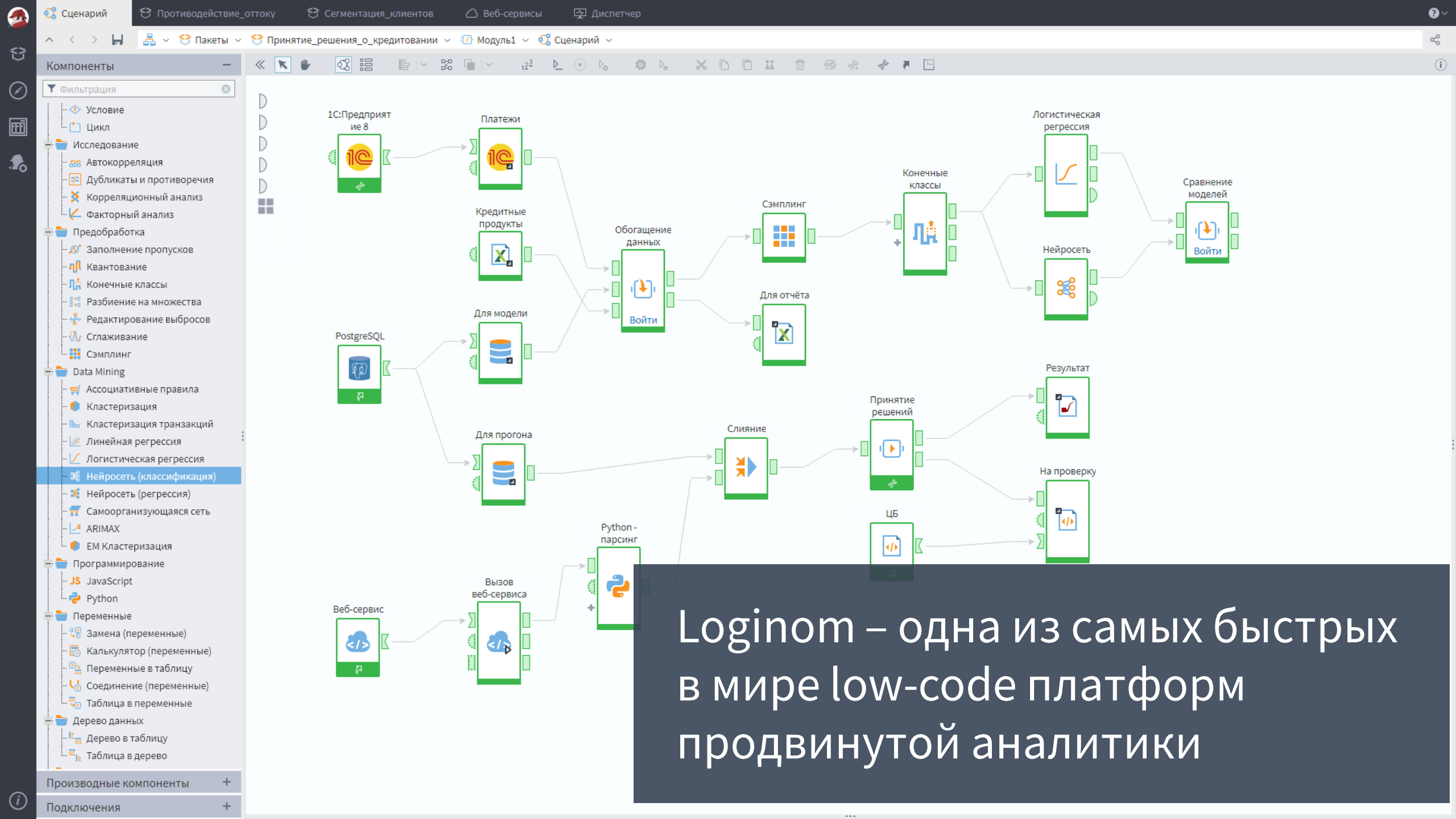
Loginom

Loginom – low-code аналитика на предельной скорости

Алексей Арустамов

Скорость – единственное свойство программ по которому есть консенсус: чем быстрее, тем лучше.

Высокая скорость должна обеспечиваться без усилий, «из коробки».



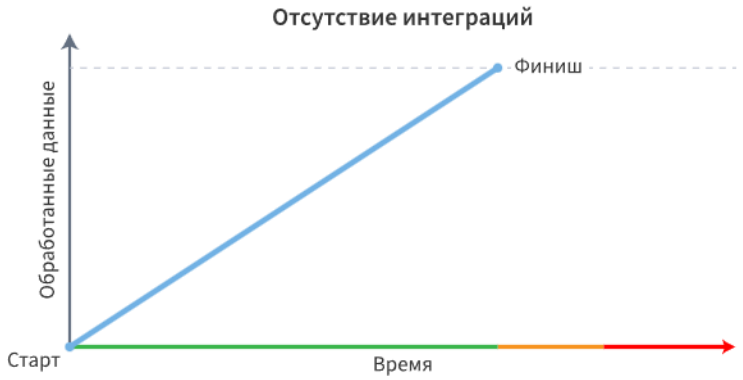
Loginom – одна из самых быстрых в мире low-code платформ продвинутой аналитики

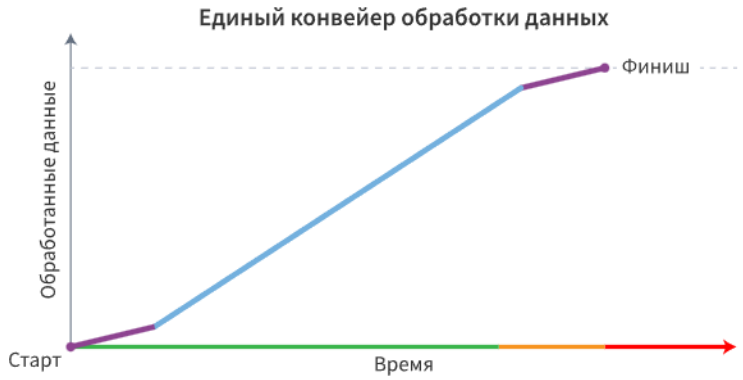
Факторы, влияющие на производительность:

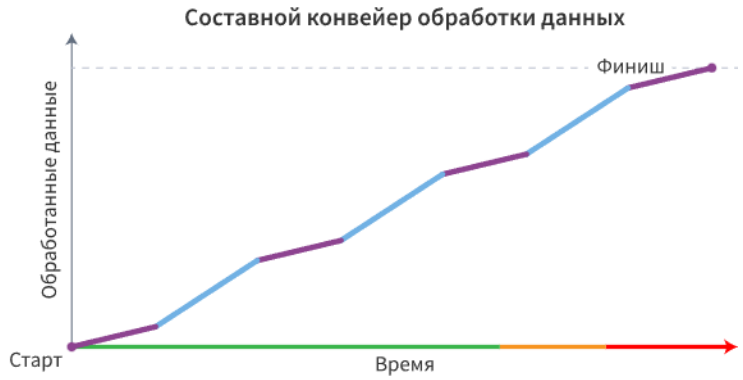
1. Архитектура платформы
2. Алгоритмы обработки
3. Способы визуализации
4. Работа с процессорами, памятью, дисками
5. Работа с источниками/приемниками данных
6. Интеграция модулей Loginom между собой

Принципы оптимизации:

1. Оценивать производительность системы в целом, а не фрагмента
2. Улучшать то, что вызывается часто
3. Учитывать возможности железа
4. Искать первоисточник потерь







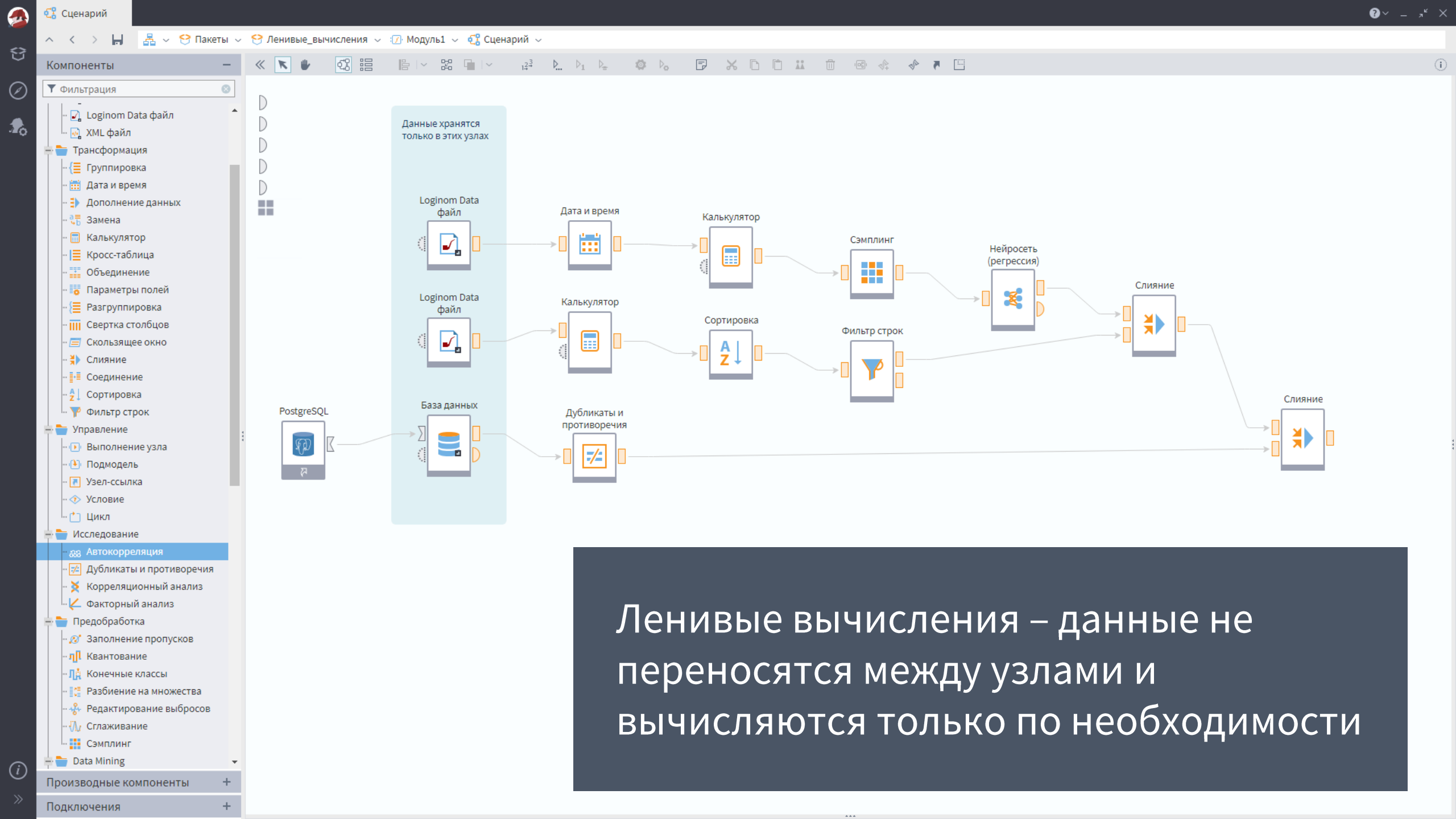
Чем больше в решении разнородных компонентов, тем больше точек интеграции и выше потери производительности

Loginom

Архитектуры платформы

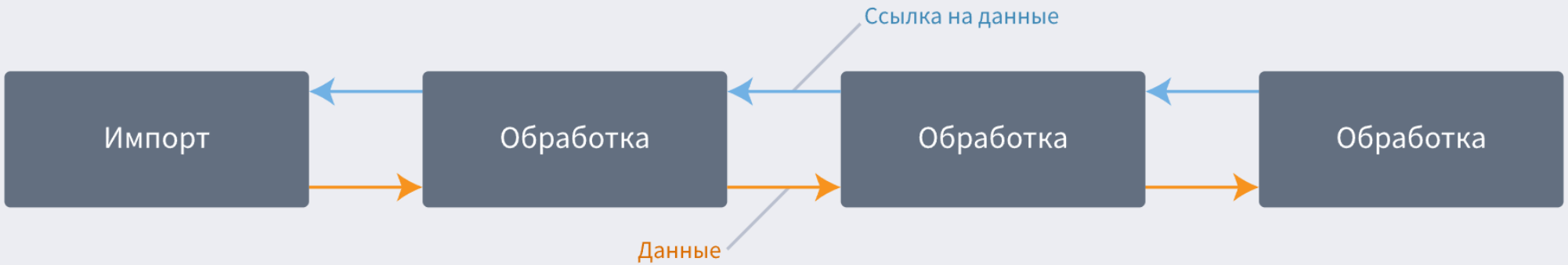
Экономить ОЗУ, т.к. при работе с большими данными она заканчивается быстро:

1. Работать с типизированными данными
2. Хранить в памяти уникальные значения
3. Использовать ленивые вычисления
4. …

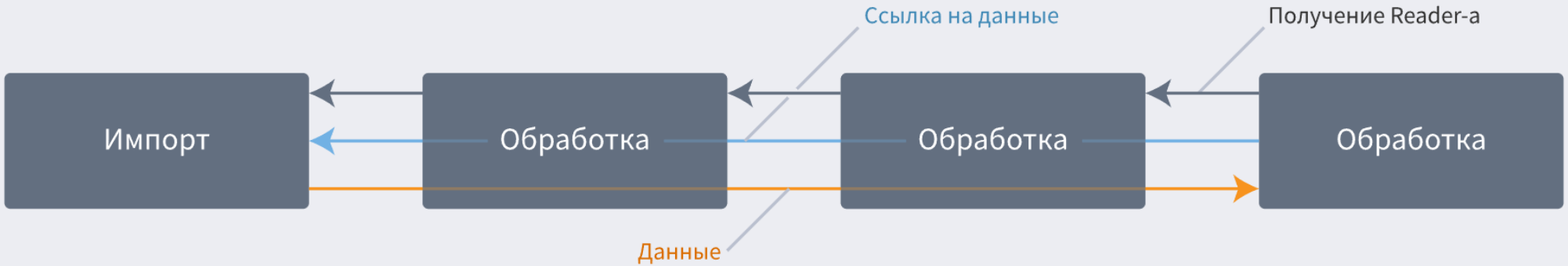

Сценарий
Пакеты
Ленивые_вычисления
Модуль1
Сценарий
Компоненты
Фильтрация
Loginom Data файл
XML файл
Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Свертка столбцов
Скользящее окно
Слияние
Соединение
Сортировка
Фильтр строк
Управление
Выполнение узла
Подмодель
Узел-ссылка
Условие
Цикл
Исследование
Автокорреляция
Дубликаты и противоречия
Корреляционный анализ
Факторный анализ
Предобработка
Заполнение пропусков
Квантование
Конечные классы
Разбиение на множества
Редактирование выбросов
Сглаживание
Сэмплинг
Data Mining
Производные компоненты
Подключения
Данные хранятся только в этих узлах
Loginom Data файл
Loginom Data файл
PostgreSQL
База данных
Дата и время
Калькулятор
Калькулятор
Сортировка
Сэмплинг
Фильтр строк
Дубликаты и противоречия
Нейросеть (регрессия)
Слияние
Слияние
Ленивые вычисления – данные не переносятся между узлами и вычисляются только по необходимости

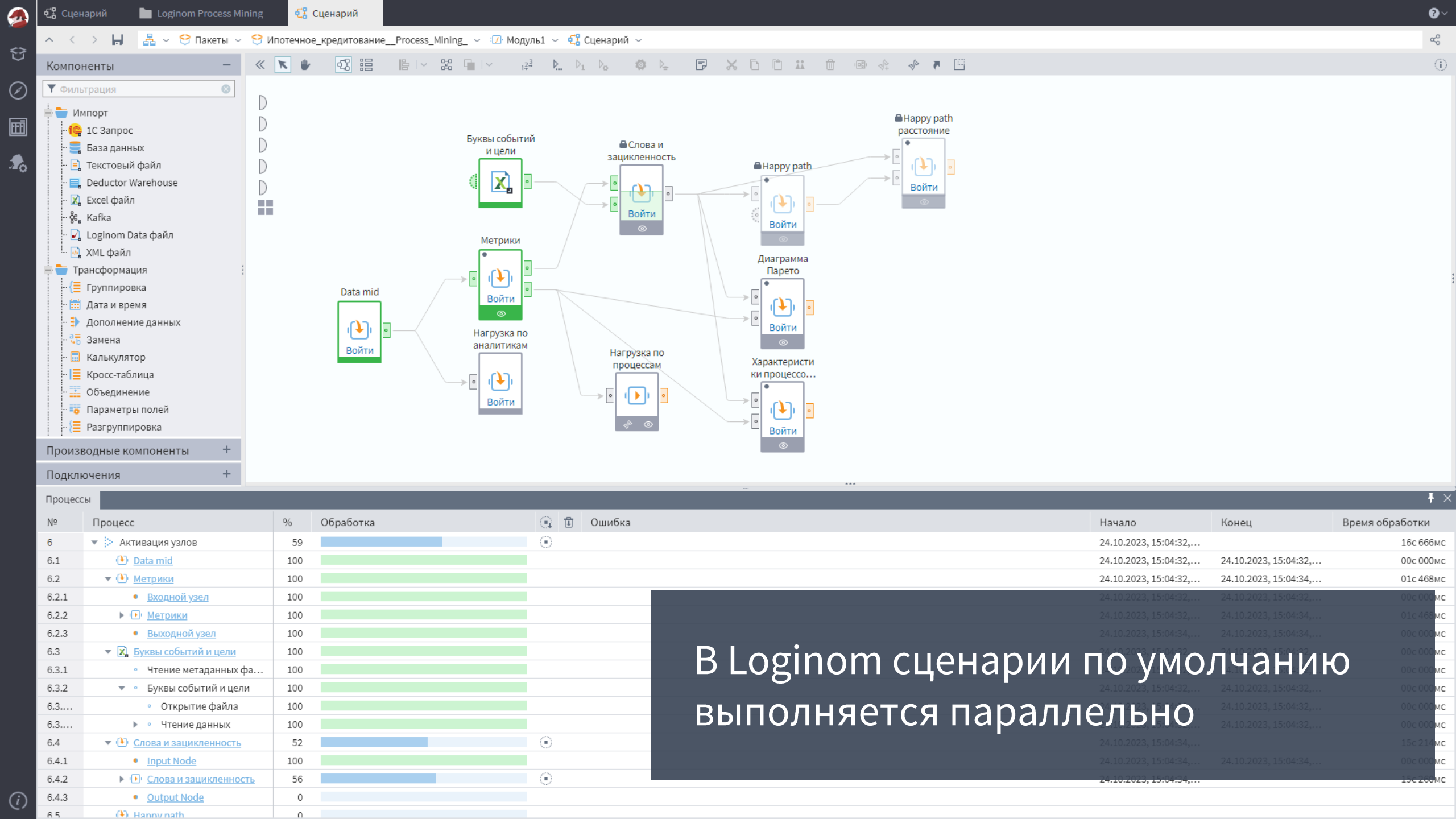
Оптимизация ленивых вычислений

Было

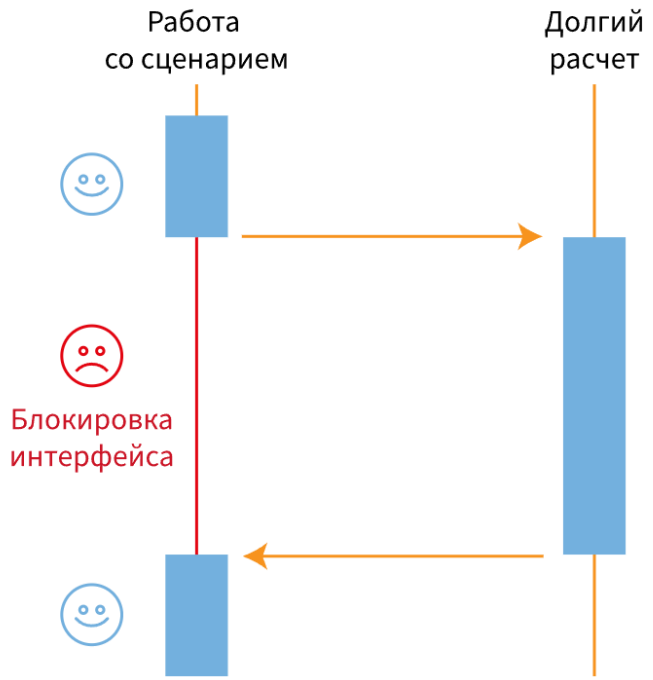


Стало

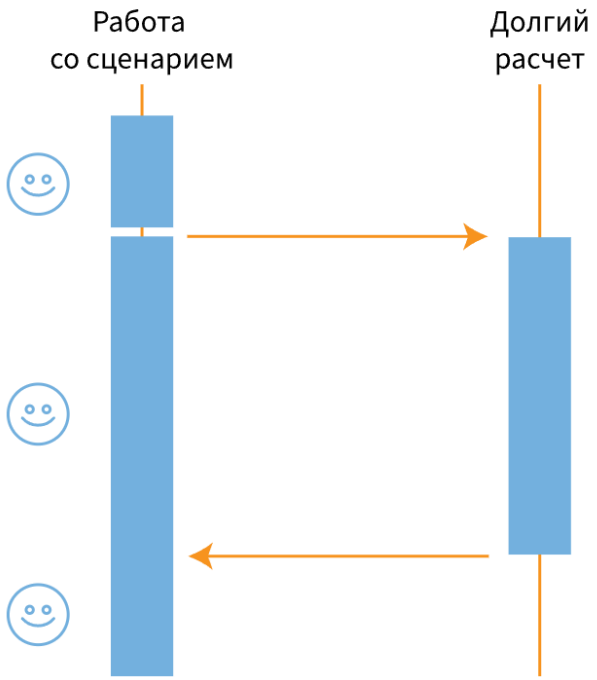



Сценарий
Loginom Process Mining
Сценарий
Пакеты
Ипотечное_кредитование__Process_Mining_
Модуль1
Сценарий
Компоненты
Фильтрация
Импорт
1С Запрос
База данных
Текстовый файл
Deductor Warehouse
Excel файл
Kafka
Loginom Data файл
XML файл
Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Производные компоненты
Подключения
Буквы событий и цели
Слова и зацикленность
Happy path
Happy path расстояние
Метрики
Диаграмма Парето
Data mid
Нагрузка по аналитикам
Нагрузка по процессам
Характеристики процессо...
Войти
Процессы
№
Процесс
%
Обработка
Ошибка
Начало
Конец
Время обработки
6 Активация узлов 59 24.10.2023, 15:04:32,... 16с 666мс
6.1 Data mid 100 24.10.2023, 15:04:32,... 24.10.2023, 15:04:32,... 00с 000мс
6.2 Метрики 100 24.10.2023, 15:04:32,... 24.10.2023, 15:04:34,... 01с 468мс
6.2.1 Входной узел 100
6.2.2 Метрики 100
6.2.3 Выходной узел 100
6.3 Буквы событий и цели 100
6.3.1 Чтение метаданных фа... 100
6.3.2 Буквы событий и цели 100
6.3.... Открытие файла 100
6.3.... Чтение данных 100
6.4 Слова и зацикленность 52
6.4.1 Input Node 100
6.4.2 Слова и зацикленность 56
6.4.3 Output Node 0
6.5 Happy path 0
В Loginom сценарии по умолчанию выполняется параллельно

Синхронный пользовательский интерфейс



Асинхронный пользовательский интерфейс



Асинхронный пользовательский интерфейс не снижает время долгих расчетов, но не блокирует работу аналитика. За счет этого повышается отзывчивость системы и комфорт от работы.

Loginom

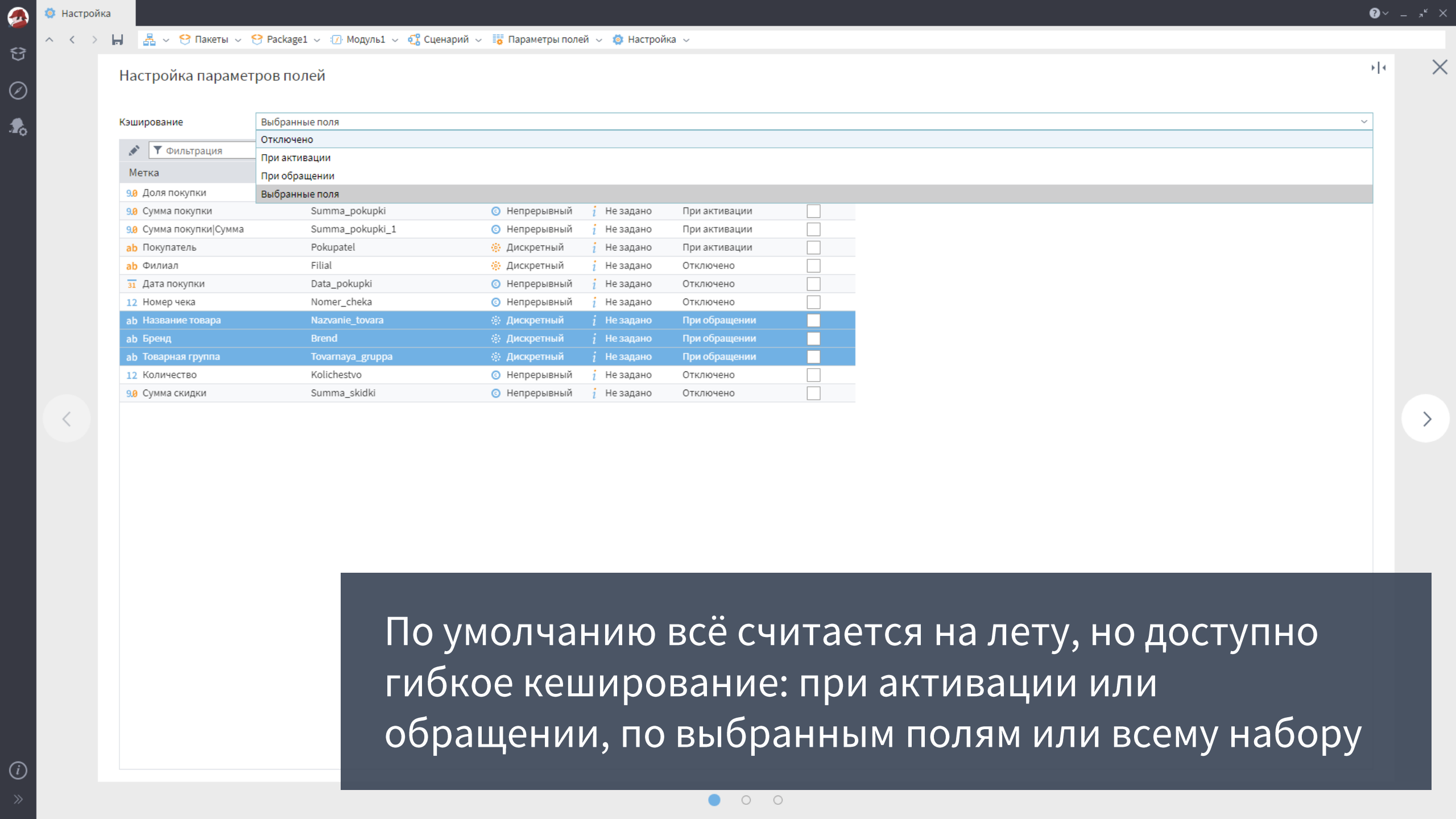
Алгоритмы обработки

Сторонние библиотеки

1. LIBLINEAR – одна из самых быстрых библиотек для построения моделей логистической регрессии
2. Intel MKL – одна из самых производительных библиотек для операций с матрицами и задач линейной алгебры
3. fast_float – библиотека для быстрого преобразования строки в вещественное число
4. ALGLIB – оптимизированный расчет статистик, обучение нейросетей
5. …

Собственная реализация алгоритмов

1. Использование параллелизма при построении индексов (слиянии и дополнении данных)
2. Приблизительные вычисления при построении моделей машинного обучения
3. Подбор гиперпараметров нейросетей
4. Алгоритмы децимации данных для диаграмм
5. Многомерные расчеты в кубе
6. …


Настройка
Пакеты
Package1
Модуль1
Сценарий
Параметры полей
Настройка
Настройка параметров полей
Кэширование
Выбранные поля
Отключено
При активации
При обращении
Выбранные поля
Фильтрация
Метка
Доля покупки
Сумма покупки Summa_pokupki Непрерывный Не задано При активации
Сумма покупки|Сумма Summa_pokupki_1 Непрерывный Не задано При активации
Покупатель Pokupatel Дискретный Не задано При активации
Филиал Filial Дискретный Не задано Отключено
Дата покупки Data_pokupki Непрерывный Не задано Отключено
Номер чека Nomer_cheka Непрерывный Не задано Отключено
Название товара Nazvanie_tovara Дискретный Не задано При обращении
Бренд Brend Дискретный Не задано При обращении
Товарная группа Tovarnaya_gruppa Дискретный Не задано При обращении
Количество Kolichestvo Непрерывный Не задано Отключено
Сумма скидки Summa_skidki Непрерывный Не задано Отключено
По умолчанию всё считается на лету, но доступно гибкое кеширование: при активации или обращении, по выбранным полям или всему набору

Визуализация

Собственный RPC протокол обмена данными между клиентом и сервером:

1. Исключение всех промежуточных слоев
2. Обмен данными в бинарном виде для снижения объема передаваемой информации и минимизации потерь на сериализацию
3. Пакетная передача с объединением мелких запросов в крупные блоки
4. Асинхронный обмен

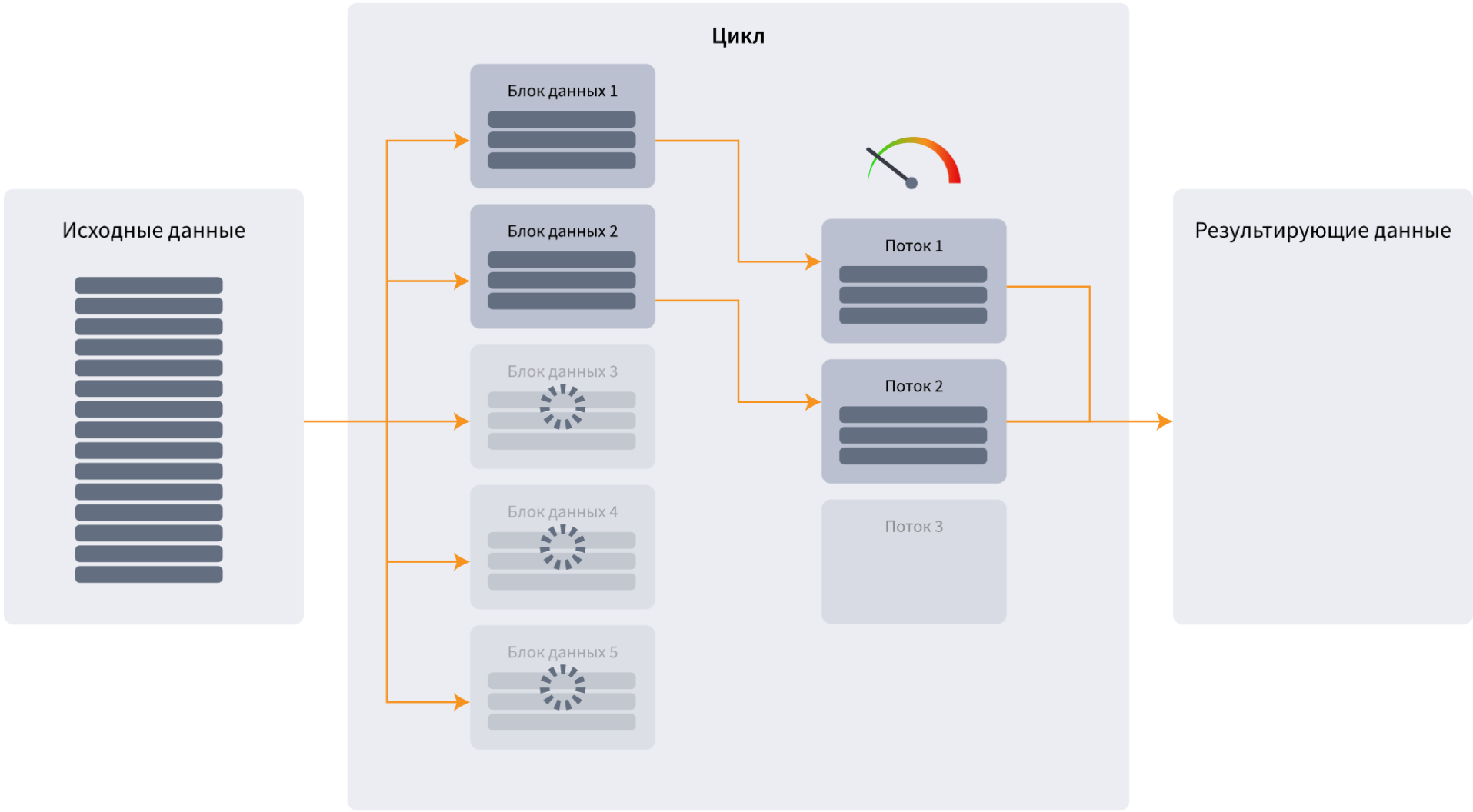
Прослойка между клиентом и сервером, учитывающая особенности визуализации проводит децимацию данных на сервере перед передачей клиенту.

Низкоуровневые операции

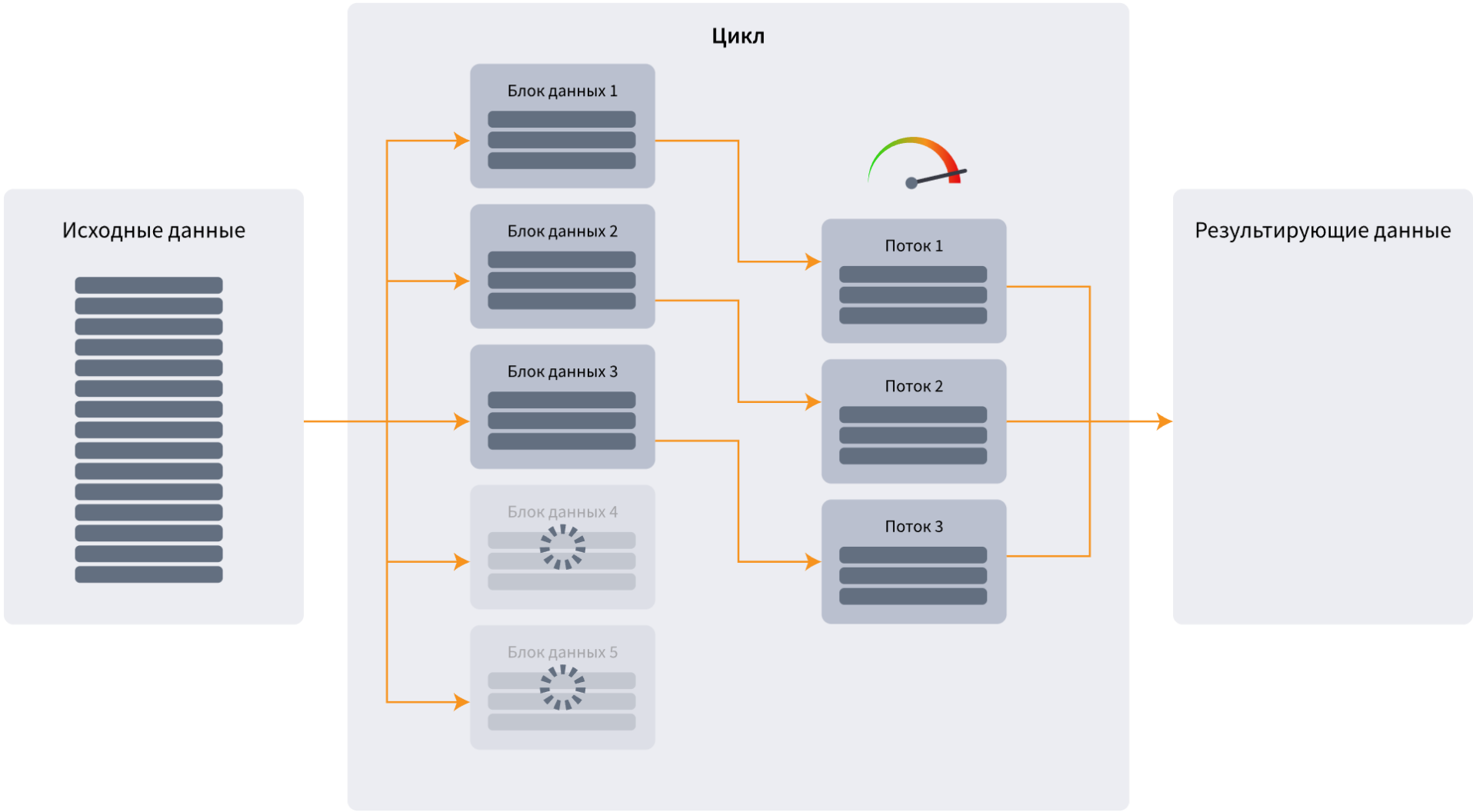
Собственная реализация многопоточной обработки:

1. Формирование своего пула потоков
2. Мониторинг загруженности процессоров для добавления потока в случае не дозагруженности
3. Выполнение задач последовательно, если излишний параллелизм не выгоден
4. Освобождение потоков, если они не используются

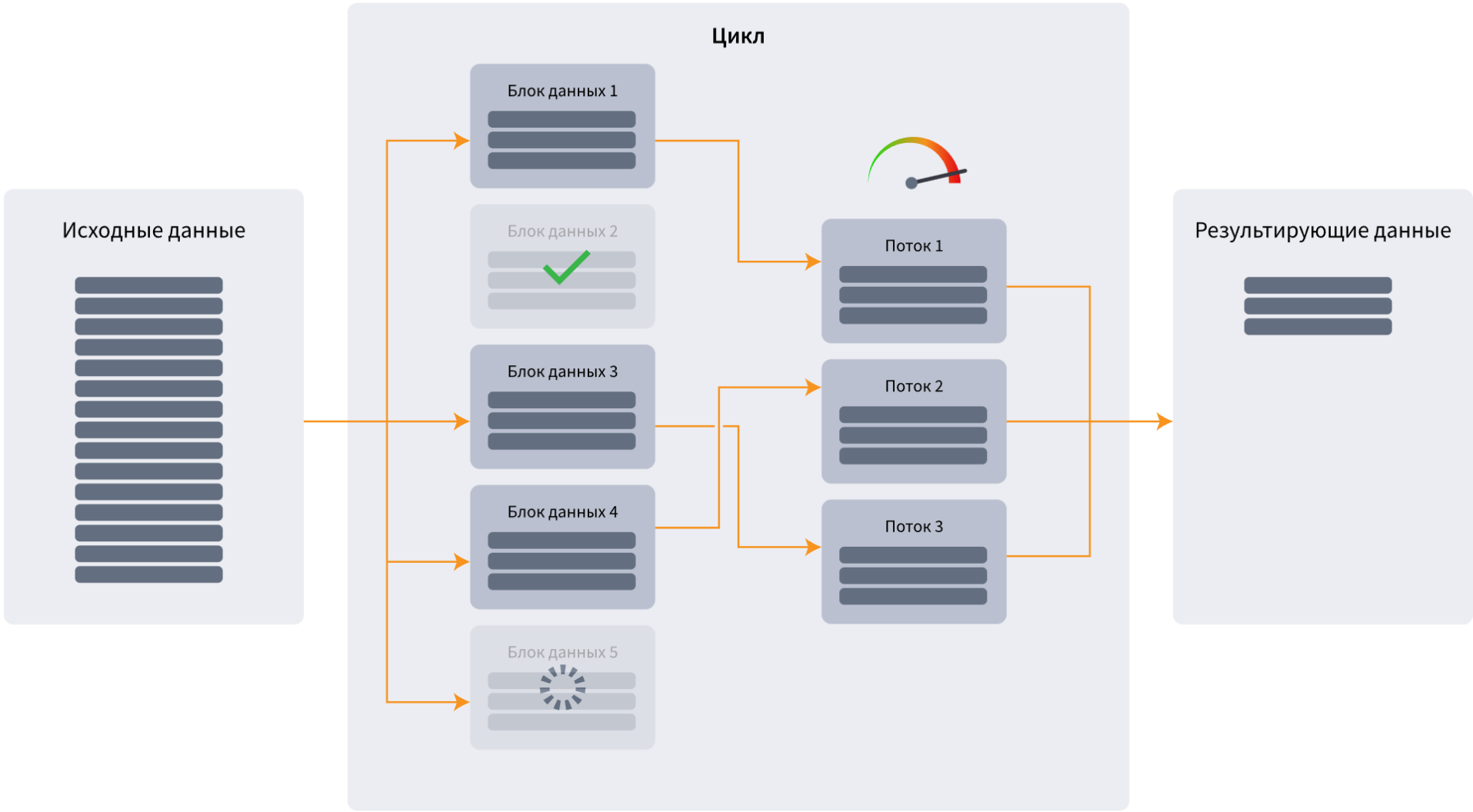
Параллельный цикл в многопроцессорных системах (1)



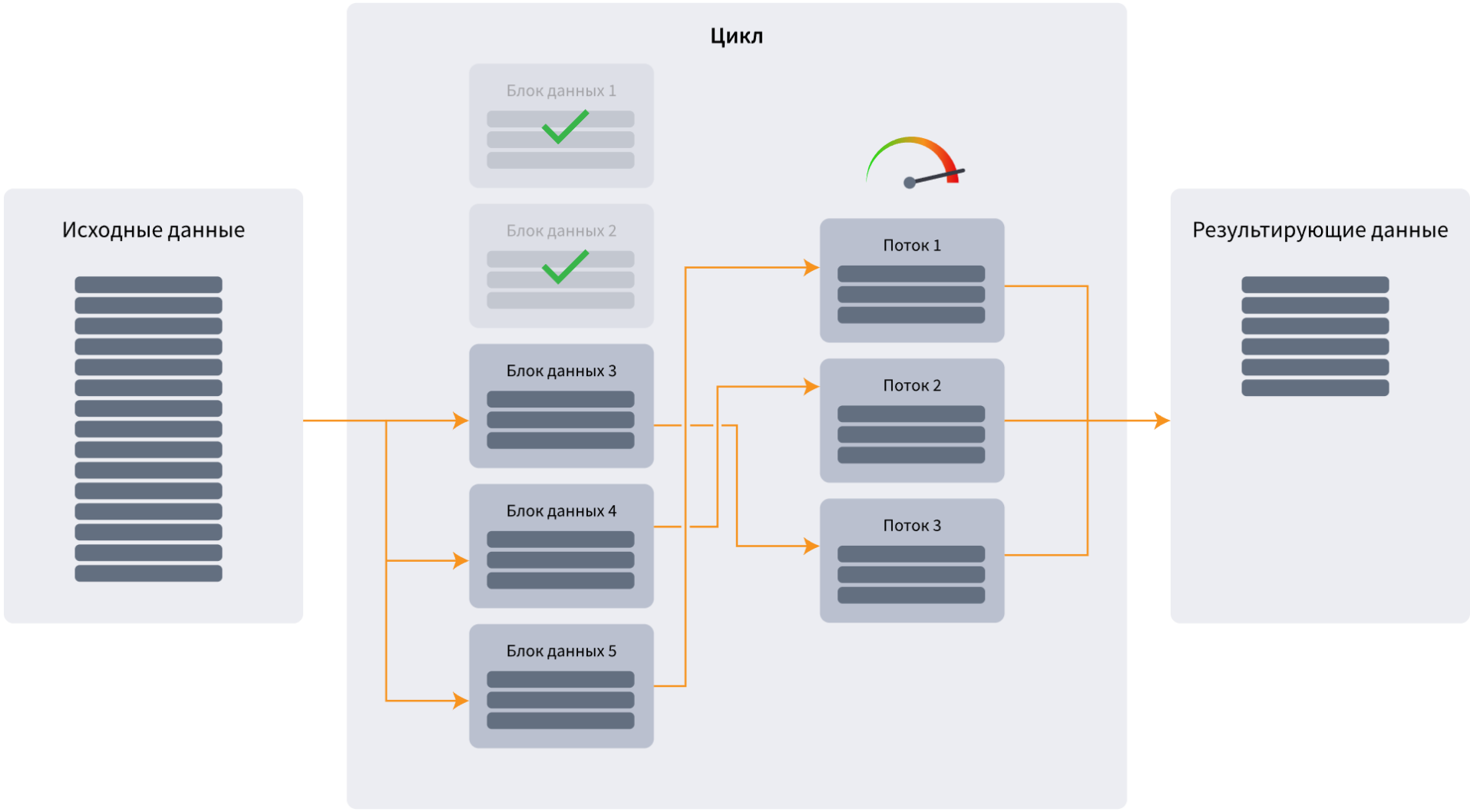
Параллельный цикл в многопроцессорных системах (2)



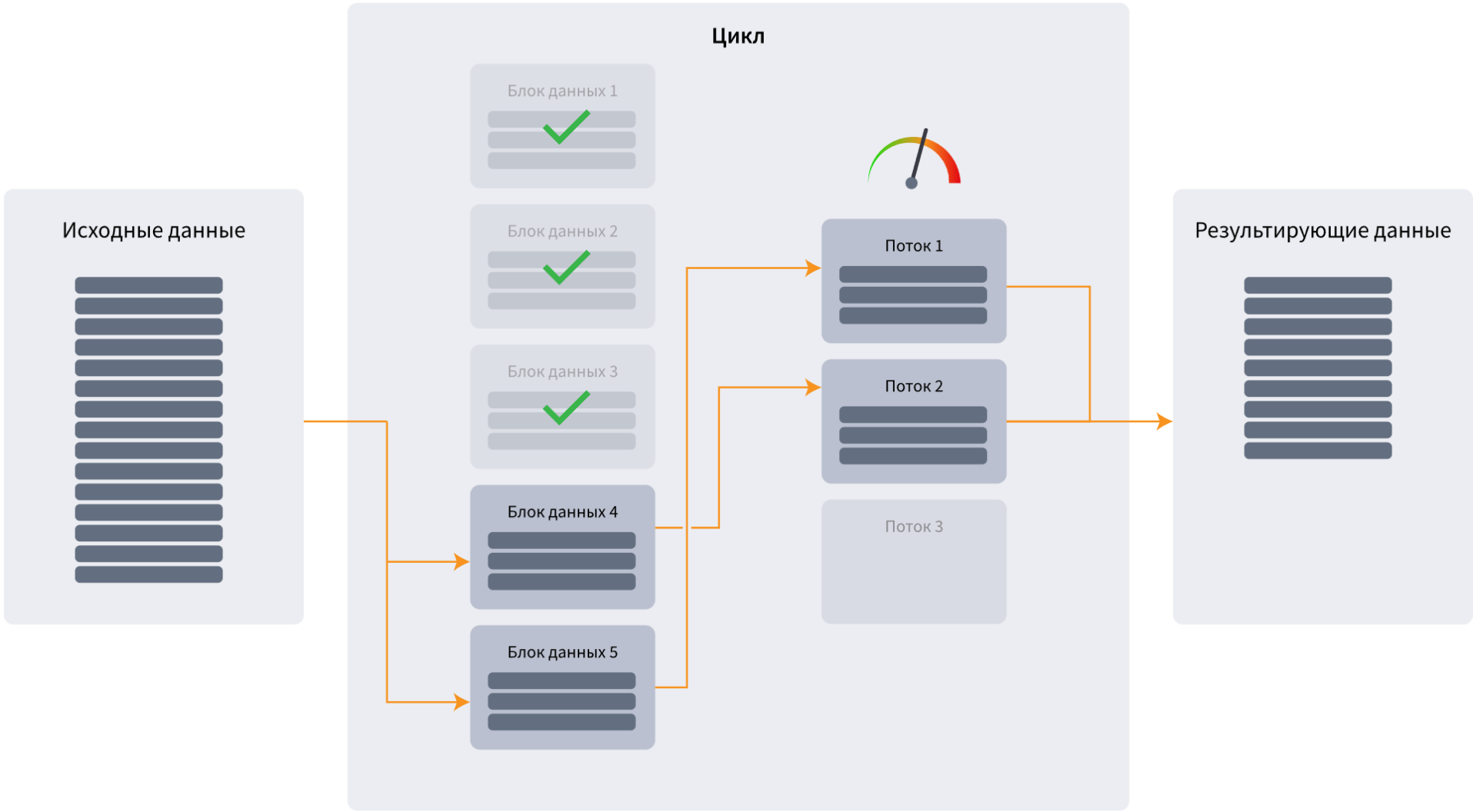
Параллельный цикл в многопроцессорных системах (3)



Параллельный цикл в многопроцессорных системах (4)



Параллельный цикл в многопроцессорных системах (5)



Параллельный цикл в многопроцессорных системах (6)

Например, для быстрой фильтрации сравнение происходит при помощи специальной функции, исключающей атомарные операции обращения к общей памяти, блокирующие конвейер всех ядер процессора. Иначе в результате блокировок даже если ядер много фактически система работает в один поток.

Результат – на многоядерных серверах увеличение скорости фильтрации примерно на 20%.

Источники/приемники данных

У всех СУБД запросы отправленные через один коннект выполняются последовательно. Для увеличения скорости требуется оптимизация:

1. При подключении к БД создается пул коннектов, за счет чего запросы выполняются параллельно
2. Для минимизации накладных расходов на коннект подключения удерживаются открытыми
3. Для экономии памяти на сервере можно настроить принудительное закрытие подключений

Оптимизация импорта из медленных источников (csv, xlsx…):

1. При чтении строковые данные кэшируются, кэш строк организован в виде хэш-таблицы.
2. Используется wyhash (самая быстрая и качественная хэш-функция, 2019 г.) — алгоритм, оптимизированный под современные процессоры.
3. Размер хеш-таблицы подобран так, чтобы он помещался в кэш-память и обеспечивал максимальную производительность.

Импорт замедлился, но уменьшился расход памяти и ускорилось выполнение последующих узлов в сценарии.

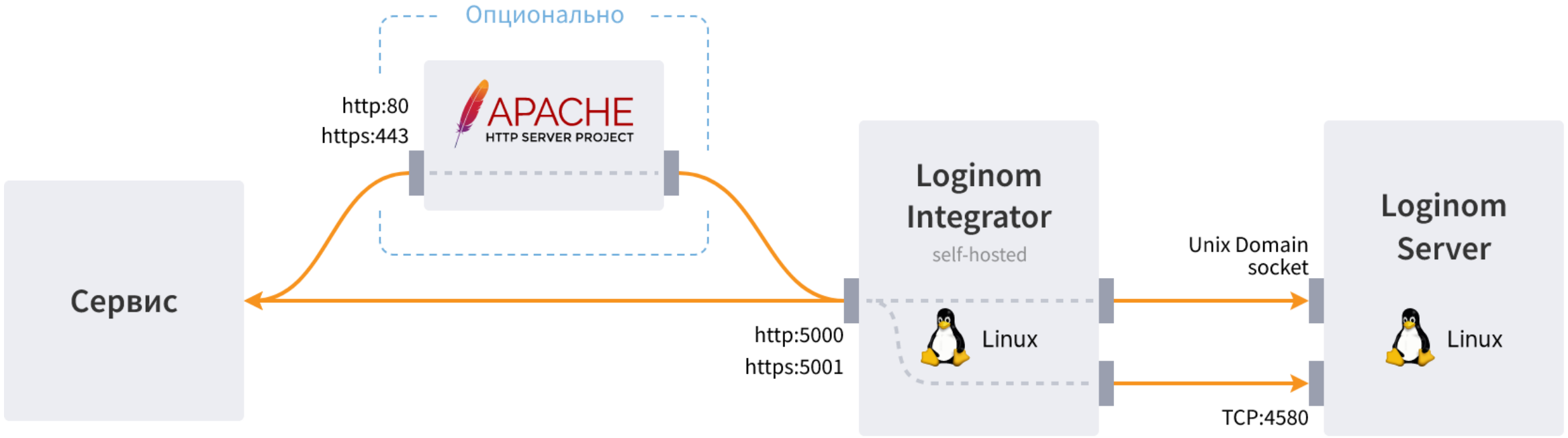
Оптимизация формата Loginom Data File:

- Замена алгоритма сжатия LZ0 на LZ4 с добавлением расчета контрольной суммы.
- Размер файла увеличился примерно на 10%, но запись быстрее на 13%, а чтение на 39%.
- В процессе экспорта в файле сохраняются уникальные строки, из-за чего скорость записи уменьшилась. При этом при импорте скорость чтения увеличилась, а расход ОЗУ снизился.

Loginom

Интеграция компонентов платформы

Обмен данными между Server-ом и Integrator-ом



Обычно Integrator и Server общаются по TCP, но при работе на одной машине можно использовать Unix domain socket. Отклик быстрее на ~15%.

Для параллельного выполнения узла Python анализируемые выборки выгружаются в файлы и запускается несколько интерпретаторов.

Наборы записываются в файлы бинарного формата с хранением по столбцам и прозрачно для аналитика загружаются в Python. Это повышает скорость обмена данными между Python и другими узлами сценария.

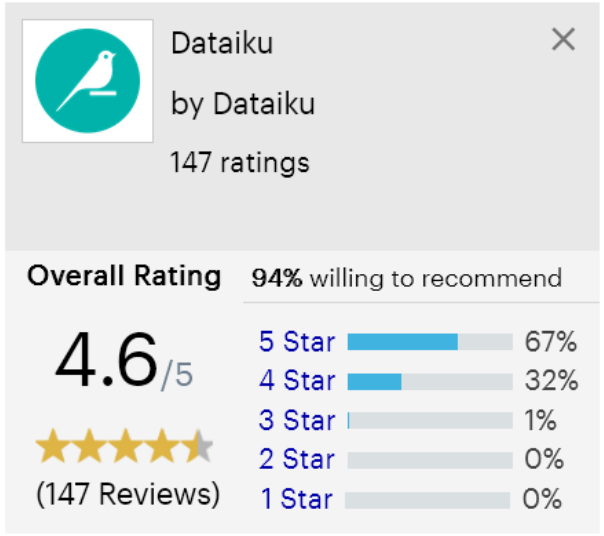
Loginom

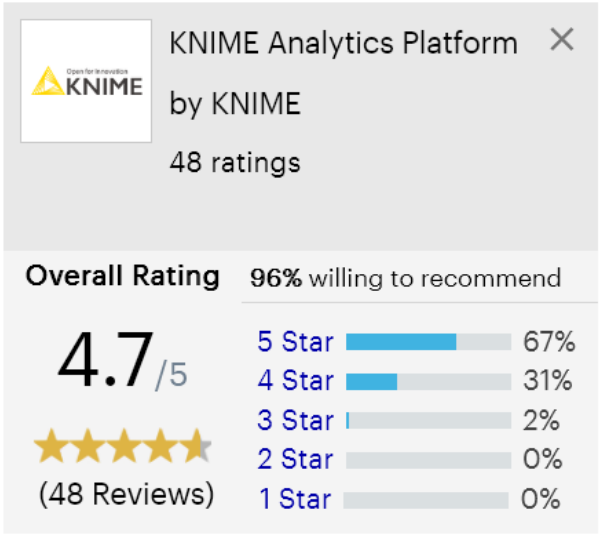
Сравнение с конкурентами

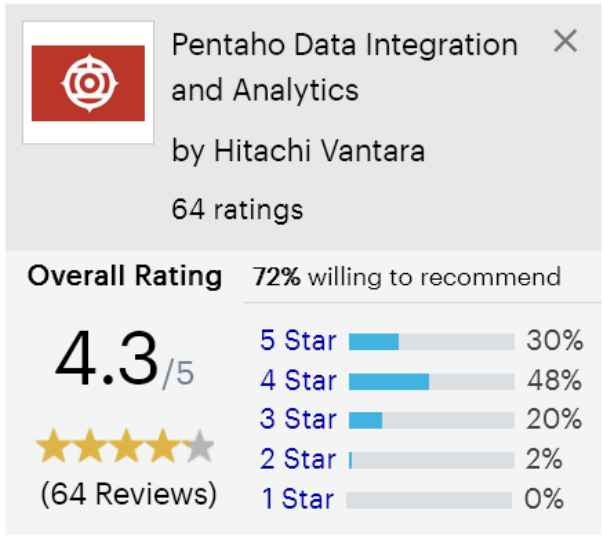
Сравнение настольных редакций популярных платформ low-code-аналитики

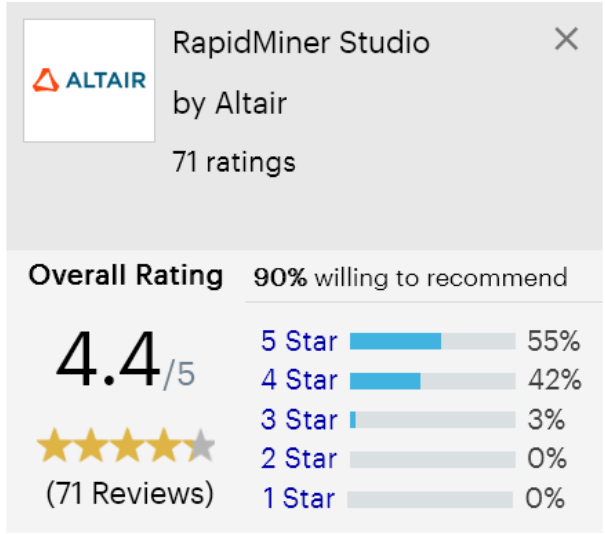
1. Alteryx
2. Dataiku
3. KNIME
4. Loginom
5. Pentaho
6. RapidMiner

С Loginom сравниваются лучше мировые продукты по мнению Gartner







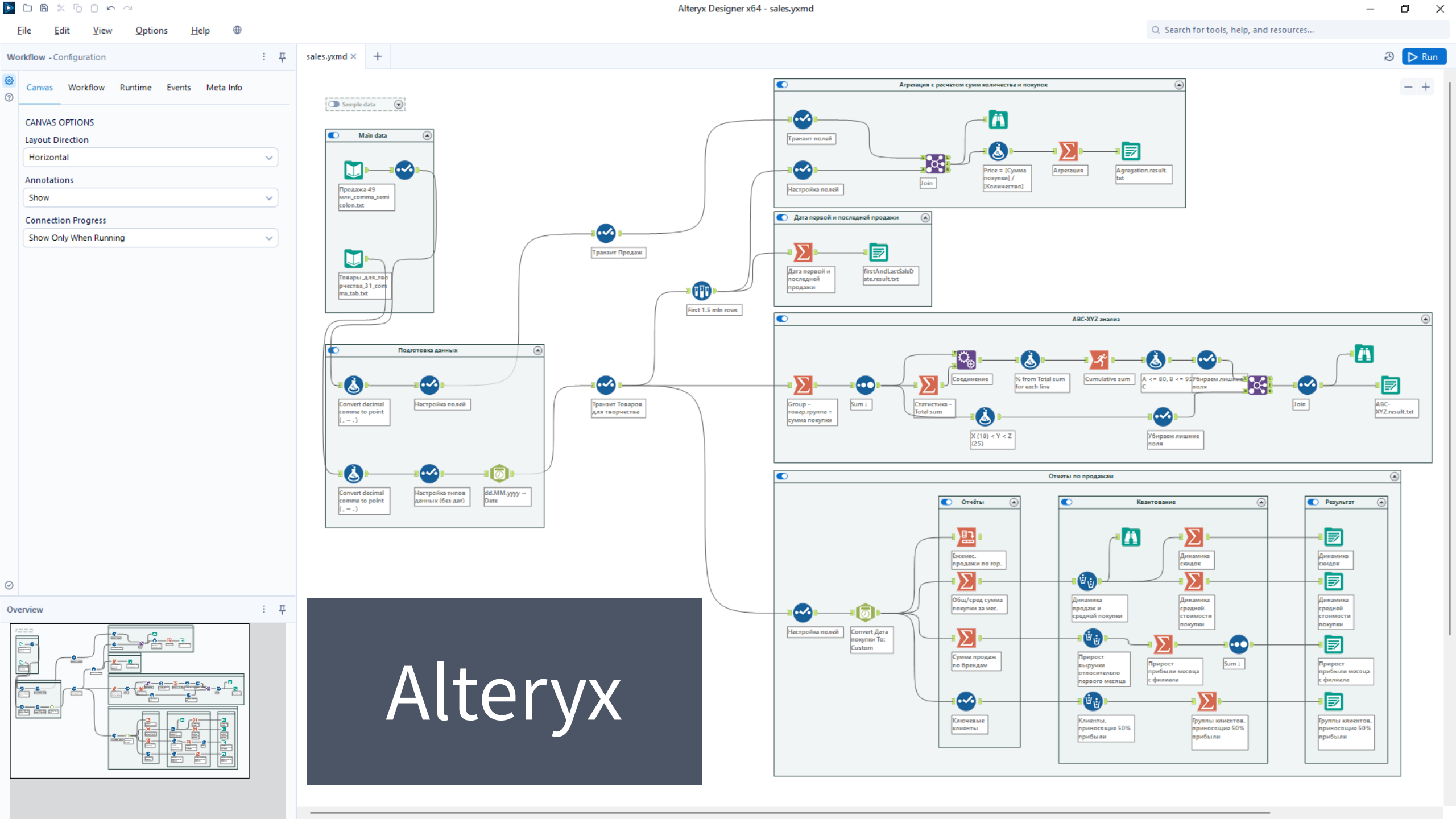


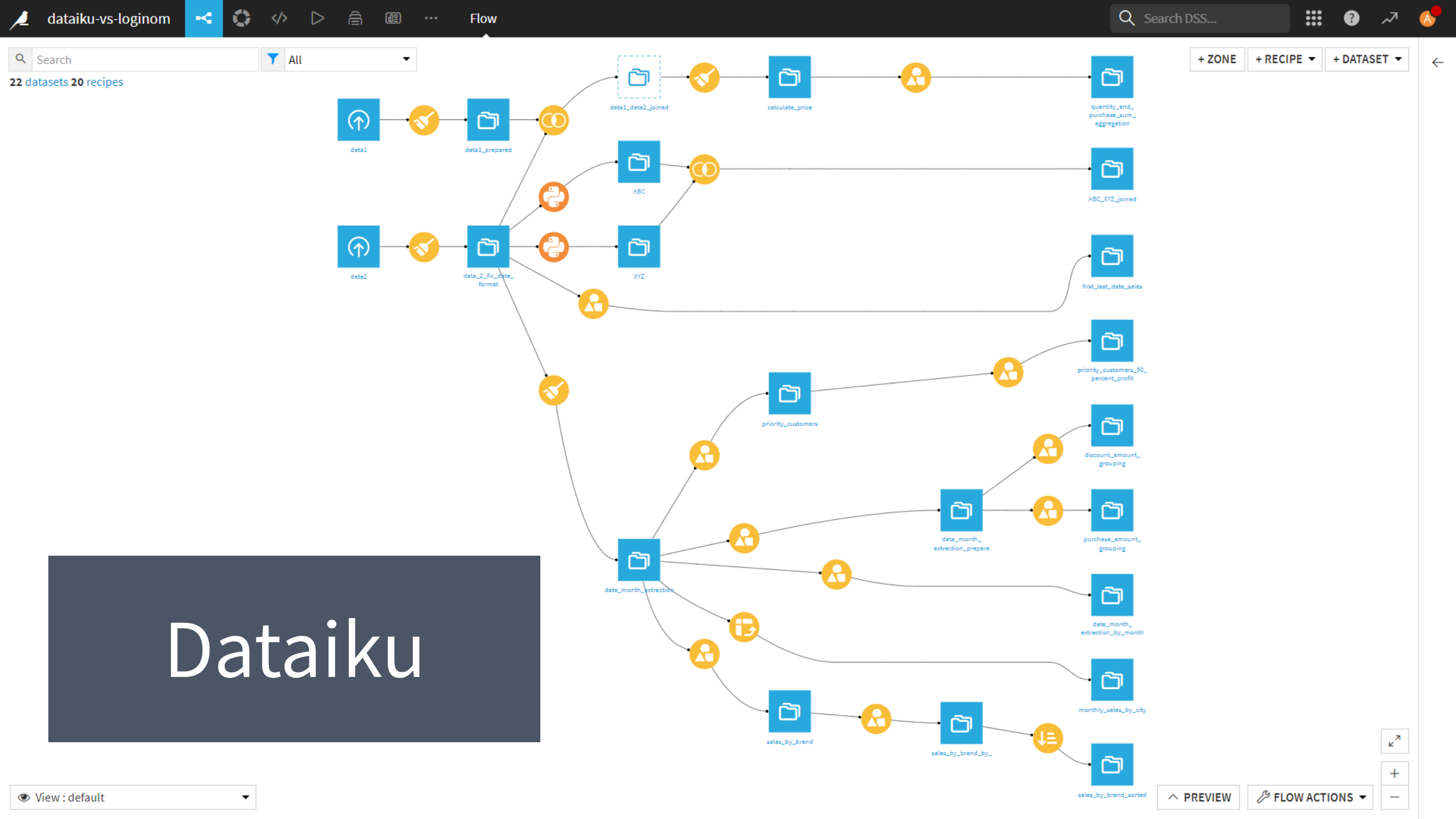
Задача 1. Простой сценарий обработки.

Построение отчетности по продажам, ABC и XYZ-анализ с минимальными настройками. Тест производительности базовых операций работы с данными:

1. Импорт csv-файлов
2. Слияние наборов данных
3. Расчет по формулам
4. Группировки, сортировки
5. Расчет простых статистик
6. Квантование
7. Экспорт csv-файлов

Параметры тестового компьютера: AMD Ryzen 5 5600G 3.9 GHz (6 ядер, 12 потоков), DDR 32 Gb 3200 MHz, SSD ADATA SP90


Alteryx Designer x64 - sales.yxmd
File
Edit
View
Options
Help
Search for tools, help, and resources...
Workflow - Configuration
Canvas
Workflow
Runtime
Events
Meta Info
CANVAS OPTIONS
Layout Direction
Horizontal
Annotations
Show
Connection Progress
Show Only When Running
Overview
sales.yxmd
Run
Sample data
Main data
Продажа 49 млн_comma_semicolon.txt
Товары_для_творчества_31_comma_tab.txt
Подготовка данных
Convert decimal comma to point (, → .)
Настройка полей
Convert decimal comma to point (, → .)
Настройка типов данных (без дат)
dd.MM.yyyy → Date
Транзит Продаж
First 1.5 mln rows
Транзит Товаров для творчества
Агрегация с расчетом сумм количества и покупок
Транзит полей
Настройка полей
Join
Price = [Сумма покупки] / [Количество]
Агрегация
Agregation.result.txt
Дата первой и последней продажи
Дата первой и последней продажи
firstAndLastSaleDate.result.txt
ABC-XYZ анализ
Group – товар.группа + сумма покупки
Sum ↓
Соединение
Статистика – Total sum
% from Total sum for each line
Cumulative sum
A <= 80, B <= 95, C
Убираем лишние поля
X (10) < Y < Z (25)
Убираем лишние поля
Join
ABC-XYZ.result.txt
Отчеты по продажам
Настройка полей
Convert Дата покупки To: Custom
Отчёты
Ежемес. продажи по гор.
Общ/сред сумма покупки за мес.
Сумма продаж по брендам
Ключевые клиенты
Квантование
Динамика продаж и средней покупки
Динамика скидок
Динамика средней стоимости покупки
Прирост выручки относительно первого месяца
Прирост прибыли месяца с филиала
Sum ↓
Клиенты, приносящие 50% прибыли
Группы клиентов, приносящие 50% прибыли
Результат
Динамика скидок
Динамика средней стоимости покупки
Прирост прибыли месяца с филиала
Группы клиентов, приносящие 50% прибыли
Alteryx


dataiku-vs-loginom
Flow
Search DSS...
Search
All
22 datasets 20 recipes
+ ZONE
+ RECIPE
+ DATASET
data1
data1_prepared
data1_data2_joined
calculate_price
quantity_and_purchase_sum_aggregation
ABC
ABC_XYZ_joined
data2
data_2_fix_date_format
XYZ
first_last_date_sales
priority_customers
priority_customers_50_percent_profit
discount_amount_grouping
date_month_extraction_prepare
purchase_amount_grouping
date_month_extraction
date_month_extraction_by_month
monthly_sales_by_city
sales_by_brand
sales_by_brand_by_
sales_by_brand_sorted
Dataiku
View : default
PREVIEW
FLOW ACTIONS

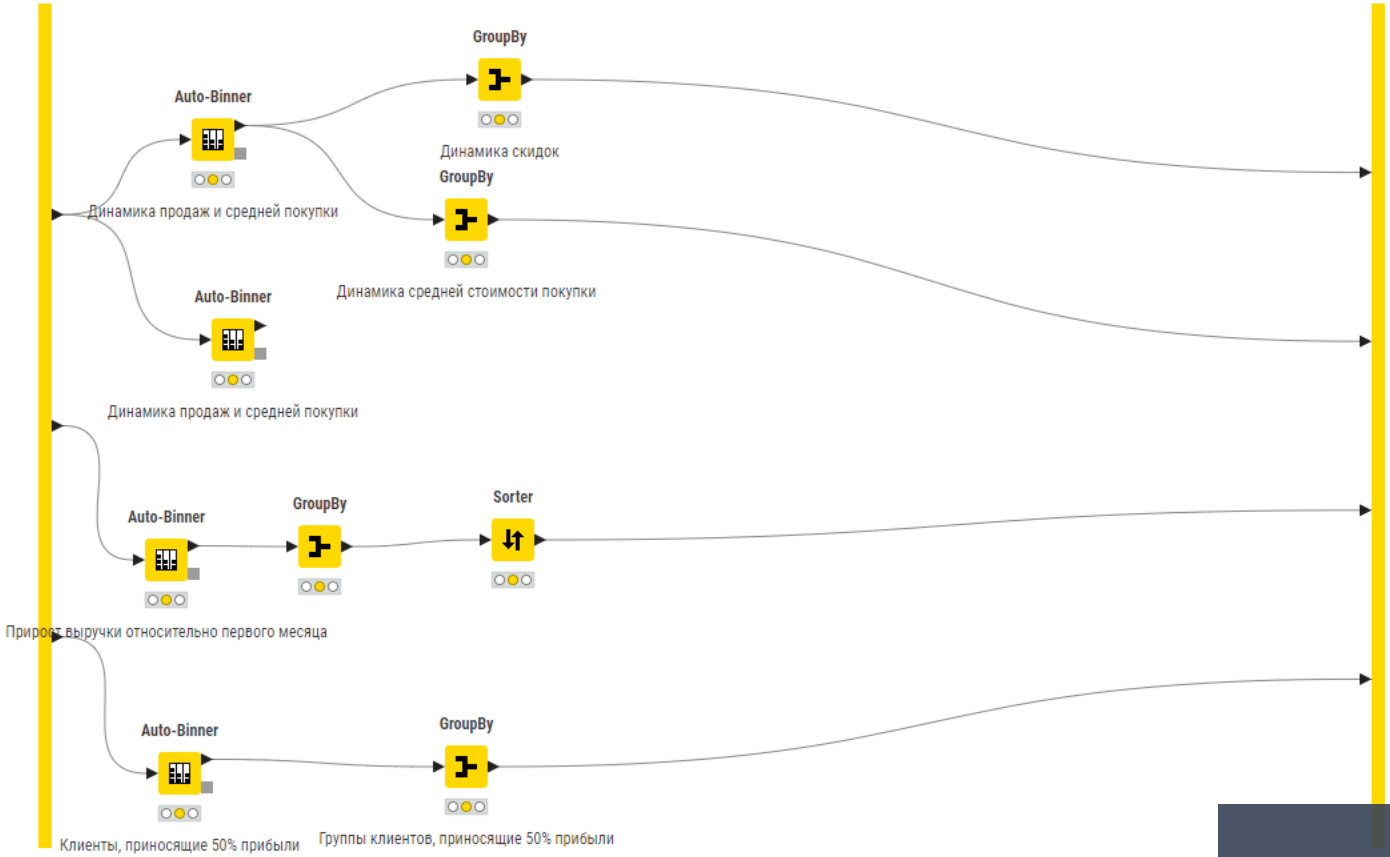
KNIME Analytics Platform

Home | compare •

Execute all | Cancel all | Reset all

compare › Metanode › Отчеты по продажам › Квантование-1

100%



KNIME

To show the node output, please select a configured or executed node.

Spoon - Один поток (changed)

File Edit View Action Tools Help

Pentaho

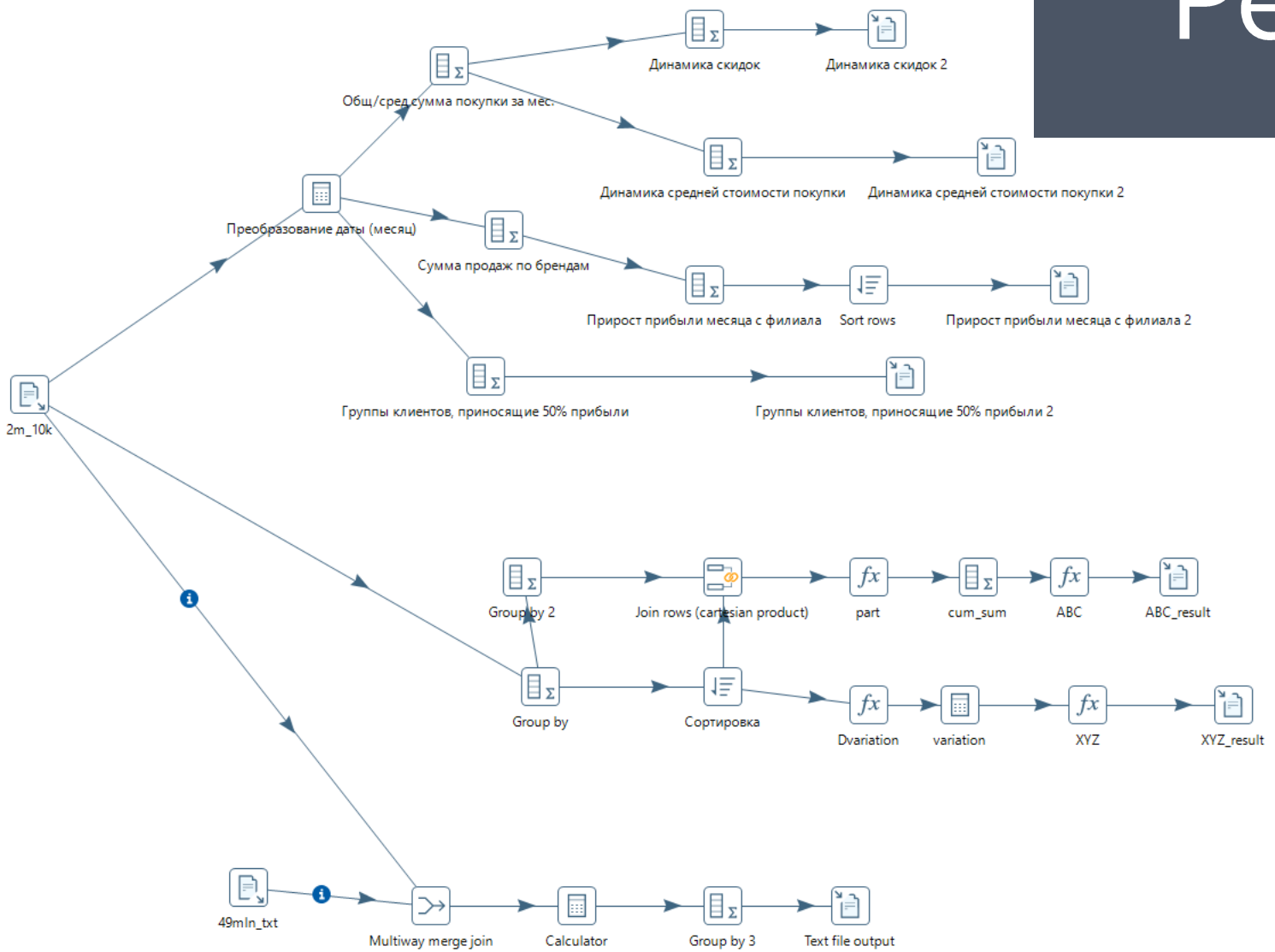
View Design

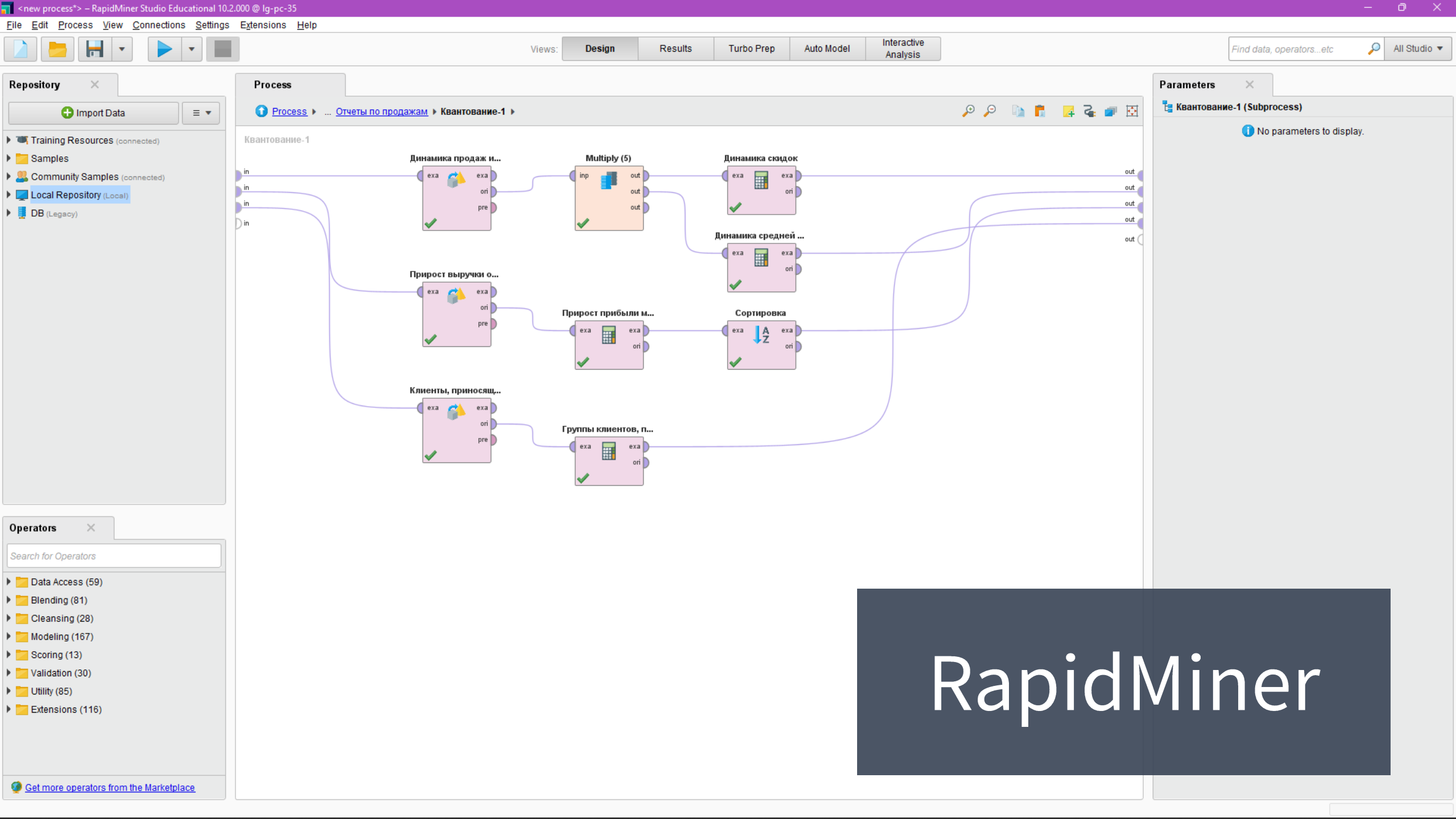
Search

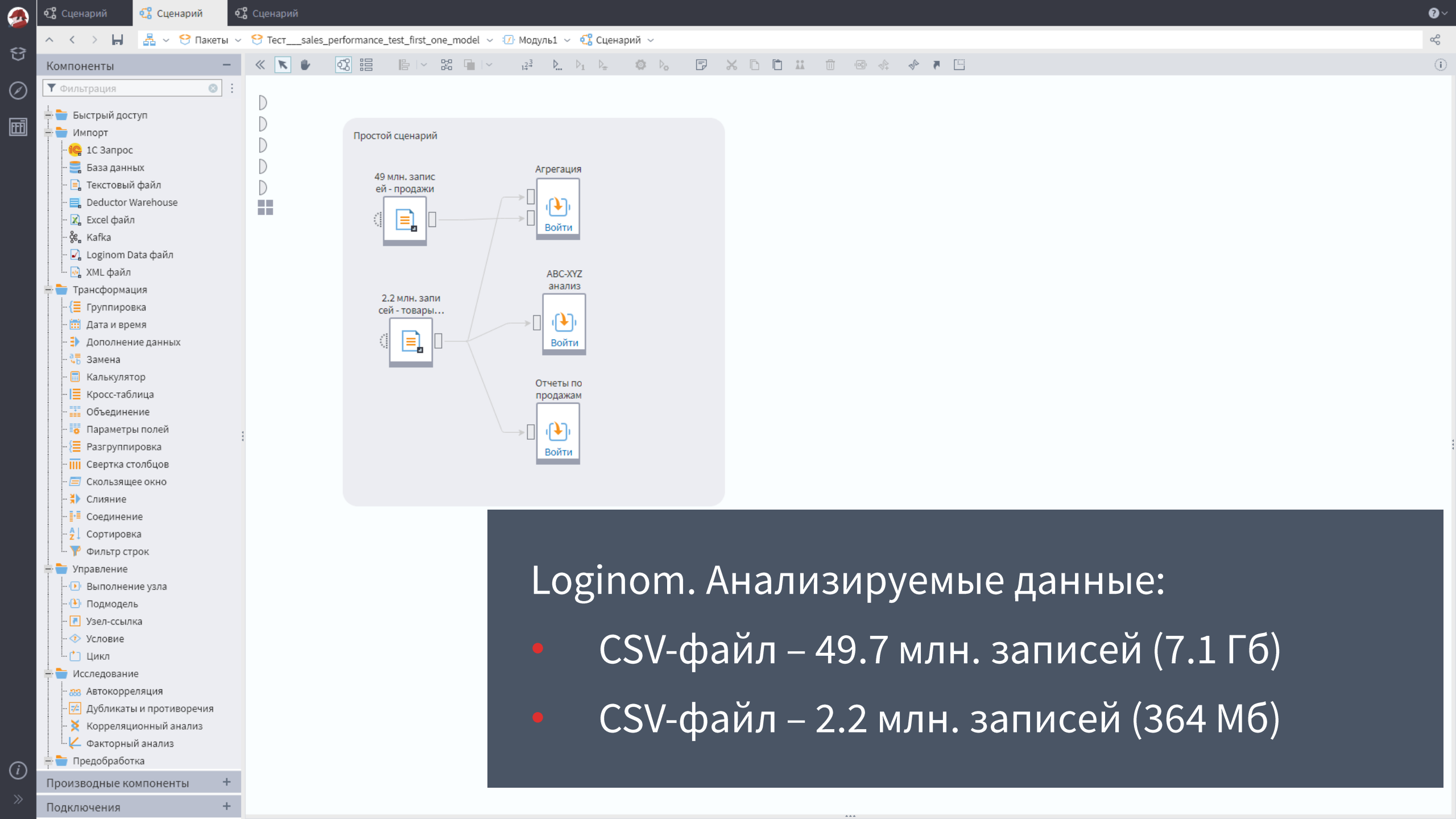
- Transformations
  - Один поток
    - Run configurations
    - Database connections
    - Steps
    - Hops
    - Partition schemas
    - Slave server
    - Kettle cluster schemas
    - VFS Connections

All1 All Один поток

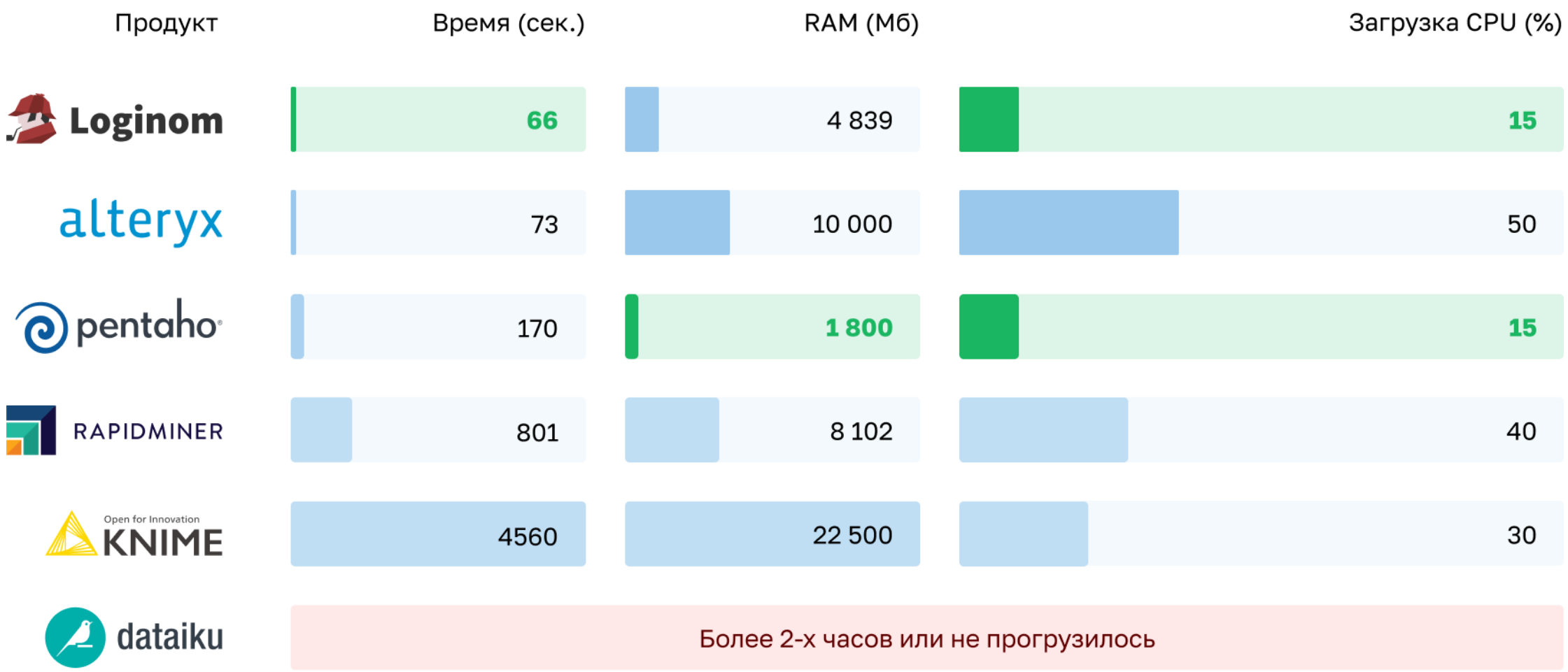
100%




<new process*> – RapidMiner Studio Educational 10.2.000 @ lg-pc-35
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Interactive Analysis
Find data, operators...etc
All Studio
Repository
Import Data
Training Resources (connected)
Samples
Community Samples (connected)
Local Repository (Local)
DB (Legacy)
Operators
Search for Operators
Data Access (59)
Blending (81)
Cleansing (28)
Modeling (167)
Scoring (13)
Validation (30)
Utility (85)
Extensions (116)
Get more operators from the Marketplace
Process
Process ... Отчеты по продажам Квантование-1
Квантование-1
Динамика продаж и...
Multiply (5)
Динамика скидок
Динамика средней ...
Прирост выручки о...
Прирост прибыли м...
Сортировка
Клиенты, приносящ...
Группы клиентов, п...
Parameters
Квантование-1 (Subprocess)
No parameters to display.
RapidMiner


Сценарий
Сценарий
Сценарий
Пакеты
Тест___sales_performance_test_first_one_model
Модуль1
Сценарий
Компоненты
Фильтрация
Быстрый доступ
Импорт
1С Запрос
База данных
Текстовый файл
Deductor Warehouse
Excel файл
Kafka
Loginom Data файл
XML файл
Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Свертка столбцов
Скользящее окно
Слияние
Соединение
Сортировка
Фильтр строк
Управление
Выполнение узла
Подмодель
Узел-ссылка
Условие
Цикл
Исследование
Автокорреляция
Дубликаты и противоречия
Корреляционный анализ
Факторный анализ
Предобработка
Производные компоненты
Подключения
Простой сценарий
49 млн. запис ей - продажи
2.2 млн. запи сей - товары...
Агрегация
ABC-XYZ анализ
Отчеты по продажам
Войти
Войти
Войти
Loginom. Анализируемые данные:
CSV-файл – 49.7 млн. записей (7.1 Гб)
CSV-файл – 2.2 млн. записей (364 Мб)

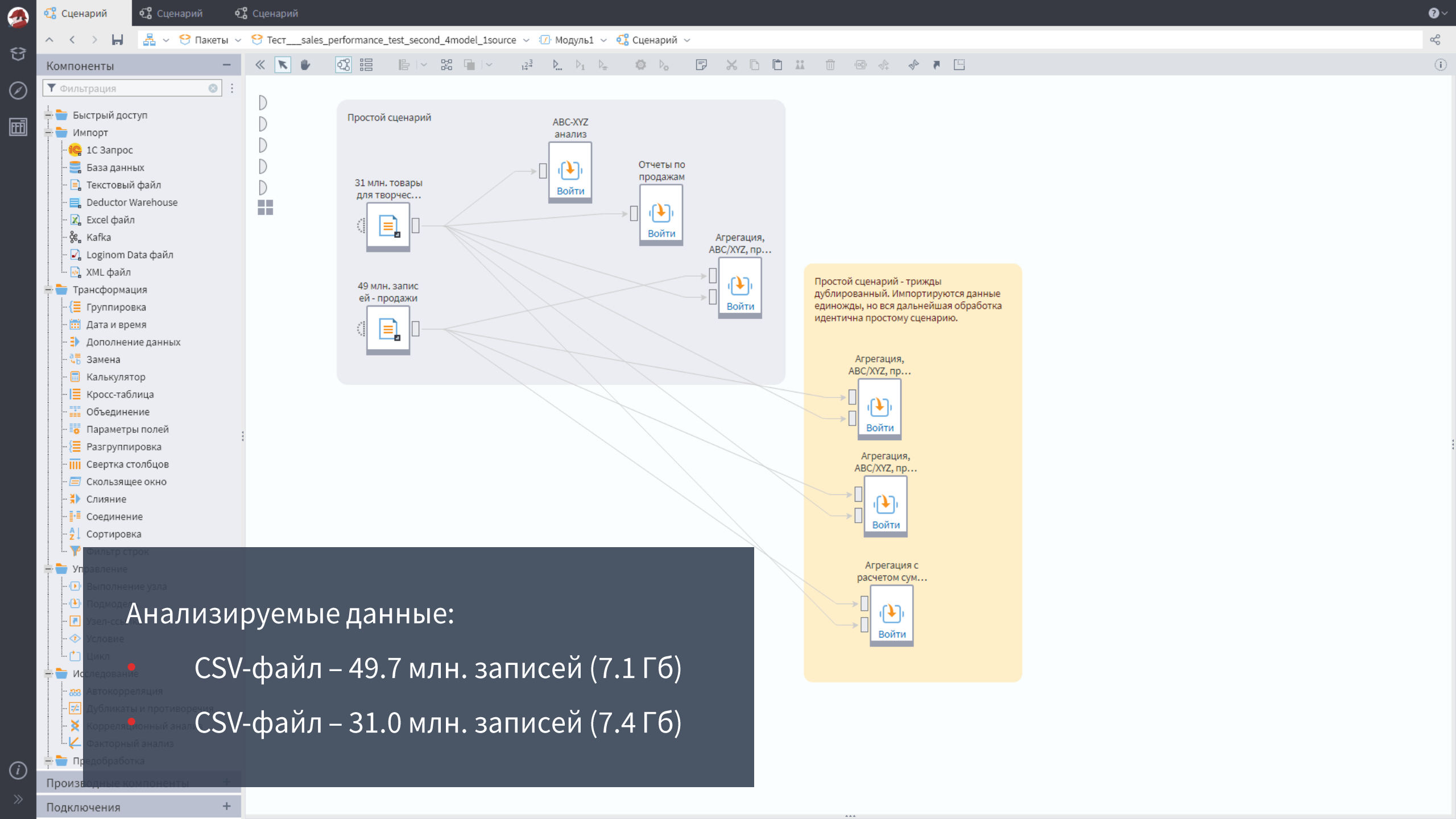
Сравнение платформ



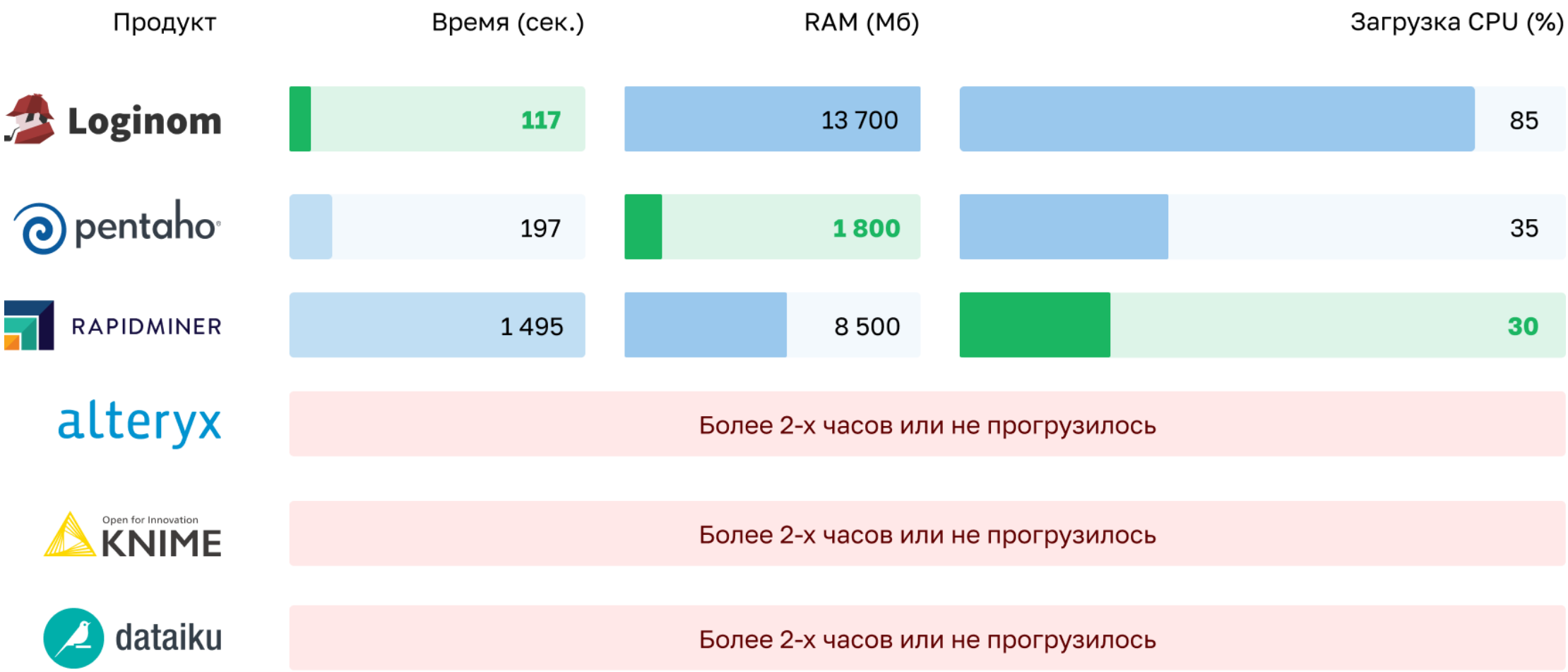
Данные необходимо загрузить в репозитарий, время загрузки суммировалось

Трижды продублированы ранее разработанные простые сценарии для демонстрации возможностей параллельной обработки.

Эмуляция ситуации при которой большие объемы данных извлекаются из нескольких источников, а затем обсчитываются множеством не очень сложных сценариев.


Сценарий
Сценарий
Сценарий
Пакеты
Тест___sales_performance_test_second_4model_1source
Модуль1
Сценарий
Компоненты
Фильтрация
Быстрый доступ
Импорт
1С Запрос
База данных
Текстовый файл
Deductor Warehouse
Excel файл
Kafka
Loginom Data файл
XML файл
Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Свертка столбцов
Скользящее окно
Слияние
Соединение
Сортировка
Фильтр строк
Управление
Выполнение узла
Подмодель
Узел-ссылка
Условие
Цикл
Исследование
Автокорреляция
Дубликаты и противоречия
Корреляционный анализ
Факторный анализ
Предобработка
Производные компоненты
Подключения
Простой сценарий
31 млн. товары для творчес...
49 млн. записей - продажи
ABC-XYZ анализ
Отчеты по продажам
Агрегация, ABC/XYZ, пр...
Войти
Простой сценарий - трижды дублированный. Импортируются данные единожды, но вся дальнейшая обработка идентична простому сценарию.
Агрегация, ABC/XYZ, пр...
Агрегация, ABC/XYZ, пр...
Агрегация с расчетом сум...
Анализируемые данные:
CSV-файл – 49.7 млн. записей (7.1 Гб)
CSV-файл – 31.0 млн. записей (7.4 Гб)

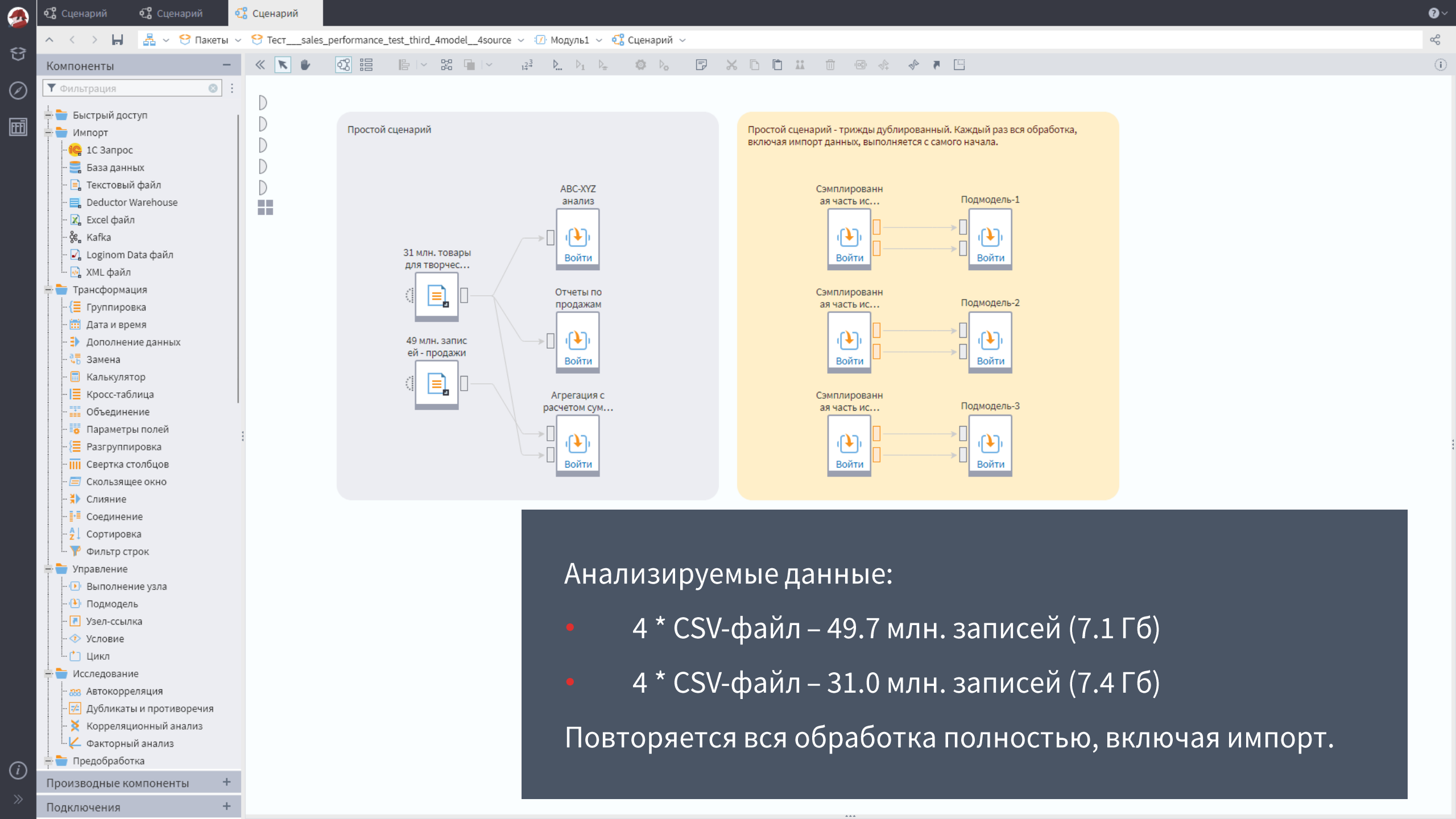
Сравнение платформ



Данные необходимо загрузить в репозитарий, время загрузки суммировалось

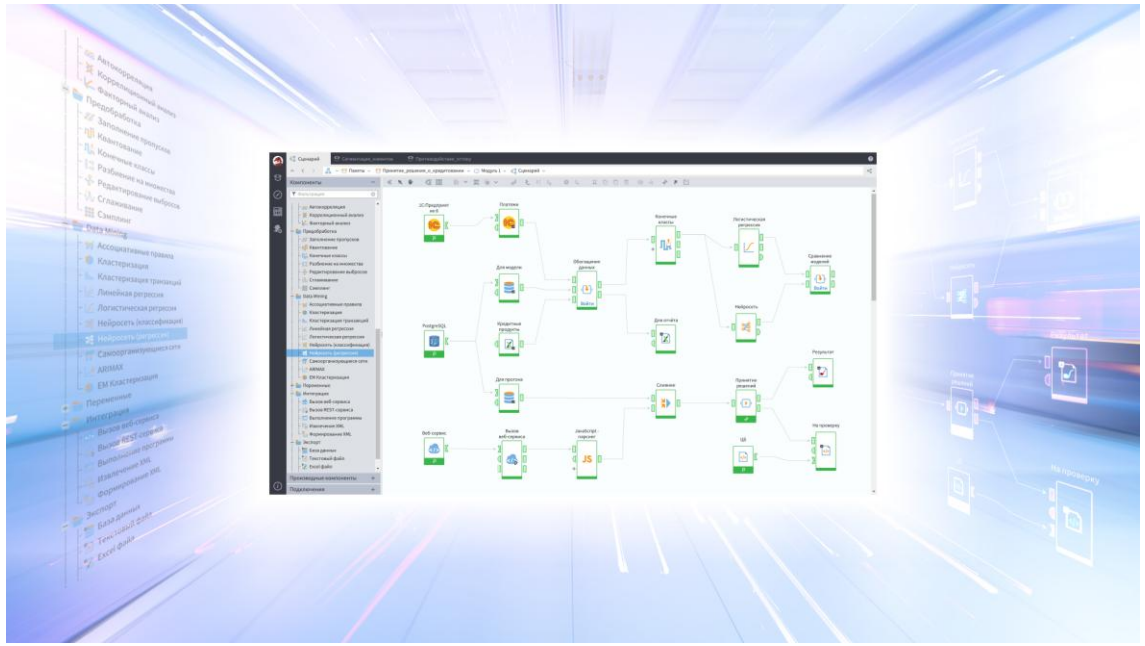
Трижды продублированы все этапы обработки от импорта данных до выгрузки результатов.

Эмуляция ситуации при которой регулярно поступают большие объемы новых данных, которые необходимо прогнать через все этапы обработки – типичный кейс работы высоконагруженных систем.


Сценарий
Сценарий
Сценарий
Пакеты
Тест___sales_performance_test_third_4model__4source
Модуль1
Сценарий
Компоненты
Фильтрация
Быстрый доступ
Импорт
1С Запрос
База данных
Текстовый файл
Deductor Warehouse
Excel файл
Kafka
Loginom Data файл
XML файл
Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Свертка столбцов
Скользящее окно
Слияние
Соединение
Сортировка
Фильтр строк
Управление
Выполнение узла
Подмодель
Узел-ссылка
Условие
Цикл
Исследование
Автокорреляция
Дубликаты и противоречия
Корреляционный анализ
Факторный анализ
Предобработка
Производные компоненты
Подключения
Простой сценарий
31 млн. товары для творчес...
49 млн. записей - продажи
ABC-XYZ анализ
Отчеты по продажам
Агрегация с расчетом сум...
Войти
Простой сценарий - трижды дублированный. Каждый раз вся обработка, включая импорт данных, выполняется с самого начала.
Сэмплированная часть ис...
Подмодель-1
Подмодель-2
Подмодель-3
Анализируемые данные:
4 * CSV-файл – 49.7 млн. записей (7.1 Гб)
4 * CSV-файл – 31.0 млн. записей (7.4 Гб)
Повторяется вся обработка полностью, включая импорт.

Сравнение платформ

Данные необходимо загрузить в репозитарий, время загрузки суммировалось



Loginom демонстрирует выдающуюся эффективность и производительность в сравнении с любой аналитической low-code платформой даже при использовании настольной редакции без тонких настроек. Чем больше источников и объемы данных, тем сильнее отрыв.