

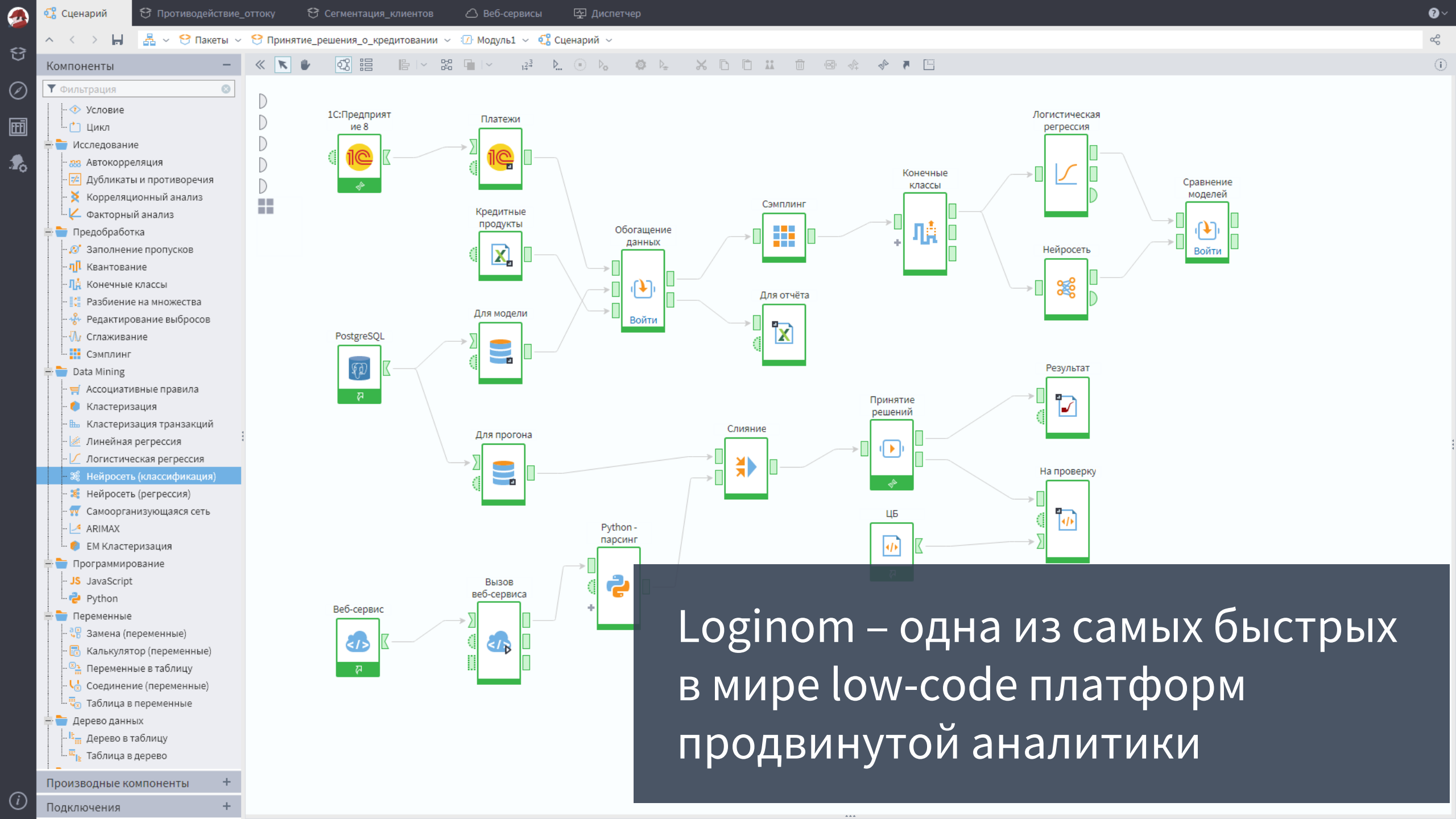


# **Loginom – low-code аналитика на предельной скорости**

Алексей Арустамов



Скорость – единственное свойство программ по которому есть консенсус: чем быстрее, тем лучше. Высокая скорость должна обеспечиваться без усилий, «из коробки».



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

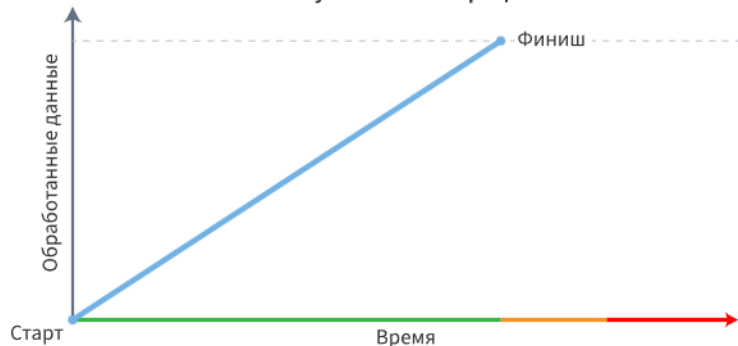
## Факторы, влияющие на производительность:

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

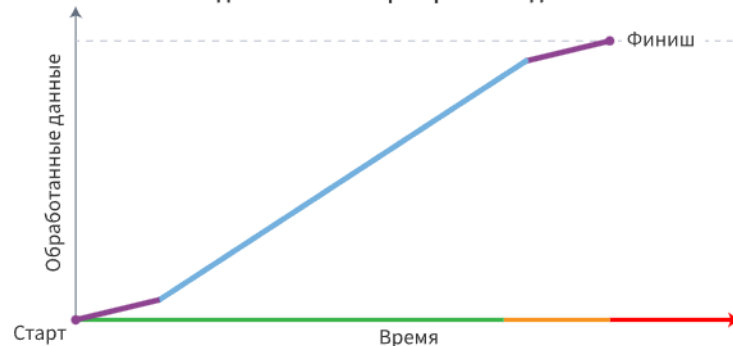
# Принципы оптимизации:

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

Отсутствие интеграций



Единый конвейер обработки данных



Составной конвейер обработки данных



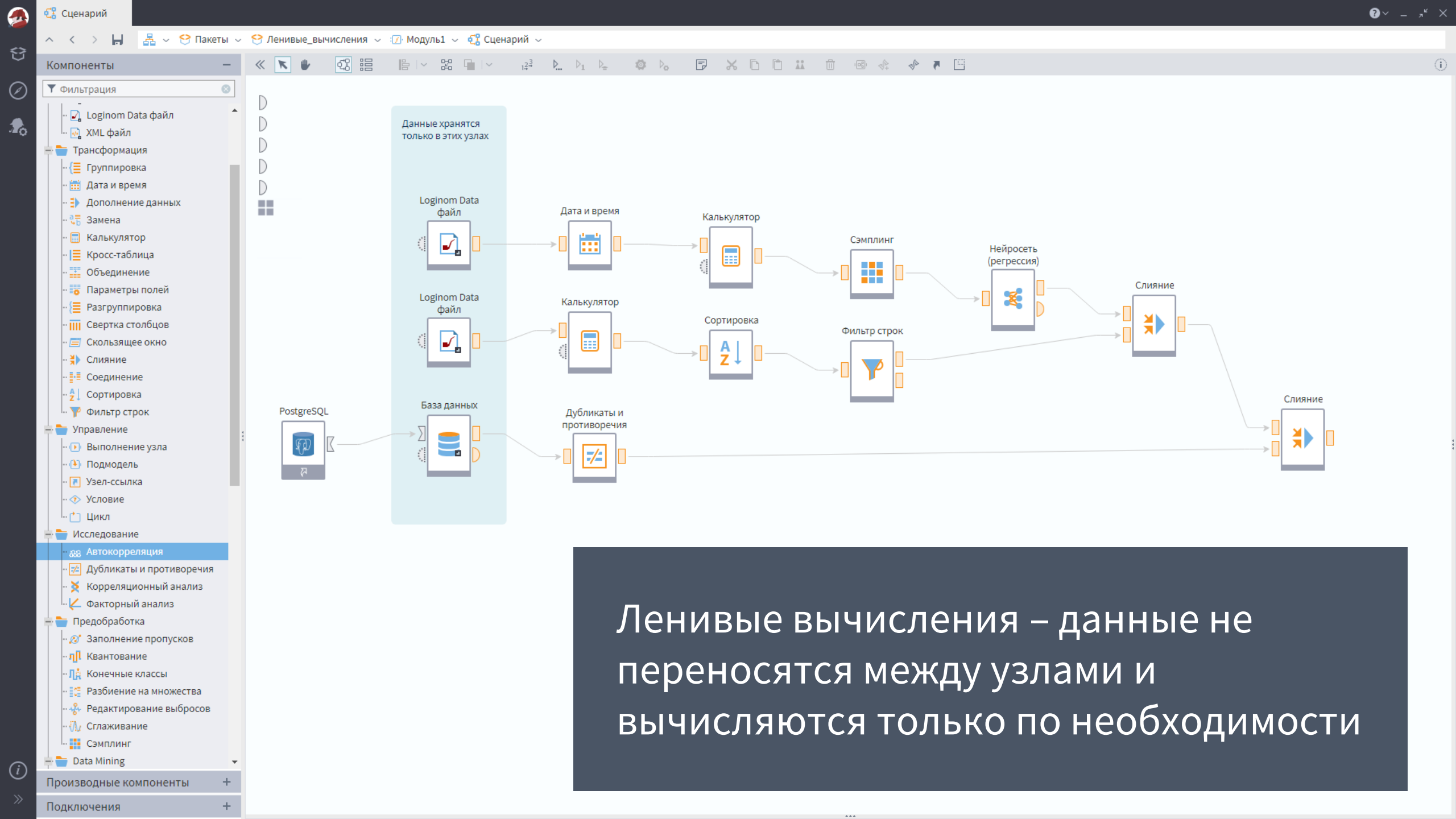
Чем больше в решении разнородных компонентов, тем больше точек интеграции и выше потери производительности

# Архитектуры платформы

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

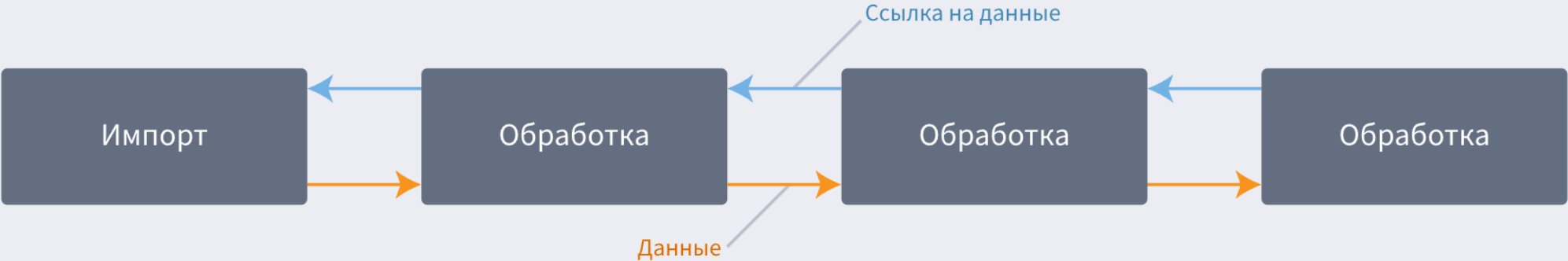
1. Работать с типизированными данными
2. Хранить в памяти уникальные значения
3. Использовать ленивые вычисления
4. ...



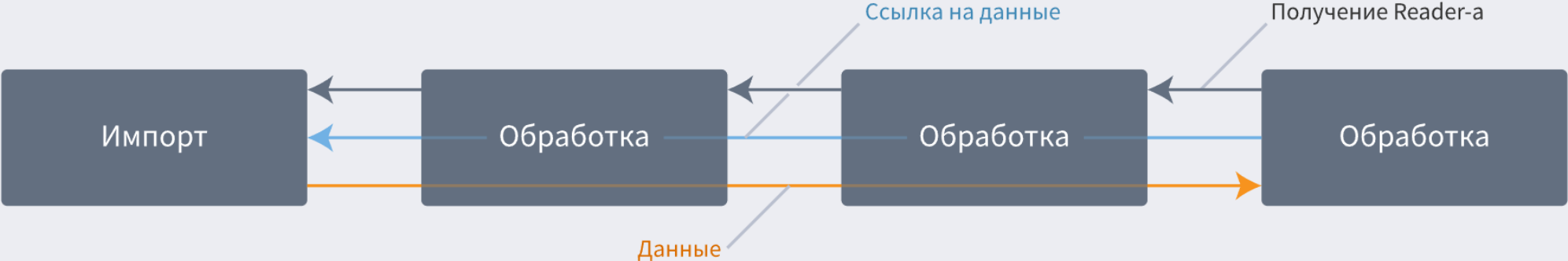


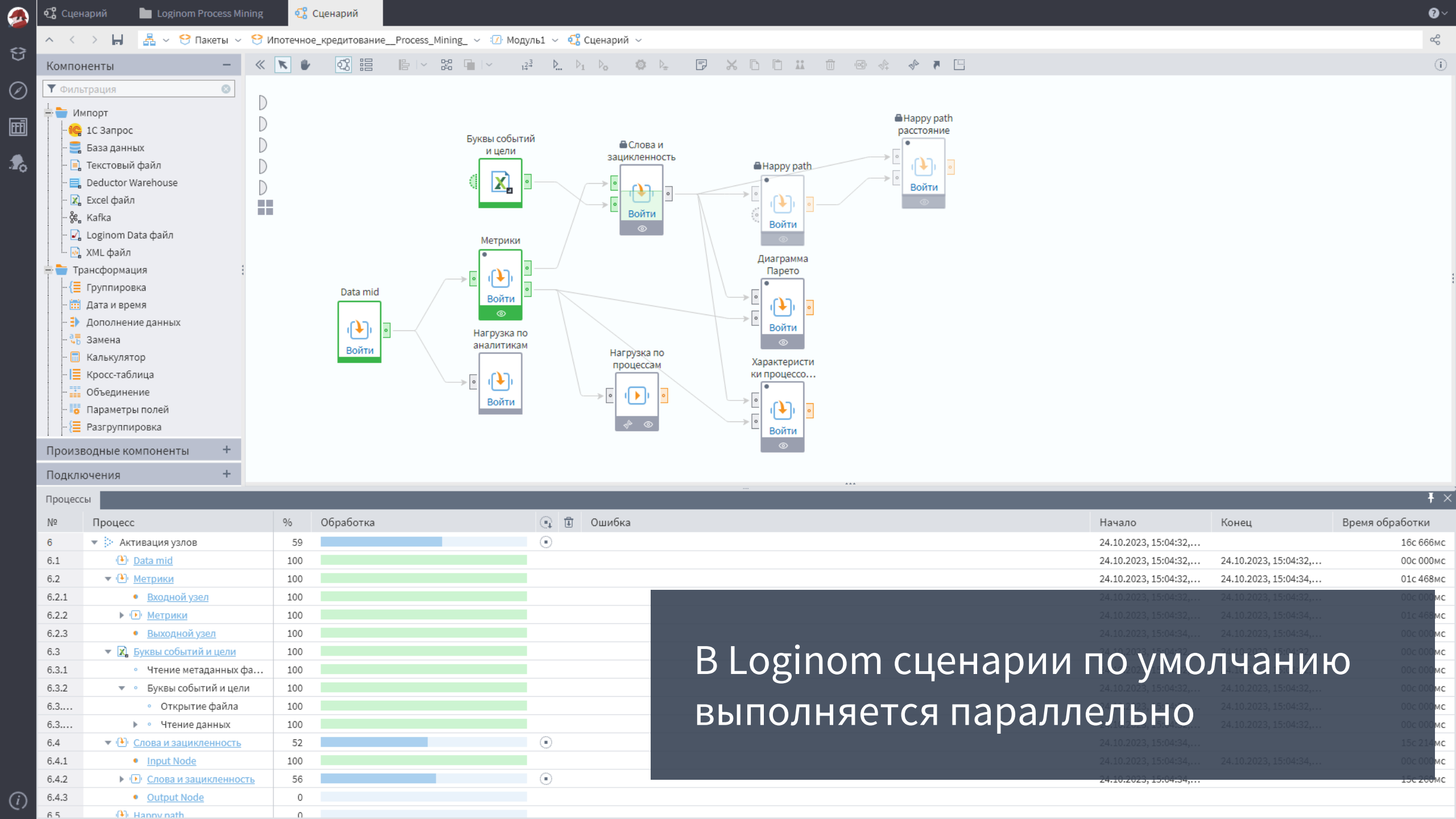
# Оптимизация ленивых вычислений

Было



Стало



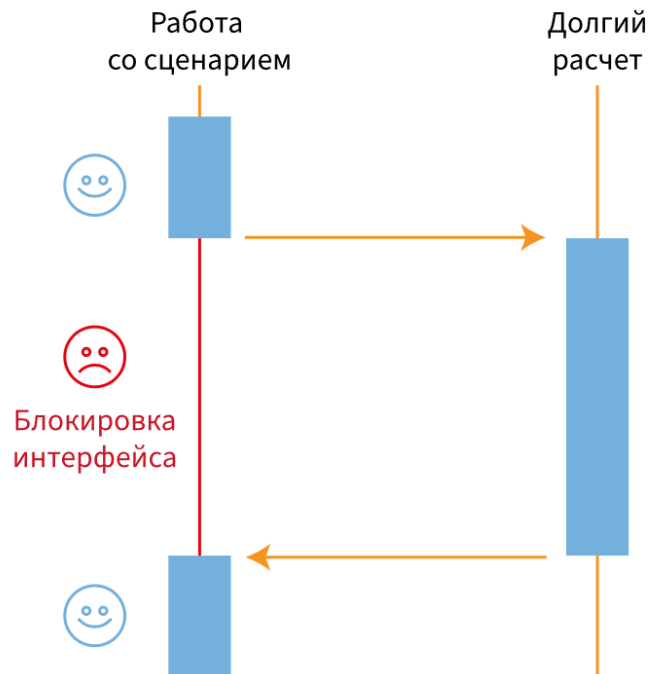


Производные компоненты +  
Подключения +

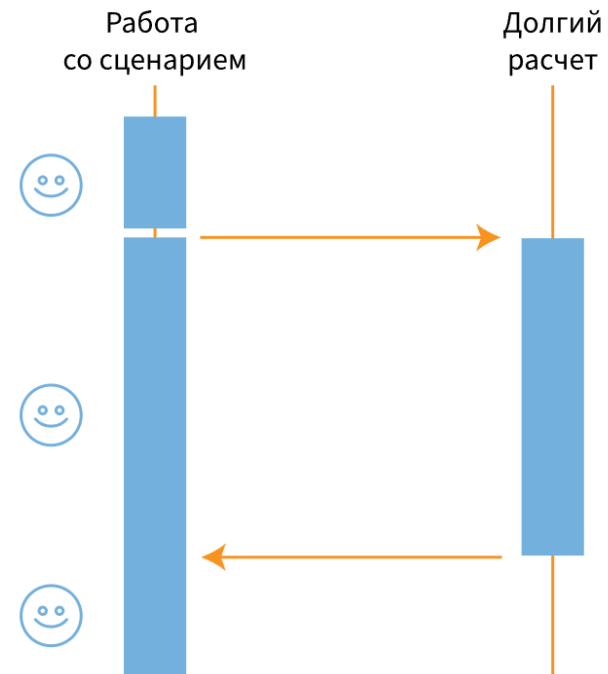
№	Процесс	%	Обработка	Ошибки	Начало	Конец	Время обработки
6	Активация узлов	59			24.10.2023, 15:04:32,...		16с 666мс
6.1	Data mid	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.2	Метрики	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:34,...	01с 468мс
6.2.1	Входной узел	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.2.2	Метрики	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:34,...	01с 468мс
6.2.3	Выходной узел	100			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	00с 000мс
6.3	Буквы событий и цели	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.3.1	Чтение метаданных фа...	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.3.2	Буквы событий и цели	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.3...	Открытие файла	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.3...	Чтение данных	100			24.10.2023, 15:04:32,...	24.10.2023, 15:04:32,...	00с 000мс
6.4	Слова и заикленность	52			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	15с 214мс
6.4.1	Input Node	100			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	00с 000мс
6.4.2	Слова и заикленность	56			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	15с 200мс
6.4.3	Output Node	0			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	15с 200мс
6.5	Happy path	0			24.10.2023, 15:04:34,...	24.10.2023, 15:04:34,...	15с 200мс

В LogiQL 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.

# 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 параметров полей

Кэширование

Выбранные поля

Отключено

При активации

При обращении

Выбранные поля

Метка	Имя поля	Тип	Значение	Состояние	Триггер	Включено
9.0 Доля покупки						
9.0 Сумма покупки	Summa_pokupki	Непрерывный	Не задано	При активации		<input type="checkbox"/>
9.0 Сумма покупки Сумма	Summa_pokupki_1	Непрерывный	Не задано	При активации		<input type="checkbox"/>
ab Покупатель	Pokupatel	Дискретный	Не задано	При активации		<input type="checkbox"/>
ab Филиал	Filial	Дискретный	Не задано	Отключено		<input type="checkbox"/>
31 Дата покупки	Data_pokupki	Непрерывный	Не задано	Отключено		<input type="checkbox"/>
12 Номер чека	Nomer_cheka	Непрерывный	Не задано	Отключено		<input type="checkbox"/>
ab Название товара	Nazvanie_tovara	Дискретный	Не задано	При обращении		<input type="checkbox"/>
ab Бренд	Brend	Дискретный	Не задано	При обращении		<input type="checkbox"/>
ab Товарная группа	Tovarnaya_gruppa	Дискретный	Не задано	При обращении		<input type="checkbox"/>
12 Количество	Kolichestvo	Непрерывный	Не задано	Отключено		<input type="checkbox"/>
9.0 Сумма скидки	Summa_skidki	Непрерывный	Не задано	Отключено		<input type="checkbox"/>

По умолчанию всё считается на лету, но доступно гибкое кэширование: 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.

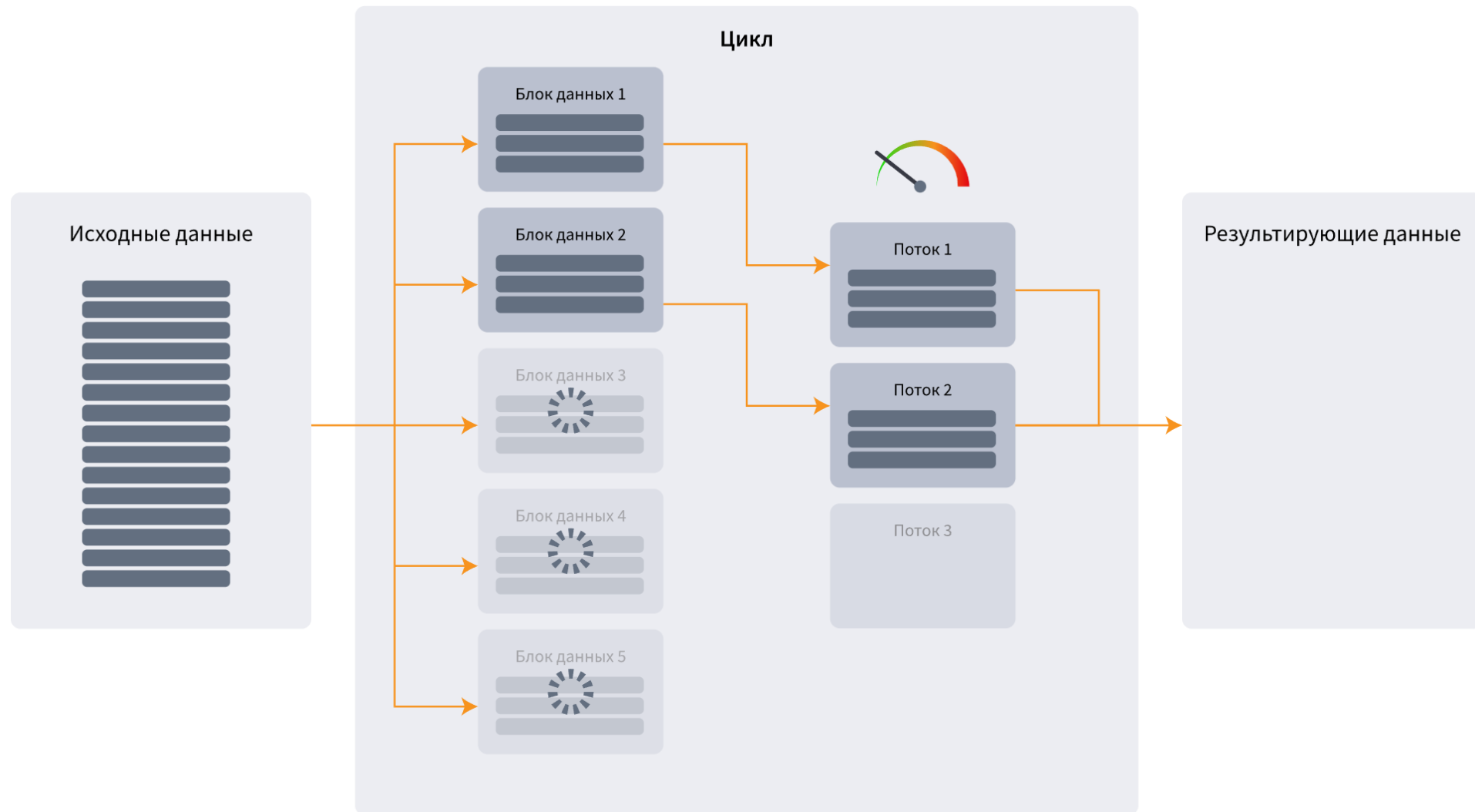


# Низкоуровневые операции

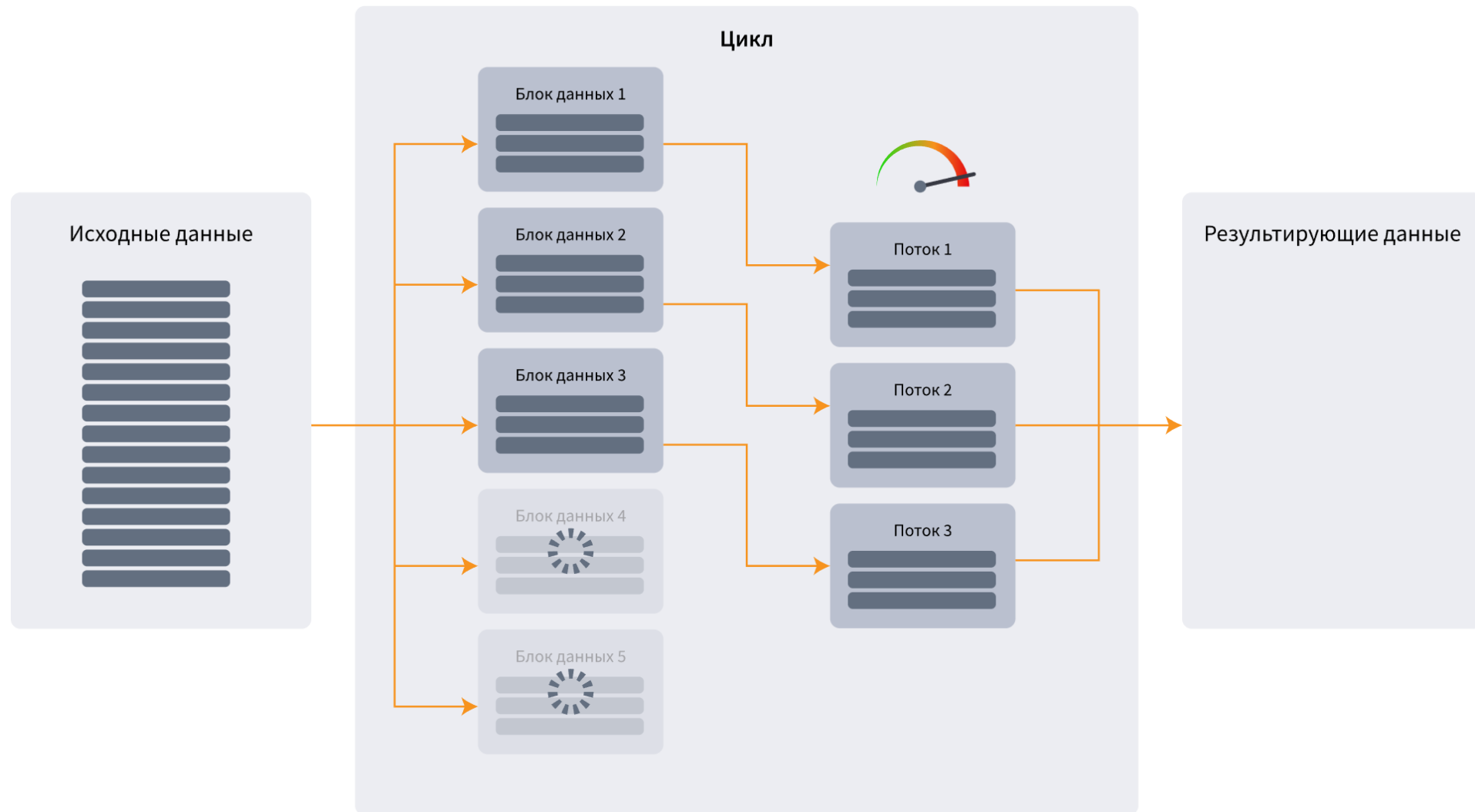
## Собственная реализация многопоточной обработки:

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

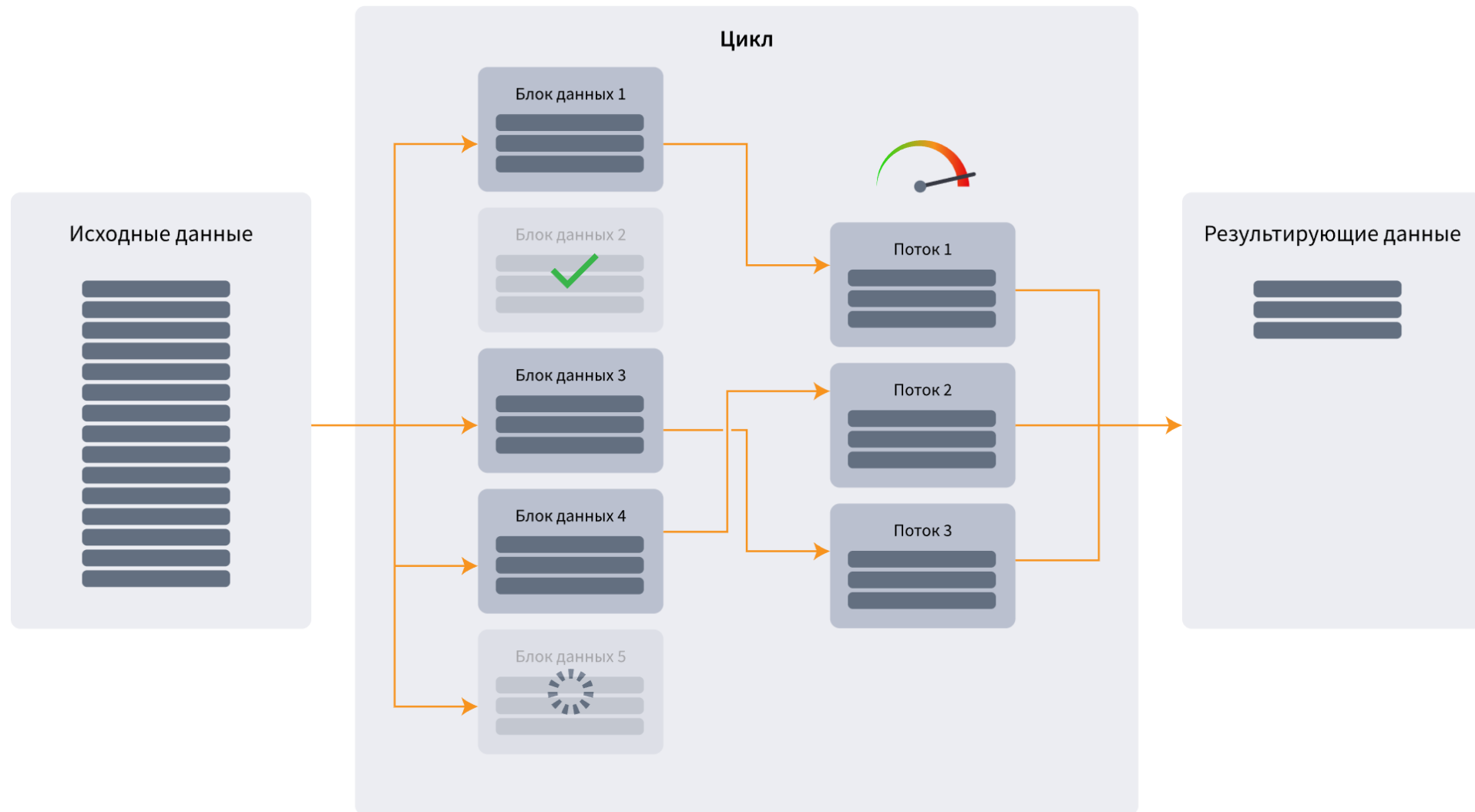
# Параллельный цикл в многопроцессорных системах (1)



## Параллельный цикл в многопроцессорных системах (2)

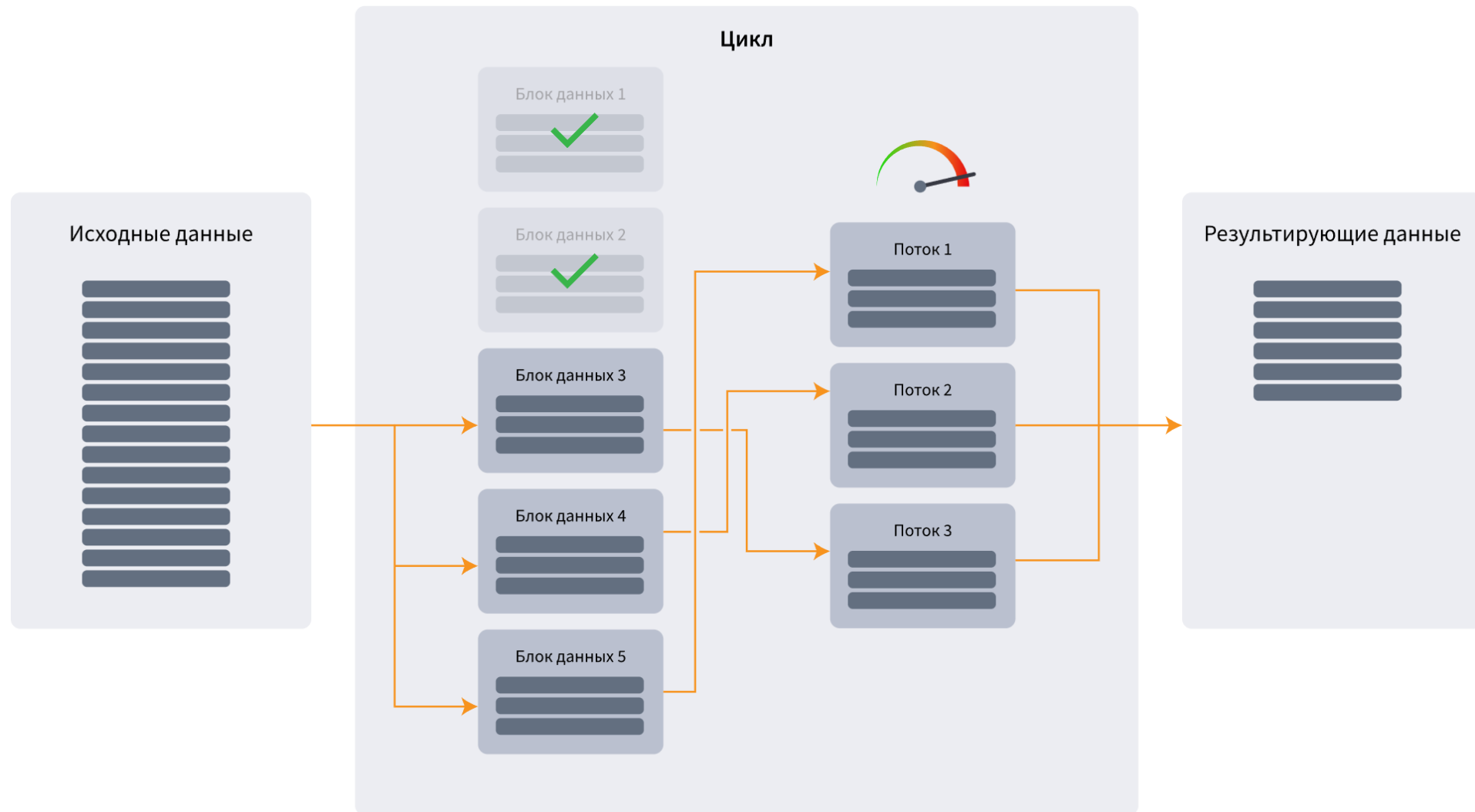


# Параллельный цикл в многопроцессорных системах (3)

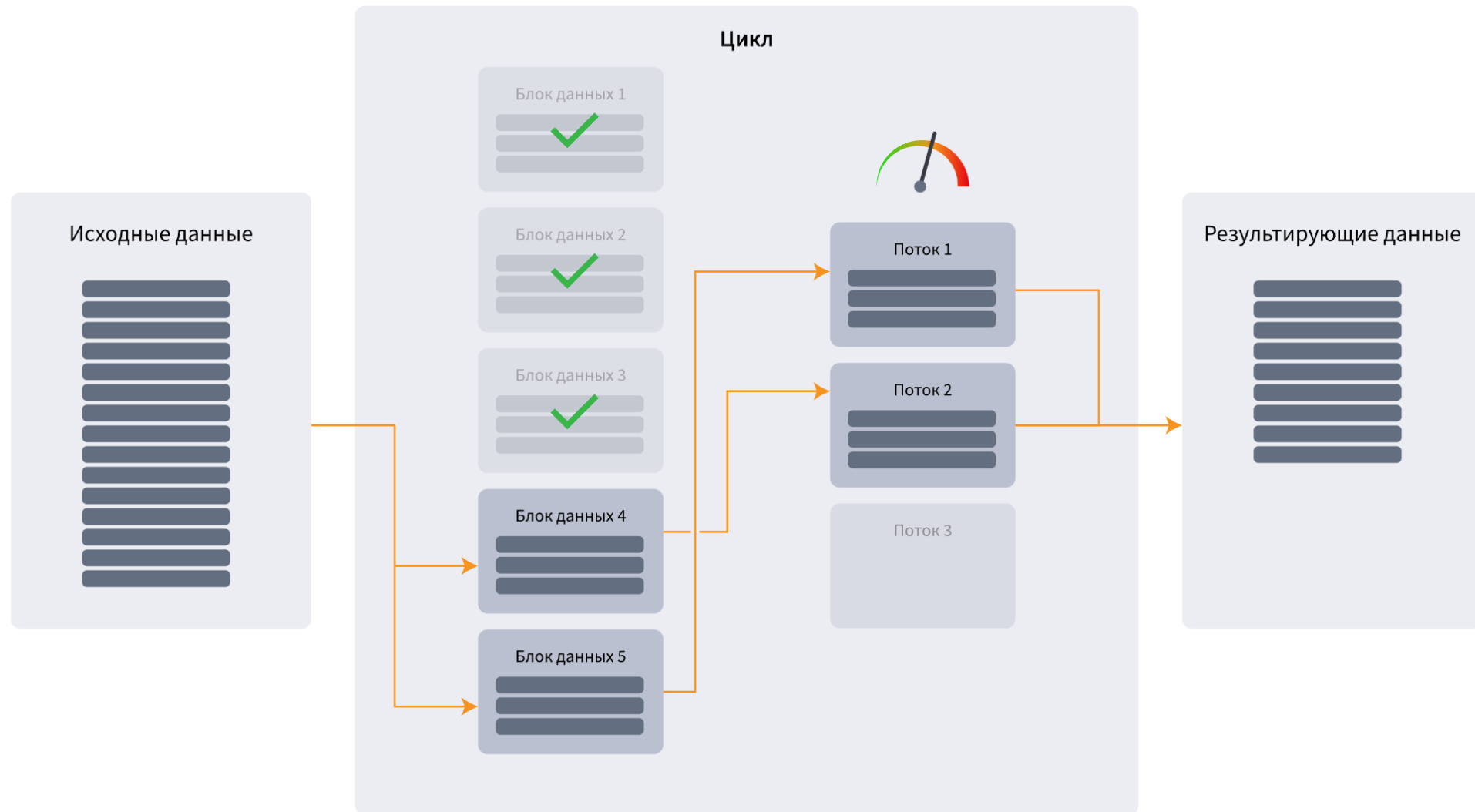




# Параллельный цикл в многопроцессорных системах (4)



# Параллельный цикл в многопроцессорных системах (5)



# Параллельный цикл в многопроцессорных системах (6)



## Эффективная работа со строками

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 `wuhash` (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.

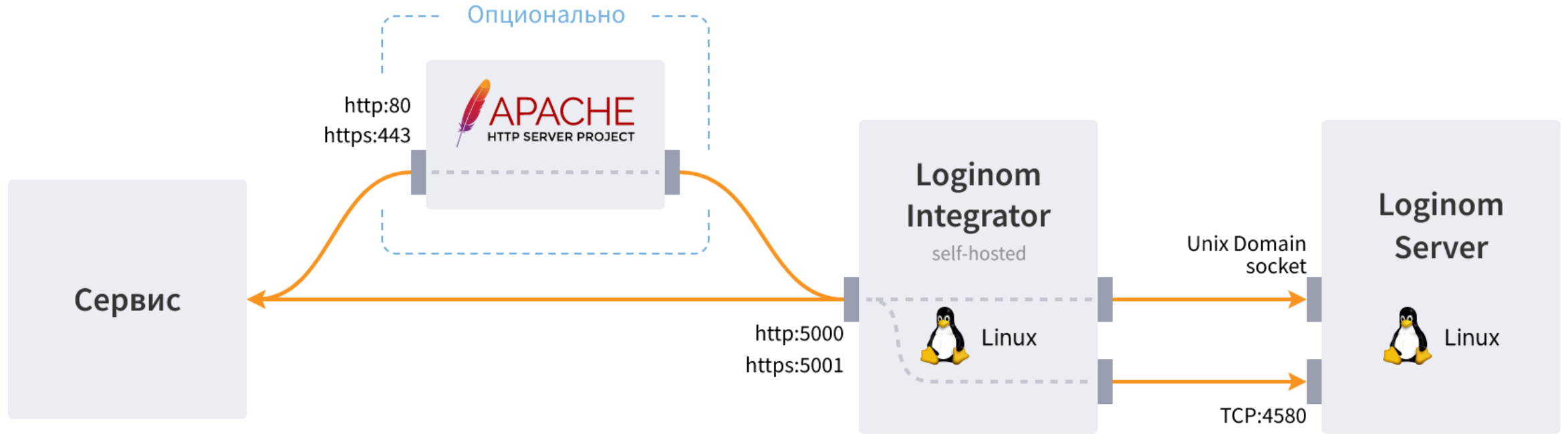
## Оптимизация формата Loginom Data File:

- 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.



# 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%.

## Встраивание Python в сценарий

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.

# Сравнение с конкурентами

Сравнение настольных редакций популярных платформ low-code-аналитики

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

# С Logiном сравниваются лучше мировые продукты по мнению Garter

 Alteryx Designer ×  
by Alteryx  
39 ratings

**Overall Rating** 87% willing to recommend

**4.2**/5

★★★★★ (39 Reviews)

5 Star	46%
4 Star	46%
3 Star	8%
2 Star	0%
1 Star	0%


 Dataiku ×  
by Dataiku  
147 ratings

**Overall Rating** 94% willing to recommend

**4.6**/5

★★★★★ (147 Reviews)

5 Star	67%
4 Star	32%
3 Star	1%
2 Star	0%
1 Star	0%


 KNIME Analytics Platform ×  
by KNIME  
48 ratings

**Overall Rating** 96% willing to recommend

**4.7**/5

★★★★★ (48 Reviews)

5 Star	67%
4 Star	31%
3 Star	2%
2 Star	0%
1 Star	0%


 Pentaho Data Integration and Analytics ×  
by Hitachi Vantara  
64 ratings

**Overall Rating** 72% willing to recommend

**4.3**/5

★★★★★ (64 Reviews)

5 Star	30%
4 Star	48%
3 Star	20%
2 Star	2%
1 Star	0%


 PolyAnalyst ×  
by Megaputer  
3 ratings

**Overall Rating** 100% willing to recommend

**4.8**/5

★★★★★ (3 Reviews)

5 Star	67%
4 Star	33%
3 Star	0%
2 Star	0%
1 Star	0%

 RapidMiner Studio ×  
by Altair  
71 ratings

**Overall Rating** 90% willing to recommend

**4.4**/5

★★★★★ (71 Reviews)

5 Star	55%
4 Star	42%
3 Star	3%
2 Star	0%
1 Star	0%

## 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

Workflow - Configuration

Canvas Workflow Runtime Events Meta Info

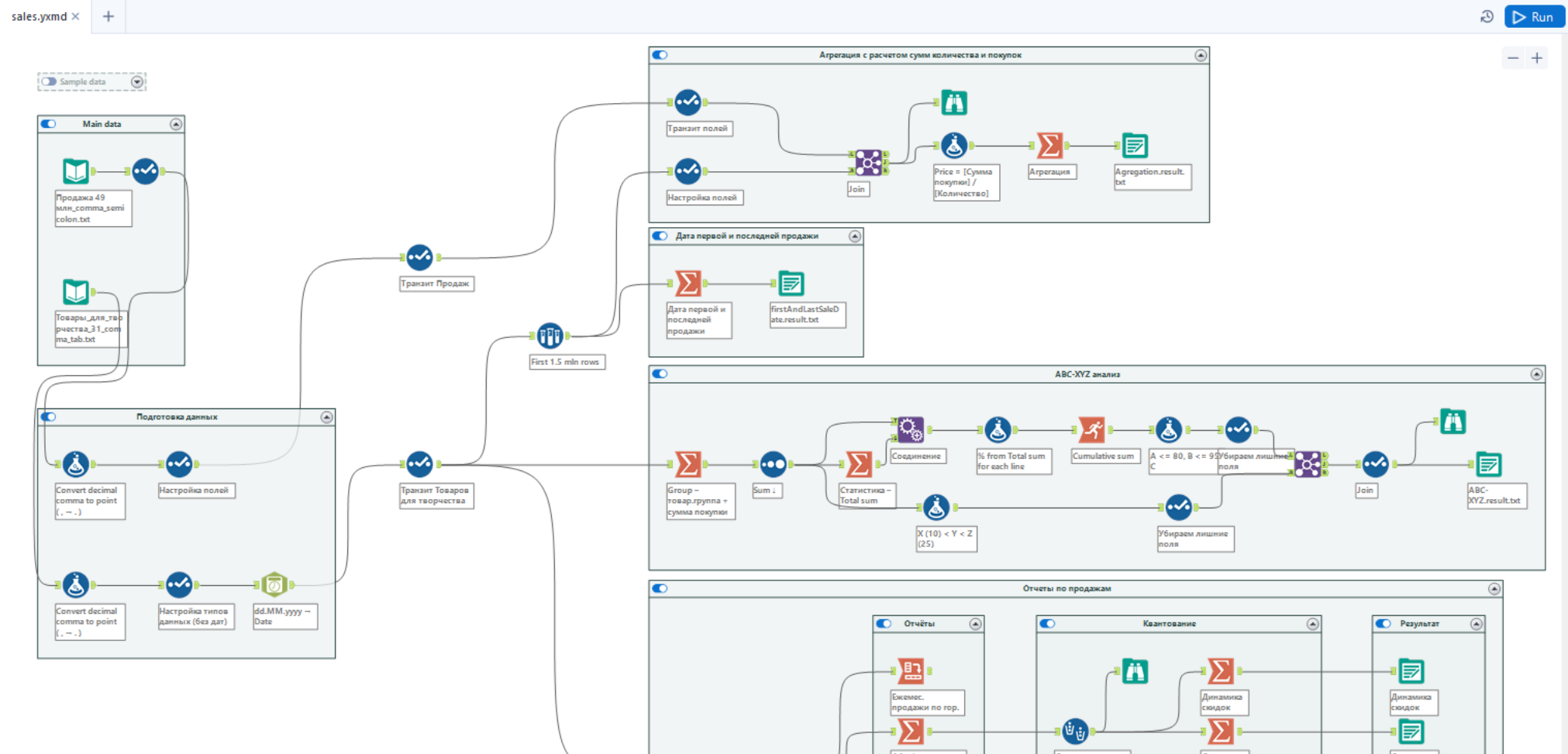
CANVAS OPTIONS

Layout Direction  
Horizontal

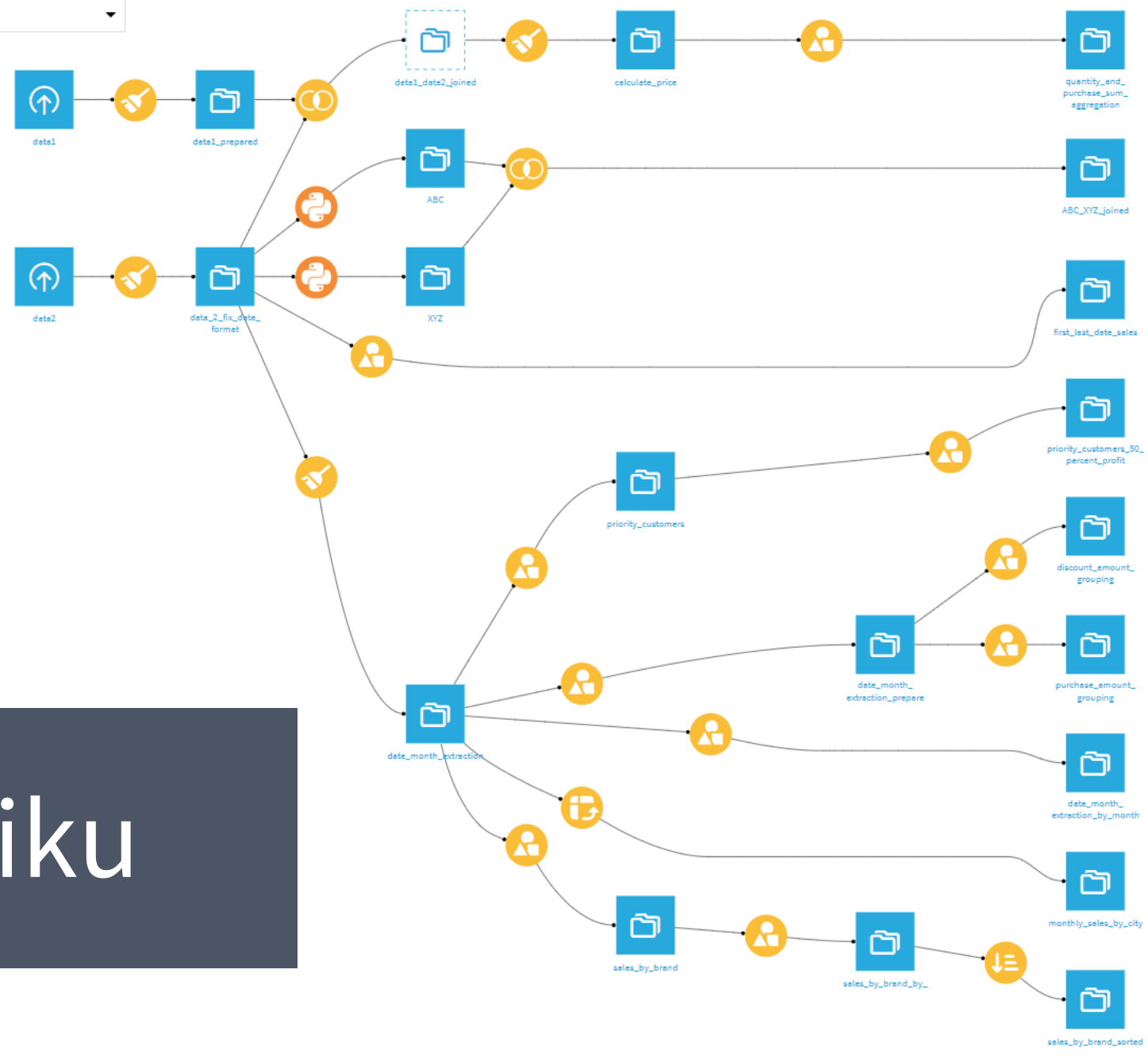
Annotations  
Show

Connection Progress  
Show Only When Running

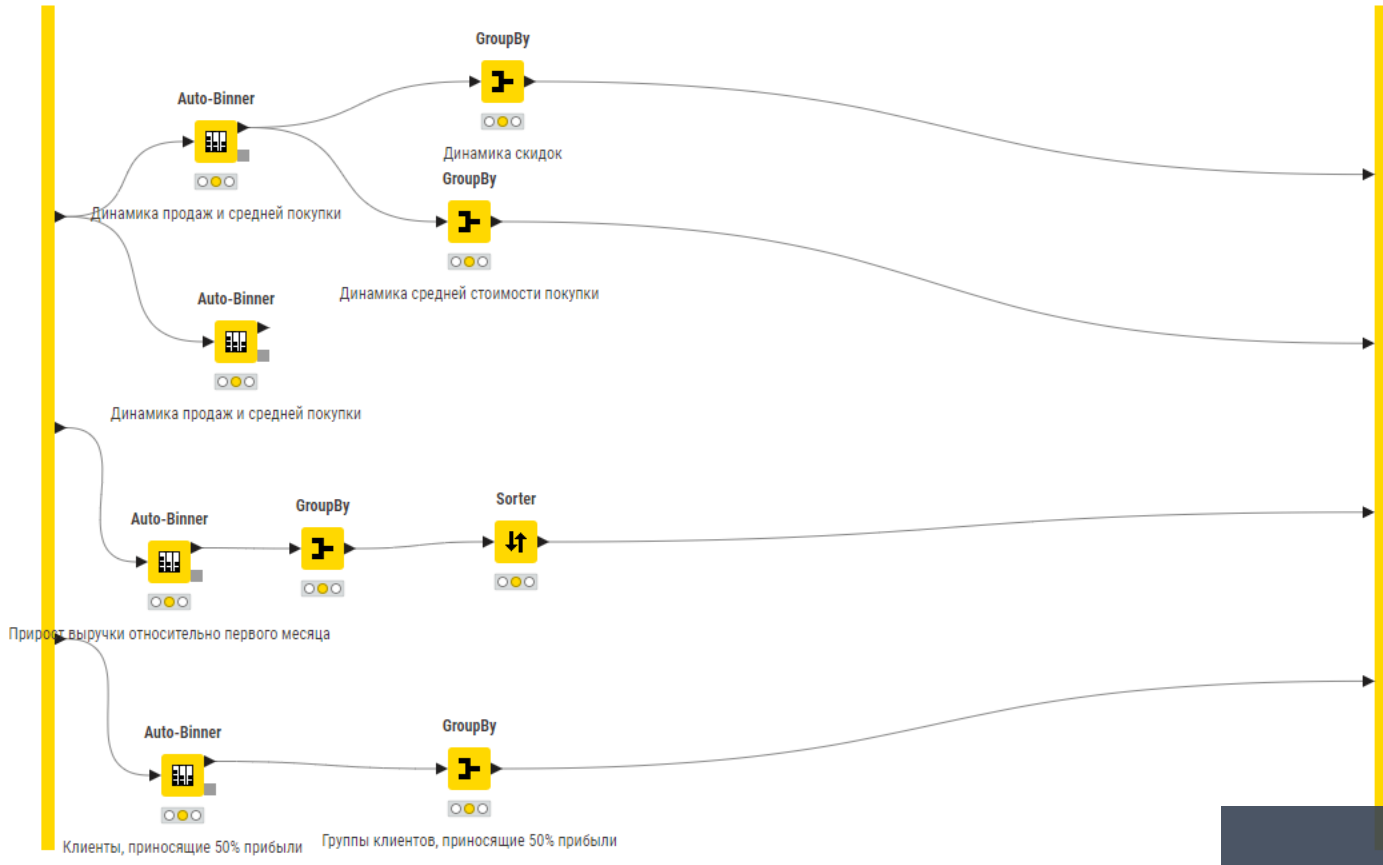
Overview







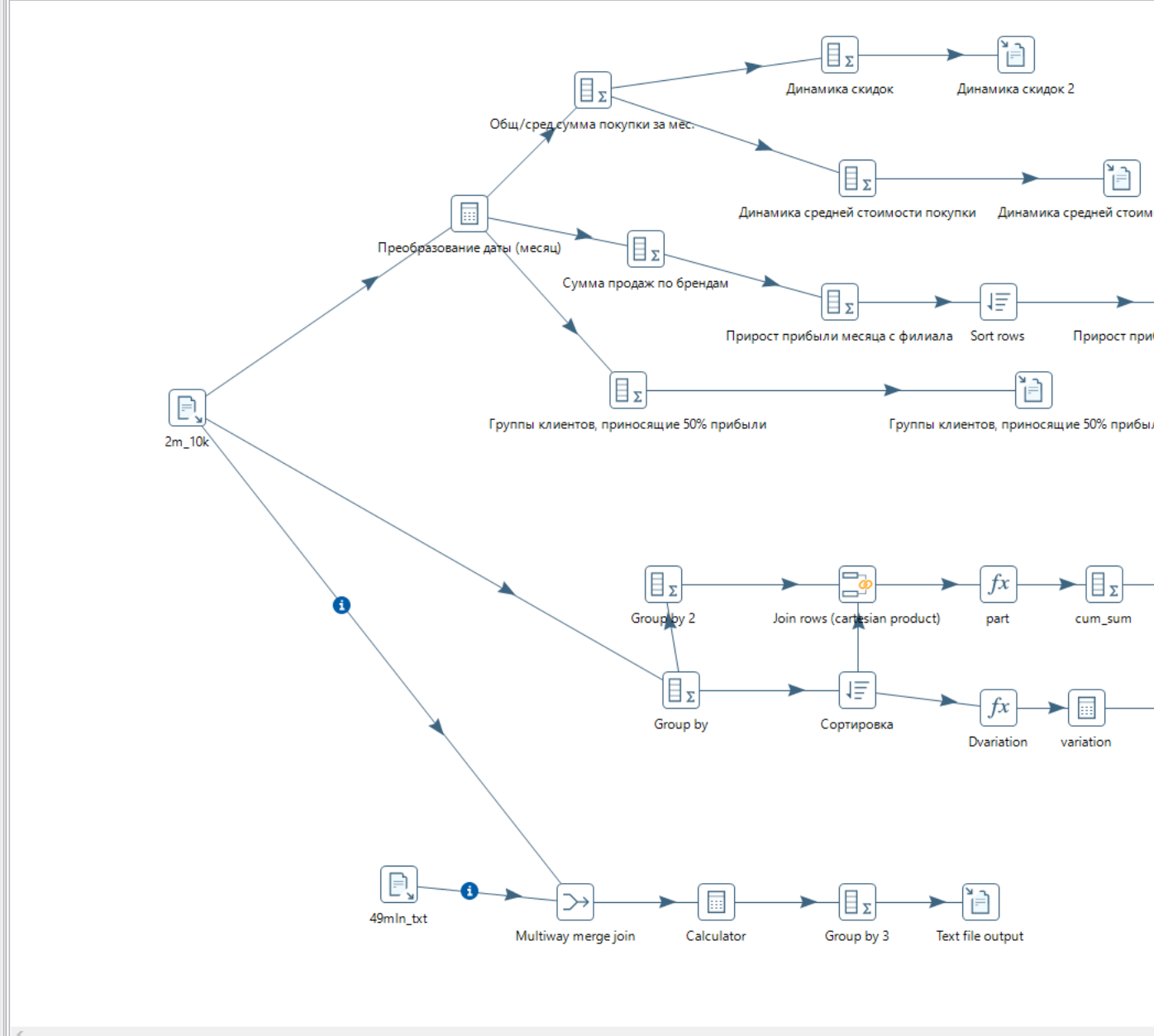
# Dataiku



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

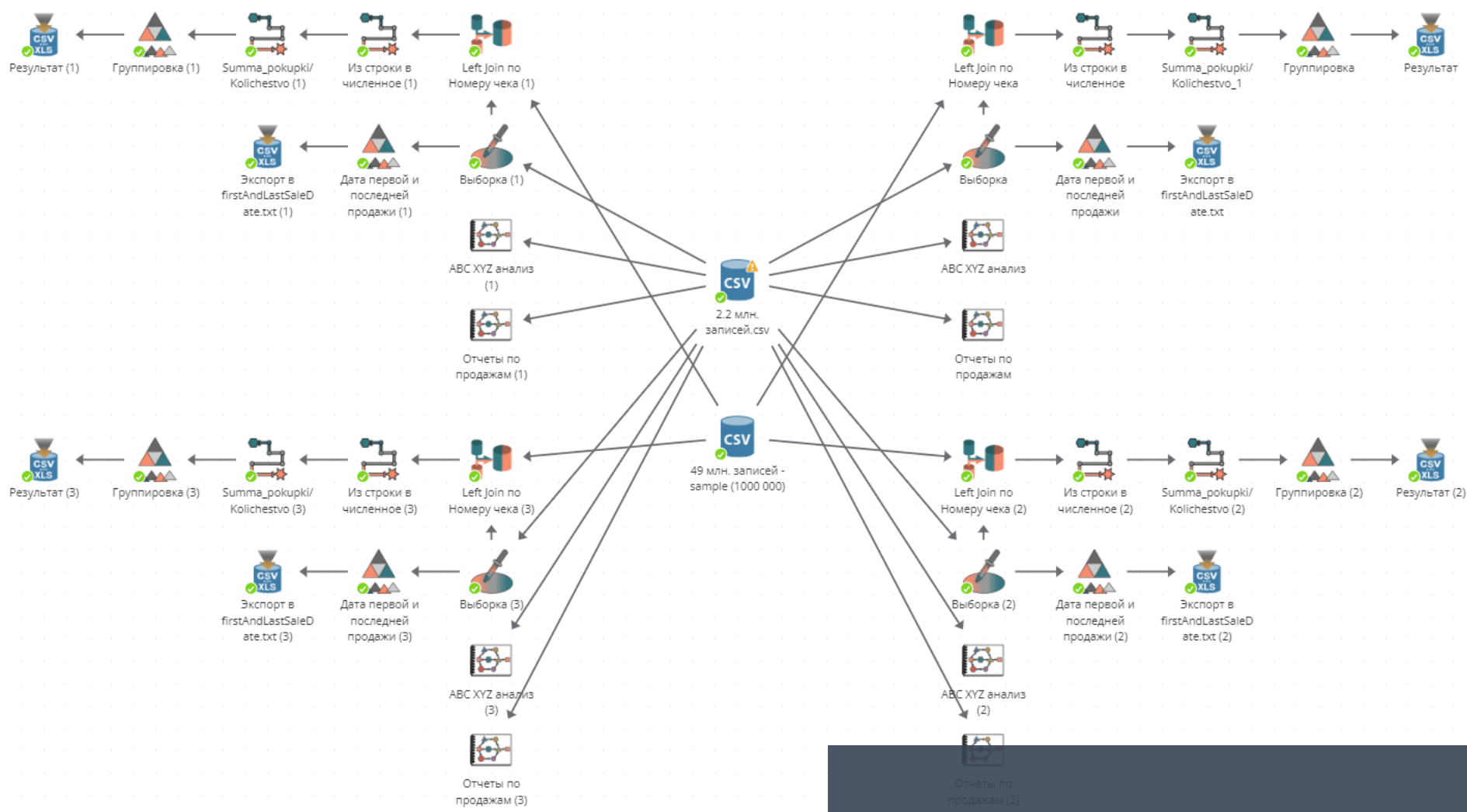
- Transformations
  - Один поток
    - Run configurations
    - Database connections
    - Steps
    - Hops
    - Partition schemas
    - Slave server
    - Kettle cluster schemas
    - VFS Connections

# Pentaho



Палитра узлов

- Анализ текстов**
- Автомат... Анализ таксоном...
  - Анализ тонально... Анализ трендов
  - Аноними... Восстано... Извлече... сущностей сущностей сущностей
  - Извлече... Извлече... Извлече... медицин... сущностей сущностей
  - Извлече... ИИ Индекс фактов Ассистент
  - Классиф... Классиф... Кластери... на основе текстов текстов
  - Метрики Объедин... Определ... текста таксоном... языка
  - Перевод Поиск... Проверка текстов запрос граммати...
  - Проверка Разметка Расшифр... орфогра... текста сокраще...
  - Резюме Связь Сравнение терминов текстов
  - Схожесть Таксоно... Тематика текстов текстов



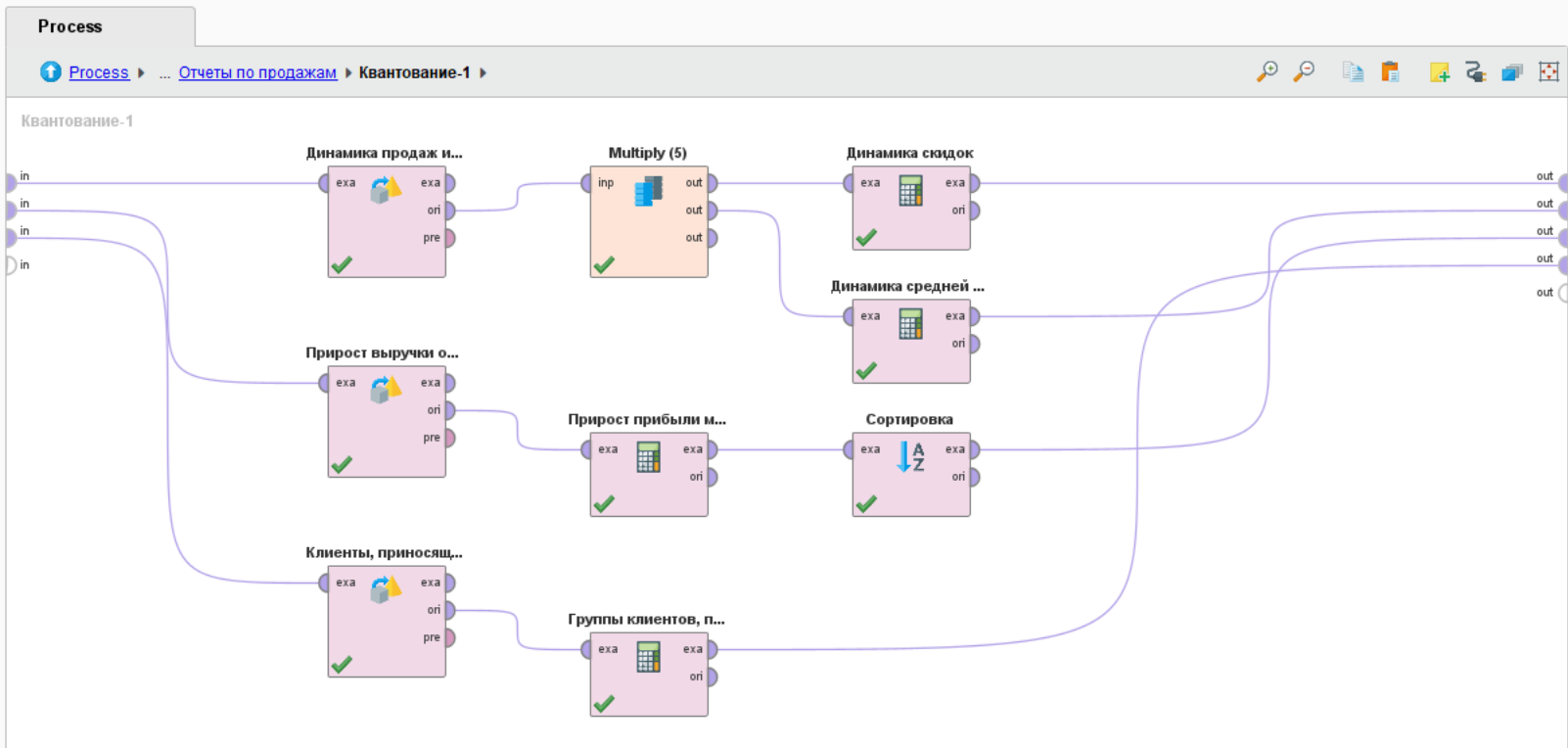
PolyAnalyst



**Repository**

Import Data

- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)



**Parameters**

Квантование-1 (Subprocess)

No parameters to display.

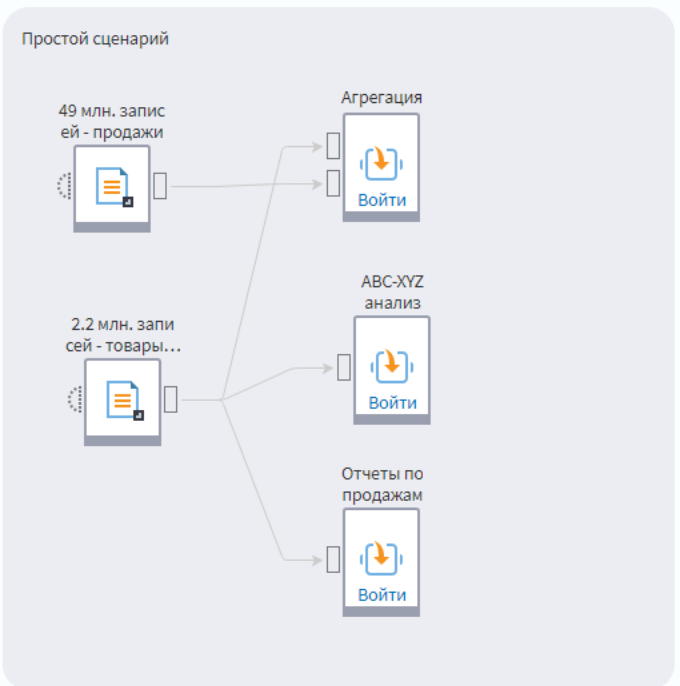
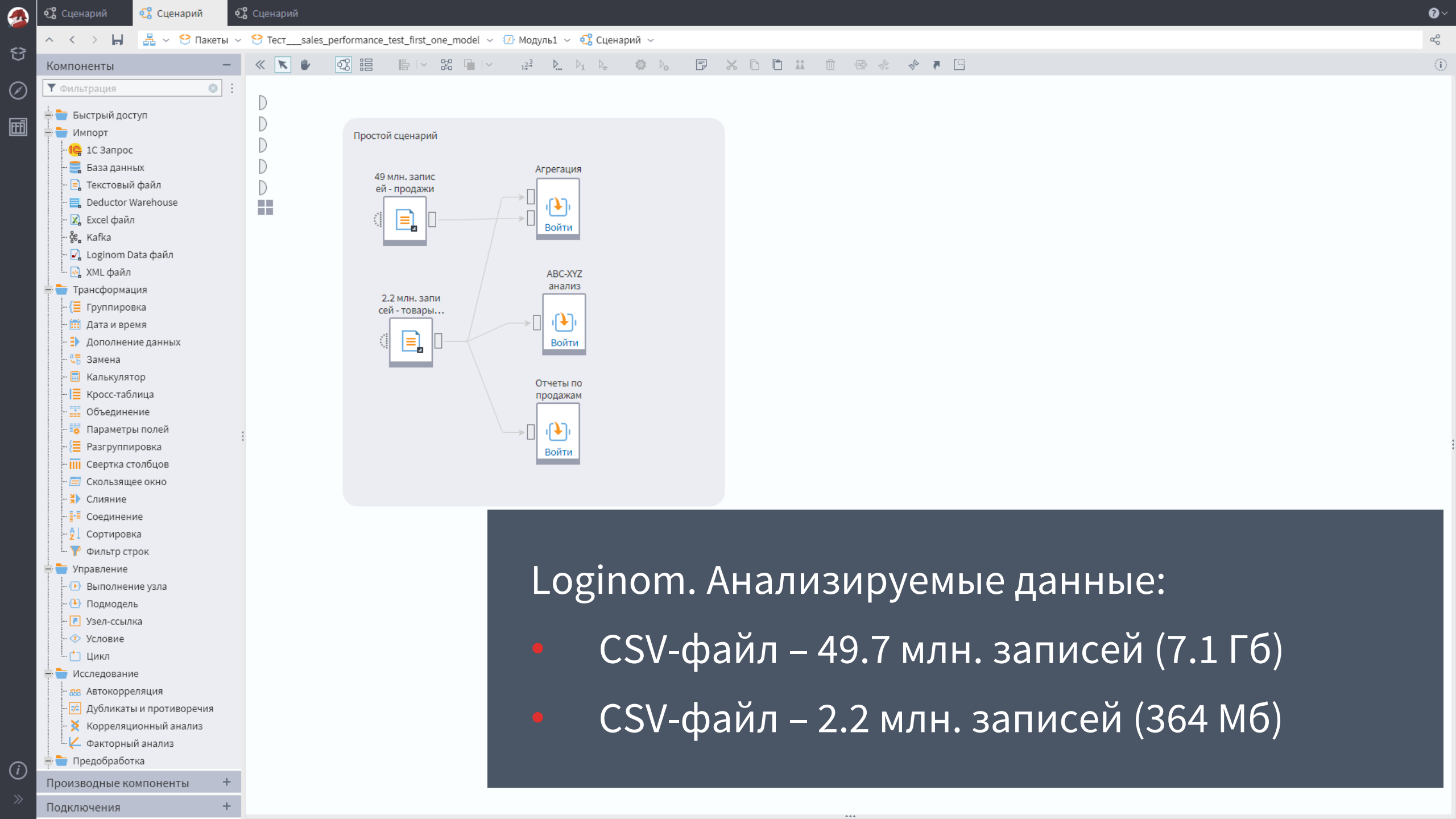
**Operators**

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](#)

# RapidMiner



## Loginom. Анализируемые данные:

- CSV-файл – 49.7 млн. записей (7.1 Гб)
- CSV-файл – 2.2 млн. записей (364 Мб)

## Сравнение платформ

Продукт	Время (сек.)	RAM (Мб)	Загрузка CPU (%)
Loginom	66	4 839	15
Alteryx	73	10 000	50
Pentaho	170	1 800	15
RapidMiner*	801	8 102	40
PolyAnalyst	920	4 720	40
KNIME	4 560	22 500	30
<b>Dataiku</b>	<b>Более 2-х часов или не прогрузилось</b>		

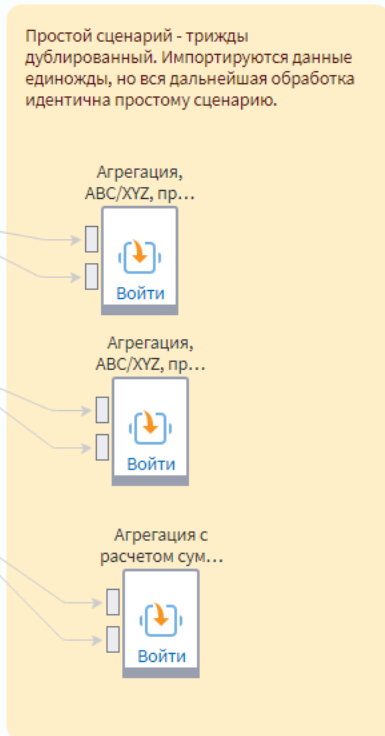
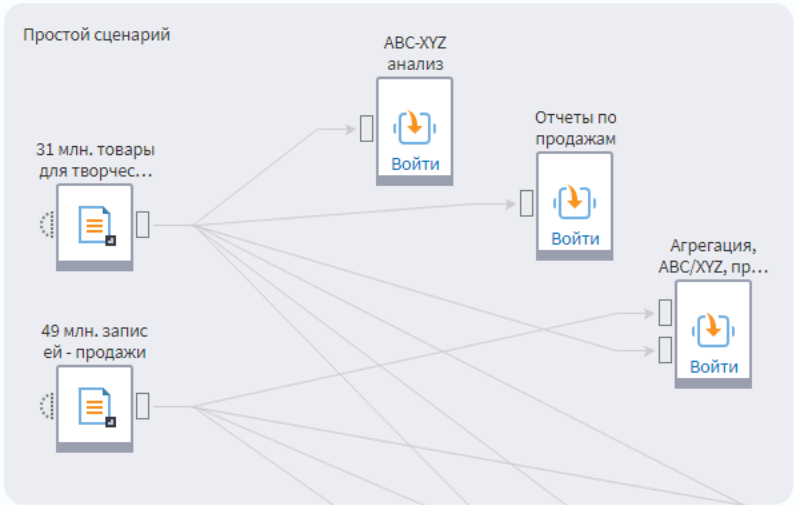
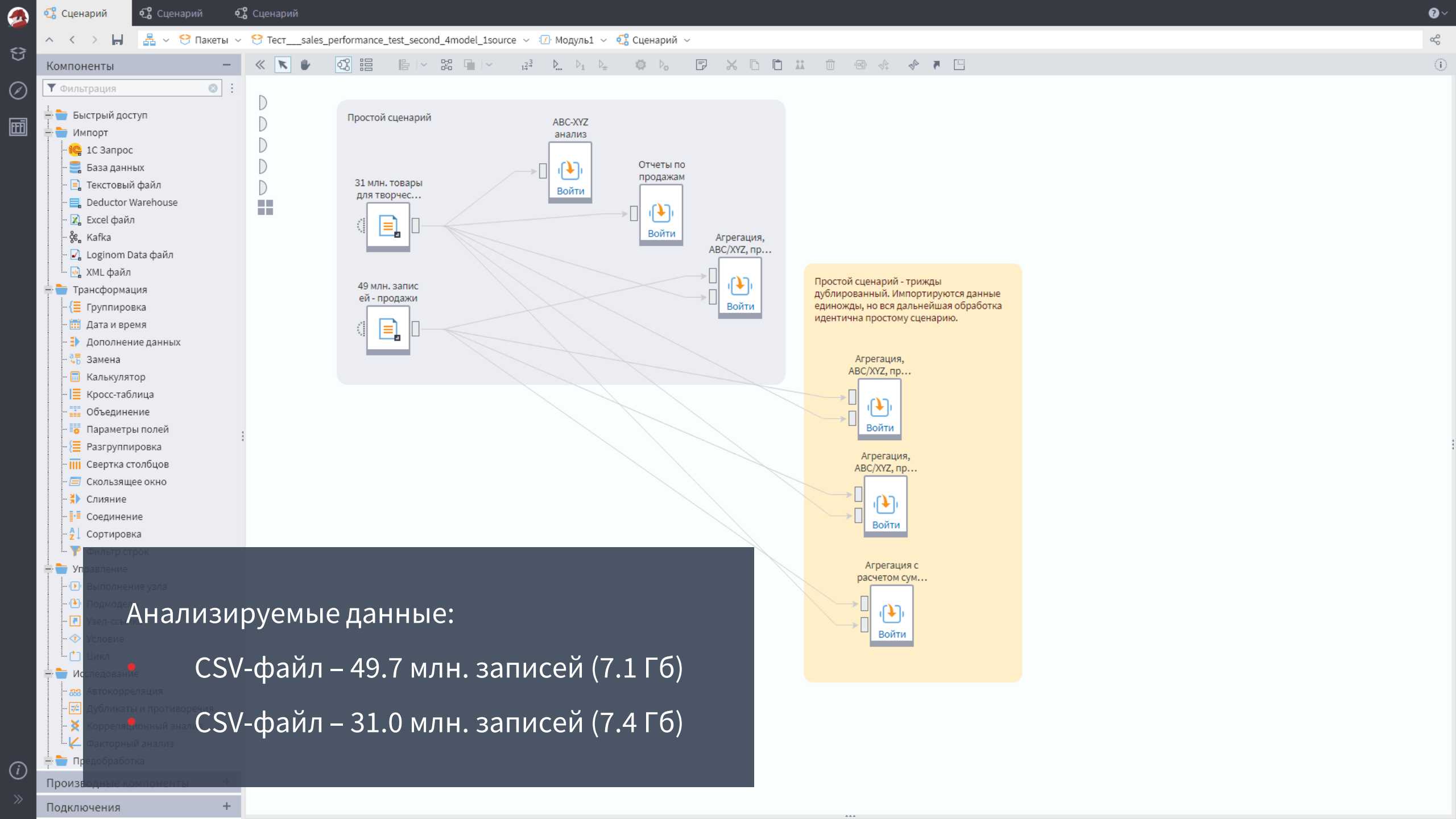
\* Данные необходимо загрузить в репозиторий, время загрузки суммировалось

Задача 2. Большие объемы данных, много расчетов

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.





Анализируемые данные:

CSV-файл – 49.7 млн. записей (7.1 Гб)

CSV-файл – 31.0 млн. записей (7.4 Гб)

## Сравнение платформ

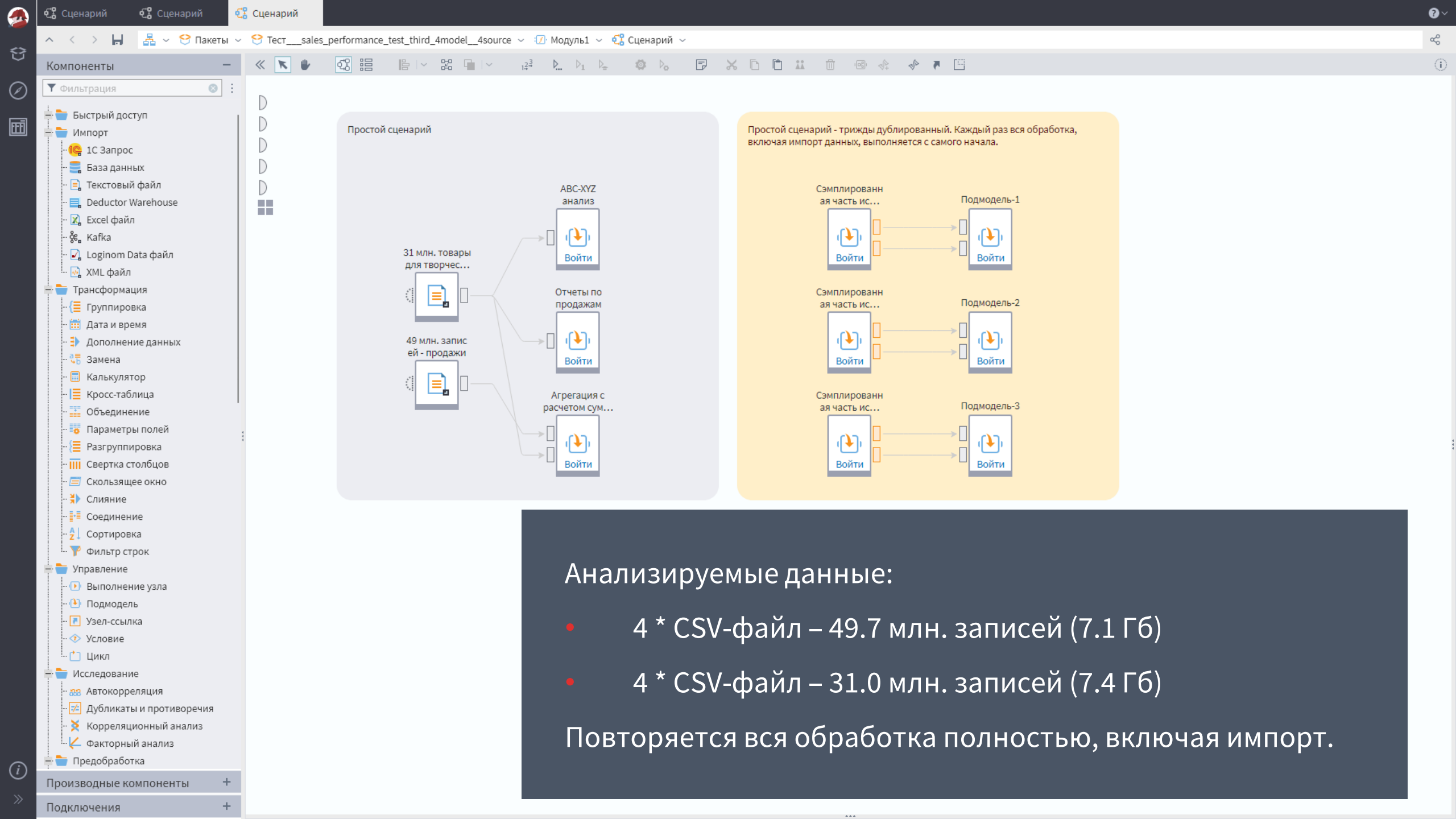
Продукт	Время (сек.)	RAM (Мб)	Загрузка CPU (%)
Loginom	117	13 700	85
Pentaho	197	1 800	35
RapidMiner*	1 495	8 500	30
Alteryx	Более 2-х часов или не прогрузилось		
PolyAnalyst	Более 2-х часов или не прогрузилось		
KNIME	Более 2-х часов или не прогрузилось		
Dataiku	Более 2-х часов или не прогрузилось		

\* Данные необходимо загрузить в репозиторий, время загрузки суммировалось

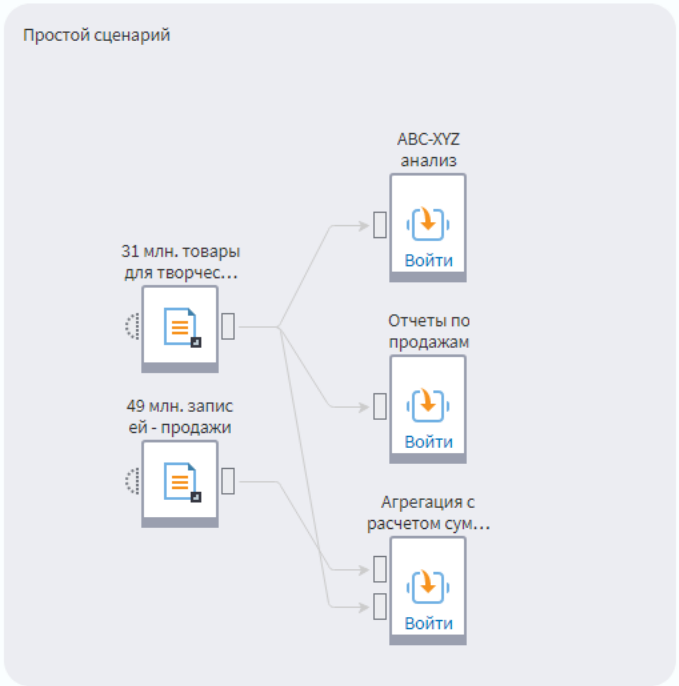
### Задача 3. Высоконагруженная обработка

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.



- 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



Анализируемые данные:

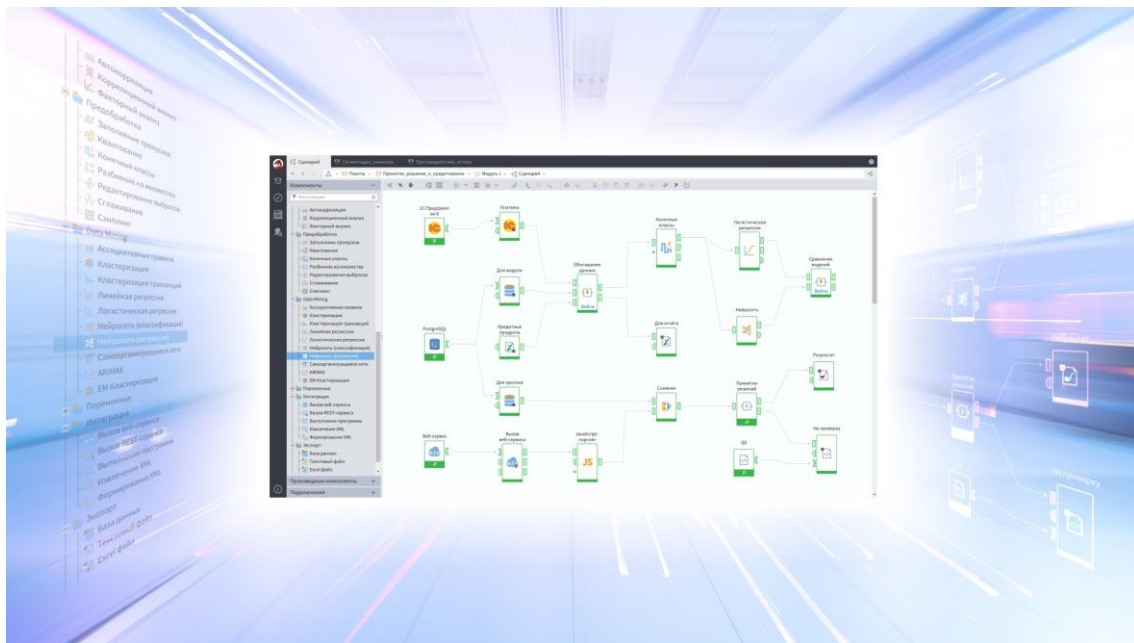
- 4 \* CSV-файл – 49.7 млн. записей (7.1 Гб)
- 4 \* CSV-файл – 31.0 млн. записей (7.4 Гб)

Повторяется вся обработка полностью, включая импорт.

## Сравнение платформ

Продукт	Время (сек.)	RAM (Мб)	Загрузка CPU (%)
Loginom	226	27 000	85
RapidMiner*	5 270	21 000	50
Pentaho	Более 2-х часов или не прогрузилось		
Alteryx	Более 2-х часов или не прогрузилось		
PolyAnalyst	Более 2-х часов или не прогрузилось		
KNIME	Более 2-х часов или не прогрузилось		
Dataiku	Более 2-х часов или не прогрузилось		

\* Данные необходимо загрузить в репозиторий, время загрузки суммировалось



LogiPlot 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.