

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

На правах рукописи

Пряхина Дарья Игоревна

**Цифровые двойники для решения задач управления и
развития распределенных центров сбора, хранения и
обработки данных**

Научная специальность 2.3.1. Системный анализ, управление и обработка
информации, статистика

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук
Кореньков Владимир Васильевич

Дубна

2024

Оглавление

Введение.....	4
Глава 1. Анализ задачи построения цифровых двойников распределенных центров сбора, хранения и обработки данных	16
1.1. Распределенные центры сбора, хранения и обработки данных	16
1.2. Состояние проблемы моделирования распределенных центров сбора, хранения и обработки данных	17
1.3. Цифровой двойник: определение, примеры применения	20
1.4. Возможности использования цифровых двойников центров хранения и обработки данных.....	22
1.5. Выводы к главе 1	24
Глава 2. Модели, методы и алгоритмы построения цифровых двойников распределенных центров сбора, хранения и обработки данных	25
2.1. Предварительные замечания.....	25
2.2. Требования к цифровым двойникам распределенных центров сбора, хранения и обработки данных	26
2.3. Метод построения цифровых двойников распределенных центров сбора, хранения и обработки данных	29
2.4. Алгоритмы для реализации метода построения цифровых двойников распределенного центра сбора, хранения и обработки данных	32
2.5. Модель данных	37
2.6. Выводы к главе 2	43
Глава 3. Реализация алгоритмов и разработка специального программного обеспечения для создания цифровых двойников и взаимодействия с ними.....	44
3.1. Средства реализации.....	44
3.2. Реализация ядра цифрового двойника	45
3.3. Реализация модуля для взаимодействия пользователя с цифровым двойником	54
3.4. Разработка специального программного обеспечения	58
3.5. Выводы к главе 3	61

Глава 4. Верификация и опытная эксплуатация специального программного обеспечения для создания цифровых двойников	62
4.1. Верификация ядра цифрового двойника.....	62
4.2. Применение специального программного обеспечения для построения цифрового двойника вычислительной инфраструктуры эксперимента VM@N комплекса NICA	75
4.3. Применение специального программного обеспечения для построения цифрового двойника вычислительной системы онлайн-фильтра данных эксперимента SPD комплекса NICA	87
4.4. Выводы к главе 4	91
Заключение	93
Словарь терминов.....	96
Список литературы	97
Приложение 1. Описание сущностей модели данных	104
Приложение 2. Инструкция для пользователя по работе со специальным программным обеспечением	110
Приложение 3. Свидетельство о регистрации программы для ЭВМ.....	119
Приложение 4. Письма о применении результатов диссертационного исследования.....	121
Приложение 5. Акт о внедрении результатов диссертационного исследования в учебный процесс	125

Введение

Актуальность темы исследования

Центры обработки данных (ЦОД) широко применяются в различных областях: в науке для хранения и дальнейшей обработки данных проводимых экспериментов; в бизнесе для предоставления услуг ИТ-аутсорсинга; в государственных организациях для автоматизации внутренних процессов, например, документооборота; в системах безопасности городской среды, транспорта и промышленных предприятий для обеспечения сбора, хранения и анализа систем контроля доступа, видеонаблюдения и т.п. Для хранения и интенсивной обработки данных существует множество вычислительных систем различного масштаба. В данном исследовании рассмотрены распределенные системы (центры), предназначенные для сбора, хранения и обработки сверхбольших объемов данных (до 10^{18} байт), как наиболее актуальные и применимые для вычислительных инфраструктур масштабных научных экспериментов.

Распределенные центры представляют собой сложный комплекс, включающий в себя не только мощные вычислительные ресурсы и ресурсы для хранения данных, но и сетевое оборудование, отвечающее за обмен данными внутри системы и связь с внешними потребителями, инженерные системы, системы безопасности, системы мониторинга и прочее. Такие сложные системы нуждаются в качественном проектировании, чтобы как можно точнее определить параметры всех процессов, которые будут происходить в рассматриваемой системе, выбрать необходимое оборудование для функционирования системы с учетом текущих потребностей и перспективы развития в будущем. Помимо этого, в период эксплуатации распределенных центров возникает необходимость в масштабировании с целью повышения эффективности использования оборудования, ускорения процессов обработки данных и т.д. В зависимости от области применения и типов решаемых задач требования к масштабированию могут быть разными, поэтому необходим инструмент, с помощью которого можно проанализировать эффективность и надежность возможных сценариев развития распределенных центров сбора, хранения и обработки данных (РЦОД). При этом необходимо учитывать все процессы, происходящие в РЦОД, включая стратегии управления потоками задач, а также параметры потоков данных для хранения и обработки.

Таким инструментом обычно являются различные программные пакеты моделирования. Основным элементом таких пакетов является ядро — компьютерная программа, которая осуществляет непосредственное моделирование систем. Для хранения входных данных и результатов моделирования некоторые пакеты используют базы данных (БД). Существующие средства моделирования не позволяют качественно проводить перечисленные выше исследования по следующим причинам:

- не учитываются современные тенденции построения распределенных центров, включающих разнородные модули обработки данных, в том числе облачные структуры и суперкомпьютеры, а также системы иерархической памяти и многое другое;
- не рассматриваются такие важные критерии функционирования системы, как потери данных в зависимости от типа выбранного оборудования, вероятности сбоев, отказов и изменений в производительности вычислительных ресурсов и хранилищ данных;
- детальное описание параметров компьютерных и сетевых систем в БД приводит к затруднениям в принятии решения по выбору системной конфигурации;
- не предусмотрен поиск конфигурации оборудования, удовлетворяющей заданным критериям;
- для каждой моделируемой инфраструктуры требуется адаптировать ядро моделирования путем изменения программного кода;
- отсутствует интерфейс для взаимодействия пользователя с программой моделирования.

Следовательно, в процессе создания и совершенствования РЦОД существует проблема решения таких важных задач, как:

- проверка различных сценариев масштабирования РЦОД, учитывая требования к потокам данных и задач;
- формирование и улучшение стратегии управления потоками задач для эффективного распределения ресурсов РЦОД при обработке данных;
- анализ используемых ресурсов и оценка необходимого количества ресурсов для конкретных задач согласно предъявленным требованиям к РЦОД.

В настоящее время активно развиваются технологии разработки проблемно-ориентированных систем управления и принятия решений на основе цифровых двойников (ЦД), которые могут быть применены к решению перечисленных задач. Таким образом, представляется актуальным разработка метода и реализация алгоритмов построения цифровых двойников РЦОД для решения задач проектирования, управления и развития РЦОД, в том числе для проверки эффективности и надежности их функционирования. Использование же ЦД на этапе проектирования РЦОД в качестве прототипов позволит проанализировать режимы работы будущих систем, повысить их надежность. Следовательно, разработанные методы и алгоритмы в последствии могут применяться к широкому классу задач построения и развития РЦОД для больших научных экспериментов и крупных проектов.

Основные требования к ЦД — это адекватность моделирования (соответствие модели РЦОД реальной системе по некоторому перечню характеристик), наглядность представления исходных данных и результатов, а также освобождение потенциального пользователя от участия в

разработке и сопровождении ядра ЦД, т.е. компьютерной программы, где реализованы алгоритмы моделирования различных процессов, происходящих в РЦОД.

Степень разработанности проблемы

Первое упоминание о ЦД прозвучало в 2002 году [1]. За прошедшие 20 лет эта технология нашла применение во всем мире практически во всех областях науки и деятельности человека [2]. В связи с прогрессивной тенденцией создания ЦОД для различных задач науки и бизнеса [3, 4, 5], технологии ЦД стали применять и в этой области.

Например, американские компании *Future Facilities* [6] и *Sunbird DCIM* [7] занимаются разработкой программного обеспечения и созданием цифровых двойников ЦОД. Но, во-первых, эти ЦД представляют собой виртуальные копии ЦОД, которые территориально расположены в одном физическом пространстве, а во-вторых, ЦОД рассматриваются только с точки зрения инженерной инфраструктуры. На данный момент ни в России, ни за рубежом не существует инструмента, позволяющего построить цифровой двойник РЦОД с учетом потоков данных и задач, который можно использовать для оценки и выбора необходимого количества ресурсов для конкретных задач согласно предъявленным требованиям к ЦОД.

Эффективность создания, использования и развития технологии ЦД обуславливается качеством построенной модели, поэтому определяющую роль при построении ЦД играют подходы к моделированию [8]. Для повышения качества модели ЦД необходимо учитывать вероятностные характеристики процессов, происходящих в моделируемой системе. Применение и использование ЦД обеспечивают различные программно-инструментальные средства и информационные технологии. Так как ЦД необходимо модернизировать в процессе эксплуатации по результатам апробации, требуется предусмотреть критерии оценки качества ЦД на основе результатов его работы.

Следует отметить, что важно учитывать такую функциональную возможность ЦД, которая позволит подобрать конфигурацию оборудования для РЦОД. В качестве критериев выбора необходимо использовать не только технические характеристики оборудования, но и другие, например, временные показатели. В итоге из множества допустимых конфигураций можно выбрать несколько потенциально конкурентоспособных вариантов.

Таким образом, из вышесказанного следует актуальность выбранной темы исследования.

Цель и задачи исследования

Целью работы является разработка метода и алгоритмов создания цифровых двойников для описания распределенных систем, принятия решений по выбору конфигураций оборудования в рамках задачи масштабирования и развития распределенных центров сбора, хранения и обработки данных, управления ресурсами и процессами, происходящими в них.

Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ предметной области.
2. Разработать метод построения цифровых двойников РЦОД, который в отличие от существующих выполняет моделирование распределенных центров с учетом характеристик потоков задач и данных для хранения и обработки, а также вероятностей появления изменений в процессах, происходящих в РЦОД.
3. Реализовать алгоритмы, структуру БД и веб-интерфейс пользователя для построения и запуска ЦД, а также предоставления графической информации о результатах его работы. На основе созданных моделей, методов и алгоритмов разработать специальное программное обеспечение (СПО), позволяющее сравнивать эффективность функционирования РЦОД в зависимости от различных конфигураций оборудования.
4. Провести верификацию и опытную эксплуатацию СПО.

Научная новизна

Новые научные результаты, полученные лично автором, состоят в следующем.

1. Предложен и разработан новый метод построения и использования цифровых двойников РЦОД, отличительной особенностью которого является возможность моделирования процессов обработки и хранения данных, учитывая характеристики потоков данных и задач, вероятности сбоев, отказов и изменений в производительности оборудования и других процессов, происходящих в моделируемой системе.

2. Разработаны и реализованы алгоритмы описания инфраструктуры распределенной системы и формирования ее виртуального образа.

3. Разработана проблемно-ориентированная система принятия решений для задач управления и оптимизации с целью улучшения технических характеристик РЦОД на основе моделей ЦД, адекватность которых подтверждена на примере вычислительной инфраструктуры эксперимента $VM@N^1$ комплекса $NICA^2$.

4. Научно обоснована конфигурация РЦОД для экспериментов $VM@N$ и SPD^3 комплекса $NICA$.

Теоретическая значимость

Теоретическая значимость исследования заключается в развитии метода моделирования распределенных вычислительных систем, которые оперируют очень большими объемами данных, требующих надежного хранения и сложной системы обработки.

¹ *Baryonic Matter at Nuclotron* – экспериментальная установка по изучению барионной материи.

² *Nuclotron based Ion Collider fAcility* – ускорительный комплекс, который создается на базе Объединенного института ядерных исследований (Дубна, Россия) с целью изучения свойств плотной барионной материи.

³ *Spin Physics Detector* – эксперимент, проводимый на детекторе спиновой физики.

Доказана применимость разработанной в диссертации методологии для создания цифровых двойников РЦОД на основе результатов верификации модели вычислительной инфраструктуры существующего эксперимента.

Практическая значимость

Практическая значимость состоит в применении полученных результатов для повышения эффективности функционирования, качества и надежности сложных систем сбора, хранения и обработки данных.

Разработанное СПО может применяться для широкого класса задач в области проектирования, построения и развития РЦОД, в том числе помогает выбрать несколько потенциально конкурентоспособных вариантов из множества допустимых конфигураций оборудования.

Получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» от 14 августа 2023.

Решена задача поиска конфигурации оборудования для системы сбора, хранения и обработки данных эксперимента *VM@N* комплекса *NISA* в Объединенном институте ядерных исследований (ОИЯИ), что подтверждается письмом о применении.

Результаты работы применяются при проектировании вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NISA* в ОИЯИ, что подтверждается письмом о применении.

Результаты работы используются в учебном процессе Федерального государственного бюджетного образовательного учреждения высшего образования «Университет «Дубна» в курсе «Распределенные вычисления и облачные технологии» для подготовки магистров по направлению 27.04.03 Системный анализ и управление по профилю «Цифровые платформы и аналитика больших данных», что подтверждается актом о внедрении.

Методология и методы исследования

Теоретические исследования построены на реализации принципов системного подхода и методов системного анализа при исследовании системных связей и закономерностей функционирования сложных систем, которыми являются современные центры сбора, хранения и обработки данных, в том числе распределенные.

Практические результаты получены на основе использования современных архитектурных решений и средств для разработки программного обеспечения, веб-приложений и БД.

Степень достоверности

Достоверность результатов диссертации обеспечивается корректным применением принципов системного подхода и методов системного анализа при исследовании системных связей и закономерностей функционирования сложных систем.

Предложенные решения основываются на изучении и критическом осмыслении научных трудов и разработок в области организации распределенного хранения и надежной передачи больших объемов данных физических экспериментов.

Достоверность рекомендаций и выводов по результатам диссертационного исследования подтверждается практикой применения разработанных методов при проектировании и развитии вычислительных инфраструктур для крупных экспериментов в области физики высоких энергий.

Апробация результатов

Разработанные в данной работе методы и СПО были проверены и показали свою эффективность при построении ЦД распределенного центра сбора, хранения и обработки данных эксперимента *BM@N* комплекса *NICA*, что подтверждается письмом о применении.

Полученные результаты обсуждались на следующих ежегодных совещаниях коллаборации эксперимента *BM@N*.

1. 5th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 20-21 April 2020 (JINR, Dubna, Russia).
2. 6th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 26-27 October 2020 (JINR, Dubna, Russia).
3. 8th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 03-08 October 2021 (JINR, Dubna, Russia).
4. 9th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 13-16 September 2022 (JINR, Dubna, Russia).
5. 10th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 14-19 May 2023 (SPbU, St. Petersburg, Russia).
6. 11th Collaboration Meeting of the *BM@N* Experiment at the *NICA* Facility, 28-30 November 2023 (JINR, Dubna, Russia).

Разработанное в данной работе СПО в настоящее время применяется для проектирования вычислительной системы эксперимента *SPD* комплекса *NICA*, в частности для онлайн-фильтра данных, что отражено в письме о применении. Полученные результаты и планы о дальнейшем сотрудничестве обсуждались на совещании коллаборации эксперимента *SPD*, а именно: VI *SPD* Collaboration Meeting and Workshop on Information Technology in Natural Sciences, 23-27 October 2023 (Samara University, Samara, Russia).

Основные положения и результаты работы докладывались, обсуждались и получили одобрение специалистов на следующих российских и международных научных мероприятиях.

1. The 8th International Conference “Distributed Computing and Grid-technologies in Science and Education” (GRID 2018), 10-14 September 2018 (JINR, Dubna, Russia).

2. Всероссийская научно-практическая конференция «Природа, общество, человек» «Университета «Дубна», 21-23 Ноября 2018 (Университет «Дубна», Дубна, Россия).

3. The International Conference “Mathematical Modeling and Computational Physics” (ММСП2019), 1-4 July 2019 (Stará Lesná, Slovakia).

4. XXVII Международная конференция «Математика. Компьютер. Образование» симпозиум с международным участием «Биофизика сложных систем: вычислительная и системная биология, молекулярное моделирование», 27 Января – 1 Февраля 2020 (Университет «Дубна», Дубна, Россия).

5. The XXIV International Scientific Conference of Young Scientists and Specialists (AYSS-2020), 09-13 November 2020 (JINR, Dubna, Russia). Получен диплом за лучший доклад конференции.

6. The 9th International Conference “Distributed Computing and Grid-technologies in Science and Education” (GRID 2021), 05-09 July 2021 (JINR, Dubna, Russia).

7. 56th meeting of the PAC for Particle Physics, 24-25 January 2022 (JINR, Dubna, Russia).

8. The XXVI International Scientific Conference of Young Scientists and Specialists (AYSS-2022), 24-28 October 2022 (JINR, Dubna, Russia).

9. 57th meeting of the PAC for Particle Physics, 23 January 2023 (JINR, Dubna, Russia).

10. The XIII Conference (with international participation) “Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems” (ИТТММ 2023), 17-21 April 2023 (RUDN, Moscow, Russia).

11. The 10th International Conference “Distributed Computing and Grid-technologies in Science and Education” (GRID 2023), 03-07 July 2023 (JINR, Dubna, Russia).

12. The XXVII International Scientific Conference of Young Scientists and Specialists (AYSS-2023); 29 October – 03 November 2023 (JINR, Dubna, Russia). Получен диплом за лучший доклад конференции.

13. 59th meeting of the PAC for Particle Physics, 22 January 2024 (JINR, Dubna, Russia).

В 2017, 2019 и 2022 годах исследование было поддержано стипендией имени М.Г. Мещерякова в Лаборатории информационных технологий (ЛИТ) ОИЯИ за работу в области поддержки экспериментальной и теоретической физики, а в 2018 и 2020 годах — стипендией имени Н.Н. Говоруна в ЛИТ за работу в области информационной, компьютерной и сетевой поддержки деятельности ОИЯИ. В 2021 и 2023 годах получен грант для молодых научных сотрудников ОИЯИ на разработку методов и комплекса программ для моделирования центров хранения и обработки данных. Лауреат II степени Премии ОИЯИ для молодых ученых и специалистов за 2023 год в номинации «Научно-технические прикладные работы».

Публикации

Основные научные результаты диссертации были опубликованы в перечисленных ниже работах.

1. Трофимов В.В., Нечаевский А.В., Ососков Г.А., Пряхина Д.И. Вероятностно-стоимостной подход к оптимизации распределенных систем хранения данных физических экспериментов // CEUR Workshop Proceedings. 2018. Т. 226. С. 393–399.

2. Кореньков В.В., Пряхина Д.И., Нечаевский А.В., Ососков Г.А., Трофимов В.В. Моделирование центров хранения и обработки данных с учетом экономических составляющих // Системный анализ в науке и образовании. 2018. № 4. С. 1–8.

3. Korenkov V., Nechaevskiy A., Ososkov G., Priakhina D., Trofimov V. A probabilistic approach of the simulation of data processing centers // European Physical Journal Web of Conferences. January 2020. Vol. 226. P. 03012.

4. Priakhina D., Trofimov V., Ososkov G., Gertsenberger K. Data center simulation for the BM@N experiment of the NICA project // AIP Conference Proceeding. 2021. Vol. 2377. P. 040007.

5. Priakhina D., Korenkov V., Gertsenberger K., Trofimov V. Simulation of Data Processing for the BM@N Experiment of the NICA Complex // CEUR Workshop Proceedings. 2021. Vol. 3041. pp. 483–487.

6. Priakhina D., Korenkov V., Trofimov V., Gertsenberger K. Simulation Results of BM@N Computing Infrastructure // Physics of Particles and Nuclei Letters. 2023. Vol. 20. No. 5. pp. 1272–1275.

7. Пряхина Д.И., Кореньков В.В. Актуальность создания цифрового двойника для управления распределенными центрами сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 262–271.

8. Пряхина Д.И., Кореньков В.В., Трофимов В.В. Метод построения цифровых двойников для решения задач эффективного управления и развития распределенных центров сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 272–281.

9. Пряхина Д.И., Кореньков В.В., Трофимов В.В., Герценбергер К.В. Верификация программы моделирования для создания цифровых двойников распределенных центров сбора, хранения и обработки данных // International Journal of Open Information Technologies. Январь 2024. Т. 12. № 1. С. 118–128.

Получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» от 14 августа 2023 года.

Структура и объем диссертации

Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы и пяти приложений.

Во введении отражена актуальность работы, поставлена цель исследования, а также задачи для ее достижения, обоснована научная новизна работы, сформулированы положения, выносимые на защиту, показана практическая значимость полученных результатов.

В первой главе сделан анализ задачи построения цифровых двойников распределенных центров хранения и обработки данных. Для этого рассмотрены примеры распределенных центров и описаны проблемы моделирования РЦОД. Помимо этого, приведен обзор понятия технологии ЦД, представлены примеры применения, а также изложены возможности использования ЦД центров хранения и обработки данных.

Во второй главе предложен новый метод построения цифровых двойников РЦОД. Представлено формальное описание метода, функциональные и нефункциональные требования к ЦД, в том числе к компьютерной программе, которая осуществляет непосредственное моделирование систем, и к модулю для взаимодействия пользователя с ЦД. Согласно описанным требованиям разработаны алгоритмы для реализации метода построения цифровых двойников РЦОД. В связи с тем, что функционирование алгоритмов невозможно без БД, приведена логическая модель данных с детальным описанием сущностей, их атрибутов и взаимосвязей.

Третья глава посвящена программной реализации алгоритмов построения цифровых двойников РЦОД. Представлены диаграммы, которые демонстрируют общую структуру иерархии разработанных классов компьютерных программ, а также жизненный цикл некоторых объектов. Показана архитектура СПО, которое включает в себя все разработанные алгоритмы и модули для реализации метода построения цифровых двойников РЦОД. Описаны средства реализации, которые использовались, как для разработки отдельных алгоритмов, так и для объединения компонент в единый комплекс. Приведены характеристики оборудования, необходимого для функционирования СПО.

В четвертой главе рассматриваются результаты верификации ядра ЦД, которые подтверждают адекватность разработанных моделей и доказывают возможность их дальнейшего использования. Представлены результаты применения разработанного СПО на примерах построения цифрового двойника РЦОД для решения задач проектирования, повышения эффективности функционирования, качества и надежности сложных систем сбора, хранения и обработки данных существующих экспериментов в области физики высоких энергий.

В заключении подведены итоги исследования, сформулированы основные выводы и возможные перспективы развития данного исследования.

Общий объем диссертации составляет 126 страниц, включая 94 рисунка, 2 таблицы и 14 формул. Список литературы включает 82 наименования.

Основные научные результаты

1. Исследованы системные связи и закономерности функционирования сложных систем, которыми являются РЦОД, с использованием принципов системного подхода и методов системного анализа. Разработаны методы описания распределенных систем, принятия решений по выбору конфигураций оборудования, управления ресурсами и процессами сложных систем. Разработаны модели, методы и алгоритмы построения цифровых двойников РЦОД. Реализованы алгоритмы, структура БД и веб-интерфейс пользователя для построения и запуска ЦД, а также предоставления графической информации о результатах его работы. Результаты изложены в следующих публикациях: **Пряхина Д.И.**, Кореньков В.В. Актуальность создания цифрового двойника для управления распределенными центрами сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 262–271 (на страницах 267–268); **Пряхина Д.И.**, Кореньков В.В., Трофимов В.В. Метод построения цифровых двойников для решения задач эффективного управления и развития распределенных центров сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 272–281 (на страницах 275–278); Трофимов В.В., Нечаевский А.В., Ососков Г.А., **Пряхина Д.И.** Вероятностно-стоимостной подход к оптимизации распределенных систем хранения данных физических экспериментов // CEUR Workshop Proceedings. 2018. Т. 226. С. 393–399 (на страницах 394–395); Кореньков В.В., **Пряхина Д.И.**, Нечаевский А.В., Ососков Г.А., Трофимов В.В. Моделирование центров хранения и обработки данных с учетом экономических составляющих // Системный анализ в науке и образовании. 2018. № 4. С. 1–8 (на странице 7); Korenkov V., Nechaevskiy A., Ososkov G., **Priakhina D.**, Trofimov V. A probabilistic approach of the simulation of data processing centers // European Physical Journal Web of Conferences. January 2020. Vol. 226. P. 03012 (на странице 03012-2).

2. На основе созданных моделей, методов и алгоритмов разработано СПО, позволяющее сравнивать эффективность функционирования РЦОД в зависимости от различных конфигураций оборудования. Получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» (от 14 августа 2023 года).

3. Проведена верификация ядра ЦД на примере центра сбора, хранения и обработки данных эксперимента *VM@N* комплекса *NISA*. Адекватность модели доказана по нескольким показателям путем сравнения результатов работы цифрового двойника со статистическими данными мониторинга вычислительной инфраструктуры рассматриваемого эксперимента. Результат изложен в следующей публикации: **Пряхина Д.И.**, Кореньков В.В., Трофимов В.В.,

Герценбергер К.В. Верификация программы моделирования для создания цифровых двойников распределенных центров сбора, хранения и обработки данных // *International Journal of Open Information Technologies*. Январь 2024. Т. 12. № 1. С. 118–128 (на странице 126).

4. Получены рекомендации, которые учтены при проектировании и развитии вычислительных инфраструктур крупных экспериментов в области физики высоких энергий. Проведена опытная эксплуатация СПО при построении ЦД вычислительной инфраструктуры эксперимента *BM@N*. Получена наиболее подходящая конфигурация оборудования для обработки данных за минимальное время, а также выбрана стратегия управления потоком задач и распределения нагрузки на вычислительные ресурсы. Проведена опытная эксплуатация СПО при построении ЦД вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NICA*. Результаты позволили оценить требуемые параметры оборудования для хранения, обработки и передачи данных онлайн-фильтра с учетом планируемых характеристик потоков данных эксперимента. Результаты изложены в следующих публикациях: **Priakhina D.**, Trofimov V., Ososkov G., Gertsenberger K. Data center simulation for the *BM@N* experiment of the *NICA* project // *AIP Conference Proceeding*. 2021. Vol. 2377. P. 040007 (на страницах 040007-3–040007-5); **Priakhina D.**, Korenkov V., Gertsenberger K., Trofimov V. Simulation of Data Processing for the *BM@N* Experiment of the *NICA* Complex // *CEUR Workshop Proceedings*. 2021. Vol. 3041. pp. 483–487 (на страницах 485–486); **Priakhina D.**, Korenkov V., Trofimov V., Gertsenberger K. Simulation Results of *BM@N* Computing Infrastructure // *Physics of Particles and Nuclei Letters*. 2023. Vol. 20. No. 5. pp. 1272–1275 (на страницах 1273–1274).

Положения, выносимые на защиту

1. Разработан метод построения цифрового двойника, который позволяет описывать распределенные центры сбора, хранения и обработки данных, учитывая потоки данных и задач, а также процессы, происходящие в РЦОД. Возможность моделирования процессов обработки и хранения данных, учитывая характеристики потоков данных и задач, вероятности сбоев, отказов и изменений в производительности оборудования и других процессов, происходящих в моделируемой системе, является отличительной особенностью нового метода. Разработанный подход к построению цифрового двойника реализует в том числе методы описания распределенных систем, принятия решений по выбору конфигураций оборудования, управления ресурсами и процессами сложных систем.

2. Разработаны алгоритмы, на основе которых создано специальное программное обеспечение, используемое для принятия решений по выбору конфигурации оборудования распределенных центров сбора, хранения и обработки данных согласно заданным требованиям. Специальное программное обеспечение позволяет сравнивать эффективность функционирования распреде-

ленных систем в зависимости от различных конфигураций оборудования. Реализация алгоритмов, структуры базы данных и веб-интерфейса пользователя для построения и запуска цифровых двойников, а также предоставления графической информации о результатах его работы, осуществлялась с использованием современных архитектурных решений и средств для разработки программного обеспечения, веб-приложений и баз данных.

3. Подтверждена адекватность построенных методов и алгоритмов на примере вычислительной инфраструктуры существующего эксперимента. Верификация проведена на примере центра сбора, хранения и обработки данных эксперимента *VM@N* комплекса *NICA*. Адекватность модели оценивалась по нескольким показателям. Получившиеся значения доказали, что отклонения результатов работы цифрового двойника от результатов работы реальных оборудования распределенных центров сбора, хранения и обработки данных позволяют использовать построенные методы и алгоритмы для решения задач управления и развития распределенных систем. Результаты опытной эксплуатации специального программного обеспечения при построении цифровых двойников вычислительной инфраструктуры эксперимента *VM@N* и вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NICA* позволили оценить требуемые параметры оборудования для хранения, обработки и передачи данных с учетом планируемых характеристик потоков данных экспериментов, выбрать стратегию управления потоком задач и распределения нагрузки на вычислительные ресурсы. Таким образом, получены рекомендации, которые учтены при проектировании и развитии вычислительных инфраструктур крупных экспериментов в области физики высоких энергий. Результаты применения специального программного обеспечения доказали работоспособность и высокое качество разработанных в диссертации моделей и алгоритмов.

Глава 1. Анализ задачи построения цифровых двойников распределенных центров сбора, хранения и обработки данных

1.1. Распределенные центры сбора, хранения и обработки данных

Развитие научных исследований в физике высоких энергий, астрофизике, биологии, науках о Земле, химии, а также в медицине, нанотехнологиях, промышленности, бизнесе и других направлениях человеческой деятельности требует совместной работы многих организаций по обработке большого объема данных в относительно короткие сроки. Именно для этого создаются и развиваются РЦОД, которые способны передавать и хранить огромные массивы данных, а также служат универсальной эффективной инфраструктурой для высокопроизводительных распределенных вычислений и обработки данных.

Распределенные системы предназначены, как правило, для сбора, хранения и обработки сверхбольших объемов данных и представляют собой географически распределенную инфраструктуру, объединяющую множество ресурсов разных типов (процессоры, долговременная и оперативная память, хранилища, сети), доступ к которым пользователь может получить из любой точки, независимо от места их расположения. В такой системе предполагается коллективный разделяемый режим доступа к ресурсам [9]. Это значит, что распределенные центры сбора, хранения и обработки данных включают в себя не только мощные гибридные вычислительные ресурсы и ресурсы для хранения данных, но и сетевое оборудование, отвечающее за обмен данными внутри системы и связь между центрами и с внешними потребителями, инженерные системы, системы безопасности, системы мониторинга и прочее.

В качестве примера РЦОД можно привести инфраструктуру для обработки данных Большого адронного коллайдера (БАК) — эксперимента физики высоких энергий в *CERN* (Европейская организация для ядерных исследований, Женева, Швейцария) [10]. Эта система имеет несколько уровней организации (*Tier*). Суть распределенной модели состоит в том, что весь объем данных с детекторов БАК после обработки в реальном времени и первичной реконструкции должен направляться для дальнейшей обработки и анализа в региональные центры. Стоит отметить, что и экспериментальные данные распределены по нескольким хранилищам данных, которые имеют разное географическое положение. В России, в частности, существуют уровни (центры) этой большой распределенной инфраструктуры [11].

Системы, аналогичные вычислительной инфраструктуре БАК, необходимо создавать и развивать в России для существующих и планируемых экспериментов класса «мегасайенс». К таким российским экспериментам можно отнести:

- проект *NICA*, состоящий из нескольких экспериментов на ускорительном комплексе для изучения свойств плотной барионной материи (г. Дубна) [12];
- исследования на реакторе ПИК (г. Гатчина) [13];
- центр коллективного пользования «Сибирский кольцевой источник фотонов» (г. Новосибирск) [14];
- исследования в области нейтринной астрофизики высоких энергий с использованием нейтринного телескопа *Baikal-GVD* (оз. Байкал) [15].

Стоит обратить внимание и на эксперименты, реализуемые совместно с другими странами, например, *JUNO* (Китай) — планируемый прецизионный эксперимент с реакторными антинейтрино нового поколения, предназначенный для определения иерархии масс нейтрино [16].

РЦОД для таких важных и сложных экспериментов должны развиваться и масштабироваться, чтобы обеспечить качественную и эффективную работу ученых, которые стремятся к быстрому получению научных результатов и новым открытиям. В зависимости от области применения и типов решаемых задач требования к масштабированию могут быть разными, поэтому необходим инструмент, который позволит анализировать эффективность и надежность различных процессов, происходящих в РЦОД, с учетом потоков данных для хранения и обработки, а также улучшить стратегию управления потоками задач. Традиционные средства моделирования не позволяют качественно проводить такие исследования. Дальнейшие материалы данной главы изложены в соответствии с опубликованной статьей [17], где подробно рассмотрены преимущества и недостатки существующих инструментов моделирования РЦОД и предложены рекомендации по разработке альтернативного решения.

1.2. Состояние проблемы моделирования распределенных центров сбора, хранения и обработки данных

На этапе проектирования РЦОД важно, как можно точнее определить параметры всех процессов, которые будут происходить в рассматриваемой системе. Для этого создаются различные модели. В процессе моделирования можно определить минимально необходимое оборудование для функционирования системы, а также выбрать несколько вариантов оборудования с учетом текущих потребностей и перспективы развития в будущем. В проводимом исследовании рассматриваются библиотеки и комплексы программ, которые позволяют прежде всего проводить моделирование потоков данных РЦОД.

До настоящего времени были созданы различные инструменты для моделирования распределенных систем, например, *Bricks*, *OptorSim* и *GridSim* [18], которые имели узкую специализацию. С помощью этих программных пакетов можно было моделировать определенные архитектуры распределенных систем. Стоит отметить, что применение таких инструментов требует знания специальных языков программирования, что существенно снижает эффективность их использования [19]. Но тем не менее существуют программные комплексы для моделирования, построенные на основе перечисленных выше инструментов.

Так, например, программный инструмент *SyMSim* разработан с использованием пакета *GridSim*, который построен на библиотеке *SimJava*, позволяющей моделировать поток дискретных событий во времени [20]. Стоит отметить, что библиотеки *GridSim* (последняя версия 5.2 вышла в 2010 году [21]) и *SimJava* (последняя версия 2.0 вышла в 2002 году [22]) в настоящее время являются устаревшими, поэтому не могут быть использованы для моделирования современных РЦОД, т.к. не позволяют учесть новые тенденции построения распределенных систем, имеющих, например, гибридную структуру, т.е. включающих разнородные модули обработки данных, в том числе облачные структуры и суперкомпьютеры, а также системы иерархической памяти и многое другое.

Тем не менее, *SyMSim* предоставляет возможность моделирования процесса обработки потока заданий распределенной системы, обладающей заданными ресурсами и правилами их резервирования и использования. *SyMSim* представляет собой имитационную модель, которая позволяет получить величину времени обработки задач и оценить, как структура вычислительной установки и производительность отдельных ее частей влияют на это время. По результатам моделирования можно сделать вывод о наибольшем значении интенсивности потока задач (данных) [23].

Главная особенность программы *SyMSim* — синтез процессов моделирования и мониторинга. Данные мониторинга работы конкретного вычислительного центра хранятся в БД, а затем используются для формирования входного потока заданий для моделирования. В случае, если для конкретного вычислительного центра отсутствуют данные мониторинга, то анализируется статистика заданий по аналогичным вычислительным центрам [24]. Затем строятся распределения заданий по времени исполнения и размеру входного файла, после чего выдвигается и проверяется гипотеза о количестве типов заданий в потоке [25]. Сгенерированные потоки данных и задач независимо от их объема хранятся в БД, что является недостатком, т.к. может привести к серьезным вычислительным затратам при работе с БД.

Средство описания вычислительной инфраструктуры в программном комплексе реализовано как БД с интерфейсом, представленном в виде веб-страницы [26]. Описанию присваивается

идентификатор, который пользователь должен указать в параметрах запуска модели. Модель получает информацию из БД, по которой строится описание вычислительной структуры [27]. Следовательно, параметры оборудования и архитектура моделируемой инфраструктуры задается через БД, что является серьезным недостатком. Не все потенциальные пользователи программного комплекса имеют навыки работы с БД. Прямое взаимодействие неопытных пользователей с БД может привести к нарушению ее целостности.

Ядром программы является модуль, который отвечает за прием заданий на обработку и «запуск» их на процессорах, а также управляет очередями, сохраняя информацию о занятых и свободных в данный момент процессорах. При эксплуатации [28, 29, 30] инструментарий *SyMSim* специально адаптируется под каждую моделируемую инфраструктуру путем изменения программного кода. Этот факт является серьезным недостатком, который не позволяет применять *SyMSim* для решения широкого класса задач при проектировании и поддержке ЦОД.

Результатом работы программы моделирования служит последовательность записей в БД, отражающая все события, происходящие в системе. К ним относится, например, поступление задания, начало и конец обработки задания, начало и конец передачи файла, манипуляции с лентами и т. д. Все события описываются в едином формате. Результаты моделирования представлены в виде ограниченного количества графиков в веб-интерфейсе, передающие только часть информации [26]: количество поступающих данных, свободное место в хранилищах данных, нагрузка на каналы передачи данных, что может быть недостаточным для пользователей.

Таким образом, существующий программный комплекс *SyMSim* несмотря на главное достоинство, которое заключается в синтезе процессов моделирования и мониторинга, имеет ряд существенных недостатков:

1. библиотеки, которые лежат в основе комплекса являются устаревшими и не позволяют учесть современные тенденции построения РЦОД;
2. веб-интерфейс комплекса позволяет устанавливать только параметры потоков задач, запускать/останавливать процесс моделирования, а также просматривать некоторые результаты на ограниченном наборе графиков (для получения дополнительных результатов данные необходимо экспортировать из БД);
3. для изменения параметров оборудования и структуры моделируемой системы необходимо напрямую взаимодействовать с БД;
4. входной поток заданий и данных независимо от объема хранится в БД;
5. ядро программы адаптируется под каждую моделируемую инфраструктуру путем изменения программного кода;

6. не учитываются такие важные критерии функционирования системы, как потери данных в зависимости от типа выбранного оборудования, вероятность осуществления событий, связанных со сбоями и отказами оборудования, изменениями в производительности вычислительных ресурсов и хранилищ данных;

7. отсутствует возможность поиска конфигурации оборудования, удовлетворяющей заданным критериям.

Можно сделать вывод, что перечисленные недостатки не позволяют применять комплекс программ *SyMSim* для решения широкого класса задач при проектировании и поддержке РЦОД. Следовательно, необходимо создать иной подход, отличный от существующих средств и программных комплексов для моделирования, который можно будет применять при проектировании, построении и развитии РЦОД с учетом характеристик оборудования, а также потоков данных и задач.

1.3. Цифровой двойник: определение, примеры применения

Цифровой двойник — понятие, которое появилось совсем недавно. Первое упоминание о нем было в 2002 году во время презентации Майкла Гривса под названием “*Conceptual Ideal for PLM*”. Американский ученый, рассказывая об управлении жизненным циклом продукта (*PLM*), неофициально представил концепцию ЦД, которая основана на идее о том, что для каждой физической системы должна существовать некоторая виртуальная система. Эта виртуальная система является зеркальным отображением (дублированием, двойником) физической системы. Предполагается, что двойник содержит всю информацию о физической системе и связан с ним на протяжении всего жизненного цикла. При этом между виртуальной и физической системами должен существовать некоторый канал, обеспечивающий обмен данными и синхронизацию [1].

Жизненный цикл любой системы начинается с проектирования, когда анализируются цели создания системы, ее поведение, характеристики, т.е. формулируются некоторые требования. Вместе с тем определяются и нежелательные формы поведения системы для разработки стратегий предотвращения их появления. На этом этапе ЦД является прототипом будущей системы и используется для проверки, насколько проект соответствует сформулированным требованиям. Как только виртуальная копия конкретной системы со всеми компонентами завершена и проверена, начинается этап непосредственного создания физической системы, где ЦД поможет скорректировать производство или установку элементов системы, тем самым предотвращая возможные проблемы при вводе системы в эксплуатацию, сокращая финансовые расходы и время. На

этапе эксплуатации системы осуществляется верификация прогнозов ЦД относительно поведения системы при различных внешних факторах. При использовании системы, как правило, возникает необходимость в замене составных частей, появляются требования по улучшению ее характеристик, поведения, добавлению новых функциональных возможностей. Именно на этом этапе необходима связь между физической и виртуальной системами. Для прогнозирования поведения системы при изменении ее конфигураций, для оценки производительности и возможных сбоев используются ЦД [31].

Конечно, в первую очередь речь идет о сложных системах, состоящих из большого количества компонентов, между которыми существуют множественные связи. Проектирование, разработку и эксплуатацию таких систем усложняет не только их архитектура, но и нетривиальные процессы передачи и обработки большого количества данных, а также разнообразные технологии, на которых они основаны. Очевидно, что сложные системы должны безупречно выполнять свои задачи, всегда давать желаемые результаты. Риск выхода такой системы из строя должен быть минимальным. Но, к сожалению, в реальной жизни в процессе эксплуатации сложных систем возникают острые проблемы, требующие быстрого решения. Поэтому необходим инструмент, который способен предупреждать о вероятности возможных сбоев в системе, чтобы их можно было своевременно исправить. Одним из таких инструментов является моделирование, которое позволяет определить, как система будет работать в разных условиях, проверить и оценить последствия изменений внутри системы, найти причину возможных сбоев и т.д. Несмотря на то, что моделирование является предшественником ЦД, этот подход не обеспечивает обмен данными между физической системой и моделью в режиме реального времени.

Таким образом, ЦД могут быть определены как компьютерные модели, которые имитируют, отражают или дублируют физическую систему, а также следуют жизненному циклу своего физического двойника, отслеживая, контролируя и улучшая его процессы и функции. ЦД постоянно прогнозирует будущие состояния системы (например, дефекты, повреждения, сбои) и позволяет моделировать и тестировать новые конфигурации. Физическая система и ее ЦД должны постоянно обмениваться информацией. Данные, полученные, например, по средствам мониторинга самой системы и условий ее эксплуатации, позволяют применять вычислительные методы для прогнозирования сбоев, тестирования результатов возможных решений и др. Таким образом формируется подход прогнозного моделирования, при котором можно предсказать возможные сбои в системе, смоделировать любые модификации системы для предотвращения ошибок или поиска наилучшего решения. Однако, важно понимать, что ЦД не всегда полностью автономны и требуют большого вмешательства человека, особенно в случаях, когда они используются для тестирования новых функций и модификаций физических систем [32].

В настоящее время достаточно много примеров применения ЦД во всем мире. ЦД разрабатывают для разных целей. Например, в транспортном машиностроении для создания и эксплуатации автомобилей [33], в аэрокосмической отрасли для профилактического обслуживания [34] и при создании внутренних систем для авиации [35], в судостроении и эксплуатации водного транспорта для проведения виртуальных испытаний [36], в железнодорожном транспорте для строения путей [37], в нефтегазовой отрасли [38] и электроэнергетике [39], в сельском хозяйстве для повышения урожайности и оптимизации процессов ухода за животными [40]. В медицине очень широкое распространение получили ЦД органов [41], важную роль для здравоохранения играют ЦД медицинских компонентов [42]. Интересным примером применения виртуальных двойников является ЦД города. Такие системы применяются как для зарубежных [43], так и для российских городов [44]. Существуют также примеры применения ЦД в логистике [45], для оптимизации жизненных циклов производства [46], больниц [47] и других организаций.

Выше представлена лишь небольшая часть от большого множества примеров применения ЦД во всех областях человеческой деятельности. В данной работе исследования сосредоточены на изучении новых тенденций в области разработки и применения ЦД, а именно для проектирования, построения и развития центров хранения и обработки данных, в частности распределенных, с учетом характеристик потоков данных и задач.

1.4. Возможности использования цифровых двойников центров хранения и обработки данных

В современном мире в любой области деятельности человека (науке, технике, производстве, социальной жизни и т.п.) прослеживается необходимость в хранении и обработке больших объемов данных. В связи с этим повсеместно начали использовать ЦОД. Например, в науке ЦОД применяют для хранения и дальнейшей обработки данных проводимых экспериментов; в бизнесе – для предоставления услуг ИТ-аутсорсинга; в государственных организациях – для автоматизации внутренних процессов, таких как, документооборот; в системах безопасности городской среды, транспорта и промышленных предприятий – для обеспечения сбора, хранения и анализа систем контроля доступа, видеонаблюдения и т.п. Для хранения и интенсивной обработки данных существует множество вычислительных систем различного масштаба. Для проектирования, создания и модификаций таких сложных систем нельзя обойтись без ЦД. Рассмотрим несколько примеров применения ЦД центров хранения и обработки данных.

Американская компания *Future Facilities* [6] занимается разработкой программного обеспечения и созданием ЦД для управления ЦОД. ЦД реализован, как 3D-копия ЦОД, которая может

имитировать его физическое поведение при любом сценарии работы. ЦД охватывает всю экосистему ЦОД, включая виртуальные представления компонентов питания, охлаждения и т.д. Такие ЦД основаны на моделировании инженерных систем и позволяют прогнозировать влияние изменений в компонентах ЦОД путем визуализации и количественной оценки производительности [48]. Рассматриваемые ЦД используются для проектирования объектов нового поколения, устранения неполадок на существующих объектах, ускорения процессов планирования мощностей и повышения общей эффективности и устойчивости.

Еще одна американская компания *Sunbird DCIM* [7] специализируется на создании 3D-копии физического ЦОД, которая отражает фактические условия работы ЦОД в режиме реального времени и предоставляет возможности удаленной визуализации и мониторинга, тем самым обеспечивая более быстрое управление ЦОД. Этот ЦД содержит всю наиболее важную информацию о физической инфраструктуре ЦОД. В виде высококачественных масштабируемых 3D-изображений в ЦД представлены сервера, сетевое оборудование, системы хранения данных, кабели. О каждом из компонент инфраструктуры в ЦД содержится информация о точном расположении, мерке, модели, размерах, конфигурации, портах и многое другое. В такой модели можно визуализировать даже те порты и кабели, которые трудно увидеть физически, например, в кабельных лотках. Помимо этого, в ЦД отображаются текущие показания датчиков окружающей среды, таких как температуры, влажности, воды, воздушного потока и перепада давления, что обеспечивает точное представление о том, что происходит в ЦОД. Такой функционал ЦД позволяет увеличить время безотказной работы за счет своевременного информирования о возможных проблемах и нагрузках на компоненты ЦОД, повысить эффективность ЦОД с помощью функции анализа текущего потребления ресурсов и оценки влияния допустимых изменений, удаленно повысить производительность и не затрачивать время и деньги на физическое присутствие в ЦОД. Благодаря автоматическому сбору, хранению и представлению отчетов можно получать и анализировать данные о работе ЦОД в режиме реального времени.

В описанных выше ЦД присутствуют существенные недостатки:

- представлены виртуальные копии ЦОД, которые территориально расположены в одном физическом пространстве;
- ЦОД рассматриваются только с точки зрения инженерной инфраструктуры.

Однако, во-первых, в последнее время чаще используются распределенные ЦОД, а во-вторых, существует необходимость моделирования потоков данных для формирования стратегии управления потоками задач, анализа количества используемых ресурсов с целью своевременного обновления оборудования, предотвращения ситуации, когда имеющегося оборудования будет

недостаточно для хранения всех данных или быстрой обработки в требуемый промежуток времени, оценки количества ресурсов, необходимых для использования ЦОД в тех или иных задачах, согласно предъявленным требованиям к ЦОД.

Таким образом, представляется актуальной разработка метода и алгоритмов построения цифровых двойников РЦОД с целью проведения исследований в области эффективности и надежности функционирования распределенных систем, проверки различных сценариев масштабирования, учитывая требования к потокам данных и задач, формирования и улучшения стратегии управления потоками задач, анализа используемых ресурсов, оценки необходимого количества ресурсов для конкретных задач согласно предъявленным требованиям к РЦОД.

1.5. Выводы к главе 1

В данной главе уточнено понятие ЦД для целей данного исследования. В качестве ЦД подразумевается компьютерная модель, которая:

- отражает архитектуру ЦОД, а также процессы, происходящие в рассматриваемой системе и связанные с потоками данных и задач;
- позволяет моделировать и тестировать новые конфигурации оборудования, тем самым предоставляя возможность прогнозировать сбои в системе и оценивать результаты решений для предотвращения подобных ситуаций.

В результате проведенного анализа существующих ЦД и средств моделирования ЦОД можно сделать вывод, что на данный момент ни в России, ни за рубежом не существует инструмента, позволяющего построить цифровой двойник РЦОД с целью формирования и улучшения стратегии управления потоками задач, анализа используемых ресурсов, оценки необходимого количества ресурсов для конкретных задач согласно предъявленным требованиям к ЦОД. Таким образом, для проведения исследований эффективности и надежности РЦОД как при проектировании, так и в процессе функционирования, управлять РЦОД, проверять различные сценарии масштабирования, учитывая требования к потокам данных и задач, существует актуальная задача разработки метода и алгоритмов построения цифровых двойников РЦОД.

Глава 2. Модели, методы и алгоритмы построения цифровых двойников распределенных центров сбора, хранения и обработки данных

2.1. Предварительные замечания

Процесс построения ЦД, как правило, реализуется поэтапно. На первом этапе осуществляется сбор данных о назначении физического объекта (системы), его составных частях и параметрах функционирования [49]. На втором этапе из полученных данных извлекается информация, необходимая для построения виртуального образа рассматриваемого объекта (системы) [50]. При построении непосредственно виртуального образа на третьем этапе используются методы моделирования. Четвертый этап, как правило, посвящен проверке получившихся моделей, верификации и валидации с физическим прототипом [51]. Это очень важный этап, так как от адекватности построенной модели зависит дальнейшее изучение рассматриваемого физического объекта (системы). При необходимости проводится корректировка модели. Последний, пятый этап построения ЦД предусматривает создание модуля демонстрации результатов работы ЦД, которые должны быть открытыми, доступными и представлены в понятном виде (диаграммы, графики, схемы и т.д.) [52]. Таким образом, основой ЦД являются модели, а непосредственное функционирование ЦД обеспечивается инструментами, которые создаются при помощи различных информационных технологий.

Естественно, эффективность создания, использования и развития технологии ЦД обуславливается качеством ядра (модели), для построения которого используются различные подходы. Существует несколько классов моделей, которые в основном применяются для создания ЦД, а именно: физическая модель, оптимизационная модель, имитационная модель [44]. Физическая модель обеспечивает компьютерное моделирование физических процессов, протекающих во времени. В основе такой модели лежат законы физики и компьютерный инженерный анализ [53]. Оптимизационная модель обеспечивает поиск оптимальных значений целевой функции при наличии ограничений с использованием математических методов. Имитационное моделирование представляет собой метод исследования, при котором изучаемая система заменяется компьютерной моделью, в которой имитируются все или некоторые процессы, происходящие в реальной системе. Для того, чтобы ЦД справлялся с поставленными перед ним задачами, при разработке ядра необходимо учитывать вероятностные характеристики процессов, происходящих в моделируемой системе [54].

В то же время важную роль играют и технологии, с помощью которых обеспечивается взаимодействие пользователя с ЦД. ЦД должен обладать функционалом, который позволит задавать, отображать и редактировать исходные данные, а также визуализировать и обрабатывать результаты моделирования. Для этого необходимо разработать методологию и соответствующий программно-технический инструментарий. Помимо этого, очень важно предусмотреть критерии оценки качества работы ЦД, которые будут использоваться в процессе создания и модернизации ЦД по результатам эксплуатации.

Таким образом, необходимо описать функциональные и нефункциональные требования к создаваемому цифровому двойнику РЦОД.

2.2. Требования к цифровым двойникам распределенных центров сбора, хранения и обработки данных

2.2.1 Требования к ядру

В рамках рассматриваемой задачи по созданию ЦД, который поможет качественно проводить исследования, связанные с созданием и развитием современных РЦОД с учетом характеристик потоков данных и задач, целесообразно разработать уникальное ядро. Уникальность ядра заключается в его универсальности и применимости для моделирования процессов передачи, хранения и обработки данных любых РЦОД. При построении ядра ЦД, в котором будет реализован процесс моделирования РЦОД, необходимо учитывать современные тенденции построения распределенных систем и придерживаться принципа разработки универсального программного пакета, который не нужно изменять под каждую моделируемую инфраструктуру. При моделировании необходимо также учитывать возможные вероятности изменения процессов функционирования системы в зависимости от типа оборудования (например, потери данных, изменение производительности). Эта особенность позволит применять ЦД для решения широкого класса задач, проводить исследования, связанные с проектированием, масштабированием, повышением производительности РЦОД, анализом эффективности и надежности различных процессов, происходящих в РЦОД в период эксплуатации.

В ЦД должна быть предусмотрена функциональная возможность, позволяющая выбирать несколько потенциально конкурентоспособных вариантов из множества допустимых конфигураций оборудования для РЦОД.

2.2.2 Требования к модулю для взаимодействия пользователя с цифровым двойником

Конечно, цифровые двойники РЦОД не могут быть полностью автономным и требуют вмешательства человека, например, для описания моделируемой инфраструктуры, при тестировании

различных модификаций распределенной системы, при поиске конфигураций оборудования согласно имеющимся требованиям. Во время проведения экспериментов и работы ЦД структура моделируемой распределенной системы может быть изменена, например, может потребоваться добавление или удаление вычислительных компонент, хранилищ данных, линий связи между компонентами РЦОД. Это значит, что разрабатываемый ЦД должен быть оснащен хорошо продуманным, удобным и доступным интерфейсом, где пользователь сможет не только задавать требуемые параметры РЦОД, но и получать результаты работы ЦД в понятном и структурированном виде, в том числе на графиках.

Таким образом, можно сформулировать следующие функциональные требования к модулю для взаимодействия пользователей с ЦД:

- взаимодействие должно осуществляться в диалоговом режиме;
- пользователь должен строить инфраструктуру РЦОД путем расположения на поле необходимых объектов, перемещения их в нужную область, соединения между собой каналами связи;
- система должна позволять сохранить изображение с инфраструктурой РЦОД на компьютер пользователя;
- модуль должен предоставлять возможность задавать параметры базовой конфигурации оборудования РЦОД;
- модуль должен предоставлять возможность задавать характеристики потоков данных для обработки на РЦОД;
- модуль должен предоставлять возможность задавать характеристики потоков задач для выполнения на РЦОД;
- пользователь должен иметь возможность редактировать инфраструктуру РЦОД, параметры оборудования, характеристики потоков данных и задач;
- модуль должен содержать функционал для запуска ЦД;
- пользователь должен создавать вычислительные эксперименты на ЦД, которые подразумевают поиск и выбор требуемой конфигурации оборудования для решения конкретных задач (например, поиск необходимого количества ресурсов для хранения данных, поступающих с экспериментальной установки; поиск необходимо количества вычислительных ресурсов для обработки данных за определенный промежуток времени и др.);
- должна быть предусмотрена возможность формировать сценарии масштабирования/модификации параметров РЦОД;
- результаты работы ЦД должны быть графически проиллюстрированы и отражать объемы данных и распределение различных типов файлов в хранилищах, использование вычислительных компонент, нагрузку на каналы связи во время передачи данных.

Диаграмма вариантов использования описанного модуля представлена на рисунке 1.

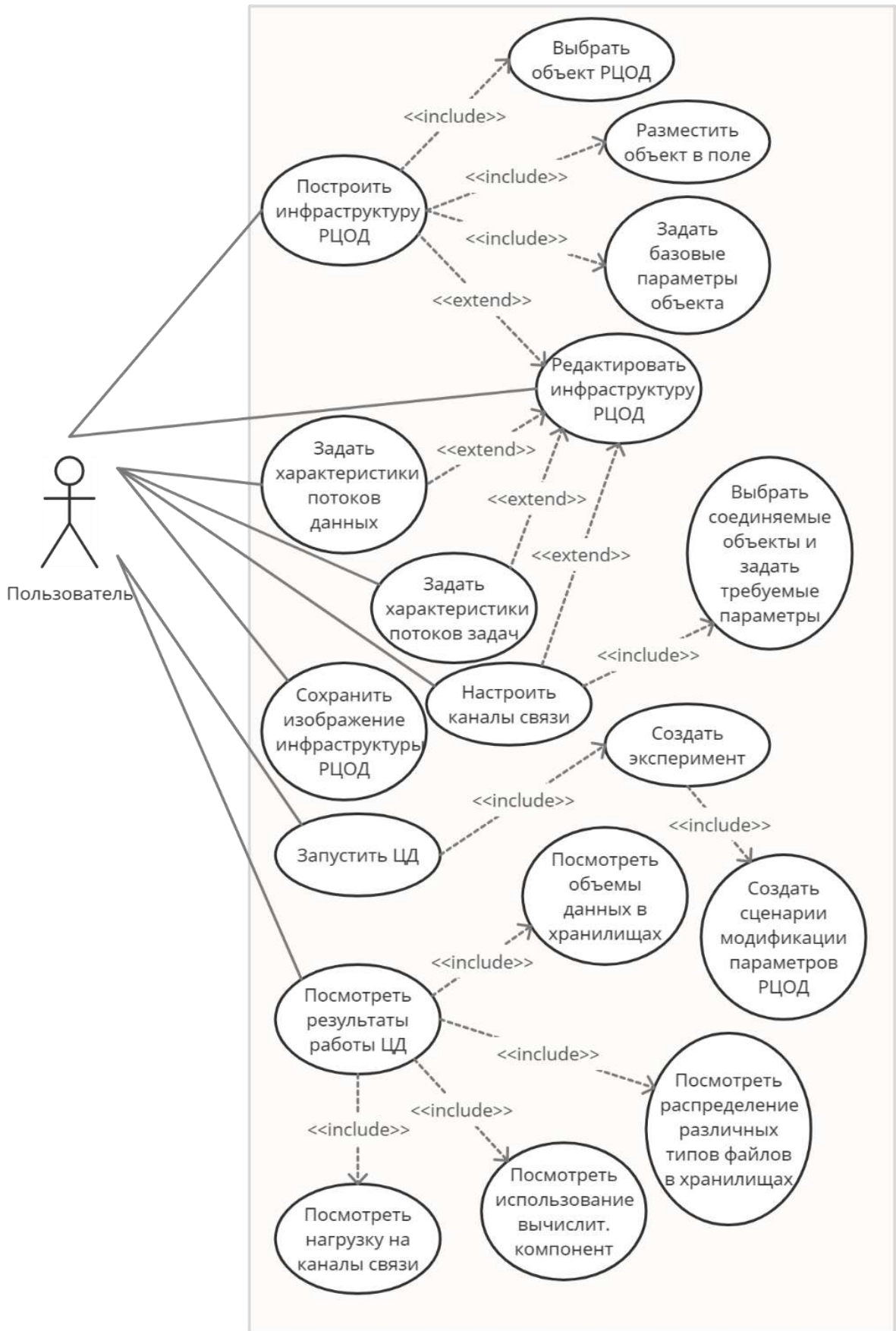


Рисунок 1. Диаграмма вариантов использования модуля для взаимодействия пользователя с ЦД

2.2.3 Нефункциональные требования к методу построения цифровых двойников РЦОД

Для метода построения цифровых двойников РЦОД в целом можно отнести следующие нефункциональные требования надежности, безопасности и производительности:

- различные параметры и результаты работы ЦД должны храниться в БД;
- скорость работы ЦД от момента запуска до получения результатов должна определяться исходя из масштаба РЦОД и процессов, происходящих в нем;
- адекватность ЦД должна быть проверена результатами работы существующего РЦОД по разным параметрам, доступным при мониторинге системы; результаты работы ЦД не должны отклоняться от среднего значения данных мониторинга более, чем на три среднеквадратических отклонения;
- работоспособность системы не должна зависеть от стороннего программного обеспечения.

2.3. Метод построения цифровых двойников распределенных центров сбора, хранения и обработки данных

Исходя из выявленных функциональных требований можно сформулировать метод построения цифровых двойников РЦОД. Метод состоит из следующих этапов.

1. Получение данных о РЦОД: параметрах оборудования, потоках данных и задач, целях построения ЦД, вероятностных характеристиках процессов, происходящих в РЦОД.
2. Описание структуры РЦОД и связей между компонентами.
3. Формирование виртуального образа РЦОД на основе полученных данных и описания его структуры.
4. Моделирование РЦОД.
5. Графическое представление результатов работы ЦД.

Основываясь на принципах системного анализа [55], в общем виде систему для реализации метода можно представить следующим образом:

$$MDT\{D, S, V, M, G, R, Q, W\}, \quad (1)$$

где D — физические объекты РЦОД, S — сущности модели данных, V — объекты виртуального образа РЦОД, M — объекты моделирования процессов в РЦОД (ядро ЦД), G — экземпляры модуля для взаимодействия пользователя с ЦД, R — отношения, определяющие связи между физическими объектами и сущностями модели данных, Q — отношения, определяющие связи между сущностями модели данных и объектами виртуального образа РЦОД, W — отношения, определяющие связи между результатами работы ЦД и их графическими представлениями.

У каждого физического объекта РЦОД ($d \in D$) существует набор параметров, которые характеризуют составляющие распределенных центров X , т.е.: $d \in D \subset X_1 \times X_2 \times \dots \times X_N$, где X_i — свойства физических объектов при $i = \overline{1, N}$ ($N \in \mathbb{N}$). Физические объекты способны переходить из одного состояния в другое под воздействием внешних факторов или процессов, происходящих в РЦОД, поэтому физические объекты формализованы в следующем виде:

$$D = \{X, \Phi, f\}, \quad (2)$$

где Φ — множество допустимых состояний объекта, f — функция перехода из одного состояния в другое, т.е. $f: \Phi \times X \rightarrow \Phi$.

Для формального описания объектов РЦОД необходима модель данных, где каждому физическому объекту ставится в соответствие определенная сущность ($s \in S$), т.е. $D \xrightarrow{R} S$, где

$$R: \{s \in S | \exists d \in D, s = R(d)\}. \quad (3)$$

Связи между компонентами РЦОД описываются в модели данных, как $P: \{s_i P_j s_k\}$, где $i, j, k = \overline{1, N}$ ($N \in \mathbb{N}$). Следовательно, формальное описание модели данных выглядит следующим образом:

$$S = \{D, R, P\}. \quad (4)$$

Виртуальный образ РЦОД должен формироваться на основе параметров реальных физических объектов и связей между ними (P), что отражено в модели данных, т.е. $S \xrightarrow{Q} V$, где

$$Q: \{v \in V | \exists s \in S, v = Q(s)\}. \quad (5)$$

Таким образом, виртуальный образ РЦОД формализован, как

$$V = \{S, Q, P\}. \quad (6)$$

Главным этапом метода построения цифровых двойников РЦОД является процесс моделирования, для реализации которого должны быть созданы объекты (генераторы данных и задач, планировщик перемещения данных, объект управления задачами обработки данных и др.), в совокупности являющиеся ядром ЦД ($m \in M$). Исходными данными этого этапа являются параметры виртуальных объектов РЦОД ($v \in V$), у каждого из которых есть множество допустимых состояний (U). Каждый виртуальный объект, подобно своему физическому образу, имеет функцию перехода для изменения состояния ($h: U \times V \rightarrow U$), которые должны учитываться при моделировании различных процессов, происходящих в РЦОД. Выходными данными рассматриваемого этапа являются формальные описания состояний объектов ($z \in Z \subset Y_1 \times Y_2 \times \dots \times Y_N$, где Y_i — свойства объектов при $i = \overline{1, N}$ ($N \in \mathbb{N}$)) в каждый момент времени (T). При этом существует функция выхода $k: U \times V \rightarrow Z$. Выход возможен в случае завершения заданного времени работы, при окончании процесса выполнения потока задач или при переполнении ресурсов, которые приводят к остановке работы РЦОД. Следовательно, формально ядро ЦД можно представить в виде:

$$M = \{T, V, U, Z, h, k\}. \quad (7)$$

Стоит отметить, для генерации таких параметров объектов, как объем данных и время выполнения задач, можно использовать равномерное (8) или нормальное (9) распределение непрерывной случайной величины.

$$pr(x) = \frac{1}{b-a}, \quad (8)$$

где $pr(x)$ — функция плотности вероятности равномерного распределения, $[a, b)$ — интервал допустимого изменения значения рассматриваемого параметра.

$$pn(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (9)$$

где $pn(x)$ — функция плотности вероятности нормального распределения, μ — среднее значение случайной величины, σ — стандартное отклонение.

К событиям изменения состояния объектов РЦОД можно отнести, например, переход оборудования в нерабочее состояние, изменение производительности и т.д. События можно описать в терминах распределения вероятностей, т.к. время их наступления является случайной величиной. Поток событий формируется согласно закону Пуассона, согласно которому распределены моменты наступления событий. Промежутки между наступлениями событий определяются по формуле экспоненциального распределения:

$$\tau = -\frac{1}{\lambda} * \ln(r), \quad (10)$$

где τ — интервал между случайными событиями, λ — среднее число событий в единицу времени, $r \in [0; 1]$ — равномерно распределенное случайное число.

Характеристики потоков данных и задач, параметры случайных событий изменения состояния объектов, типы распределения вероятностей определяет пользователь, который является экспертом в области администрирования РЦОД. Оценивая надежность системы и вероятность отказа по данным различных систем мониторинга, пользователь может изменять настройки параметров ЦД с целью получения информации о времени обработки данных, о загруженности ресурсов и т.д. для принятия решения о стратегии управления потоками задач или возможном перераспределении ресурсов.

Для взаимодействия пользователя с ЦД в диалоговом режиме должны быть предусмотрены соответствующие интерфейсы ($g \in G$), в частности для задания различных параметров и графического представления результатов работы ЦД, которые формально можно представить в виде:

$$G = \{Z, W\}, \quad (11)$$

где Z — формальные описания состояний объектов, информация о которых сохранялась в процессе работы ЦД, W — отношения, определяющие связи между результатами работы ЦД и их дальнейшим представлениями ($Z \xrightarrow{W} G$), а именно:

$$W: \{g \in G | \exists z \in Z, g = W(z)\}. \quad (12)$$

По результатам работы ЦД эксперт может принять решение по выбору требуемой конфигурации оборудования из некоторого множества альтернатив $\{\psi\}$, основываясь на принципе выбора Ω . Т.е. задачу принятия решения можно формализовать в виде:

$$\{\{\psi\}, \Omega\} \rightarrow \psi^*, \quad (13)$$

где ψ^* — выбранная альтернатива. Принцип выбора зависит от цели, которой может быть поиск конфигурации оборудования и стратегии управления потоком задач, обеспечивающей обработку всех данных за минимальное количество времени, с использованием наименьшего количества ресурсов и др. Значит, необходимо найти такую альтернативу, которая доставляет минимум одному, наиболее предпочтительному критерию (например, времени), при условии, что значения остальных критериев будут не более некоторых заданных заранее величин:

$$criteria_1(\psi) \rightarrow \min; criteria_j(\psi) \leq c_j; j = \overline{2, N} (N \in \mathbb{N}), \quad (14)$$

где $criteria_1(\psi)$ — наиболее предпочтительный критерий, $criteria_j(\psi)$ — множество критериев, c_j — заданное допустимое значение j -го критерия.

Выбранная альтернатива должна обеспечить более эффективное использование ресурсов и надежные сценарии масштабирования и управления потоками данных и задач РЦОД. В общем виде показатель эффективности имеет вид $\alpha = \langle H_{ц}, T \rangle$, где $H_{ц}$ — целевые эффекты, T — затраты времени.

Для реализации каждого этапа метода необходимо разработать алгоритмы, а также модель данных, где будут храниться исходные данные о РЦОД и результаты работы ЦД. Дальнейшее изложение материала представлено в соответствии с опубликованной статьей [56].

2.4. Алгоритмы для реализации метода построения цифровых двойников распределенного центра сбора, хранения и обработки данных

2.4.1 Алгоритм работы ядра

Ключевым элементом в методе построения цифровых двойников РЦОД является ядро, или программа моделирования распределенных систем, учитывающая параметры потоков данных и задач для обработки, а также вероятностные характеристики процессов, происходящих в РЦОД. Поэтому прежде всего необходимо описать алгоритм работы программы моделирования (см. рис. 2).

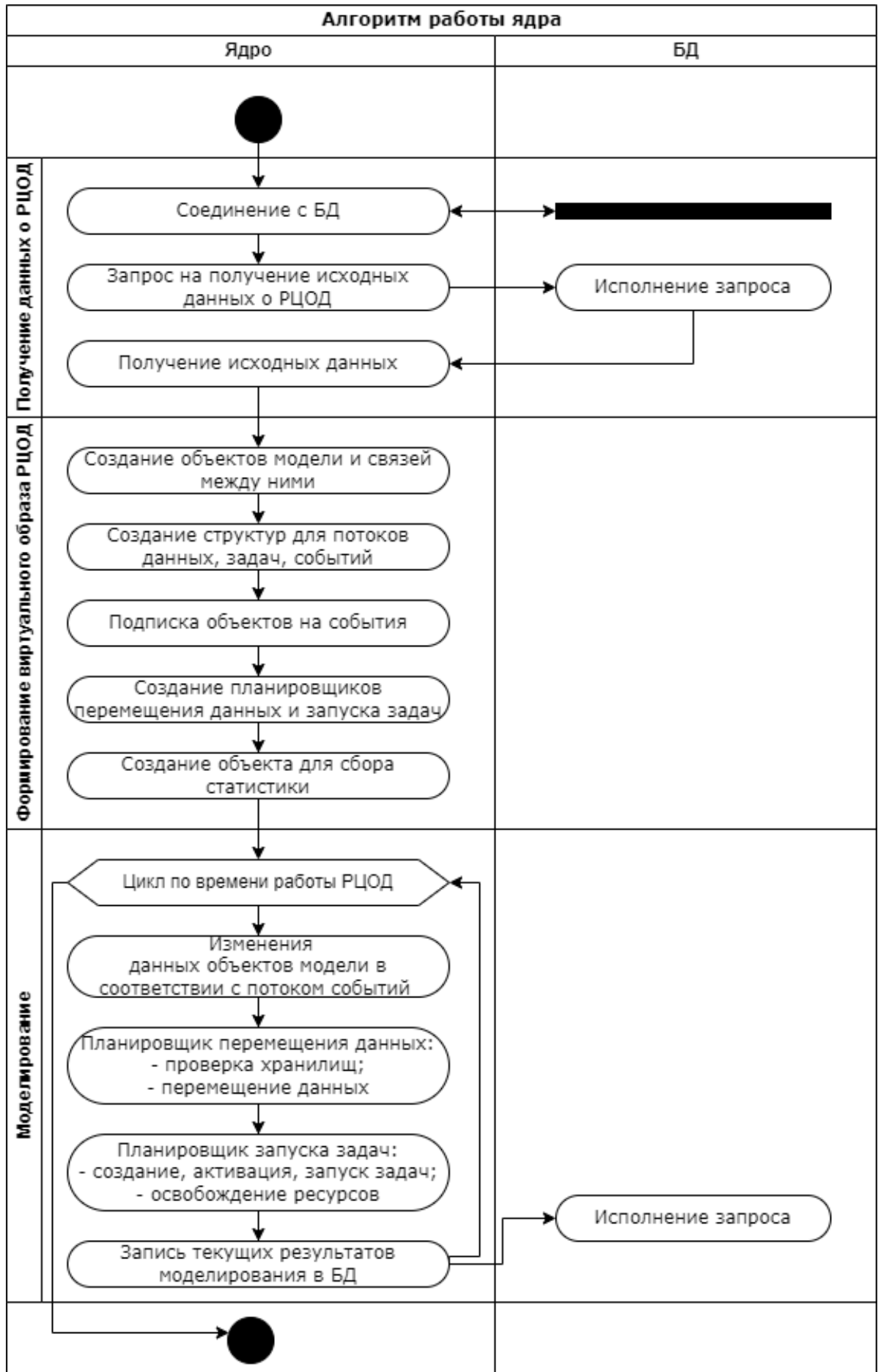


Рисунок 2. Алгоритм работы ядра цифрового двойника РЦОД

Для того, чтобы ядро было универсальным и могло применяться для моделирования процессов передачи, хранения и обработки данных любых РЦОД, необходимо придерживаться принципа разработки универсального программного пакета, который не нужно изменять под каждую моделируемую инфраструктуру. В связи с этим все входные параметры, которые описывают структуру РЦОД, конфигурации входящего в его состав оборудования, характеристики потоков данных и задач, должны храниться в БД. Помимо этого, в БД должна содержаться информация о возможных событиях, происходящих в РЦОД. Под событием подразумевается, например, плановая остановка оборудования, его временное прекращение работы, изменение производительности и т.д. События не привязываются к конкретному объекту, т.к. могут произойти с любым элементом РЦОД. Событие может произойти независимо от объекта, например, при остановке всего РЦОД, а может быть инициировано процессами объекта, например, переполнение хранилища данных, которое наступает, когда записываются данные по объему превышающие объем имеющихся ресурсов. В БД должны быть описаны сценарии моделирования и модификации параметров оборудования РЦОД, а также критерии сбора статистики о конкретных объектах в процессе работы будущего ЦД. Таким образом, первым этапом алгоритма является настройка соединения с БД и получение всей необходимой информации о моделируемой системе.

Второй этап алгоритма — формирование виртуального образа рассматриваемого РЦОД. На этом этапе по исходным данным, полученным из БД, осуществляется создание объектов будущей модели и связей между ними. Далее определяются структуры для потоков данных и задач, вырабатывается поток событий. В связи с тем, что события могут произойти с любым объектом, целесообразно ввести понятие «подписка объекта на событие», благодаря которому можно управлять одновременно всеми элементами РЦОД в момент наступления того или иного события. Таким образом, объекты подписываются на любое количество событий. На этом же этапе создаются объекты, которые должны отвечать за планирование процессов перемещения данных и задач между элементами РЦОД, а также объекты, которые будут собирать и записывать в БД статистическую информацию о процессах, происходящих в различных элементах РЦОД на протяжении третьего этапа алгоритма — моделирования.

Моделирование проводится по постоянным временным единицам, минимальное значение которых составляет 1 секунда. Для каждого построенного на предыдущем этапе алгоритма объекта модели в соответствии с потоком событий производятся изменения в данных, которые соответствуют временной единице работы РЦОД. Начало и продолжительность каждого события определяется согласно одному из распределений вероятности (например, Пуассона). Планировщики перемещения данных и управления задачами являются ключевыми объектами, которые описывают процессы, происходящие в РЦОД. В каждую единицу времени планировщик переме-

щения данных проверяет все хранилища на наличие новых данных, которые генерируются согласно выбранному закону распределения вероятностей. В случае их обнаружения запускается процесс передачи данных из текущего хранилища в пункты назначения согласно описанию структуры моделируемого РЦОД. Задача второго планировщика заключается в управлении потоком задач для обработки имеющихся данных. В каждую единицу времени планировщик создает некоторое количество задач, время выполнения которых является случайной величиной, распределенной согласно одному из законов (например, равномерно), активирует и отправляет на исполнение ранее созданные задачи. Существует две основные модели запуска задач на вычислительных ресурсах. Первая модель представляет собой отправку задач в очереди, которые расположены непосредственно на счетных узлах вычислительных компонент. Во второй модели существуют общие очереди для каждого класса задач, не закрепленные под конкретные вычислительные ресурсы. Поэтому на вычислительных компонентах имеются индивидуальные пилоты, которые контролируют процесс запуска задач. Под пилотом понимается алгоритм, который анализирует количество свободных ядер на вычислительной компоненте, выделенных под определенный тип задач. Если доступны свободные ресурсы, то пилот берет задачу из соответствующей очереди и отправляет ее на исполнение. Далее задача удаляется из очереди, занимает вычислительные ресурсы (слоты) и начинает исполняться, т.е. обрабатывать входной файл при его наличии в хранилище. Каждый слот включает в себя некоторое подмножество ядер. Для выполнения задачи может потребоваться как одно, так и несколько ядер. В результате выполнения задачи выходной файл записывается в хранилище, с которым связан вычислительный ресурс. После завершения работы каждой задачи планировщик освобождает вычислительные ресурсы. На протяжении всего этапа моделирования осуществляется запись данных в БД, где отражена статистическая информация о всех происходящих процессах.

2.4.2 Алгоритм работы модуля для взаимодействия пользователя с цифровым двойником

Алгоритм работы модуля для взаимодействия пользователя с ЦД представлен на диаграмме на рисунке 3. Взаимодействие пользователя с ЦД начинается с построения инфраструктуры РЦОД. Согласно выдвинутым требованиям, пользователю необходимо выбрать из списка объект, соответствующий оборудованию РЦОД, расположить его в некоторой области и настроить базовую конфигурацию этого устройства. После этого вся информация о добавленном в инфраструктуру объекте сохраняется в БД. После того, как все необходимые объекты будут добавлены, пользователь должен настроить каналы связи между ними. Под каналами связи понимаются логические соединения между компонентами РЦОД, по которым передаются данные. Для настройки каналов следует выбрать объекты, между которыми осуществляется соединение, и задать требуемые параметры. Информация о каналах связи также добавляется в БД. Далее требуется создать потоки данных для обработки на РЦОД и потоки задач, которые будут обрабатывать эти данные.

Пользователь может редактировать инфраструктуру РЦОД, базовую конфигурацию оборудования, изменять параметры потоков данных и задач до перехода на следующий этап, который заключается непосредственно в создании ЦД.

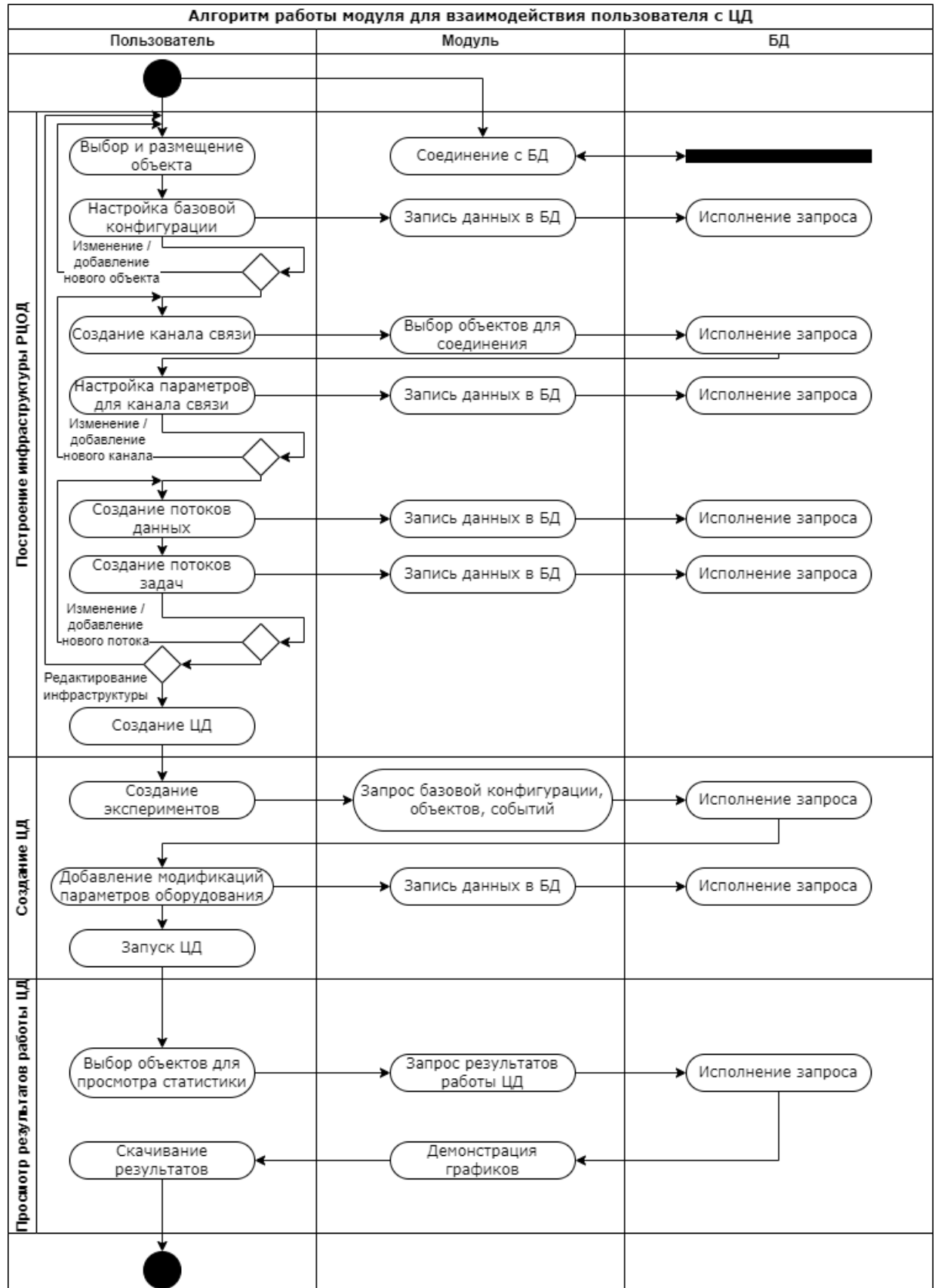


Рисунок 3. Алгоритм работы модуля для взаимодействия пользователя с цифровым двойником РЦОД

ЦД могут создаваться для разных задач, например, для проектирования РЦОД; для проверки работоспособности существующего РЦОД с текущими параметрами оборудования и поиска проблемных мест его функционирования; для масштабирования РЦОД и поиска конфигурации оборудования, которая будет удовлетворять некоторым требованиям, и т.д. В связи с этим пользователю необходимо создать вычислительные эксперименты, где ЦД будет применяться для решения конкретной задачи. При добавлении вычислительного эксперимента на ЦД важно указать объекты и события, информация о которых будет сохраняться в БД. В каждом таком эксперименте будет доступ к базовой конфигурации оборудования, которая создавалась при построении инфраструктуры РЦОД. При этом пользователь может добавить дополнительные модификации параметров оборудования, если это необходимо для решения поставленной задачи. После настройки вычислительных экспериментов осуществляется запуск ЦД. Если существует несколько модификаций, то ЦД для каждой модификации могут быть запущены одновременно.

В процессе или по завершении работы ЦД пользователю доступны результаты для просмотра и дальнейшего анализа с целью оценки эффективности решения задач построения и развития РЦОД. Пользователю необходимо выбрать тип интересующего оборудования, после чего будут построены интерактивные графики по данным из БД, которые записывались в процессе работы ЦД. Например, графики могут отражать объемы данных и распределение различных типов файлов в хранилищах, использование вычислительных компонент, нагрузку на каналы связи во время передачи данных и т.д. Интерактивные графики подразумевают возможность масштабирования и выбора данных для просмотра. Пользователь может сохранить результаты работы ЦД для какой-либо модификации в виде изображений с графиками, которые отражают изменения, происходящие в РЦОД.

2.5. Модель данных

БД играет ключевую роль при разработке алгоритмов для реализации метода построения цифровых двойников РЦОД. В БД должна храниться информация об архитектуре РЦОД, параметрах оборудования, входящего в его состав, характеристиках потоков данных и задач, событиях, происходящих в РЦОД, сценариях возможного масштабирования системы, результатах работы ЦД. В связи с этим необходимо разработать модель данных, которая должна включать следующие сущности:

- *DataStorages* — хранилища данных РЦОД;
- *ComputingComponents* — вычислительные компоненты РЦОД;

- *Pilots* — пилоты, т.е. объекты, которые будут осуществлять запуск задач на вычислительных ресурсах РЦОД;
- *Slots* — слоты, которые входят в состав вычислительных компонент РЦОД;
- *Links* — каналы связей между компонентами РЦОД;
- *DataTags* — типы данных, которые хранятся и обрабатываются в РЦОД;
- *DataFlows* — потоки данных для хранения и обработки в РЦОД;
- *JobQueues* — потоки задач, которые будут обрабатывать имеющиеся в РЦОД данные;
- *TransportJobs* — объекты передачи потоков данных между хранилищами РЦОД;
- *Events* — возможные события, происходящие в РЦОД;
- *Event_Object* — параметры распределений вероятностей возникновения событий для каждого объекта, с которым может произойти то или иное событие;
- *Sensors* — объекты для сбора статистики в процессе работы ЦД;
- *Experiments* — варианты запуска ЦД для различных задач (вычислительные эксперименты);
- *Modifications* — модификации (сценарии масштабирования) базовой конфигурации оборудования РЦОД;
- *SimulationReport* — результаты работы ЦД.

Атрибутами сущности *DataStorages* являются параметры хранилищ данных РЦОД: идентификатор, название, описание, объем, название объекта для сбора статистики. Для источника генерации данных, т.е. активного хранилища, необходимы атрибуты, где будет указываться факт активности и скорость появления данных. Для каждого хранилища также должно быть установлено значение приоритета, которое указывает на последовательность просмотра устройств в ходе планирования процессов передачи данных.

Атрибутами сущности *ComputingComponents* являются параметры вычислительных компонент РЦОД: идентификатор, название, описание, количество вычислительных ресурсов (ядер), название объекта для сбора статистики. Если на вычислительной компоненте предусмотрена возможность уменьшения времени, затрачиваемого на выполнение задачи, необходим атрибут, который будет содержать значение коэффициента ускорения.

Атрибутами сущности *Pilots* являются параметры пилотов, которые будут осуществлять запуск задач на вычислительных ресурсах РЦОД: идентификатор, название, описание, название объекта для сбора статистики. Для идентификации ресурса, на котором будет работать пилот, необходим атрибут, в котором указывается название вычислительной компоненты. Согласно алгоритму работы ядра, для каждого класса задач на вычислительных компонентах имеются индивидуальные пилоты, поэтому рассматриваемая сущность должна содержать атрибут, в котором

указывается название потока задач. У каждого пилота должно быть установлено значение приоритета, которое указывает на последовательность просмотра объектов в ходе планирования процессов запуска задач обработки данных. Для разделения потоков задач на несколько вычислительных компонент, введен атрибут, где будет указываться доля задач из общего потока, которые должен забирать пилот из очереди. В связи с тем, что в РЦОД может быть несколько типов ресурсов для хранения данных, добавлены атрибуты, указывающие на название хранилища данных для чтения исходных данных, для обработки и записи результатов выполнения вычислительных задач.

Атрибутами сущности *Slots* являются параметры слотов, которые входят в состав вычислительных компонент РЦОД: идентификатор, название, количество ядер, выделенных для конкретного пилота, название объекта для сбора статистики. Важно отметить, что суммарное количество ядер во всех слотах, принадлежащих одной вычислительной компоненте, не должно превышать общего количества ядер на ресурсе. Атрибут, отвечающий за активность слота, необходим для реализации возможности изменения количества вычислительных ресурсов в процессе функционирования РЦОД.

Атрибутами сущности *Links* являются параметры каналов связей между компонентами РЦОД: идентификатор, название, описание, пропускная способность, название объекта для сбора статистики. Как правило, канал связи соединяет объекты разных сущностей, имена которых обозначаются в соответствующих атрибутах. Для реализации возможного события отключения канала связи, предусмотрен атрибут, с помощью которого сущность становится неактивной.

Атрибутами сущности *DataTags* являются параметры типов данных, которые хранятся и обрабатываются в РЦОД: идентификатор, название, описание. Атрибутами сущности *DataFlows* являются параметры потоков данных для хранения и обработки в РЦОД: идентификатор, название, описание, тип данных, название объекта для сбора статистики. Каждый поток привязывается к конкретному хранилищу, в которое будут записываться данные. Под каждый поток в хранилище выделяется определенный объем ресурсов, при переполнении которого статус потока изменится на неактивный для записи. Важно отметить, что суммарный допустимый объем потоков данных, принадлежащих одному хранилищу, не должен превышать общий объем ресурса хранения. При построении ЦД может возникнуть необходимость в рассмотрении варианта работы РЦОД, при котором в хранилища данных уже записано некоторое количество файлов для обработки. Для описанного случая в данной сущности предусмотрен соответствующий атрибут.

Атрибутами сущности *JobQueues* являются параметры потоков задач, которые будут обрабатываться имеющиеся в РЦОД данные: идентификатор, название, описание, название объекта для сбора статистики. Для каждого типа задач должны быть определены такие значения, как: тип

входных/выходных данных, средний объем входных/выходных данных, среднее время выполнения задачи. К тому же, количество задач в потоках может отличаться, некоторые задачи могут запускаться чаще, а некоторые начинают исполняться только после завершения других. Все перечисленные детали нашли свое отражение в атрибутах. Немаловажно предусмотреть возможность случайной генерации файлов с данными и задач для их обработки, в связи с чем в сущность добавлены атрибуты, определяющие допустимые границы для изменения тех или иных значений.

Атрибутами сущности *TransportJobs* являются параметры объектов передачи потоков данных между хранилищами РЦОД: название, имена ресурсов хранения, между которыми передаются данные, и соединяющего их канала связи, тип передаваемых данных. Для каждого объекта должно быть установлено значение приоритета, которое указывает на последовательность просмотра устройств в ходе планирования процессов передачи данных. Разделение потоков данных на несколько распределенных хранилищ возможно с помощью атрибута, где будет указываться доля данных из общего потока для передачи на определенные ресурсы. Атрибут отложенного старта передачи данных введен для реализации ситуации перераспределения ресурсов хранения.

Атрибутами сущности *Events* являются параметры, описывающие возможные события, происходящие в РЦОД: идентификатор, название, описание, название объекта для сбора статистики. Такими событиями могут быть, например, остановка/запуск одного из элементов РЦОД, увеличение/уменьшение количества доступных вычислительных ресурсов или ресурсов хранения и другие. Атрибутами сущности *Event_Object* являются параметры распределений вероятностей возникновения событий для каждого объекта, с которым может произойти какое-либо событие: идентификатор события, название и тип объекта, которыми могут являться хранилища данных, вычислительные компоненты, слоты, каналы связи. Главными атрибутами являются тип распределения и значение вероятности возникновения события, которое может отличаться не только в зависимости от типа объекта, но и от его назначения. Для реализации зависимых событий, предусмотрены дополнительные атрибуты.

Атрибутами сущности *Sensors* являются параметры объектов для сбора статистики в процессе работы ЦД. Для каждого типа элемента, входящего в состав РЦОД: хранилища данных, вычислительные компоненты, пилоты, слоты, каналы связи, потоки данных, потоки задач, события — предусмотрен особый программный объект сбора статистики, имеющий идентификатор, имя и описание. Собранная информация должна усредняться и сохраняться с заданной частотой, что обеспечит ускорение процесса записи и сэкономит место в БД.

Атрибутами сущности *Experiments* являются параметры вариантов запуска ЦД для различных задач: идентификатор, название, описание, параметры запуска, дата и время создания, а также список идентификаторов объектов сбора статистики, которые могут меняться в соответ-

ствии с потребностями использования ЦД. Атрибутами сущности *Modifications* являются параметры модификаций (сценариев масштабирования) базовой конфигурации оборудования РЦОД: идентификатор, параметры оборудования РЦОД, которые изменяются в соответствии со сценарием относительно базовой конфигурации, статус, а также дата и время создания, запуска, завершения работы ЦД. Для идентификации записей с результатами работы ЦД по тому или иному сценарию добавлен соответствующий атрибут, который содержит значение идентификаторов запуска ЦД и сценария.

Атрибутами сущности *SimulationReport* являются параметры, описывающие результаты работы ЦД. Каждая запись содержит идентификатор, системное время сбора статистики, идентификатор и название объекта, входящего в состав РЦОД, информация о результатах работы которого содержится в записи. Для каждого типа объектов записываются данные, которые представляют наибольший интерес для отслеживания процессов происходящих в РЦОД в соответствии с описанием объектов сбора статистики. Результаты всех вариантов запуска ЦД с различными модификациями хранятся в одной таблице, поэтому предусмотрен атрибут, позволяющий идентифицировать запись каждого сценария.

Полное описание атрибутов перечисленных сущностей представлено в Приложении 1. Для каждого атрибута добавлены примечания, согласно которым разработана физическая модель данных для реализации БД. На рисунке 4 представлена логическая модель данных, где отражены все описанные сущности, их атрибуты и взаимосвязи.

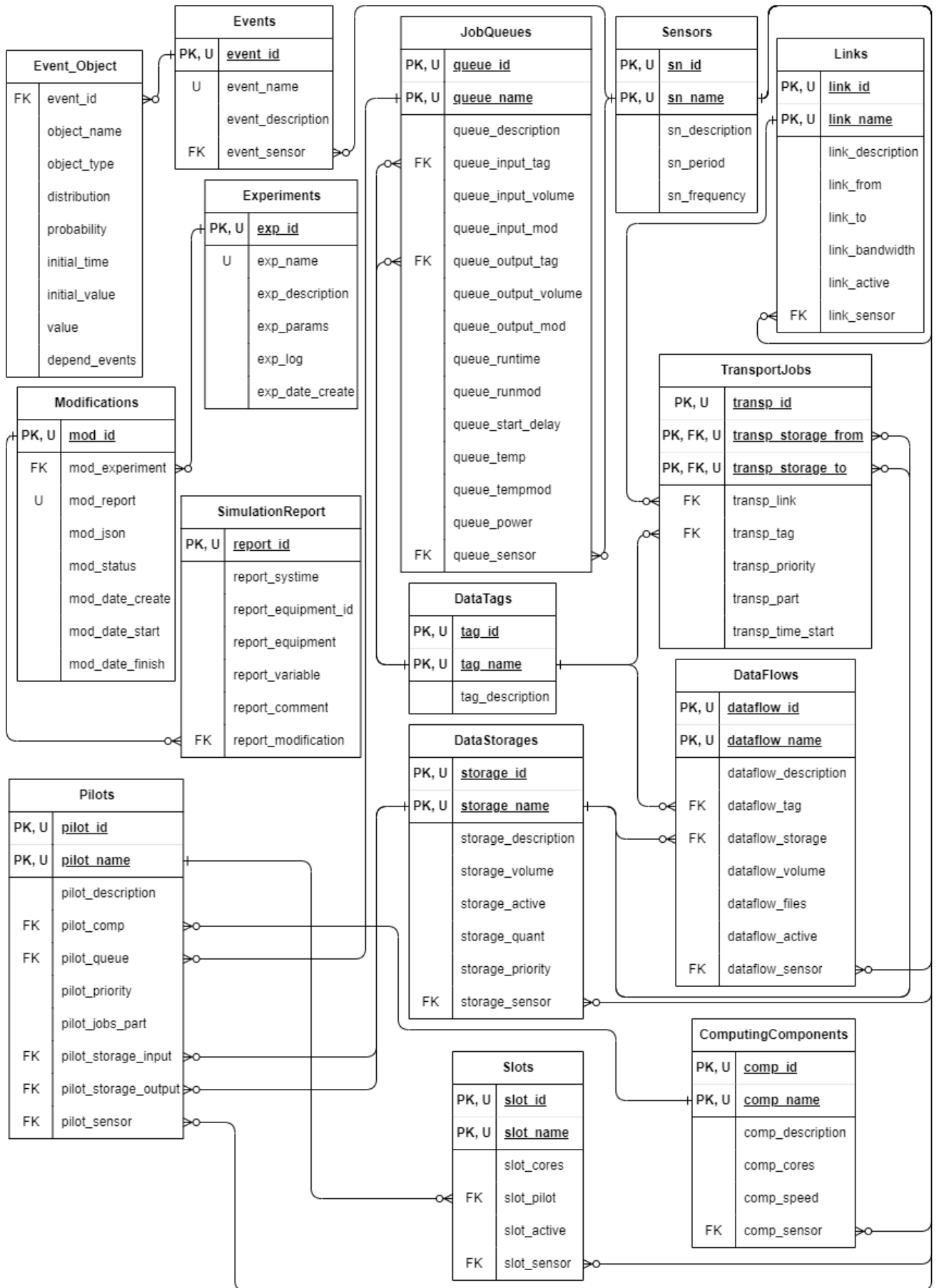


Рисунок 4. Логическая модель данных

2.6. Выводы к главе 2

Определены функциональные и нефункциональные требования к цифровым двойникам РЦОД. Требования включают в себя требования к ядру, которое является программой моделирования, и требования к модулю для взаимодействия пользователя с ЦД. Основными требованиями ядра является универсальность, адекватность моделирования с учетом характеристик потоков данных и задач, вероятностных процессов функционирования системы, а также применимость для моделирования процессов передачи, хранения и обработки данных любых РЦОД. К требованиям к модулю для взаимодействия пользователя с ЦД относится наглядность представления исходных данных, процесса моделирования и результатов, а также освобождение потенциального пользователя от участия в разработке и сопровождении ядра ЦД. Требования сформулированы в соответствии с вариантами использования будущего ЦД, основной задачей которого является оценка эффективности решения задач построения и развития РЦОД.

В результате сформулирован новый метод построения и использования ЦД для решения задач управления и развития РЦОД, в том числе улучшения их технических характеристик. Представлено формальное описание метода. Предложенный метод отличается от существующих возможностью моделировать процессы обработки и хранения данных, учитывая характеристики потоков данных и задач, вероятностей сбоев, отказов и изменений в производительности оборудования и других процессов, происходящих в моделируемой системе.

Для реализации метода разработаны алгоритмы работы ядра и модуля для пользователя. Алгоритм работы ядра можно разделить на три этапа: получение данных о РЦОД, формирование виртуального образа РЦОД, моделирование. Алгоритм работы модуля для взаимодействия пользователя с ЦД включает в себя построение инфраструктуры РЦОД, создание ЦД, просмотр результатов работы ЦД.

Создана модель данных, представлены сущности, которые описывают архитектуру РЦОД, входящее в его состав оборудование, потоки данных и задач, события, происходящих в РЦОД, сценарии возможного масштабирования системы, результаты работы ЦД.

Доказано первое положение, выносимое на защиту: «Разработан метод построения цифрового двойника, который позволяет описывать распределенные центры сбора, хранения и обработки данных, учитывая потоки данных и задач, а также процессы, происходящие в РЦОД».

Глава 3. Реализация алгоритмов и разработка специального программного обеспечения для создания цифровых двойников и взаимодействия с ними

3.1. Средства реализации

Для реализации метода построения цифровых двойников РЦОД требуется определить перечень технологий. Прежде всего необходимо выбрать язык программирования, который позволит создать кроссплатформенный программный продукт. Из множества существующих языков выбран *Python* — интерпретируемый объектно-ориентированный высокоуровневый язык программирования с динамической типизацией и автоматическим управлением памятью [57]. Его преимуществами для разработки сложных алгоритмов с множеством методов является наличие разнообразных библиотек, которые можно использовать для обработки аргументов командной строки, кодирования и декодирования данных в различные форматы, работы с высокоуровневыми математическими функциями и, наконец, взаимодействия с системами управления базами данных (СУБД). Более подробно используемые библиотеки языка *Python* будут описаны в следующих разделах при детальном описании разработанных алгоритмов.

Язык программирования *Python* может использоваться для создания веб-сервисов, которые будет удобно использовать для взаимодействия пользователей с ЦД. Так, например, свободно распространяемый веб-фреймворк *Django* с открытым исходным кодом поддерживает технологию объектно-реляционного отображения для доступа к сущностям БД (*ORM* — *Object-Relational Mapping*) [58]. Архитектура *Django* называется *Model Template View (MVT, модель-шаблон-представление)*, так как модели отображают структуры данных, необходимые приложению, на таблицы в БД, шаблоны отвечают за представление данных пользователю, а представление форматирует данные из моделей в определенный вид и передает их шаблонам [59]. Пользовательский интерфейс в таком случае удобно реализовать в виде адаптивных веб-страниц с помощью языка программирования *JavaScript* и библиотек *Bootstrap*, *Cytoscape.js* и *Plotly*. Более подробно используемые библиотеки будут описаны в следующих разделах при детальном описании разработанных алгоритмов.

Для хранения данных, необходимых для создания, запуска и функционирования ЦД выбрана свободно распространяемая объектно-реляционная СУБД *PostgreSQL* [60]. Реляционная БД позволяет удобно хранить данные, представленные в виде объектов, так как каждой сущности

БД можно поставить в соответствие определенного вида объект. Эффективная поддержка целостности данных, атомарных операций и параллельный доступ к БД – особенности реляционной БД, которые важны для реализуемых алгоритмов: должна быть возможность одновременно записывать данные в БД в процессе работы ЦД и читать данные из БД для их отображения пользователю. Физическая реализация БД основана на детальном описании атрибутов сущностей разработанной модели данных, что представлено в Приложении 1.

3.2. Реализация ядра цифрового двойника

В связи с тем, что ядро ЦД, в котором будет реализован процесс моделирования РЦОД, должно быть универсальным для моделирования процессов передачи, хранения и обработки данных любых РЦОД, необходимо придерживаться принципа разработки универсального программного пакета, который не нужно изменять под каждую моделируемую инфраструктуру. Для разработки ядра ЦД используется объектно-ориентированный подход.

Все объекты из модели данных описываются абстрактным базовым классом *Object_DC* (см. рис. 5), в который входят общие свойства элементов РЦОД, такие как идентификатор (*id*), имя (*name*), описание (*description*), приоритет (*priority*), статус активности (*active*) и список событий (*events*), которые могут произойти. В классе реализован конструктор и методы получения статистической информации о функционировании объекта (*GetStatistics*), добавления события (*AddEvent*) и сортировка событий по ближайшему времени наступления (*SortEventByMinTime*). Методы редактирования параметров объекта (*EditObject*) и обработки событий (*ProcessEvent*) являются абстрактными и реализуются в наследниках базового класса: *DataStorages*, *ComputingComponents*, *Pilots*, *Slots*, *Links*, *DataFlows*, *JobQueues*, *TransportJobs*, *Events*, *Sensor*.

На рисунке 6 представлена диаграмма, демонстрирующая общую структуру иерархии классов ядра, которые описывают сущности описанной модели данных, а также дополнительные объекты, позволяющие реализовать процесс моделирования работы РЦОД.

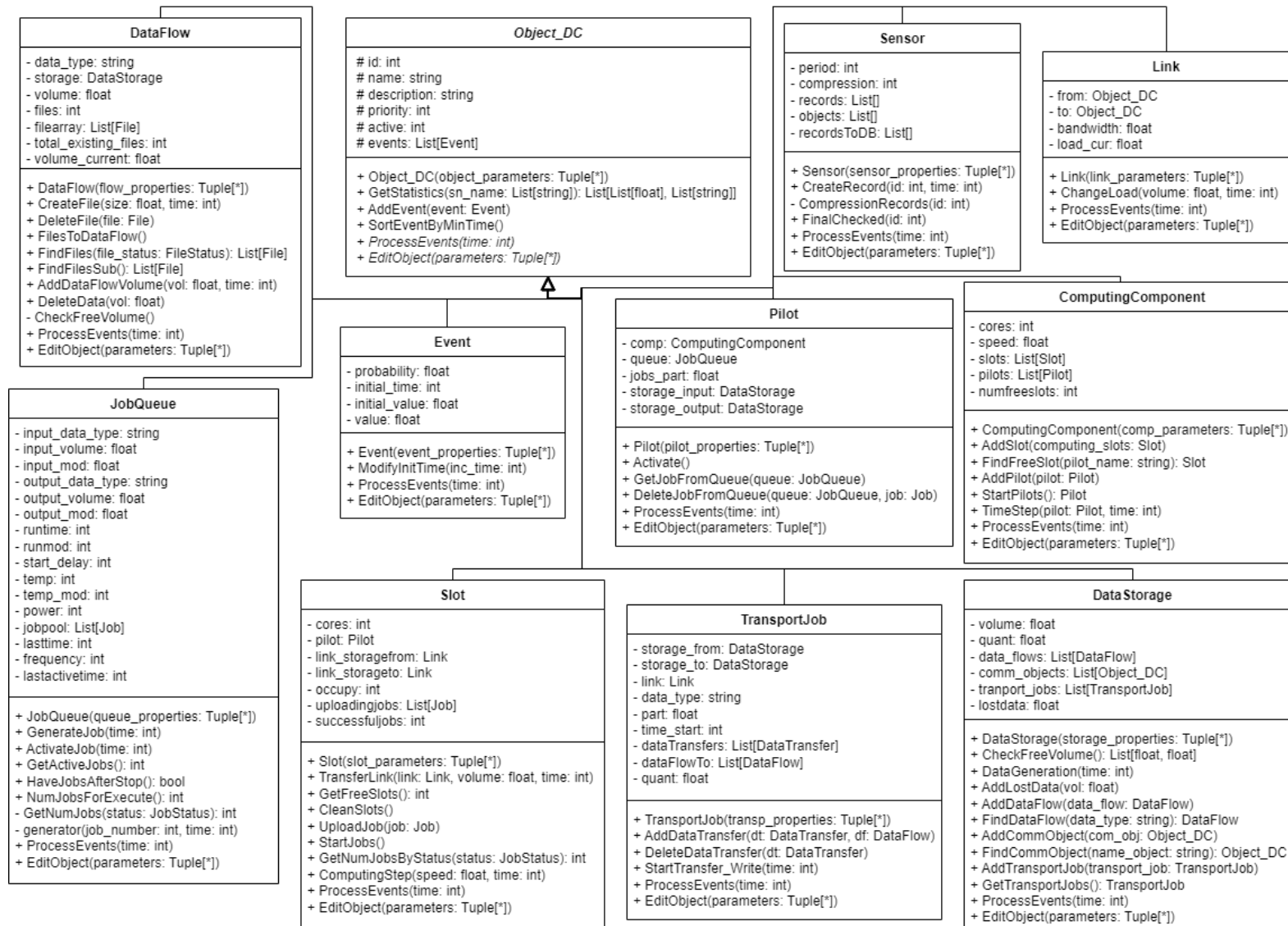


Рисунок 5. Диаграмма классов, описывающая иерархию наследования

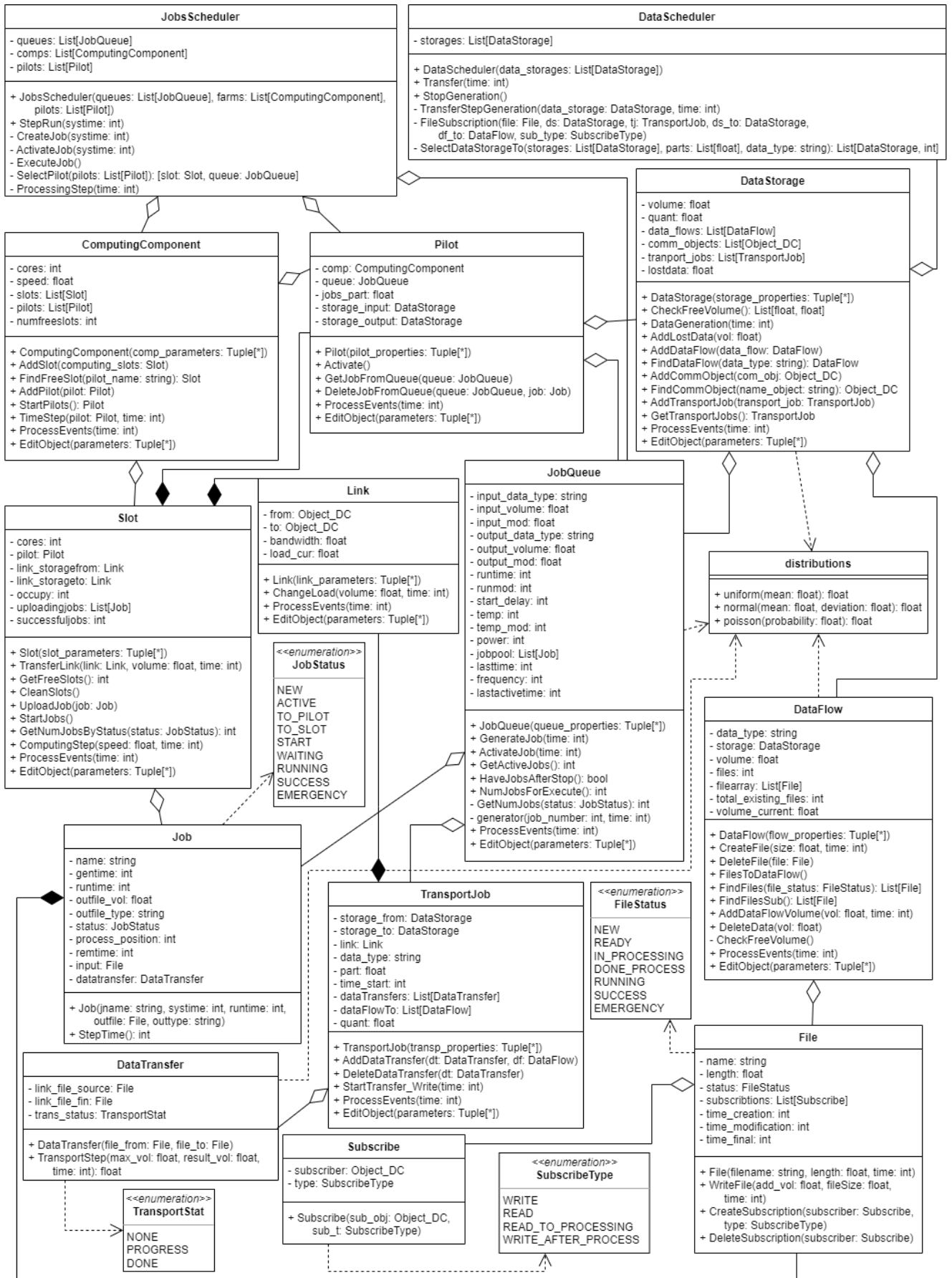


Рисунок 6. Детализированная диаграмма классов для ядра ЦД

Класс *DataStorage* описывает хранилища данных РЦОД. Отличительными свойствами объекта являются максимальный объем хранилища (*volume*) и скорость генерации данных в единицу времени (*quant*), если объект описывает источник данных. Источник данных может быть подписан на событие, меняющее его параметры. Источник данных не имеет памяти для хранения информации, поэтому если к началу очередного события генерации объем данных источника не равен нулю, то информация считается потерянной. В связи с этим необходим счетчик потерь данных (*lostdata*). Данными в хранилищах управляют потоки, которые описаны в классе *DataFlow*. Каждый поток привязывается к конкретному хранилищу (*storage*), в которое будут записываться данные определенного типа (*data_type*). Под каждый поток в хранилище выделяется определенный объем ресурсов (*volume*), при переполнении которого статус потока изменится на неактивный для записи. Данные в потоке хранятся в виде файлов (*files*).

Данные между хранилищами перемещаются с помощью объектов, описанных в классе *TransportJob*. Полями являются тип передаваемых данных (*data_type*), указатели на хранилища, между которыми перемещаются данные (*storage_from* и *storage_to*), указатель на канал передачи данных (*link*). Для реализации процесса разделения потоков данных на несколько хранилищ создано поле (*part*), содержащее значение доли данных из общего потока для передачи на определенные ресурсы. Существует поле отложенного старта передачи данных (*time_start*) для реализации ситуации перераспределения ресурсов хранения. Для детализации процесса передачи файлов с данными создан класс *DataTransfer*. Полями являются указатели на файлы для передачи (*link_file_source*) и приема (*link_file_fin*) данных, а также статус передачи (*trans_status*). Возможные статусы перечислены в элементе *TransportStat* (нет передачи, идет процесс передачи, передача завершена). Каналы связи между объектами РЦОД для передачи данных описаны в классе *Link*. Характеристиками канала связи являются пропускная способность (*bandwidth*) и ссылки на объекты инфраструктуры, между которыми выполняется связь (*from* и *to*).

Класс *File* описывает файлы с данными. У каждого файла есть имя (*name*), размер (*length*), время создания (*time_creation*), время изменения (*time_modification*), время завершения записи (*time_final*). Так как файл может использоваться для записи или чтения для обработки, добавлено поле, отражающее его статус (*status*), а также поле, описывающее объект-подписчик (*subscriptions*), требующий файл. В элементе *FileStatus* перечислены возможные статусы файла: новый, готов для использования, в процессе передачи, передан, в процессе обработки, обработан успешно, обработать не удалось. Объектом-подписчиком может являться как хранилище данных, так и вычислительная компонента, поэтому введен элемент *SubscribeType*, в котором перечислены типы допустимого использования файла: запись, чтение, чтение для обработки, запись после обработки.

Класс *ComputingComponent* описывает вычислительные компоненты РЦОД. Каждая вычислительная компонента содержит некоторое количество ядер (*cores*), на которых будут выполняться различные задачи. Для возможного уменьшения времени, затрачиваемого на выполнение задачи на вычислительной компоненте, создано поле, описывающее коэффициент ускорения (*speed*). Ядра вычислительной компоненты объединены в слоты, которые описаны в классе *Slot*. Именно на них происходит выполнение задач по обработке данных. На задачи разного типа выделяется определенное количество ядер (*cores*). В данном случае рассматривается модель с общими очередями для каждого класса задач, не закрепленные под конкретные вычислительные ресурсы, поэтому процесс запуска задач контролируют пилоты, которые описывает класс *Pilot*. Каждый пилот работает только в рамках одной вычислительной компоненты (*comp*) и управляет только определенным типом задач, которые формируются в некоторых очередях (*queue*). Так как все задачи из очереди могут быть разделены между несколькими пилотами, в класс добавлено поле (*jobs_part*), где будет указываться доля задач, которые должен забирать пилот из очереди. Дополнительно в класс добавлены поля, указывающие на хранилища данных для чтения исходных данных для обработки (*storage_input*) и записи результатов выполнения вычислительных задач (*storage_output*).

Задачи по обработке данных описаны в классе *Job*. Задачу характеризуют такие поля класса, как имя (*name*), время генерации (*gentime*), время выполнения (*runtime*), объем выходных данных (*outfile_vol*), тип выходных данных (*outfile_type*), счетчик выполненных операций (*process_position*). Задача должна обрабатывать файл, поэтому добавлено поле, которое содержит ссылку на входные данные для обработки (*input*), а также поле, которое содержит ссылку на объект детализации процесса передачи файла из хранилища (*datatransfer*). Для отслеживания статуса выполнения задачи (*status*) введен элемент *JobStatus*. Возможные варианты статуса задачи: новая, активирована, взята пилотом, отправлена на слот для исполнения, запущена, ожидает файл для обработки, выполняется, завершена успешно, завершена с ошибкой. После создания все задачи попадают в очереди (потoki), которые описывает класс *JobQueue*. Для каждой очереди определены такие характеристики, как: тип входных/выходных данных (*input_data_type* и *output_data_type*), средний объем входных/выходных данных (*input_volume* и *output_volume*), среднее время выполнения задачи (*runtime*). К тому же, количество задач в потоках может отличаться, некоторые задачи могут запускаться чаще, а некоторые начинают исполняться только после завершения других, поэтому в класс добавлены соответствующие поля (*power*, *temp*, *start_delay*). В связи с тем, что файлы с данными и задачи для их обработки генерируются в процессе моделирования РЦОД, в класс добавлены поля для значений допустимой модификации объемов входных/выходных данных (*input_mod* и *output_mod*), время выполнения задачи (*runtime_mod*) и частоты запуска (*temp_mod*).

Для описания возможных событий, происходящих в РЦОД, создан класс *Event*. Полями класса являются вероятность возникновения события (*probability*), время наступления (*initial_time*) и допустимые значения для изменения свойств объектов (*initial_value*, *value*). Для сбора статистики в процессе работы ЦД требуются объекты класса *Sensor*, которые характеризуются периодом усреднения собранной информации (*period*) и частотой записи в БД (*compression*).

Ключевыми элементами в реализуемом ядре являются планировщик перемещения данных (класс *DataScheduler*) и планировщик управления задачами (класс *JobsScheduler*). Жизненный цикл этих элементов и их основные функции представлены на рисунках 7 и 8 соответственно. Полем класса *DataScheduler* является список ссылок на объекты (*storages*), которые являются хранилищами данных, так как планировщик перемещения данных должен контролировать любые операции с файлами. В каждую единицу времени планировщик перемещения данных проверяет работоспособность всех хранилищ, которые в свою очередь совершают обработку событий, а затем дают ответ о статусе своей активности. Затем осуществляется проверка хранилищ на наличие новых данных, которые генерируются в потоках данных согласно требуемому закону распределения вероятностей. В случае их обнаружения запускается процесс передачи данных из текущего хранилища (*Transfer()*) или из генератора (*TransferStepGeneration()*), если он предусмотрен вычислительной инфраструктурой, в пункты назначения (*SelectDataStorageTo()*) по списку объектов, запрашивающих определенный файл (*FileSubscription()*). Полями класса *JobsScheduler* являются списки ссылок на объекты, которые содержат очереди задач на исполнение (*queues*), и отвечают за запуск задач (*pilots*) на определенных вычислительных ресурсах (*comps*). В каждую единицу времени планировщик создает некоторое количество задач (*CreateJob()*), время выполнения которых является случайной величиной, распределенной по заданному закону, активирует ранее созданные задачи (*ActivateJob()*) и посылает в очередь на исполнение на вычислительные компоненты. Далее происходит выбор пилота, который будет запускать задачу на вычислительной компоненте (*SelectPilot()*). Пилот анализирует количество свободных ядер на вычислительной компоненте. Если доступны свободные ресурсы, то осуществляется запуск задачи (*ExecuteJob()*): пилот берет задачу из соответствующей очереди и отправляет ее на исполнение на вычислительные ресурсы, в процессе чего происходит обработка входного файла при его наличии в хранилище (*ProcessingStep()*). После завершения работы каждой задачи планировщик удаляет ее из очереди и освобождает вычислительные ресурсы.

Генерация всех данных и задач, времен наступления событий осуществляется с использованием класса *distributions*, где с помощью модуля работы с высокоуровневыми математическими функциями и многомерными массивами *NumPy* [61] реализованы методы генерации случайных значений согласно различным распределениям (равномерное (*uniform()*), нормальное (*normal()*), Пуассона (*poisson()*)).

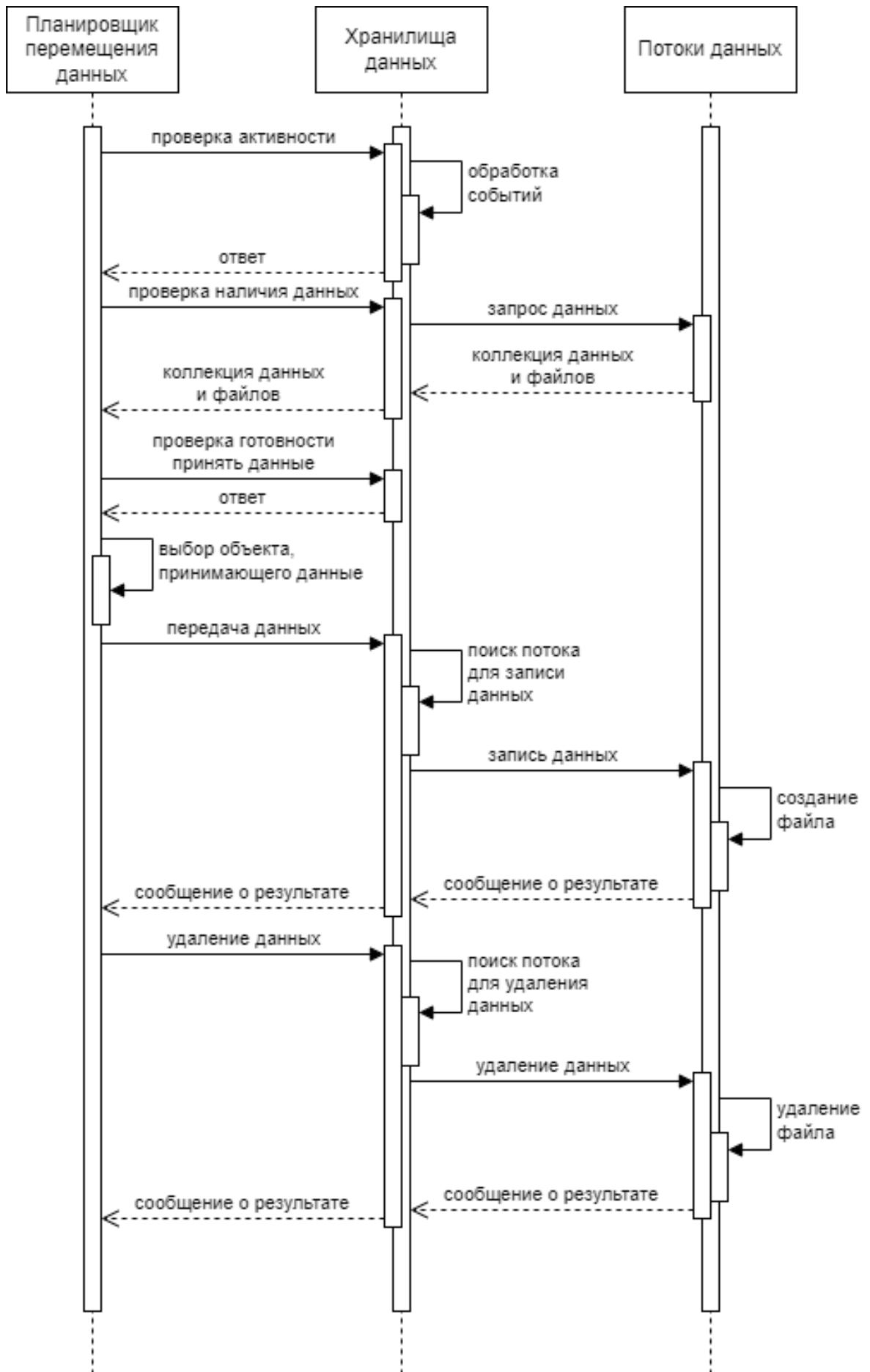


Рисунок 7. Диаграмма последовательности для планировщика перемещения данных

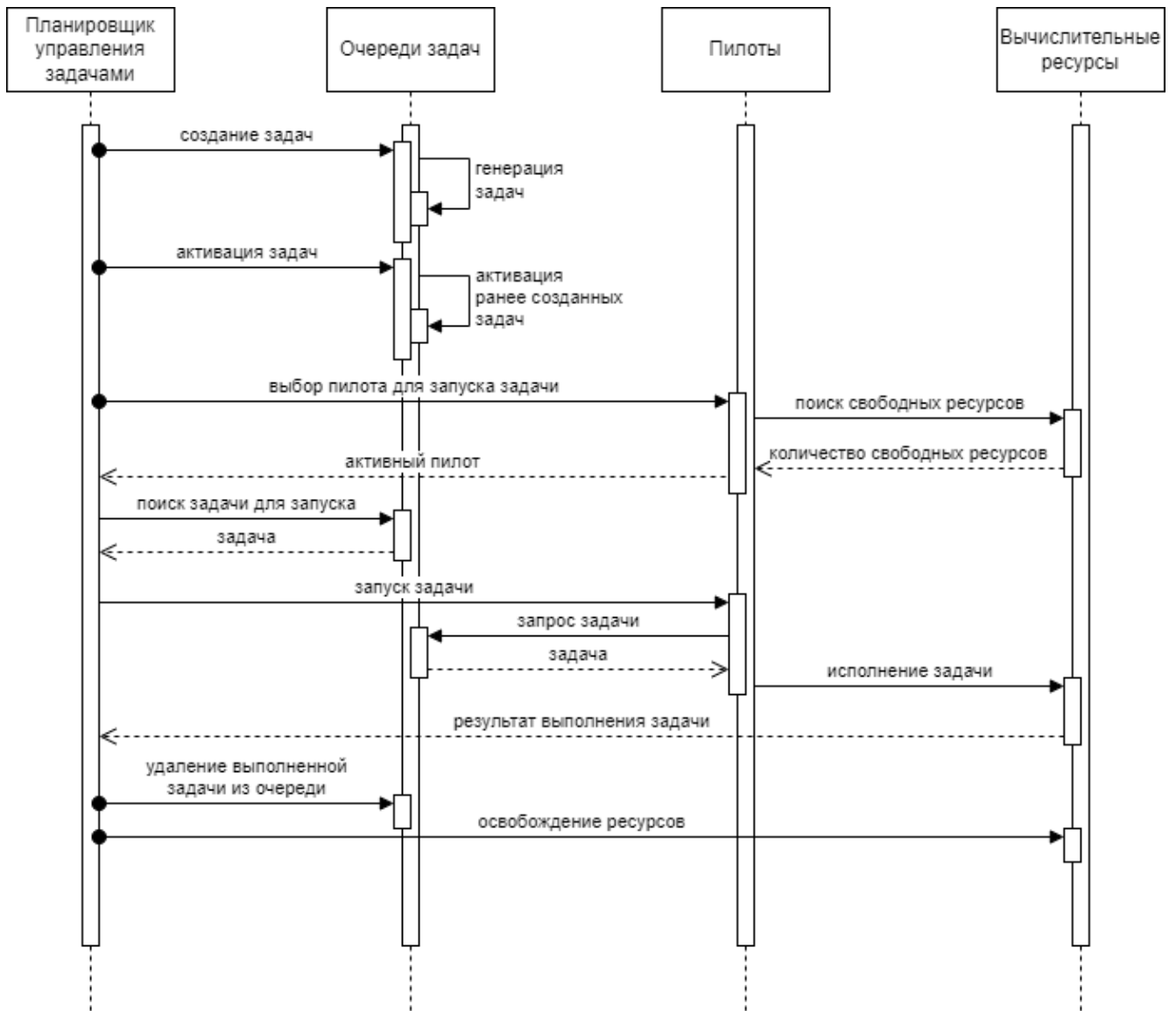


Рисунок 8. Диаграмма последовательности для планировщика управления задачами

Теперь рассмотрим общий принцип работы ядра, его связи с описанными классами и дополнительные параметры, которые необходимы для его работы, что отражено на рисунке 9.

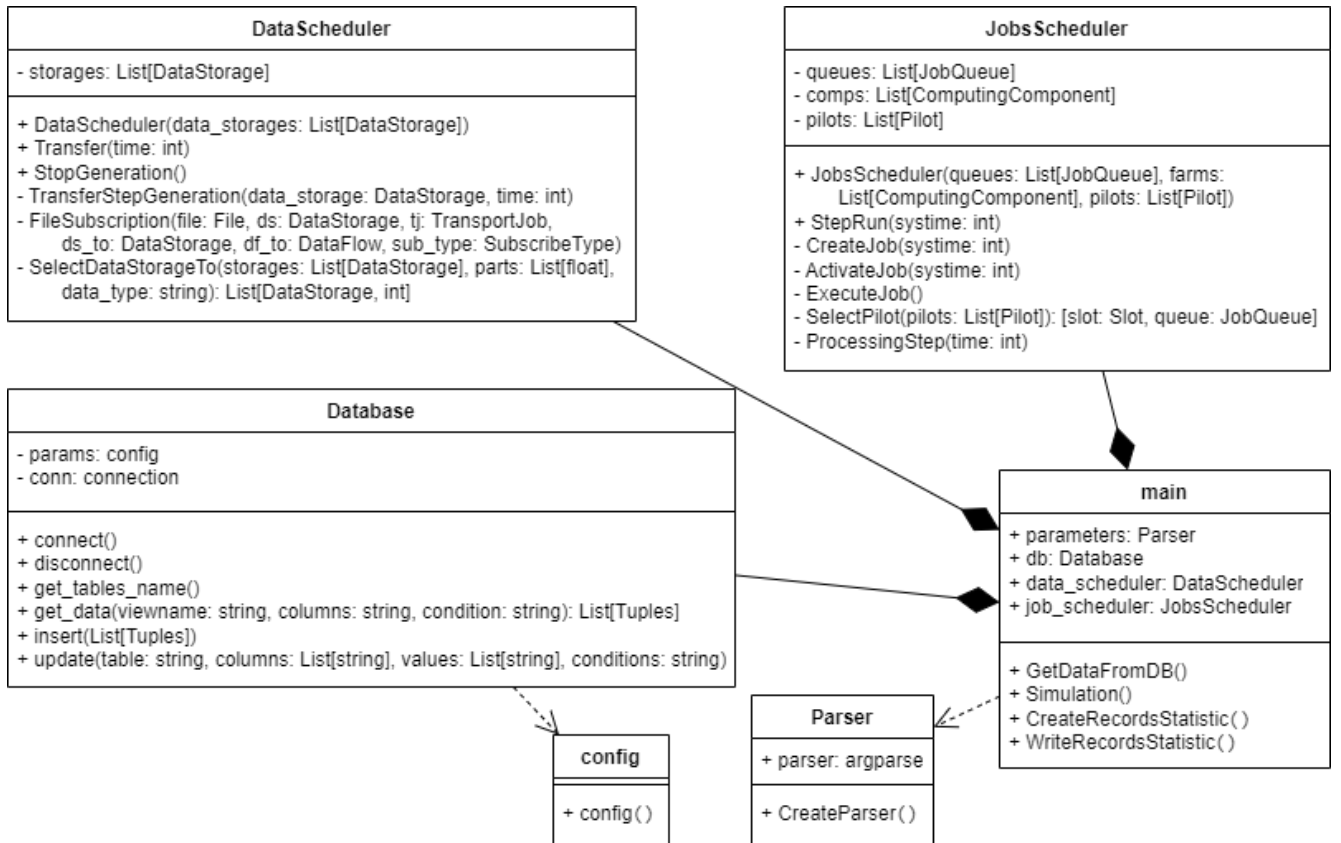


Рисунок 9. Общая диаграмма классов для ядра ЦД

Разработанная программа должна запускаться с помощью командной строки, поэтому разработан класс *Parser*. Единственный метод класса с помощью библиотеки *argparse* [62] обрабатывает заданные аргументы, которые необходимы для дальнейшей работы программы. К таким аргументам относятся: время работы экспериментальной установки в часах, для которой создается вычислительная инфраструктура (*simulation_time*); коэффициент временного разрешения (*resolution_factor*), который необходим для ускорения работы программы и получения результатов в более короткие сроки (расчеты производятся не для каждой секунды астрономического времени, а для времени, на которое указывает этот коэффициент); идентификатор ЦД для ведения журнала записей результатов его работы (*log_id*).

Первым шагом после запуска программы является настройка соединения с БД и получение всей необходимой информации о РЦОД. Настройки подключения к БД описываются в специальном файле с расширением *.ini*, где указываются такие параметры, как хост, порт, имя пользователя, пароль и имя БД. Единственный метод класса *config* с помощью модуля *configparser* [63] разбирает параметры конфигурационного файла. Класс *Database* описывает соединение к БД и с помощью модуля *psycopg2* реализует методы взаимодействия с СУБД *PostgreSQL*: подключение и отключение (*connect()* и *disconnect()*), получение списка таблиц (*get_tables_name()*), получение данных из заданной таблицы (*get_data()*), добавление новых данных (*insert()*), обновление данных (*update()*).

Второй этап алгоритма — формирование виртуального образа рассматриваемого РЦОД. На этом этапе по исходным данным, полученным из БД, осуществляется создание объектов будущей модели и связей между ними. Далее определяются структуры для потоков данных и задач, вырабатывается поток событий, на которые подписываются объекты РЦОД. Создаются планировщики процессов перемещения данных и задач, а также объекты, которые будут собирать и записывать в БД статистическую информацию о процессах, происходящих в различных элементах РЦОД.

Третий этап — непосредственно моделирование. Моделирование проводится по постоянным временным единицам, минимальное значение которых составляет 1 секунда. Так как программа предназначена для анализа потока данных и задач в системе, вызываются методы планировщика перемещения данных (*data_scheduler.Transfer()*) и планировщика управления задачами (*job_scheduler.StepRun(system_time)*). Для каждого построенного на предыдущем этапе алгоритма объекта модели в соответствии с потоком событий производятся изменения в данных, которые соответствуют временной единице работы РЦОД. На протяжении всего этапа моделирования осуществляется получение статистической информации о всех происходящих процессах (*CreateRecordsStatistic()*) и запись данных в БД (*WriteRecordsStatistic()*).

Таким образом реализовано ядро ЦД с учетом всех объявленных требований.

3.3. Реализация модуля для взаимодействия пользователя с цифровым двойником

Модуль для взаимодействия пользователя с ЦД функционирует в диалоговом режиме. Модуль реализован в виде адаптивных веб-страниц с учетом всех предъявленных требований. Использовались технологии *HTML*, *CSS*, *Bootstrap* и *JavaScript*. Каждая страница описана в отдельном шаблоне для удобства дальнейшего использования при сборке проекта в СПО.

Взаимодействие пользователя с ЦД начинается с построения инфраструктуры РЦОД (см. рис. 10). На странице представлен список объектов, соответствующий возможному оборудованию РЦОД, а также поле для расположения нужных объектов и соединения их в общую инфраструктуру. При перемещении объекта на поле открывается форма для ввода его параметров. При нажатии на кнопку «Добавить устройству», вся информация об объекте сохраняется в БД. Таким образом происходит настройка базовой конфигурации. После того, как все необходимые объекты будут добавлены, пользователь должен настроить каналы связи между ними. Под каналами связи понимаются логические соединения между компонентами РЦОД, по которым пере-

даются данные. Для настройки каналов следует выбрать объекты, между которыми осуществляется соединение, и задать требуемые параметры (см. рис. 11). Для создания потоков данных и потоков задач по аналогии созданы формы для ввода требуемых параметров. Информация о каналах связи, потоках данных и потоках задач также должна быть добавлена в БД. Пользователь может редактировать инфраструктуру РЦОД, базовую конфигурацию оборудования, изменять параметры потоков данных и задач до перехода на следующий этап, который заключается непосредственно в создании ЦД.

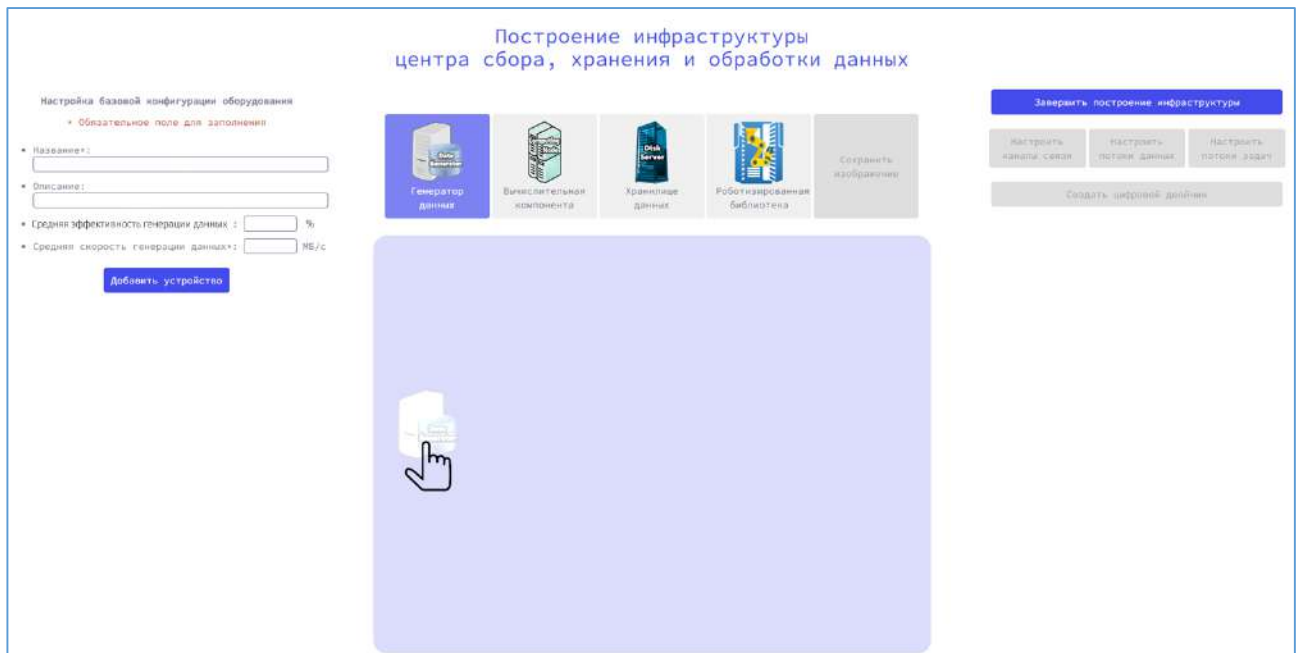


Рисунок 10. Страница для построения инфраструктуры РЦОД и настройки базовой конфигурации оборудования



Рисунок 11. Страница для построения инфраструктуры РЦОД и настройки каналов связи между объектами

Построение инфраструктуры РЦОД является одной из важнейших функций. Процесс подразумевает анимированное расположение объектов на форме (*HTML*-элемент *canvas*), по которой их можно двигать, соединять, выделять. Построение конфигурации напоминает процесс построения взвешенного графа, где добавленные объекты — это вершины, а связи между ними, которые описывают каналы связи с пропускной способностью — ребра с весами. В связи с этой особенностью для реализации функционала выбрана библиотека *Cytoscape.js*, которая ориентирована на построение графов и сетей. Библиотека имеет встроенный рендеринг с жестами и событиями, а также позволяет манипулировать высоко настраиваемыми и интерактивными графиками [64].

Для описания целей создания ЦД и ввода дополнительных параметров для процесса моделирования создана соответствующая простая *HTML*-страница для добавления вычислительных экспериментов на ЦД (см. рис. 12). В каждом вычислительном эксперименте имеется доступ к базовой конфигурации оборудования, которая создавалась при построении инфраструктуры РЦОД, также пользователь может добавить дополнительные модификации параметров оборудования и ввести события, происходящие в РЦОД. После настройки вычислительных экспериментов осуществляется запуск ЦД (см. рис. 13).

Добавление эксперимента

Заполните поля формы, чтобы добавить новый эксперимент для поиска оптимальной конфигурации оборудования

* Обязательное поле для заполнения

Название эксперимента *

Описание эксперимента

Параметры моделирования

- Продолжительность работы моделируемой инфраструктуры – ч.
- Ускорение процесса моделирования в раз.

Параметры логирования

Выберите объекты и события, о которых необходимо сохранять информацию во время моделирования

- Объекты моделируемой инфраструктуры
 - Хранилища данных
 - Вычислительные компоненты
 - Каналы связи
- События
 - Генерация данных
 - Потери данных
 - Работа с файлами
 - Генерация, запуск, выполнение задач

Добавить

Очистить

Отмена

Рисунок 12. Страница для добавления вычислительного эксперимента

Информация об эксперименте

Дата создания: 7 февраля 2023 г. 10:36

Название эксперимента
Test 1

Описание эксперимента
Поиск оптимального количества ресурсов для хранения данных.

Параметры моделирования

- Продолжительность работы моделируемой инфраструктуры – 800 ч.
- Ускорение процесса моделирования в 1000 раз.

Параметры логирования

- Объекты моделируемой инфраструктуры
 - Хранилища данных
 - Вычислительные компоненты
 - Каналы связи
- События
 - Генерация данных
 - Потери данных
 - Работа с файлами
 - Генерация, запуск, выполнение задач

Посмотреть результаты

Выбрать другой эксперимент

Базовая конфигурация

Хранилища данных		
Название	Описание	Объем (ТБ)
trigger	Trigger SWM	10000,0
buffer	Data description buffer	5400,0
eoslhp	Main storage LHEP	1000,0
eoslit	Main storage LIT	1000,0
dcach	DC	1000,0

Вычислительные компоненты		
Название	Описание	Количество ядер
tzlit	LIT T2 farm	596
ncxlhp	LHEP main farm	1299
super	Governor	199

Каналы связи		
Название	Описание	Пропускная способность (ГБ/с)
raw0	trigger - buffer	100,0
raw1	buffer - lhp	10,0
raw2	buffer - lit	10,0
compute0	lhp - farm lhp	10,0
compute1	lit - governor	10,0
compute2	lit - farm lit	10,0
dataeoslhp	eoslhp - eoslit	10,0
dataeoslitlhp	eoslit - eoslhp	10,0

Добавить модификацию

Список модификаций

#	Статус	Дата обновления	
16	NEW	9 марта 2023 г. 14:52	<div style="display: flex; gap: 5px;"> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Просмотр</div> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Запуск</div> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Результаты</div> </div>
15	DONE	10 марта 2023 г. 10:18	<div style="display: flex; gap: 5px;"> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Просмотр</div> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Запуск</div> <div style="background-color: #0056b3; color: white; padding: 2px 5px;">Результаты</div> </div>

Список событий

Добавить событие

Рисунок 13. Страница для просмотра информации о вычислительном эксперименте и запуска ЦД

Согласно требованиям, в процессе или по завершении работы ЦД пользователю должны быть доступны для просмотра результаты ЦД. Для реализации этой функции использована библиотека *Plotly* [65]. На странице пользователю открываются интерактивные графики в зависимости от типа выбранного оборудования (см. рис. 14). Интерактивные графики подразумевают возможность масштабирования и выбора данных для просмотра, что помогает определить несколько потенциально конкурентоспособных вариантов из множества допустимых конфигураций оборудования. Пользователь может сохранить результаты работы ЦД для какой-либо модификации в виде изображений с графиками. Все эти возможности предоставляются библиотекой *Plotly*.



Рисунок 14. Страница для просмотра результатов работы ЦД

Таким образом реализован модуль для взаимодействия пользователя с ЦД с учетом всех объявленных требований.

3.4. Разработка специального программного обеспечения

Реализованные модели, методы, модули и алгоритмы объединены в единый программный пакет для дальнейшего использования уже в составе СПО, которое является проблемно-ориентированной системой управления, принятия решений и оптимизации на основе моделей ЦД. Архитектура СПО представлена на рисунке 15. СПО содержит три основных блока: хранение данных, моделирование, представление результатов. Хранение данных — накопление структурированной информации для обеспечения процессов моделирования и анализа результатов. Моделирование — алгоритмы работы ядра ЦД. Представление результатов — модуль для взаимодействия пользователя с ЦД, с помощью которого можно задавать исходные данные, описывающие РЦОД и параметры для моделирования, а также просматривать результаты работы ЦД.



Рисунок 15. Архитектура СПО

Для реализации описанных ранее методов, модулей и алгоритмов использовался язык программирования *Python*, поэтому для объединения всех компонент в единое СПО как веб-сервис использовался фреймворк *Django*. Этот фреймворк поддерживает технологию объектно-реляционного отображения для доступа к сущностям БД, а особенности архитектуры позволяют запустить проект как серверное приложение. Проект с *Django* состоит из директории проекта с общими настройками и каталогов подключаемых и отключаемых от проекта приложений. Приложения подключаются в файле общих настроек проекта.

Каждой странице, которая разработана для модуля взаимодействия пользователя с ЦД, задается шаблон, который определяет, как будет отображена страница пользователю, представление, которое принимает запрос на получение страницы, и *url*-адрес, на котором эта страница доступна. Для определения *URL*-адресов приложения, используется модуль *Python URLconf* (конфигурация *URL*). Этот модуль содержит код *Python*, который с помощью регулярных выражений отображает *URL*-шаблоны, и связанные функции *Python* (представления). Конфигурации могут ссылаться друг на друга и создаваться динамически [66]. В результате у страницы имеется шаблон, который определяет, как будет отображена страница пользователю, ее представление, которое принимает запрос на получение страницы, и адрес, на котором эта страница доступна.

Для работы с БД у фреймворка *Django* существует файл настроек проекта (*settings.py*), где необходимо указать соответствующий используемому типу СУБД драйвер. В этом же файле ука-

зываются имя, хост, порт и данные для подключения к БД. Каждой сущности, которая необходима для хранения данных реализуемой системы, в *Django* сопоставляются классы-модели, описанные в файле *models.py*. Модель содержит определенные поля, у каждого из которых есть название, тип и могут быть дополнительные свойства, соответствующие возможным свойствам атрибутов БД: например, свойство уникальности значений или указание на то, что поле является внешним ключом. *Django* предоставляет некоторые встроенные методы моделей для доступа к данным в БД, а также дает возможность брать данные из БД, используя вручную написанные *SQL*-запросы.

Наибольший интерес представляет серверная часть процесса построения инфраструктуры РЦОД. При выборе устройства из списка, открывается форма для ввода его параметров. Этот функционал реализован с помощью *AJAX*-запросов, чтобы после добавления параметров очередному устройству не перезагружалась страница. С помощью *JavaScript* собираются данные с формы, и формируется *AJAX*-запрос, который отправляет полученные данные на представление, где данные обрабатываются и сохраняются в БД. Затем приходит ответ от сервера, что данные успешно добавлены в БД, либо произошла ошибка, о чем сообщается пользователю.

Подробная инструкция для пользователя с детальным описанием всех возможных действий по работе со СПО находится в Приложении 2.

По окончании разработки и реализации проблемно-ориентированной системы управления, принятия решений и оптимизации на основе моделей ЦД получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» от 14 августа 2023 года (Приложение 3).

Для установки и функционирования СПО предъявляются следующие минимальные системные требования:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 2 ГГц;
- 16 ГБ оперативной памяти;
- 1 ГБ свободного дискового пространства.

Для хранения результатов работы ЦД потребуется дополнительное дисковое пространство из расчета не менее 30 МБ на один вычислительный эксперимент.

3.5. Выводы к главе 3

Согласно описанным требованиям, реализованы алгоритмы построения цифровых двойников РЦОД. Используются современные архитектурные решения, принципы объектно-ориентированного программирования и средства для разработки программного обеспечения, веб-приложений и БД.

Ядро ЦД представляет собой программу, состоящую из большого количества классов, которые описывают всевозможные объекты, которые могут существовать в различных РЦОД, а также дополнительные элементы, позволяющие реализовать процесс моделирования работы РЦОД. При этом учтен принцип разработки универсального программного пакета, который не нужно изменять под каждую моделируемую инфраструктуру.

Модуль для взаимодействия пользователя с ЦД реализован в виде веб-сервиса с адаптивными веб-страницами для обеспечения работы в диалоговом режиме. При необходимости в сервис можно добавлять дополнительные страницы по разработанным шаблонам, что упрощает дальнейшее развитие проекта.

Для хранения данных, необходимых для создания, запуска и функционирования ЦД реализована реляционная БД. Доступ к данным осуществляется по средствам библиотек, реализованных для языка программирования *Python*, который использован для реализации всех алгоритмов.

Разработано СПО, которое является проблемно-ориентированной системой управления, принятия решений и оптимизации на основе моделей ЦД. СПО используется для построения цифровых двойников РЦОД и позволяет сравнивать эффективность функционирования РЦОД в зависимости от различных конфигураций оборудования. СПО включает в себя БД, ядро ЦД и модуль для взаимодействия пользователя с ЦД.

СПО реализует методы построения ЦД центров хранения и обработки данных, моделирования таких центров, генерации потоков данных и задач, визуализации. СПО может применяться для широкого класса задач в области проектирования, построения и развития РЦОД, в том числе помогает выбрать несколько потенциально конкурентоспособных вариантов из множества допустимых конфигураций оборудования.

Получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» от 14 августа 2023 года (Приложение 3).

Доказано второе положение, выносимое на защиту: «Разработаны алгоритмы, на основе которых создано специальное программное обеспечение, используемое для принятия решений по выбору конфигурации оборудования распределенных центров сбора, хранения и обработки данных согласно заданным требованиям».

Глава 4. Верификация и опытная эксплуатация специального программного обеспечения для создания цифровых двойников

4.1. Верификация ядра цифрового двойника

4.1.1 Постановка задачи

Согласно представленным требованиям к цифровым двойникам РЦОД, адекватность ЦД должна быть проверена результатами работы существующего РЦОД по разным параметрам, доступным при мониторинге системы. При этом результаты работы ЦД не должны отклоняться от среднего значения данных мониторинга более, чем на три среднеквадратических отклонения. Поэтому необходимо провести верификацию программы моделирования, которая является ядром ЦД. Дальнейшее изложение материала представлено в соответствии с опубликованной статьей [67].

Верификация программы моделирования осуществлялась на примере вычислительной инфраструктуры эксперимента *VM@N* [68] ускорительного комплекса *NICA* [12], который создается в России на базе ОИЯИ в городе Дубна Московской области. Экспериментальная установка *VM@N* является одним из элементов первого этапа реализации комплекса *NICA*. После проведения серии технических сеансов эксперимента зимой 2022–2023 года состоялся первый физический сеанс, в котором было набрано более 550 миллионов событий взаимодействия пучка ионов ксенона на мишени цезий-йод, подлежащих дальнейшей обработке и физическому анализу полученных экспериментальных данных [69]. Вычислительная инфраструктура эксперимента включает в себя различные ресурсы, а именно:

1. кластер *NICA*, который располагается в Лаборатории физики высоких энергий имени В.И. Векслера и А.М. Балдина (ЛФВЭ) ОИЯИ;
2. компоненты распределенной грид-инфраструктуры Многофункционального информационно-вычислительного комплекса (МИВК) ОИЯИ [70]: ресурсный центр первого уровня *Tier1* ЛИТ и ресурсный центр второго уровня *Tier2* ЛИТ;
3. Суперкомпьютер «Говорун», входящий в состав гетерогенной платформы *HybriLIT* [71] (МИВК ОИЯИ);
4. хранилище данных на распределенной файловой системе *EOS* (ЛИТ ОИЯИ).

Для интеграции объектов инфраструктуры и обеспечения единого доступа к ним с целью запуска задач массовой обработки данных используется система *DIRAC* [72]. Мониторинг и оценка производительности вычислительных ресурсов в результате выполнения задач эксперимента также осуществляется с помощью программного обеспечения *DIRAC Interware* [73].

Таким образом, в качестве исходных данных для верификации использовалась статистика, полученная по результатам мониторинга с помощью программного обеспечения *DIRAC Interware* при запуске задач преобразования полученных «сырых» (необработанных) экспериментальных данных в формат *digit* (далее *RawToDigit* задачи) и задач преобразования данных формата *digit* в данные реконструированных событий столкновения частиц формата *DST* (далее *DigitToDst* задачи).

Для достижения поставленной цели проведено три этапа моделирования. Задача первого этапа заключается в осуществлении моделирования процесса сбора экспериментальных данных, определении количества ресурсов, необходимых для их хранения. Задачей второго и третьего этапа является моделирование процесса запуска задач *RawToDigit* и *DigitToDst* соответственно, измерение таких показателей, как загрузка вычислительных ресурсов, общее время выполнения задач, скорость передачи данных.

4.1.2 Описание данных эксперимента и результатов мониторинга

Рассматривается физический сеанс работы эксперимента *BM@N*, который проходил с декабря 2022 года по февраль 2023 года. Общее время сбора данных составило примерно 720 часов. «Сырые» необработанные данные (далее *raw*-данные) поступали с установки со скоростью, усредненной по всему времени проведения сеанса, 142 МБ/с. Согласно результатам мониторинга [74], по окончании сеанса суммарный объем физических *raw*-данных составил 379 ТБ (см. рис. 16). В течение каждого запуска при наборе данных эксперимента в буфере приема и хранения данных осуществлялась запись экспериментальных данных в *raw*-файлы. Объем отдельного файла составил 15 ГБ. По готовности *raw*-файлы в полном объеме копировались в хранилище данных на файловой системе *EOS*. Объем полученных экспериментальных *raw*-данных, записанных в хранилище, соответствует 25 800 *raw*-файлам для обработки. Обработка экспериментальных данных представляет собой преобразование и последующее получение реконструированных данных событий.

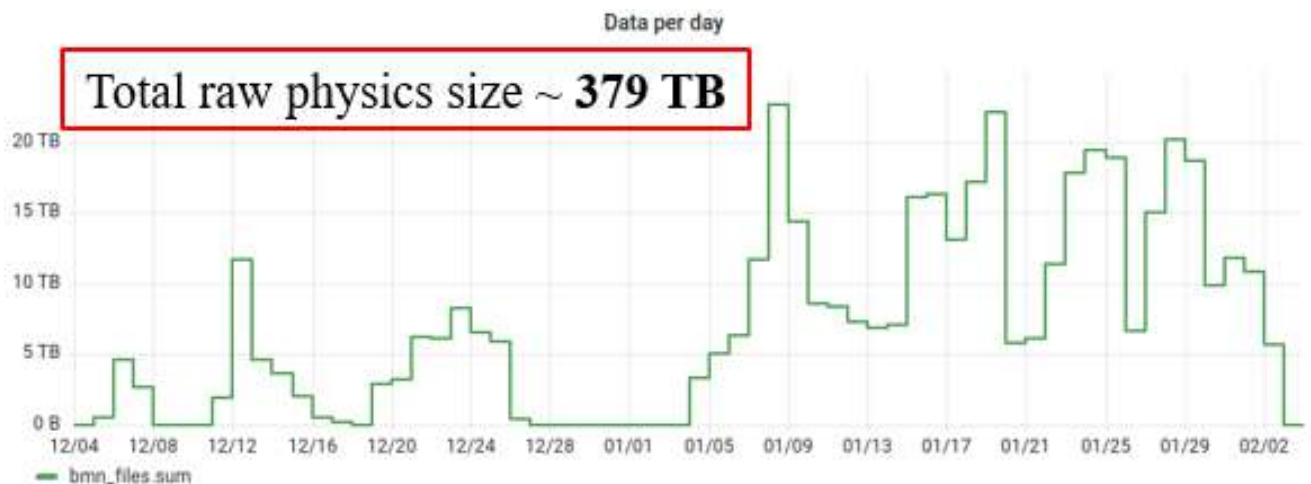


Рисунок 16. Реальный объем поступающих *raw*-данных эксперимента *BM@N*

Преобразование *raw*-файлов в *digit*-файлы (*RawToDigit* задачи) осуществлялась на вычислительных ресурсах эксперимента: кластере *NICA* и *Tier1* ЛИТ. Каждая задача *RawToDigit* обрабатывает 1 файл «сырых» экспериментальных данных один раз. Размер *digit*-файла составляет в среднем 870 МБ. Все *RawToDigit* задачи были отправлены на выполнение одновременно. Общее время обработки всех *raw*-файлов составило примерно 36 часов. На рисунках 17 и 18 изображены графики, отражающие количество выполненных задач на вычислительных компонентах кластера *NICA* ЛФВЭ и *Tier1* ЛИТ соответственно за указанный период времени. По графику на рисунке 17 видно, что вычислительные ресурсы кластера *NICA* ЛФВЭ загружены равномерно, каждый час выполнялось примерно по 100 задач. Это позволило сделать предположение о том, что на кластере *NICA* ЛФВЭ предоставлялось 100 ядер. По графику, представленному на рисунке 18, можно сделать вывод о неравномерном использовании ресурсов *Tier1* ЛИТ. Количество выполненных задач в час изменяется примерно от 200 до 1 500, что свидетельствует о постепенной загрузке предоставляемых для обработки 1 500 ядер. Следует отметить, что на кластере *NICA* ЛФВЭ всего было обработано 4 844 задачи, что составляет примерно 19% от общего количества, а на *Tier1* ЛИТ — 20 956 задач (81%). Мониторинг производительности процессоров при выполнении всех *RawToDigit* задач показал, что среднее время выполнения одной задачи составляет примерно 2 500 секунд. Суммарный объем получившихся *digit*-файлов в хранилище данных составил 23 ТБ (без учета зеркалирования данных). График на рисунке 19 отражает скорость передачи данных. Видно, что скорость передачи данных от хранилища данных до вычислительных ресурсов кластера *NICA* ЛФВЭ составляет в среднем 0.5 ГБ/с. В связи с тем, что задачи на вычислительные компоненты *Tier1* ЛИТ поступали неравномерно, аналогичная скорость передачи данных изменяется от 1 ГБ/с до 8 ГБ/с.



Рисунок 17. Количество выполненных задач на кластере *NICA* ЛФВЭ



Рисунок 18. Количество выполненных задач на ресурсах *Tier1* ЛИТ

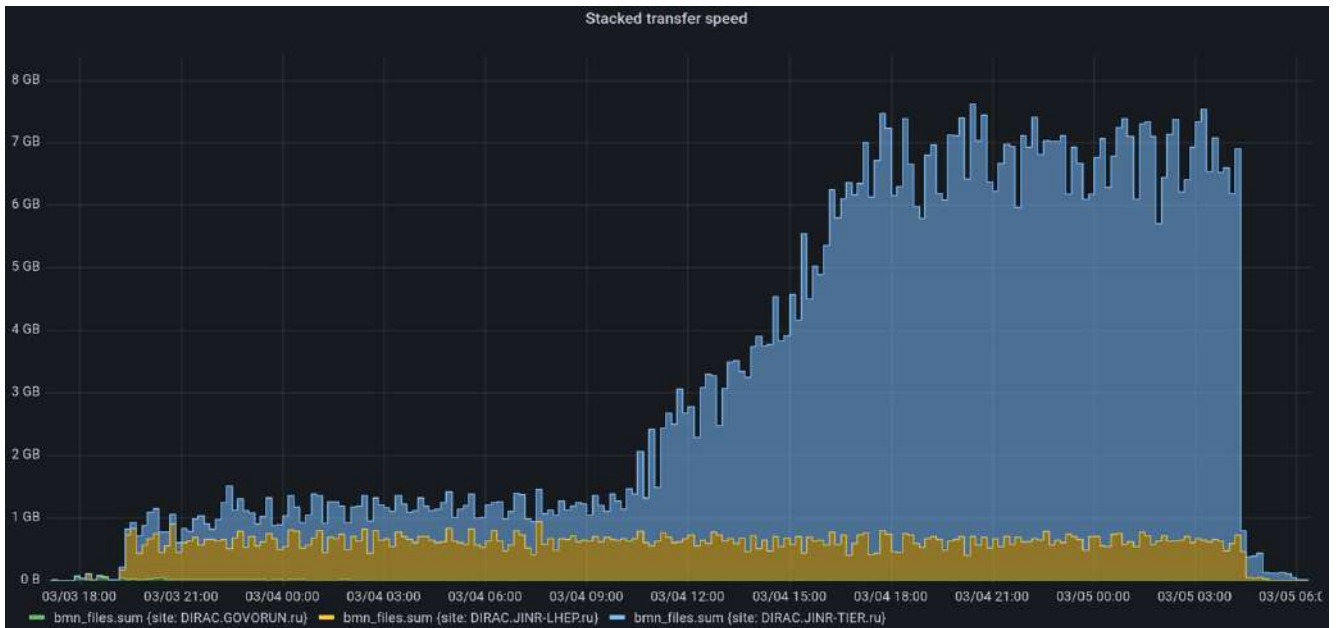


Рисунок 19. Мониторинг скорости передачи данных: желтый – от хранилища данных до вычислительных ресурсов кластера *NICA* ЛФВЭ; голубой – от хранилища данных до ресурсов *Tier1* ЛИТ

Получение реконструированных данных формата *DST* (*DigitToDst* задачи) — преобразование *digit*-файлов в *dst*-файлы — осуществлялось уже на большем количестве вычислительных компонент, а именно: кластере *NICA* ЛФВЭ, *Tier1* ЛИТ, *Tier2* ЛИТ, Суперкомпьютере «Говорун». Аналогично, каждая задача *DigitToDst* обрабатывает 1 *digit*-файл один раз. Размер каждого *dst*-файла составил в среднем 2 000 МБ. Все *DigitToDst* задачи были отправлены на выполнение одновременно. Общее время обработки всех *digit*-файлов составило примерно 73 часа. На рисунках 20-23 изображены графики, отражающие количество выполненных задач на вычислительных компонентах кластера *NICA* ЛФВЭ, *Tier1* ЛИТ, *Tier2* ЛИТ, Суперкомпьютера «Говорун» соответственно за указанный период времени. Можно сделать вывод, что все вычислительные ресурсы загружены неравномерно. На кластере *NICA* ЛФВЭ для запуска *DigitToDst* задач выделено 300 ядер (см. рис. 20), при этом имеется промежуток времени, когда используется меньше ресурсов (от 10 до 200 ядер). На *Tier1* ЛИТ выделено 1 500 ядер, вычислительная компонента постепенно загружается до максимума, а затем количество используемых ресурсов сокращается до 10 ядер (см. рис. 21). Аналогичная ситуация просматривается на *Tier2* ЛИТ, где выделено 1 000 ядер (см. рис. 22). Суперкомпьютер «Говорун», где выделено 500 ядер, используется для запуска задач только в первую половину рассматриваемого промежутка времени (см. рис. 23). Таким образом на кластере *NICA* ЛФВЭ всего было обработано 5 315 задач, что составляет примерно 21% от общего количества, на *Tier1* ЛИТ — 9 289 задач (36%), на *Tier2* ЛИТ — 9 016 задач (35%), на Суперкомпьютере «Говорун» — 2 180 задач (8%). Мониторинг производительности

процессоров при выполнении всех *RawToDigit* задач показал, что среднее время выполнения одной задачи составляет примерно 10 000 секунд. Суммарный объем получившихся *dst*-файлов в хранилище данных составил 53 ТБ.



Рисунок 20. Количество выполненных задач на кластере *NICA* ЛФВЭ

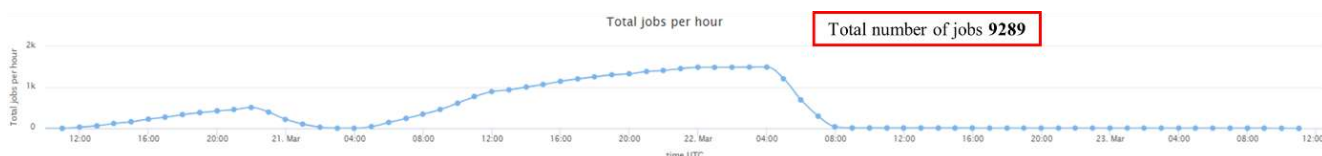


Рисунок 21. Количество выполненных задач на *Tier1* ЛИТ



Рисунок 22. Количество выполненных задач на *Tier2* ЛИТ

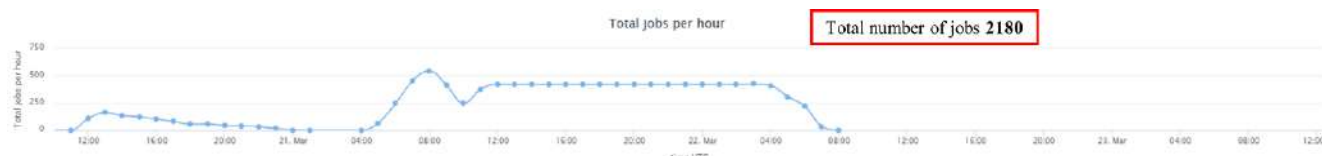


Рисунок 23. Количество выполненных задач на Суперкомпьютер «Говорун»

4.1.3 Моделирование процесса получения и хранения экспериментальных данных *VM@N*

Моделируемая система получения и хранения данных эксперимента *VM@N* представлена на рисунке 24. Исходными данными для запуска программы моделирования являются скорость генерации данных и пропускная способность каналов связи между объектами инфраструктуры. Требуется определить количество ресурсов, которые необходимы для хранения всех поступающих данных на буфере и в хранилище данных *EOS*.

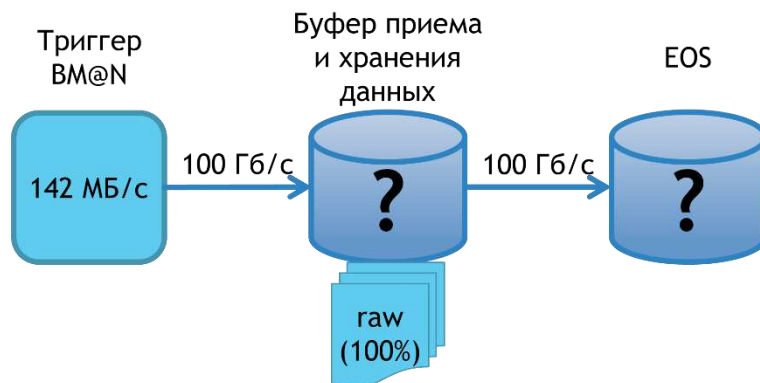


Рисунок 24. Моделируемая система получения и хранения данных эксперимента *VM@N*

Результаты моделирования показали, что объем накопленных за 720 часов данных составит примерно 363 ТБ (см. рис. 25). В хранилище *EOS* будет передан весь объем данных, упакованных в *raw*-файлы (см. рис. 26). Для приема и хранения экспериментальных данных при заданной средней частоте генерации будет достаточно буфера объемом 400 ТБ (без учета зеркалирования данных).

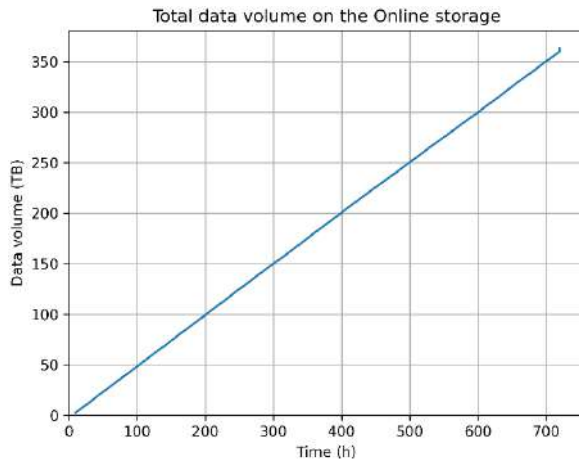


Рисунок 25. Объем данных на буфере приема и хранения данных

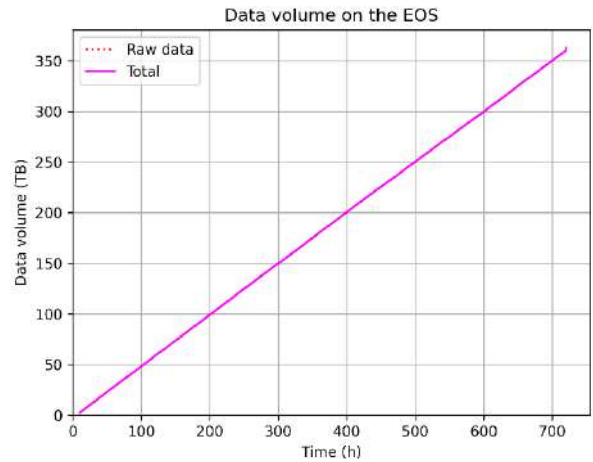


Рисунок 26. Объем *raw*-файлов в хранилище данных *EOS*

4.1.4 Моделирование процесса выполнения задач преобразования экспериментальных данных

Исходя из результатов мониторинга промоделирована система обработки экспериментальных данных, представленная на рисунке 27. Во время моделирования требуется рассчитать загрузку вычислительных ресурсов в процессе выполнения *RawToDigit* задач и скорость передачи данных при следующих условиях. На кластере *NICA* ЛФВЭ выделено 100 ядер, на *Tier1* ЛИТ — 1 500 ядер. Количество выполненных задач в час на *Tier1* ЛИТ изменяется примерно от 200 до 1 500, поэтому при моделировании процесса выполнения *RawToDigit* задач следует установить параметры для изменения количества используемых ядер. Такими параметрами являются вероятность наступления события изменения количества свободных ядер, доступных для выполнения задач, а также диапазон допустимых значений. В данном случае вероятность наступления события увеличения количества ресурсов составляет 0.005, при этом количество ядер изменяется от 200 до 1 500, в среднем на 100 единиц при каждом событии. Для моделирования используются: равномерное распределение времени выполнения задач со средним значением равным 2 500 секунд, средний размер *raw*-файла равный 15 ГБ, средний размер *digit*-файла — 870 МБ. Дополнительно необходимо определить общее время выполнения всех *RawToDigit* задач.

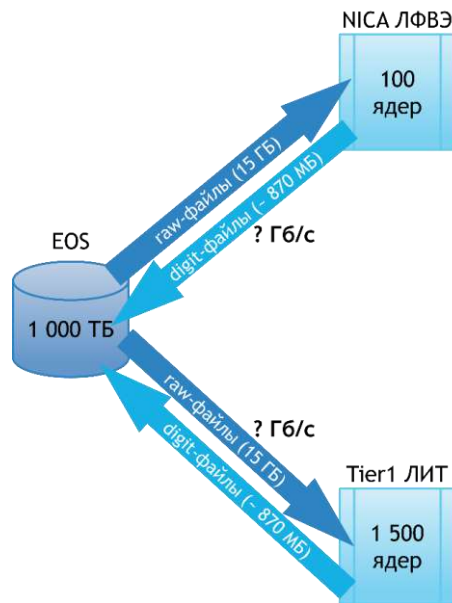


Рисунок 27. Моделируемая вычислительная инфраструктура эксперимента *BM@N* для выполнения задач преобразования экспериментальных данных

Все 25 800 *RawToDigit* задачи генерируются одновременно. Процесс запуска задач во время моделирования контролируют пилоты. Под пилотом понимается алгоритм, который подготавливает задачу к запуску, в том числе анализирует количество свободных ядер на вычислительной компоненте. Если доступны свободные ресурсы, то пилот берет очередную задачу из очереди и отправляет ее на исполнение. Далее задача занимает процессорное ядро и начинает исполняться, то есть обрабатывать входной *raw*-файл из хранилища данных *EOS*. В результате выполнения задачи выходной *digit*-файл также записывается в *EOS*.

Рассмотрим полученные результаты моделирования. На рисунке 28 представлены графики, отражающие общее количество выполненных задач на кластере *NICA* ЛФВЭ и *Tier1* ЛИТ соответственно в каждый момент времени. Можно сделать вывод, что все *RawToDigit* задачи выполняются примерно за 30 часов, при этом на кластере *NICA* ЛФВЭ обработано 3 875 задач, что составляет примерно 15% от общего количества, а на *Tier1* ЛИТ — 21 924 задачи (85%).

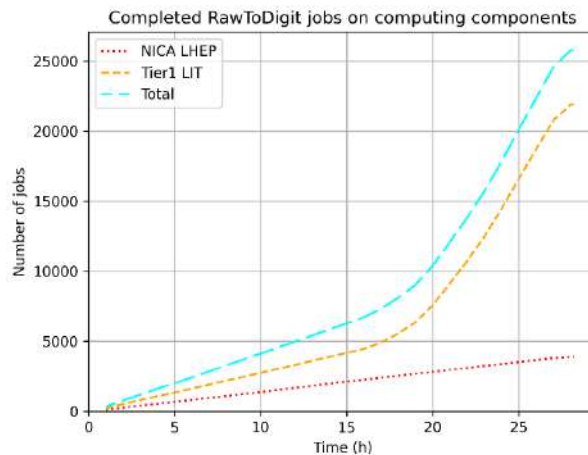


Рисунок 28. Количество выполненных *RawToDigit* задач на вычислительных ресурсах при моделировании

На рисунках 29 и 30 представлены графики, отражающие количество используемых ядер на вычислительных компонентах кластера *NICA* ЛФВЭ и *Tier1* ЛИТ соответственно в каждый момент времени. Делаем вывод, что ресурсы кластера *NICA* ЛФВЭ равномерно загружены на 100%, однако ресурсы *Tier1* ЛИТ используются не в полном объеме, то есть сначала загружены на 15%, а затем постепенно начинают заполняться до 100%.

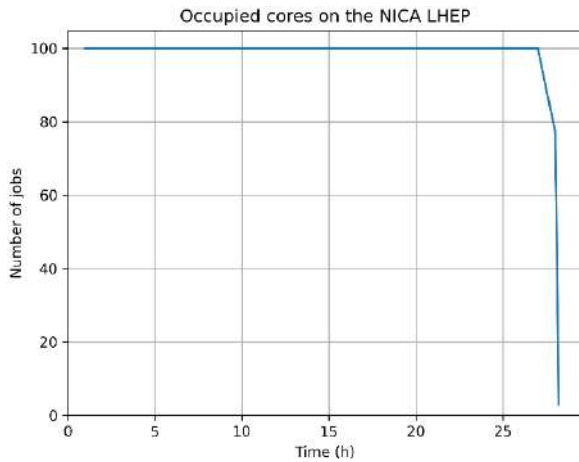


Рисунок 29. Использование ресурсов вычислительной компоненты кластера *NICA* ЛФВЭ

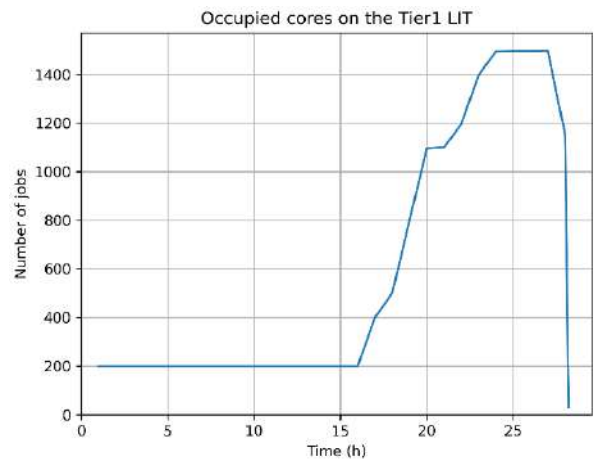


Рисунок 30. Использование ресурсов вычислительной компоненты *Tier1* ЛИТ

Графики, показывающие нагрузку на каналы связи между хранилищем и вычислительными компонентами кластера *NICA* ЛФВЭ и *Tier1* ЛИТ, представлены на рисунках 31 и 32 соответственно. Можно сделать вывод, что средняя скорость передачи данных между *EOS* и кластером *NICA* ЛФВЭ составляет $5 \text{ Гб/с} = 0.63 \text{ ГБ/с}$; между *EOS* и *Tier1* ЛИТ скорость изменяется от $8 \text{ Гб/с} = 1 \text{ ГБ/с}$ до $64 \text{ Гб/с} = 8 \text{ ГБ/с}$. Таким образом, изменение скорости передачи данных между *EOS* и *Tier1* ЛИТ соответствует изменению количества ядер, свободных для запуска задач.

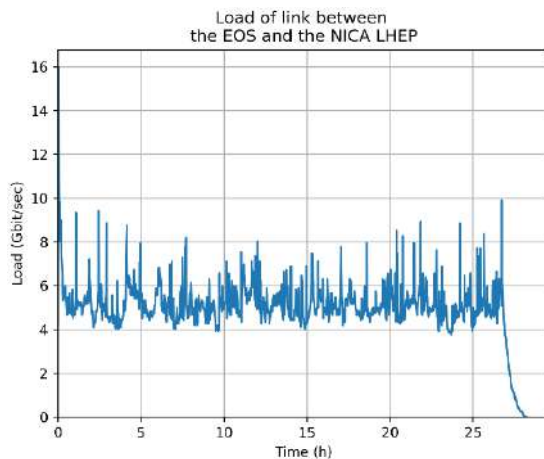


Рисунок 31. Загрузка канала связи между *EOS* и кластером *NICA* ЛФВЭ

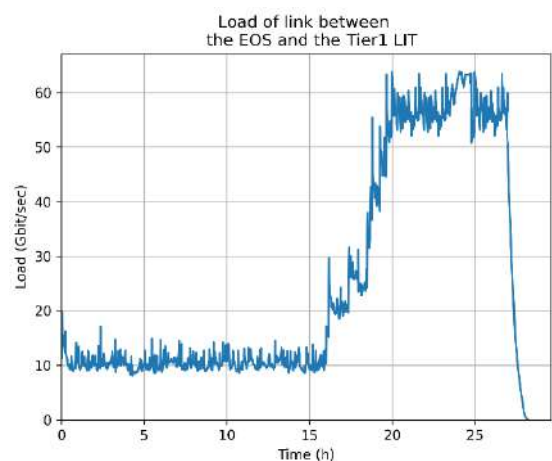


Рисунок 32. Загрузка канала связи между *EOS* и *Tier1* ЛИТ

Стоит отметить, что размер *digit*-файла при моделировании является равномерно распределенной случайной величиной со средним значением 870 МБ (см. рис. 33), а суммарный объем всех *digit*-файлов составил примерно 22 ТБ (см. рис. 34).

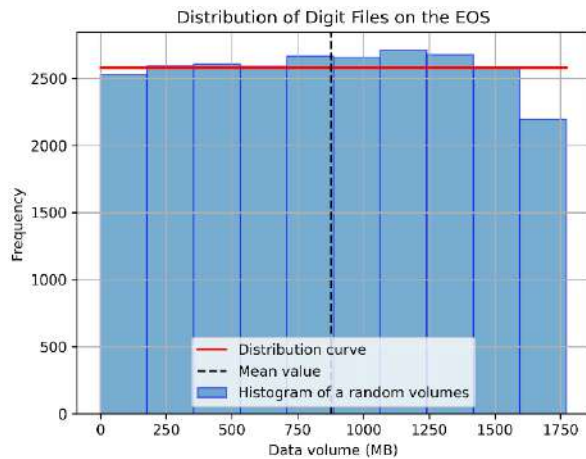


Рисунок 33. Равномерное распределение размеров *digit*-файлов в хранилище EOS

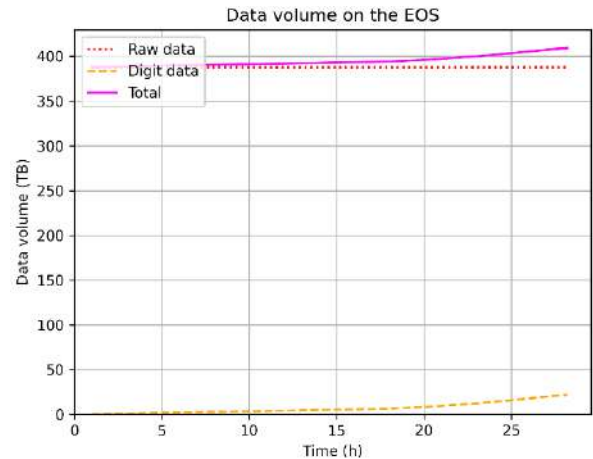


Рисунок 34. Объем данных в хранилище EOS после преобразования экспериментальных данных

4.1.5 Моделирование процесса выполнения задач реконструкции событий

Инфраструктура, которая использовалась для реконструкции событий столкновения частиц по полученным *digit*-данным, представлена на рисунке 35. Во время моделирования требуется рассчитать загрузку вычислительных ресурсов в процессе выполнения *DigitToDst* задач при следующих условиях. На кластере NICA ЛФВЭ выделено 300 ядер, на *Tier1* ЛИТ — 1 500 ядер, на *Tier2* ЛИТ — 1 000 ядер, на Суперкомпьютере «Говорун» — 500 ядер. При этом кластер NICA ЛФВЭ загружен полностью 60% времени, в остальное время используется от 10 до 200 ядер. *Tier1* ЛИТ постепенно загружается до максимума, а затем оставшиеся 40% времени количество используемых ресурсов сокращается до 10 ядер. Аналогичная ситуация просматривается на *Tier2* ЛИТ. Суперкомпьютер «Говорун» используется для запуска задач половину рассматриваемого промежутка времени. Таким образом, при моделировании процесса выполнения *DigitToDst* задач установлены параметры для изменения количества используемых ядер (вероятности и диапазон изменений) для всех вычислительных ресурсов. В данном случае вероятность наступления событий увеличения и уменьшения количества ресурсов составляет 0.001, при этом количество ядер на кластере NICA ЛФВЭ изменяется от 10 до 300, на *Tier1* ЛИТ — от 10 до 1 500, на *Tier2* ЛИТ — от 10 до 1 000, на Суперкомпьютере «Говорун» — от 50 до 500. Для моделирования используются: равномерное распределение времени выполнения задач со средним значением равным 10 000 секунд, средний размер *digit*-файла равный 870 МБ, средний размер *dst*-файла — 2 000 МБ. Дополнительно необходимо определить общее время выполнения всех *DigitToDst* задач. Все 25 800 *DigitToDst* задачи генерируются одновременно. Процесс запуска *DigitToDst* задач во время моделирования аналогичен процессу запуска *RawToDigit* задач.

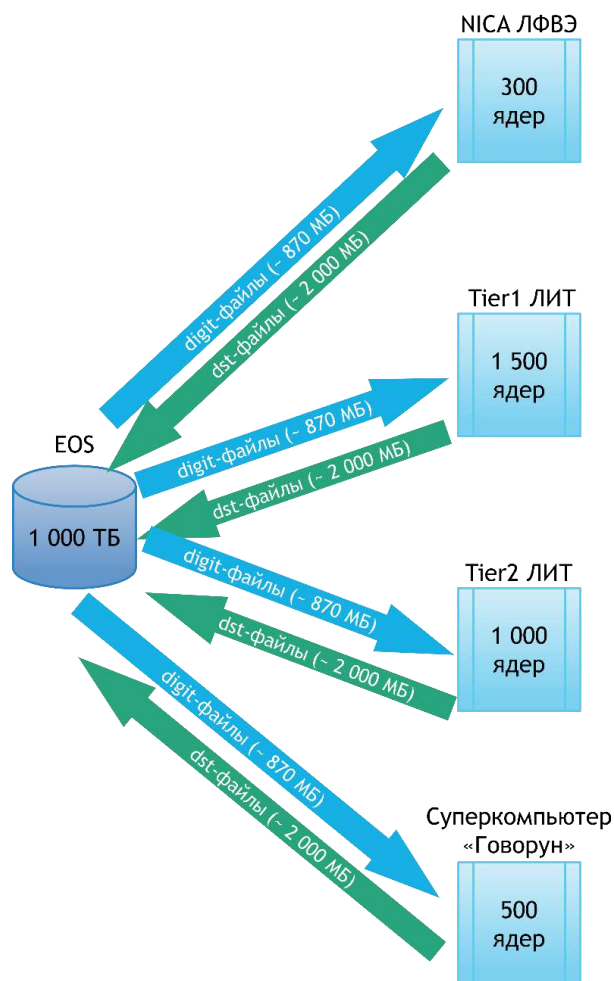


Рисунок 35. Моделируемая вычислительная инфраструктура эксперимента *VM@N* для выполнения задач реконструкции событий

Рассмотрим полученные результаты моделирования. На рисунке 36 представлены графики, отражающие общее количество выполненных задач на кластере *NICA* ЛФВЭ, *Tier1* ЛИТ, *Tier2* ЛИТ, Суперкомпьютере в каждый момент времени. Делаем вывод, что все *DigitToDst* задачи выполняются примерно за 80 часов. При этом на кластере *NICA* ЛФВЭ обработано 5 906 задач, что составляет примерно 24% от общего количества, на *Tier1* ЛИТ — 8 872 задачи (34%), на *Tier2* ЛИТ — 8 598 задач (33%), на Суперкомпьютере — 2 424 задач (9%).

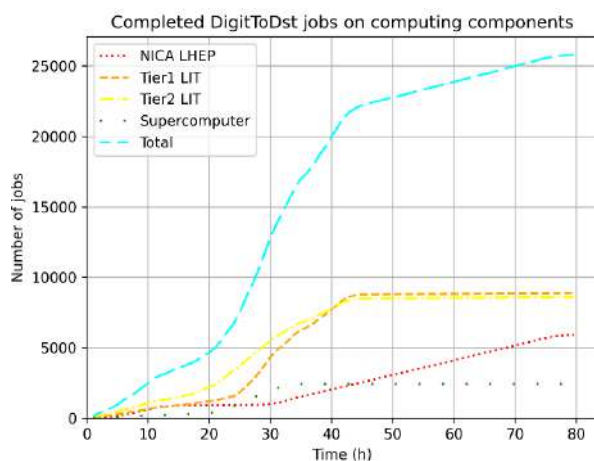


Рисунок 36. Количество выполненных *DigitToDst* задач на вычислительных ресурсах при моделировании

На рисунках 37-40 представлены графики, показывающие количество используемых ядер на вычислительных компонентах кластера *NICA* ЛФВЭ, *Tier1* ЛИТ, *Tier2* ЛИТ, Суперкомпьютера соответственно в каждый момент времени. Делаем вывод, что ресурсы кластера *NICA* ЛФВЭ равномерно загружены полностью в течении 60% времени, в остальное время используется от 10 до 200 ядер. Ресурсы *Tier1* ЛИТ и *Tier2* ЛИТ постепенно загружаются до максимума, а 40% времени используется в среднем 10 ядер. Суперкомпьютер «Говорун» выполняет запуск задач только в первую половину рассматриваемого промежутка времени, при этом сначала используется 10% ресурсов, а затем прослеживается загрузка до 100%.

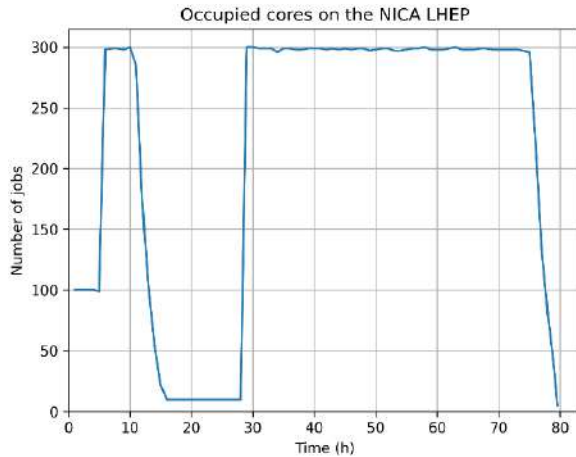


Рисунок 37. Использование ресурсов вычислительной компоненты кластера *NICA* ЛФВЭ

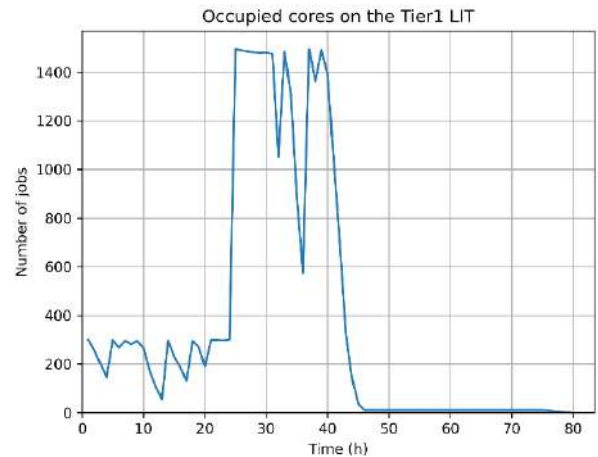


Рисунок 38. Использование ресурсов вычислительной компоненты *Tier1* ЛИТ

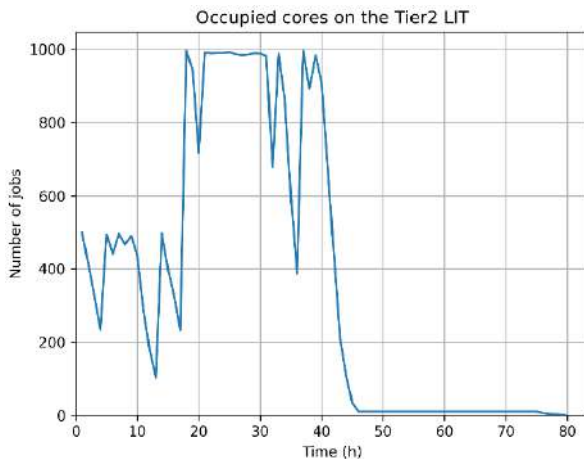


Рисунок 39. Использование ресурсов вычислительной компоненты *Tier2* ЛИТ

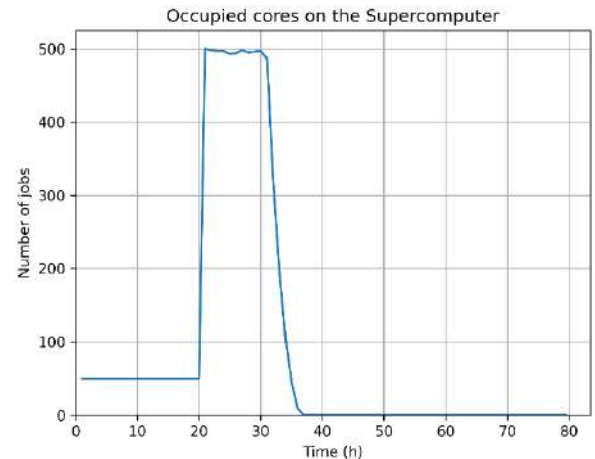


Рисунок 40. Использование ресурсов вычислительной компоненты Суперкомпьютер «Говорун»

Стоит отметить, что размер *dst*-файла при моделировании является равномерно распределенной случайной величиной со средним значением 2 000 МБ (см. рис. 41), а суммарный объем всех *dst*-файлов составил примерно 52 ТБ (см. рис. 42).

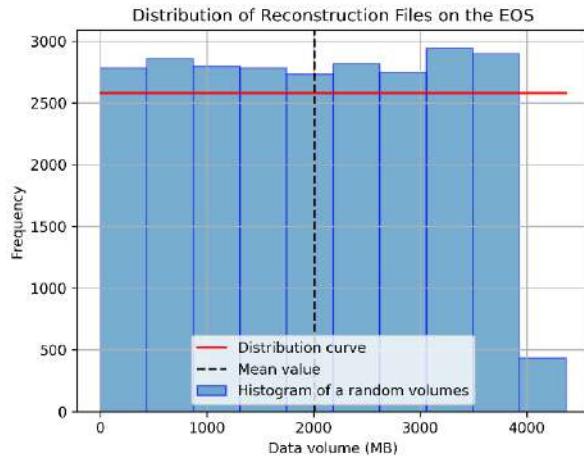


Рисунок 41. Равномерное распределение размеров *digit*-файлов в хранилище *EOS*

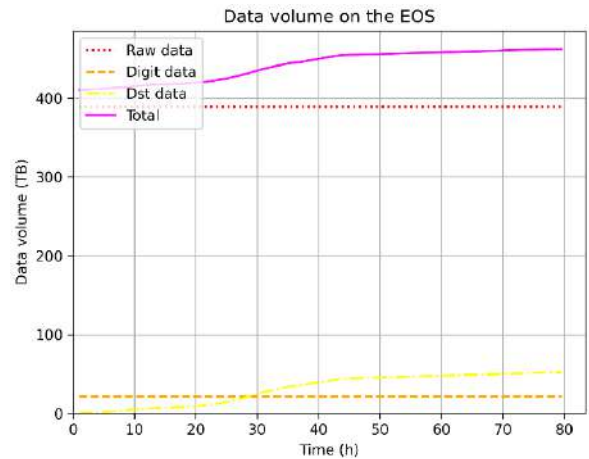


Рисунок 42. Объем данных в хранилище *EOS* после получения данных реконструкции

4.1.6 Выводы по результатам верификации

Верификация программы моделирования, которая является ядром СПО по созданию цифровых двойников РЦОД, проведена на примере вычислительной инфраструктуры эксперимента *VM@N* проекта *NICA*. Рассматриваемая вычислительная инфраструктура использовалась для получения, хранения и обработки данных сеанса эксперимента, который проходил с декабря 2022 года по февраль 2023 года. В качестве исходных данных для моделирования использовалась статистика, полученная по результатам мониторинга вычислительной инфраструктуры эксперимента с помощью программного обеспечения *DIRAC Interware*.

В процессе моделирования определены такие показатели, как количество ресурсов, необходимых для хранения поступающих данных, загрузка вычислительных ресурсов при преобразовании экспериментальных данных и получении по ним реконструированных данных событий, общее время выполнения задач, скорость передачи данных между объектами инфраструктуры. В таблице 1 приведены результаты мониторинга и моделирования для более удобного сравнения.

Результаты верификации доказали корректную работу программы моделирования. Адекватность оценивалась по нескольким показателям. Стоит отметить, что отклонения результатов ЦД от среднего значения, полученного по результатам мониторинга, **не превысят трех средне-квадратических отклонений** статистических данных мониторинга. Такое значение точности достаточно для дальнейшего использования программы моделирования в составе СПО для создания цифровых двойников РЦОД, которые будут применяться для решения задач проектирования и развития вычислительной инфраструктуры научных экспериментов класса «мегасайенс».

Программа моделирования ранее применялась также и для других конфигураций вычислительной инфраструктуры *VM@N*. Некоторые результаты применения опубликованы [75, 76, 77].

Таблица 1. Сравнение результатов мониторинга и моделирования

	Мониторинг	Моделирование	Средне-квадратическое отклонение (σ)
Объем экспериментальных данных, накопленных за 720 часов работы установки (ТБ)	379	363	10
Общее время выполнения всех <i>RawToDigit</i> задач (ч.)	36	30	5
Количество выполненных задач на кластере <i>NICA</i> ЛФВЭ / % от общего числа задач	4 844 / 19%	3 875 / 15%	500
Количество выполненных задач на <i>Tier1</i> ЛИТ / % от общего числа задач	20 956 / 81%	21 924 / 85%	500
Скорость передачи данных между <i>EOS</i> и кластером <i>NICA</i> ЛФВЭ (ГБ/с)	0.5	0.63	0,1
Скорость передачи данных между <i>EOS</i> и <i>Tier1</i> ЛИТ (ГБ/с)	от 1 до 8	от 1 до 8	1
Суммарный объем <i>digit</i> -файлов (ТБ)	23	22	2
Общее время выполнения всех <i>DigitToDst</i> задач (ч.)	73	80	5
Количество выполненных задач на кластере <i>NICA</i> ЛФВЭ / % от общего числа задач	5 315 / 21%	5 906 / 24%	500
Количество выполненных задач на <i>Tier1</i> ЛИТ / % от общего числа задач	9 289 / 36%	8 872 / 34%	500
Количество выполненных задач на <i>Tier2</i> ЛИТ / % от общего числа задач	9 016 / 35%	8 598 / 33%	500
Количество выполненных задач на Суперкомпьютере / % от общего числа задач	2 180 / 8%	2 424 / 9%	500
Суммарный объем <i>dst</i> -файлов (ТБ)	53	52	2

4.2. Применение специального программного обеспечения для построения цифрового двойника вычислительной инфраструктуры эксперимента $VM@N$ комплекса NICA

Результаты первого физического сеанса эксперимента $VM@N$ комплекса NICA, который проходил с декабря 2022 года по февраль 2023 года, показали, что в вычислительную инфраструктуру эксперимента необходимо внести изменения. В связи с этим требуется построить ЦД с несколькими видами архитектуры системы сбора, хранения и обработки данных рассматриваемого эксперимента, а также с различными параметрами входящего в ее состав оборудования.

Главная цель построения ЦД — выбор конфигурации оборудования, которая обеспечит хранение и обработку данных с учетом планируемых параметров потоков данных будущих сеансов эксперимента $VM@N$. Наиболее предпочтительным критерием выбора является время обработки всех данных, которое должно быть минимальным из всех возможных альтернатив.

Распределенная система сбора, хранения и обработки данных эксперимента $VM@N$ включает в себя следующие компоненты:

1. источник данных (*Trigger*) — средняя скорость генерации данных 140 МБ/с;
2. промежуточное хранилище данных (*Buffer*) — хранение «сырых» необработанных экспериментальных данных и запись их в raw-файлы, средний объем которых составляет 15 ГБ;
3. постоянной хранилище данных на распределенной файловой системе (*EOS*) — 1 000 ТБ для хранения всех файлов, которые содержат не только экспериментальные данные, но и данные смоделированных событий, а также результаты каждого этапа обработки перечисленных типов данных;
4. вычислительный кластер обработки экспериментальных данных в процессе работы установки (*Online farm*) — для обработки данных по завершении сеанса может быть использовано 1 000 ядер;
5. вычислительный кластер ЛФВЭ ОИЯИ (*NICA LHEP*) — для обработки данных может быть использовано 1 000 ядер;
6. ресурсный центр первого уровня МИВК ЛИТ ОИЯИ (*Tier1 LIT*) — для обработки данных может быть использовано 750 ядер;
7. ресурсный центр второго уровня МИВК ЛИТ ОИЯИ (*Tier2 LIT*) — для обработки данных может быть использовано 500 ядер;
8. суперкомпьютер «Говорун» ЛИТ ОИЯИ (*Govorun*) — для обработки данных может быть использовано 200 ядер.

Процесс построения инфраструктуры РЦОД представлен на рисунке 43.

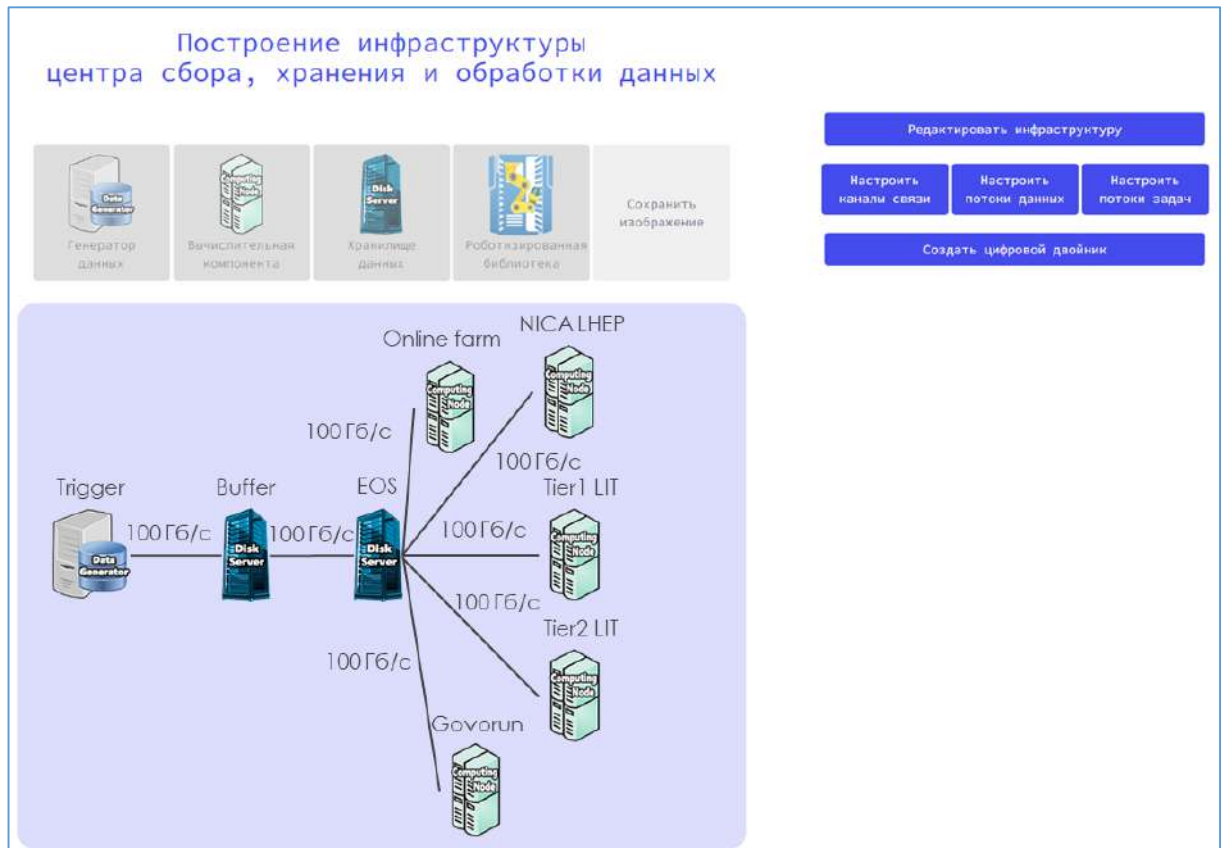


Рисунок 43. Распределенная система сбора, хранения и обработки данных эксперимента *BM@N*

Следует отметить, что пропускная способность каналов связи составляет 100 Гбит/с. Рассматривается случай работы РЦОД при бесперебойном функционировании оборудования после окончания сеанса эксперимента, т.е. в период набора данных вычислительные задачи не запускались.

Процесс обработки экспериментальных данных включает в себя два этапа:

1. преобразование данных из *raw*-файлов (25 800 шт.) в формат *digit*, средний размер результирующего *digit*-файла составляет 870 МБ — *RawToDigit* задачи, среднее время выполнения одной задачи составляет 2 500 с;
2. преобразование данных формата *digit* в данные реконструированных событий столкновения частиц формата *DST*, средний размер результирующего *dst*-файла составляет 2 000 МБ — *DigitToDst* задачи, среднее время выполнения одной задачи составляет 86 400 с.

Процесс обработки сгенерированных модельных данных включает в себя два этапа:

1. получение из сгенерированных данных в *generation*-файлах, или *gen*-файлах (60 000 шт.), средний размер которых составляет 4 МБ, смоделированных событий после столкновений, средний размер результирующего *sim*-файла составляет 300 МБ — *GenToSim* задачи, среднее время выполнения одной задачи составляет 5 400 с;

2. преобразование данных формата *sim* в данные реконструированных событий столкновения частиц формата *DST*, средний размер результирующего *dst*-файла составляет 300 МБ — *SimToDst* задачи, среднее время выполнения одной задачи составляет 5 400 с.

Характеристики, указанные при перечислении этапов обработки данных, используются для настройки параметров потоков данных и потоков задач. Для анализа загруженности ресурсов хранения данных и вычислительных компонент при обработке только экспериментальных данных при разном количестве свободных ядер, а также при параллельной обработке экспериментальных и смоделированных данных добавлены соответствующие вычислительные эксперименты на ЦД.

При первом вычислительном эксперименте на ЦД рассчитывается требуемый объем данных в хранилище *EOS* и анализируется загруженность вычислительных ресурсов в процессе обработки *raw*-данных, т.е. при выполнении *RawToDigit* и *DigitToDst* задач. При этом задачи *RawToDigit* выполняются только на ресурсах *Tier1 LIT*, а задачи *DigitToDst* — на всех имеющихся вычислительных компонентах. Рассмотрено две конфигурации вычислительного оборудования:

а. *Online farm* — 0 ядер, *NICA LHEP* — 250 ядер, *Tier1 LIT* — 750 ядер, *Tier2 LIT* — 500 ядер, *Govorun* — 200 ядер;

б. *Online farm* — 1 000 ядер, *NICA LHEP* — 1 000 ядер, *Tier1 LIT* — 750 ядер, *Tier2 LIT* — 500 ядер, *Govorun* — 200 ядер.

Второй вычислительный эксперимент на ЦД посвящен поиску требуемого объема данных в хранилище *EOS* и анализу загруженности вычислительных ресурсов. Но в этом случае рассматривается как процесс обработки экспериментальных данных (выполнение *RawToDigit* и *DigitToDst* задач), так и процесс обработки модельных данных (выполнение *GenToSim* и *SimToDst* задач). По аналогии с первым вычислительным экспериментом, задачи *RawToDigit* выполняются только на ресурсах *Tier1 LIT*, задачи *DigitToDst* — на всех имеющихся вычислительных компонентах. Задачи *GenToSim* и *SimToDst* выполняются только на ресурсах *Online farm* и *NICA LHEP*. При этом используется только базовая конфигурация оборудования, которая задана на этапе построения инфраструктуру перед созданием ЦД (*Online farm* — 1 000 ядер, *NICA LHEP* — 1 000 ядер, *Tier1 LIT* — 750 ядер, *Tier2 LIT* — 500 ядер, *Govorun* — 200 ядер).

ЦД для обоих вычислительных экспериментов запущены одновременно. Через 20 часов получены все результаты. Рассмотрим результаты более подробно на изображениях, которые экспортированы с веб-сервиса.

Результаты первого вычислительного эксперимента показали, что с конфигурацией оборудования, представленной на рисунке 44, все экспериментальные данные будут полностью обработаны примерно за 432 часа (см. рис. 45), что составляет 18 дней. При этом задачи *RawToDigit*

выполнятся за 60 часов (см. рис. 46). График выполнения задач *DigitToDst* представлен на рисунке 47. Графики на рисунках 48-51 отражают полную загрузку вычислительных компонент. Объем данных в хранилище *EOS* по завершении обработки экспериментальных данных составляет 460 ТБ (см. рис. 52).

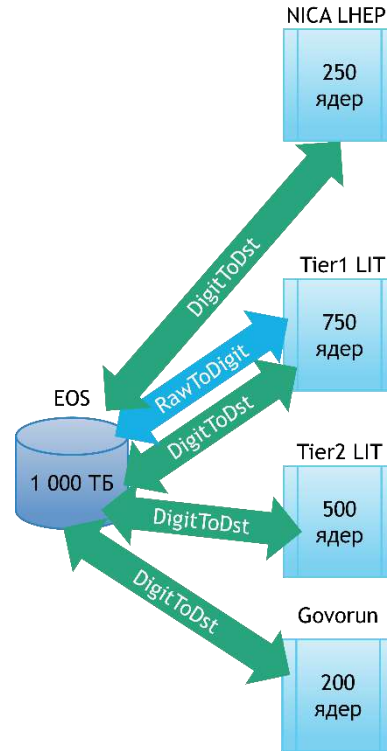


Рисунок 44. Схема конфигурации (а) первого вычислительного эксперимента на ЦД

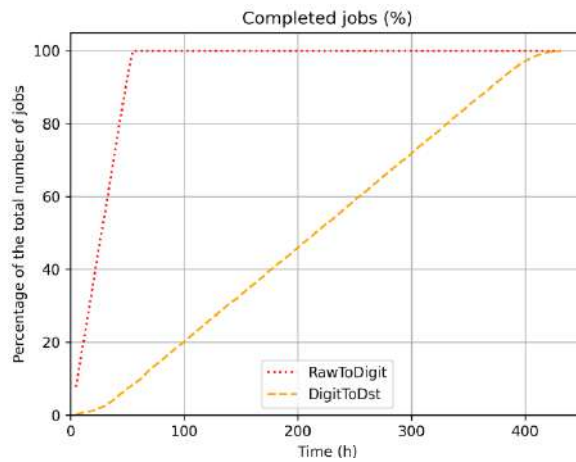


Рисунок 45. Количество выполненных задач обработки экспериментальных данных на вычислительных ресурсах

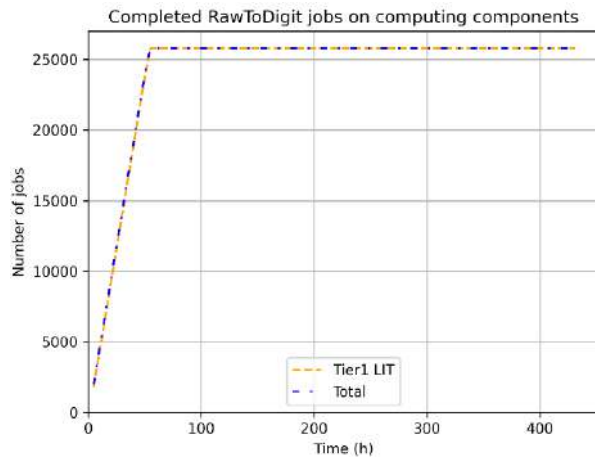


Рисунок 46. Количество выполненных *RawToDigit* задач на вычислительных ресурсах

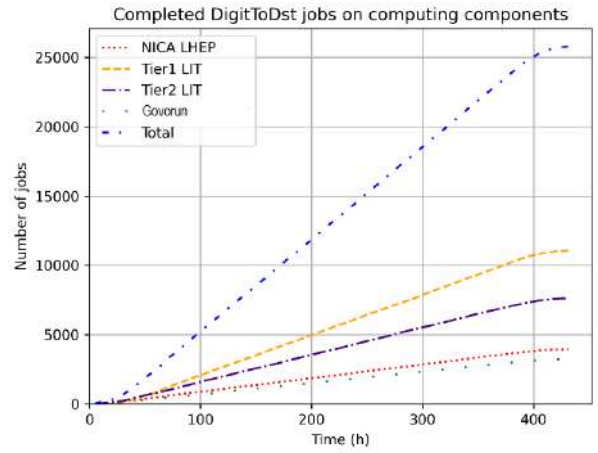


Рисунок 47. Количество выполненных *DigitToDst* задач на вычислительных ресурсах

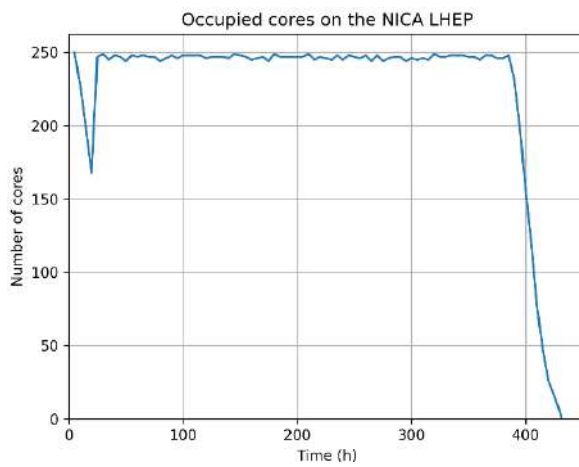


Рисунок 48. Использование ресурсов вычислительной компоненты *NICA LHEP*

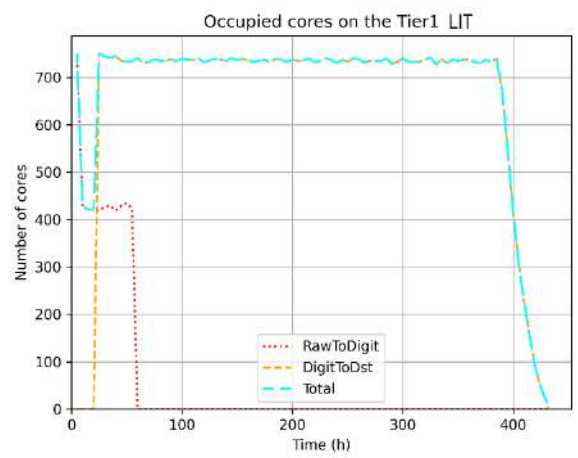


Рисунок 49. Использование ресурсов вычислительной компоненты *Tier1 LIT*

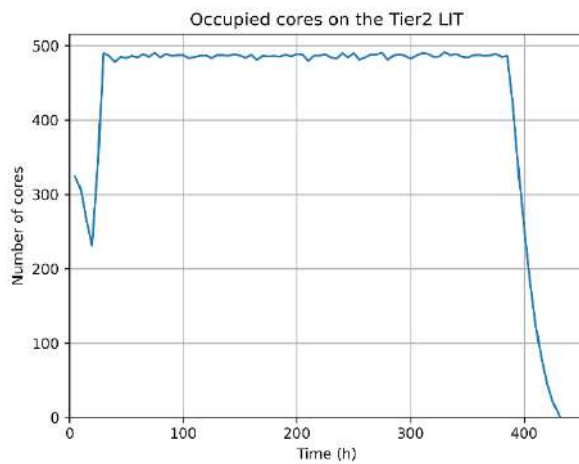


Рисунок 50. Использование ресурсов вычислительной компоненты *Tier2 LIT*

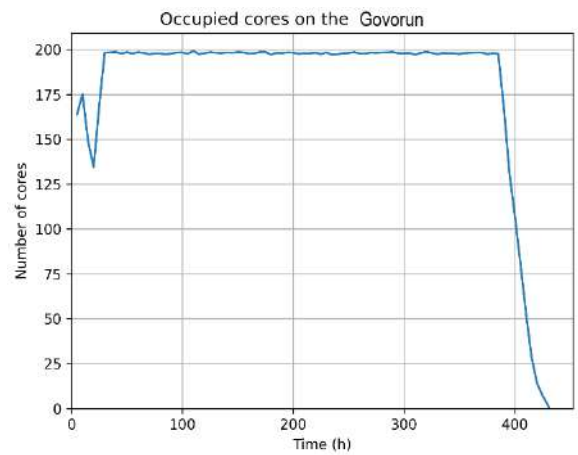


Рисунок 51. Использование ресурсов вычислительной компоненты *Govorun*

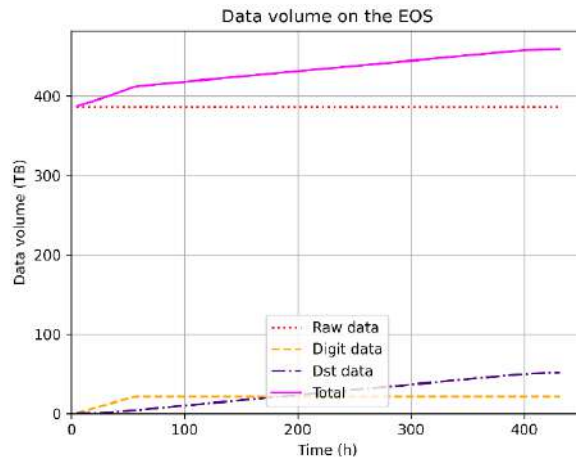


Рисунок 52. Объем данных в хранилище *EOS* после обработки экспериментальных данных

Результаты работы ЦД с обновленной конфигурацией оборудования (см. рис. 53) показали, что все экспериментальные данные будут полностью обработаны примерно за 240 часов (см. рис. 54), что составляет 10 дней. При этом задачи *RawToDigit* выполнятся за 28 часов (см. рис. 55). График выполнения задач *DigitToDst* представлен на рисунке 56. Графики на рисунках 57-61 отражают полную загрузку вычислительных компонент. Стоит отметить, что данная конфигурация позволяет обработать экспериментальные данные почти в 2 раза быстрее. Увеличение количества вычислительных ресурсов не влияет на накопленный объем данных в хранилище *EOS*.

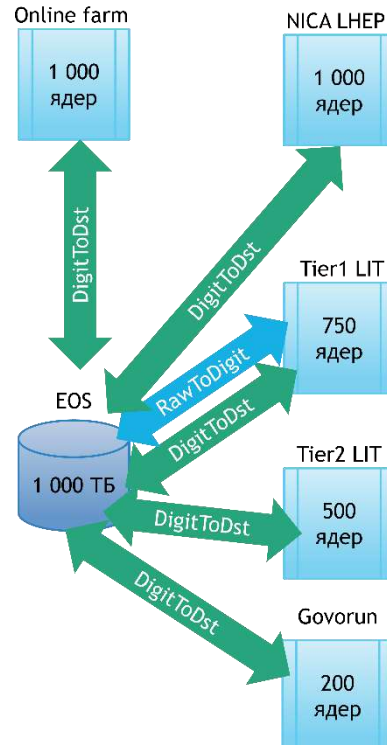


Рисунок 53. Схема конфигурации (б) первого вычислительного эксперимента на ЦД

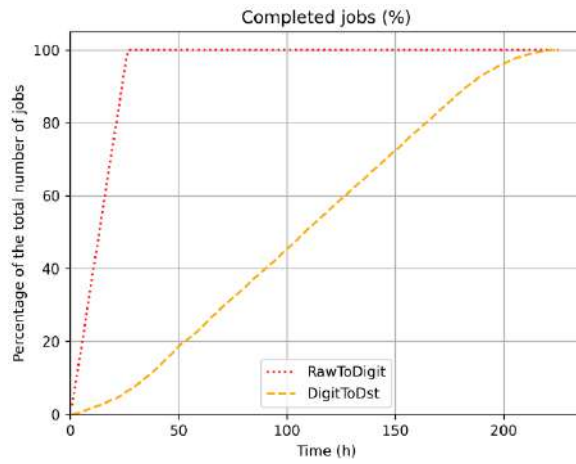


Рисунок 54. Количество выполненных задач обработки экспериментальных данных на вычислительных ресурсах

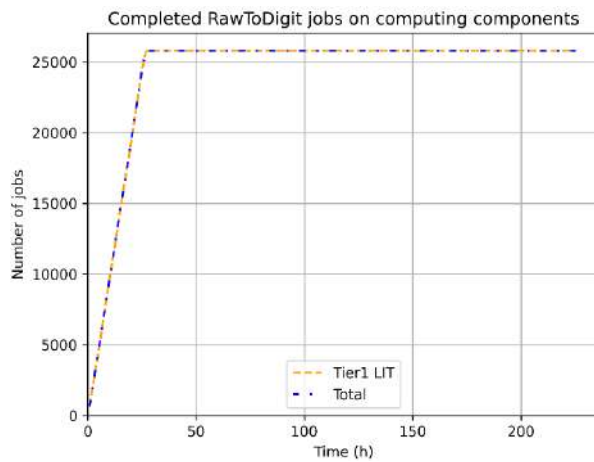


Рисунок 55. Количество выполненных *RawToDigit* задач на вычислительных ресурсах

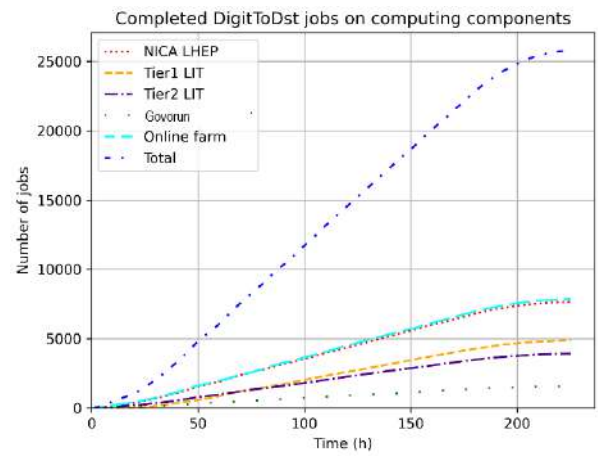


Рисунок 56. Количество выполненных *DigitToDst* задач на вычислительных ресурсах

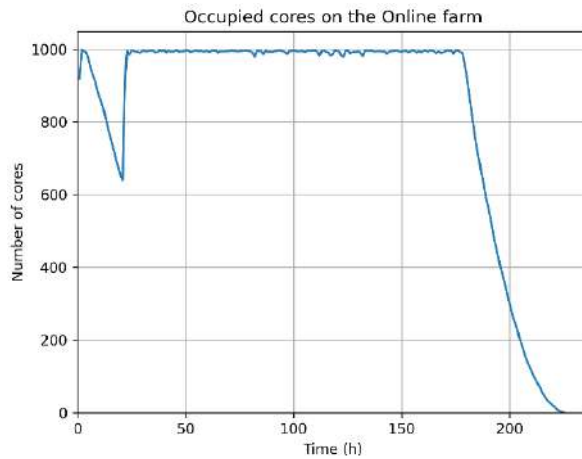


Рисунок 57. Использование ресурсов вычислительной компоненты *Online farm*

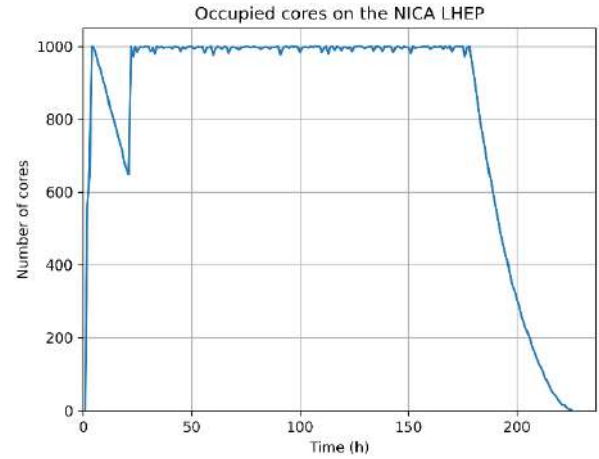


Рисунок 58. Использование ресурсов вычислительной компоненты *NICA LHEP*

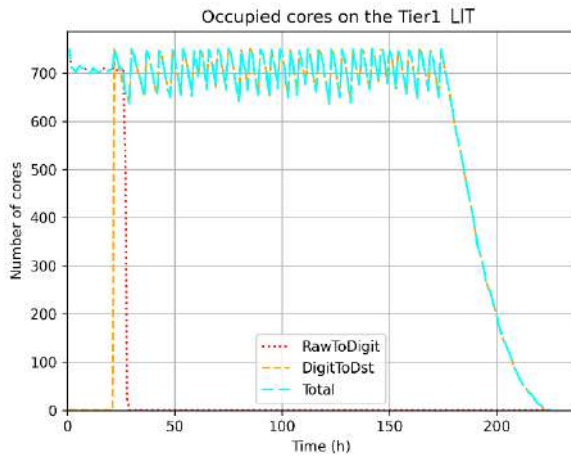


Рисунок 59. Использование ресурсов вычислительной компоненты *Tier1 LIT*

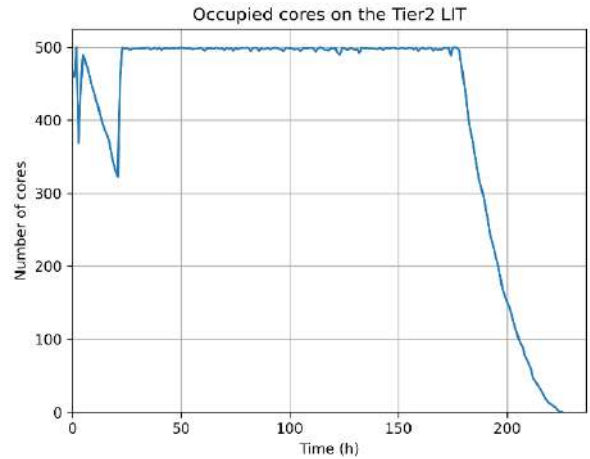


Рисунок 60. Использование ресурсов вычислительной компоненты *Tier2 LIT*

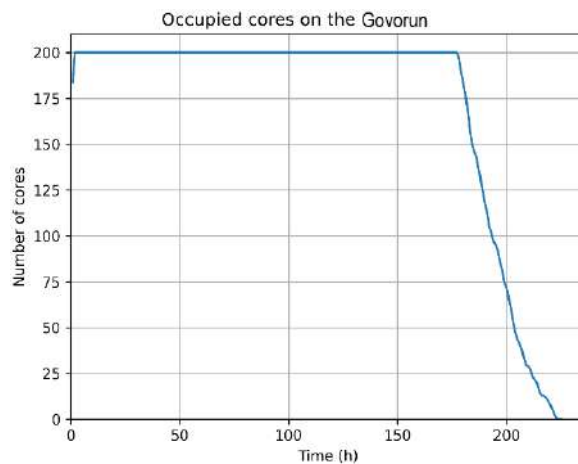


Рисунок 61. Использование ресурсов вычислительной компоненты *Govorun*

Результаты второго вычислительного эксперимента на ЦД по обработке всех типов данных на вычислительной инфраструктуре, представленной на рисунке 62, показали, что экспериментальные данные будут полностью обработаны примерно за 367 часов (15 дней), а смоделированные данные — за 130 часов (6 дней) (см. рис. 63). Процесс выполнения *RawToDigit* задач аналогичен такому же процессу в первом вычислительном эксперименте с конфигурацией (б). Графики выполнения *DigitToDst*, *GenToSim* и *SimToDst* задач представлены на рисунках 64-66 соответственно. Графики на рисунках 67-71 отражают полную загрузку вычислительных компонент. Стоит отметить, несмотря на то, что на обработку экспериментальных данных затрачивается немного больше времени в сравнении с конфигурацией (б) первого вычислительного эксперимента, на обработку смоделированных данных требуется только 6 дней. Объем данных в хранилище *EOS* по завершении обработки экспериментальных и смоделированных данных составляет 500 ТБ (см. рис. 72).

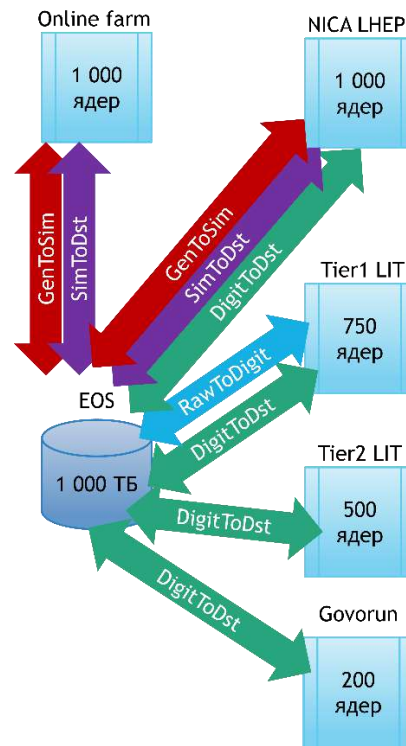


Рисунок 62. Схема конфигурации второго вычислительного эксперимента на ЦД

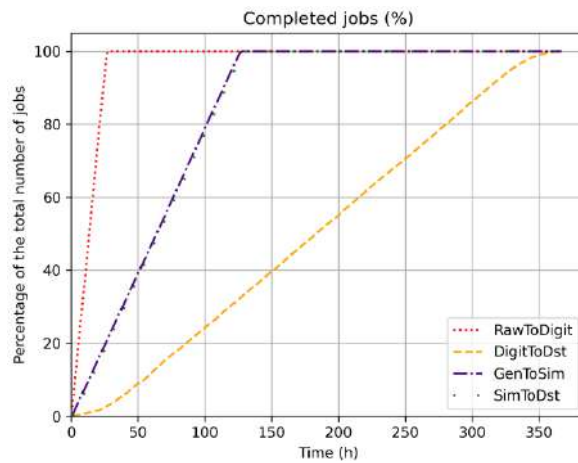


Рисунок 63. Количество выполненных задач обработки экспериментальных и смоделированных данных на вычислительных ресурсах

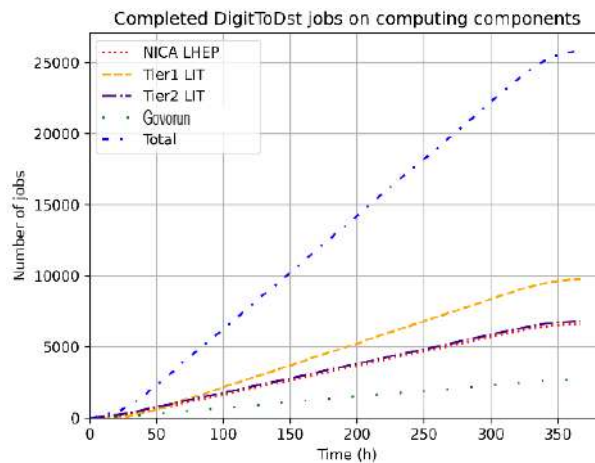


Рисунок 64. Количество выполненных *DigitToDst* задач на вычислительных ресурсах

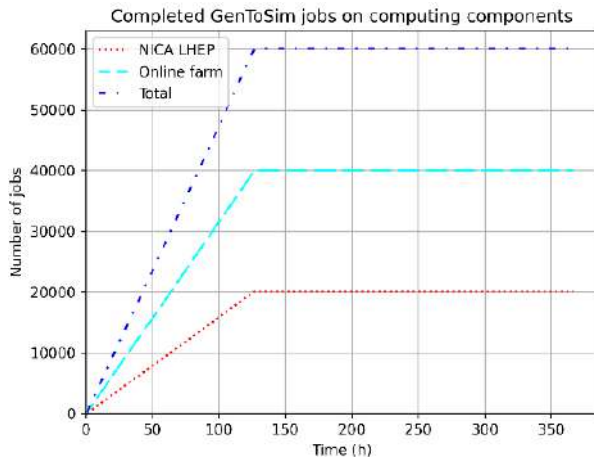


Рисунок 65. Количество выполненных *GenToSim* задач на вычислительных ресурсах

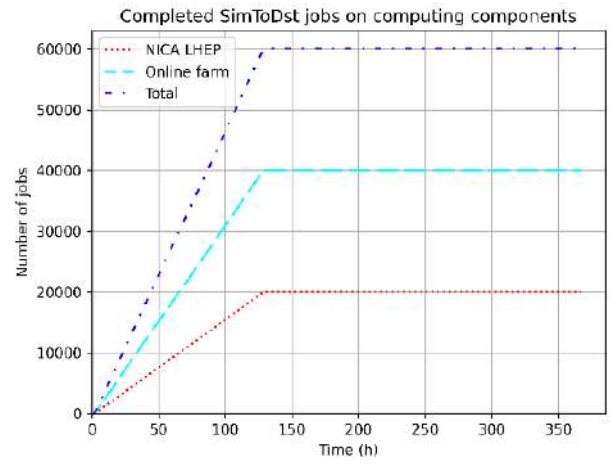


Рисунок 66. Количество выполненных *SimToDst* задач на вычислительных ресурсах

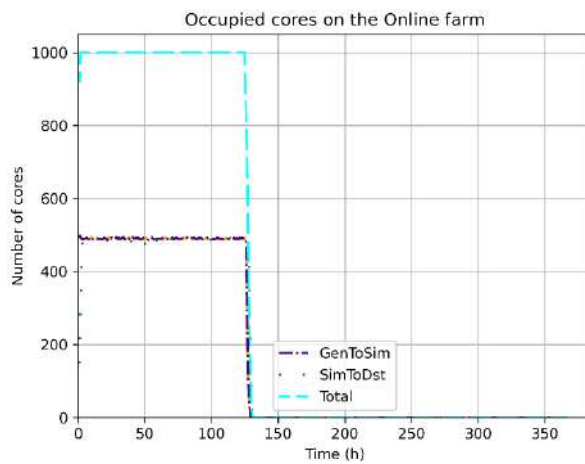


Рисунок 67. Использование ресурсов вычислительной компоненты *Online farm*

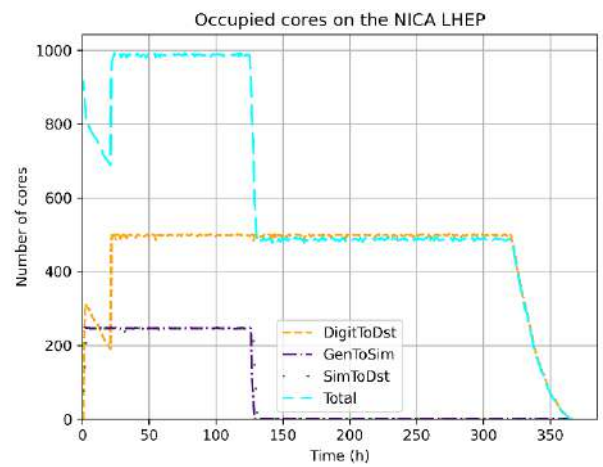


Рисунок 68. Использование ресурсов вычислительной компоненты *NICA LHEP*

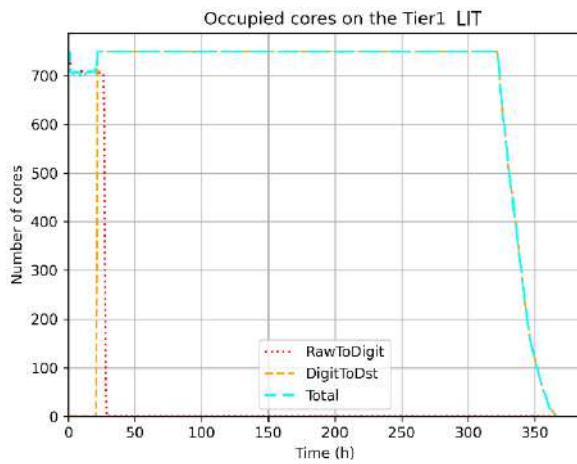


Рисунок 69. Использование ресурсов вычислительной компоненты *Tier1 LIT*

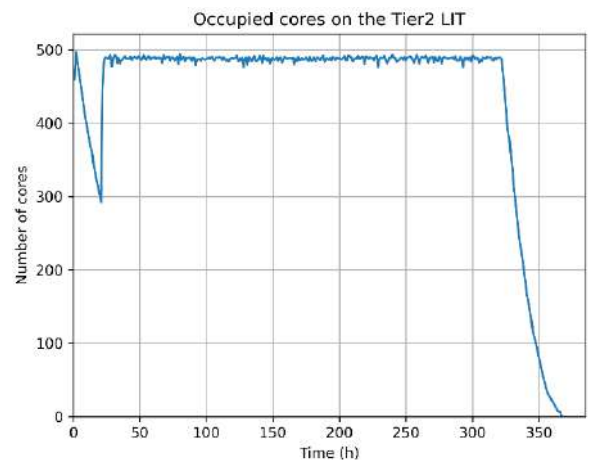


Рисунок 70. Использование ресурсов вычислительной компоненты *Tier2 LIT*

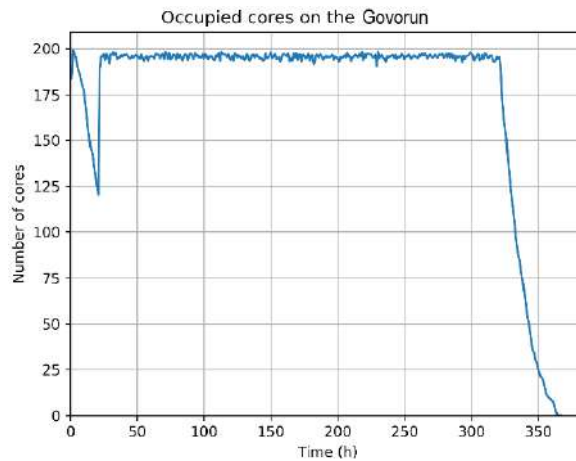


Рисунок 71. Использование ресурсов вычислительной компоненты *Govorun*

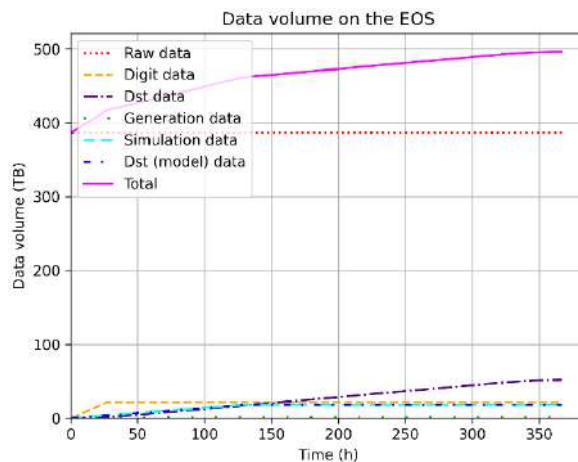


Рисунок 72. Объем данных в хранилище *EOS* после обработки экспериментальных и смоделированных данных

В таблице 2 приведены результаты проведенных вычислительных экспериментов на ЦД для более удобного сравнения. Построение ЦД выполнялось для выбора конфигурации оборудования, которая обеспечит хранение и обработку данных с учетом планируемых параметров потоков данных будущих сеансов эксперимента *BM@N*. Наиболее предпочтительный критерий выбора — время обработки всех данных, которое должно быть наименьшим.

По результатам работы ЦД сделан вывод о наилучшей конфигурации оборудования, которая позволит за наименьшее количество времени обработать все экспериментальные (367 часов) и смоделированные данные (130 часов) с использованием всех имеющихся ресурсов. В состав конфигурации должны входить следующие компоненты: *Online farm* — 1 000 ядер, *NICA LHEP* — 1 000 ядер, *Tier1 LIT* — 750 ядер, *Tier2 LIT* — 500 ядер, *Govorun* — 200 ядер, *EOS* — не менее 500 ТБ. При этом целесообразно запускать задачи *RawToDigit* только на ресурсах *Tier1 LIT*, задачи *DigitToDst* — на всех имеющихся вычислительных компонентах, задачи *GenToSim* и *SimToDst* — только на ресурсах *Online farm* и *NICA LHEP*. Выбранная альтернатива с учетом полной загрузки имеющихся ресурсов обеспечит **ускорение в 2 раза** процесса преобразования *raw*-данных. **Предложенная тактика управления потоком задач** разных типов при распределении их на вычислительные ресурсы позволит за то же время (3 недели после окончания эксперимента)

обработать не только экспериментальные, но и модельные данные. Следовательно, достигнуто более эффективное использование ресурсов, выбраны надежные сценарии масштабирования и управления потоками данных и задач.

Таблица 2. Сравнение результатов работы ЦД при разных конфигурациях оборудования вычислительной инфраструктуры эксперимента *VM@N*

	Конфигурации первого вычислительного эксперимента		Конфигурация второго вычислительного эксперимента
	а	б	
Преобразование всех <i>raw</i> -данных в формат <i>digit</i> (<i>RawToDigit</i> задачи)	60 часов (2,5 дня)	28 часов (1 день)	28 часов (1 день)
Время полной обработки всех экспериментальных данных	432 часа (18 дней)	240 часов (10 дней)	367 часов (15 дней)
Время полной обработки всех модельных данных	--	--	130 часов (6 дней)

В рассмотренном примере применения СПО для построения ЦД вычислительной инфраструктуры эксперимента *VM@N* комплекса *NISA* решена задача поиска конфигурации оборудования для системы сбора, хранения и обработки данных. Важность проведенного исследования подтверждается письмом о применении результатов в эксперименте *VM@N* (Приложение 4). Результаты работы имеют особую практическую значимость, так как позволили оценить требуемое количество ресурсов для хранения и обработки данных с учетом планируемых параметров потоков данных будущих сеансов эксперимента *VM@N*. Найдено, сколько времени потребуется на обработку экспериментальных и модельных данных по окончании сеанса эксперимента. Полученные результаты помогли принять правильное решение в процессе управления потоками задач и более корректно распределить нагрузку на вычислительные ресурсы. Общее время работы ЦД для всех вычислительных экспериментов (от момента запуска до получения результатов) составило 20 часов, что не только удовлетворяет заданным требованиям, но и позволяет в короткие сроки проверить большое количество вариантов для модернизации РЦОД. Это, безусловно, способствует более оперативному принятию решения о развитии инфраструктуры, от которой напрямую зависят новые достижения ученых.

4.3. Применение специального программного обеспечения для построения цифрового двойника вычислительной системы онлайн-фильтра данных эксперимента SPD комплекса NICA

В составе комплекса *NICA* есть компоненты, которые находятся на стадии строительства, например, детектор *SPD*. Эксперимент *SPD* создается для изучения спиновой структуры протон и дейтрон и других спиновых явлений с поляризованными пучками протонов и дейтронов при энергии столкновения до 27 ГэВ и светимости до $10^{32} \text{ см}^{-2}\text{с}^{-1}$ [78]. Согласно технической документации, с новой установки будут поступать данные со скоростью 20 ГБ/сек, что соответствует 200 ПБ/год [79]. Проектирование масштабной системы для хранения и обработки такого огромного объема экспериментальных данных требует особого внимания. В связи с этим было предложено использовать разработанное СПО для создания ЦД вычислительной инфраструктуры эксперимента *SPD* с целью тестирования подсистем, в частности онлайн-фильтра данных, с разными вариантами параметров оборудования, потоков данных и потоков задач.

Основная цель онлайн-фильтра данных — быстрая реконструкция событий, возникающих при столкновении частиц, и подавление фоновых событий как минимум в 20 раз в режиме реального времени. Учитывая, что скорость поступления данных планируется на уровне 20 ГБ/сек, быстрая реконструкция и фильтрация данных детектора *SPD* не могут быть выполнены на одном вычислительном узле (процессоре или ядре). Это значит, что вычислительная система онлайн-фильтра должна представлять собой специализированный высокопроизводительный кластер, включающий несколько систем хранения данных (для получения данных детектора и для временного хранения отфильтрованных данных перед передачей их в долговременное хранилище) и большое количество одинаковых рабочих узлов обработки данных [79].

Онлайн-фильтр будет работать следующим образом. «Сырые» необработанные *raw*-данные детектора со скоростью 20 ГБ/сек будут поступать в буфер приема данных, где запланирована запись экспериментальных данных в *raw*-файлы. Объем отдельного *raw*-файла составляет 4 ГБ. Процесс подавления фоновых событий, который выполняется на некотором наборе вычислительных ресурсов в режиме реального времени со скоростью 1 000 событий/сек, включает в себя три этапа обработки: декодирование, частичная реконструкция и непосредственно фильтрация данных. На каждом этапе формируются результирующие файлы: *dec*, *prec* и *filtered* соответственно, которые записываются на ресурсы временного хранилища данных. Объем отдельного *dec*-файла составляет 4 ГБ, *prec*-файла — 8 ГБ, *filtered*-файла — 450 МБ [79].

Таким образом, для обеспечения быстрой реконструкции событий и фильтрации данных в режиме реального времени необходимо спроектировать эффективную вычислительную систему, предусмотрев возможности для ее дальнейшего развития и оптимизации. Рассмотрим результаты

использования разработанного СПО для построения ЦД системы сбора, хранения и обработки данных для онлайн-фильтра эксперимента *SPD*.

Первое применение программного комплекса для создания ЦД вычислительной системы онлайн-фильтра обусловлено необходимостью рассчитать требуемые параметры оборудования для хранения и обработки данных, а также дать оценку загруженности сети передачи данных. В рассматриваемом примере поставлена задача построения ЦД для сбора и фильтрации данных эксперимента, который будет работать на протяжении 24 часов со следующей периодичностью: 1 час работы и 3 часа перерыв. Дополнительное требование выдвинуто к времени обработки *raw*-файла: все три этапа до получения отфильтрованного файла должны длиться не более 10 минут. Файлы в системах хранения данных не удаляются.

На рисунке 73 представлен элемент веб-интерфейса СПО, который позволяет описать инфраструктуру рассматриваемой вычислительной системы и параметры входящего в ее состав оборудования. В состав инфраструктуры добавлены такие объекты, как источник данных (*Trigger*), вычислительная компонента (*Computing*) и два хранилища данных (*Buffer* и *Intermediate*). Настройка потоков данных и задач проведена в соответствии с представленным описанием процесса работы онлайн-фильтра.



Рисунок 73. Вычислительная система онлайн-фильтра данных эксперимента *SPD*

После построения инфраструктуры можно переходить непосредственно к созданию ЦД, для запуска которого необходимо настроить некоторые параметры: задать продолжительность работы вычислительной системы, добавить вероятностные события, которые могут происходить

в системе, а также указать объекты и события для записи логов. Важно отметить, что при заданной продолжительности проведения эксперимента (24 часа) дано ограничение на время активного получения данных (1 час работы и 3 часа перерыв), в связи с чем в настройках ЦД указана эффективность генерации данных (20%).

Итоговые результаты работы ЦД были получены через 40 минут после его запуска. Рассмотрим результаты более подробно на изображениях, которые экспортированы с веб-сервиса.

При интенсивности генерации 20% с установки *SPD* за 24 часа будет получено примерно 400 ТБ экспериментальных данных (см. рис. 74). Для обработки всех *raw*-файлов в режиме реального времени потребуется примерно 1 400 вычислительных узлов (см. рис. 75), 120 из которых будут заняты задачами декодирования экспериментальных данных (*RawToDec*), 430 — задачами частичной реконструкции декодированных данных (*DecToPrec*) и 850 — непосредственно фильтрацией реконструированных данных (*PrecToFilt*). Требуемый объем данных для промежуточного хранилища, где располагаются результирующие файлы трех этапов обработки, составляет примерно 1 250 ТБ (см. рис. 76). При этом 400 ТБ будут занимать декодированные данные (*Dec*), 800 ТБ — данные частичной реконструкции (*Prec*) и 50 ТБ — отфильтрованные данные (*Filt*). Загрузка каналов связи при передаче данных отражена на рисунках 77-79. Делаем вывод, что между установкой *SPD* и буфером приема данных необходимо обеспечить канал связи пропускной способностью не менее 50 Гбит/с (см. рис. 77), между буфером приема данных и вычислительными ресурсами — 40 Гбит/с (см. рис. 78), между вычислительными ресурсами и временным хранилищем данных — 250 Гбит/сек (см. рис. 79).

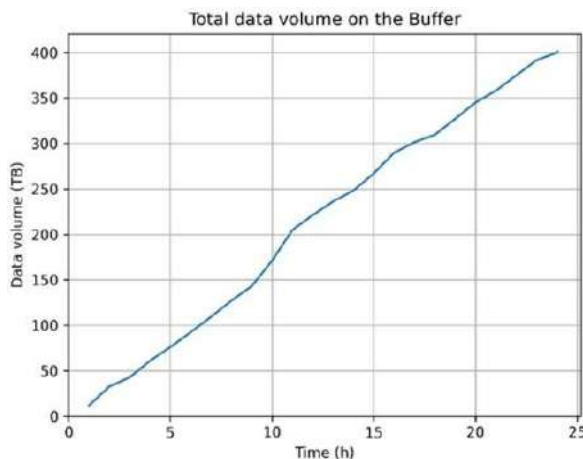


Рисунок 74. Объем накопленных экспериментальных данных за 24 часа

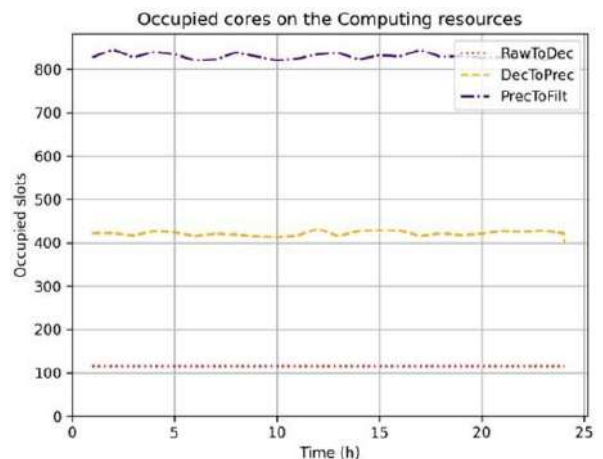


Рисунок 75. Использование вычислительных ресурсов для обработки данных для подавления фоновых событий

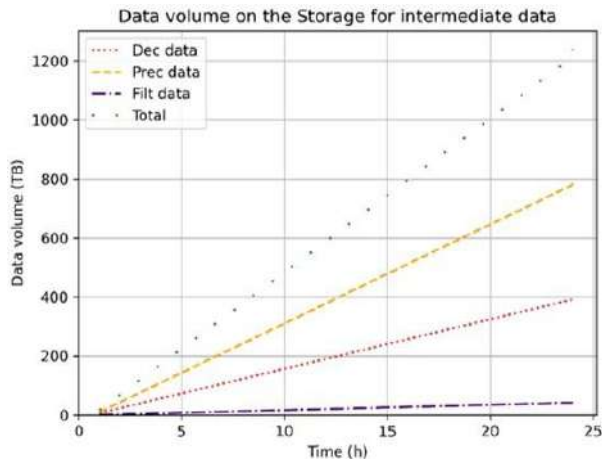


Рисунок 76. Объем результирующих данных в промежуточном хранилище

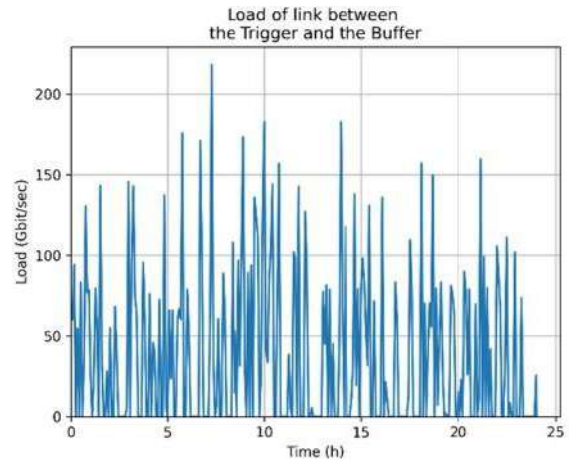


Рисунок 77. Загрузка канала передачи данных между установкой SPD и буфером приема данных

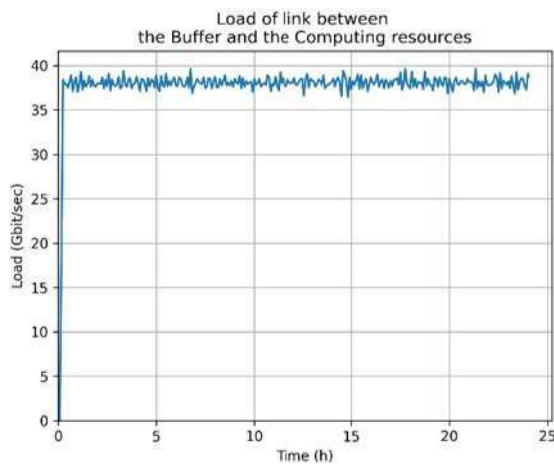


Рисунок 78. Загрузка канала передачи данных между буфером приема данных и вычислительными ресурсами

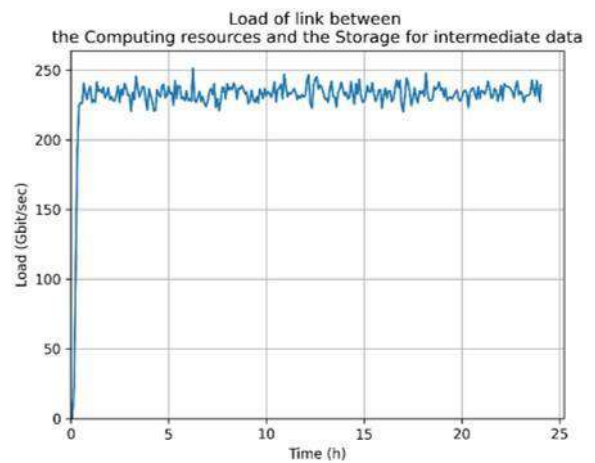


Рисунок 79. Загрузка канала передачи данных между вычислительными ресурсами и временным хранилищем данных

По результатам проведенной работы сделаны следующие выводы. С учетом 20% эффективности генерации данных за 24 часа работы эксперимента будет получено 400 ТБ данных, обработка которых требует не менее 1 500 вычислительных узлов для обеспечения быстрой реконструкции событий и фильтрации данных в режиме реального времени, при условии обеспечения пропускной способности каналов связи на уровне 50 Гбит/сек. В промежуточном хранилище для временного хранения отфильтрованных данных перед передачей их в долговременное хранилище используется 1 250 ТБ. Скорость передачи данных от вычислительных ресурсов до промежуточного хранилища составляет 250 Гбит/с.

Результаты применения СПО для построения ЦД вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NICA* доказали возможность использования разработки для решения задачи проектирования РЦОД. Высокая практическая значимость проведенной работы подтверждается письмом о применении результатов исследований в эксперименте *SPD* (Приложение 4). Результаты исследования позволили оценить требуемые параметры оборудования для

хранения, обработки и передачи данных онлайн-фильтра с учетом планируемых характеристик потоков данных эксперимента *SPD*. Время работы ЦД (от момента запуска до получения результатов) составило 40 минут, что не только удовлетворяет заданным требованиям, но и позволит в короткие сроки провести тестирование вычислительной системы онлайн-фильтра с разными вариантами параметров оборудования, потоков данных и задач. Это, несомненно, обеспечит качественное развитие и поддержку вычислительной инфраструктуры. Все вышесказанное подтверждает, что полученные результаты позволяют проводить масштабирование и трансформацию системы под различные задачи и требования.

4.4. Выводы к главе 4

Проведена верификация и опытная эксплуатация СПО. В процессе верификации экспериментально подтверждена адекватность разработанных моделей ЦД центра сбора, хранения и обработки данных эксперимента *VM@N* комплекса *NICA*. Рассматриваемая вычислительная инфраструктура использовалась для получения, хранения и обработки данных последнего сеанса эксперимента, который проходил с декабря 2022 года по февраль 2023 года. Результаты верификации доказали корректную работу программы моделирования. Адекватность оценивалась по нескольким показателям. Получившиеся значения доказали, что отклонения результатов работы ЦД от результатов работы реальных РЦОД будут не более **трех среднеквадратических отклонений** статистических данных мониторинга.

Достоверность рекомендаций и выводов по результатам диссертационного исследования подтверждается практикой применения разработанных методов при проектировании и развитии вычислительных инфраструктур для крупных экспериментов в области физики высоких энергий.

Проведена опытная эксплуатация СПО на примерах построения цифрового двойника вычислительной инфраструктуры эксперимента *VM@N* и цифрового двойника вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NICA*. Показана применимость СПО и возможность использования ЦД для проектирования, повышения эффективности функционирования, качества и надежности сложных систем сбора, хранения и обработки данных.

С помощью разработанного СПО успешно построены ЦД, по результатам работы которых были найдены характеристики оборудования, необходимого для хранения, обработки и передачи данных с учетом планируемых параметров потоков данных будущих сеансов экспериментов комплекса *NICA*. Главный критерий – минимизация времени обработки данных. Полученные результаты обеспечили качественную проверку функционирования РЦОД, а также помогли принять правильное решение в процессе управления потоками задач и более корректно распределить

нагрузку на вычислительные ресурсы. Научное обоснование выбора конфигураций РЦОД подтверждает важность проведенного исследования. Применение результатов диссертационного исследования для решения задачи поиска конфигурации оборудования для вычислительных инфраструктур экспериментов *VM@N* и *SPD* комплекса *NISA* в ОИЯИ подтверждается письмами о применении (Приложение 4).

Время получения результатов работы ЦД (от момента запуска до получения результатов), которое зависит от масштаба РЦОД и процессов, происходящих в нем, **не превышает 24 часов**. Это является дополнительным преимуществом, т.к. способствует более оперативному принятию решения о развитии РЦОД.

Доказано третье положение, выносимое на защиту: «Подтверждена адекватность построенных методов и алгоритмов на примере вычислительной инфраструктуры существующего эксперимента».

Заключение

В диссертации разработан новый метод построения и использования цифровых двойников РЦОД с целью улучшения их технических характеристик. Предложенный метод отличается от существующих возможностью моделировать процессы обработки и хранения данных, учитывая характеристики потоков данных и задач, вероятностей сбоев, отказов и изменений в производительности оборудования и других процессов, происходящих в моделируемой системе.

Основные результаты диссертационной работы заключаются в следующем.

1. Исследованы системные связи и закономерности функционирования сложных систем, которыми являются РЦОД, с использованием принципов системного подхода и методов системного анализа.

2. Разработаны методы описания распределенных систем, принятия решений по выбору конфигураций оборудования, управления ресурсами и процессами сложных систем.

3. Разработаны модели, методы и алгоритмы построения цифровых двойников РЦОД.

4. Реализованы алгоритмы, структура БД и веб-интерфейс пользователя для построения и запуска ЦД, а также предоставления графической информации о результатах его работы. Используются современные архитектурные решения и средств для разработки программного обеспечения, веб-приложений и БД.

5. На основе созданных моделей, методов и алгоритмов разработано СПО, позволяющее сравнивать эффективность функционирования РЦОД в зависимости от различных конфигураций оборудования.

6. Получено свидетельство о государственной регистрации программы для ЭВМ №2023667305 «Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных» (Приложение 3).

7. Проведена верификация ядра ЦД на примере центра сбора, хранения и обработки данных эксперимента *VM@N* комплекса *NISA*. Адекватность модели оценивалась по нескольким показателям. Получившиеся значения доказали, что отклонения результатов работы ЦД от результатов работы реальных РЦОД будут не более **трех среднеквадратических отклонений** статистических данных мониторинга.

8. Проведена опытная эксплуатация СПО при построении ЦД вычислительной инфраструктуры эксперимента *VM@N*. Получена наиболее подходящая конфигурация оборудования для обработки данных за минимальное время, а также выбрана стратегия управления потоком задач и распределения нагрузки на вычислительные ресурсы. Выбранная альтернатива обеспечит уско-

рение в 2 раза процесса преобразования *raw*-данных. **Предложенная тактика управления потоком задач** позволит за то же время обработать не только экспериментальные, но и модельные данные.

9. Проведена опытная эксплуатация СПО при построении ЦД вычислительной системы онлайн-фильтра данных эксперимента *SPD* комплекса *NICA*. Результаты позволили оценить требуемые параметры оборудования для хранения, обработки и передачи данных онлайн-фильтра с учетом планируемых характеристик потоков данных эксперимента.

10. Определено время получения результатов работы ЦД (от момента запуска до получения результатов) с учетом масштаба РЦОД и происходящих в нем процессов, которое **не превышает 24 часов**. Это является дополнительным преимуществом, т.к. способствует более оперативному принятию решения о развитии РЦОД.

11. Получены рекомендации, которые учтены при проектировании и развитии вычислительных инфраструктур крупных экспериментов в области физики высоких энергий. Результаты применения СПО доказали работоспособность и высокое качество разработанных в диссертации моделей и алгоритмов, что подтверждено письмами о применении (Приложение 4).

12. Результаты диссертационной работы используются в учебном процессе Федерального государственного бюджетного образовательного учреждения высшего образования «Университет «Дубна» в курсе «Распределенные вычисления и облачные технологии» для подготовки магистров по направлению 27.04.03 Системный анализ и управление по профилю «Цифровые платформы и аналитика больших данных», что подтверждается актом о внедрении (Приложение 5).

Результаты диссертационного исследования являются важным вкладом в развитии научных исследований, которые оказывают существенную помощь в проектировании, создании, поддержке, а также решении задач управления и развитии центров сбора, хранения и обработки данных для крупных научных проектов. Разработанный метод построения ЦД позволяет создавать прототип вычислительной инфраструктуры, который помогает дать оценку эффективности функционирования РЦОД, способствует оперативному принятию решения о масштабировании распределенной системы и изменении характеристик входящего в его состав оборудования, что приводит к повышению качества и надежности сложных систем сбора, хранения и обработки данных.

В дальнейшем разработанное СПО [80] может применяться для широкого класса задач области проектирования, построения и развития РЦОД для больших научных экспериментов и крупных проектов.

В перспективе развития данной работы планируется совершенствование разработанного метода и добавление функции многокритериальной оптимизации при выборе конфигурации оборудования для РЦОД. В качестве критериев должны будут учитываться не только технические,

но и стоимостные параметры оборудования, входящего в состав РЦОД [81, 82]. Для повышения удобства применения СПО необходимо добавить личный кабинет пользователя, а для обеспечения безопасности — модуль для регистрации, т.е. предоставлять доступ к СПО для построения ЦД только после прохождения процедуры авторизации.

Словарь терминов

1. **адекватность:** Соответствие модели распределенного центра сбора, хранения и обработки данных реальной системе по некоторому перечню характеристик.

2. **верификация:** Процедура проверки компьютерной программы на корректность реализации поставленной задачи путем оценки результатов ее работы по заданным критериям; в данной работе сравнение результатов работы программы моделирования распределенной системы осуществляется с результатами мониторинга этой системы.

3. **вычислительный эксперимент:** Эксперимент на цифровом двойнике распределенной системы сбора, хранения и обработки данных для изучения ее поведения при определенном сценарии с заданными параметрах оборудования, потоков данных и задач.

4. **специальное программное обеспечение:** Комплекс программ, реализующий метод решения конкретных задач, а именно в данной работе: новый метод создания, в т. ч. построения, настройки, запуска цифровых двойников распределенных центров сбора, хранения и обработки данных, а также демонстрации результатов его работы.

5. **ядро цифрового двойника:** компьютерная программа, где реализованы алгоритмы моделирования различных процессов, происходящих в системе, в частности, распределенном центре сбора, хранения и обработки данных.

Список литературы

1. Grieves M. Digital Twin: Manufacturing Excellence Through Virtual Factory Replication. Digital Twin White Paper // ResearchGate. 2014. URL: https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication (дата обращения: 15.03.2023).
2. Прохоров А., Лысачев М. Цифровой двойник. Анализ, тренды, мировой опыт. 1-е изд. М.: ООО «АльянсПринт», 2020. 401 с.
3. Корнилов В.В., Исаев В.А., Исаев К.А. Перспективы использования центров обработки данных при решении задач математической биологии и биоинформатики // Математическая биология и биоинформатика. Февраль 2015. Т. 10. № 1. С. 60–71.
4. Ляшенко М.А. Содержание стратегии развития центра обработки данных // Интернет-журнал «Науковедение». Июль – август 2015. Т. 7. № 4.
5. Business Ecosystems. Центры обработки данных. Повышение масштабируемости, гибкости и безопасности бизнеса // Business Ecosystems. URL: https://becsys.ru/uploads/files/solutions/technological-solutions/3/Business_Ecosystems_Data_Centers.pdf (дата обращения: 15.03.2023).
6. The Digital Twin Company [Электронный ресурс] // Future Facilities: [сайт]. URL: <https://www.futurefacilities.com/> (дата обращения: 20.12.2022).
7. Sunbird Software, Inc. DCIM - Data Center Infrastructure Management Software System, Cable Management, Infrastructure Design & Optimization Companies [Электронный ресурс] // Sunbird DCIM: [сайт]. URL: <https://www.sunbirdcim.com/> (дата обращения: 20.12.2022).
8. Петров А.В. Имитационное моделирование как основа технологии цифровых двойников // Вестник Иркутского государственного технического университета. 2018. Т. 22. № 10. С. 56–66.
9. Foster I., Kesselman C. The grid: blueprint for a new computing infrastructure. San Francisco (CA): Morgan Kaufmann Publishers Inc., 1999. 593 pp.
10. CERN. Welcome to the Worldwide LHC Computing Grid [Электронный ресурс] // WLCG: [сайт]. URL: <https://wlcg.web.cern.ch/> (дата обращения: 10.12.2022).
11. Berezhnaya A., Dolbilov A., Piyin V., Korenkov V., Lazin Y., Lyalin I., Mitsyn V., Ryabinkin E., Shmatov S., Strizh T., et al. LHC Grid Computing in Russia: present and future // Journal of Physics: Conference Series. 2014. Vol. 513. No. 6. P. 062041.

12. Kekelidze V., Kovalenko A., Lednicky R., Matveev V., Meshkov I., Sorin A., Trubnikov G. Status of the NICA project at JINR // European Physical Journal Web of Conferences. March 2017. Vol. 138. P. 01027.
13. Serebrov A.P., Vassiljev A.V., Varlamov V.E., Geltenbort P., Gridnev K.A., Dmitriev S.P., Dovator N.A., Egorov A.I., Ezhov V., Zherebtsov O.M., et al. Program for studying fundamental interactions at the PIK reactor facilities // Physics of Atomic Nuclei. June 2016. Vol. 79. No. 3. pp. 293–303.
14. Баранов Г.Н., Богомягков А.В., Левичев Е.Б., Синяткин С.В. Оптимизация магнитной структуры источника синхротронного излучения четвертого поколения СКИФ в Новосибирске // Сибирский физический журнал. 2020. Т. 15. № 1. С. 5–23.
15. Avrorin A.D., Avrorin A.V., Aynutdinov V.M., Bannash R., Belolaptikov I.A., Brudanin V.B., Budnev N.M., Doroshenko A.A., Domogatsky G.V., Dvornický R., et al. Baikal-GVD: status and prospects // European Physical Journal Web of Conferences. October 2018. Vol. 191. P. 01006.
16. Fengpeng A., et al. Neutrino physics with JUNO // Journal of Physics G: Nuclear and Particle Physics. February 2016. Vol. 43. No. 3. P. 030401.
17. Пряхина Д.И., Кореньков В.В. Актуальность создания цифрового двойника для управления распределенными центрами сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 262–271.
18. Нечаевский А.В., Кореньков В.В. Пакеты моделирования DataGrid // Системный анализ в науке и образовании. 2009. № 1. С. 21–35.
19. Самоваров О.И., Кузюрин Н.Н., Грушин Д.А., Аветисян А.И., Михайлов Г.М., Рогов Ю.П. Проблемы моделирования GRID-систем и их реализации // Научный сервис в сети Интернет: решение больших задач. Москва. 2008. С. 83–88.
20. Кореньков В.В., Нечаевский А.В., Ососков Г.А., Пряхина Д.И., Трофимов В.В., Ужинский А.В. Моделирование грид и облачных сервисов как средство повышения эффективности их разработки // CEUR Workshop Proceedings. 2014. Т. 1297. С. 13–19.
21. GridSim: A Grid Simulation Toolkit 5.2 [Электронный ресурс] // Soft 112: [сайт]. [2010]. URL: <https://gridsim-a-grid-simulation-toolkit.soft112.com/> (дата обращения: 10.10.2022).
22. The University of Edinburgh. API Specification [Электронный ресурс] // SimJava v2.0: [сайт]. [2002]. URL: <https://www.icsa.inf.ed.ac.uk/research/groups/hase/simjava/doc/index.html> (дата обращения: 23.12.2022).
23. Кореньков В.В., Нечаевский А.В., Ососков Г.А., Пряхина Д.И., Трофимов В.В., Ужинский А.В. Синтез процессов моделирования и мониторинга для развития систем хранения и

- обработки больших массивов данных в физических экспериментах // Компьютерные исследования и моделирование. 2015. Т. 7. № 3. С. 691–698.
24. Kadochnikov I., Korenkov V., Mitsyn V., Pelevanyuk I., Strizh T. Service monitoring system for JINR Tier-1 // The European Physical Journal Conferences. Sep 2019. Vol. 214. P. 08016.
 25. Кореньков В.В., Нечаевский А.В., Ососков Г.А., Пряхина Д.И., Трофимов В.В., Ужинский А.В. Моделирование грид и облачных сервисов как важный этап их разработки // Системы и средства информатики. 2015. Т. 25. № 1. С. 4–19.
 26. Нечаевский А.В., Пряхина Д.И., Ужинский А.В. Разработка веб-сервиса для моделирования систем хранения и обработки данных физических экспериментов // Системный анализ в науке и образовании. 2015. № 4. С. 1–7.
 27. Korenkov V., Nechaevskiy A., Ososkov G., Pryahina D., Trofimov V., Uzhinskiy A., Balashov N. Web-Service Development of the Grid-Cloud Simulation Tools // Procedia Computer Science. 2015. Vol. 66. pp. 533–539.
 28. Korenkov V., Nechaevskiy A., Ososkov G., Pryahina D., Trofimov V., Uzhinskiy A. Simulation concept of NICA-MPD-SPD Tier0-Tier1 computing facilities // Particles and Nuclei Letters. 2016. Vol. 13. No. 5. pp. 1074–1083.
 29. Кутовский Н.А., Нечаевский А.В., Ососков Г.А., Пряхина Д.И., Трофимов В.В. Моделирование межпроцессорного взаимодействия при выполнении MPI-приложений в облаке // Компьютерные исследования и моделирование. 2017. Т. 9. № 6. С. 955–963.
 30. Nechaevskiy A., Ososkov G., Pryahina D., Trofimov V., Li W. Simulation approach for improving the computing network topology and performance of the China IHEP Data Center // European Physical Journal Web of Conferences. 2019. Vol. 214. P. 08018.
 31. Grieves M., Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems // In: Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches. Springer, Cham, 2017. pp. 85-113.
 32. Barricelli B.R., Casiraghi E., Fogli D. Definitions, Characteristics, Applications, and Design Implications // IEEE Access. November 2019. Vol. 7. pp. 167653–167671.
 33. Денисов А.С., Куверин И.Ю. Цифровые двойники как основа цифровой трансформации технической эксплуатации автомобилей в рамках четвёртой технологической революции // Техническое регулирование в транспортном строительстве. 2020. Т. 3. № 42. С. 165–168.
 34. Yang J., Zhang W., Liu Y. Subcycle fatigue crack growth mechanism investigation for aluminum alloys and steel // 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. Boston. 2013. P. 1499.

35. Шалумов А.С., Шалумова Н.А., Шалумов М.А. Цифровой двойник авиационной электроники: моделирование физических процессов при формировании электронной модели // Автоматизация. Современные технологии, Т. 75, № 9, 2021. С. 403–415.
36. Филатов А.Р. Цифровой двойник корпуса судна. Назначение и основные принципы построения // Труды Крыловского государственного научного центра. 2021. Т. 4. № 398. С. 87–92.
37. Шевченко Д.В. Методология построения цифровых двойников на железнодорожном транспорте // Вестник Научно-исследовательского института железнодорожного транспорта. 2021. Т. 80. № 2. С. 91–99.
38. Быкова В.Н., Ким Е., Гаджиалиев М.Р., Мусиенко В.О., Оруджев А.О., Туровская Е.А. Применение цифрового двойника в нефтегазовой отрасли // Актуальные проблемы нефти и газа. 2020. Т. 28. № 1. С. 8.
39. Тихонов А.И., Стулов А.В., Каржевин А.А., Подобный А.В. Разработка нелинейной модели трехфазного трансформатора для исследования влияния несимметрии магнитной системы на работу устройства в произвольных режимах // Вестник Ивановского государственного энергетического университета. 2020. № 1. С. 22–31.
40. Абрамов В.И., Столяров А.Д. Цифровые двойники в сельском хозяйстве: возможности и перспективы // АПК России: образование, наука, производство. Сборник статей II Всероссийской (национальной) научно-практической конференции. Пенза. 2021. С. 3–9.
41. Bruynseels K., Santoni de Sio F., Hoven J. Digital Twins in Health Care: Ethical Implications of an Emerging Engineering Paradigm // *Frontiers in genetics*. February 2018. Vol. 9. No. 31.
42. Меньшутина Н.В. Многоуровневое моделирование аэрогелей и их производства // Сверхкритические флюиды, фундаментальные основы, технологии, инновации. Сочи. 2017. С. 124–126.
43. Lehtola V.V., Koeva M., Elberink S.O., Raposo P., Virtanen J.P., Vahdatikhaki F., Borsci S. Digital twin of a city: Review of technology serving city needs // *International Journal of Applied Earth Observation and Geoinformation*. 2022. Vol. 114. P. 102915.
44. Иванов С.А., Никольская К.Ю., Радченко Г.И., Соколинский Л.Б., Цымблер М.Л. Концепция построения цифрового двойника города // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 9. № 4. С. 5–23.
45. Амирханян А.Г. Цифровые двойники в логистике // *Modern science*. 2020. № 1–2. С. 37–40.
46. Rosen R., Von Wichert G., Bettenhausen K.D. About the importance of autonomy and digital twins for the future of manufacturing // *IFAC-PapersOnLine*. 2015. Vol. 48. No. 3. pp. 567–572.

47. Polyniak K., Matthews J. The Johns Hopkins Hospital Launches Capacity Command Center to Enhance Hospital Operations // John Hopkins Medicine. 2016. URL: https://www.hopkinsmedicine.org/news/media/releases/the_johns_hopkins_hospital_launches_capacity_command_center_to_enhance_hospital_operations (дата обращения: 18.12.2022).
48. Nemati K., Zabalegui A., Bana M., Seymour M.J. Quantifying data center performance // 34th Thermal Measurement, Modeling & Management Symposium. 2018. pp. 141–147.
49. Batty M. Digital twins. Environment and Planning B // Urban Analytics and City Science. 2018. Vol. 5. No. 45. pp. 817–820.
50. Boschert S., Rosen R. Digital Twin — The Simulation Aspect // In: Mechatronic Futures. Springer, Cham, 2016. pp. 59–74.
51. Cimino C., Negri E., Fumagalli L. Review of digital twin applications in manufacturing // Computers in Industry. December 2019. Vol. 113. P. 103130.
52. Halenar I. 20th International Carpathian Control Conference // Virtualization of Production Using Digital Twin Technology. Krakow-Wieliczka, Poland. 2019. pp. 1–5.
53. Хитрых Д.П. Цифровые двойники: прошлое, настоящее и будущее // CADFEM REVIEW – Научно-технический журнал от компании КАДФЕМ. 2021. Т. 8. № 31. С. 30–32.
54. Korenkov V., Nechaevskiy A., Ososkov G., Priakhina D., Trofimov V. A probabilistic approach of the simulation of data processing centers // European Physical Journal Web of Conferences. January 2020. Vol. 226. P. 03012.
55. Романов В.Н. Основы системного анализа / учебно-методический комплекс. Санкт-Петербург: СЗТУ, 2011. 298 с.
56. Пряхина Д.И., Кореньков В.В., Трофимов В.В. Метод построения цифровых двойников для решения задач эффективного управления и развития распределенных центров сбора, хранения и обработки данных // Современные информационные технологии и ИТ-образование. 2023. Т. 19. № 2. С. 272–281.
57. Ramalho L. Fluent Python: Clear, Concise, and Effective Programming. 1st ed. O'Reilly Media, 2015.
58. Rubio D. Beginning Django: Web Application Development and Deployment with Python. Mexico: Ensenada, 2017.
59. Дронов В.А. Django 2.1. Практика создания веб-сайтов на Python. Санкт-Петербург: БХВ-Петербург, 2019.

60. Новиков Б.А. Основы технологий баз данных: учеб. пособие. 2-е изд. Москва: ДМК Пресс, 2020.
61. Führer C., Solem J.E., Verdier O. Scientific Computing with Python. 2nd ed. Packt Publishing, 2021.
62. Argparse Tutorial [Электронный ресурс] // Python 3.12.1 documentation: [сайт]. URL: <https://docs.python.org/3/howto/argparse.html> (дата обращения: 17.07.2021).
63. configparser — Configuration file parser [Электронный ресурс] // Python 3.12.1 documentation: [сайт]. URL: <https://docs.python.org/3/library/configparser.html> (дата обращения: 17.07.2021).
64. Graph theory (network) library for visualisation and analysis [Электронный ресурс] // Cytoscape.js: [сайт]. URL: <https://js.cytoscape.org/> (дата обращения: 20.05.2022).
65. Даббас Э. Интерактивные дашборды и приложения с PLOTLY и DASH. ДМК Пресс, 2023.
66. Django documentation [Электронный ресурс] // Django: [сайт]. URL: <https://docs.djangoproject.com/en/5.0/> (дата обращения: 17.05.2022).
67. Пряхина Д.И., Кореньков В.В., Трофимов В.В., Герценбергер К.В. Верификация программы моделирования для создания цифровых двойников распределенных центров сбора, хранения и обработки данных // International Journal of Open Information Technologies. Январь 2024. Т. 12. № 1. С. 118–128.
68. Kapishin M., BM@N Collaboration. Studies of baryonic matter at the BM@N experiment (JINR) // Nuclear Physics A. 2019. Vol. 982. P. 967.
69. JINR. NICA Bulletin // ЛАБОРАТОРИЯ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ – имени В.И.Векслера и А.М.Балдина. 2023. URL: https://lhep.jinr.ru/wp-content/uploads/2023/03/nica_vypusk9.pdf (дата обращения: 20.04.2023).
70. Baginyan A., Balandin A., Balashov N., Dolbilov A., Gavrish A., Golunov A., Gromova N., Kashunin I., Korenkov V., Kutovskiy N., et al. Current status of the MICC: an overview // CEUR Workshop Proceeding. 2021. Vol. 3041. pp. 1–8.
71. Adam G., Bashashin M., Belyakov D., Kirakosyan M., Matveev M., Podgainy D., Sapozhnikova T., Streltsova O., Torosyan S., Vala M., et al. IT ecosystem of the HybriLIT heterogeneous platform for high performance computing and training of IT specialists // CEUR Workshop Proceeding. 2018. Vol. 2267. pp. 638–644.
72. Korenkov V., Pelevanyuk I., Tsaregorodtsev A. DIRAC at JINR as a general purpose system for massive computations // Journal of Physics: Conference Series. 2023. Vol. 2438. P. 012029.
73. Pelevanyuk I. Performance evaluation of computing resources with DIRAC interware // AIP Conference Proceedings. September 2021. Vol. 2377. No. 1. P. 040006.

74. Gertsenberger K., Pelevanyuk I. BM@N Mass Data Production on distributed infrastructure for Run 8 using DIRAC, 10th Collaboration Meeting of the BM@N Experiment at the NICA Facility , St. Petersburg, Presentation 2023.
75. Priakhina D., Trofimov V., Ososkov G., Gertsenberger K. Data center simulation for the BM@N experiment of the NICA project // AIP Conference Proceeding. 2021. Vol. 2377. P. 040007.
76. Priakhina D., Korenkov V., Gertsenberger K., Trofimov V. Simulation of Data Processing for the BM@N Experiment of the NICA Complex // CEUR Workshop Proceedings. 2021. Vol. 3041. pp. 483–487.
77. Priakhina D., Korenkov V., Trofimov V., Gertsenberger K. Simulation Results of BM@N Computing Infrastructure // Physics of Particles and Nuclei Letters. 2023. Vol. 20. No. 5. pp. 1272–1275.
78. The SPD proto-collaboration. Conceptual design of the Spin Physics Detector // International spin physics collaboration at the collider NICA. 2021. URL: <http://spd.jinr.ru/wp-content/uploads/2021/04/2102.00442.pdf> (дата обращения: 25.09.2023).
79. The SPD collaboration. Technical Design Report of the Spin Physics Detector // International spin physics collaboration at the collider NICA. 2022. URL: http://spd.jinr.ru/wp-content/uploads/2023/03/TechnicalDesignReport_SPD2023.pdf (дата обращения: 25.09.2023).
80. Кореньков В.В., Пряхина Д.И., Трофимов В.В. Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных, Свидетельство государственной регистрации программы для ЭВМ 2023667305 , Август 14, 2023.
81. Трофимов В.В., Нечаевский А.В., Ососков Г.А., Пряхина Д.И. Вероятностно-стоимостной подход к оптимизации распределенных систем хранения данных физических экспериментов // CEUR Workshop Proceedings. 2018. Т. 226. С. 393–399.
82. Кореньков В.В., Пряхина Д.И., Нечаевский А.В., Ососков Г.А., Трофимов В.В. Моделирование центров хранения и обработки данных с учетом экономических составляющих // Системный анализ в науке и образовании. 2018. № 4.

Приложение 1. Описание сущностей модели данных

№	Название атрибута	Описание атрибута	Тип	Примечание
<i>Сущность DataStorages</i>				
1	<i>storage_id</i>	Идентификатор хранилища	целое число	первичный ключ; уникальное значение
2	<i>storage_name</i>	Название хранилища	символьная строка	
3	<i>storage_description</i>	Описание хранилища	символьная строка	
4	<i>storage_volume</i>	Максимальный объем хранилища (ТБ)	вещественное число	
5	<i>storage_active</i>	Активность хранилища (1 – активное; 0 – неактивное)	целое число	
6	<i>storage_quant</i>	Скорость генерации данных в активном хранилище (ТБ/с)	вещественное число	
7	<i>storage_priority</i>	Значение приоритета хранилища	целое число	
8	<i>storage_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
<i>Сущность ComputingComponents</i>				
1	<i>comp_id</i>	Идентификатор вычислительной компоненты	целое число	первичный ключ; уникальное значение
2	<i>comp_name</i>	Название вычислительной компоненты	символьная строка	
3	<i>comp_description</i>	Описание вычислительной компоненты	символьная строка	
4	<i>comp_cores</i>	Общее количество ядер	целое число	
5	<i>comp_speed</i>	Коэффициент ускорения	вещественное число	
6	<i>comp_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
<i>Сущность Pilots</i>				
1	<i>pilot_id</i>	Идентификатор пилота	целое число	первичный ключ; уникальное значение
2	<i>pilot_name</i>	Название пилота	символьная строка	
3	<i>pilot_description</i>	Описание пилота	символьная строка	
4	<i>pilot_comp</i>	Название вычислительной компоненты	символьная строка	внешний ключ
5	<i>pilot_queue</i>	Название потока задач	символьная строка	внешний ключ
6	<i>pilot_priority</i>	Значение приоритета пилота	целое число	
7	<i>pilot_jobs_part</i>	Доля задач для обработки	вещественное число	

№	Название атрибута	Описание атрибута	Тип	Примечание
8	<i>pilot_storage_input</i>	Название хранилища с данными для обработки	символьная строка	внешний ключ
9	<i>pilot_storage_output</i>	Название хранилища для сохранения данных после обработки	символьная строка	внешний ключ
10	<i>pilot_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
Сущность <i>Slots</i>				
1	<i>slot_id</i>	Идентификатор слота	целое число	первичный ключ;
2	<i>slot_name</i>	Название слота	символьная строка	уникальное значение
3	<i>slot_cores</i>	Количество ядер в слоте	целое число	
4	<i>slot_pilot</i>	Название пилота	символьная строка	внешний ключ
5	<i>slot_active</i>	Активность слота (1 – активен; 0 – неактивен)	целое число	
6	<i>slot_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
Сущность <i>Links</i>				
1	<i>link_id</i>	Идентификатор канала связи	целое число	первичный ключ;
2	<i>link_name</i>	Название канала связи	символьная строка	уникальное значение
3	<i>link_description</i>	Описание канала связи	символьная строка	
4	<i>link_from</i>	Название объекта, от которого передаются данные	символьная строка	
5	<i>link_to</i>	Название объекта, до которого передаются данные	символьная строка	
6	<i>link_bandwidth</i>	Пропускная способность канала связи (ТБ/с)	вещественное число	
7	<i>link_active</i>	Активность канала связи (1 – активен; 0 – неактивен)	целое число	
8	<i>link_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
Сущность <i>DataTags</i>				
1	<i>tag_id</i>	Идентификатор типа данных	целое число	первичный ключ;
2	<i>tag_name</i>	Название типа данных	символьная строка	уникальное значение
3	<i>tag_description</i>	Описание типа данных	символьная строка	
Сущность <i>DataFlows</i>				
1	<i>dataflow_id</i>	Идентификатор потока данных	целое число	первичный ключ;
2	<i>dataflow_name</i>	Название потока данных	символьная строка	уникальное значение

№	Название атрибута	Описание атрибута	Тип	Примечание
3	<i>dataflow_description</i>	Описание потока данных	символьная строка	
4	<i>dataflow_tag</i>	Название типа данных	символьная строка	внешний ключ
5	<i>dataflow_storage</i>	Название хранилища	символьная строка	внешний ключ
6	<i>dataflow_volume</i>	Максимальный объем потока данных (ТБ)	вещественное число	
7	<i>dataflow_files</i>	Количество файлов в потоке данных	целое число	
8	<i>dataflow_active</i>	Активность потока данных (1 – активен; 0 – неактивен)	целое число	
9	<i>dataflow_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
<i>Сущность JobQueues</i>				
1	<i>queue_id</i>	Идентификатор потока задач	целое число	первичный ключ;
2	<i>queue_name</i>	Название потока задач	символьная строка	уникальное значение
3	<i>queue_description</i>	Описание потока задач	символьная строка	
4	<i>queue_input_tag</i>	Название типа входных данных	символьная строка	внешний ключ
5	<i>queue_input_volume</i>	Средний объем входных данных (ТБ)	вещественное число	
6	<i>queue_input_mod</i>	Допустимое значение изменения объема входных данных (ТБ)	вещественное число	
7	<i>queue_output_tag</i>	Название типа выходных данных	символьная строка	внешний ключ
8	<i>queue_output_vol</i>	Средний объем выходных данных (ТБ)	вещественное число	
9	<i>queue_output_mod</i>	Допустимое значение изменения объема выходных данных (ТБ)	вещественное число	
10	<i>queue_runtime</i>	Среднее время выполнения задачи (с)	целое число	
11	<i>queue_runmod</i>	Допустимое значение изменения времени выполнения задачи (с)	целое число	
12	<i>queue_start_delay</i>	Задержка процесса запуска задач (с)	целое число	
13	<i>queue_temp</i>	Среднее время появления задачи (с)	целое число	
14	<i>queue_temp_mod</i>	Допустимое значение изменения времени появления задачи (с)	целое число	
15	<i>queue_power</i>	Общее количество задач	целое число	
16	<i>queue_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ

№	Название атрибута	Описание атрибута	Тип	Примечание
<i>Сущность TransportJobs</i>				
1	<i>transp_id</i>	Идентификатор объекта передачи	целое число	первичный ключ; уникальное значение
2	<i>transp_storage_from</i>	Название хранилища, от которого передаются данные	символьная строка	первичный ключ;
3	<i>transp_storage_to</i>	Название хранилища, до которого передаются данные	символьная строка	внешний ключ; уникальное значение
4	<i>transp_link</i>	Название канала связи	символьная строка	внешний ключ
5	<i>transp_tag</i>	Название типа данных	символьная строка	внешний ключ
6	<i>transp_priority</i>	Значение приоритета объекта передачи	целое число	
7	<i>transp_part</i>	Доля данных для передачи	вещественное число	
8	<i>transp_time_start</i>	Время начала передачи (с)	целое число	
<i>Сущность Events</i>				
1	<i>event_id</i>	Идентификатор события	целое число	первичный ключ; уникальное значение
2	<i>event_name</i>	Название события	символьная строка	уникальное значение
3	<i>event_description</i>	Описание события	символьная строка	
4	<i>event_sensor</i>	Название объекта сбора статистики	символьная строка	внешний ключ
<i>Сущность Event_Object</i>				
1	<i>event_id</i>	Идентификатор события	целое число	внешний ключ
2	<i>object_name</i>	Название объекта	символьная строка	
3	<i>object_type</i>	Тип объекта	символьная строка	
4	<i>distribution</i>	Тип распределения	символьная строка	
5	<i>probability</i>	Вероятность возникновения события	вещественное число	
6	<i>initial_time</i>	Время возникновения первого события	целое число	
7	<i>initial_value</i>	Исходное значение параметра объекта, с которым происходит событие	вещественное число	

№	Название атрибута	Описание атрибута	Тип	Примечание
8	<i>value</i>	Допустимое значение изменения параметра объекта, с которым происходит событие	вещественное число	
9	<i>depend_events</i>	Список зависимых событий	символьная строка	
<i>Сущность Sensors</i>				
1	<i>sn_id</i>	Идентификатор объекта сбора статистики	целое число	первичный ключ;
2	<i>sn_name</i>	Название объекта сбора статистики	символьная строка	уникальное значение
3	<i>sn_description</i>	Описание объекта сбора статистики	символьная строка	
4	<i>sn_period</i>	Период усреднения собранной информации	целое число	
5	<i>sn_frequency</i>	Частота записи собранной информации в БД	целое число	
<i>Сущность Experiments</i>				
1	<i>exp_id</i>	Идентификатор запуска ЦД	целое число	первичный ключ; уникальное значение
2	<i>exp_name</i>	Название запуска ЦД	символьная строка	уникальное значение
3	<i>exp_description</i>	Описание запуска ЦД	символьная строка	
4	<i>exp_params</i>	Параметры запуска ЦД	символьная строка	
5	<i>exp_log</i>	Список идентификаторов объектов сбора статистики	символьная строка	
6	<i>exp_date_create</i>	Дата и время создания	дата и время	
<i>Сущность Modifications</i>				
1	<i>mod_id</i>	Идентификатор сценария	целое число	первичный ключ; уникальное значение
2	<i>mod_experiment</i>	Идентификатор запуска ЦД	целое число	внешний ключ
3	<i>mod_report</i>	Идентификатор сценария запуска ЦД	целое число	уникальное значение
4	<i>mod_json</i>	Параметры оборудования РЦОД	текстовые данные <i>JSON</i>	
5	<i>mod_status</i>	Статус работы (0 – создан; 1 – в работе; 2 – завершено)	целое число	
6	<i>mod_date_create</i>	Дата и время создания	дата и время	
7	<i>mod_date_start</i>	Дата и время запуска	дата и время	

№	Название атрибута	Описание атрибута	Тип	Примечание
8	<i>mod_date_finish</i>	Дата и время завершения работы	дата и время	
<i>Сущность SimulationReport</i>				
1	<i>report_id</i>	Идентификатор записи	целое число	первичный ключ; уникальное значение
2	<i>report_systime</i>	Системное время сбора статистики	целое число	
3	<i>report_equipment_id</i>	Идентификатор объекта, входящего в состав РЦОД, информация о результатах работы которого содержится в записи	целое число	
4	<i>report_equipment</i>	Название объекта, входящего в состав РЦОД, информация о результатах работы которого содержится в записи	символьная строка	
5	<i>report_variable</i>	Значение, описывающее состояние объекта, входящего в состав РЦОД	вещественное число	
6	<i>report_comment</i>	Комментарий	символьная строка	
7	<i>report_modification</i>	Идентификатор сценария запуска ЦД	целое число	внешний ключ

Приложение 2. Инструкция для пользователя по работе со специальным программным обеспечением

1. При входе в систему открывается страница, на которой пользователю предлагается построить инфраструктуру РЦОД (см. рис. 80). На странице есть «активные» кнопки, на которых изображены объекты, доступные для добавления в инфраструктуру РЦОД, а именно: генератор данных, вычислительная компонента, хранилище данных, роботизированная библиотека. Необходимо выбрать из списка объект (1) и расположить его в выделенной для рисования области (2), после чего откроется форма для настройки базовой конфигурации устройства (3). Требуется ввести все необходимые данные и нажать на кнопку «Добавить устройство». На рисунке 81 изображен пример добавления хранилища данных в инфраструктуру РЦОД. По завершении построения инфраструктуры следует нажать на соответствующую кнопку, после чего кнопки с объектами инфраструктуры станут «неактивными» (см. рис. 82). Изображение с получившейся инфраструктурой можно сохранить на локальное устройство.

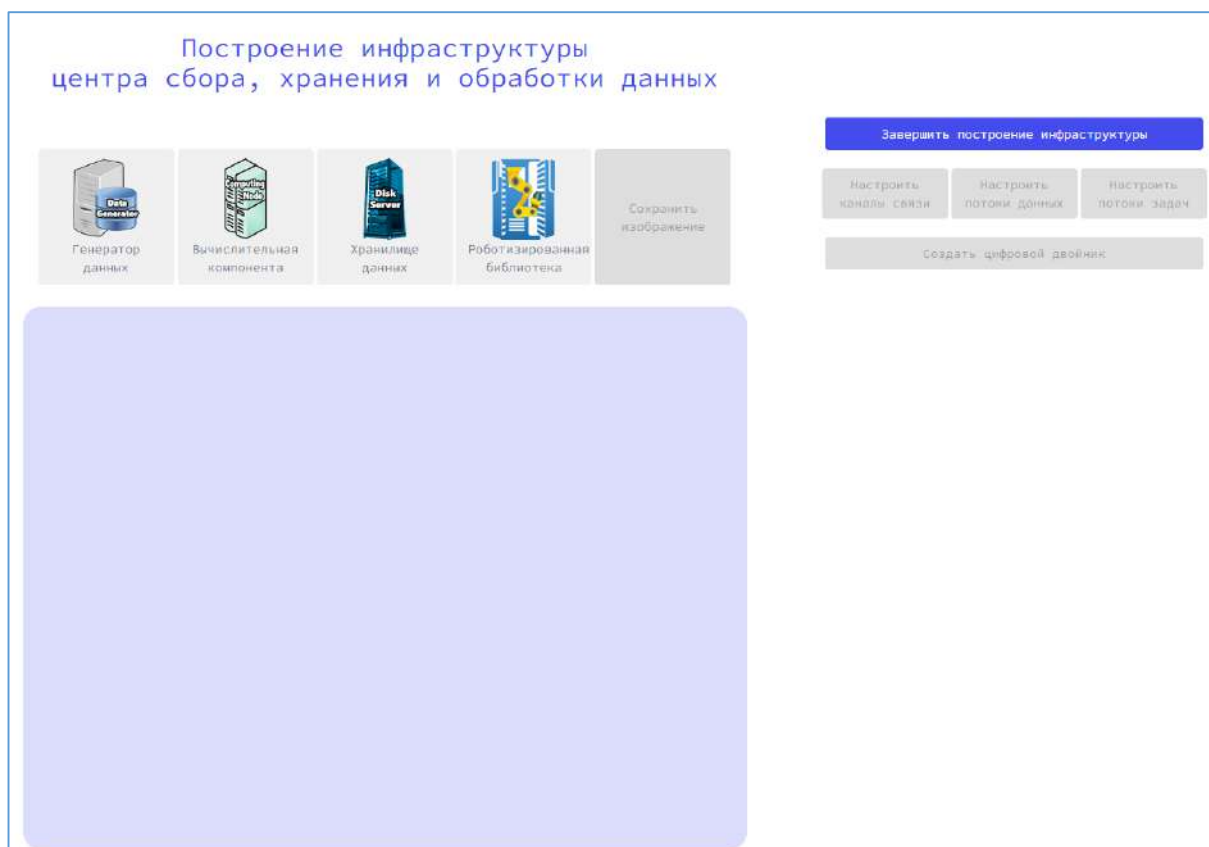


Рисунок 80. Построение инфраструктуры РЦОД

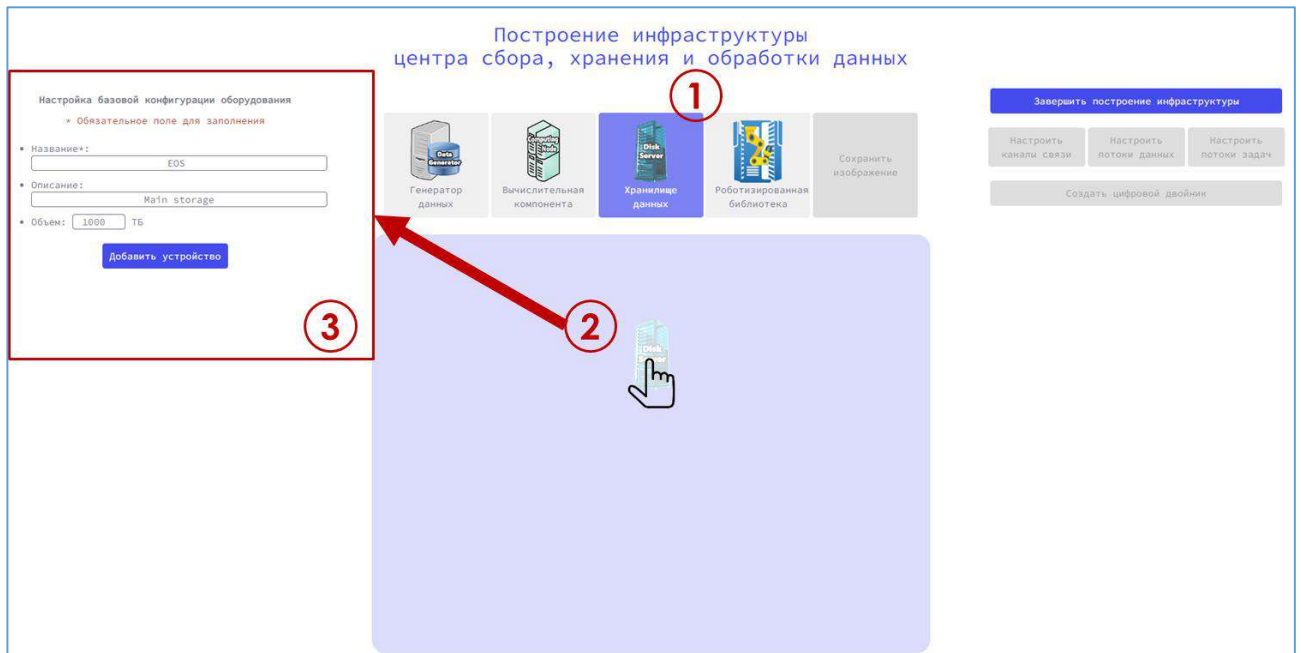


Рисунок 81. Добавление хранилища данных в инфраструктуру ЦОД и настройка его параметров



Рисунок 82. Результат построения инфраструктуры ЦОД

2. Построенную инфраструктуру можно отредактировать, например, изменить месторасположение объектов или параметры базовой конфигурации. Для этого необходимо нажать на кнопку «Редактировать инфраструктуру», затем в области для рисования выбрать объект для перемещения или изменения его параметров (см. рис. 83). Все изменения необходимо сохранить, нажав на соответствующую кнопку.




Рисунок 83. Перемещение одного из хранилищ данных в процессе редактирования инфраструктуры РЦОД

3. После того, как все необходимые объекты будут добавлены, пользователь должен настроить каналы связи между ними (см. рис. 84). Для настройки каналов следует выбрать объекты, между которыми осуществляется соединение, и задать требуемые параметры. В существующие каналы связи можно вносить изменения (см. рис. 87).




Рисунок 84. Настройка каналов связи


Построение инфраструктуры центра сбора, хранения и обработки данных




Генератор данных



Вычислительная компонента




Хранилище данных




Роботизированная библиотека

Сохранить изображение




Trigger




Buffer


100 Гб/с



NCX LHEP



Tier LIT



EOS

Редактировать инфраструктуру

Настроить каналы связи
Настроить потоки данных
Настроить потоки задач

Создать цифровой двойник

Настройка каналов связи

* Обязательное поле для заполнения

Выберите существующий канал из списка, чтобы изменить параметры


- Название*:
- Описание:
- Канал от*: до*:
- Пропускная способность: Гб/с


Сохранить изменения


Рисунок 85. Редактирование каналов связей


4. Далее требуется создать потоки данных для обработки на РЦОД (см. рис. 86) и потоки задач, которые будут обрабатывать эти данные (см. рис. 87). Для создания потоков необходимо нажать на соответствующую кнопку, после чего заполнить открывшуюся форму.


Построение инфраструктуры центра сбора, хранения и обработки данных

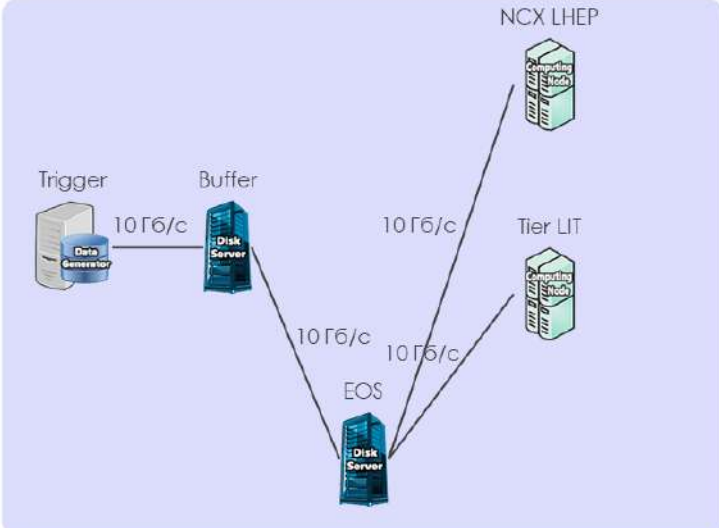

 Генератор данных


 Вычислительная компонента


 Хранилище данных


 Роботизированная библиотека


 Сохранить изображение



Редактировать инфраструктуру

Настроить каналы связи
Настроить потоки данных
Настроить потоки задач

Создать цифровой двойник

Настройка потоков данных

* Обязательное поле для заполнения

- Название*:
- Описание:
- Канал передачи данных*:
- Объем данных для передачи: %

Добавить

Рисунок 86. Настройка потоков данных

Построение инфраструктуры центра сбора, хранения и обработки данных


 Генератор данных


 Вычислительная компонента


 Хранилище данных


 Роботизированная библиотека


 Сохранить изображение



Редактировать инфраструктуру

Настроить каналы связи
Настроить потоки данных
Настроить потоки задач

Создать цифровой двойник

Настройка потоков задач

* Обязательное поле для заполнения

- Название*:
- Описание:
- Тип обрабатываемых данных*:
- Объем входных данных для задачи*: ГБ
- Объем входных данных (допустимое отклонение): ГБ
- Тип результирующих данных*:
- Объем выходных данных*: ГБ
- Объем выходных данных (допустимое отклонение): ГБ
- Распределение вероятности для генерации данных:
- Среднее время выполнения задачи*: с
- Среднее время выполнения (допустимое отклонение):
- Распределение вероятности для генерации задач:
- Общее количество задач *

Добавить

Рисунок 87. Настройка потоков задач

5. Пользователь может редактировать инфраструктуру РЦОД, базовую конфигурацию оборудования, изменять параметры потоков данных и задач до перехода на следующий этап, который заключается непосредственно в создании ЦД.

6. По нажатию на кнопку «Создать цифровой двойник» откроется окно, изображенное на рисунке 88. Пользователю необходимо добавить вычислительный эксперимент на ЦД, нажав на соответствующую кнопку, после чего откроется форма, представленная на рисунке 89. Пользователю предлагается ввести название и описание вычислительного эксперимента, параметры моделирования и логирования. Можно добавлять несколько вычислительных экспериментов для решения задач поиска требуемой конфигурации оборудования определенного типа (см. рис. 90).

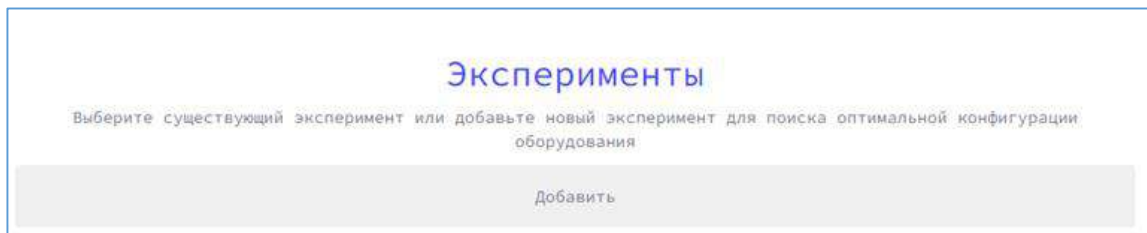


Рисунок 88. Добавление вычислительного эксперимента на ЦД

Рисунок 89. Настройка параметров вычислительного эксперимента на ЦД

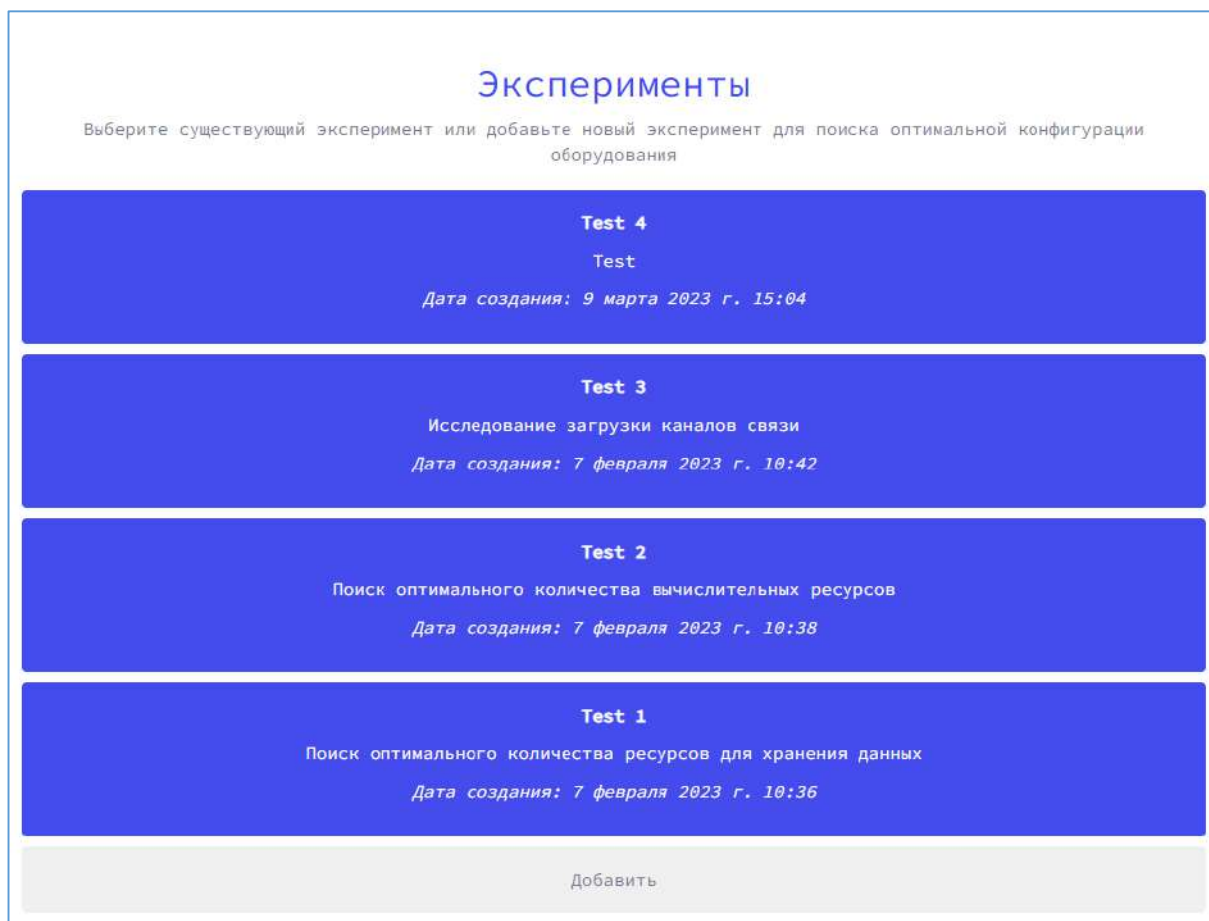


Рисунок 90. Список добавленных вычислительных экспериментов на ЦД

7. По нажатию на один из имеющихся вычислительных экспериментов откроется страница с информацией, которая отражает все параметры и базовую конфигурацию оборудования (см. рис. 91). Для добавления дополнительных модификаций оборудования необходимо нажать на кнопку «Добавить модификацию». Для добавления и настройки событий, которые могут произойти в РЦОД, необходимо нажать на кнопку «Добавить событие».

Информация об эксперименте
Дата создания: 7 февраля 2023 г., 10:36

Название эксперимента
Test 1

Описание эксперимента
Поиск оптимального количества ресурсов для хранения данных.

Параметры моделирования

- Продолжительность работы моделируемой инфраструктуры – 800 ч.
- Ускорение процесса моделирования в 1000 раз.

Параметры логирования

- Объекты моделируемой инфраструктуры
 - Хранилища данных
 - Вычислительные компоненты
 - Каналы связи
- События
 - Генерация данных
 - Потери данных
 - Работа с файлами
 - Генерация, запуск, выполнение задач

[Посмотреть результаты](#)

[Выбрать другой эксперимент](#)

Базовая конфигурация

Хранилища данных		
Название	Описание	Объем (ТБ)
trigger	Trigger BSM	10000,0
buffer	Data reception buffer	5400,0
eos1hep	Main storage LHEP	1000,0
eos1it	Main storage LIT	1000,0
dcach	pp	1000,0

Вычислительные компоненты		
Название	Описание	Количество ядер
ezlit	LIT T2 farm	500
ncslhep	LHEP main farm	1200
super	Governor	100

Каналы связи		
Название	Описание	Пропускная способность (ГБ/с)
raw0	trigger - buffer	100,0
raw1	buffer - lhep	10,0
raw2	buffer - lit	10,0
compute0	lhep - farm lhep	10,0
compute1	lit - governor	10,0
compute2	lit - farm lit	10,0
dataeoslheplit	eoslhep - eoslit	10,0
dataeoslitlhep	eoslit - eoslhep	10,0

[Добавить модификацию](#)

Список модификаций

Список событий

[Добавить событие](#)

Рисунок 91. Просмотр информации о вычислительном эксперименте на ЦД

8. После настройки вычислительных экспериментов осуществляется запуск ЦД. Если в вычислительном эксперименте существует несколько модификаций (см. рис. 92), то ЦД для каждой модификации можно запускать одновременно.

Информация об эксперименте
Дата создания: 7 февраля 2023 г., 10:36

Название эксперимента
Test 1

Описание эксперимента
Поиск оптимального количества ресурсов для хранения данных.

Параметры моделирования

- Продолжительность работы моделируемой инфраструктуры – 800 ч.
- Ускорение процесса моделирования в 1000 раз.

Параметры логирования

- Объекты моделируемой инфраструктуры
 - Хранилища данных
 - Вычислительные компоненты
 - Каналы связи
- События
 - Генерация данных
 - Потери данных
 - Работа с файлами
 - Генерация, запуск, выполнение задач

[Посмотреть результаты](#)

[Выбрать другой эксперимент](#)

Базовая конфигурация

Хранилища данных		
Название	Описание	Объем (ТБ)
trigger	Trigger BSM	10000,0
buffer	Data reception buffer	5400,0
eos1hep	Main storage LHEP	1000,0
eos1it	Main storage LIT	1000,0
dcach	pp	1000,0

Вычислительные компоненты		
Название	Описание	Количество ядер
ezlit	LIT T2 farm	500
ncslhep	LHEP main farm	1200
super	Governor	100

Каналы связи		
Название	Описание	Пропускная способность (ГБ/с)
raw0	trigger - buffer	100,0
raw1	buffer - lhep	10,0
raw2	buffer - lit	10,0
compute0	lhep - farm lhep	10,0
compute1	lit - governor	10,0
compute2	lit - farm lit	10,0
dataeoslheplit	eoslhep - eoslit	10,0
dataeoslitlhep	eoslit - eoslhep	10,0

[Добавить модификацию](#)

Список модификаций

#	Статус	Дата обновления	
16	NEW	9 марта 2023 г., 14:52	Просмотр Запуск Результаты
15	NEW	9 марта 2023 г., 14:52	Просмотр Запуск Результаты

Список событий

Название	Описание	
decrease	уменьшение количества ядер	Подробнее

[Добавить событие](#)

Рисунок 92. Информация о вычислительном эксперименте со списком доступных модификаций и происходящих в РЦОД событий

9. В процессе или по завершении работы ЦД пользователю доступны результаты для просмотра. По нажатию на кнопку «Посмотреть результаты» откроется страница, где следует выбрать тип интересующего оборудования, после чего будут построены интерактивные графики. Например, на рисунке 93 представлена страница с графиками, отражающими объемы данных в

хранилищах, а на рисунке 94 — нагрузку на каналы связи во время передачи данных. Для просмотра доступны также графики распределений различных типов файлов в хранилищах и использования вычислительных компонент. Интерактивные графики подразумевают возможность масштабирования и выбора данных для просмотра, т.е. скрывание результатов одной или нескольких модификаций (см. рис. 94). Пользователь может сохранить результаты работы ЦД для какой-либо модификации в виде изображений с графиками.



Рисунок 93. Просмотр результатов работы ЦД – общий объем данных в хранилищах



Рисунок 94. Просмотр результатов работы ЦД – загрузка каналов передачи данных

Приложение 3. Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ
№ 2023667305

**Программный комплекс для создания цифровых
двойников распределенных центров сбора, хранения и
обработки данных**

Правообладатель: **Объединенный Институт Ядерных
Исследований (RU)**

Авторы: **Кореньков Владимир Васильевич (RU), Пряхина
Дарья Игоревна (RU), Трофимов Владимир Валентинович
(RU)**

Заявка № **2023665363**
Дата поступления **20 июля 2023 г.**
Дата государственной регистрации
в Реестре программы для ЭВМ **14 августа 2023 г.**

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов



РОССИЙСКАЯ ФЕДЕРАЦИЯ



RU2023667305

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2023667305
Дата регистрации: 14.08.2023
Номер и дата поступления заявки:
2023665363 20.07.2023
Дата публикации и номер бюллетеня:
14.08.2023 Бюл. № 8
Контактные реквизиты:
Нет

Автор(ы):
Кореньков Владимир Васильевич (RU),
Пряхина Дарья Игоревна (RU),
Трофимов Владимир Валентинович (RU)
Правообладатель(и):
Объединенный Институт Ядерных
Исследований (RU)

Название программы для ЭВМ:

Программный комплекс для создания цифровых двойников распределенных центров сбора, хранения и обработки данных

Реферат:

Программный комплекс предназначен для создания цифровых двойников (ЦД) распределенных центров сбора, хранения и обработки данных (РЦОД). Созданные ЦД могут применяться для решения задач построения, совершенствования и развития РЦОД. Основным компонентом программного комплекса является алгоритм, реализующий моделирование распределенных центров с учетом характеристик потоков данных и задач для хранения и обработки, а также вероятностей появления изменений в процессах, происходящих в РЦОД. В программе также реализованы процессы получения информации о РЦОД (архитектура, параметры оборудования, характеристики потоков данных и задач, события, происходящих в РЦОД, сценарии масштабирования системы), структурирования и сохранения результатов работы ЦД. Для пользователя разработана возможность графического построения инфраструктуры РЦОД, запуска ЦД и визуализации результатов его работы. Тип ЭВМ: IBM PC-совмест. ПК на базе процессора Intel/AMD; ОС: Windows 7 и выше; Linux.

Язык программирования:

Python 3.7 с использованием библиотек argparse, json, matplotlib, numpy, plotly, psycopg2, scipy и фреймворка Django; JavaScript, SQL

Объем программы для ЭВМ:

133,14 МБ

Приложение 4. Письма о применении результатов диссертационного исследования

ПИСЬМО

о применении результатов диссертационной работы Пряжиной Д. И.
«Цифровые двойники для решения задач управления и развития распределенных
центров сбора, хранения и обработки данных»
в эксперименте VM@N проекта NICA

В диссертации на соискание ученой степени кандидата технических наук Пряжиной Д. И. «Цифровые двойники для решения задач управления и развития распределенных центров сбора, хранения и обработки данных» разработан метод построения и использования цифровых двойников распределенных центров сбора, хранения и обработки данных. Актуальность работы подтверждается необходимостью создания цифровых двойников для проектирования аппаратных решений, оценки эффективности функционирования, а также поиска проблемных точек сложных систем сбора, хранения и обработки данных, которыми являются крупные распределенные вычислительные инфраструктуры современных научных проектов.

Предложенный в диссертационной работе подход направлен на моделирование распределенных вычислительных систем, оперирующих большими объемами данных. Программа моделирования, которая является основным элементом цифрового двойника, учитывает характеристики потоков данных и задач, вероятности сбоя и изменения в производительности используемой аппаратной части распределенных систем обработки и хранения данных. Важным результатом диссертационной работы является специальное программное обеспечение для создания цифровых двойников, которое может применяться для широкого класса задач в области проектирования, построения и развития распределенных систем, в том числе с целью выбора наиболее целесообразных вариантов из множества рассматриваемых конфигураций аппаратных платформ.

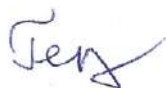
Достоверность подхода и выводов диссертационного исследования подтверждается практикой применения разработанных методов при анализе и прогнозировании работы вычислительной инфраструктуры крупного эксперимента в области физики высоких энергий. Верификация компонентов программного обеспечения проходила в процессе создания цифрового двойника вычислительной инфраструктуры эксперимента VM@N проекта NICA Объединенного института ядерных исследований. Результаты, полученные разработанной в диссертации моделью, показали хорошую согласованность с практическими значениями.

Реализованный метод построения цифрового двойника позволил как описать вычислительную инфраструктуру эксперимента VM@N, учитывая реальные потоки данных и задач, так и применить его для оценки характеристик работы имеющихся

центров обработки при решении задач эксперимента на выбранной инфраструктуре. Результаты работы цифрового двойника, построенного с помощью разработанного специального программного обеспечения, позволили оценить требуемое количество ресурсов для обработки данных с учетом планируемых параметров потоков данных будущих сеансов эксперимента VM@N, проведена оценка времени, необходимого для полной обработки экспериментальных данных по окончании сеанса эксперимента, как с учетом, так и без учета дополнительной обработки имеющихся моделированных данных. В дальнейшем специальное программное обеспечение планируется применять для формирования требований по развитию вычислительной инфраструктуры эксперимента VM@N.

Пряхина Д. И. регулярно представляет для обсуждения и формирования дальнейшей стратегии текущие результаты данной работы на коллаборационных совещаниях эксперимента VM@N.

Выражаю свою поддержку диссертационной работе Д. И. Пряхиной.



к.т.н., Герценбергер Константин Викторович
координатор разработки программного обеспечения
эксперимента VM@N проекта NICA,
начальник группы математического и программного обеспечения
научно-экспериментального отдела физики столкновений
тяжелых ионов на комплексе NICA,
Лаборатория физики высоких энергий им. В.И. Векслера и А.М. Балдина,
Объединенный институт ядерных исследований

« 12 » 02 2024 г.

ПИСЬМО

о применении результатов диссертационной работы Пряхиной Д.И. «Цифровые двойники для решения задач управления и развития распределенных центров сбора, хранения и обработки данных» в эксперименте SPD комплекса NICA

В работах, легших в основу диссертации на соискание ученой степени кандидата технических наук Пряхиной Д.И. «Цифровые двойники для решения задач управления и развития распределенных центров сбора, хранения и обработки данных», был разработан и реализован программный инструментарий для создания цифровых двойников распределенных центров сбора, хранения и обработки данных. Распределенная обработка играет ключевую роль в получении научных результатов крупных экспериментов класса «мегасайенс». Хорошим примером являются эксперименты на ускорительном комплексе NICA. В рамках проектирования физической установки, а так же в процессе проводимых исследований эксперименты производят большой объем данных, для хранения и своевременной обработки которых используются крупные вычислительные системы. К таким комбинированным системам предъявляются достаточно высокие требования по производительности при оптимальных затратах, поэтому работа Пряхиной Д.И. по созданию цифровых двойников, которые помогают обеспечить качественное проектирование, постоянное совершенствование и масштабирование систем хранения и обработки данных, является очень актуальной.

Разработанный в диссертации новый метод построения и использования цифровых двойников сложных систем отличается возможностью моделировать процессы обработки данных, учитывая характеристики потоков данных и вычислительных задач, вероятностей сбоев, отказов и изменений в производительности различных составляющих распределенной системы. Эта особенность является существенным преимуществом работы, так как позволяет применять разработку при принятии решений по улучшению технических характеристик сложных вычислительных систем.

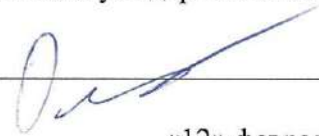
Экспериментальная установка SPD комплекса NICA находится в стадии создания, одновременно реализуется и система первичной обработки данных эксперимент: SPD online filter. Проектирование специализированной вычислительной системы для высокопропускной обработки огромного объема, включает в себя анализ архитектуры и прогнозирование затрат на оборудование, поэтому разработанный Пряхиной Д.И. программный комплекс используется для создания цифрового двойника SPD Online filter. Главная задача цифрового двойника — моделирование подсистем с разными вариантами параметров оборудования, потоков данных и типов обработки. Результаты работы цифрового двойника на данный момент позволили первоначально оценить требуемые параметры оборудования для специализированной системы предварительной обработки с учетом планируемых характеристик потоков данных эксперимента SPD комплекса NICA. Применение результатов диссертационного исследования Пряхиной Д.И. в обозримом будущем позволит обеспечить качественное развитие и поддержку вычислительной инфраструктуры эксперимента SPD.

Полученные результаты первого варианта цифрового двойника и планы развития данного направления получили одобрение на совещании коллаборации эксперимента SPD в

2023 году (VI SPD Collaboration Meeting and Workshop on Information Technology in Natural Sciences).

Выражаю поддержку диссертационной работе Пряхиной Д.И.

к.т.н., Олейник Данила Анатольевич
заместитель координатора разработки программного обеспечения
эксперимента SPD комплекса NICA,
старший научный сотрудник научно-технического отдела внешних коммуникаций и
распределенных информационных систем,
Лаборатория информационных технологий им. М.Г. Мещерякова,
Объединенный институт ядерных исследований



«12» февраля 2024 г.

Приложение 5. Акт о внедрении результатов диссертационного исследования в учебный процесс

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Университет «Дубна»

(государственный университет «Дубна»)



УТВЕРЖДАЮ

И.о. проректора по учебной работе
Государственного университета «Дубна»

О.А. Крейдер

«12» 02 2024 г.



АКТ

о внедрении результатов диссертационной работы Пряхиной Д.И. «Цифровые двойники для решения задач управления и развития распределенных центров сбора, хранения и обработки данных» в Институте системного анализа и управления (ИСАУ) Федерального государственного бюджетного образовательного учреждения высшего образования «Университет «Дубна»

Комиссия в составе:

д.т.н., проф. Е.Н. Черемисина, научный руководитель ИСАУ, зав. каф. системного анализа и управления,

к.т.н., доц. Е.Ю. Кирпичева, директор ИСАУ,

к.ф.-м.н., доц. Н.А. Токарева, зав. каф. информационных технологий,

к.т.н., О.Ю. Тятюшкина, доцент каф. системного анализа и управления,

П.П. Сычев, доцент каф. распределенных информационно-вычислительных систем

составила настоящий акт о внедрении результатов диссертационного исследования Пряхиной Д.И. в учебный процесс Государственного университета «Дубна». Развитый в диссертации метод построения и использования цифровых двойников распределенных центров сбора, хранения и обработки данных используется при проведении лекций и семинарских занятий для студентов магистратуры по направлению 27.04.03 Системный анализ и управление по профилю «Цифровые платформы и аналитика больших данных» в курсе «Распределенные вычисления и облачные технологии».

Использование в учебном процессе методических рекомендаций и выводов диссертации Пряхиной Д.И. в части подхода к разработке модели распределенных вычислительных систем с учетом характеристик потоков задач и данных для хранения и обработки, а также вероятностей появления изменений в процессах, происходящих в таких системах, позволяет повысить уровень профессиональных компетенций студентов в решении задач производственной и технологической деятельности с учетом современных достижений науки и техники.

Черемисина Е.Н.

Научный руководитель ИСАУ,
зав. каф. системного анализа и управления,
проф., д.т.н.



Кирпичева Е.Ю.

Директор ИСАУ,
доц., к.т.н.



Токарева Н.А.

зав. каф. информационных технологий,
доц., к.ф.-м.н.



Тятюшкина О.Ю.

доцент каф. системного анализа и управления,
к.т.н.



Сычев П.П.

доцент каф. распределенных информационно-вычислительных систем

