

# LogiNot: высокая производительность

Алексей Субботин

BaseGroup Labs



# Причины низкой скорости

1. Медленный доступ к данным
2. Неэффективная утилизация «железа»
3. Медленные алгоритмы




Для обеспечения высокой  
производительности  
необходимо расширять все  
«узкие места»

Доступ к данным

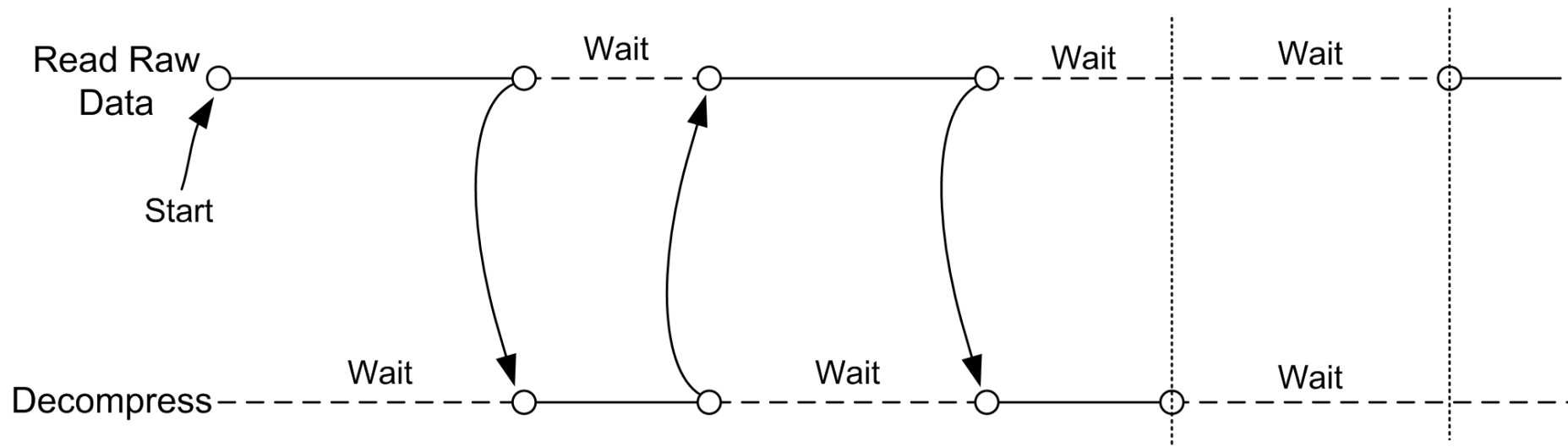


# LGD – внутренний формат

1. Быстрое чтение
  2. Быстрая запись
  3. Экономия места на диске
  4. Хранение метаданных
- 

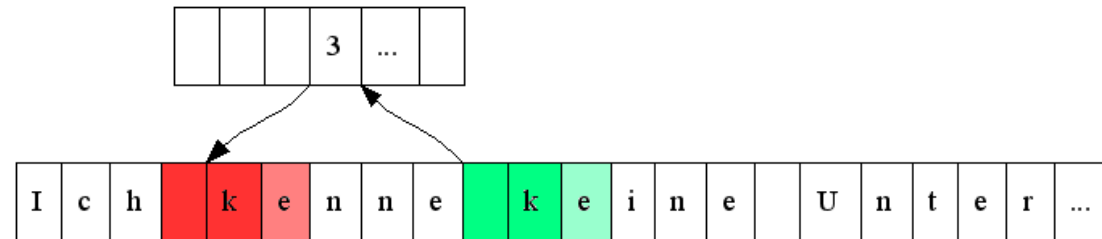
# DDF файл (предыдущая версия)

Узкое место – синхронное чтение/запись:  
канал загружен, процессор простаивает.

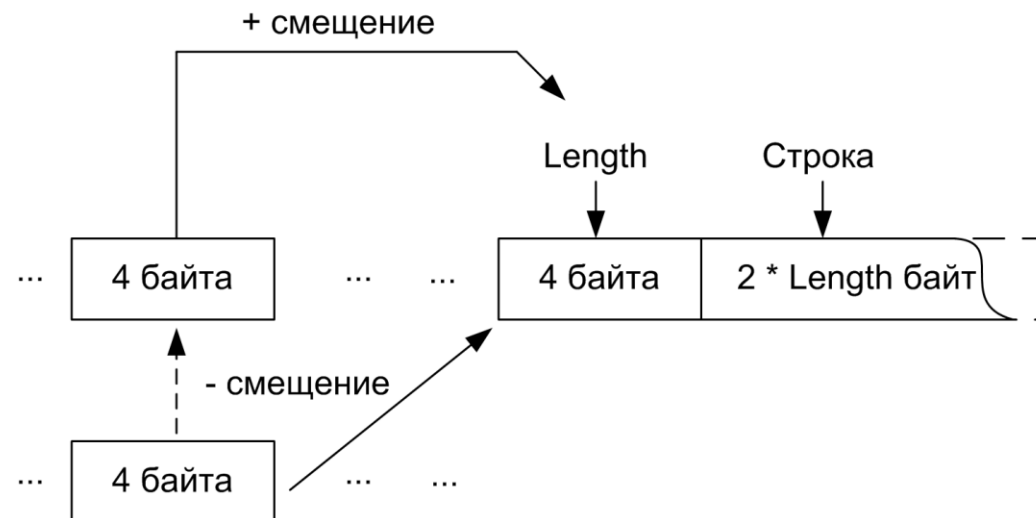


# LGD файл (новая версия)

Потоковое  
сжатие

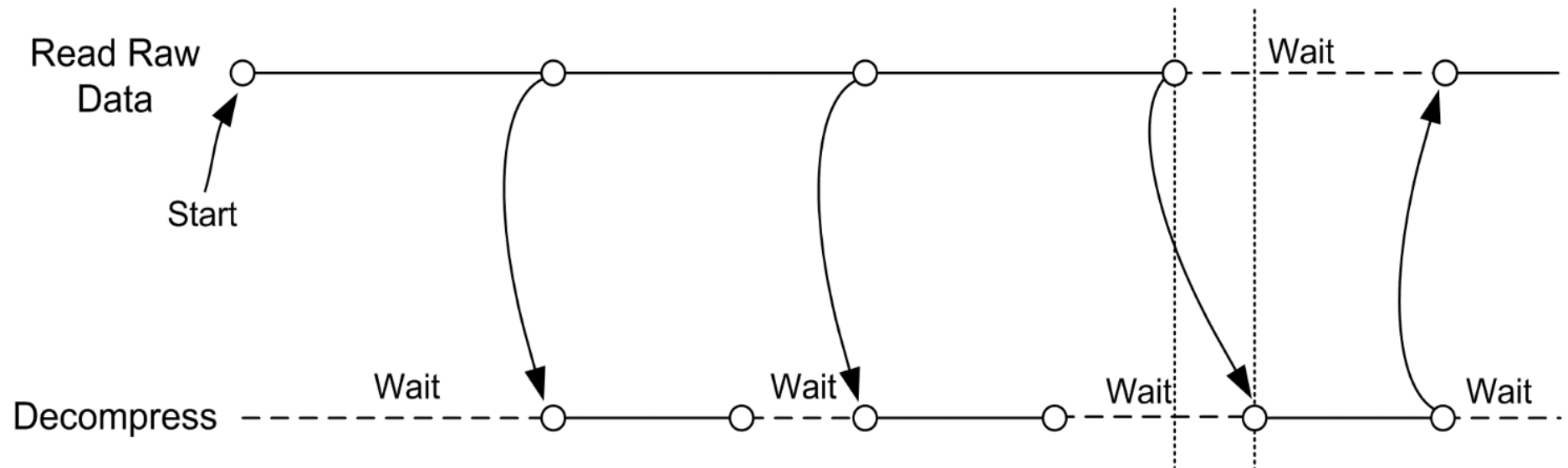
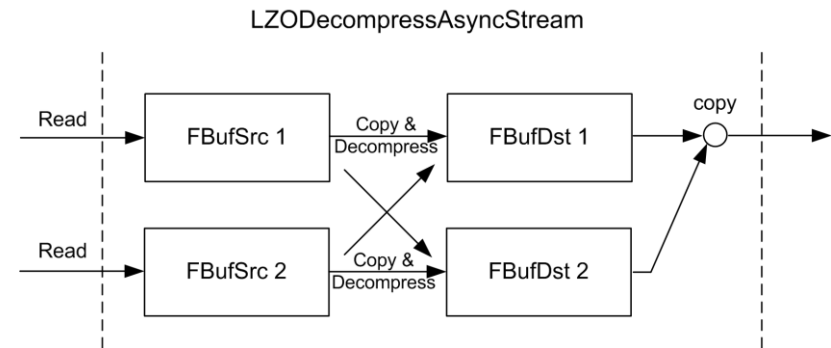


Оптимальное  
хранение строк



# LGD файл

## Асинхронное чтение





# Скорость импорта-экспорта\*

| Время, с**   |         | MySQL<br>LZ4, shared<br>memory | SQLite | CSV | LGD<br>LZ4, 64bit |
|--|---------|--------------------------------|--------|-----|-------------------|
| Вещественные<br>числа, 1 млн<br>записей, 4 столбца | Импорт  | 2,9                            | 0,4    | 0,8 | 0,1               |
|  | Экспорт | 8,5                            | 2,9    | 2,0 | 0,6               |
| Случайные строки,<br>1 млн записей, 4<br>столбца   | Импорт  | 5,9                            | 1,6    | 1,6 | 0,7               |
|  | Экспорт | 35,6                           | 25,3   | 0,8 | 2,5               |

\* Intel Pentium CPU G3240 @3.0GHz, 8GiB RAM, Windows 7 64bit

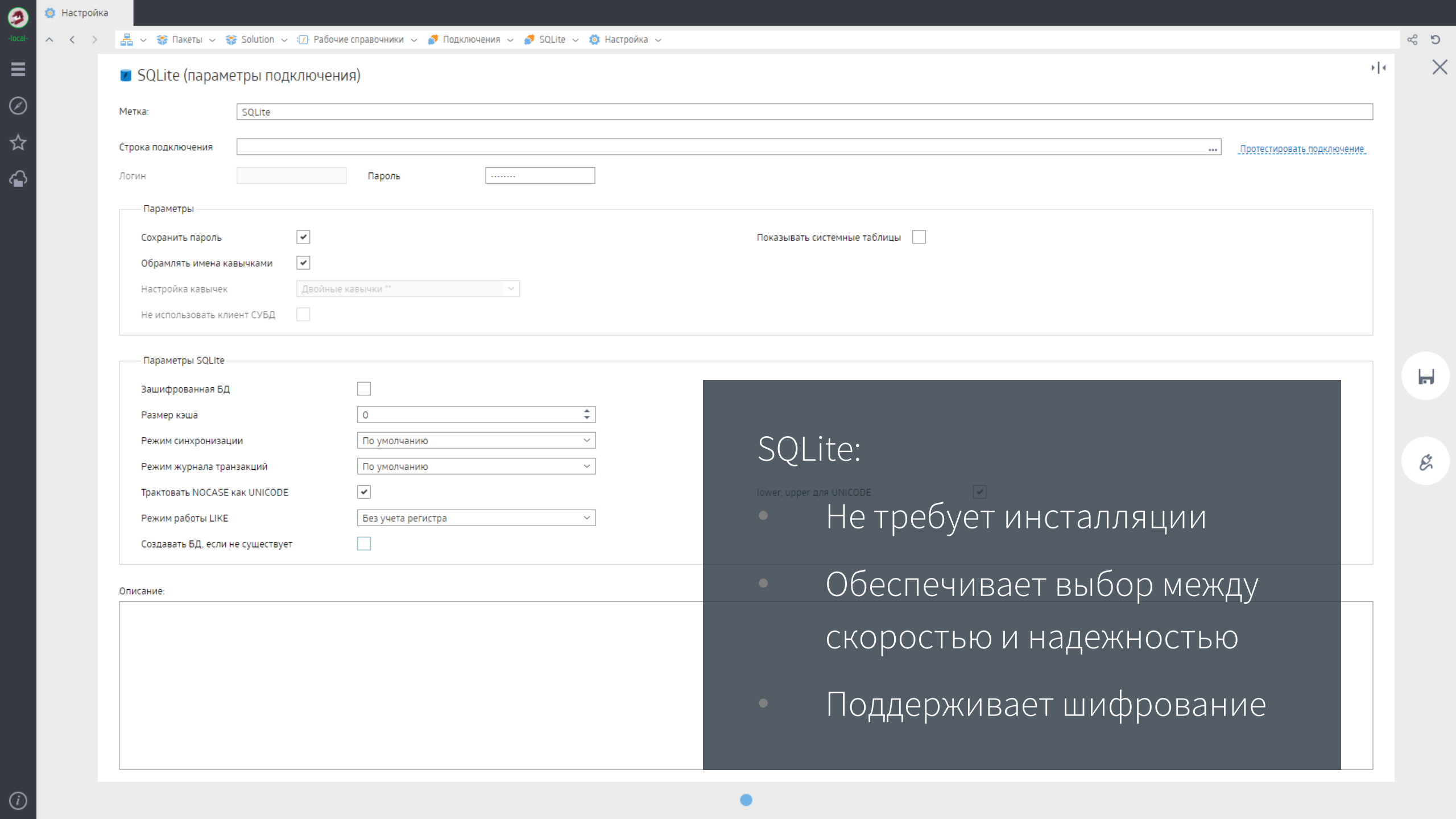
\*\* Лучший результат на 10 прогонах

| База данных | Batch read/write              | Shared memory                    |
|-------------|-------------------------------|----------------------------------|
| Firebird    | ●                             | ●<br>для режима embedded         |
| Interbase   | ●                             | ●<br>для режима embedded         |
| MS SQL      | ●                             |                                  |
| MySQL       | ●                             | ●<br>при работе на одном сервере |
| ODBC        | ●<br>при поддержке клиента БД |                                  |
| Oracle      | ●                             |                                  |
| PostgreSQL  | ●                             |                                  |
| SQLite      |                               | ●                                |



Встроенный SQLite:

- Высокая скорость
- Удобные операции со множествами
- Промежуточное хранилище данных



## SQLite (параметры подключения)

Метка:

Строка подключения  [Протестировать подключение](#)

Логин  Пароль

### Параметры

- Сохранить пароль
- Обрамлять имена кавычками
- Настройка кавычек
- Не использовать клиент СУБД
- Показывать системные таблицы

### Параметры SQLite


- Зашифрованная БД
- Размер кэша
- Режим синхронизации
- Режим журнала транзакций
- Трактовать NOCASE как UNICODE
- Режим работы LIKE
- Создавать БД, если не существует

Описание:

## SQLite:

lower, upper для UNICODE

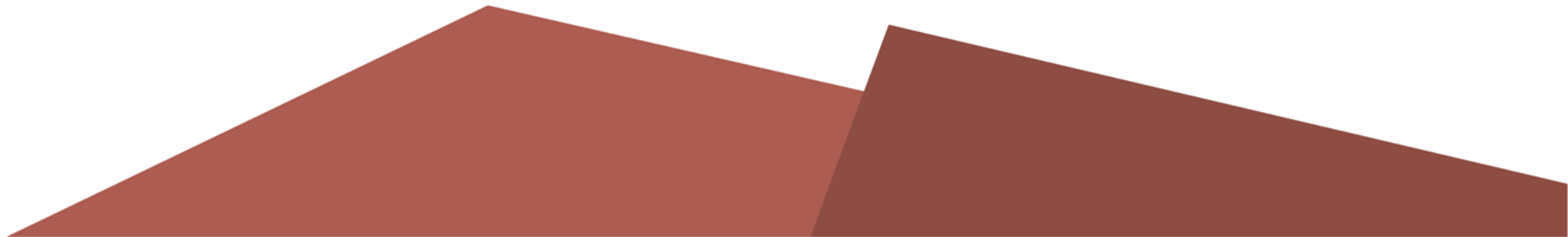
- Не требует инсталляции
- Обеспечивает выбор между скоростью и надежностью
- Поддерживает шифрование

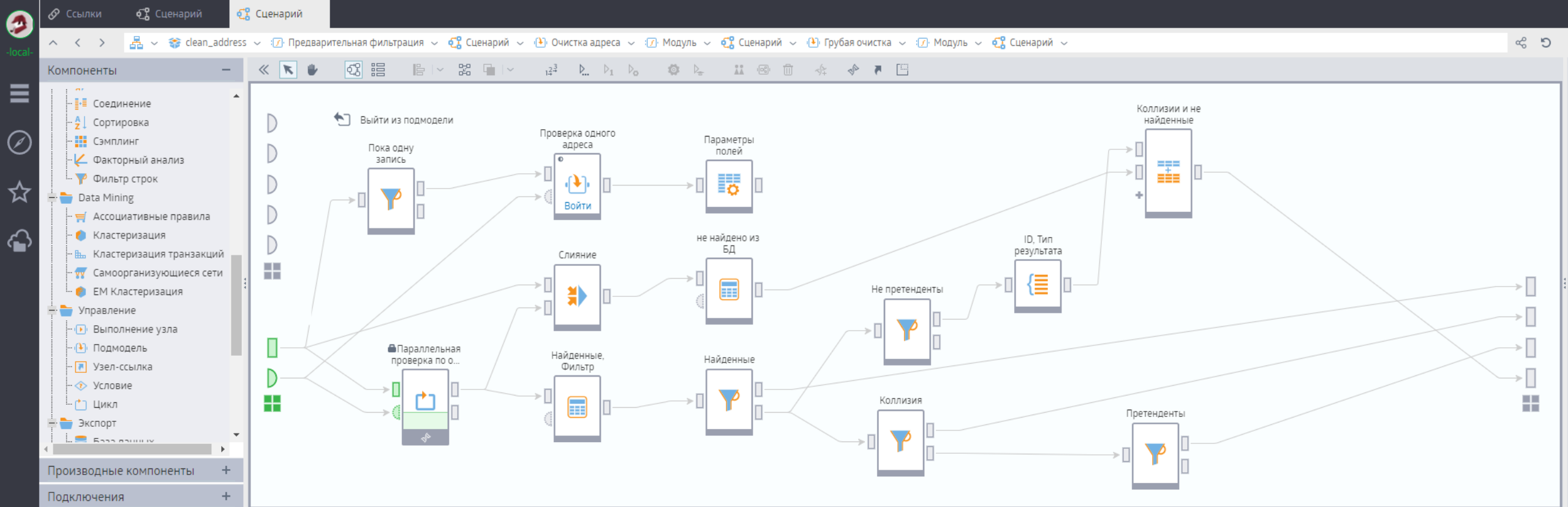


# Быстрый XML

1. Поточный SAX  
парсер
2. Оптимизированное  
хранение XSD
3. Режим упрощенной  
валидации

# Утилизация ресурсов



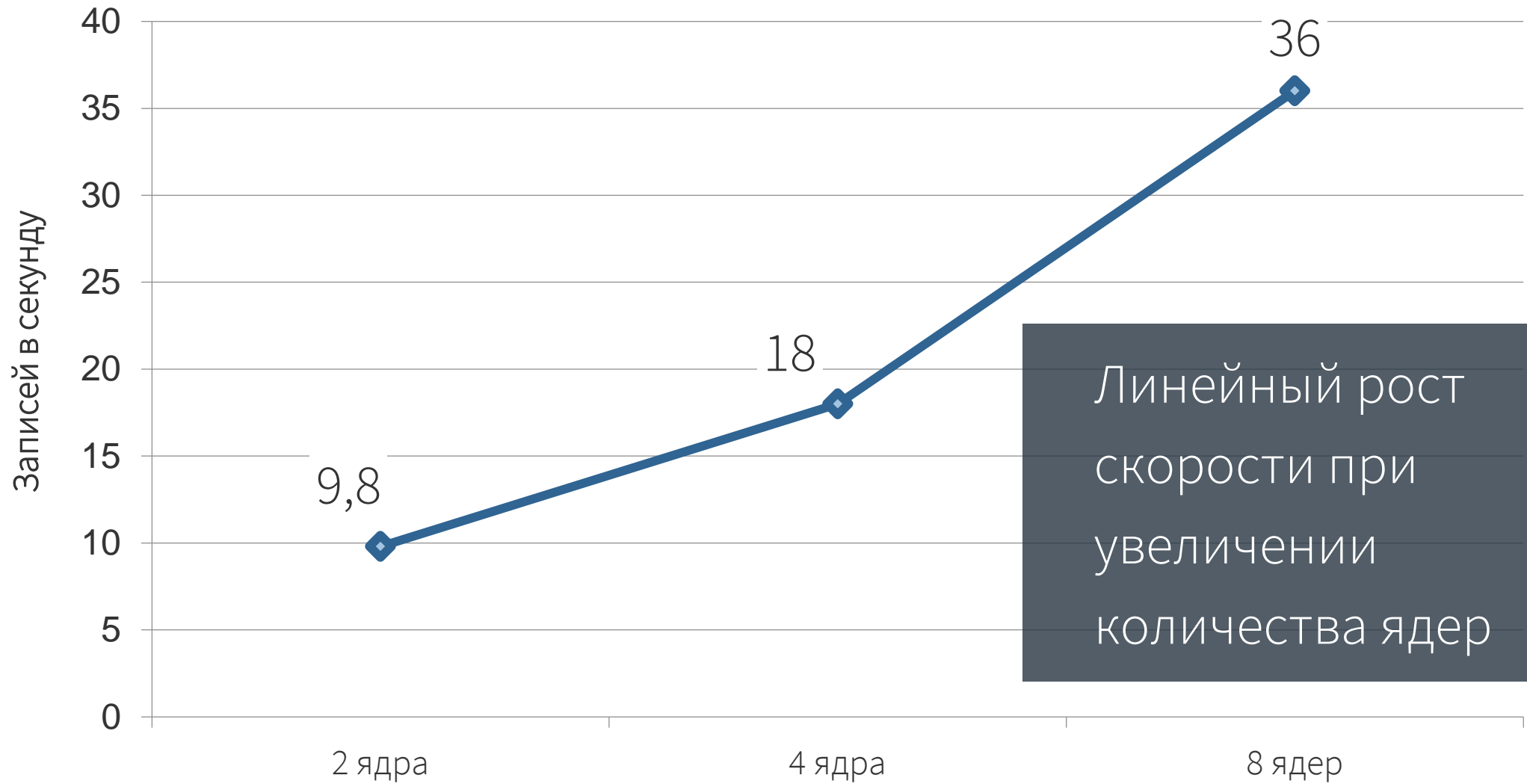


Процессы Ошибки

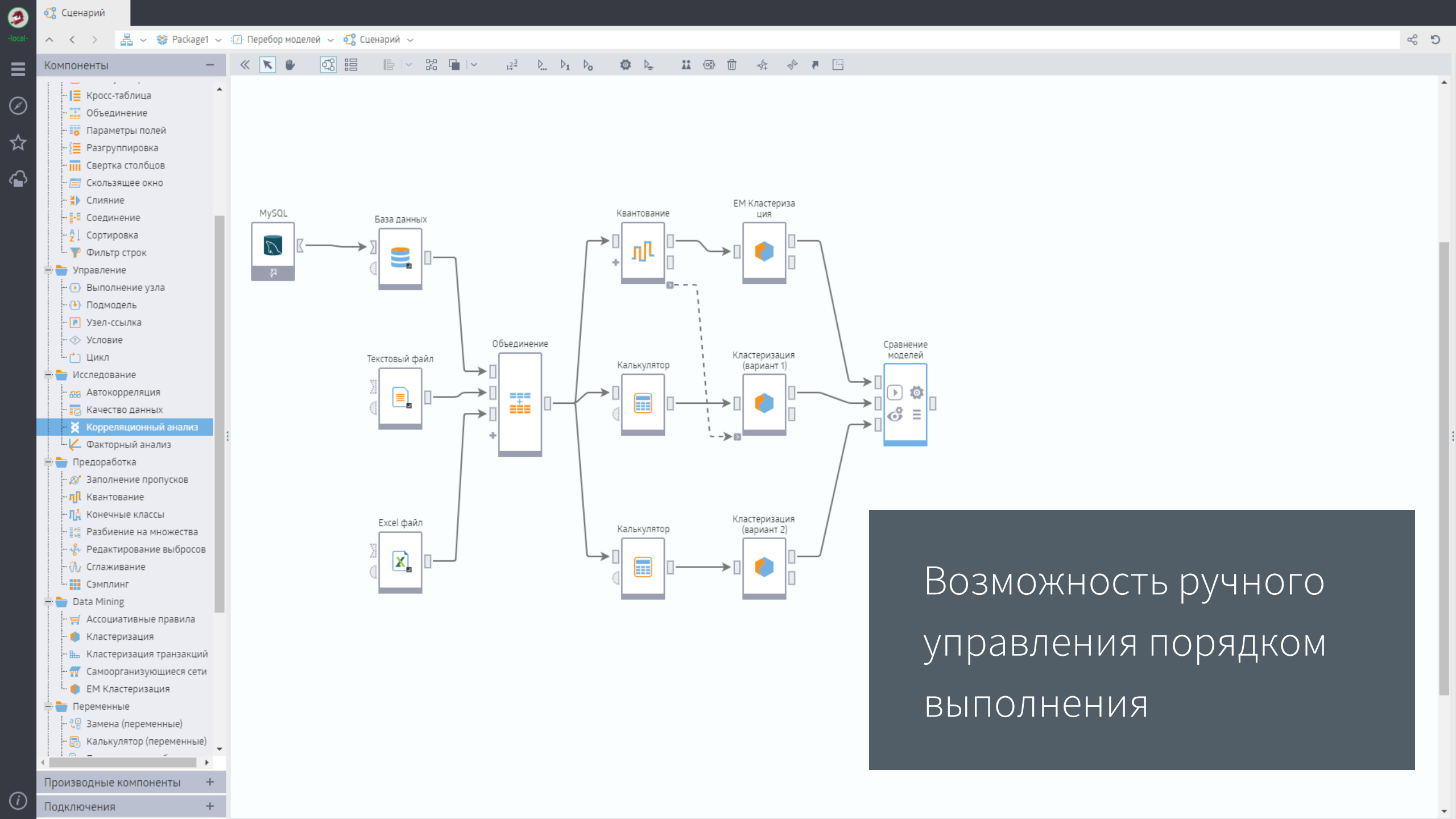
| №      | Процесс                     | %   | Обработка | Ошибка | Начало               | Конец                | Время обработки |
|--------|-----------------------------|-----|-----------|--------|----------------------|----------------------|-----------------|
| 2      | Активация узлов             | 40  |           |        | 25.10.2017, 11:29:39 |                      | 05с 684мс       |
| 2.1    | Параллельная проверка по... | 41  |           |        | 25.10.2017, 11:29:39 |                      | 05с 762мс       |
| 2.1.17 | Итерация №16                | 42  |           |        | 25.10.2017, 11:29:42 |                      | 00с 134мс       |
| 2.1.18 | Итерация №17                | 23  |           |        | 25.10.2017, 11:29:42 |                      | 00с 031мс       |
| 2.1.19 | Итерация №18                | 0   |           |        |                      |                      |                 |
| 2.1.20 | Итерация №19                | 42  |           |        | 25.10.2017, 11:29:42 |                      | 00с 302мс       |
| 2.1.21 | Итерация №20                | 38  |           |        | 25.10.2017, 11:29:43 |                      | 00с 177мс       |
| 2.1.22 | Итерация №21                | 100 |           |        | 25.10.2017, 11:29:43 | 25.10.2017, 11:29:43 | 00с 391мс       |
| 2.1.23 | Итерация №22                | 100 |           |        | 25.10.2017, 11:29:43 | 25.10.2017, 11:29:43 | 00с 200мс       |
| 2.1.24 | Итерация №23                | 38  |           |        | 25.10.2017, 11:29:43 |                      | 00с 126мс       |
| 2.1.25 | Итерация №24                | 0   |           |        | 25.10.2017, 11:29:43 |                      | 00с 000мс       |
| 2.1.26 | Итерация №25                | 0   |           |        |                      |                      |                 |
| 2.1.27 | Итерация №26                | 0   |           |        |                      |                      |                 |
| 2.1.28 | Итерация №27                | 41  |           |        | 25.10.2017, 11:29:44 |                      | 00с 102мс       |
| 2.1.29 | Итерация №28                | 38  |           |        | 25.10.2017, 11:29:44 |                      | 00с 083мс       |
| 2.1.30 | Итерация №29                | 23  |           |        | 25.10.2017, 11:29:44 |                      | 00с 014мс       |

Автоматическое  
распараллеливание  
выполнения сценария

# Пример: Очистка почтовых адресов



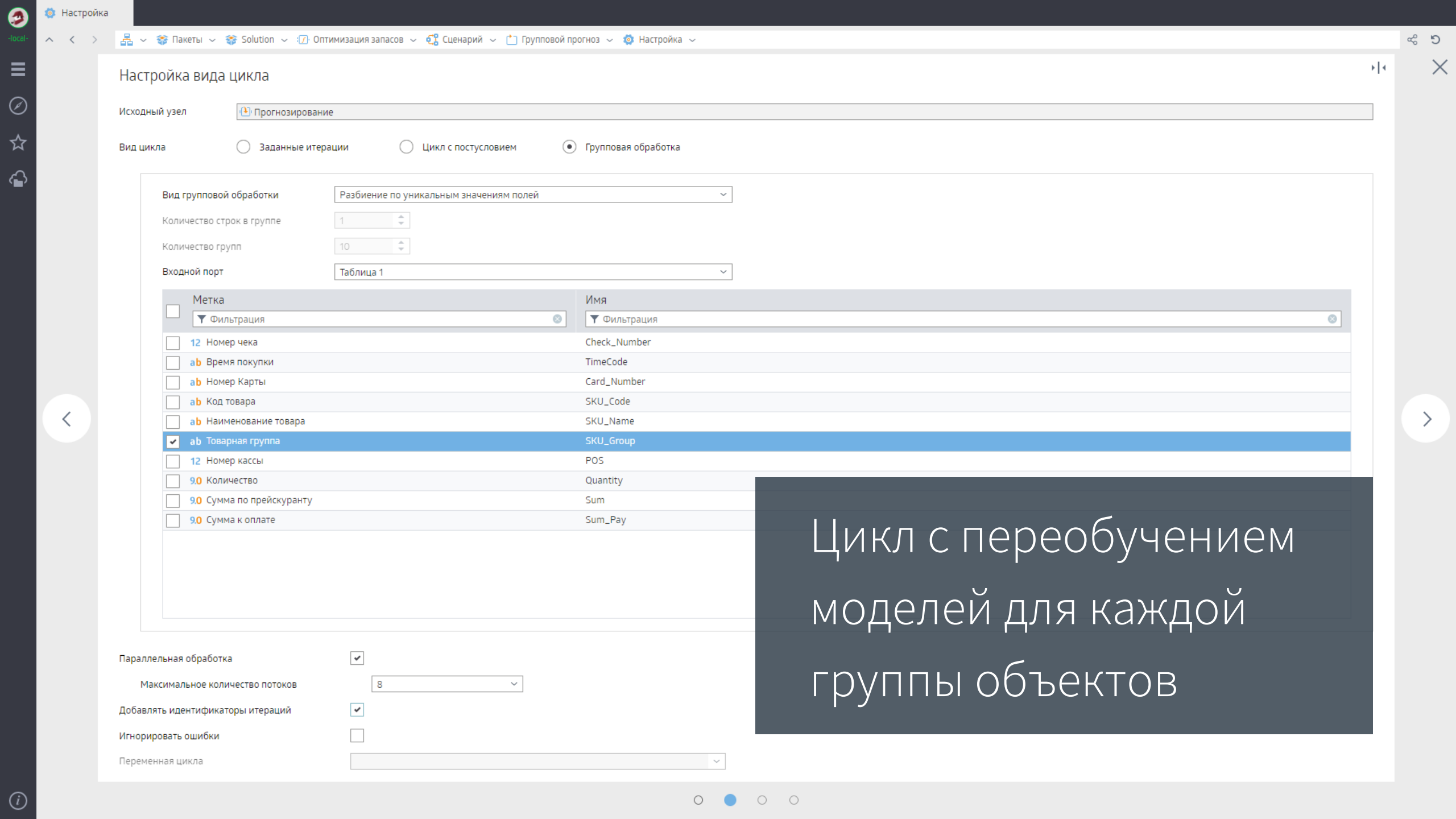




- Кросс-таблица
- Объединение
- Параметры полей
- Разгруппировка
- Свертка столбцов
- Скользящее окно
- Слияние
- Соединение
- Сортировка
- Фильтр строк
- Управление
  - Выполнение узла
  - Подмодель
  - Узел-ссылка
  - Условие
  - Цикл
- Исследование
  - Автокорреляция
  - Качество данных
  - Корреляционный анализ**
  - Факторный анализ
- Предобработка
  - Заполнение пропусков
  - Квантование
  - Конечные классы
  - Разбиение на множества
  - Редактирование выбросов
  - Сглаживание
  - Сэмплинг
- Data Mining
  - Ассоциативные правила
  - Кластеризация
  - Кластеризация транзакций
  - Самоорганизующиеся сети
  - EM Кластеризация
- Переменные
  - Замена (переменные)
  - Калькулятор (переменные)

Производные компоненты +

Подключения +



## Настройка вида цикла

Исходный узел:

Вид цикла:  Заданные итерации  Цикл с постусловием  Групповая обработка

Вид групповой обработки:

Количество строк в группе:

Количество групп:

Входной порт:

| Метка  | Имя          |
|--|--------------|
| <input type="checkbox"/> Фильтрация                    | Фильтрация   |
| <input type="checkbox"/> 12 Номер чека                 | Check_Number |
| <input type="checkbox"/> ab Время покупки              | TimeCode     |
| <input type="checkbox"/> ab Номер Карты                | Card_Number  |
| <input type="checkbox"/> ab Код товара                 | SKU_Code     |
| <input type="checkbox"/> ab Наименование товара        | SKU_Name     |
| <input checked="" type="checkbox"/> ab Товарная группа | SKU_Group    |
| <input type="checkbox"/> 12 Номер кассы                | POS          |
| <input type="checkbox"/> 9.0 Количество                | Quantity     |
| <input type="checkbox"/> 9.0 Сумма по прейскуранту     | Sum          |
| <input type="checkbox"/> 9.0 Сумма к оплате            | Sum_Pay      |

Параллельная обработка:

Максимальное количество потоков:

Добавлять идентификаторы итераций:

Игнорировать ошибки:

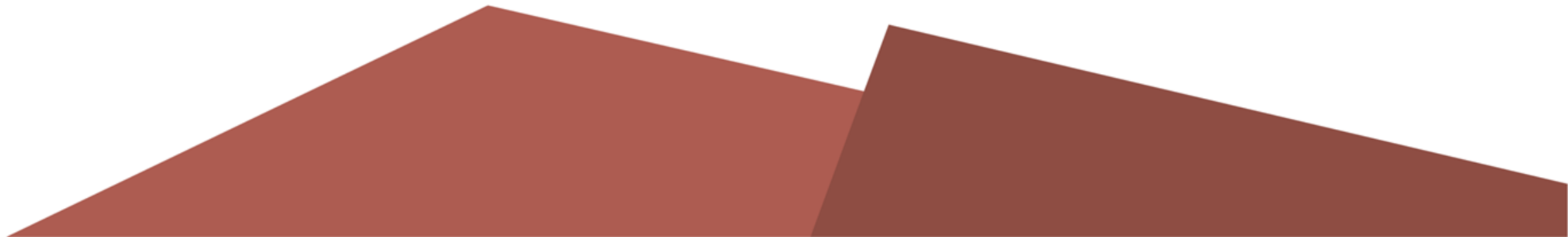
Переменная цикла:

Цикл с переобучением  
моделей для каждой  
группы объектов

# Работа с памятью

1. Стараемся все данные держать в памяти
2. По умолчанию храним только уникальные данные
3. При необходимости можно задать точки кэширования данных
4. Оптимизировано хранение наборов
5. Кэшируем последние загруженные пакеты

# Быстрые алгоритмы



# Формулы/код в калькуляторе:

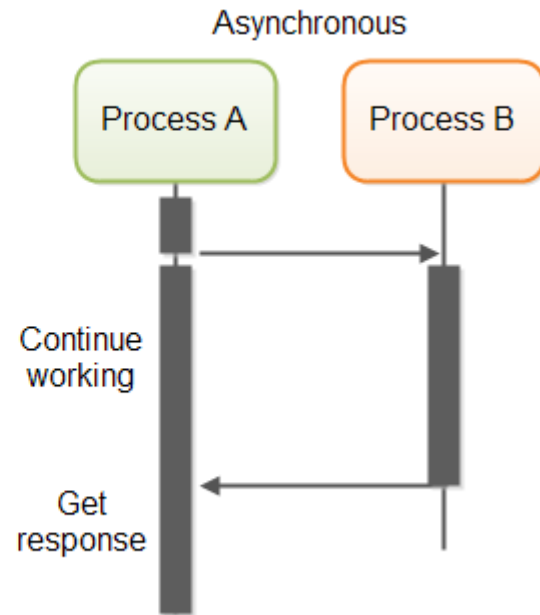
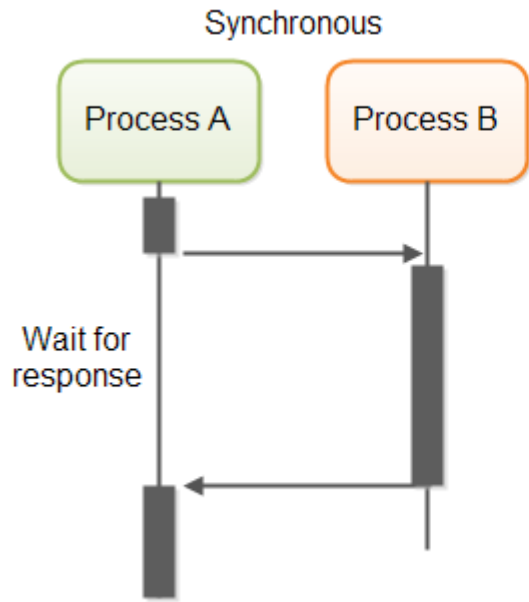
1. Построение и кэширование синтаксического дерева выражения
2. JIT компиляция и кэширование регулярных выражений
3. JIT компиляция и кэширование JavaScript кода

# Оптимизация алгоритмов:

- Применение самых быстрых библиотек
- Хранение данных в специальных структурах с учетом особенностей каждого алгоритма
- Обработка данных окнами
- Реализация алгоритмов для оптимальной работы в многоядерных системах

# Работа с памятью:

1. По возможности чтение и сохранение в память крупными блоками
2. Использование структур данных, вмещающихся в кэш процессора



Ощущение комфорта  
даже при  
выполнении долгих  
операций сервером



[basegroup.ru](http://basegroup.ru)

