходящего потока данных в систему приемник ($SO_{oS} \in [0; 1]$),

fS – Количество всех используемых фреймворков для консорт-сервисов,

SF_{fS} – Framework Stability i -ого консорт-сервиса – стабильность i -ого фреймворка, на основе которого реализован консорт-сервис ($WF_{fS}, SF_{fS} \in [0; 1]$),

aS – Количество всех используемых целевых консорт-сервисов,

SA_{aS} – Applied Service Stability i -ого консорт-сервиса – стабильность i -ого целевого или обслуживающего консорт-сервиса ($W_{aS}, SA_{aS} \in [0; 1]$),

hf – Количество всех сотрудников, эксплуатирующих систему управления,

SF_{hf} – Human Factor Stability hf -ного консорт-сервиса – надежность эксплуатации системы управления hf -ным сотрудником ($W_{hf}, SF_{hf} \in [0; 1]$).

Устойчивостью языка программирования, на основе которого написан фреймворк, можно пренебречь, так как его стабильность уже может быть включена в стабильность работы самого фреймворка.

В формуле выше сам фреймворк является сервисом инфраструктуры, и его устойчивость будет уже учтена в инфраструктурной части. Поэтому им можно пренебречь. Также стабильность исходящего потока данных будет зависеть от тех

же сервисов инфраструктуры и сети, и их показатели будут учтены при определении инфраструктурной устойчивости. Среди входящих потоков данных необходимо учитывать все внешние входящие потоки данных для сервисов интеграций и слияния данных. Для всех внутренних сервисов, которые являются потребителями данных других сервисов, стабильность входящих потоков учтена в рамках учета стабильности работы самого консорт-сервиса, который является источником данных. От формулы (15) остается:

$$S_{CS} = \frac{\sum_1^{is} SI_{is}}{is} \frac{\sum_1^{as} SA_{as}}{as} \frac{\sum_1^{hf} SH_{hf}}{hf}, \quad (16)$$

где:

S_{CS} - Стабильность всех консорт-сервисов ($S_{CS} \in [0; 1]$),

is – Количество всех входящих потоков данных от внешних систем-источников (загружаемых сущностей),

SI_{is} – Inbound Stability i -того консорт-сервиса – стабильность входящего потока данных из системы источника ($W_{is}, SI_{is} \in [0; 1]$),

as – Количество всех используемых целевых или обслуживающих консорт-сервисов,

SA_{as} – Applied Service Stability i -того консорт-сервиса – стабильность i -того целевого или обслуживающего консорт-сервиса ($W_{as}, SA_{as} \in [0; 1]$),

hf – Количество всех сотрудников, эксплуатирующих систему управления,

SH_{hf} – Human Factor Stability hf -ного консорт-сервиса – надежность эксплуатации системы управления hf -ным сотрудником ($W_{hf}, SF_{hf} \in [0; 1]$).

3.3.2. Стабильность функционирования консортов инфраструктуры

Стабильность вычислительной инфраструктуры ЦЭС представляет собой стабильность работы всех *компонентов инфраструктуры* и их *взаимосвязи*. Для

достижения устойчивости системы управления она должна быть обязательно распределенной, так как в противном случае выход из строя любой машины приведет к выходу из строя всей системы управления.

В общем виде к таким компонентам инфраструктуры будем относить как различные программные модули, так и аппаратные:

$$S_i = S_s S_h \quad (S_i \in [0; 1]), \quad (17)$$

где: S_i – общая инфраструктурная устойчивость,

S_s – устойчивость программной инфраструктуры ($S_s \in [0; 1]$),

S_h – устойчивость аппаратной инфраструктуры ($S_h \in [0; 1]$).

К *программным модулям* будем относить: системы управления базами данных (СУБД) и Брокеры сообщений, фреймворки распределенной обработки данных и обучения моделей прогнозирования, фреймворки оркестрации, а также сервисы визуализации, операционные системы (хостовую и гостевую), ПО управления виртуальными машинами и облаком, ПО управления контейнерами.

К *аппаратным модулям* будем относить: сеть, диски, ядра, оперативную память, графические процессоры, коммутаторы и шины.

3.3.2.1. Инфраструктурно-аппаратная стабильность

В общем виде показатель инфраструктурно-аппаратной стабильности будет иметь вид:

$$S_a = \frac{\sum_1^u S_u}{u} \frac{\sum_1^o S_o}{o} \frac{\sum_1^d S_d}{d} \frac{\sum_1^t S_t}{t} \frac{\sum_1^y S_y}{y} \frac{\sum_1^h S_h}{h} \frac{\sum_1^f S_f}{f}, \quad (18)$$

где:

S_a – общая инфраструктурно-аппаратная стабильность ЦЭС ($S_a \in [0; 1]$),

u – количество ядер в системе управления ($u \in N$),

S_u – стабильность функционирования u -ого ядра в системе управления ($S_u \in [0; 1]$),

o – Количество плат оперативной памяти в системе управления ($o \in N$),

S_o – Стабильность функционирования o -ой платы памяти в системе управления ($S_o \in [0; 1]$),

d – Количество дисков в системе управления ($d \in N$),

S_d – Стабильность функционирования d -ого диска в системе управления ($S_d \in [0; 1]$),

t – Количество каналов связи в системе управления ($t \in N$),

S_t – Стабильность функционирования t -ого канала связи в системе управления ($S_t \in [0; 1]$),

y – Количество графических процессоров в системе управления ($y \in N$),

S_y – Стабильность функционирования y -ого графического процессора в системе управления ($S_y \in [0; 1]$),

h – Количество коммутаторов или шин в системе управления ($h \in N$),

S_h – Стабильность функционирования h -ого коммутатора или шины в системе управления ($S_h \in [0; 1]$),

s – Количество людей, эксплуатирующих аппаратную часть системы управления ($s \in N$),

S_s – Надежность эксплуатации аппаратной части системы управления s -ым человеком ($S_s \in [0; 1]$).

Стабильность функционирования каждого элемента может быть получена экспериментально либо экспертно. Для определения стабильности работы сети между двумя интерфейсами можно подсчитать общее время корректной работы

сети за год и суммарное время простоев или сбоев сети за тот же период. На основе данных метрик - получить значение показателя надежности работы 1 сетевого интерфейса.

Для определения степени стабильности функционирования физических машин (также и CPU, RAM, GPU) предлагается взять за основу тип используемых дисков, срок службы, статистику по отказам оборудования такого типа и высчитать на основе полученных данных вероятность отказа машины в течение года (возможно также использование экспертных оценок).

Для определения степени стабильности работы виртуальных машин предлагается посчитать статистику отказов и сбоев виртуальных машин, возможный срок службы и другие экспертные оценки, после чего подсчитать вероятность отказа машины в течение года.

Для операционной системы предлагается определить вероятность отказов и критичных сбоев системы, которые могут повлиять на стабильное функционирование прикладного ПО, в составе которого будут функционировать и сервисы прогнозирования. Для подсчета вероятности критичных сбоев в течение года возможно также привлечение экспертов.

В целом, виртуальные машины и операционные системы можно считать достаточно устойчивыми с точки зрения отказов и сбоев. Эти параметры достаточно сложно поддаются прогнозированию, и в общем описании устойчивости системы управления до определенного значения ими можно пренебречь.

Во многом вероятность отказов оборудования (физических машин) также носит случайный характер и сложно поддается прогнозированию. Для возможности сведения к минимуму влияния на стабильность работы системы управления ЦЭС выхода из строя физических машин необходимо использовать резервирования на уровне дата центров, что может быть достаточно дорого. Другой способ – это использование прикладных сервисов, которые обеспечат отказо-

стабильность на программном уровне приложений, что и является важной задачей системы управления: по сути, это сведение к минимуму влияния выхода из строя физических машин на общую стабильность системы управления ЦЭС.

Таким образом, влияние стабильности работы сети остается одним основным параметром из формулы выше, при условии обеспечения отказоустойчивости сервисов системы управления на прикладном уровне приложений. В этом и состоит важная задача проектирования такой системы управления ЦЭС, при которой влияние таких случайных параметров, как выход из строя физических машин, ОС или виртуальных машин будет сведено к минимуму за счет выделения отдельной системы управления отказостабильностью, реализуемой на инфраструктурно-программном уровне.

3.3.2.2. Инфраструктурно-программная стабильность

В общем виде инфраструктурно-программная стабильность будет иметь вид:

$$S_s = \frac{\sum_1^n S_n}{n} \frac{\sum_1^m S_m}{m} \frac{\sum_1^k S_k}{k} \frac{\sum_1^l S_l}{l} \frac{\sum_1^g S_g}{g} \frac{\sum_1^c S_c}{c} \frac{\sum_1^r S_r}{r} \frac{\sum_1^b S_b}{b}, \quad (19)$$

где:

S_s – Общая инфраструктурно-программная стабильность ЦЭС ($S_s \in [0; 1]$),

n – Количество баз данных или брокеров сообщений в системе управления ($n \in N$),

S_n – Стабильность функционирования n -ной базы данных ($S_n \in [0; 1]$),

m – Количество распределенных сервисов обработки данных и обучения моделей прогнозирования ($m \in N$),

S_m – Стабильность функционирования m -ого распределенного сервиса обработки данных ($S_m \in [0; 1]$),

k – Количество сервисов визуализаций ($k \in N$),

S_k – Стабильность функционирования k -ого сервиса визуализации данных ($S_k \in [0; 1]$),

l – Количество хостовых и гостевых операционных систем ($l \in N$),

S_l – Стабильность функционирования l -ной гостевой и хостовой ОС ($S_l \in [0; 1]$).

g – Количество систем управления виртуализацией и облаком ($g \in N$),

S_g – Стабильность функционирования g -ой системы управления виртуализацией и облаком ($S_g \in [0; 1]$),

S_c – Стабильность функционирования c -ой системы управления контейнеризацией ($S_c \in [0; 1]$),

r – Количество фреймворков оркестрации ($r \in N$),

S_r – Стабильность функционирования r -ого фреймворка оркестрации ($S_r \in [0; 1]$),

b – Количество людей, эксплуатирующих программную часть системы управления ($s \in N$),

S_b – Надежность эксплуатации программной части системы управления s -ым человеком ($S_b \in [0; 1]$).

Все базовые программные компоненты представляют собой основные программные комплексы более низкого уровня, на основе которых строятся целевые консорты управления ЦЭС и вспомогательные консорт-сервисы для обеспечения нормализации данных, фильтрации и обогащения. Под более низким уровнем понимается уровень абстракции программных компонент, который используется как основной для реализации необходимого сервиса. Например, для реализации интеллектуального сервиса на Python базовым компонентом будет являться Python, а при реализации нормализации данных на базе фреймворка обработки данных Apache Spark – Apache Spark и т. д. Для подсчета устойчивости базового программного компонента (фреймворка) за год может быть использован

экспертный подход. Также возможен анализ количества зарегистрированных багов по данной компоненте на официальной странице сообщества и анализ влияния данных багов на разработанный сервис, вероятность возникновения проблем с кодом во время промышленной эксплуатации решения.

Для подсчета стабильности вспомогательного сервиса нормализации, фильтрации и обогащения, который функционирует на базе базового программного компонента (фреймворка), возможно определять: вероятность сбоев при загрузке данных, некорректного формата данных, сбоев при пиковых нагрузках и возможных потенциальных проблемах с прикладным кодом нормализации, фильтрации или агрегации данных. К таким проблемам могут относиться: возможные проблемы памяти, неэффективное исполнение кода, локальные проблемы с функциями, использование старых библиотек, и прочее.

Стабильность самого целевого консорта, решающего задачи непосредственно управления, можно определить, принимая во внимание объемы данных (что возможно обработать сервисом), увеличение размерности данных, необходимость переобучения в режиме реального времени и риски проблем с памятью при переобучениях модели, эффективность организации кода, использование актуальных библиотек и прочее.

Для обеспечения стабильности при использовании различных библиотек и пакетов необходимо принимать во внимание, насколько актуальные библиотеки и пакеты используются, их стабильность работы, количество багов на официальной странице сообщества, общее время существования данных библиотек, количество разработчиков в этом сообществе.

Уровень стабильности всей системы управления этим классом консортов можно вычислить, используя анализ уровня резервирования основных мастер машин и рабочих машин, количества мастер машин и рабочих машин, используемые алгоритмы резервирования ключевых компонентов, количества

багов на официальной странице сообщества, общее время существования данного решения, количество разработчиков данного сообщества.

3.4. ВЫВОДЫ ПО ГЛАВЕ 3

В третьей главе диссертационной работы предложены концептуальная архитектура системы управления ресурсами производства в ЦЭС предприятия, функциональная архитектура консорт-сервисного слоя, описаны основные аспекты и функции каждого слоя архитектуры, предложен подход к определению устойчивости системы управления, который складывается из устойчивости целевых консортов – решателей целевых задач управления, а также стабильности работы обслуживающих консортов и консортов инфраструктуры.

В разделе 3.1 описывается концептуальная архитектура системы управления. Показано, что вычислительная инфраструктура и управление цифровыми сервисами консортами являются важнейшей частью обеспечения стабильности и устойчивости всей системы управления. От их устойчивости, в том числе, зависит, насколько устойчиво будет функционировать вся система управления.

На схеме концептуальной архитектуры отражены основные инфраструктурные слои всей системы управления – от слоя управления аппаратным обеспечением и сетью – до слоя принятия решения человеком. Отражены основные принципы и свойства каждого слоя системы управления для ее устойчивого функционирования с точки зрения инфраструктурной стабильности.

В разделе 3.2 представлена функциональная архитектура консорт-сервисного слоя системы управления. В нем осуществляются основные операции по преобразованиям данных – фильтрации, обогащения, нормализации, слияние данных из всех внешних систем, а также управление производственным объектом.

В разделе 3.3 формулируется подход к определению стабильного функционирования системы управления, в которой учитывается как консорт-

сервисная, так и инфраструктурная стабильность. Инфраструктурная стабильность (надежность) будет состоять из надежности аппаратной и программной. В аппаратную часть будут входить все компоненты аппаратного обеспечения системы управления – диски, сеть, процессорные и графические ядра, оперативная память, в программную – все программные фреймворки обработки данных и моделирования, СУБД, системы обмена сообщениями, системы оркестрации и системы визуализации данных.

ГЛАВА 4. РАЗРАБОТКА ПРОТОТИПА СИСТЕМЫ ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ ЗАРУЗКИ СЕРВЕРНОГО КОМПЛЕКСА ДЛЯ УПРАВЛЕНИЯ РЕСУРСАМИ В ЦИФРОВОЙ ПРОИЗВОДСТВЕННОЙ ЭКОСИСТЕМЕ ПАО «КАМАЗ»

4.1. ОПИСАНИЕ ТЕХНОЛОГИЧЕСКИХ КОМПОНЕНТОВ ПРОТОТИПА СИСТЕМЫ УПРАВЛЕНИЯ РЕСУРСАМИ В ЦИФРОВОЙ ПРОИЗВОДСТВЕННОЙ ЭКОСИСТЕМЕ ПАО «КАМАЗ»

Предложенный подход к синтезу цифровой производственной системы был реализован для подсистемы управления вычислительными ресурсами ПАО КАМАЗ. Был создан прототип Системы прогнозирования состояния загрузки серверного комплекса.

Разработка сервисов цифровой экосистемы выполнена на основе систем и фреймворков с открытым исходным кодом (Open-Source). Выбор технологий реализации сервисов осуществлен на основании эффективности данных технологий, масштабируемости, отказоустойчивости, стабильности работы, развивающегося сообщества разработчиков, апробации данных технологий в решении схожих задач, а также наличия основных свойств данных систем, описанных ранее в разделах 3.1, 3.2.

В данном разделе описываются компоненты, которые использовались для моделирования прототипа системы управления ресурсами. При моделировании системы управления были использованы следующие компоненты (таблица 1):

Таблица 1. Характеристики компонентов прототипа системы управления

№	Компонент	Подсистема	Назначение в системе управления	Описание	Версия
---	-----------	------------	---------------------------------------	----------	--------

1	Apache Ignite	Горячего хранения	Сервис горячего хранения данных	Распределенная In-Memory база данных для высокопроизводительных вычислений с данными [165]	2.10.0
2	Apache NIFI	Интеграции/ Слияния данных	Сервис интеграций, фильтраций, агрегаций и нормализаций данных	Распределенный, высокопроизводительный, горизонтально масштабируемый фреймворк обработки данных, обладающий визуальным интерфейсом для настройки ETL и большим количеством встроенных процессоров обработки данных [166]	1.13.2
3	Python scikit-learn	Прогнозирования	Сервис управляющего воздействия и моделирования	Модуль Python для машинного обучения, построенный на основе SciPy и распространяемый по лицензии BSD [167]	0.23
4	Grafana	Аналитики и Визуализации	Сервис визуализации прогнозов	Система визуализации метрик с открытым исходным кодом, которая имеет встроенные коннекторы ко многим базам данных [168]	7.5.5

5	PostgreSQL	Аналитики	Сервис хранения прогнозов	Система управления объектно-реляционными базами данных с открытым исходным кодом, которая активно развивалась более 30 лет и заработала прочную репутацию благодаря надежности, функциональной устойчивости и производительности [169].	13.2
6	Docker	Администрирования и эксплуатации	Сервис управления развертыванием, масштабированием и эксплуатацией системы управления	Программное обеспечение для автоматизации развёртывания и управления всеми сервисами на основе контейнеризации [170]	20.10.6

Компонентная схема в имитационном моделировании прототипа системы управления изображена на рисунке 8.

В следующих разделах диссертационной работы описываются:

- задача прогнозирования состояния серверного комплекса ресурсов;
- разворачивание сервиса горячего хранения данных (Apache Ignite) и создание таблиц хранения данных;
- разворачивание сервиса интеграций (Apache NIFI) и разработка пайплайна трансформаций данных до их загрузки в сервис горячего хранения, модель данных;

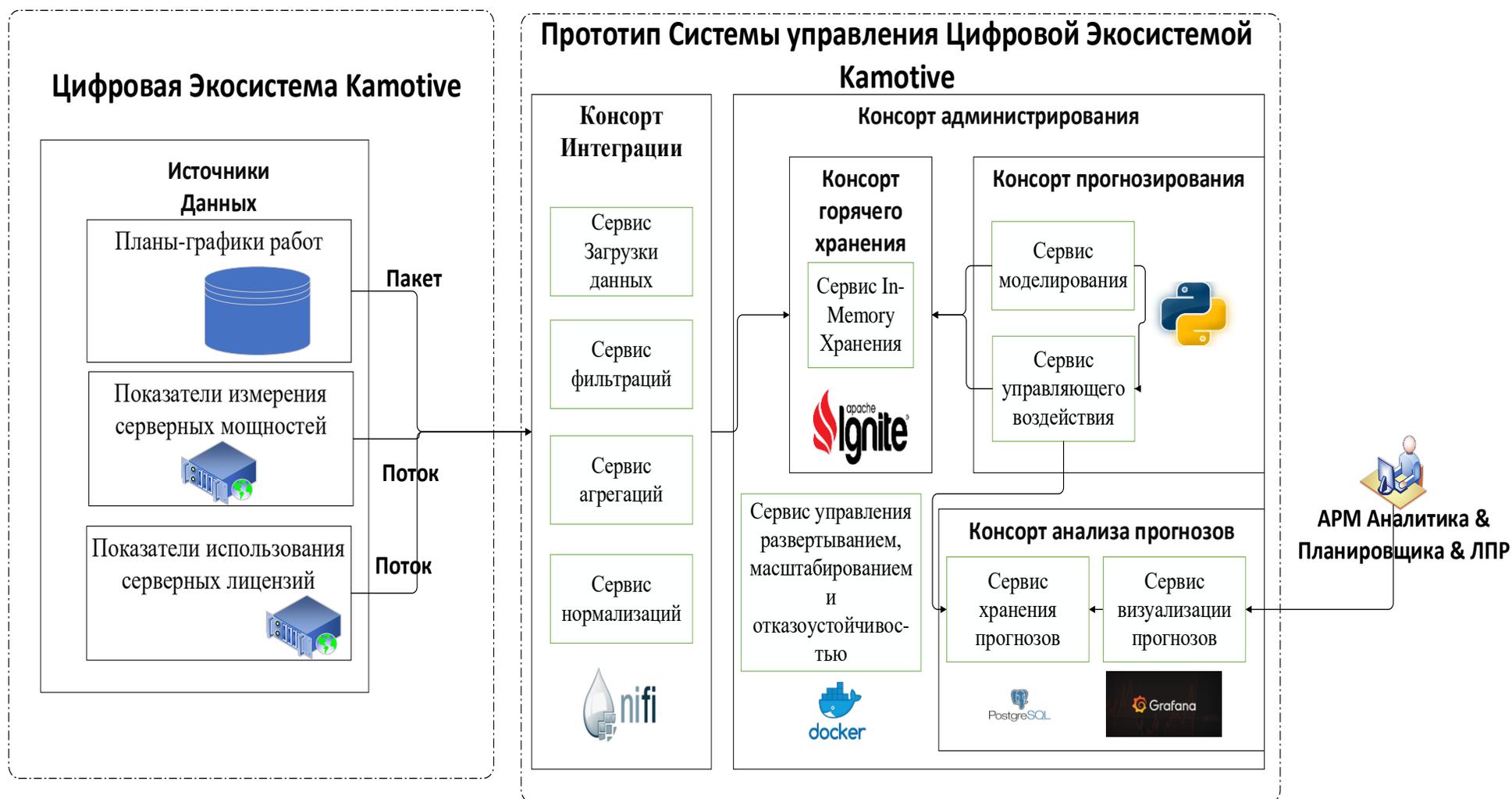


Рисунок 8. Компонентная схема модели прототипа системы управления ресурсами в производственной экосистеме ПАО «КАМАЗ»

- разворачивание сервиса визуализации прогнозов (Grafana) и хранения прогнозов (PostgreSQL);
- результаты построения прогнозирующих графиков в Grafana;
- расчет стабильного функционирования обслуживающих консортов и консортов инфраструктуры.

4.2. РЕАЛИЗАЦИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ СЕРВЕРНОГО КОМПЛЕКСА РЕСУРСОВ

Источником данных для задачи прогнозирования состояния серверного комплекса ресурсов является СУБД MySQL, развернутая из контейнера следующей командой (под суперпользователем Linux):

```
sudo docker run -p 3307:3307 --name mysql-container
-v /home/prognoz/mysql/data:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=pr0gn0zadm%!
-d mysql/mysql-server:latest
```

Далее была восстановлена база данных телеметрии серверного комплекса ресурсов объемом более 300 Гб.

Основные задачи прогнозирования состояния серверного комплекса ресурсов:

- разработка методов с использованием машинного обучения, которые на основе анализа данных или метрик определяют закономерности динамики функционирования исследуемых процессов и функциональные взаимосвязи, построение соответствующих идентификационных моделей;
- прогнозирование динамики основных показателей на основе исторических данных – для поддержки своевременного принятия решений;
- визуализация прогнозов изменения состояния комплекса ресурсов;

– хранение прогнозов изменения состояния комплекса ресурсов.

На рисунке 9 изображены основные метрики с датчиков комплекса серверных ресурсов и их описание:

№	Наименование датчика	Тип	Описание	Ед. изм.	Тип инфо	Таблица	Триггер
1	RAM Memory Used	Generic Memc	Используемая оп	Gb	Числовой	history	
2	RAM Memory Load	Generic Memc	Загрузка оператив	%	Числовой	history	
3	CPU Core#1 Load	CPU информ	Загрузка CPU	%	Числовой	history	
4	CPU Core Temperature	CPU информ	Температура CPU	°C	Числовой	history	Превышение допустимого значения 80 °C в течении 10 минут
5	GPU Core Load	GPU информ	Загрузка процесс	%	Числовой	history	
6	GPU Memory Load	GPU информ	Загрузка памяти	%	Числовой	history	
7	GPU Memory Used	GPU информ	Используемая па	Mb/Gb	Числовой	history	
8	GPU Memory Total	GPU информ	Общий объем па	Mb/Gb	Числовой	history	
9	GPU Core Temperature	GPU информ	Температура про	°C	Числовой	history	Превышение допустимого значения 85 °C в течении 10 минут
10	Host Ping	Техническая	Доступность узла сети		Числовой	history_uint	Узел недоступен более 21 день
11	OpenHardwareMonitor Webser	Техническая	Проверка статуса службы		Числовой	history_uint	
12	Материнская плата (Модель)	Техническая	Модель материнской плат		Текст	history_text	
13	Операционная система (Арх	Техническая	Разрядность операционно		Текст	history_text	
14	Процессор (модель)	Техническая	Модель процессора		Текст	history_text	
15	Computer Name	Техническая	Производитель компьюте		Текст	history_text	
16	Материнская плата (Произв	Техническая	Производитель материнск		Текст	history_text	
17	Computer Identifying Number	Техническая	Серийный номер компью		Текст	history_text	
18	Операционная система (Наи	Техническая	Вид операционной систе		Текст	history_text	
19	Оперативная память (Объем	Техническая	Объем оперативн	Gb	Числовой	history	Изменение объема оперативной памяти
20	Монитор (Модель, серийный	Техническая	Модель и серийный номе		Текст	history_text	
21	Жесткий диск (модель)	Техническая	Модель жесткого диска		Текст	history_text	Изменение списка жестких дисков
22	Полное имя пользователя	Техническая	ФИО пользователя		Текст	history_text	
23	Видеокарта (модель)	Техническая	Модель видеокарты		Текст	history_text	
24	Доменное имя пользователя	Техническая	Доменное имя пользовате		Текст	history_text	
25	Teamcenter	Windows Proc	Объем используе	Mb/Gb	Числовой	history	
26	NX process	Windows Proc	Объем используе	Mb/Gb	Числовой	history	

Рисунок 9. Пороговые значения идентификации проблемных ситуаций

Выполнено сопоставление индексов датчиков и имен параметров, которые датчики считывают. Данные телеметрии по объектам, которые записываются в базу данных, имеют следующую структуру:

- объекты (компьютеры, сервера, устройства, оборудование) помечаются индексами «hostid», всего 2109 уникальных объектов;
- каждый объект (host) имеет набор датчиков, которым присвоен уникальный идентификатор «itemid», всего 185758 уникальных идентификаторов с датчиков;
- каждому идентификатору датчика соответствует имя датчика «name», всего 598 уникальных имен датчиков.

На рисунке 10 приведено описание датчиков:

Архив с данными телеметрии серверного комплекса ресурсов (БД Zabbix) содержит 149 таблиц.

Сопоставление приведенных данных с данными базы zabbix выполняется с помощью таблицы items (рисунок 12):

№	Наименование датчика	Тип	Описание
1	RAM Memory Used	Generic Memory информация	Используемая оперативная па
2	RAM Memory Load	Generic Memory информация	Загрузка оперативной памяти
3	CPU Core#1 Load	CPU информация	Загрузка CPU
4	CPU Core Temperature	CPU информация	Температура CPU
5	GPU Core Load	GPU информация	Загрузка процессора GPU
6	GPU Memory Load	GPU информация	Загрузка памяти GPU
7	GPU Memory Used	GPU информация	Используемая память GPU
8	GPU Memory Total	GPU информация	Общий объем памяти GPU
9	GPU Core Temperature	GPU информация	Температура процессора GPU
10	Host Ping	Техническая информация	Доступность узла сети
11	OpenHardwareMonitor Webservice	Техническая информация	Проверка статуса службы
12	Материнская плата (Модель)	Техническая информация	Модель материнской платы
13	Операционная система (Архитектура)	Техническая информация	Разрядность операционной с
14	Процессор (модель)	Техническая информация	Модель процессора
15	Computer Name	Техническая информация	Производитель компьютера
16	Материнская плата (Производитель)	Техническая информация	Производитель материнской п
17	Computer Identifying Number	Техническая информация	Серийный номер компьютера
18	Операционная система (Наименование)	Техническая информация	Вид операционной системы
19	Оперативная память (Объем)	Техническая информация	Объем оперативной памяти

Рисунок 12. Сопоставление названий и идентификаторов ресурсов

Основная таблица history содержит основные данные телеметрии (рисунок 13):

Index	log time	ID item	Value
0	2020-07-08 09:11:00	215534	7.8982
1	2020-07-08 09:12:00	215534	7.8933
2	2020-07-08 09:13:00	215534	7.8969
3	2020-07-08 09:14:00	215534	7.9357
4	2020-07-08 09:15:00	215534	7.9191
5	2020-07-08 09:16:00	215534	7.93
6	2020-07-08 09:17:00	215534	7.9062
7	2020-07-08 09:18:00	215534	7.9155
8	2020-07-08 09:19:00	215534	7.9029
9	2020-07-08 09:20:00	215534	7.9042
10	2020-07-08 09:21:00	215534	7.9091

Рисунок 13. Тестовая выборка из таблицы history

Были реализованы функции прогноза, для примера был выполнен прогноз значений 8-ми типов датчиков по 10 машинам.

Реализованы функции формирования входной выборки данных для прогноза определения целевых прогнозируемых нештатных состояний (рисунок 14).

На этапе разработки модуля прогнозирования с целью настройки и проверки работы алгоритмов выбран тип данных «history», отобраны параметры прогноза по этому типу: clock, itemid, value.

На вход алгоритма подаются данные в формате транзакций, для этого переводим таблицу к формату: [Время]; [Параметр=Значение] (рисунок 14):

Name	Type	Size	
conn	engine.base.Connection	1	Connection object of sqlalchemy.engine.base module
df	DataFrame	(985707, 2)	Column names: log_time, feature
engine	engine.base.Engine	1	Engine object of sqlalchemy.engine.base module
history_uint	DataFrame	(1000000, 4)	Column names: itemid, clock, value, ns

Index	itemid	clock	value	ns
516001	100686	1559104339	33	852844821
516558	100686	1559104266	32	902454672
517087	100686	1559104207	31	597434546
517607	100686	1559104147	31	841133753
518125	100686	1559104086	30	825720285
518616	100686	1559104027	29	793686959
519106	100686	1559103966	28	905531636
519579	100686	1559103907	27	684542344
520045	100686	1559103847	27	60370363
520492	100686	1559103787	25	920152189
520940	100686	1559103727	24	924143164

Index	log_time	feature
0	21.05.2019 04:20	100686=28
1	21.05.2019 04:21	100686=28
2	21.05.2019 04:21	100687=33
3	21.05.2019 04:22	100686=29
4	21.05.2019 04:22	100687=33
5	21.05.2019 04:23	100686=30
6	21.05.2019 04:23	100687=34
7	21.05.2019 04:24	100686=30
8	21.05.2019 04:24	100687=34
9	21.05.2019 04:25	100686=31
10	21.05.2019 04:25	100687=35

Рисунок 14. Выбор параметров для алгоритма прогнозирования

Приводим данные к формату транзакций: [Время]; [Список параметров].

4.3. РАЗВОРАЧИВАНИЕ СЕРВИСА ГОРЯЧЕГО ХРАНЕНИЯ ДАННЫХ

Для горячего хранения данных и переобучения интеллектуальных моделей был развернут Apache Ignite, в котором данные хранились в оперативной памяти. Он был развернут из докер-контейнера следующей командой:

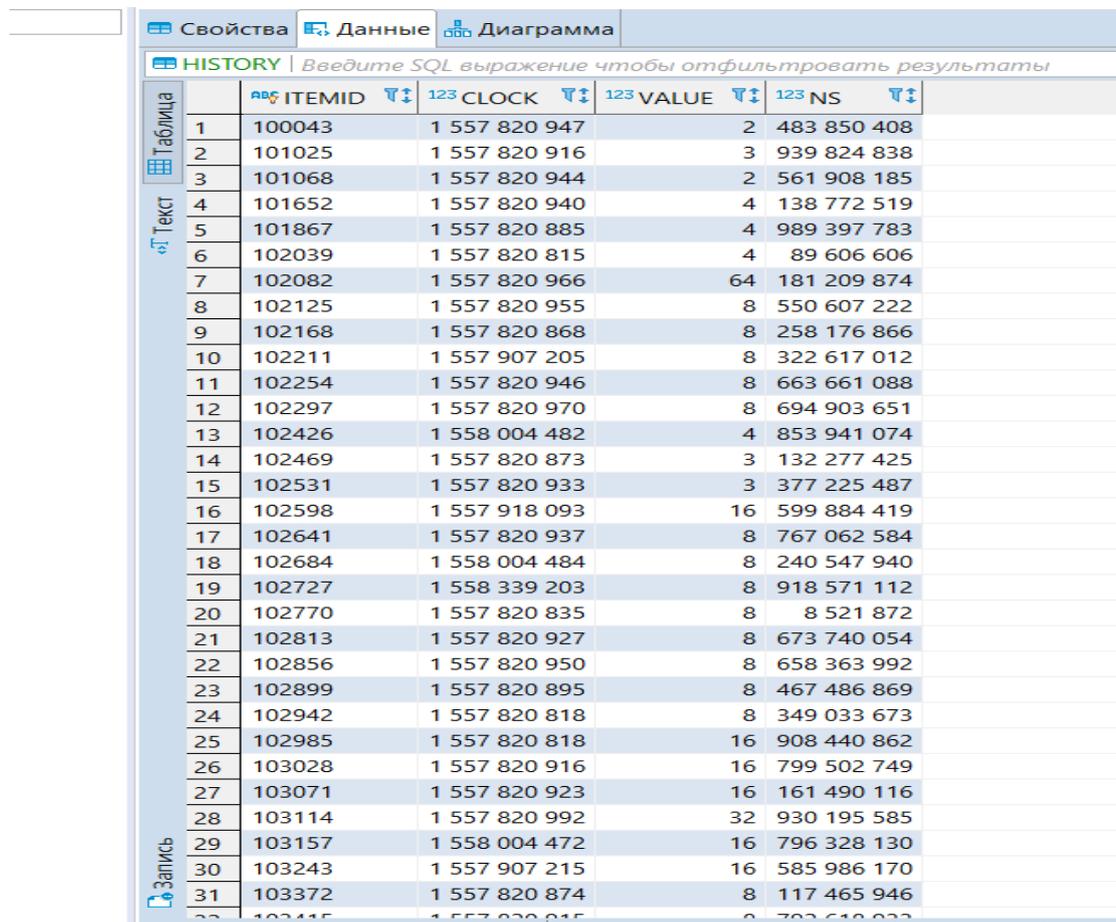
```
docker run -d --name=ignite -p 10800:10800 -v /opt/ignite_data:/storage -e
IGNITE_WORK_DIR=/storage apacheignite/ignite:2.10.0
```

В Ignite была создана таблица по результатам обработки данных:

```
CREATE TABLE PUBLIC.HISTORY (
  ITEMID VARCHAR(100),
  CLOCK BIGINT,
  VALUE DOUBLE,
  NS BIGINT,
```

);
CONSTRAINT ITEMID PRIMARY KEY (ITEMID)

Наполнение данными таблицы из подсистемы обработки данных (рисунок 15):



	ITEMID	CLOCK	VALUE	NS
1	100043	1 557 820 947	2	483 850 408
2	101025	1 557 820 916	3	939 824 838
3	101068	1 557 820 944	2	561 908 185
4	101652	1 557 820 940	4	138 772 519
5	101867	1 557 820 885	4	989 397 783
6	102039	1 557 820 815	4	89 606 606
7	102082	1 557 820 966	64	181 209 874
8	102125	1 557 820 955	8	550 607 222
9	102168	1 557 820 868	8	258 176 866
10	102211	1 557 907 205	8	322 617 012
11	102254	1 557 820 946	8	663 661 088
12	102297	1 557 820 970	8	694 903 651
13	102426	1 558 004 482	4	853 941 074
14	102469	1 557 820 873	3	132 277 425
15	102531	1 557 820 933	3	377 225 487
16	102598	1 557 918 093	16	599 884 419
17	102641	1 557 820 937	8	767 062 584
18	102684	1 558 004 484	8	240 547 940
19	102727	1 558 339 203	8	918 571 112
20	102770	1 557 820 835	8	8 521 872
21	102813	1 557 820 927	8	673 740 054
22	102856	1 557 820 950	8	658 363 992
23	102899	1 557 820 895	8	467 486 869
24	102942	1 557 820 818	8	349 033 673
25	102985	1 557 820 818	16	908 440 862
26	103028	1 557 820 916	16	799 502 749
27	103071	1 557 820 923	16	161 490 116
28	103114	1 557 820 992	32	930 195 585
29	103157	1 558 004 472	16	796 328 130
30	103243	1 557 907 215	16	585 986 170
31	103372	1 557 820 874	8	117 465 946

Рисунок 15. Наполнение данными таблицы по телеметрии состояния комплекса производственных ресурсов в Apache Ignite из Apache NIFI

Также был создан справочник для обогащения данных, с помощью которого можно идентифицировать имя хоста и имя метрики для анализа (рисунок 16):

ITEMID	TYPE	SNMP_COMMUNITY	SNMP_OID	HOSTID	NAME	KEY	DELAY	HISTORY
1	10 009	0		10 001	Number of processes	proc.num[]	1m	3d
2	10 010	0		10 001	Processor load (1 min average per core)	system.cpu.load[percpu,avg1]	1m	3d
3	10 013	0		10 001	Number of running processes	proc.num[,run]	1m	3d
4	10 014	0		10 001	Free swap space	system.swap.size[,free]	1m	3d
5	10 016	0		10 001	Number of logged in users	system.users.num	1m	3d
6	10 019	0		10 001	Checksum of \$1	vfs.file.cksum[/etc/passwd]	1h	3d
7	10 020	0		10 001	Agent ping	agent.ping	1m	3d
8	10 025	0		10 001	System uptime	system.uptime	10m	3d
9	10 026	0		10 001	Total memory	vm.memory.size[total]	1h	3d
10	10 030	0		10 001	Total swap space	system.swap.size[,total]	1h	3d
11	10 055	0		10 001	Maximum number of processes	kernel.maxproc	1h	3d
12	10 056	0		10 001	Maximum number of opened files	kernel.maxfiles	1h	3d
13	10 057	0		10 001	Host name	system.hostname	1h	3d
14	10 058	0		10 001	System information	system.uname	1h	3d
15	10 059	0		10 001	Version of zabbix_agent(d) running	agent.version	1h	3d
16	17 318	0		10 001	Host boot time	system.boottime	10m	3d
17	17 350	0		10 001	Free swap space in %	system.swap.size[,pfree]	1m	3d
18	17 352	0		10 001	Host local time	system.localtime	1m	3d
19	17 354	0		10 001	CPU \$2 time	system.cpu.util[,idle]	1m	3d
20	17 356	0		10 001	CPU \$2 time	system.cpu.util[,user]	1m	3d
21	17 358	0		10 001	CPU \$2 time	system.cpu.util[,nice]	1m	3d
22	17 360	0		10 001	CPU \$2 time	system.cpu.util[,system]	1m	3d
23	17 362	0		10 001	CPU \$2 time	system.cpu.util[,iowait]	1m	3d
24	22 181	0		10 001	Available memory	vm.memory.size[available]	1m	3d
25	22 183	5		10 047	Zabbix \$2 write cache, % free	zabbix[wcache,history,pfree]	1m	3d
26	22 185	5		10 047	Zabbix \$2 write cache, % free	zabbix[wcache,trend,pfree]	1m	3d
27	22 187	5		10 047	Values processed by Zabbix server per second	zabbix[wcache,values]	1m	3d
28	22 189	5		10 047	Zabbix configuration cache, % free	zabbix[r-cache,buffer,pfree]	1m	3d
29	22 191	5		10 047	Zabbix value cache, % free	zabbix[v-cache,buffer,pfree]	1m	3d
30	22 196	5		10 047	Zabbix value cache hits	zabbix[v-cache,cache,hits]	1m	3d
31	22 199	5		10 047	Zabbix value cache misses	zabbix[v-cache,cache,misses]	1m	3d

Рисунок 16. Наполнение данными таблицы по справочнику комплекса ресурсов в Apache Ignite из Apache NIFI

4.4. РАЗВОРАЧИВАНИЕ СЕРВИСА ИНТЕГРАЦИЙ И СЛИЯНИЯ ДАННЫХ

В подсистеме интеграции были выполнены основные преобразования данных - загрузка из источника, нормализация и вся обработка данных перед тем, как поместить данные в целевое хранилище (Apache Ignite).

Подсистема построена на основе Apache NIFI, которая также работает из докер-контейнера:

```
docker run -d --name=nifi -p 8080:8080 -v /opt/nifi_data:/opt/nifi/nifi-toolkit-current apache/nifi:1.13.2
```

Обработка данных начинается с их загрузки из СУБД MySQL, далее происходит их обработка, нормализация, конвертация в Json, затем в SQL, запись

SQL в БД Apache Ignite – такие преобразования делаются для каждой строки данных (рисунок 17):

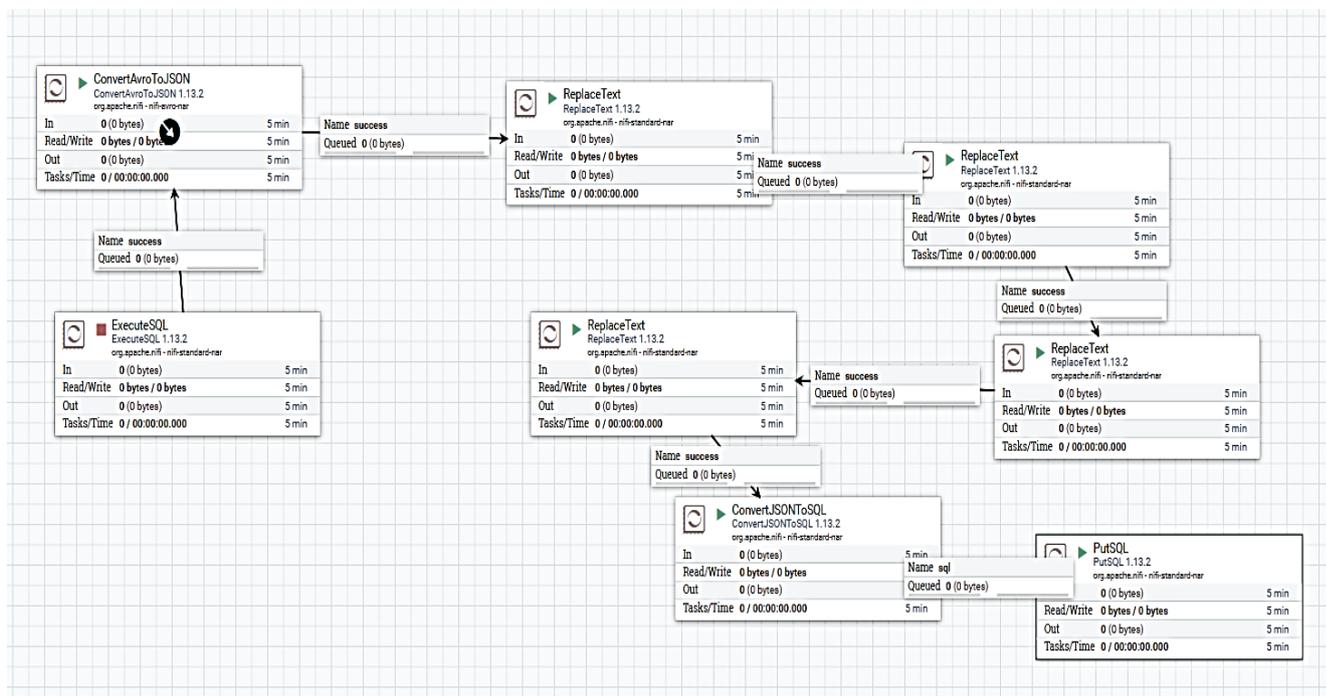


Рисунок 17. Обработка данных состояния комплекса производственных ресурсов в Apache NIFI

Сущность ITEMS обновляется достаточно редко и является справочником. Исходя из этого было принято решение грузить данные в Apache Ignite вручную.

Далее представлена разработка пайплайна в Apache NIFI для загрузки данных телеметрии комплекса производственных ресурсов (таблица HISTORY) – основной сущности хранения всех данных телеметрии для дальнейшего анализа.

Процесс преобразований данных в Apache NIFI следующий. Для извлечения данных из MySQL используем процессор в NIFI “ExecuteSQL”, настройки процессора приведены на рисунке 18:

Configure Processor

Stopped

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field +

Property	Value
Database Connection Pooling Service	MY_SQL_connect
SQL Pre-Query	No value set
SQL select query	SELECT * FROM history WHERE CLOCK > \${time_start} LI...
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	SNAPPY
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0

Рисунок 18. Настройка процессорной группы “ExecuteSQL”

Также необходимо настроить коннектор к БД MySQL в разделе Controller Services (рисунок 19):

Controller Service Details

SETTINGS **PROPERTIES** COMMENTS

Required field

Property	Value
Database Connection URL	jdbc:mysql://192.168.21.2:3306/zabbix
Database Driver Class Name	com.mysql.cj.jdbc.Driver
Database Driver Location(s)	/opt/nifi/mysql-connector-java-8.0.25.jar
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Password	No value set
Database User	prognozz
Password	Sensitive value set
Max Wait Time	500 millis
Max Total Connections	8
Validation query	No value set
Minimum Idle Connections	0
Max Idle Connections	8
Max Connection Lifetime	-1

Рисунок 19. Настройка коннектора к СУБД MySQL

Данные получены, далее их нужно преобразовать в читаемый формат, поскольку при извлечении большого числа записей необходимо сделать преобразования с помощью процессора “ConvertAvroToJSON”(со стандартными настройками).

Названия столбцов регистрозависимые, поэтому необходимо заменить название полей на заглавный регистр. Для этого используем 4 процессора “ReplaceText” (рисунок 20):

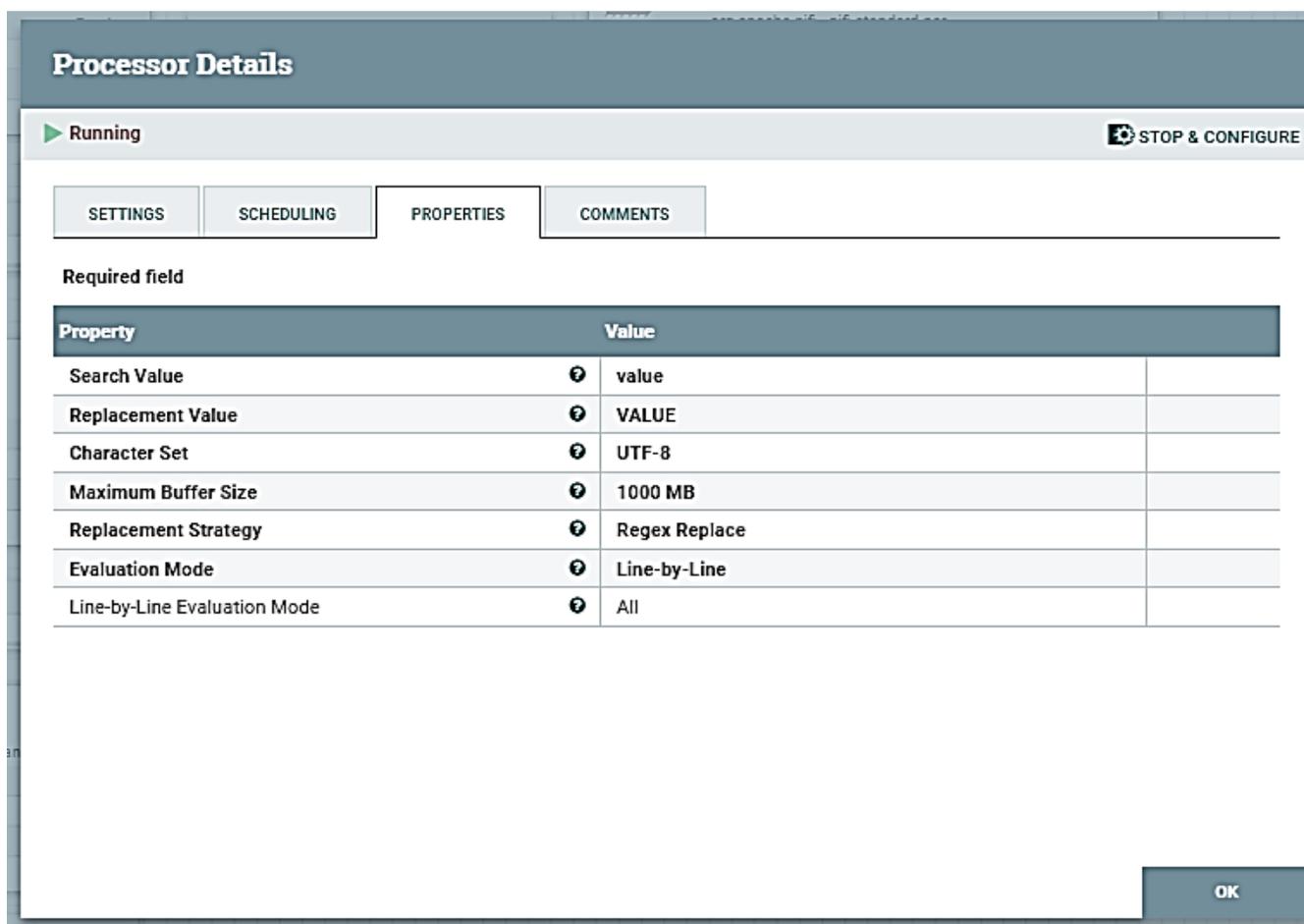


Рисунок 20. Настройка процессорной группы “ReplaceText”

Далее записываем данные в Apache Ignite. Для начала формируем SQL запрос из преобразованных данных: процессор “JSONtoSQL” (рисунок 21):

Processor Details

▶ Running ⚙ STOP & CONFIGURE

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field

Property	Value
JDBC Connection Pool	ignite_connect
Statement Type	INSERT
Table Name	HISTORY
Catalog Name	No value set
Schema Name	No value set
Translate Field Names	true
Unmatched Field Behavior	Ignore Unmatched Fields
Unmatched Column Behavior	Fail on Unmatched Columns
Update Keys	No value set
Quote Column Identifiers	false
Quote Table Identifiers	false
SQL Parameter Attribute Prefix	sql

OK

Рисунок 21. Настройка процессорной группы “JSONtoSQL”

Далее выполняем запись в Ignite, используя процессор “PutSQL” (рисунок 22):

Processor Details

▶ Running ⚙ STOP & CONFIGURE

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field

Property	Value
JDBC Connection Pool	ignite_connect
SQL Statement	No value set
Support Fragmented Transactions	true
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	100
Obtain Generated Keys	false
Rollback On Failure	false

Рисунок 22. Настройка процессорной группы “PutSQL”

Таким образом, заканчивается обработка данных, по результату которой данные помещены в Apache Ignite для долгосрочного хранения в памяти. При поступлении новых данных с источника, данные будут автоматически загружены

в Apache Ignite, так как весь пайплайн обработки данных может работать в режиме, близкому к реальному времени, постоянно опрашивая СУБД MySQL на наличие новых данных.

4.5. РАЗВОРАЧИВАНИЕ СЕРВИСА ВИЗУАЛИЗАЦИЙ И РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ

Подсистема визуализации прогнозов реализована на основе Grafana, развернутой из докер-контейнера:

```
- docker run -d --name=grafana_2 -p 3000:3000 -v /opt/grafana_data:/var/lib/grafana grafana/grafana:7.5.5
```

В Grafana хранятся результаты прогнозов по состоянию комплекса ресурсов, а также аналитика по текущим данным. Grafana настроена на источник хранения прогнозов - PostgreSQL, в который попадают метрики прогнозирования.

Созданы примеры графиков прогнозирования состояний комплекса ресурсов для следующих показателей:

- Загрузка CPU.
- Загрузка процессора GPU.
- Загрузка памяти GPU.
- Температура CPU.
- Температура жесткого диска.
- Температура процессора GPU.

На рисунке 23 изображен результат прогнозирования состояния серверного комплекса ресурсов на примере температуры центрального процессора (дневной интервал). График показывает высокую точность метода ассоциативного поиска для данной задачи:

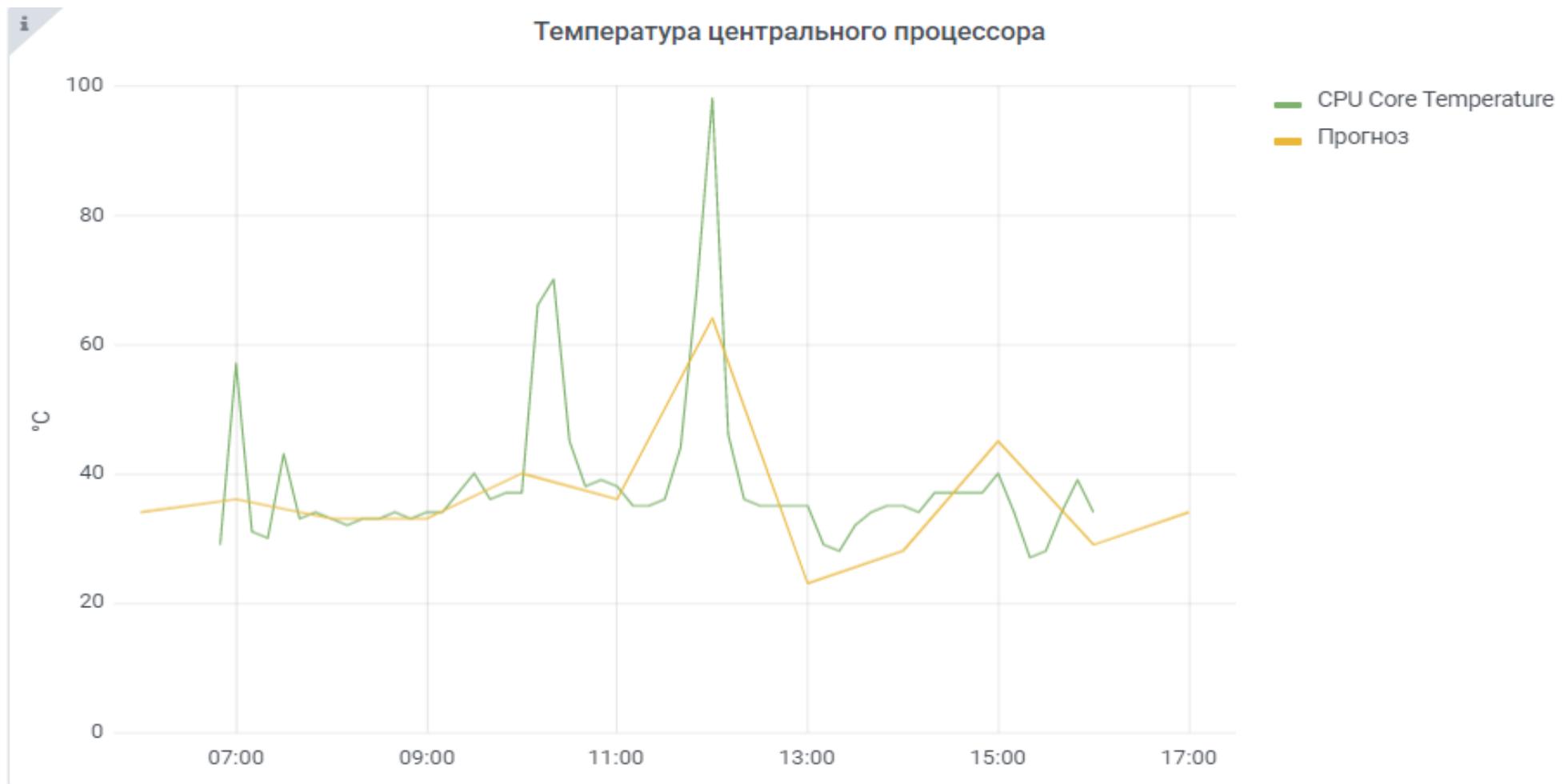


Рисунок 23. Результат прогнозирования состояния серверного комплекса ресурсов на примере температуры центрального процессора (дневной интервал)

4.6. СТАБИЛЬНОСТЬ ПОЛУЧЕННОГО ПРОТОТИПА СИСТЕМЫ УПРАВЛЕНИЯ

Для подсчета стабильности работы системы управления была проанализирована стабильность работы следующих частей:

- Инфраструктурно-аппаратный слой – для тестирования использовались новые сервера, и исходя из официальной документации вендоров для метрик CPU, RAM, GPU, диски можно указать надежность как число, большее 99.99999, поэтому можно округлить данные показатели до 1. Для устойчивости сети можно взять экспертную оценку текущих сетевых интерфейсов в 99.9999.

- Инфраструктурно-программный слой: для расчета устойчивости программных компонент необходимо экспертно оценить все программные компоненты, использованные во внедрении системы управления:

- СУБД хранения данных в памяти – несмотря на то, что прототип сделан на 1 машине, диски являются новыми и ранее мы уже сделали допущение, что можно зафиксировать надежность системы в первый год как 1. Однако устойчивость самой СУБД Apache Ignite, используемой в прототипе, можно оценить как 99.99, так как СУБД еще недостаточно совершенна, в сообществе достаточно много открытых багов, отдельные случаи работы СУБД для больших объемов могут вызывать непредвиденные баги.

- Фреймворк интеграций и слияния. В прототипе использовался Apache NIFI, который является достаточно стабильным решением. Однако, так как он работает на Java, то могут возникать проблемы с памятью при новых интеграциях и больших количествах данных. Экспертно можно оценить устойчивость данного консорта как 99.99.

- Фреймворк визуализации данных экспертно можно оценить как устойчивый и стабильный, можно принять оценку устойчивости как 99.99.

- Фреймворк распределенной обработки данных при обучении модели прогнозирования – 99.9999.
- Консорт-сервисный слой – обслуживающие консорты:
 - Стабильность функционирования входящих потоков данных: 99.99.
 - Сервис визуализации: 99.99999.
 - Сервис управляющего воздействия: 99.9999.
 - Сервис интеграций и слияния данных: 99.999.

Человеческий фактор эксплуатации инфраструктуры и сервисного слоя – 99.99.

Итого, в результате получаем стабильность функционирования всей системы на основе инфраструктурных консортов и обслуживающих как:

$$99.99 * 99.99 * 99.99 * 99.9999 * 99.9999 * 99.99 * 99.99999 * 99.9999 * 99.999 * 99.99 = 99.49.$$

С учетом допущений, что разработанная система является прототипом, данное значение стабильности функционирования ЦЭС может быть увеличено путем организации программной отказоустойчивости консортов разного уровня (обслуживающих, инфраструктурных).

4.7. ВЫВОДЫ ПО ГЛАВЕ 4

В 4-ой главе диссертационной работы разработан прототип системы прогнозирования состояния загрузки серверного комплекса для управления ресурсами в цифровой производственной экосистеме ПАО «КАМАЗ». В главе дается характеристика основных программных средств с открытым исходным кодом, использованных при внедрении, а также представляется модель программной (технологической) архитектуры прототипа системы управления, считается показатель стабильности функционирования ЦЭС на основе консортов разного типа – инфраструктурных и обслуживающих консортов.

В разделе 4.1 описываются используемые технологии больших данных с открытым исходным кодом, используемые при разработке прототипа системы управления. При реализации были использованы следующие технологии: СУБД In-Memory хранения Apache Ignite, СУБД хранения прогнозов PostgreSQL, фреймворк интеграции и слияния данных Apache NIFI, язык реализации моделей прогнозирования состояния комплекса ресурсов – python, фреймворк визуализации прогнозов – Grafana, управление контейнеризацией – docker.

В разделе 4.2 описана задача прогнозирования состояния ресурсов производства в ПАО «КАМАЗ» - прогнозирование состояния комплекса ресурсов. К прогнозируемым метрикам загрузки мощностей производственного оборудования относятся процессорные ядра, графическая память процессора, загрузка оперативной памяти процессора, температура процессора, температура графического процессора, температура жестких дисков.

В разделе 4.3 описывается разворачивание сервиса горячего In-Memory хранения данных – Apache Ignite. Формулируются основные команды разворачивания сервиса из docker-контейнера и описывается создание основных таблиц в СУБД, используемых далее в качестве источников данных.

В разделе 4.4 описывается разработка сервиса интеграций и слияний данных. Приводятся команды разворачивания сервиса из docker-контейнера, а также описывается весь пайплайн обработки данных от загрузки с источника (внешняя СУБД MySQL), трансформации, конвертации форматов данных, фильтрация, до загрузки данных в In-Memory СУБД Apache Ignite для исторического хранения данных.

В разделе 4.5 представлено разворачивание фреймворка визуализации данных на базе Grafana, а также созданные прототипы визуализации прогнозов состояния загрузки серверного комплекса ресурсов.

В разделе 4.6 представляется описание расчета стабильности работы всей ЦЭС – инфраструктурных и обслуживающих консортов. Рассчитанная

стабильность показала высокую надежность прототипа созданной системы управления. Однако стоит отметить, что возможно сильно увеличить данный показатель надежности путем применения дополнительных кластерных технологий для всех типов консортов за счет технологий высокой доступности для используемых фреймворков и дополнительного программного резервирования консорт-сервисов, которые не были использованы во время реализации прототипа.

ЗАКЛЮЧЕНИЕ

В диссертационной работе были решены основные поставленные задачи:

- Проанализированы результаты исследований по теме разработки и развития цифровых экосистем.
- Разработаны методы анализа и прогнозирования показателей производственных процессов и ситуаций на основе цифровых алгоритмов идентификации.
- Разработаны цифровые ассоциативные методы идентификации ЦЭС как мультиконсортных систем.
- Разработана концептуальная архитектура и функциональная архитектура сервисного слоя системы управления цифровой экосистемой промышленного предприятия, сформулированы основные необходимые свойства и выявлены уровни цифровых сервисов системы управления.
- Разработан подход к обеспечению стабильного функционирования ЦЭС, учитывающий как факторы внешних экосистем, так и надежность вычислительной инфраструктуры, стабильного функционирования обслуживающих консорт-сервисов и устойчивость подсистем управления.
- Исследованы и проанализированы системы класса больших данных для обработки разнородных данных в режиме реального времени в оперативной памяти (Big Data In-Memory Systems), системы обработки сообщений в потоковом режиме, технологии виртуализации и контейнеризации.
- Разработан прототип системы прогнозирования состояния загрузки серверного комплекса для управления ресурсами в цифровой производственной экосистеме ПАО «КАМАЗ».

ЛИТЕРАТУРА

1. Сулейкин А. С., Бахтадзе Н. Н. Модели архитектуры цифровых экосистем в сфере управления цепочками поставок // Информационные технологии и вычислительные системы. – 2019. – №4. – С. 21-34.
2. Suleykin A., Bakhtadze N., Pavlov B., Pyatetsky V. Digital Energy Ecosystems // IFAC PapersOnLine. – 2019. – Vol. 52, N 13. – P. 30–35. – DOI: <https://doi.org/10.1016/j.ifacol.2019.11.088>.
3. Babiolakis M. Forget Products. Build Ecosystems. How products are transforming to open interconnectable interfaces [Электронный ресурс]. – Режим доступа: <https://medium.com/@manolisbabiolakis/forget-products-build-ecosystems-792dea2cc4f2>.
4. Senyo P. K., Kecheng L., Effah J. Understanding Behaviour Patterns of Multi-agents in Digital Business Ecosystems: An Organisational Semiotics Inspired Framework // Advances in Human Factors, Business Management and Society. – 2018. – DOI: 10.1007/978-3-319-94709-9_21.
5. Международный банк реконструкции и развития [Электронный ресурс]. – Режим доступа: <https://openknowledge.worldbank.org/bitstream/handle/10986/30584/AUS0000158-RU.pdf?sequence=4&isAllowed=yMark>.
6. Dong H., Hussain F.K., Chang E. An Integrative view of the concept of Digital Ecosystem // Proceedings of the Third International Conference on Networking and Services. Washington, DC, USA. IEEE, 2007. – P. 42—44.
7. Qin S. J., Badgwell T. A. Approximate genealogy of linear MPC algorithms. // Control Engineering Practice. – 2003. – P. 733–764.
8. Nachira F., Dini P., Nicolai A.A. Network of Digital Business Ecosystems for Europe: Roots, Processes and Perspectives // Digital Business Ecosystems. – 2007.
9. Chang E., West M. Digital Ecosystems: A Next Generation of the Collaborative Environment // iiWAS. – 2006. – P. 3—24.

10. Baker K.S., Bowker G.C. Information ecology: open system environment for data, memories, and knowing // *J. Intell. Inf. Syst.*, 2007. – Vol. 29, N 1. – P. 127—144.
11. Кастельс М. Информационная эпоха: экономика, общество и культура — М.: ГУ ВШЭ. – 2000. – С. 129.
12. Fuller M. *Media Ecologies: Materialist Energies in Art and Technoculture* (Leonardo Books) // The MIT Press. – 2007.
13. Papaioannou T., Wield D., Chataway J. Knowledge ecologies and ecosystems. An empirically grounded reflection on recent developments in innovation systems theory // *Environment and Planning C: Government and Policy*. – 2009. – Vol. 27, N 2. – P. 319—339.
14. Dmitry A. Ivanov, Suresh P. Sethi, Alexandre Dolgui, Boris V. Sokolov: A survey on control theory applications to operational systems, supply chain management, and Industry 4.0. // *Annual Reviews in Control*. – 2018. – Vol. 46. – P. 134-147.
15. Seuring S. A review of modeling approaches for sustainable supply chain management // *Decision Support Systems*. – 2012. – Vol. 54, N 4. – P. 1–8.
16. Markus M. L., Loebbecke C. Commoditized digital processes and business community platforms: new opportunities and challenges for digital business strategies // *Mis Q.* – 2013. – Vol. 37, N 2. – P. 649–654.
17. Iansiti M., Levien R. Strategy as Ecology // *Harvard Business Review*. – 2004. – Vol. 82, N 3. – P. 68–81.
18. Porter M. E. *Competitive Advantage: Creating and Sustaining Superior Performance* // New York: Free Press. – 1985.
19. Brandenburger A. M., Stuart H. W. J. Value-based business strategy // *Journal of Economics & Management Strategy*. – 1996. – N 5. – P. 2–25.
20. Christopher M. *Logistics and Supply Chain Management* // *Financial Times*. – 1992.
21. Pagani M. Digital Business Strategy and Value Creation: Framing the Dynamic Cycle of Control Points // *MIS Q.* – 2013. – Vol. 37, N 2. – P. 617–632.

22. Averian A. Supply Chain Modelling as digital ecosystem // International Scientific Conference ITEMА, Budapest, Hungary. – 2017. – DOI: <https://doi.org/10.31410/EMAN.2018.620>.
23. Boley H., Chang E. Digital ecosystems: Principles and semantics // Inaugural International Conference on Distributed Event-Based Systems. IEEE, 2007. – P. 398–403.
24. Chang E., West M. Digital Ecosystems and comparison to existing collaboration environment // WSEAS Transactions on Environment and Development. – 2006. – Vol. 2, N 11. – P. 1396–1404.
25. Chang E., West M. Digital Ecosystems A Next Generation of the Collaborative Environment // Proceedings of iiWAS. – 2006. – Vol. 214. – P. 3–23.
26. Camarinha-Matos L. M., Afsarmanesh H., Galeano N., Molina A. Collaborative networked organizations - Concepts and practice in manufacturing enterprises // Computers and Industrial Engineering. – 2009. – Vol. 57, N 1. – P. 46–60.
27. Walsh W. E., Wellman M. P. Modeling Supply Chain formation in Multiagent Systems // Lecture Notes in Artificial Intelligence. – 1999. – Vol. 1788. – P. 94–101.
28. Walsh W. E., Wellman M. P. Decentralized Supply-Chain Formation: A Market Protocol and Competitive Equilibrium Analysis // Journal of Artificial Intelligence Research. – 2003. – Vol. 19. – P. 513–567.
29. Roto V., Heikkilä M. New Value Transactions: Understanding and Designing for Distributed Autonomous Organisations // Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems. ACM. – 2017. – P. 352–355. – DOI: <https://doi.org/10.1145/3064857.3064862>.
30. Официальный сайт Cargo Stream [Электронный ресурс]. – Режим доступа: <https://www.cargostream.net/>.
31. Официальный сайт Oracle Transportation Management [Электронный ресурс]. – Режим доступа: [http://www.tadviser.ru/Oracle_Transportation_Management_\(OTM\)](http://www.tadviser.ru/Oracle_Transportation_Management_(OTM)).

32. Pettey C. Prepare for the impact of digital twins [Электронный ресурс]. – 2017. — Режим доступа: <https://go.nature.com/2krzbjd>.
33. Kotilainen K., Sommarberg M., Aalto P., Järventausta P. Prosumer centric digital energy ecosystem framework // Proceedings of the 8th International Conference on Management of Digital EcoSystems. – 2016. – P. 47–51. – DOI: <https://doi.org/10.1145/3012071.3012080>.
34. Niet, Irene & Dekker, Romy & Est, Rinie. (2021). Seeking Public Values of Digital Energy Platforms // Science, Technology & Human Values. 10.1177/01622439211054430.
35. Kashevarova N., Shiboldenkov V. Production ecosystems and platforms development in the industry digital transformation conditions // MATEC Web of Conferences. – 2020. – Vol. 311, N 1. – DOI: [10.1051/matecconf/202031102017](https://doi.org/10.1051/matecconf/202031102017).
36. Bakhtadze N., Suleykin A. Industrial digital ecosystems: Predictive models and architecture development issues // Annual Reviews in Control. – 2021. – Vol. 51. – P. 56-64. – DOI: <https://doi.org/10.1016/j.arcontrol.2020.11.001>.
37. Villalonga, A., Negri E., Giacomo B., Fernando C., Guerra H., Elias R., Fumagalli L., Macchi M. A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins // Annual Reviews in Control. – 2021. – Vol. 51. – DOI: <https://doi.org/10.1016/j.arcontrol.2021.04.008>.
38. Negri E., Pandhare V., Cattaneo L., Singh J., Macchi M., Lee J. Field-synchronized Digital Twin framework for production scheduling with uncertainty // Journal of Intelligent Manufacturing. – 2021. – Vol. 32. – DOI: <https://doi.org/10.1007/s10845-020-01685-9>.
39. Darvish M., Kidd M. et al. Integrated production-distribution systems: trends and perspectives // Pesquisa Operacional. – 2021. – Vol. 41, N 1. – DOI: [10.1590/0101-7438.2021.041s1.00246080](https://doi.org/10.1590/0101-7438.2021.041s1.00246080).
40. Dhungana D., Falkner A., Haselböck A., Taupe R. Enabling Integrated Product and Factory Configuration in Smart Production Ecosystems, Vienna, Austria. // IEEE, 2017. – P. 266-273. – DOI: [10.1109/SEAA.2017.26](https://doi.org/10.1109/SEAA.2017.26).

41. Wimberley E. T. *Nested Ecology: The Place of Humans in the Ecological Hierarchy* // The Johns Hopkins University Press. – 2009.
42. Ivanov D., Dolgui A. A digital supply chain twin for managing the disruptions risks and resilience in the era of Industry 4.0 // *Production Planning and Control*. – 2020. – DOI: 10.1080/09537287.2020.1768450.
43. Poslad S., Buckle P., Hadingham R. The FIPA-OS agent platform: Open Source for Open Standards // *Proceedings of 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM)*, Manchester, UK. – 2000. – P. 355–368.
44. Nguyen N., Nahavandi S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications // *IEEE*, 2020. – Vol. 50. – P. 3826–3839.
45. Дозорцев В.М. Цифровые двойники в промышленности: Генезис, состав, технологии, платформы, перспективы // *Автоматизация в промышленности*, 2020. – № 9 – С. 3-11. – DOI: 10.25728/avtprom.2020.09.01.
46. Фролов Е.Б., Нестеров П.А., Косьяненко А.В. Правила оптимального выбора исполнительной производственной системы для машиностроительного предприятия // *Ритм машиностроения*. – 2019. – № 8 (96). – С. 10-15.
47. Jajodia, S., Albanese M., Liu P., Jajodia S., Wang C. An Integrated Framework for Cyber Situation Awareness. *Theory and Models for Cyber Situation Awareness* // *Lecture Notes in Computer Science*. – 2017. – Vol. 10030. P. 29–46.
48. Эйкхофф, П. Основы идентификации систем управления. - М.: Мир, 1975. – С. 683.
49. Райбман Н.С. Что такое идентификация? - М., Наука, 1970. – С.118.
50. Райбман Н.С. Дисперсионная идентификация. - М., Наука, Главная редакция физико-математической литературы, 1981. – С. 336.
51. Бунич А.Л., Бахтадзе Н.Н. Синтез и применение дискретных систем управления с идентификатором. - М.: Наука, 2003. – С. 232.

52. Цыпкин Я. З. Основы информационной теории идентификации. - М. Наука. – 1984. – С. 320.
53. Hunt E. Cognitive Science: Definition, Status and Questions // Annual Review of Psychology. – 1989. – Vol. 40. – P. 603-629.
54. Гроп Д. Методы идентификации систем. - М.: Мир, 1979. – С. 302.
55. Сейдж Э.П., Мелса Дж.Л. Идентификация систем управления. - М.: Наука. – 1974. – С. 248.
56. Сейдж Э.П., Мелса Дж.Л. Теория оценивания и ее применение в связи и управлении. - М.: Связь, 1976. – С. 496.
57. Карабутов Н.Н. Структурная идентификация систем: анализ динамических структур. - МГИУ, 2008. – С. 159.
58. Бунич А.Л. Вырожденные задачи синтеза многомерных дискретных систем // Труды 4-й Международной конференции «Параллельные вычисления и задачи управления» (РАСО'2008, Москва). М.: ИПУ РАН, 2008. – С. 56-60.
59. Бунич А.Л. Прогноз, инвариантность, робастность и синтез систем управления, Москва // Труды VII конференции SICPRO «Идентификация систем и задачи управления». – 2008. – С. 1264.
60. Бунич А.Л. Вырожденные задачи синтеза и робастность систем управления для объектов с ограниченной неопределенностью // Труды IV Международной конференции по проблемам управления. М.: ИПУ РАН им. В.А.Трапезникова, М.: январь 2009. - С.260-272.
61. Бунич А.Л. Идентификация линейных стационарных объектов с большим отношением сигнал/шум // Труды VIII Международной конференции SICPRO «Идентификация систем и задачи управления», М.: Январь 2009. - С.406-411.
62. Бунич А.Л. Вырожденные задачи синтеза систем управления линейными дискретными объектами. - Проблемы управления, 2009. – №5. - С. 2 – 8.
63. Бахтадзе Н.Н. Идентификационный подход к синтезу внутренних динамических моделей в задачах робастно-оптимального управления //

- Тезисы докладов международной конференции «Идентификация систем и задачи управления (SICPRO'04)». – 2004.
64. Перельман И.И. Оперативная идентификация объектов управления. - М.: Энергоиздат. – 1982. – С. 272.
 65. Дозорцев. В.М., Кнеллер Д.В. APC - Усовершенствованное управление технологическими процессами // Датчики и системы. – 2005. – №10. – С.56-62.
 66. Райбман Н.С., Чадеев В.М. Построение моделей процессов производства, Москва // «Энергия». – 1975. – С. 376.
 67. Райбман Н.С., Чадеев В.М. О концепции адаптивных систем управления с идентификатором // Автоматика и телемеханика. – 1982. – № 3. – С. 54–60.
 68. Torgashov A. Robust Decentralized Control of Reactive Distillation Process in Dimethylacetamide Production // Proceeding of 16th World Congress of IFAC. – 2005. – Vol. 16, N 1. – P. 1609.
 69. Bakhtadze N., Lototsky V., Maximov E., Pavlov B. Associative Search Models in Industrial systems // IFAC Workshop of Intelligent Manufacturing Systems, Alicante, Spain. – 2007. – P. 156-161.
 70. Natalya Bachtadze, Vladimir V. Kulba, Vladimir A. Lototsky, Evgeny M. Maximov. IDENTIFICATON-BASED APPROACH TO SOFT SENSORS DESIGN // IFAC International Workshop Intelligent Assembly and Disassembly. Alicante, Spain. 2007.
 71. Лотоцкий В.А., Максимов Е.М., Валиахметов Р.Т., Бахтадзе Н.Н. Модели ассоциативного поиска в производственных системах // Автоматизация в промышленности. – 2007. – № 10. – С. 19-21.
 72. Бахтадзе Н.Н., Валиахметов Р.Т. Применение моделей ассоциативного поиска для прогнозирования в задачах трейдинга // Проблемы управления. – 2007. – №6. – С.15-20.
 73. Кульба В.В., Максимов Е.М., Павлов Б.В. Модели ассоциативного поиска в виртуальных анализаторах технологических процессов // Труды

- Международной научно-технической конференции «Информационные технологии и математическое моделирование систем», Москва. М.: Радиотехника. – 2008. – С. 201-202.
74. Бахтадзе Н.Н., Власов С.А., Лотоцкий В.А. Идентификационный анализ в задачах автоматизации технологических процессов // Труды Международной научно-технической конференции «Информационные технологии и математическое моделирование систем», Москва. М.: Радиотехника. – 2009. – С.203-204.
75. Бахтадзе Н.Н., Герасимов А.В. Автоматизация принятия решения водителем автомобиля // Материалы 15-й Международной научно-технической конференции «Информационные системы и технологии» (ИСТ-2009, Нижний Новгород). Н. Новгород: Нижегородский государственный технический университет им. Р.Е.Алексеева. – 2009. – С. 145-150.
76. Бахтадзе Н.Н., Лотоцкий В.А., Максимов Е.М., Валиахметов Р.Т. Применение моделей ассоциативного поиска в виртуальных анализаторах // Труды VI Международной конференции «Идентификация систем и задачи управления» (SICPRO'07). М.: ИПУ РАН. – 2007. – С. 751-759.
77. Bakhtadze N., Maximov E., Valiakhmetov R. FUZZY SOFT SENSORS FOR CHEMICAL AND OIL REFINING PROCESSES // Proceedings of 17 IFAC World Congress, Seoul, Korea. – 2008. – P. 4246-4250.
78. Bakhtadze N., Lototsky V., Valiakhmetov R. Associative Search Models in Trading // Proceedings of 17 IFAC World Congress, Seoul, Korea. – 2008. – P. 4280-4284.
79. Bakhtadze N., Kulba V., Lototsky V., Maximov E., Yadylin I. Intelligent control of power generation states // Proceedings of 9th IFAC Workshop on Intelligent Manufacturing Systems. – 2008. – P. 25-32.
80. Бахтадзе Н.Н., Власов С.А., Лотоцкий В.А. Идентификационный анализ в задачах автоматизации технологических процессов // Материалы международной научно-технической конференции «Информационные

- технологии и математическое моделирование систем». М.: Радиотехника. – 2009. – С. 203-204.
81. Бахтадзе Н.Н., Лотоцкий В.А. Современные методы управления производственными процессами // Проблемы управления. – 2009. – № 3. – С.56-64.
82. Кульба В.В., Лотоцкий В.А., Бахтадзе Н.Н., Максимов Е.М. Применение моделей ассоциативного поиска в виртуальных анализаторах для задач энергетики // Международная научно-техническая конференция «Информационные технологии и математическое моделирование систем», Греция. – 2009. – С. 201-206.
83. Бахтадзе Н.Н., Лотоцкий В.А., Боровских Л.П. Новое об информационных технологиях в промышленности // Проблемы управления. – 2010. – №1. – С.79-83.
84. Бахтадзе Н.Н., Максимов Е.М. Модели ассоциативного поиска в системах управления энергосистемами // Труды Международной конференции «Идентификация систем и задачи управления (SICPRO'09)», Москва. – 2009. – С. 1628-1633.
85. Bakhtadze N., Yadikin I., Kulba V., Lototsky V., Maximov E. INTELLIGENT CONTROL OF POWER GENERATION STATES // Proceedings of 9th IFAC Workshop on Intelligent Manufacturing Systems, Szczecin, Poland. – 2008. – P. 24-32.
86. Bakhtadze N., Lototsky V., Maximov E., Pavlov B. Associative Search Models in Power Grids // Proceedings of 13th IFAC Symposium on INFORMATION CONTROL PROBLEMS IN MANUFACTURING (INCOM'09), Moscow, Russia. – 2009. – P. 2169-2173.
87. Bakhtadze N., Kulba V., Yadikin I., Lototsky V., Maximov E. Identification methods based on associative search procedure // Control and Cybernetics. – 2011. – Vol. 2, N 3. – P. 6-18.

88. Бахтадзе Н.Н., Максимов Е.М., Максимова Н.Е. Интеллектуальные алгоритмы идентификации состояния энергообъектов // Информационные технологии и вычислительные системы. – 2011. – №3. – С. 45-50.
89. Ljung, Lennart. Perspectives on System Identification // Proceeding of 17th IFAC World Congress, Seoul (Korea). – 2008. – DOI: DOI:10.1016/j.arcontrol.2009.12.001.
90. Ljung L. System Identification - Theory for the User // PTR Prentice Hall, Upper Saddle River. – 1999.
91. Pouria A., Loggerenberg F., Lang T. Big Data and Big Data Technologies. Springer. – 2017. – DOI: 10.1007/978-3-319-62990-2_3.
92. Lee H. Big Challenge in Big Data Research: Continual Dispute on Big Data Analysis // Korean circulation journal. – 2020. – Vol. 50. – P. 69-71. – DOI: 10.4070/kcj.2019.0349.
93. Hassanien E., Azar A., Snasel V., Kacprzyk, J. Big Data in Complex Systems: Challenges and Opportunities // Studies in Big Data. –2015. – Vol. 9. – DOI: 10.1007/978-3-319-11056-1.
94. Shirinzadeh S., Drechsler R. In-Memory Computing: The Integration of Storage and Processing // Springer. – 2020. – DOI: 10.1007/978-3-030-19262-4_3.
95. Luo L., Liu Y., Qian D. Survey on in-memory computing technology // Journal of Software. – 2016. – Vol. 27. – P. 2147-2167. – DOI: 10.13328/j.cnki.jos.005103.
96. Lung Hsiang-Lan. AI: from deep learning to in-memory computing // Proceedings of the SPIE. – 2019. – Vol. 53. – id 109591J. – P. 6. – DOI: 10.1117/12.2517237.
97. Yongguo J., Qiang L., Changshuai Q., Jian S., Qianqian L. Message-oriented Middleware: A Review // IEEE, 2019. – P. 88-97. – DOI: 10.1109/BIGCOM.2019.00023.
98. Sachs K., Kounev S., Appel S., Buchmann A. Benchmarking of message-oriented middleware // Proceedings of the Third ACM International Conference on Distributed Event-Based Systems. – 2009. – Vol. 44. – P. 1-2. – DOI: 10.1145/1619258.1619313.

99. Pujolle Guy. *Virtualization* // ISTE Ltd and John Wiley & Sons. – 2020. – DOI: 10.1002/9781119694748.ch1.
100. Piper Ben. *Network Virtualization* // John Wiley & Sons. – 2020. – DOI: 10.1002/9781119658795.ch10.
101. Pahl C., Jamshidi P., Zimmermann O. *Microservices and Containers* // *Software Engineering*. – 2020. – P. 115-116. – DOI: 10.18420/SE2020_34.
102. Lim J., Nieh J. *Optimizing Nested Virtualization Performance Using Direct Virtual Hardware* // *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. – 2020. – P. 557-574. – DOI: 10.1145/3373376.3378467.
103. Sharma V., Saxena H., Singh A. *Docker for Multi-containers Web Application* // *Proceedings of 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. – 2020. – P. 589-592. – DOI: 10.1109/ICIMIA48430.2020.9074925.
104. Ortega José Manuel. *DevOps and Containers Security: Security and Monitoring in Docker Containers* // BPB Publications, 1st edition. – 2020.
105. De Benedictis Marco, Liroy Antonio. *Integrity verification of Docker containers for a lightweight cloud environment* // *Future Generation Computer Systems*. – 2020. – DOI: 10.1016/j.future.2019.02.026.
106. Zadka Moshe. *Docker* // *DevOps in Python*. – 2019. – P. 147-150. DOI: 10.1007/978-1-4842-4433-3_12.
107. Rodríguez-Mazahua L., Rodríguez-Enríquez C., Sánchez-Cervantes, J., Cervantes J., García-Alcaraz, J., Alor-Hernández G. *A general perspective of Big Data: Applications, tools, challenges and trends* // *J. Supercomput.* – 2016. – Vol. 72. – P. 3073–3113. – DOI: doi:10.1007/s11227-015-1501-1.
108. Ahmed E., Yaqoob I., Hashem I., Khan I., Ahmed A., Imran M., Vasilakos A. *The role of big data analytics in Internet of Things* // *Computing Network*. – 2017 – Vol. 129. – P. 459–471.

109. Aggarwal C. Data Streams: Models and Algorithms (Advances in Database Systems) // Springer. – 2006.
110. Gama J., Bouchachia A. et al. A Survey on Concept Drift Adaptation // ACM Computing Survey. – 2014. – Vol. 46. – DOI: doi:10.1145/2523813.
111. Gepperth A., Hammer B. Incremental learning algorithms and applications // Proceedings of the European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium. – 2016.
112. Bifet A., Holmes G., Kirkby R., Pfahringer B. MOA: Massive Online Analysis. J. Mach. Learn. Res. – 2010. – Vol. 11. – P. 1601–1604.
113. Manyika, J., Chui M., Bisson P., Woetzel J., Dobbs R., Bughin J., Aharon D. Unlocking the Potential of the Internet of Things // McKinsey Global Institute: New York, NY, USA. – 2015.
114. Tu D.Q., Kayes A., Rahayu, W., Nguyen K. ISDI: A New Window-Based Framework for Integrating IoT Streaming Data from Multiple Sources // Proceedings of the International Conference on Advanced Information Networking and Applications, Matsue, Japan. Springer. – 2019. – P. 498–511.
115. Zhang L., Xiao N., Yang W., Li J. Advanced Heterogeneous Feature Fusion Machine Learning Models and Algorithms for Improving Indoor Localization // Sensors. – 2019. – Vol. 19. – P. 125. – DOI: doi:10.3390/s19010125.
116. Bouguelia M.R., Karlsson A., Pashami S., Nowaczyk S., Holst A. Mode tracking using multiple data streams // Inf. Fus. – 2018. – Vol. 43. –P. 33–46. – DOI: doi:10.1016/j.inffus.2017.11.011.
117. Kong J.L., Wang Z.N., Jin X. B., Wang, X.Y., Su T.L., Wang J.L. Semi-Supervised Segmentation Framework Based on Spot-Divergence Supervoxelization of Multi-Sensor Fusion Data for Autonomous Forest Machine Applications // Sensors. – 2018. – Vol. 18. – P. 61. – DOI: doi:10.3390/s18093061.
118. Wu J., Feng Y., Sun P. Sensor Fusion for Recognition of Activities of Daily Living // Sensors. – 2018. –Vol. 18. – P. 4029.

119. Ma M., Song Q., Gu Y., Li Y., Zhou Z. An Adaptive Zero Velocity Detection Algorithm Based on Multi-Sensor Fusion for a Pedestrian Navigation System // *Sensors*. – 2018. – Vol. 18. – P. 3261. – DOI: doi:10.3390/s18103261.
120. Zhou Y., Xue W. A Multisensor Fusion Method for Tool Condition Monitoring in Milling // *Sensors*. – 2018. – Vol. 18. – P. 3866. – DOI: doi:10.3390/s18113866.
121. Shi P., Li G.; Yuan, Y., Kuang, L. Data Fusion Using Improved Support Degree Function in Aquaculture Wireless Sensor Networks // *Sensors*. – 2018. – Vol. 18. – P. 3851. – DOI: doi:10.3390/s18113851.
122. Zhou F., Hu P., Yang S., Wen C. A Multimodal Feature Fusion-Based Deep Learning Method for Online Fault Diagnosis of Rotating Machinery// *Sensors*. – 2018. – Vol. 18. – P. 3521. – DOI: doi:10.3390/s18103521.
123. Lu K., Yang, L., Seoane F., Abtahi F., Forsman M., Lindecrantz K. Fusion of Heart Rate, Respiration and Motion Measurements from a Wearable Sensor System to Enhance Energy Expenditure Estimation // *Sensors*. – 2018. – Vol. 18. – P. 3092. – DOI: doi:10.3390/s18093092.
124. Hu J., Huang T., Zhou J., Zeng J. Electronic Systems Diagnosis Fault in Gasoline Engines Based on Multi-Information Fusion // *Sensors*. – 2018. – Vol. 18. – P. 2917. – DOI: doi:10.3390/s18092917.
125. Wu B., Huang T., Jin Y., Pan J., Song K. Fusion of High-Dynamic and Low-Drift Sensors Using Kalman Filters. // *Sensors*. – 2019. – Vol. 19. – P. 186. – DOI: doi:10.3390/s19010186.
126. Akbar A., Kousiouris G., Pervaiz H., Sancho J., Ta-Shma P., Carrez F., Moessner K. Real-Time Probabilistic Data Fusion for Large-Scale IoT Applications // *IEEE*, 2018. – Vol. 6. – P. 10015–10027.
127. Kayes A., Rahayu W., Dillon T., Chang E., Han J. Context-aware access control with imprecise context characterization for cloud-based data resources // *Future Generation Computing Systems*. – 2019. – Vol. 93. – P. 237–255. – DOI: doi:10.1016/j.future.2018.10.036.

128. Colombo P., Ferrari E. Fine-Grained Access Control Within NoSQL Document-Oriented Datastores // Springer. – 2016. – Vol. 1. – P. 127–138. – DOI: doi:10.1007/s41019-016-0015-z.
129. Kayes A.S.M., Rahayu W., Dillon T. Critical situation management utilizing IoT-based data resources through dynamic contextual role modeling and activation // Computing. – 2018. – doi:10.1007/s00607-018-0654-1.
130. Colombo P., Ferrari E. Access Control Enforcement Within MQTT-based Internet of Things Ecosystems. In Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies, New York // ACM. – 2018. – P. 223–234. – DOI: doi:10.1145/3205977.3205986.
131. Zhang K., Li X.R., Zhu Y. Optimal update with out-of-sequence measurements // IEEE, 2005. – Vol. 53. – P. 1992–2004.
132. Khaleghi B., Khamis A., Karray F. Multisensor Data Fusion: A Data-Centric Review of the State of the Art and Overview of Emerging Trends In Multisensor Data Fusion: From Algorithms and Architectural Design to Applications // Fourati. – 2015. – P. 15–33.
133. Lahat D., Adali T., Jutten C. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects // IEEE, 2015. – Vol. 103. – P. 1449–1477. – DOI: doi:10.1109/JPROC.2015.2460697.
134. García S., Ramírez-Gallego S., Luengo J., Benítez J.M., Herrera, F. Big data preprocessing: Methods and prospects // Big Data Anal. – 2016. – Vol. 1. – P. 9. – DOI: doi:10.1186/s41044-016-0014-0.
135. Zliobaite I., Gabrys B. Adaptive Preprocessing for Streaming Data // IEEE, 2014. – Vol. 26. – P. 309–321. – DOI: doi:10.1109/TKDE.2012.147.
136. Stonebraker M., Çetintemel U., Zdonik S. The 8 Requirements of Real-time Stream Processing // ACM Sigmod. – 2005. – Vol. 34. – P. 42–47. –DOI: doi:10.1145/1107499.1107504.

137. Gaber M.M., Zaslavsky A., Krishnaswamy S. Mining Data Streams: A Review // ACM Sigmod. – 2005. – Vol. 34. – P. 18–26. – DOI: doi:10.1145/1083784.1083789.
138. Domingos P., Hulten G. Mining high-speed data streams. In Proceedings of the KDD 2000 // Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Boston, USA. – 2000. – Vol. 2. – P. 4.
139. Manapragada C., Webb G.I., Salehi M. Extremely Fast Decision Tree. In Proceedings of the KDD 2018 // Proceedings of 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK. – ACM. – 2018. – P. 1953–1962. – DOI: doi:10.1145/3219819.3220005.
140. Kourtellis N., Morales G.D.F., Bifet A., Murdopo A. VHT: Vertical Hoeffding Tree // Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA. – 2016. – P. 915–922.
141. Parisi G.I., Kemker R., Part J.L., Kanan C., Wermter S. Continual lifelong learning with neural networks: A review // Neural Network. – 2019. – Vol. 113. – P. 54 – 71. – DOI: doi:10.1016/j.neunet.2019.01.012.
142. Gama J., Sebastião R., Rodrigues P.P. On evaluating stream learning algorithms // Machine Learning. – 2013. – Vol. 90. – P. 317–346.
143. Zaharia M., Xin R.S., Wendell P., Das T., Armbrust M., Dave A., Meng X., Rosen J., Venkataraman S., Franklin M.J. et al. Apache spark: A unified engine for big data processing // ACM . – 2016. – Vol. 59. – P. 56–65.
144. Kleppmann M.A., Krepis J. Kafka, Samza and the Unix philosophy of distributed data // IEEE, 2015. – Vol. 38. – P. 4–14.
145. Carbone P., Katsifodimos A., Ewen S., Markl V., Haridi S., Tzoumas K. Apache flink: Stream and batch processing in a single engine // IEEE, 2015. – Vol. 38. – P. 28-38.
146. Pathak H., Rathi M., Parekh A. Introduction to Real-Time Processing in Apache Apex // International Journal of Research in Advent Technology. NCPCL. – 2016. – P. 107-111.

147. Suleykin A., Panfilov P. Distributed Big Data Driven Framework for Cellular Network Monitoring Data // Proceedings of 2019 24th Conference of Open Innovations Association (FRUCT), Moscow. IEEE, 2019. – P. 430-436. – URL: <https://ieeexplore.ieee.org/document/8711912>.
148. Suleykin A., Panfilov P. Harnessing the Complexity of Mobile Network Data with Smart Monitoring // Proceedings of the MACSPro Workshop 2019 (MACSPro 2019), Vienna. CEUR-WS.org. – 2019. – Vol. 2478. – P.149-161.
149. Suleykin A., Panfilov P., Bakhtadze N. Industrial track: Architecting railway KPIs data processing with Big Data technologies // Proceedings of 2019 IEEE International Conference on Big Data (Big Data), Los Angeles. IEEE, 2019. – P. 2047-2056. – URL: <https://ieeexplore.ieee.org/document/9006196>.
150. Suleykin A., Panfilov P. Implementing Big Data Processing Workflows using Open Source Technologies // Proceedings of the 30th DAAAM International Symposium. Curran Associates. – 2019. Vol. 30, N 1. – P.0394-0404.
151. Bifet A., Zhang J., Fan W., He C., Zhang J., Qian J., Holmes G., Pfahringer B. Extremely Fast Decision Tree Mining for Evolving Data Streams // Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Canada. ACM. – 2017. – P. 1733–1742. – DOI: [doi:10.1145/3097983.3098139](https://doi.org/10.1145/3097983.3098139).
152. Montiel J., Read J., Bifet A., Abdessalem T. Scikit-Multiflow: A Multi-output Streaming Framework // J. Mach. – 2018. – Vol. 19. – P. 2915–2914.
153. Fortuna B., Rupnik J., Brank J., Fortuna C., Jovanoski V., Karlovcec M., Kazic B., Kenda K., Leban G., Mladenić, D., et al. QMiner: Data Analytics Platform for Processing Streams of Structured and Unstructured Data, Montreal, Canada. // Proceedings of the Software Engineering for Machine Learning Workshop. Neural Information Processing Systems. – 2014. – DOI: [10.13140/RG.2.1.4054.1283](https://doi.org/10.13140/RG.2.1.4054.1283).
154. Kenda K., Kazic B., Novak E., Mladenić D. Streaming Data Fusion for the Internet of Things // Sensors. – 2019. – Vol. 19. – P. 1955. – DOI: [10.3390/s19081955](https://doi.org/10.3390/s19081955).

155. Yi W., Teng F., Xu J. Novel Stream Data Mining Framework Under the Background of Big Data // *Cybernetics and Information Technologies*. – 2016. – Vol. 16. – P. 69–77. – DOI: doi:10.1515/cait-2016-0053.
156. Marz N., Warren J. *Big Data: Principles and Best Practices Of Scalable Real-Time Data Systems* // Manning Publications. – 2015.
157. Ta-Shma P., Akbar A., Gerson-Golan G., Hadash G., Carrez F., Moessner K. An Ingestion and Analytics Architecture for IoT Applied to Smart City Use Cases // *IEEE*, 2018. – Vol. 5. – P. 765–774.
158. Kolomvatsos K., Anagnostopoulos C., Hadjiefthymiades S. Data Fusion and Type-2 Fuzzy Inference in Contextual Data Stream Monitoring // *IEEE*, 2017. – Vol. 47. – P. 1839–1853. – DOI: doi:10.1109/TSMC.2016.2560533.
159. Bakhtadze N., Lototsky V. *Knowledge-Based Models of Nonlinear Systems Based on Inductive Learning* // *New Frontiers in Information and Production Systems Modelling and Analysis Incentive Mechanisms, Competence Management, Knowledge-Based Production*. Springer. – 2016. – P. 85–104.
160. Moore E. On the reciprocal of the general algebraic matrix // *Bulletin of American Mathematical Society*. – 1920. – Vol. 26. – P. 394–395.
161. Penrose R. A generalized inverse for matrices // *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press. – 1955. – Vol. 51, N 3. – P. 406-413.
162. MacQueen J. Some methods for classification and analysis of multivariate observations // *Berkeley Symposium on Mathematical Statistics and Probability*. – 1967. – Vol. 5, N 1. – P. 281-297.
163. Colombani C., Legarra A., Fritz S., Guillaume F., Croiseau P., Ducrocq V., Robert-Granié C. Application of Bayesian least absolute shrinkage and selection operator (LASSO) and Bayes methods for genomic selection in French Holstein and Montbéliarde breeds // *Journal of Dairy Science*. – 2013. – Vol. 96, N 1. – P. 575-591.

164. Bakhtadze N., Lototsky V., Vlasov S., Sakrutina E. Associative Search and Wavelet Analysis Techniques in System Identification // Proceedings of the 16-th IFAC Symposium on System Identification. IFAC Publication. – 2012. – P. 1227-1232.
165. Официальный сайт Apache Ignite: [сайт]. Электронный ресурс – Режим доступа: <https://ignite.apache.org/>
166. Официальный сайт Apache NIFI: [сайт]. Электронный ресурс – Режим доступа: <https://nifi.apache.org/>.
167. Официальный сайт Scikit-learn: [сайт]. Электронный ресурс – Режим доступа: <https://scikit-learn.org/stable/>.
168. Официальный сайт Grafana: [сайт]. Электронный ресурс – Режим доступа: <https://grafana.com/>.
169. Официальный сайт PostgreSQL: [сайт]. Электронный ресурс – Режим доступа: <https://postgresql.org/>.
170. Официальный сайт Docker: [сайт]. Электронный ресурс – Режим доступа: <https://docker.com/>.

ПРИЛОЖЕНИЕ 1

УТВЕРЖДАЮ

Заместитель генерального директора
ПАО «КАМАЗ» - директор по
развитию



И.Ф. Гумеров

«14» 09 2022 г.

АКТ ВНЕДРЕНИЯ

результатов научно-исследовательской работы
Сулейкина Александра Сергеевича

Настоящим актом подтверждается, что результаты диссертационной работы «Методы анализа и синтез архитектуры цифровых производственных экосистем», представленной на соискание ученой степени кандидата технических наук по специальности 2.3.3 – «Автоматизация и управление технологическими процессами и производствами (технические науки)» использованы в ПАО «КАМАЗ» при создании прототипа модуля прогнозирования в части:

- цифровой инфраструктуры для хранения, обработки, анализа, прогнозирования и визуализации гетерогенных данных.

В прототип модуля прогнозирования входят следующие результаты диссертации: концептуальная и функциональная архитектуры системы управления цифровой экосистемой производственного предприятия (ЦЭС); прототип системы прогнозирования состояния загрузки серверного комплекса для управления ресурсами.

Директор департамента цифровых
систем проектирования - главный
конструктор цифровых систем
проектирования

А.В. Пуртов

Руководитель службы
конструкторских
и научно-исследовательских
расчетов, к.т.н., доцент

В.С. Карабцев