PostgresPro

Как горизонтально масштабировать СУБД Postgres Pro

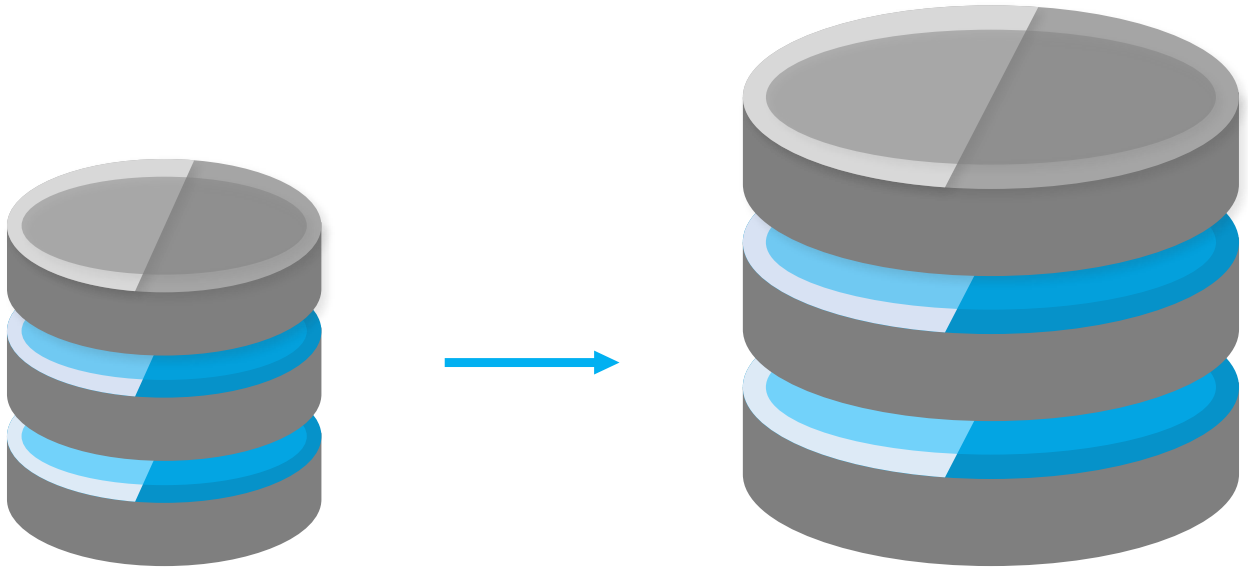
Виктор Васильев

28 января 2025 года

PGProDay

Подходы масштабирования

Вертикальное

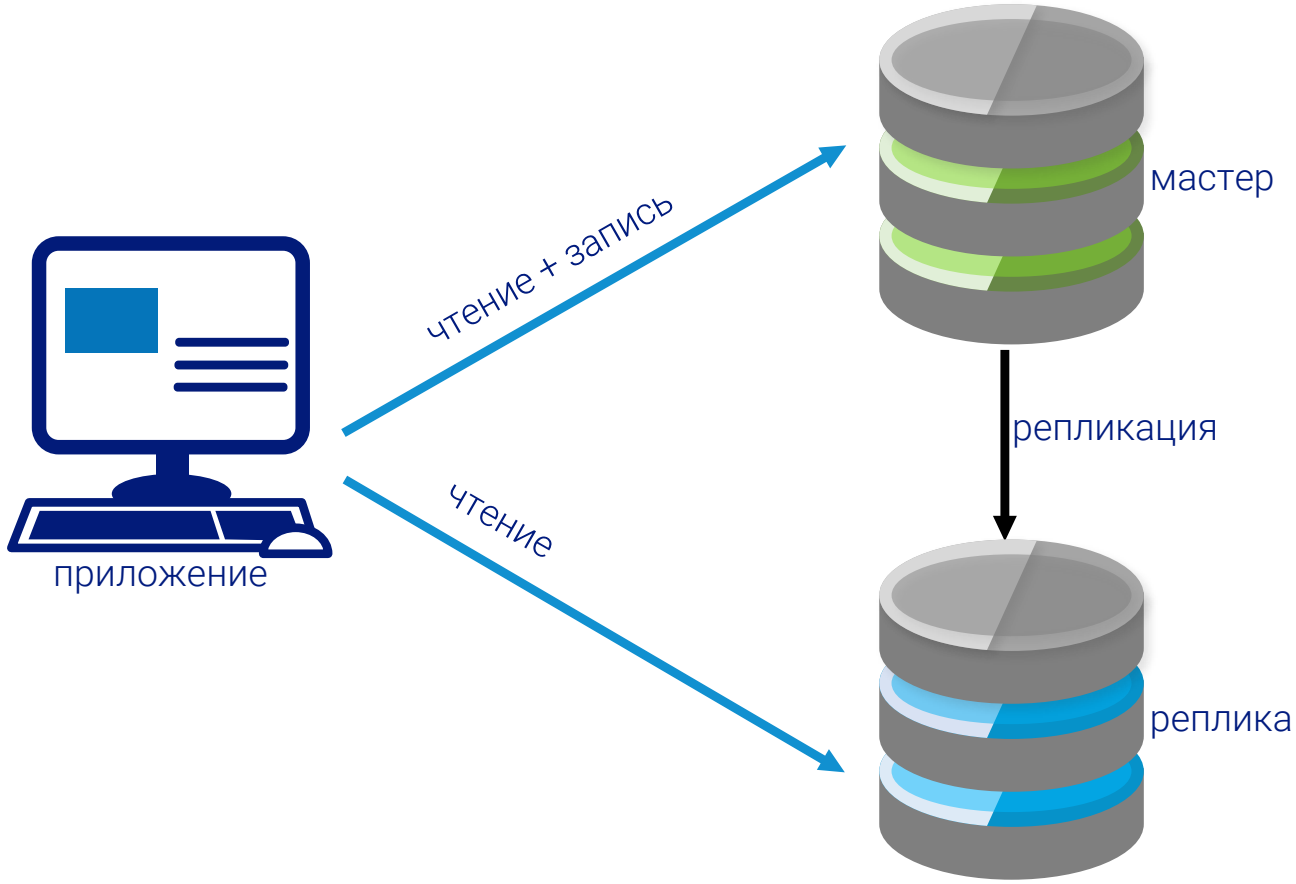


Подходы масштабирования

Горизонтальное

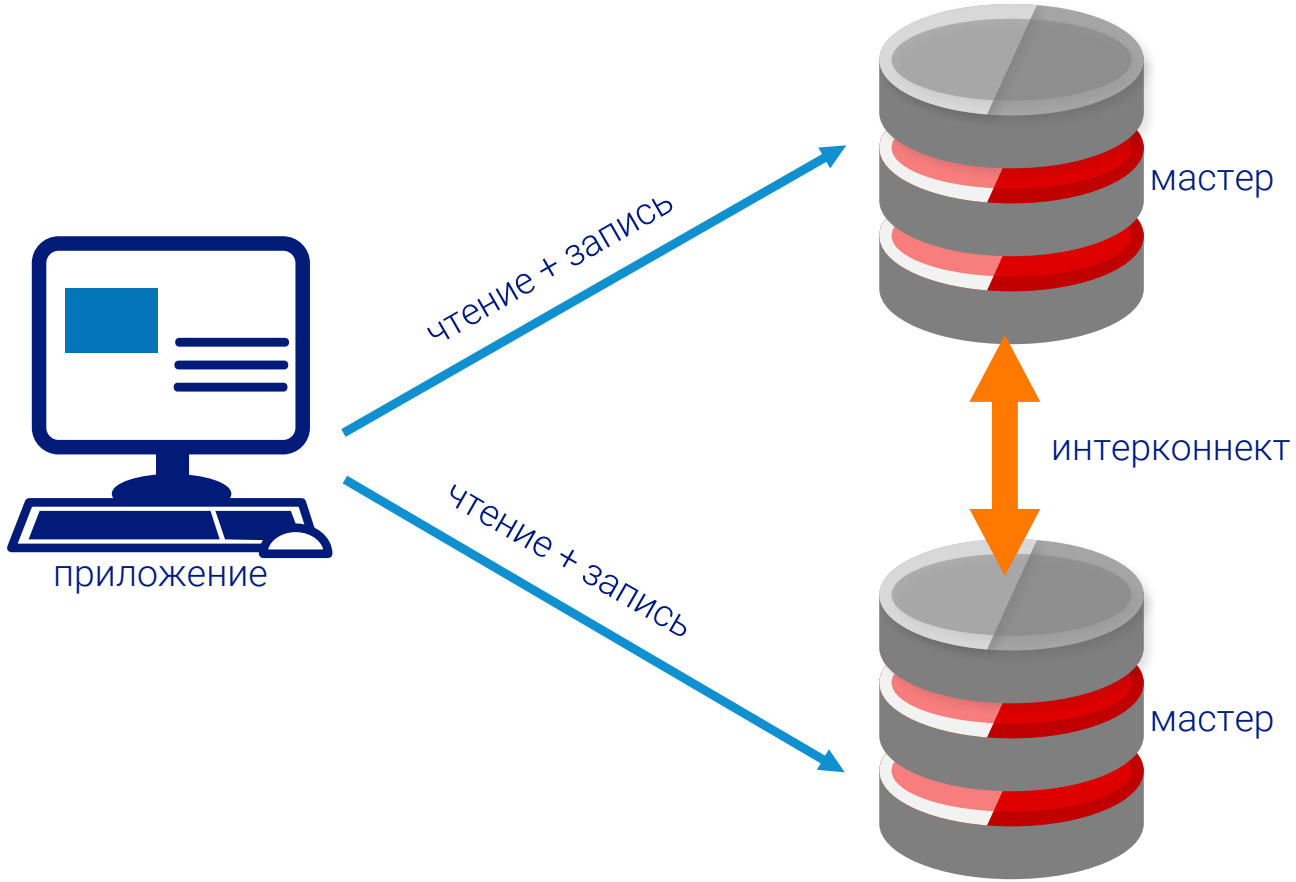
Горизонтальное масштабирование

Репликация



Горизонтальное масштабирование

Шардирование



Когда нужно шардирование?

- Когда уже есть большая и нагруженная СУБД, использующая все доступные ресурсы сервера
- Вертикальное масштабирование экономически или технически невозможно
- Невозможность изменять архитектуру доступа и работы с данными
- Когда планируем строить КХД, предполагающее массивно-параллельную архитектуру
- Встроенное горизонтальное масштабирование
- Мультитенантность

Варианты шардирования в Postgres Pro

- **Shardman**

Патч к ядру на основе Postgres Pro, реализованный по принципам массивно-параллельной архитектуры, обеспечивающий горизонтальное масштабирование

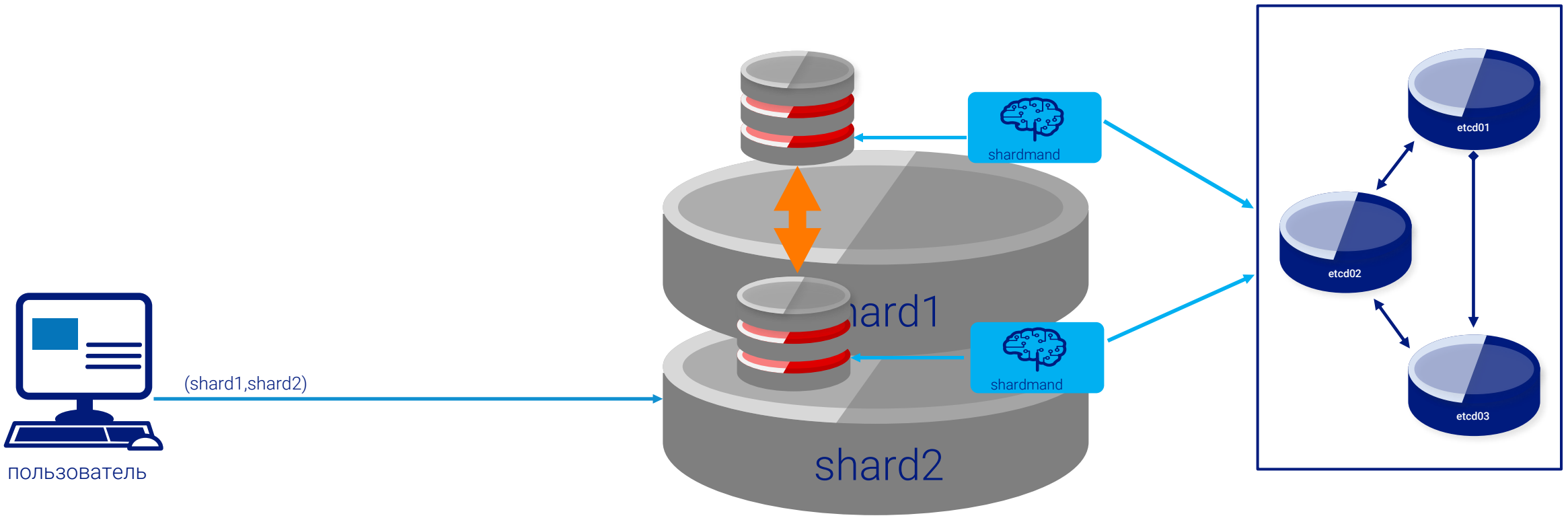
- **Citus**

Расширение для Postgres Pro, предоставляющее горизонтальное масштабирование и колоночное хранение

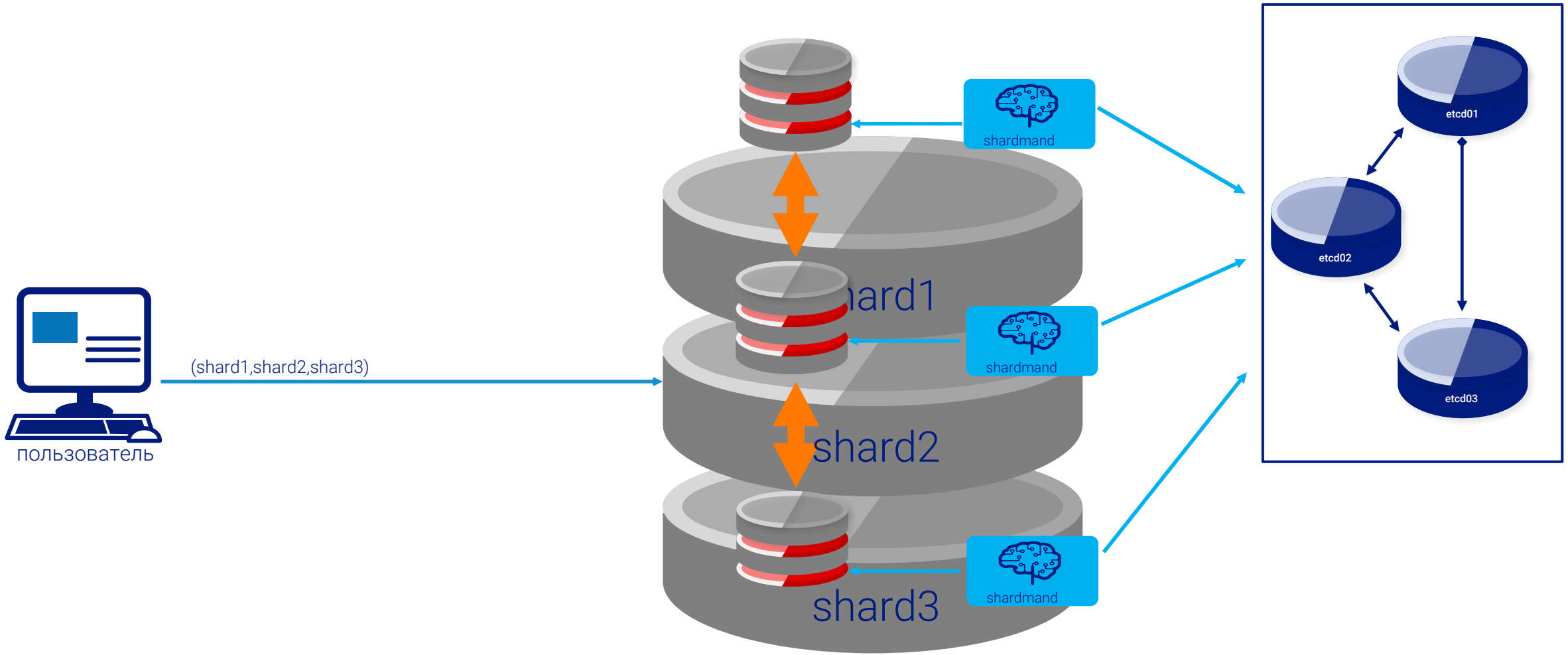
Особенности Shardman и Citus

	Shardman	Citus
Целостность данных (ACID)	На уровне кластера	На уровне узла
Профиль нагрузки	OLTP (ограниченно OLAP)	OLAP (ограниченно OLTP)
Точка входа пользователя	Любой узел	Любой узел
Управляющая точка входа	Любой узел	Координатор
Колоночное хранение	Нет	Да
ФСТЭК	Да	Нет

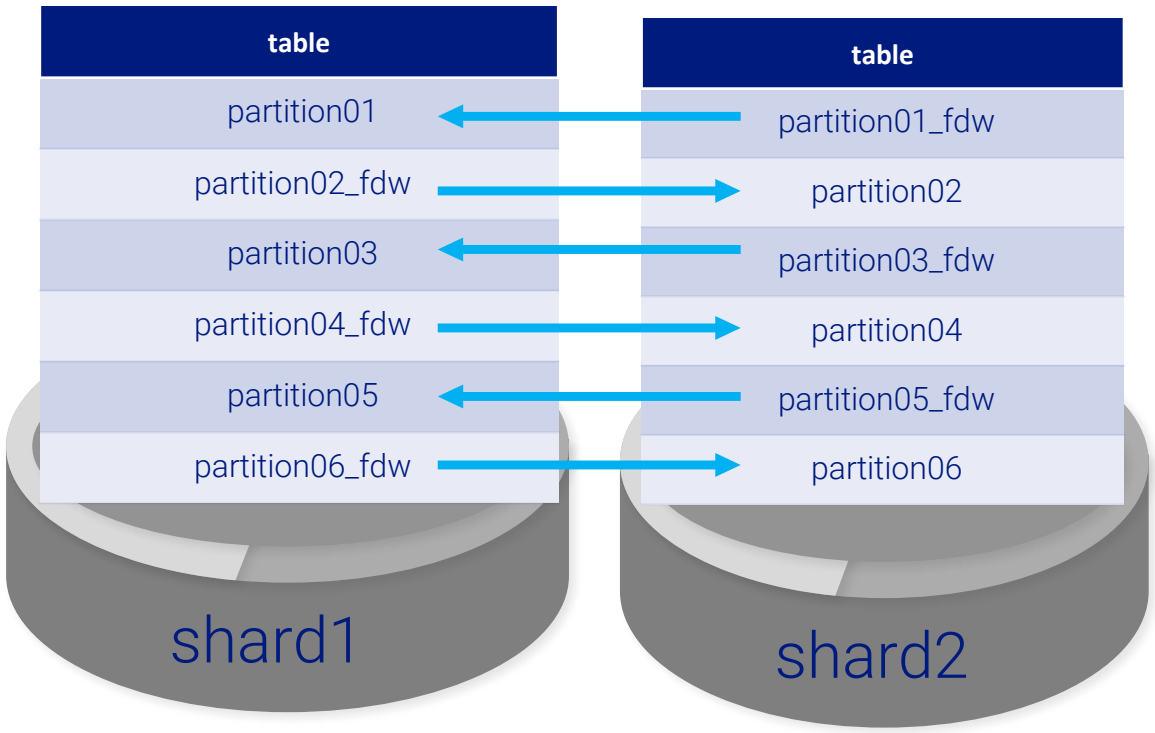
Shardman: Архитектура



Shardman: Масштабирование

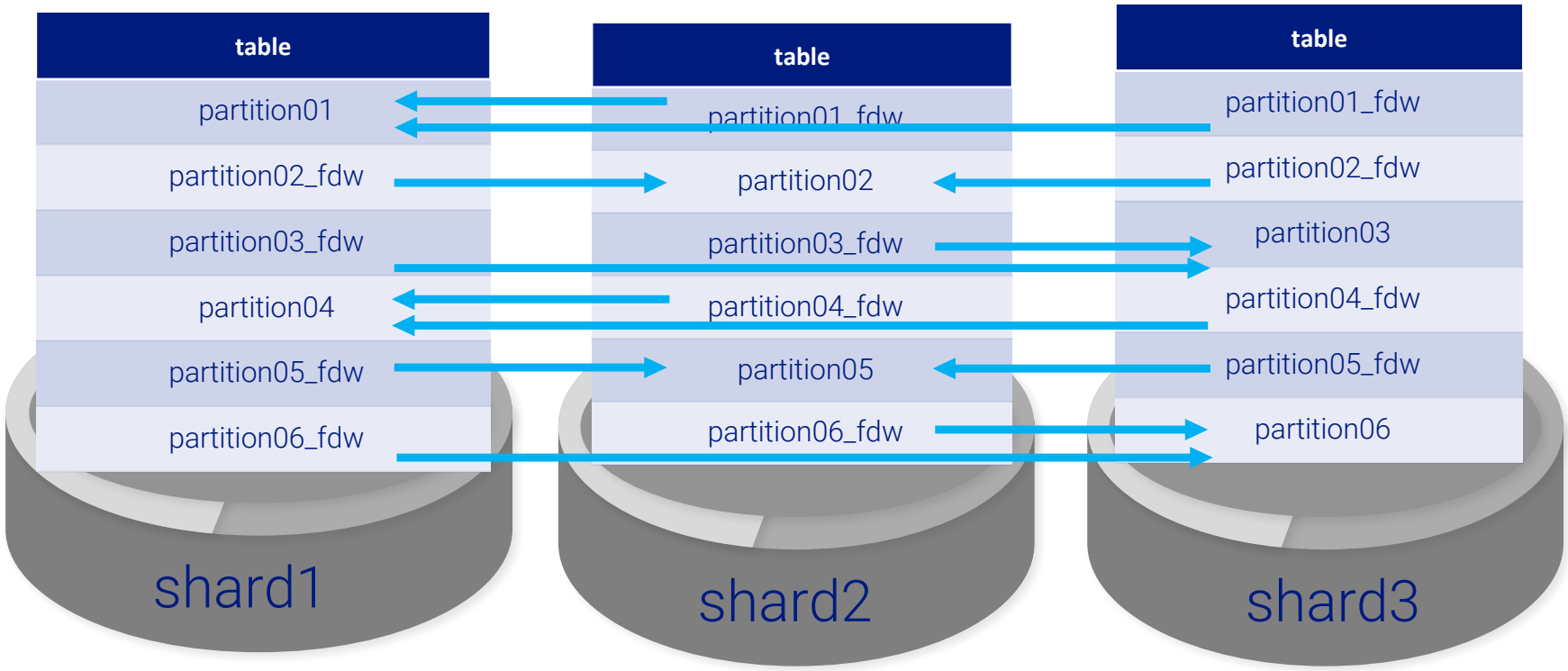


Shardman: Шардирование на основе строк



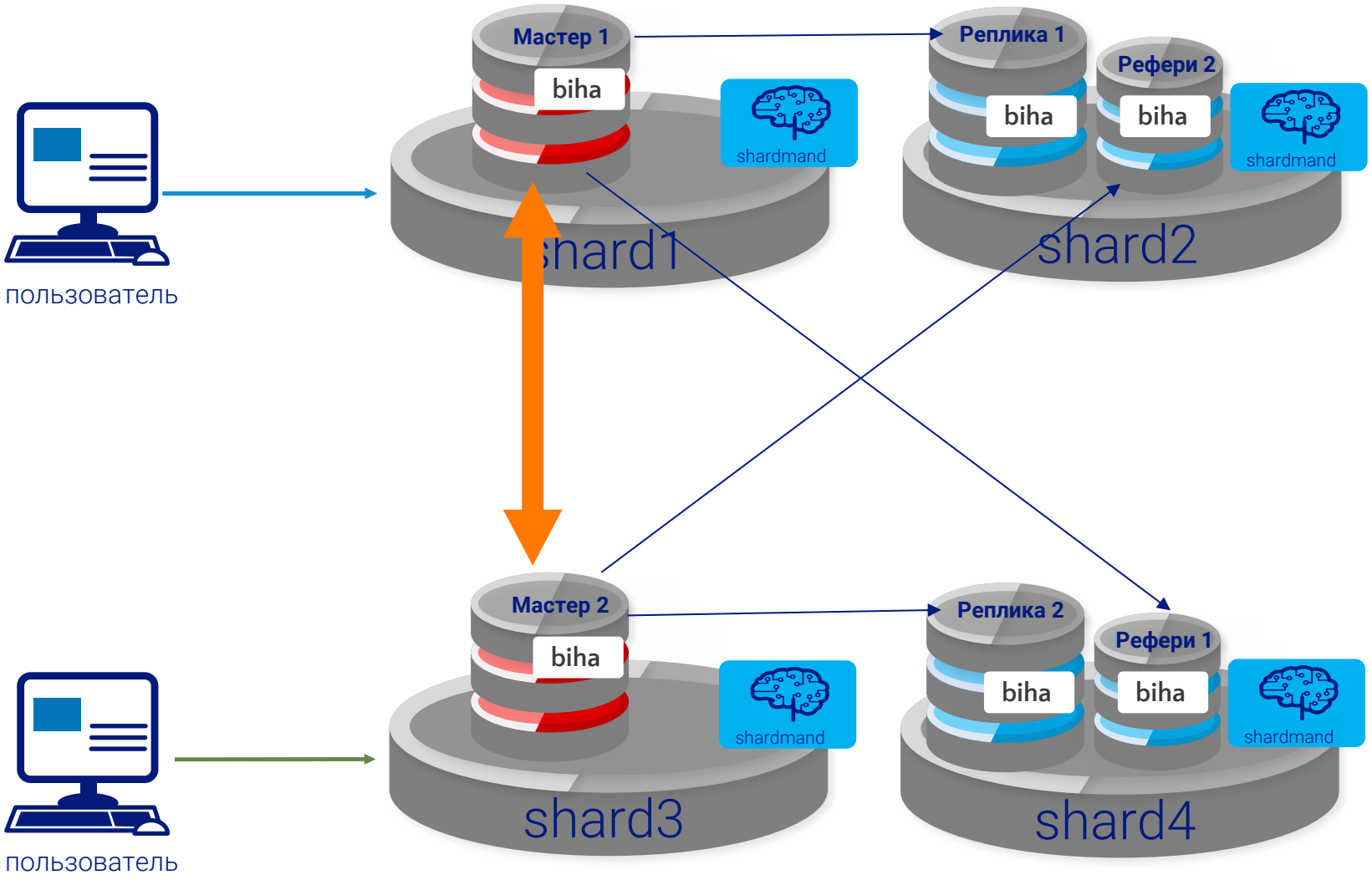
- Шардированная таблица состоит из набора секций, которые находятся физически на разных узлах кластера
- Число секций фиксированное при создании таблицы и неизменно
- На каждом узле содержится только часть данных таблиц
- Доступ к данным других секций обеспечивается через FDW
- Равномерное распределение данных по HASH значению определенных полей таблицы

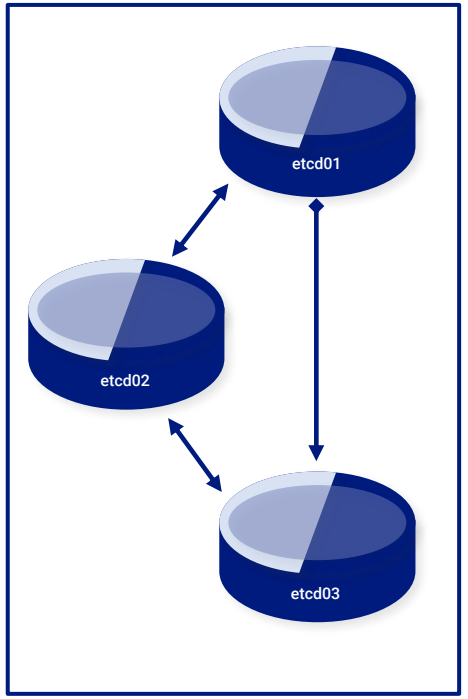
Shardman: Распределение данных при масштабировании



- При добавлении узла часть партиций переезжает
- Данные переливаются посредством логической репликации
- Каждый узел обрабатывает меньше количества данных
- Производительность линейно увеличивается

Shardman: Отказоустойчивость





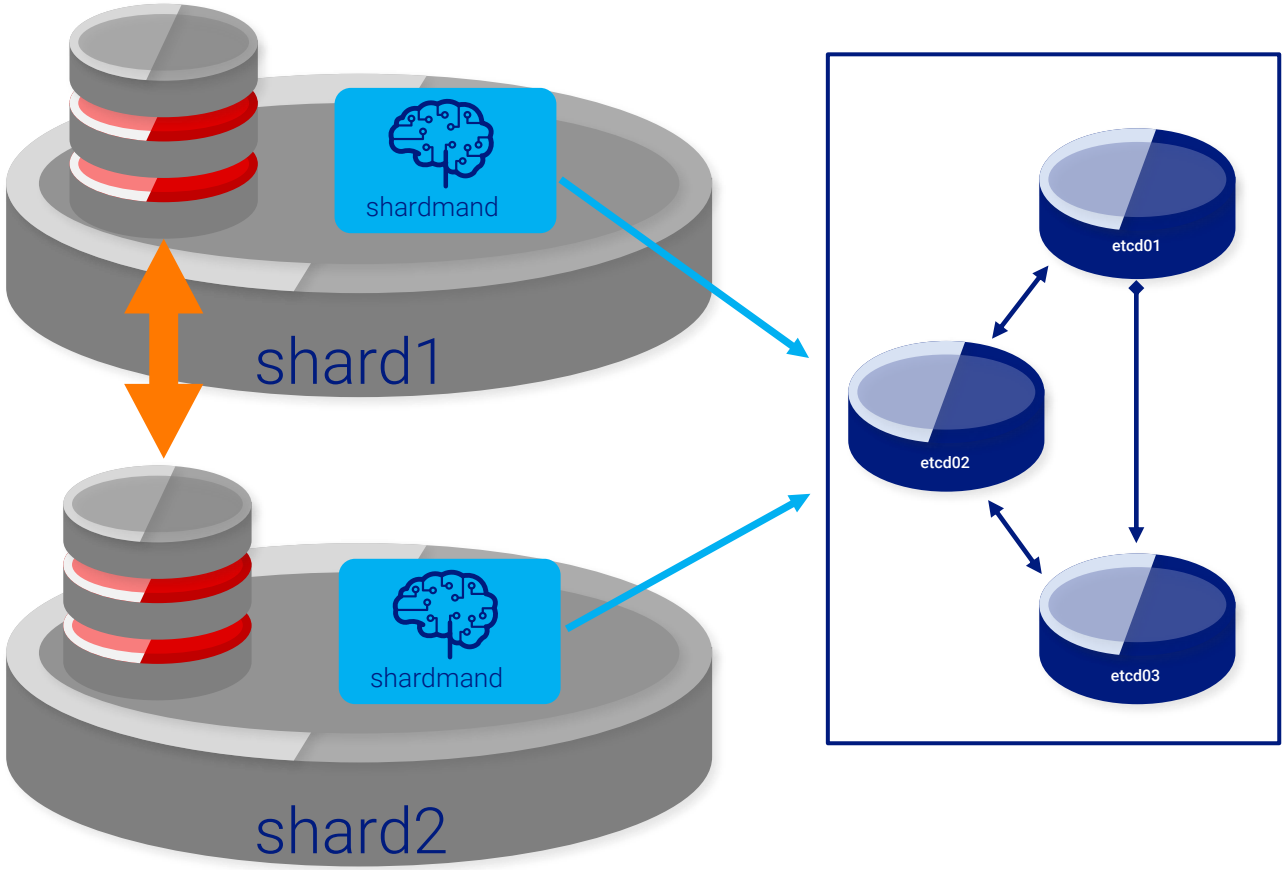
Shardman: Резервная копия

shardmanctl probackup backup

Создается точка согласованности по LSN и определяются значения LSN для каждой группы репликации.

shardmanctl probackup restore

Восстановление на LSN точки синхронизации каждой группы репликации с помощью pg_probackup.



Shardman: Что нового в 17-ой версии?

- Основан на ядре Postgres Pro Enterprise 17
- CFS
- PTRACK
- 64-битный XID
- AQO — адаптивный планировщик запросов
- BiHA — отказоустойчивость из коробки
- Средства диагностики, включая пакеты pgpro_pwr, pgpro_stats
- И другие полезные мелочи

Shardman: Roadmap

Q2 2025: выпуск базовой версии

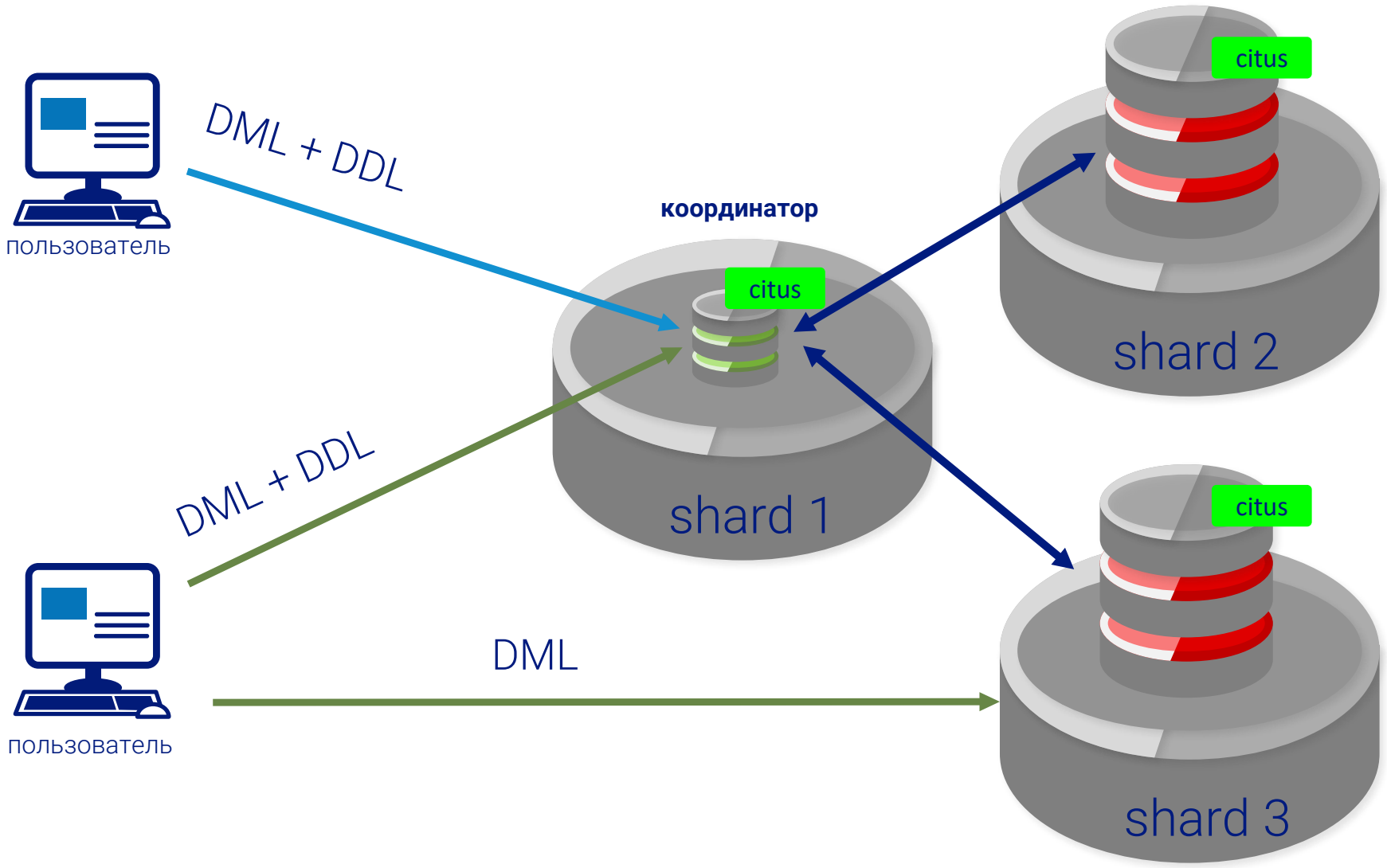
- Новая модель отказоустойчивости на базе BiHA
- Ядро заменено на Postgres Pro Enterprise 17 (с поддержкой ключевого функционала)
- Расширенная поддержка утилит аудита и мониторинга (pgpro_pwr)
- Поддержка AQO, AQE
- 64-битные счётчики транзакций

Q3 2025: выпуск второй версии

- Новый принцип организации транспортного уровня и работы с соединениями (повышенная устойчивость к большому числу подключений)
- Генерализованные оптимизации планировщика
- Повышение надежности переходных состояний конфигурации кластера
- Поддержка pg_upgrade на последующие версии

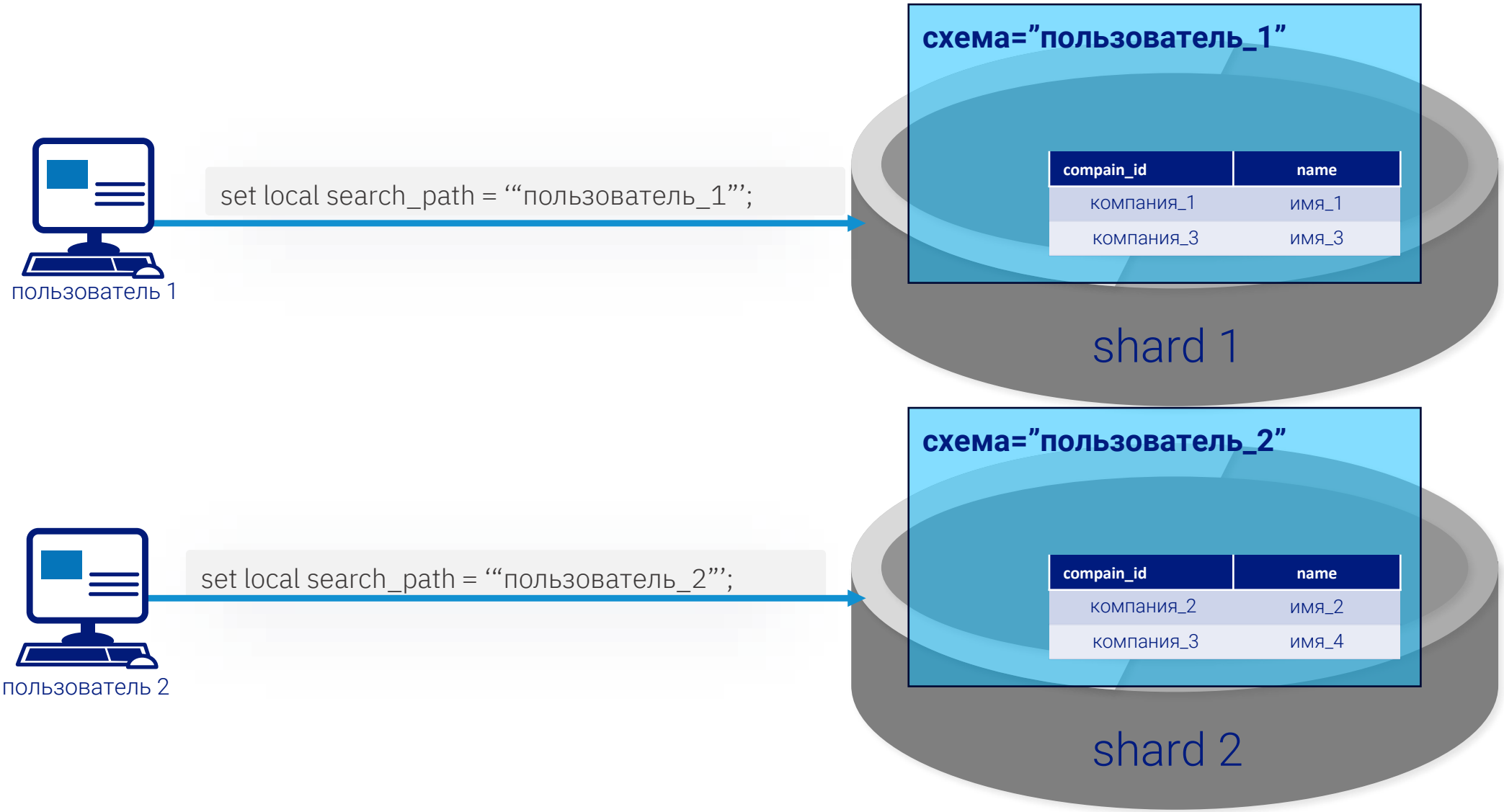
Q4 2025-Q1-2026: выпуск Shardman 18

Citus: Архитектура



Citus: Шардирование на основе строк

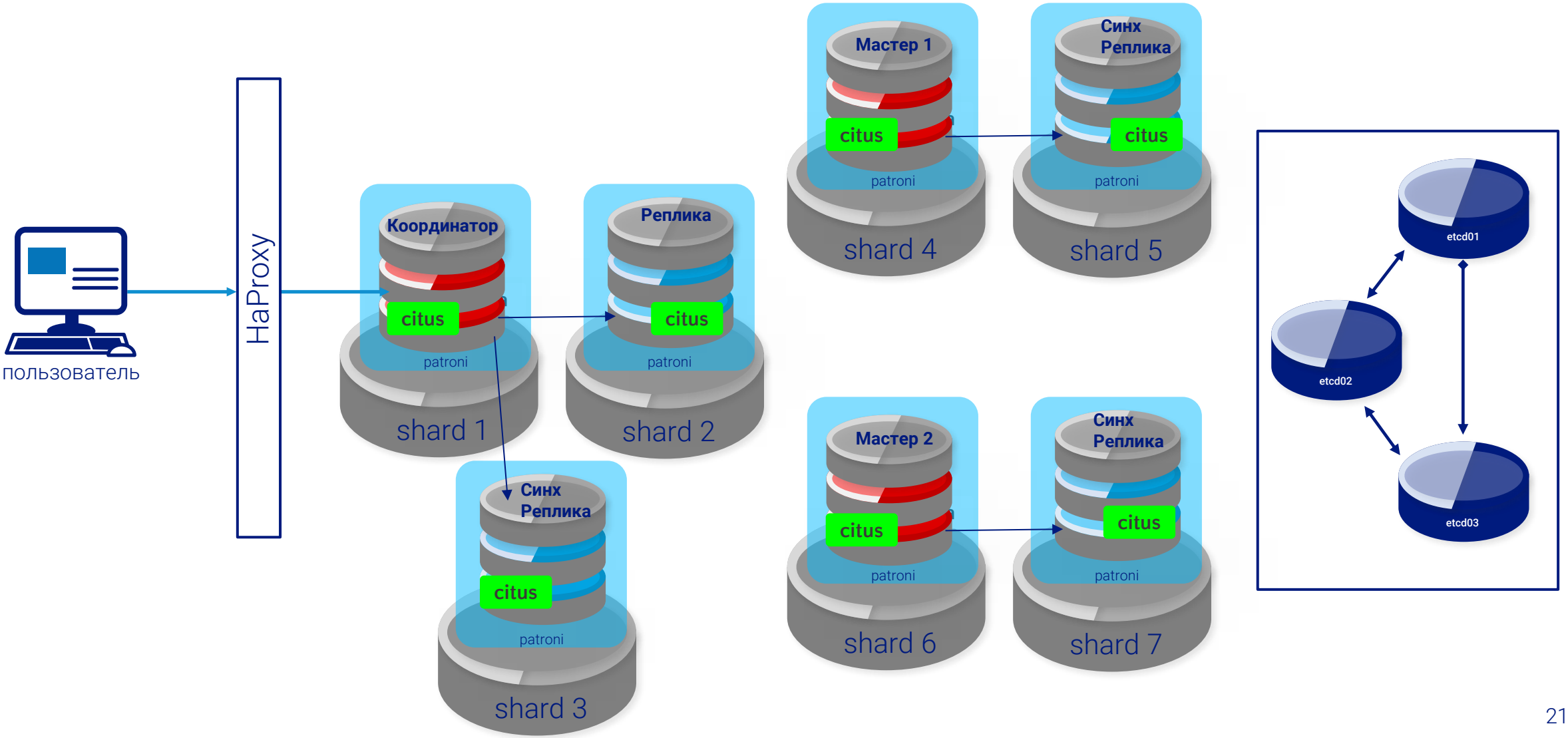
Citus: Шардирование на основе схемы



Citus: Отказоустойчивость

- Нет встроенной отказоустойчивости
- **patroni** (начиная с 4.0.3 поддерживает Enterprise)
- **pg_autofailover**
  - установка dev пакетов энтерпрайза
  - сборка pg_autofailover из исходников

Citus: Patroni



Citus: Резервное копирование

- Нет автоматизации
- В ручном режиме
  - **Для каждого узла**
    - Резервная копия (pg_basebackup, и т.д.)
    - Архивация WAL
  - Согласованность данных в кластере
    - citus_create_restore_point() — глобальная именованная точка

Итоги

- Шардирование, используя Shardman 17
  - OLTP
  - Встроенная отказоустойчивость шард
  - Консистентная резервная копия кластера
- Шардирование, используя Citus 12
  - OLAP
  - Отказоустойчивость шард, используя внешние компоненты
  - Консистентная резервная копия кластера

PostgresPro

Спасибо за внимание!