

www.picodata.io

Picodata: обзор возможностей, кейсы, план развития

Константин Осипов, основатель и СТО

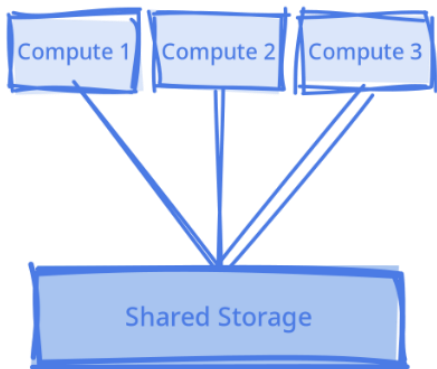
Вебинар Arenadata
Апрель 2026



Picodata: PostgreSQL-совместимая (in-memory) СУБД

Shared-Storage

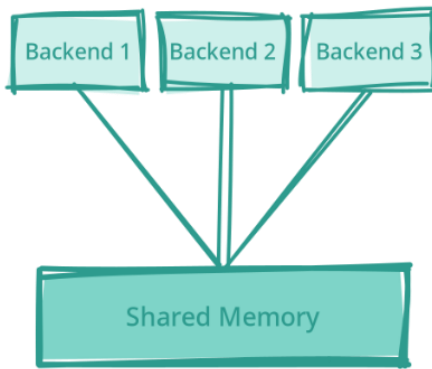
Neon, Aurora



compute \square storage
over network

Shared-Memory

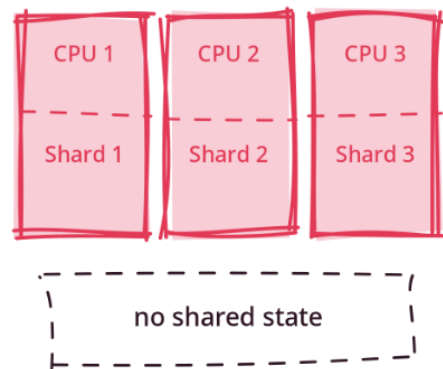
PostgreSQL



backends contend on
shared memory

Shared-Nothing

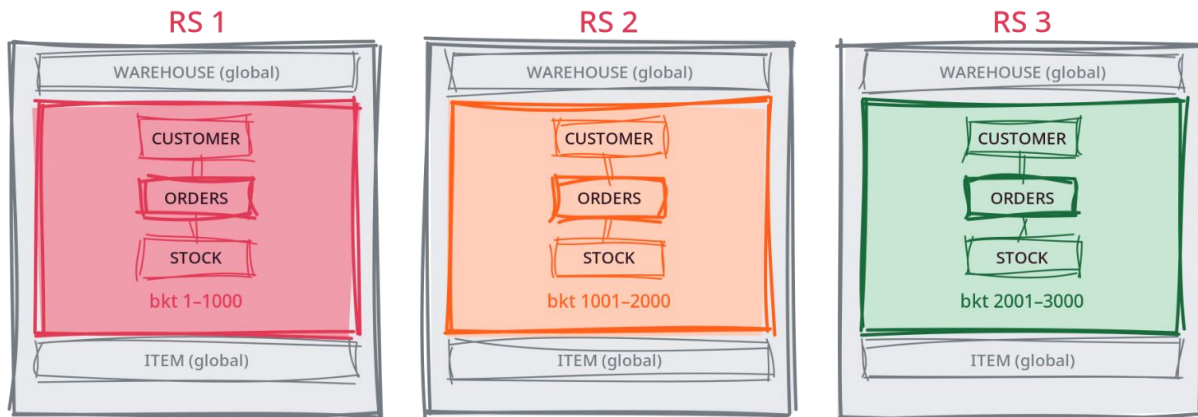
Picodata



shard-per-core:
no locks, no contention

Shared-nothing: каждое ядро CPU владеет своим шардом данных — без блокировок, без contention, без разделяемого состояния

Распределение данных

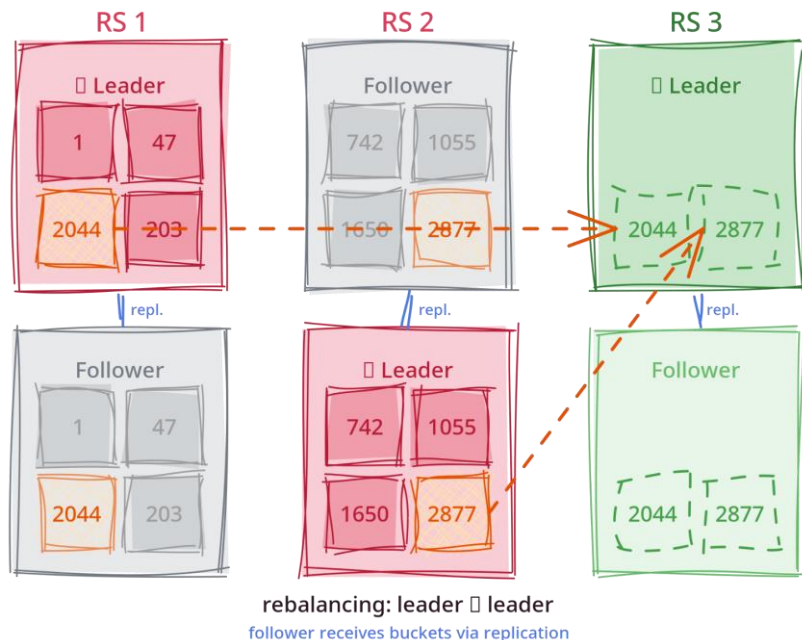


- co-distributed tables O JOIN locally, zero network hops
- global tables I dimension data available on every node

```
CREATE TABLE orders (...) DISTRIBUTED BY (o_w_id)
CREATE TABLE customer (...) DISTRIBUTED BY (c_w_id)
CREATE TABLE warehouse (...) DISTRIBUTED GLOBALLY
```

- Таблицы с общим ключом распределения — co-distributed: JOIN'ы выполняются локально
- Глобальные таблицы реплицируются на каждый узел — идеально для справочников

Распределение данных



- Данные разбиты на 3 000 бакетов (по умолчанию)
- Таблицы с общим ключом распределения — co-distributed: JOIN'ы выполняются локально
- Глобальные таблицы (справочники) реплицируются на каждый узел
- Smart-драйверы кэшируют карту bucket→shard и маршрутизируют запросы напрямую

- 3 000 buckets by default - fixed-size units of rebalancing
- rebalancing begins when new RS is fully online (RF satisfied)
- drivers cache bucketshard map O queries routed directly

PostgreSQL-совместимый SQL и экосистема

Экосистема PostgreSQL —
работает без изменений



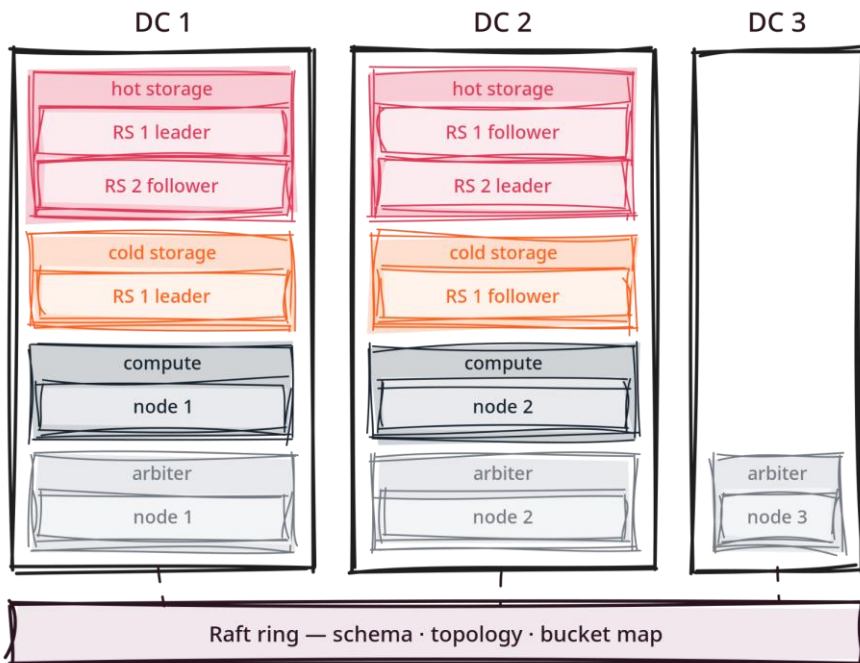
Нативные драйверы Picodata — *shard-aware*,
topology-aware



- Кэшируют карту bucket→shard
- Маршрутизируют запросы напрямую на нужный узел
- Автоматически обновляют топологию

- JOIN'ы, CTE, оконные функции, подзапросы, агрегации
- Система типов: INTEGER, TEXT, BOOLEAN, DOUBLE, DECIMAL, UUID, DATETIME, JSON
- DISTRIBUTED BY / DISTRIBUTED GLOBALLY для управления размещением данных

Доступность и масштабирование



- Синхронная запись на диск — durable persistence
- Синхронная репликация — полная копия БД в 2+ ДЦ
- ACID с уровнем изоляции SERIALIZABLE
- Тирры: горячее хранение (RAM), холодное (Vinyl LSM), compute, arbiter
- Production: 10 000+ TPS/ядро, 2–100 ТБ, до 2000+ узлов

Безопасность сверх требований ФСТЭК



Шифрование

Полное шифрование
внутри кластера:

PostgreSQL трафик,
HTTP (метрики и WebUI)



Аутентификация

SCRAM-SHA-256
(взаимная авторизация)

LDAP + TLS, PostgreSQL
протокол + TLS



Аудит

Полный аудит
избранных ролей

До 128 ролей

Picodata Radix

Плагин Radix — реализация Redis-протокола на базе СУБД Picodata, предназначенная для миграции с существующих инсталляций Redis



- Поддержка типов данных и 180+ команд Redis
- Автоматическое масштабирование (Redis Cluster)
- Политика вытеснения ключей (volatile-ttl в Redis)
- До 90% производительности Redis на одном узле
- Линейный рост производительности при масштабировании
- Логирование событий

Picodata Sirin

Плагин Sirin — реализация Cassandra-протокола на базе СУБД Picodata, предназначенная для миграции с существующих инсталляций Casandra



- Поддержка всех основных типов и команд
- отсутствие REPAIR
- **снижение ТСО и экономия оборудования**
- безопасная альтернатива для IoT и промышленности

Picodata Silver

Автоматизация процессов распределения доходов и затрат,
направленный на повышение прибыльности бизнеса и управление
его маржинальностью

Замена для технологий:

SAP FSPER
IBM COGNOS

Скорость расчета выше

в 5 раз

Затраты на оборудование ниже

в 2 раза

Ключевые бизнес пользователи:



Финансовый директор



Директор по закупкам



Главный бухгалтер



Директор по производству

Кейс: аналитика в реальном времени

1

запрос ФОИВ

7

SQL-запросов

~3 000

строк распределённого SQL

190

JOIN'ов

81

CTE

42

оконных функции

38

UNION

Все запросы выполняются конкурентно на всех узлах кластера

Каждый из 7 запросов: 340–500 строк SQL, 11–13 CTE, 23–32 JOIN'ов, 3–10 оконных функций

Технологический запрос

Витрины исторических выписок для ФОИВ —
предоставление выписок и ответов на запросы от федеральных органов исполнительной власти

СПЕЦИФИКА ЗАПРАШИВАЕМЫХ ДАННЫХ:

- Каждый запрос ФОИВ порождает **5–7** запросов к БД
- Сложная структура поиска, выборка из таблиц с 300 млн — **1 млрд записей**

ЦЕЛЕВЫЕ ПОКАЗАТЕЛИ:

- Среднее: **20 запросов/с**, пиковое: **300 запросов/с**
- Целевой объём: **1 200 000 запросов/сутки**



Решение

Выделение слоя «горячих данных» на in-memory СУБД Picodata

Достигнутые результаты

Сложность запросов	RPS
Простые (JOIN < 5)	29 230
Средние (5 < JOIN < 10)	11 430
Сложные (10 < JOIN < 20)	6 340

Исходная производительность (без Picodata)

400 бизнес-запросов/с

Достигнутая производительность (с Picodata)

3 000 бизнес-запросов/с

7.5x

Аналитические запросы

3x

Импорт данных через Spark

Кейс: low-code витрина данных

Возможности:

- Импортёры данных

быстрая настройка интеграции с системами в ландшафте предприятия; форматы parquet, csv, flat file, Oracle GG, Kafka, Dataflot

- Витрины данных

предоставляют необходимый срез данных через RESTful API

- Фильтры и триггеры

гибкая реакция на данные, поступающие из источников

- конфигурирование через yaml

конфигурация управляется системой контроля версий

- UI для работы с конфигурацией,
- Интеграция с SSO/ OpenId Connect

Решаемые задачи:

- Виртуализация данных.
- Кеш оперативных данных .
- Высокопроизводительное хранилище мастер-данных.

Кейс: тарифный калькулятор

- 01 Гибкая настройка продуктов и тарифов без программирования
- 02 Быстрое формирование индивидуальных предложений
- 03 Онлайн подключение тарифных планов и пакетов услуг
- 04 Быстрая интеграция со всеми смежными системами
- 05 Высокая скорость онлайн калькуляции стоимости продукта



Производительность хранения: Vinyl

Превращаем Vinyl в production-ready движок для OLTP/HTAP

Сделано (26.1–26.2)

- Новый планировщик компактификации (shape-based)
- Учёт read amplification при выборе уровня
- Слияние мелких и пустых range
- Slab-аллокатор для кортежей vinyl

В разработке

- Snapshot Isolation — настоящий repeatable read без ложных аборт транзакций
- Словарная компрессия (dictionary compression)
- LCP-сжатие ключей в индексных страницах
- Новый page index cache, fuse фильтры

Горизонт

- io_uring + libev extension — асинхронный I/O
- TTL — нативная поддержка устаревших данных
- Глобальный IO-планировщик
- Shared memory interconnect между процессами

SQL и совместимость с PostgreSQL

Снижаем порог перехода с PostgreSQL: оптимизатор, типы данных, транзакции

SQL →

Parse →

Optimize →

Route →

Execute

26.2 Оптимизатор

- LIMIT pushdown в подзапросы
- Fast-path локальная маршрутизация
- Propagation равенств через JOIN
- WAIT APPLIED для ALTER SYSTEM
- Fencing multi-bucket SQL
- Расширенная поддержка JSON
- LET, IF в транзакционных блоках

26.3 PG-совместимость

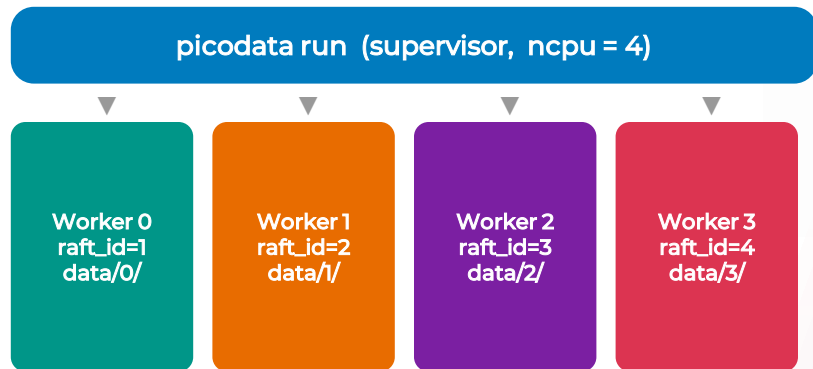
- SERIAL — автоинкрементные колонки
- PostgreSQL COPY — пакетная загрузка
- Logical dump/restore (SQL формат)
- VIEWS (представления)

26.4+ Распределённый SQL

- Accord — распределённые транзакции
- Новый SQL парсер (PEST)
- Кросс-тип SQL запросы
- Foreign data wrappers
- Cost-based optimizer

Масштабирование и эксплуатация

От одного ядра к десяткам — масштабирование одной строкой конфига



:3301 (SO_REUSEPORT) • IPC: Unix socketpairs

Кластер и репликация

- Синхронная репликация шардированных спейсов (26.2 flagship)
- Новая реализация шардирования (26.2 flagship)
- Rolling upgrade 25.x → 26.x
- `picodata nuke` — удаление узлов после потери majority (26.3)

Удобство эксплуатации

- Kubernetes health probes — настраиваемые `/health/*` endpoint'ы
- Endpoint `/memory` — мониторинг потребления памяти
- WebUI: управление кластером, фильтрация, состояние replicaset
- `picodata doctor` — диагностика кластера (26.4)
- Просмотр статуса и отмена длительных запросов (26.4)

Экосистема плагинов: Radix и Sirin

Одна платформа — несколько протоколов: Redis, Cassandra, PostgreSQL



Radix (Redis)

- Версия 0.14 → 1.0
- ACL, MULTI/EXEC, Lua-скриптинг
- STRING, LIST, HASH, ZSET, SET
- Redis Sentinel эмуляция
- Впереди: BITMAP, Streams, JSON
- Vector sets, Time series
- Diskless mode, eviction

Sirin (Cassandra)

- Версия 1.1 → 1.2 (апр. 2026)
- Production: Tibbo, Цифра
- CQL: SELECT, INSERT, UPDATE
- Авторизация, prepared stmts
- Впереди: DISTINCT, DESCRIBE
- TTL без доп. индексов
- IPC для псри, YCSB тесты

Замена Redis + Cassandra + PostgreSQL на единую платформу: единое управление, мониторинг, резервное копирование, шардирование из коробки

www.picodata.io



@picodataru
@kostja_osipov

Почему именно Picodata?

Требование	Описание
Высокая скорость	In-memory вычисления, задержки в миллисекунды
Сложные аналитические запросы	Оконные функции, распределённые JOIN, агрегации
Распределённая архитектура	Горизонтальное масштабирование, автошардирование, отказоустойчивость
PostgreSQL диалект SQL	Минимальный порог входа для команды заказчика
Регуляторные требования	Сертификация ФСТЭК, реестр ПО

Функционал	Picodata	Tarantool	YDB
In-memory режим	✓ Да	✓ Да	- Нет
Управление распределением данных	✓ Да	- Нет	- Нет
pgwire протокол	✓ Да	- Нет	o Эксп.
Распределённый SQL	✓ Да	- Нет	✓ Да (YQL)
Оконные функции	✓ Да	- Нет	✓ Да (YQL)
LDAP	✓ Да	o Частично	✓ Да
Реестр ПО / ФСТЭК	✓ Да	✓ Да	✓ Да