

**Описание функциональных характеристик  
ПО “Tengri Data Platform”**

**Москва**

**2025**

## **Аннотация**

Настоящий документ содержит описание функциональных характеристик ПО “Tengri Data Platform”, в том числе описание ПО, информацию о назначении ПО, описание основных возможностей ПО, а также задачи, реализуемые при помощи платформы.

## СОДЕРЖАНИЕ

Описание и назначение ПО .....	4
Основные функции, которые выполняет ПО .....	5
Разделение Compute & Storage .....	6
Хранение .....	7
Apache Iceberg в S3 .....	7
Колоночный формат хранения .....	7
Вычисление .....	8
Вычислители .....	8
Ускорение вычислений .....	10
Morsel-driven parallelism .....	10
SIMD-инструкции .....	10
Диалект и протоколы PostgreSQL .....	11
PostgreSQL .....	11
Загрузка данных .....	12
Развитая система разграничения прав .....	13
Поддерживаемые принципы .....	13
DAC .....	13
RBAC .....	13
Ключевые концепты системы разграничения прав .....	13
Защищаемый объект .....	13
Роль .....	13
Привилегия .....	13
Пользователь .....	14
Контакты .....	15

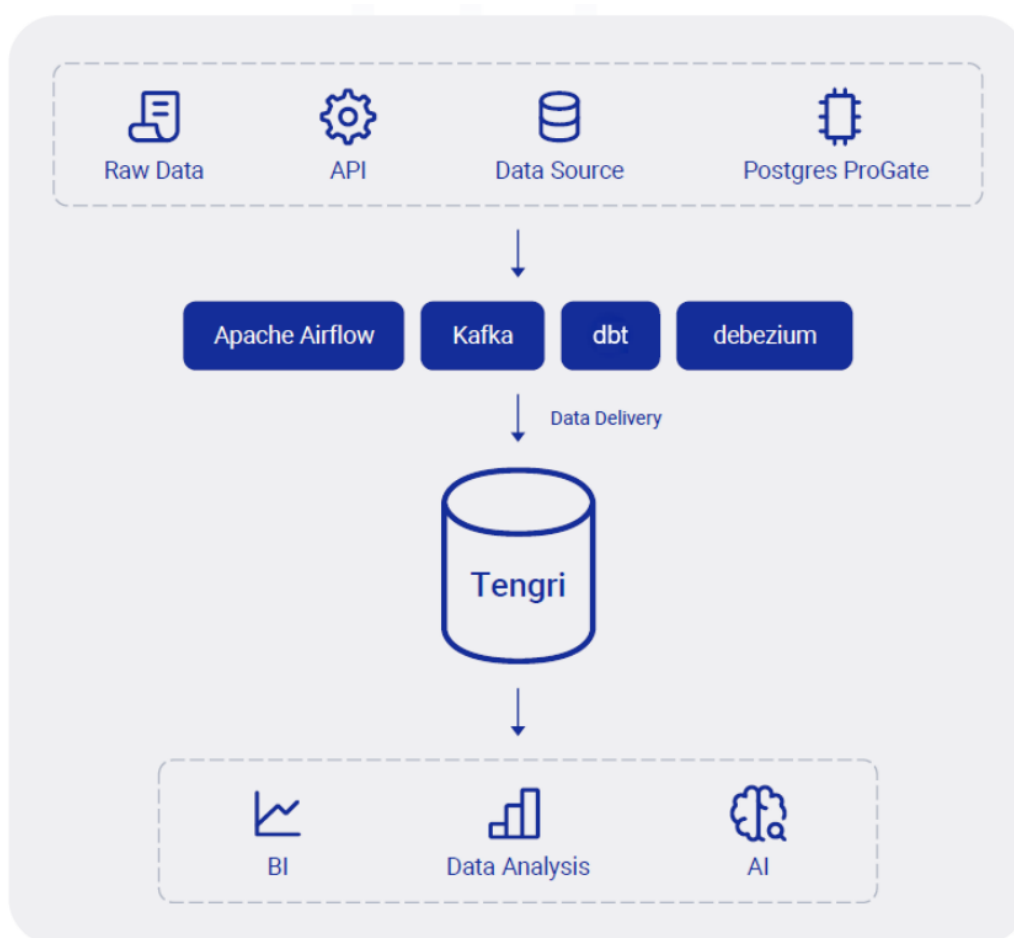
## Описание и назначение ПО

ПО “Tengri Data Platform” — это корпоративная аналитическая платформа для работы с данными.

ПО поддерживает любые операции с данными на языках SQL и Python в единой среде: загрузка данных, трансформация данных, оркестрация расчетов, визуализация, машинное обучение и AI. Все операции с данными покрываются системой прав и разрешений, все действия логируются. Операции над данными максимально эффективны и изолированы «из коробки» за счёт использования современных технологий. Ресурсы вычисления и хранения данных независимо эластично масштабируются. On-premise, в облаках или гибридно. Данные хранятся в формате Iceberg, доступном для стороннего использования через Iceberg REST API; файлы Iceberg лежат в S3. Вычисление выполняется на эластичном кластере под управлением Tengri, кластер может быть представлен физическими или виртуальными машинами или K8S.

## Основные функции, которые выполняет ПО

1. хранение данных;
2. получение данных из множества разнообразных источников (API, CRM, ERP, JSON, биллинги, другие источники);
3. объединение данных в единую платформу;
4. анализ и агрегация данных;
5. создание отчетов и графиков;
6. вычисление, ускорение вычислений.



В Tengri реализованы:

- ✓ Поддержка OLAP SQL-запросов
- ✓ Простая интеграция с ETL/BI/ML-системами
- ✓ Неограниченное масштабирование 1
- ✓ Гарантии целостности и сохранности данных
- ✓ Система разграничения прав доступа

## Разделение Compute & Storage

Tengri — аналитическая платформа нового поколения, основанная на парадигме разделения систем хранения и вычисления (Compute & Storage).

- Данные хранятся в универсальном масштабируемом кластере S3.
- Вычислительные мощности подбираются динамически при помощи изолированных вычислителей, запускаемых на платформе контейнеризации.



Такой подход позволяет:

- Гибко и независимо масштабировать хранение и вычислительные ресурсы без кратного увеличения расходов на инфраструктуру;
- Адаптивно встраиваться в современные гибридные инсталляции предприятий.

Описание систем вычисления и хранения представлено в разделах Вычисление и Хранение соответственно.

## **Хранение**

### **Apache Iceberg в S3**

Данные таблиц в Tengri хранятся в кластере S3 в открытом формате Apache Iceberg.

- Iceberg поддерживает операции INSERT, UPDATE и DELETE, а также изменение схемы таблиц.
- Iceberg поддерживает ACID — набор требований к транзакционной системе, обеспечивающий наиболее надёжную и предсказуемую её работу.
- Iceberg-структура формируется на основе формата Parquet с поколоночной компрессией данных.
- Iceberg — это тысячи успешных внедрений крупнейшими мировыми компаниями для работы с сотнями ТБ и десятками ПБ данных.

### **Колоночный формат хранения**

Благодаря использованию формата Parquet в Tengri поддерживаются: Колоночный формат хранения данных OLAP-запросы работают быстрее за счет чтения только нужных колонок таблиц. Поколоночная компрессия данных OLAP-запросы работают быстрее в 5 и более раз благодаря сжатию данных таблиц.

## Вычисление

В Tengri вычисление производится в виртуальных вычислительных пулах — вычислителях.

## Вычислители

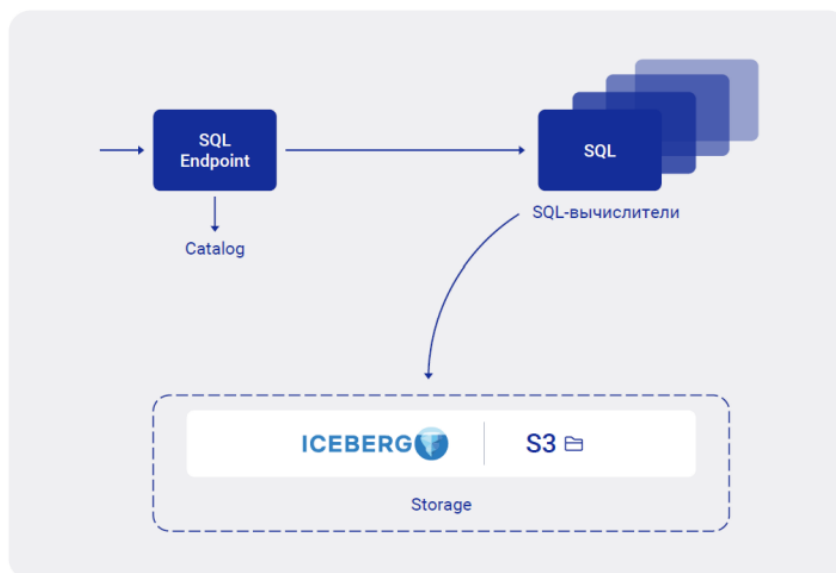
SQL-вычислитель — это легковесный контейнер, запускаемый на аппаратном кластере за ~100 миллисекунд.

SQL-вычислитель:

- Получает эксклюзивные ядра (vCPU) и RAM-бюджет
- Подгружает данные и выполняет SQL-запросы
- Отключается системой после некоторого периода ожидания (если простаивает без нагрузки)

За счет такой архитектуры вычислителей:

- Каждый аналитик работает со своим SQL-вычислителем
- Аналитики не мешают друг другу

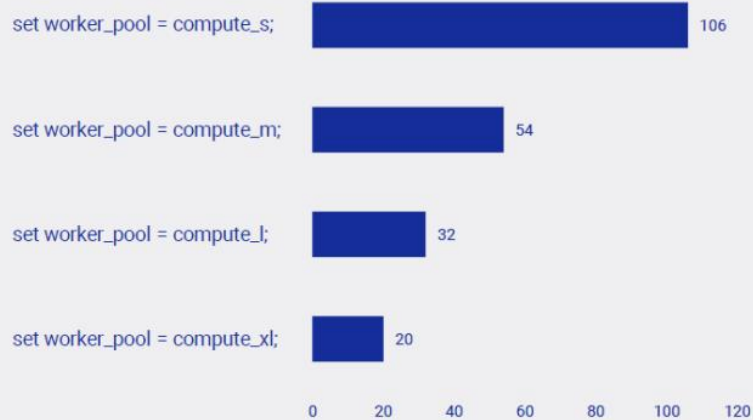


Каждый аналитик в каждой сессии может выбрать размер вычислителя для SQL-запроса: XS / S / M / L / XL. Каждый следующий размер вычислителя в два раза больше предыдущего.

Время выполнения теста TPC-H на вычислителях разного размера:



### Время выполнения (в секундах), TPC-H, sf=100



*Код, использованный для выполнения этого теста*

```
SELECT profit.nation AS nation,  
profit.o_year AS o_year,  
SUM(profit.amount) AS sum_profit  
  
FROM (SELECT s100_nation.n_name AS nation,  
  
EXTRACT(YEAR FROM s100_orders.o_orderdate) AS o_year,  
  
s100_lineitem.l_extendedprice * (1 - s100_lineitem.l_discount) -  
s100_partsupp.ps_supplycost * s100_lineitem.l_quantity AS amount  
  
FROM s100_part, s100_supplier, s100_lineitem,  
s100_partsupp, s100_orders, s100_nation  
  
WHERE s100_supplier.s_suppkey = s100_lineitem.l_suppkey  
AND s100_partsupp.ps_suppkey = s100_lineitem.l_suppkey  
AND s100_partsupp.ps_partkey = s100_lineitem.l_partkey  
AND s100_part.p_partkey = s100_lineitem.l_partkey  
AND s100_orders.o_orderkey = s100_lineitem.l_orderkey  
AND s100_supplier.s_nationkey = s100_nation.n_nationkey  
AND s100_part.p_name LIKE '%' || 'black' || '%')  
AS profit  
  
GROUP BY profit.nation, profit.o_year ORDER BY nation, o_year  
DESC NULLS FIRST LIMIT 1000;
```

В Tengri также поддерживаются вычислители Python.

## **Ускорение вычислений**

Для ускорения вычислений в Tengri поддерживаются:

### **Morsel-driven parallelism**

OLAP SQL-запросы работают быстрее за счет использования Morsel-алгоритмов, позволяющих загружать все vCPU, выделенные системой.

### **SIMD-инструкции**

OLAP SQL-запросы работают быстрее за счет использования SIMD-инструкций современных CPU.

- Тесты производительности Tengri приведены в разделе Результаты тестов.
- Подробности о работе с вычислителями можно узнать в разделе Вычислительные пулы.

## Диалект и протоколы PostgreSQL

### PostgreSQL

В Tengri реализована поддержка SQL (диалект PostgreSQL).

В частности поддерживаются:

- Основные элементы синтаксиса SQL диалекта PostgreSQL: SELECT, INSERT, UPDATE, DELETE, JOIN, WITH, UNION
- Оконные функции
- Работа с JSON посредством SQL
- Принципы ACID
- Транзакции
- Транзакционная целостность данных
- Изоляция данных на уровне Repeatable read (Повторяющееся чтение)

JDBC-драйвер PostgreSQL.

## **Загрузка данных**

Гарантируется работоспособность инструментов загрузки данных (Airflow, dbt) и инструментов аналитики (BI, ML) за счет возможности подключаться через JDBC-драйвер PostgreSQL.

Все BI- и ETL-инструменты работают так, как если бы это был PostgreSQL.

Аналитикам не нужно переучиваться на новые инструменты.

## **Развитая система разграничения прав**

В Tengri реализована развитая система разграничения прав доступа.

### **Поддерживаемые принципы**

В Tengri поддержаны принципы DAC и RBAC.

#### **DAC**

**Избирательное управление доступом:** у каждого объекта есть владелец, который может предоставить права доступа (привилегии) на данный объект другим пользователям.

#### **RBAC**

**Управление доступом на основе ролей:** права доступа (привилегии) приписаны ролям, которые, в свою очередь, приписаны пользователям

## **Ключевые концепты системы разграничения прав**

### **Защищаемый объект**

Сущность, к которой может быть предоставлен доступ (привилегии). Если доступ к данному защищаемому объекту не разрешен, он будет запрещен.

### **Роль**

Сущность, которой могут быть предоставлены права доступа (привилегии). Роли могут назначаться пользователям или другим ролям. Назначение роли другой роли создает иерархию ролей.

### **Привилегия**

Определенный уровень доступа к объекту. Назначаются пользователям или ролям. Привилегии, назначенные ролям или пользователям, позволяют получить доступ к объектам, подлежащим защите. Могут быть отозваны у ролей или пользователей.

Для управления детализацией предоставляемого доступа может одновременно использоваться несколько различных привилегий.

## **Пользователь**

Идентификатор, связанный с человеком или службой. Пользователь является объектом, которому могут быть предоставлены привилегии.

Подробности о работе с системой разграничения прав можно узнать в разделах:

- [Операции с пользователями](#)
- [Операции с ролями](#)
- [Операции с привилегиями](#)

## Контакты

E-mail: [sales@postgrespro.ru](mailto:sales@postgrespro.ru)