

Автоматизация обработки потока данных эксперимента VM@N

на основе платформы оркестрации задач

**Никита Ильин, магистрант, ГУ «Дубна»
Константин Герценбергер, к.т.н.,
начальник группы, ЛФВЭ ОИЯИ**

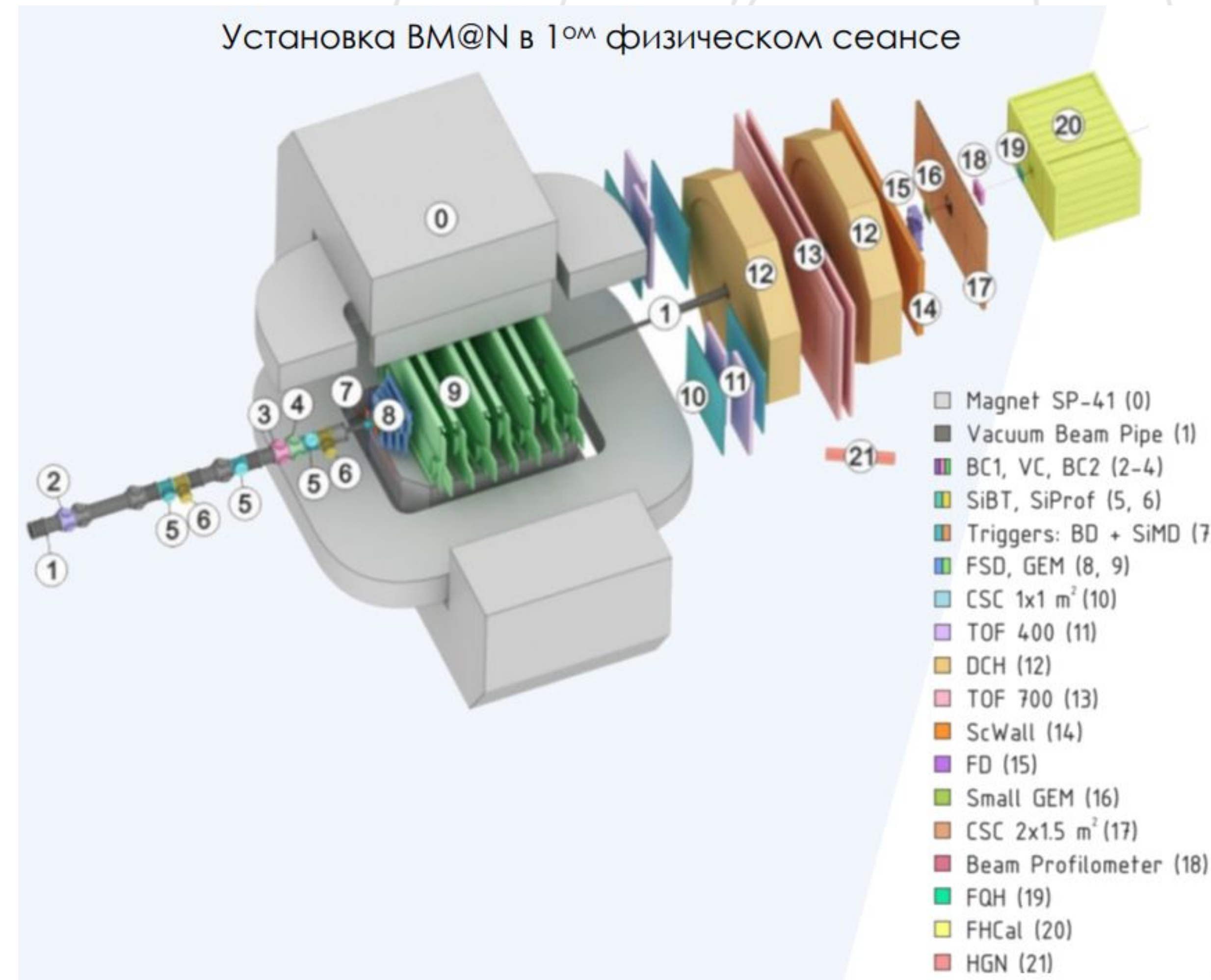
BM@N – 1ый эксперимент проекта NICA

BM@N (Baryonic Matter at Nuclotron) – первый эксперимент ускорительного комплекса NICA на фиксированной мишени на выведенном с Нуклотрона пучке. Полученные в ходе эксперимента данные позволяют детально исследовать свойства сверхплотной ядерной материи, образующейся в ядро-ядерных столкновениях при энергиях 1-6 АГэВ.



#1ый физический сеанс VM@N зима 2022/2023 год

Название метрики	Значение
Продолжительность сеанса	50 дн
Время работы (сбора данных) детектора	720 ч
Размер файла необработанных данных	15 ГБ
Суммарный объем необработанных данных за сеанс	400 ТБ
Скорость набора данных (средняя по всему сеансу)	300 МБ/с



Постановка задачи

Необходимо осуществить копирование полученных необработанных данных эксперимента VM@N с онлайн фермы на хранилища данных кластера NISA и МИВК ОИЯИ, запустить их распределенную обработку, обеспечить копирование на ленты для архивного хранения.

Для этого требуется описать выполняющиеся бизнес-процессы, выбрать инструменты и методы автоматизации обработки потока задач, развернуть систему для оркестрации задач и их автоматизации, описать потоки задач копирования и обработки данных эксперимента, провести тестирование разработанной системы.

В ходе копирования данных необходимо проверять целостность файлов по контрольной сумме. Помимо этого, требуется реализовать систему мониторинга выполнения потока задач и рассылку уведомлений о завершении важных этапов обработки данных эксперимента.



Начальная диаграмма реализуемого потока данных



Автоматизация с помощью системы оркестрации задач

Оркестраторы – это инструменты, которые управляют периодическими и вызываемыми по событиям задачами в системах обработки данных. Такие системы позволяют: планировать задачи с учетом различных условий запуска, выполнять репроцессинг, осуществлять мониторинг, управлять ресурсами аппаратной платформы.

Настройка и выполнение задач происходит посредством работы с графическим интерфейсом и / или написанием программных компонентов на поддерживаемом системой языке программирования.



Сводная таблица инструментов для оркестрации задач

Название системы	Простота	Популярность	яп	Безопасность	Отказоустойчивость	Масштабируемость	GUI
Prefect	+	14,4	Python	+	-	+	+
Airflow	+	34,3	Python	+	+	+	+
Dagster	+	10,1	Python	+	+	-	+
NiFi	-	4,4	Java	+	+	+	+
AWS Step Functions	-	1	Python, C++, etc.	+	+	+	+
Luigi	+	17,3	Python	-	-	+	+

* Простота – простота установки, настройки и эксплуатации. Популярность – количество звезд на GitHub. Безопасность – защищенные протоколы, наличие уязвимостей, Отказоустойчивость – возможно сохранения / продолжения работы при отказе отдельных компонентов

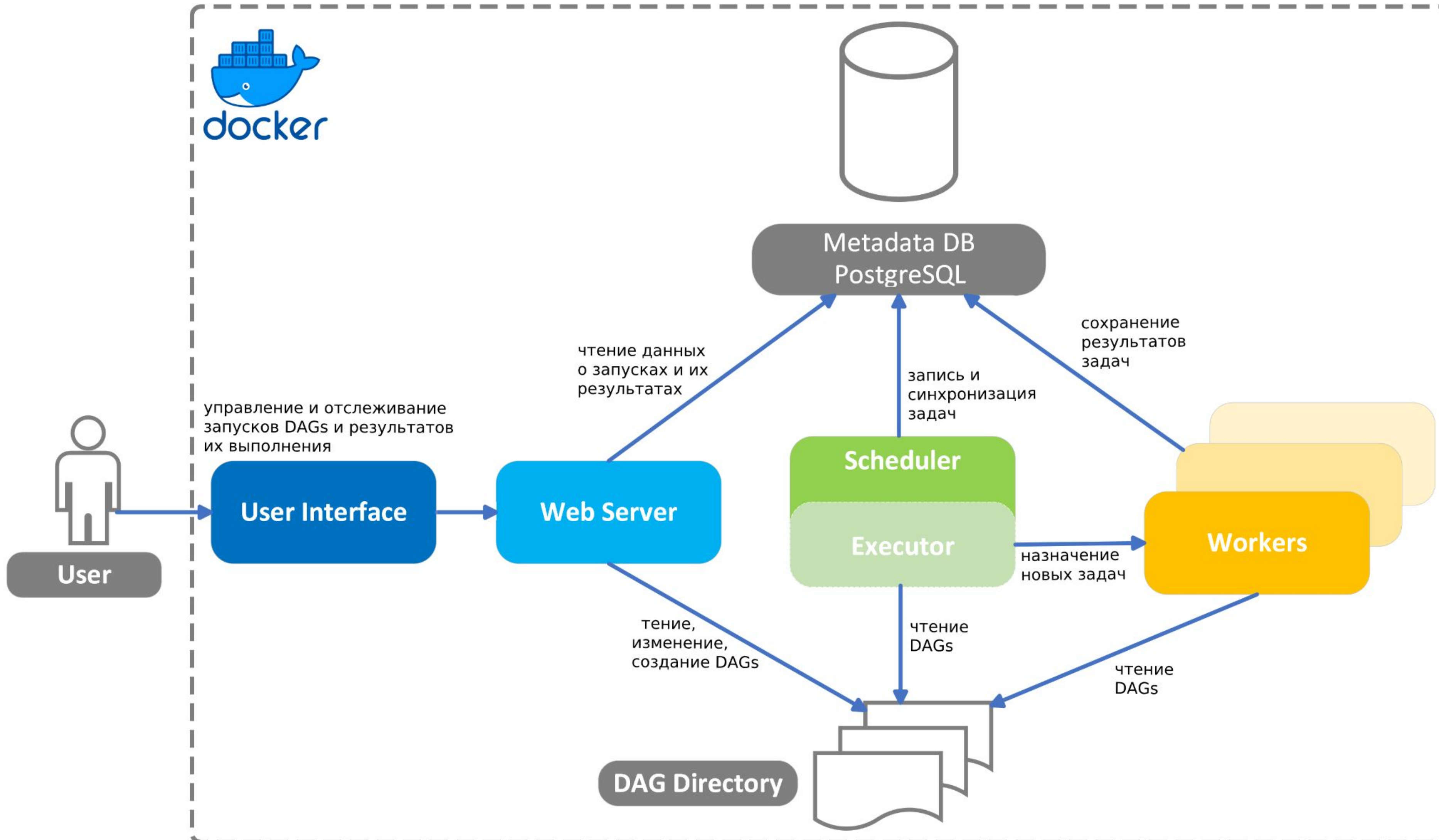
Используемые инструменты

- >> Apache Airflow – выбранная система оркестрации задач
- >> ЯП Python – основной язык разработки в Airflow
- >> Celery – инструмент, реализующий асинхронную очередь задач, часть экосистемы Apache Airflow
- >> PostgreSQL – выбранная СУБД для функционирования БД Airflow, часть экосистемы Apache Airflow
- >> SQLAlchemy – ORM* для взаимодействия с разрабатываемой БД через Python для работы со статусами, системной информации о файлах, а также истории копирования
- >> Docker – платформа контейнеризации для автоматизации развертывания и управления приложениями в различных средах

* ORM (Object-Relational Mapping) – технология, которая связывает БД с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную БД».



Развернутая архитектура



* DAG (Directed Acyclic Graph) – направленный ациклический граф; объединяет задачи, упорядоченные с помощью зависимостей и взаимосвязей, чтобы указать, как они должны выполняться.

Веб-интерфейс развернутой системы автоматизации

Основная панель для запуска / остановки реализованного DAG, отслеживания текущих запусков копирования данных, расписания, даты последнего и следующего старта, статус выполнения, а также меню расширенного функционала.

The screenshot displays the Apache Airflow web interface. At the top, there is a navigation bar with the Airflow logo and menu items: DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The current time is 20:13 UTC, and there is an AA icon for accessibility.

The main section is titled "DAGs". Below the title, there are filters for DAG status: All (55), Active (2), Paused (53), Running (9), and Failed (0). There is also a "Filter DAGs by tag" input field, a "Search DAGs" search bar, and an "Auto-refresh" toggle switch.

The DAGs list is shown in a table with the following columns: DAG, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, Actions, and Links. Two DAGs are visible:

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
creating_files_dag	tunsmm	1553 (3 active, 1 failed)	0:10:00	2023-11-08, 23:07:20	2023-11-08, 21:57:20	3	▶ 🗑️	⋮
monitoring_dag	tunsmm	248 (7 active)	0:05:00	2023-11-02, 01:34:00	2023-11-02, 00:34:00	5 (2 active)	▶ 🗑️	⋮

At the bottom left, there is a pagination control showing page 1 of 1. At the bottom right, it says "Showing 1-2 of 2 DAGs".

Веб-интерфейс для мониторинга работы реализованного графа

The screenshot displays the Airflow web interface for monitoring a DAG named 'monitoring_dag'. The interface includes a navigation bar with options like DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The current time is 21:05 UTC. The DAG's schedule is 0:31:00, and the next run is scheduled for 2023-11-02, 02:45:00. The interface is set to 'Grid' view, showing a list of runs for the task 'task_monitoring'. The runs are color-coded by status: deferred (purple), failed (red), queued (grey), removed (grey), restarting (pink), running (green), scheduled (orange), skipped (pink), success (green), up_for_reschedule (cyan), up_for_retry (yellow), upstream_failed (orange), and no_status (grey). A bar chart shows the duration of the runs, with the longest run lasting 10d00:51:42. The right-hand panel provides a summary of the DAG runs, including the total number of runs displayed (25), total success (12), total running (13), first run start (2024-03-30, 20:13:10 UTC), last run start (2024-04-09, 21:03:45 UTC), max run duration (10d00:51:42), mean run duration (4d19:37:03), and min run duration (00:00:22).

DAG: monitoring_dag DAG for monitoring files

Schedule: 0:31:00 Next Run ID: 2023-11-02, 02:45:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details Code Audit Log

09.04.2024 21:05:21 25 All Run Types All Run States Clear Filters Auto-refresh

Press **shift** + **/** for Shortcuts

deferred failed queued removed restarting running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

DAG monitoring_dag

Details Graph Gantt Code

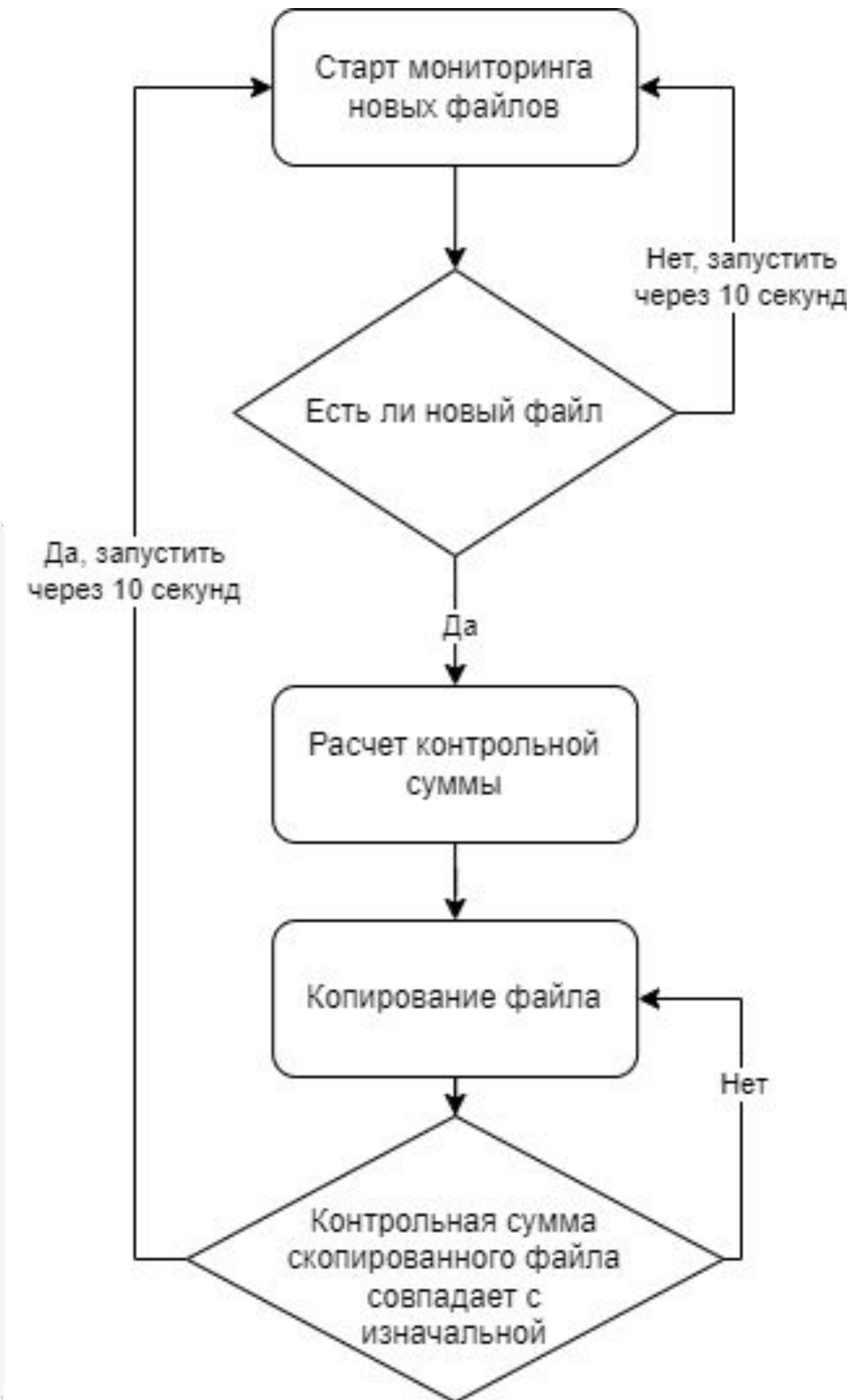
DAG Runs Summary

Total Runs Displayed	25
Total success	12
Total running	13
First Run Start	2024-03-30, 20:13:10 UTC
Last Run Start	2024-04-09, 21:03:45 UTC
Max Run Duration	10d00:51:42
Mean Run Duration	4d19:37:03
Min Run Duration	00:00:22

DAG Summary

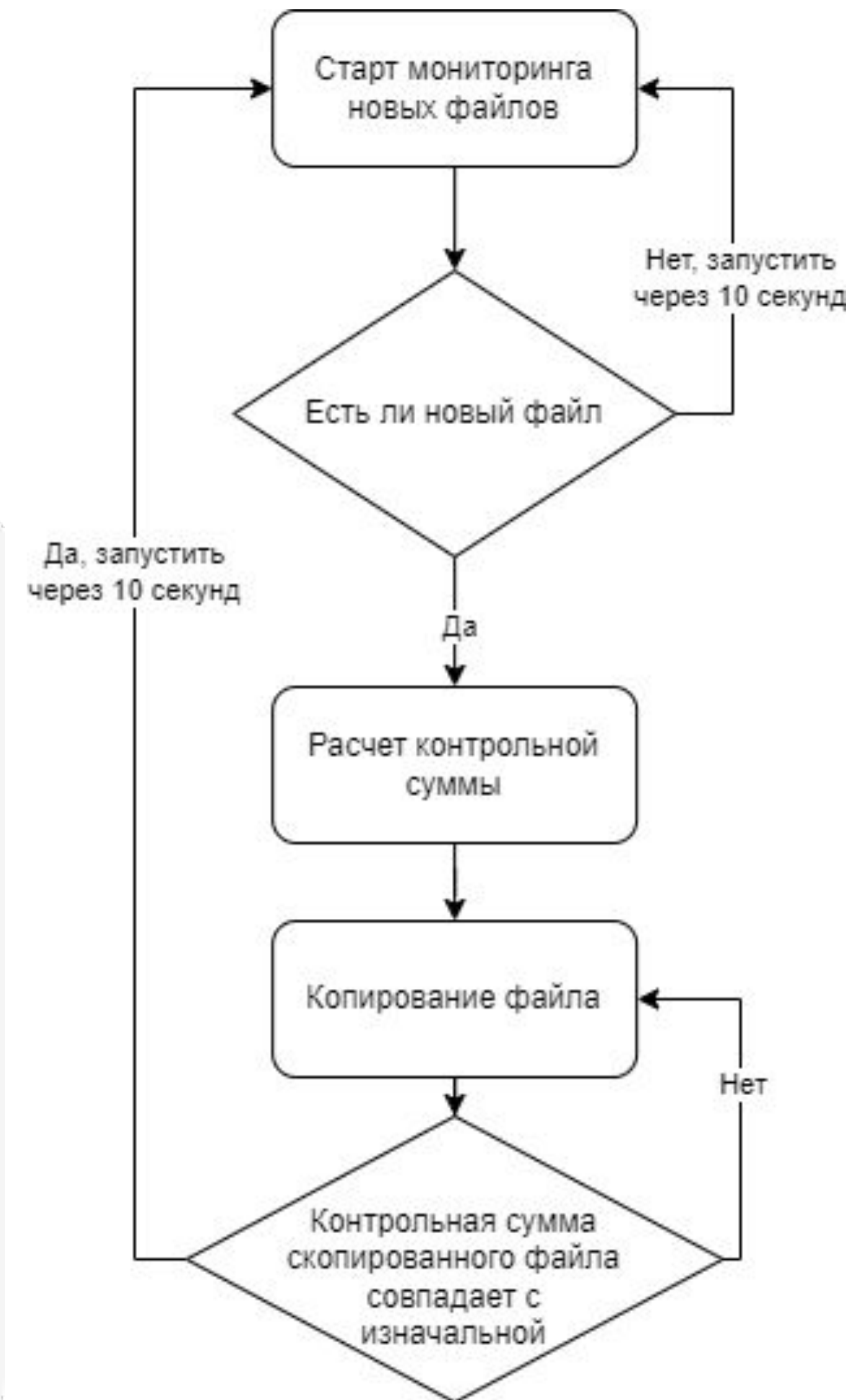
Поток задач для реализации копирования

1. Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;
2. Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;
3. Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;
4. Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.
5. Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.



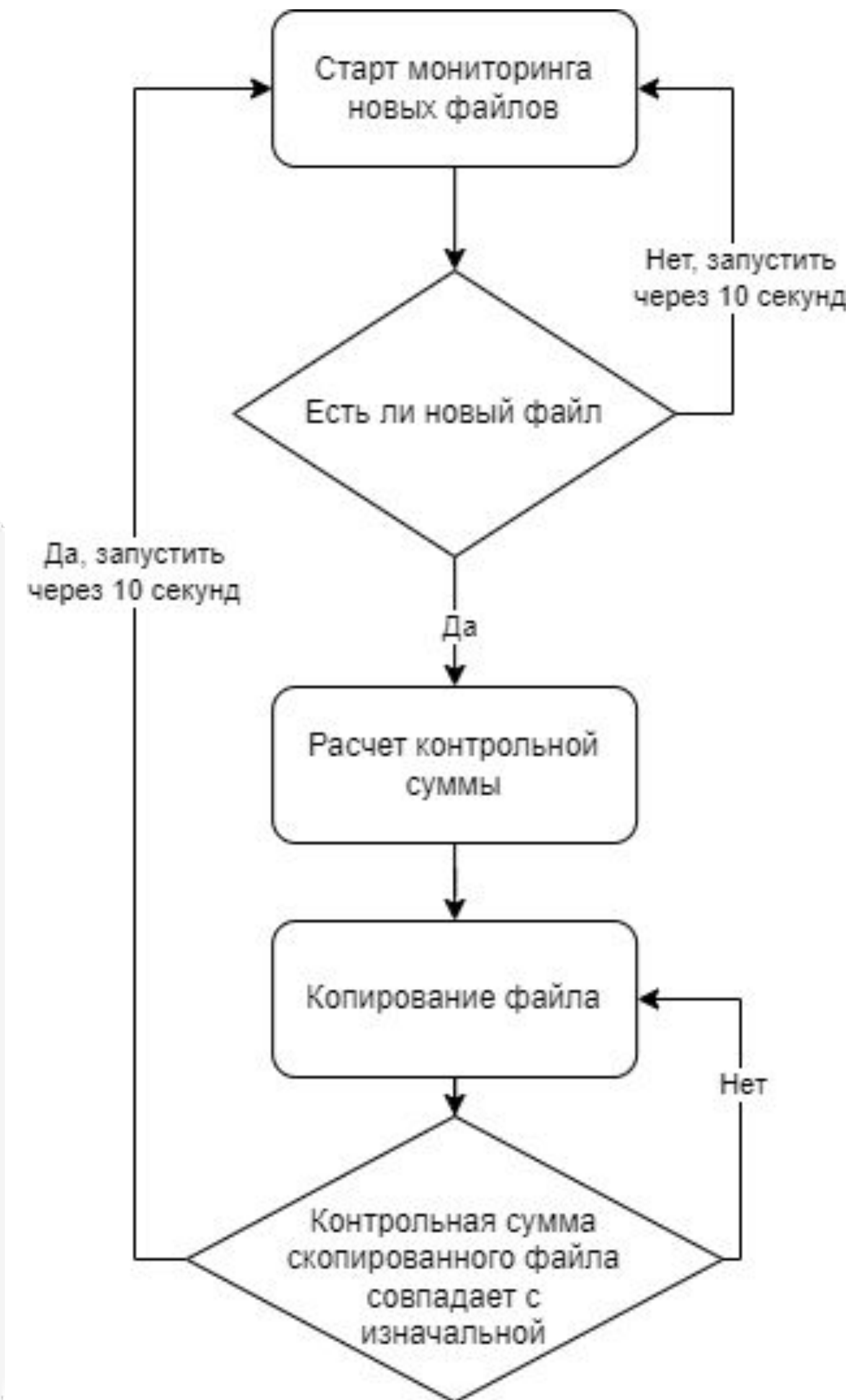
Поток задач для реализации копирования

1. **Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;**
2. Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;
3. Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;
4. Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.
5. Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.



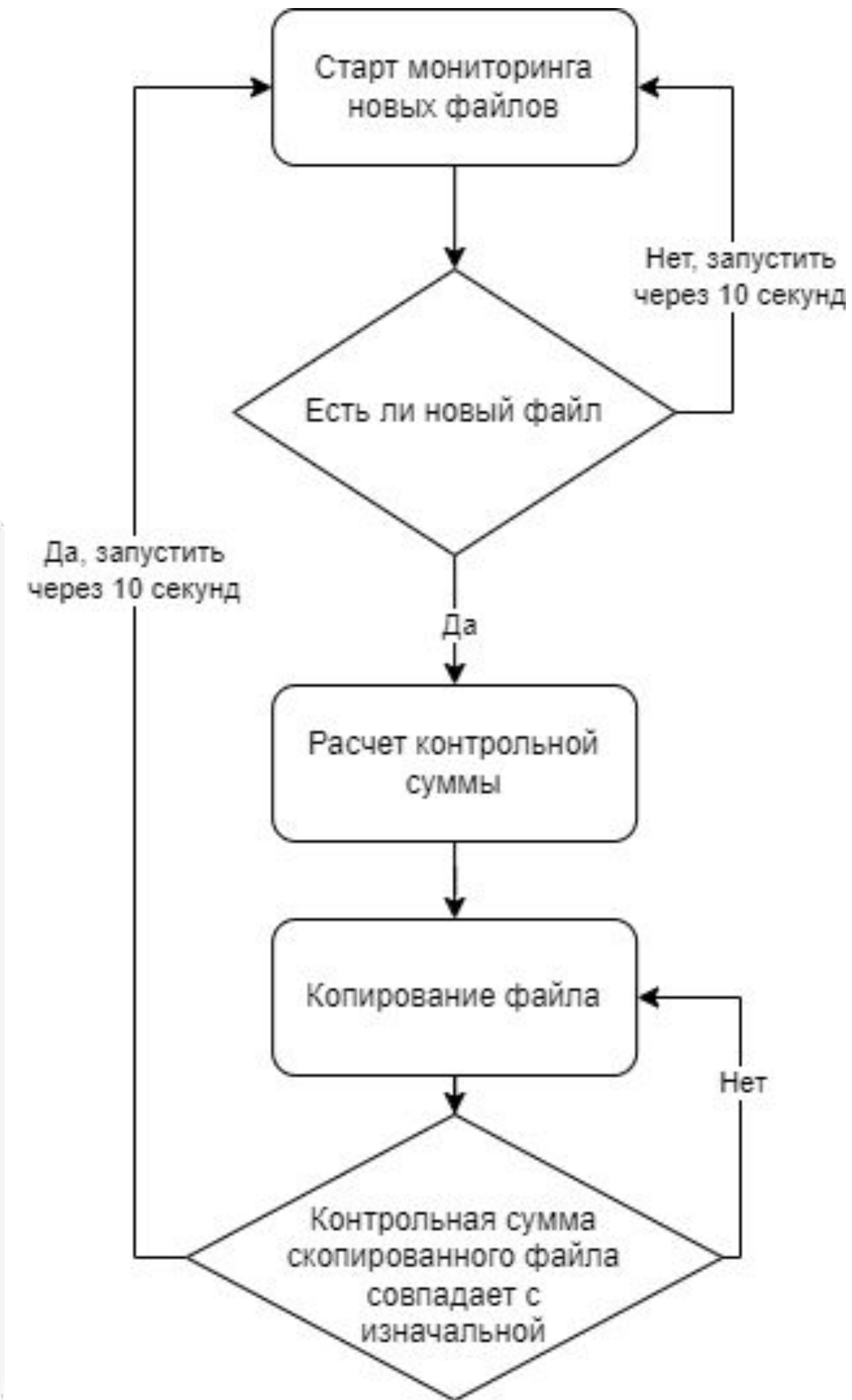
Поток задач для реализации копирования

1. Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;
2. **Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;**
3. Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;
4. Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.
5. Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.



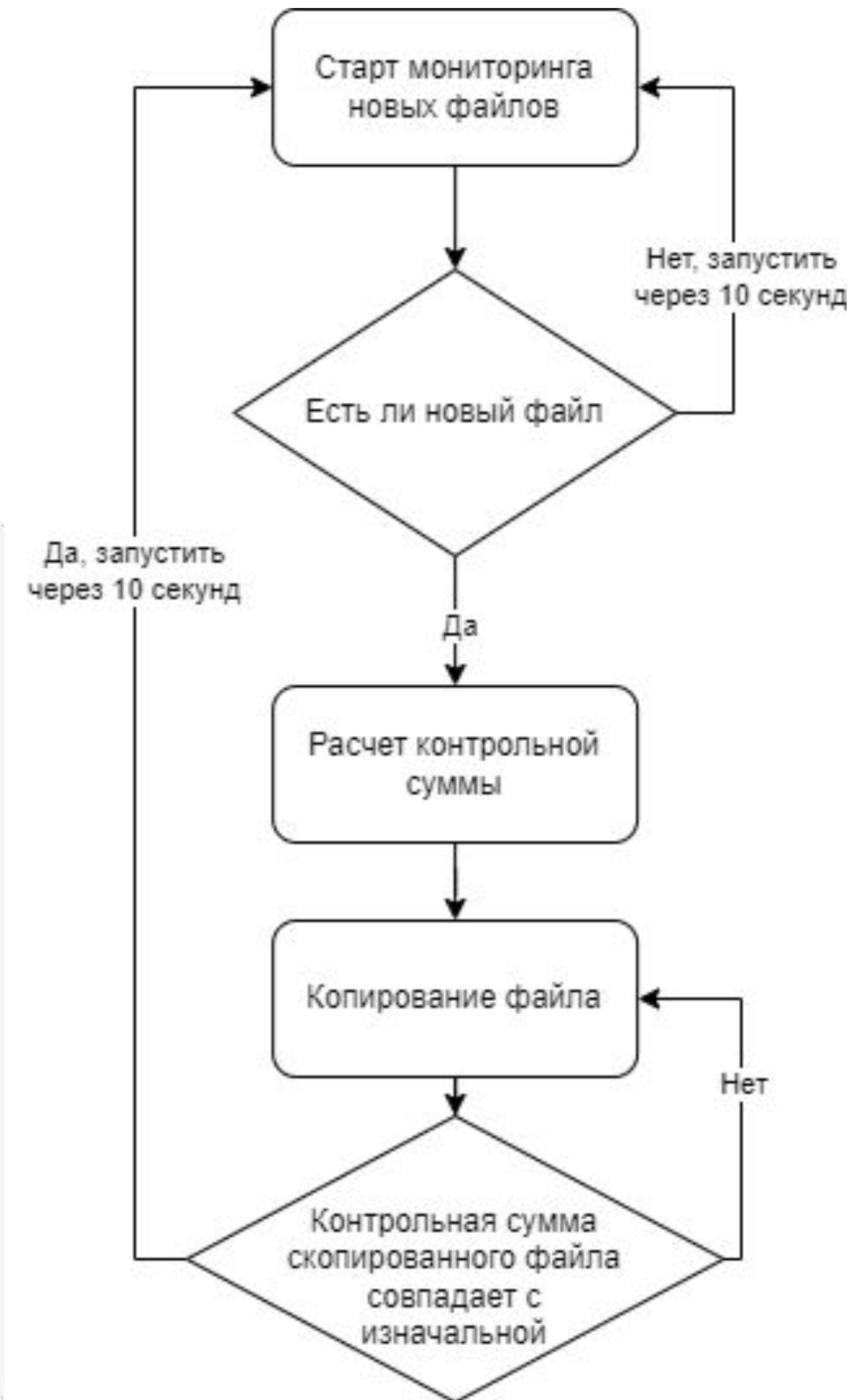
Поток задач для реализации копирования

1. Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;
2. Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;
3. **Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;**
4. Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.
5. Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.



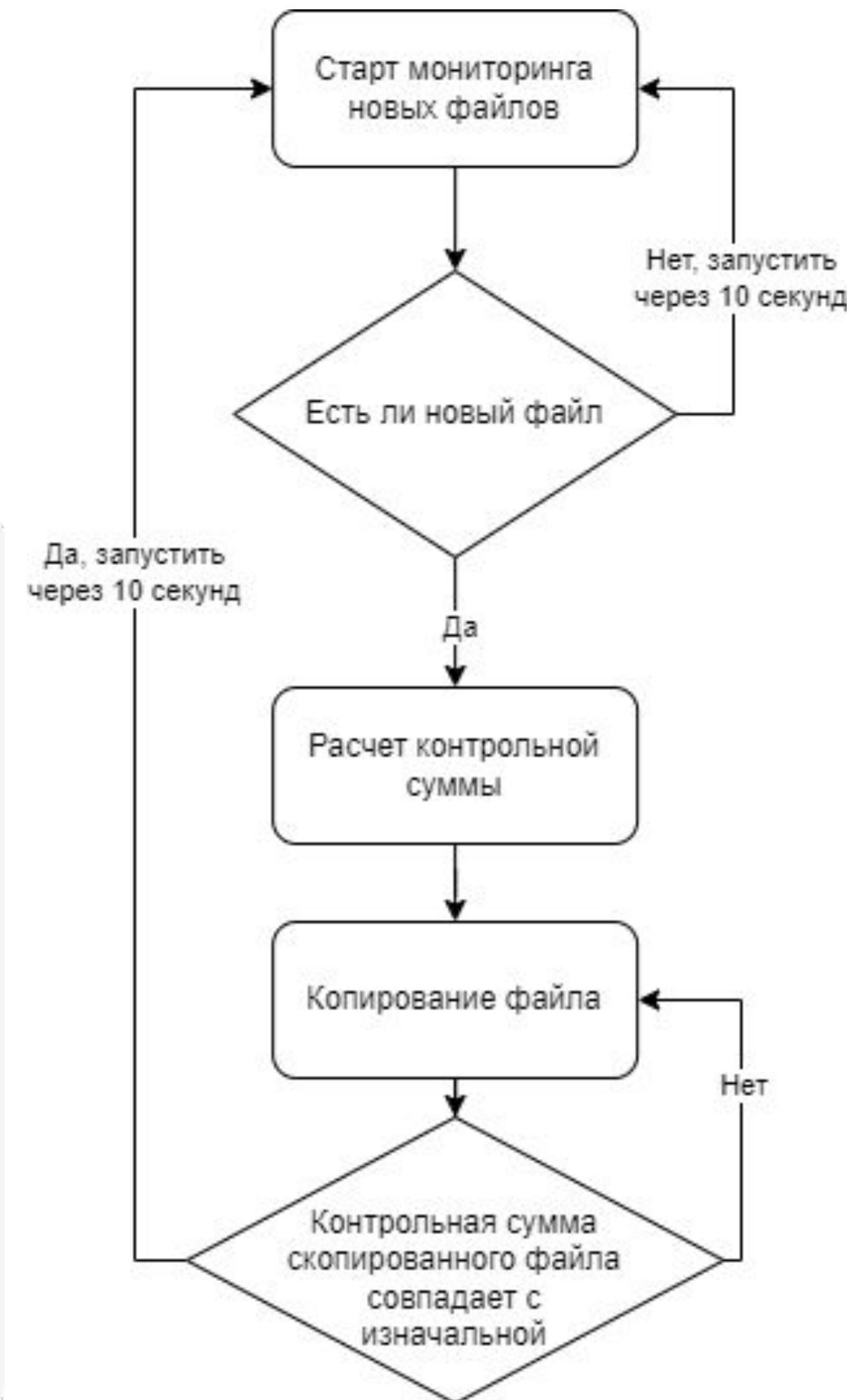
Поток задач для реализации копирования

1. Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;
2. Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;
3. Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;
4. **Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.**
5. Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.



Поток задач для реализации копирования

1. Мониторинг наличия новых файлов в онлайн ферме каждые 10 секунд;
2. Расчет контрольной суммы новых файлов через применение алгоритма Adler-32, сохранение расчета в разрабатываемую БД;
3. Запуск многопоточного копирования через набор потоков (*ThreadPool*) на хранилища данных кластера NICA и МИВК ОИЯИ посредством сетевого протокола *SSH*;
4. Расчет и сверка контрольной суммы на конечных серверах со значением в разрабатываемой БД. Если значения не совпадают, отправка запроса на повторное копирование таких файлов.
5. **Логирование системной информации о процессе работы, запись статуса работы с файлом в разрабатываемую БД, а также истории его копирования. Отправка уведомления в Телеграм.**



Тестирование компонентов

1. Модульное тестирование работоспособности компонентов разрабатываемой системы.
2. Нагрузочное тестирование на локальном сервере: DAG по созданию файлов по 2 ГБ каждые 5 секунд.



Дальнейшие шаги

1

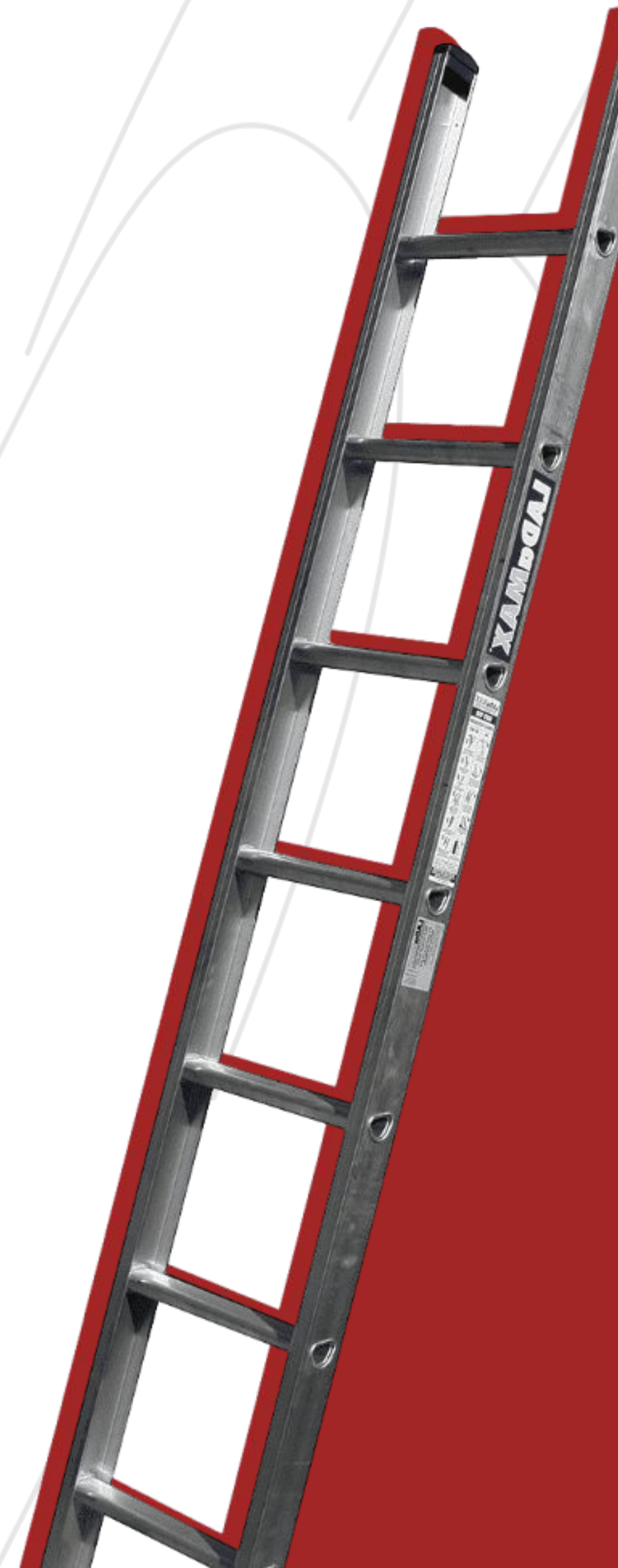
Развертывание прототипа разрабатываемой системы в облаке, тестирование и отладка процесса копирования данных VM@N

2

Завершение разработки системы логирования и уведомления о завершении важных этапов обработки

3

Развитие системы автоматизации с целью добавления в развернутую систему оркестрации задач распределенной обработки данных и их копирования на ленты для архивного хранения



Спасибо за внимание



Никита Ильин

Telegram

t.me/tunsmm



Никита Ильин

Вконтакте

vk.com/tunsmm