PostgresPro

Эволюция средств обеспечения отказоустойчивости СУБД Postgres Pro

Андрей Забелин, Дарья Репина

28 января 2025 года

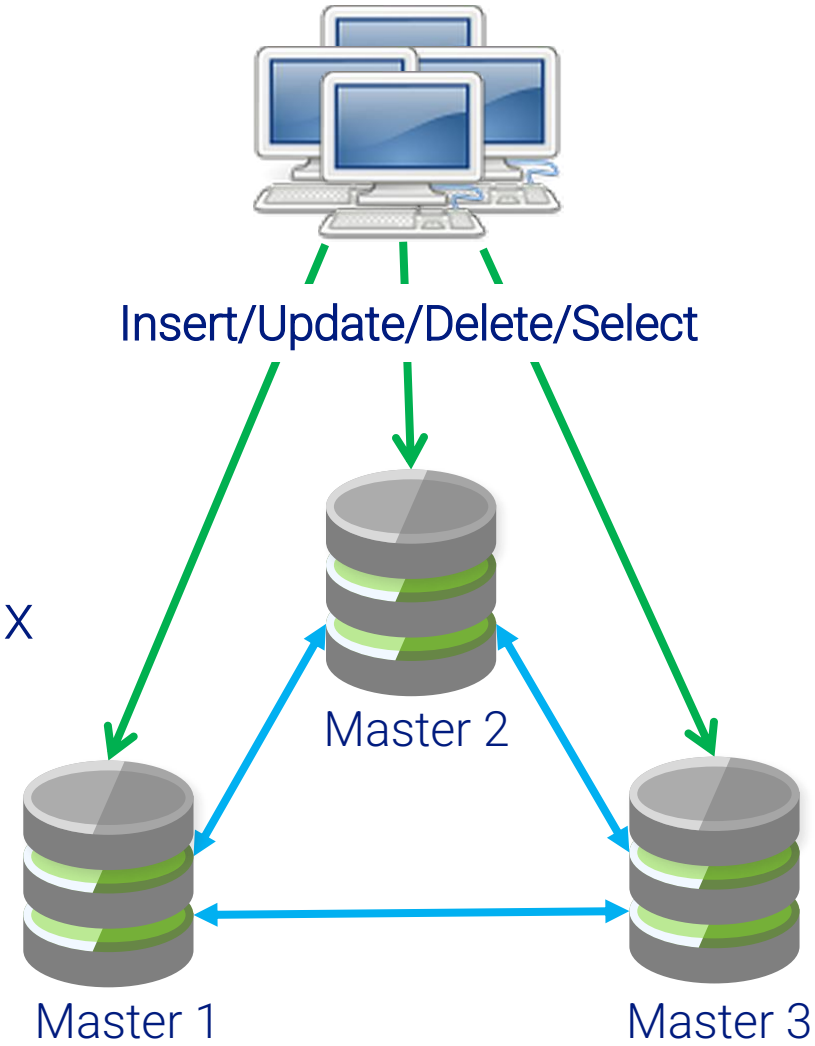
PGProDay

Технологии отказоустойчивости Postgres Pro

- Мультимастер
- BiHA
- Proxima

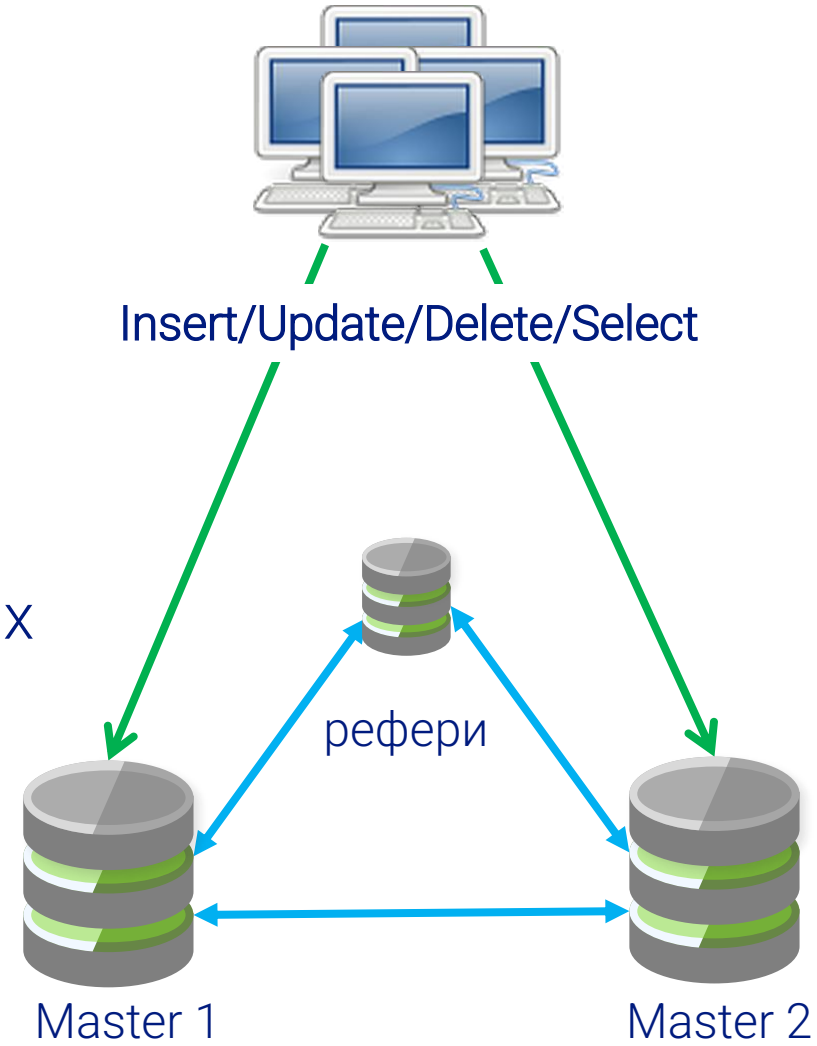
Мультимастер

- Основан на логической репликация
- Все узлы доступны на запись
- Масштабирование на чтение
- Управление транзакциями
- Гарантия синхронности данных на всех узлах
- Репликация изменений DDL

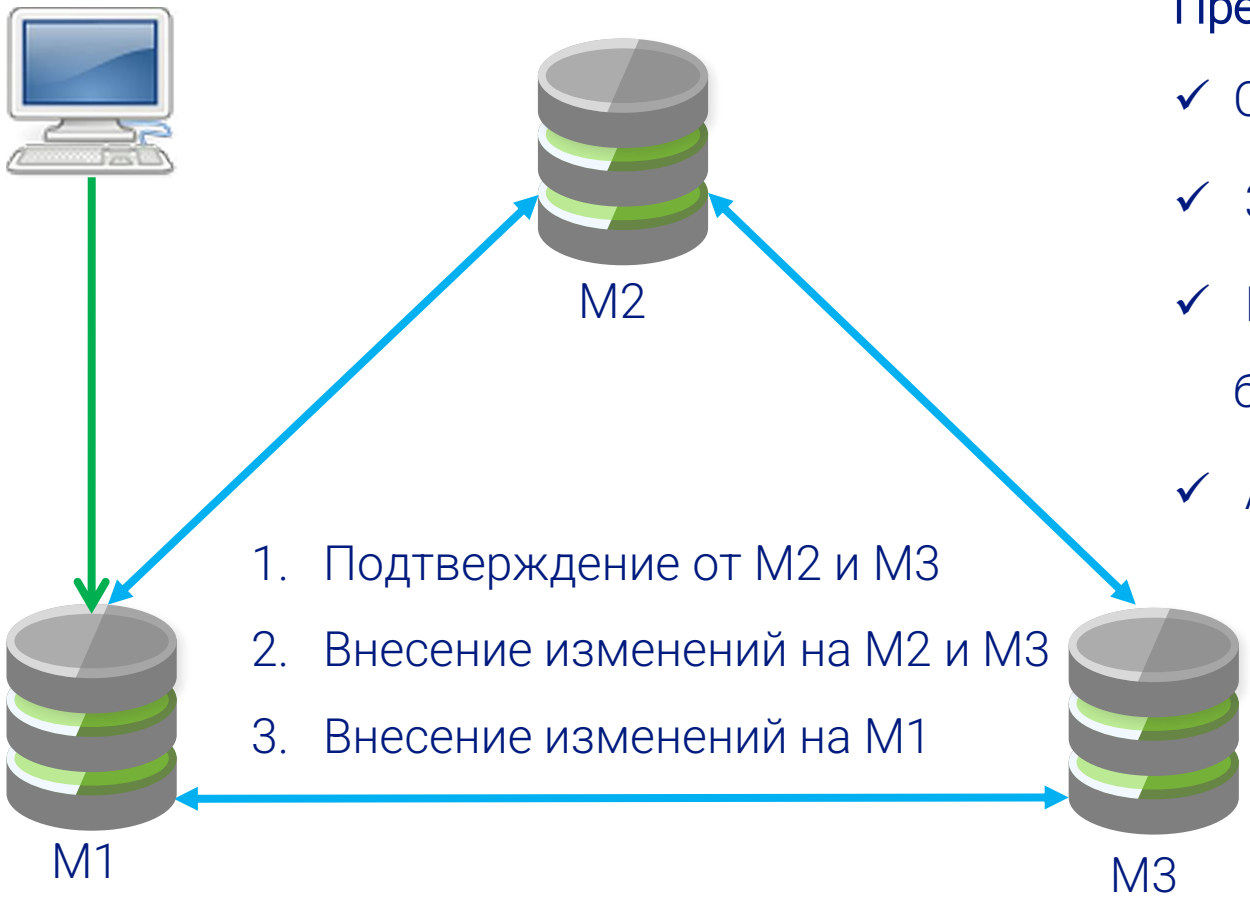


Мультимастер: 2+1

- Основан на логической репликация
- Все узлы доступны на запись
- Масштабирование на чтение
- Управление транзакциями
- Гарантия синхронности данных на всех узлах
- Репликация изменений DDL
- Возможность использовать рефери



Мультимастер : Управление транзакциями



Преимущества:

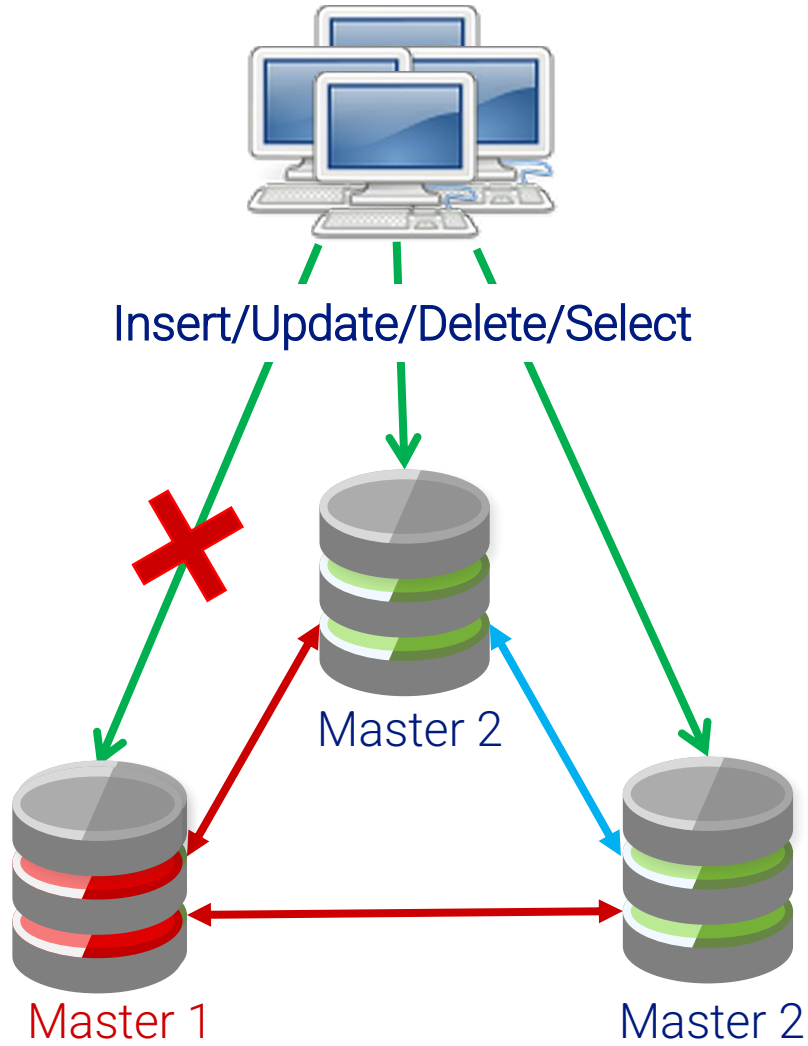
- ✓ Сохранение порядка транзакций
- ✓ Защита от повторного применения транзакций
- ✓ Предотвращение распределенных взаимных блокировок,
- ✓ Автоматическое разрешение конфликтов

3-х фазная фиксация транзакции:

1. PREPARE
2. PRECOMMIT
3. COMMIT

Мультимастер : Обработка сбоев

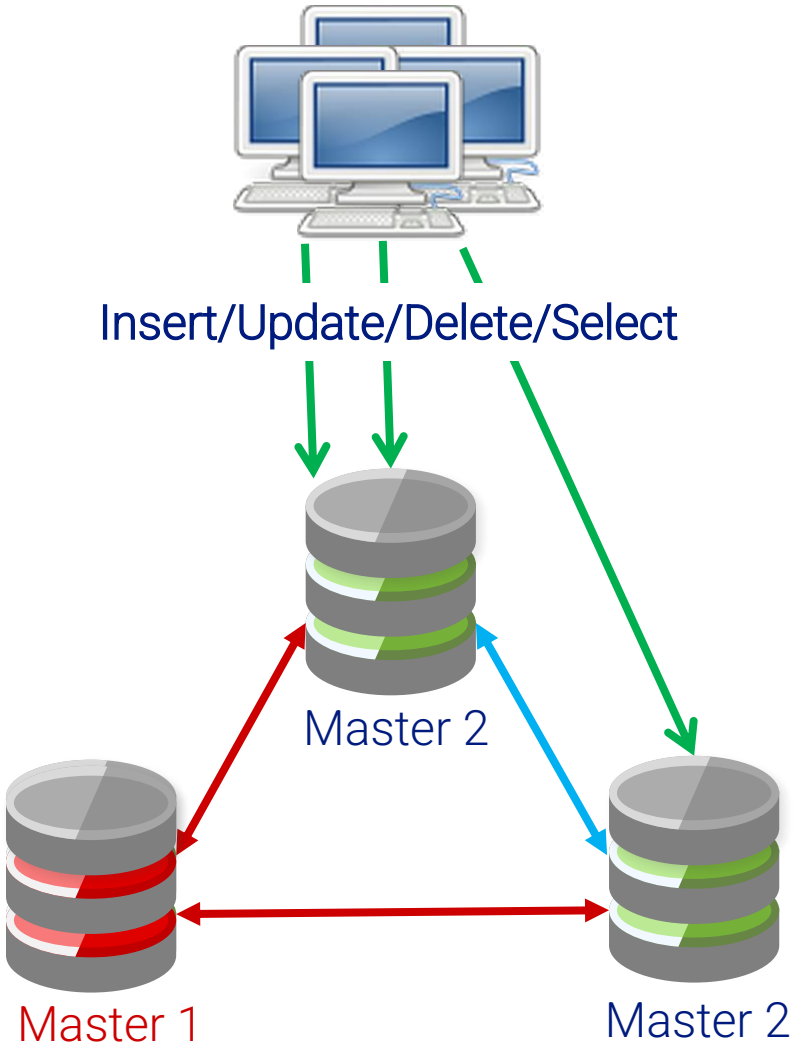
- Быстрое переключение
- Нет потери данных
- Исключение split-bran:
  - ✓ при нёчетном количестве узлов
  - ✓ при чётном количестве узлов + рефери (2+1)



Мультимастер : Обработка сбоев

- Быстрое переключение
- Нет потери данных
- Исключение split-bran:
  - ✓ при нёчетном количестве узлов
  - ✓ при чётном количестве узлов + рефери (2+1)
- Автоматическое восстановление узла
  - Режим восстановления данных (CatchUp)

В Enterprise 17 время восстановления узла ускорено за счет параллельного применения неконфликтующих транзакций

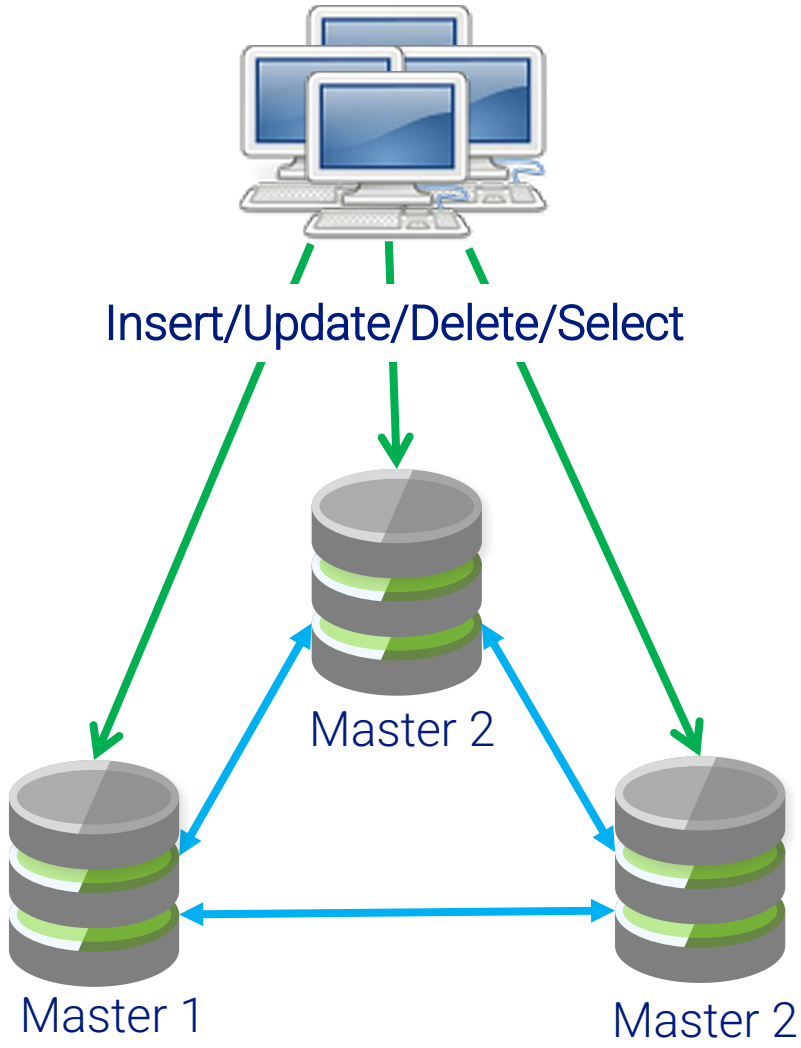


Мультимастер : Обработка сбоев

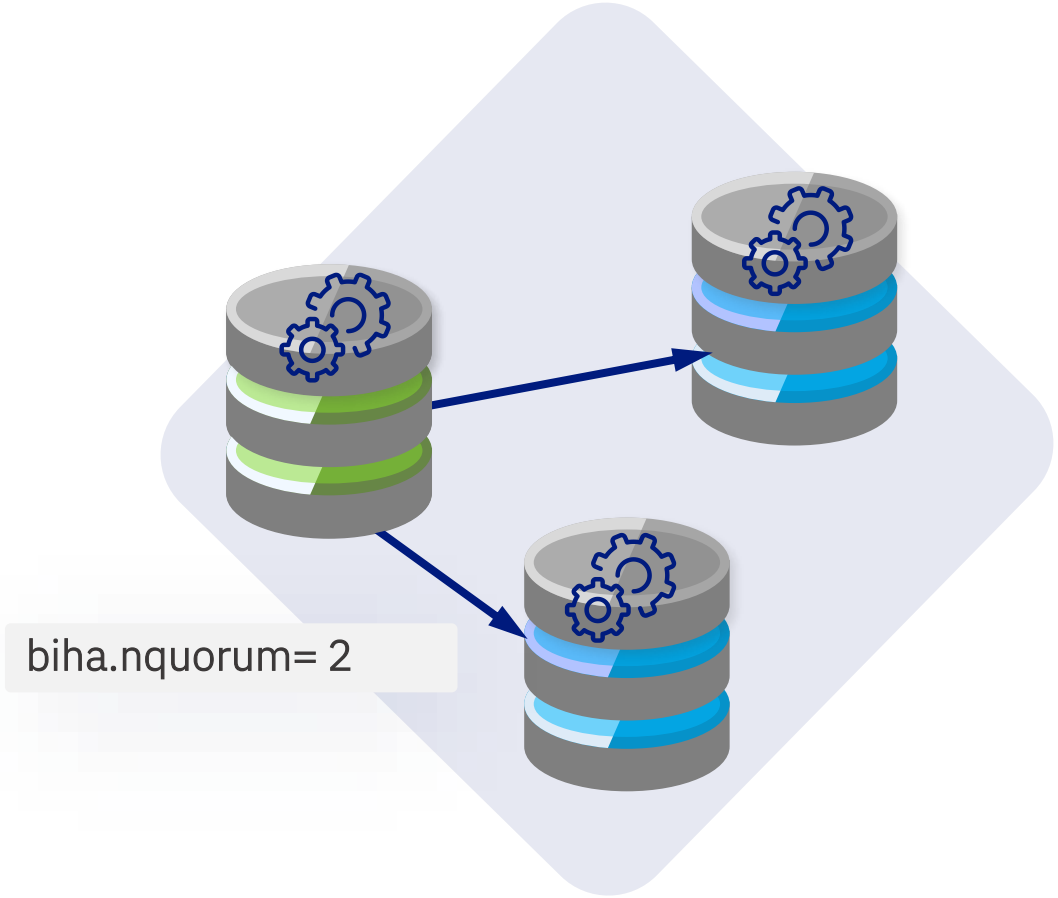
- Быстрое переключение
- Нет потери данных
- Исключение split-bran:
  - ✓ при нёчетном количестве узлов
  - ✓ при чётном количестве узлов + рефери (2+1)
- Автоматическое восстановление узла
  - Режим восстановления данных (CatchUp)

В Enterprise 17 время восстановления узла ускорено за счет параллельного применения неконфликтующих транзакций

  - Переход в активный режим Active



BiHA: встроенный отказоустойчивый кластер



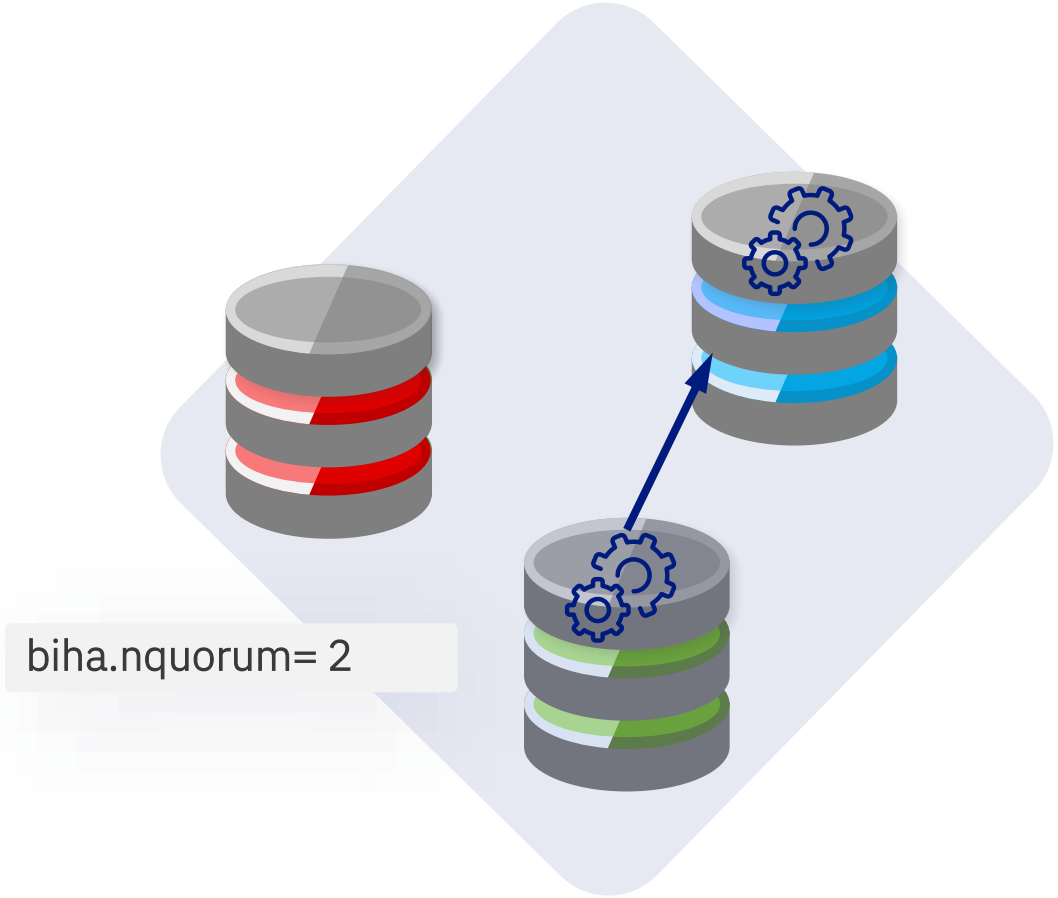
Встроен в ядро Postgres Pro Enterprise начиная с версии 16

Простая установка и конфигурирование

Не требуется установка дополнительного ПО

Лидер продолжает работать, если соблюдается кворум

BiHA: встроенный отказоустойчивый кластер



Встроен в ядро Postgres Pro Enterprise начиная с версии 16

Простая установка и конфигурирование

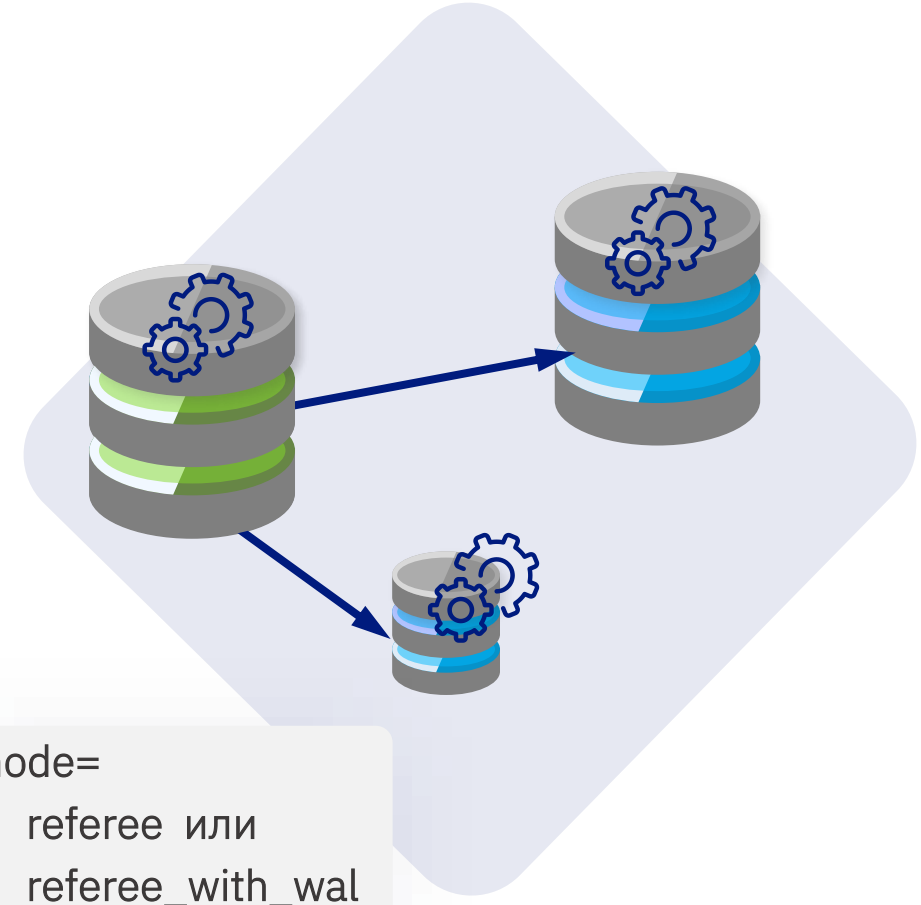
Не требуется установка дополнительного ПО

Лидер продолжает работать, если соблюдается кворум

В случае сбоя ведомые предлагают себя кандидатами и организуются выборы нового лидера

Защита от split-brain

BiHA: узел рефери



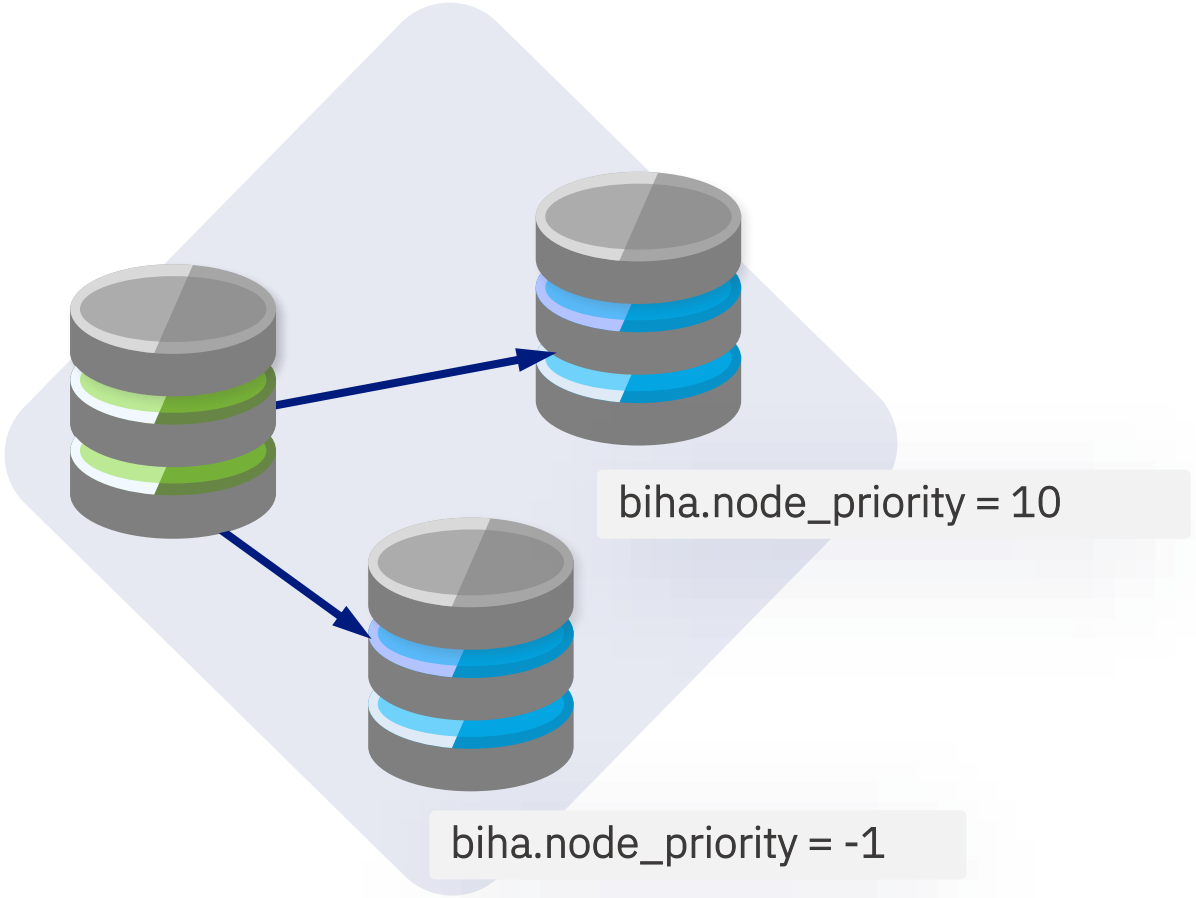
mode=
referee или
referee_with_wal

Рефери – легковесный экземпляр, который не содержит пользовательских данных, но является членом кластера BiHA:
сам кандидатом на звание нового лидера
не выступает, но участвует в голосовании

Рефери может работать в режиме referee_with_wal :

- Получает весь WAL и фильтрует его, применяются только системные записи WAL без пользовательских данных
- При сбое лидера может отправлять WAL кандидату на нового лидера, если тот отстаёт от рефери

BiHA: приоритеты кандидатов в лидеры

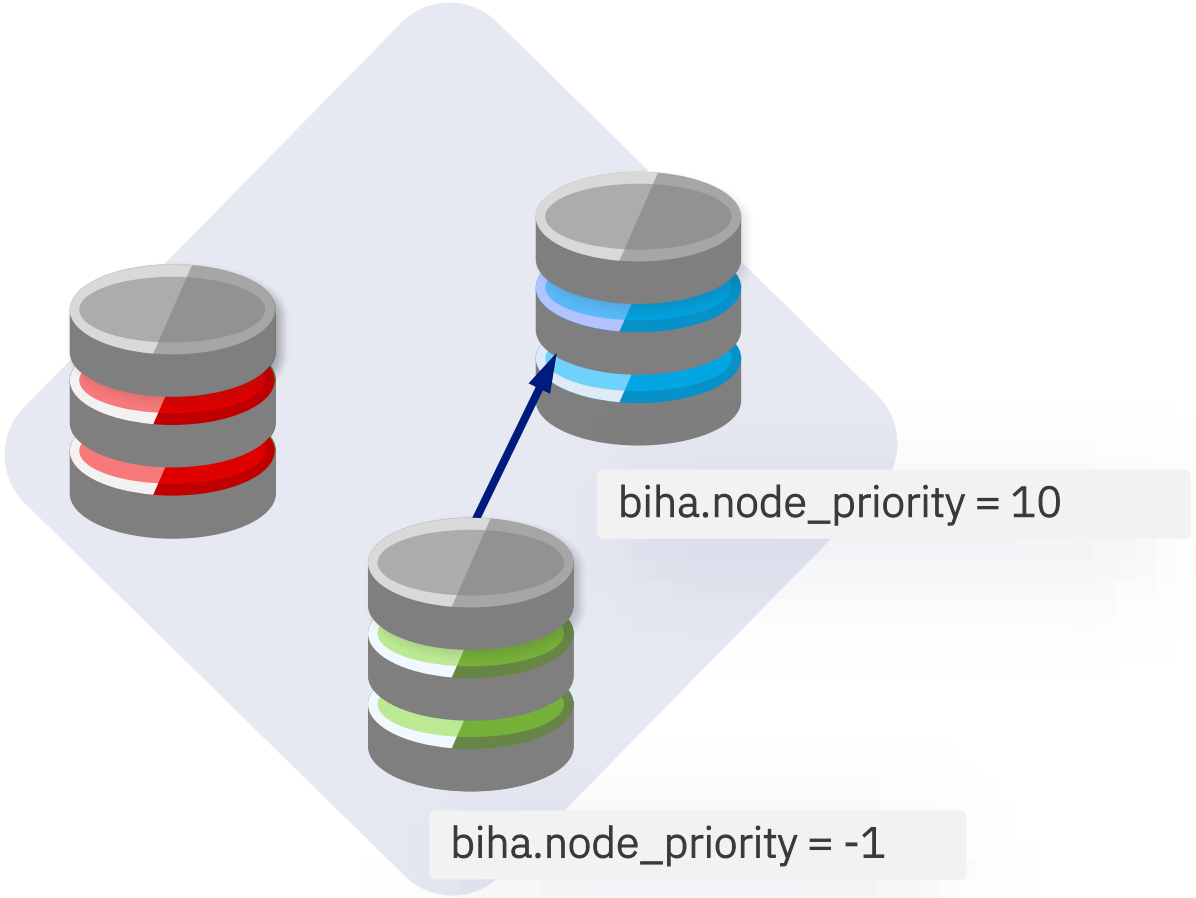


Можно установить приоритеты для выдвижения узла в качестве кандидата

Приоритет узла в секундах определяет тайм-аут, по достижении которого узел предложит себя в качестве кандидата на выборах.

Только в синхронном кластере

BiHA: приоритеты кандидатов в лидеры

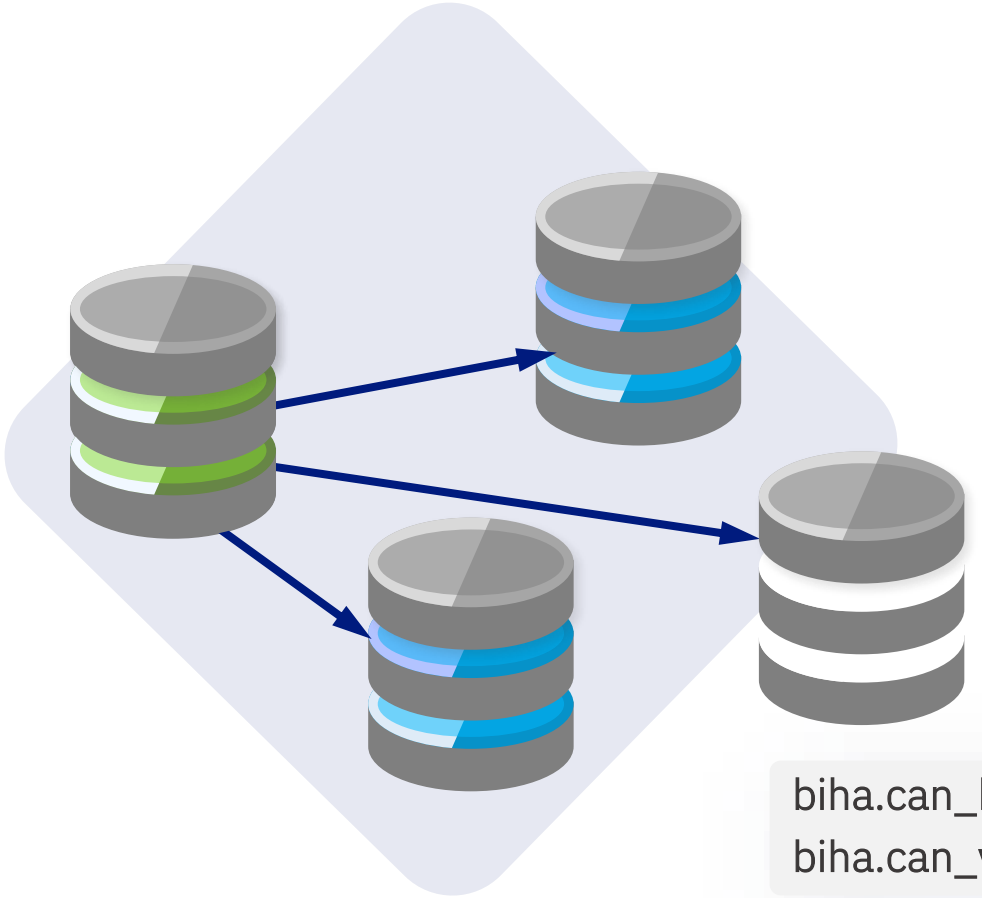


Можно установить приоритеты для выдвижения узла в качестве кандидата

Приоритет узла в секундах определяет тайм-аут, по достижении которого узел предложит себя в качестве кандидата на выборах.

Только в синхронном кластере

BiHA: узел, который не может стать лидером



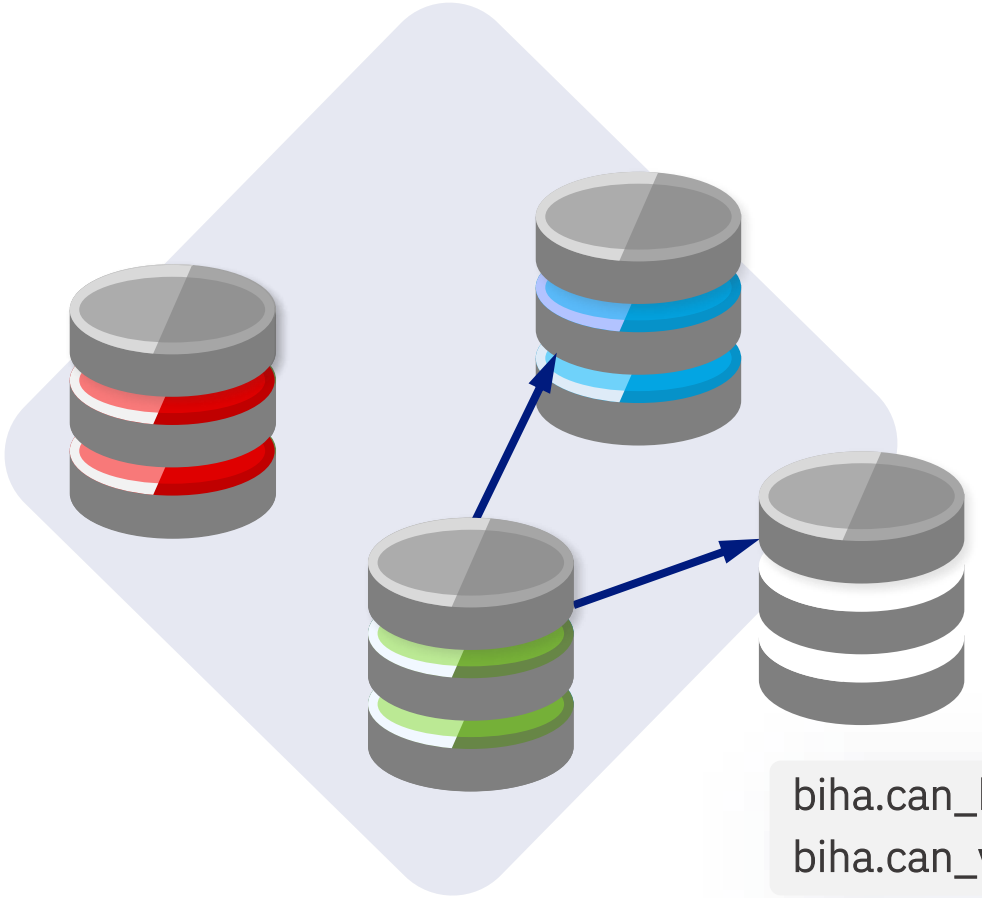
```
biha.can_be_leader = false
biha.can_vote = false
```

Если вы считаете, что один из серверов по какой-то причине не должен никогда стать лидером, например:

- недостаточно производительный,
- подключён к другой сети,
- включено отложенное применение WAL,

то можно запретить узлу становиться кандидатом и даже голосовать

BiHA: узел, который не может стать лидером



```
biha.can_be_leader = false
biha.can_vote = false
```

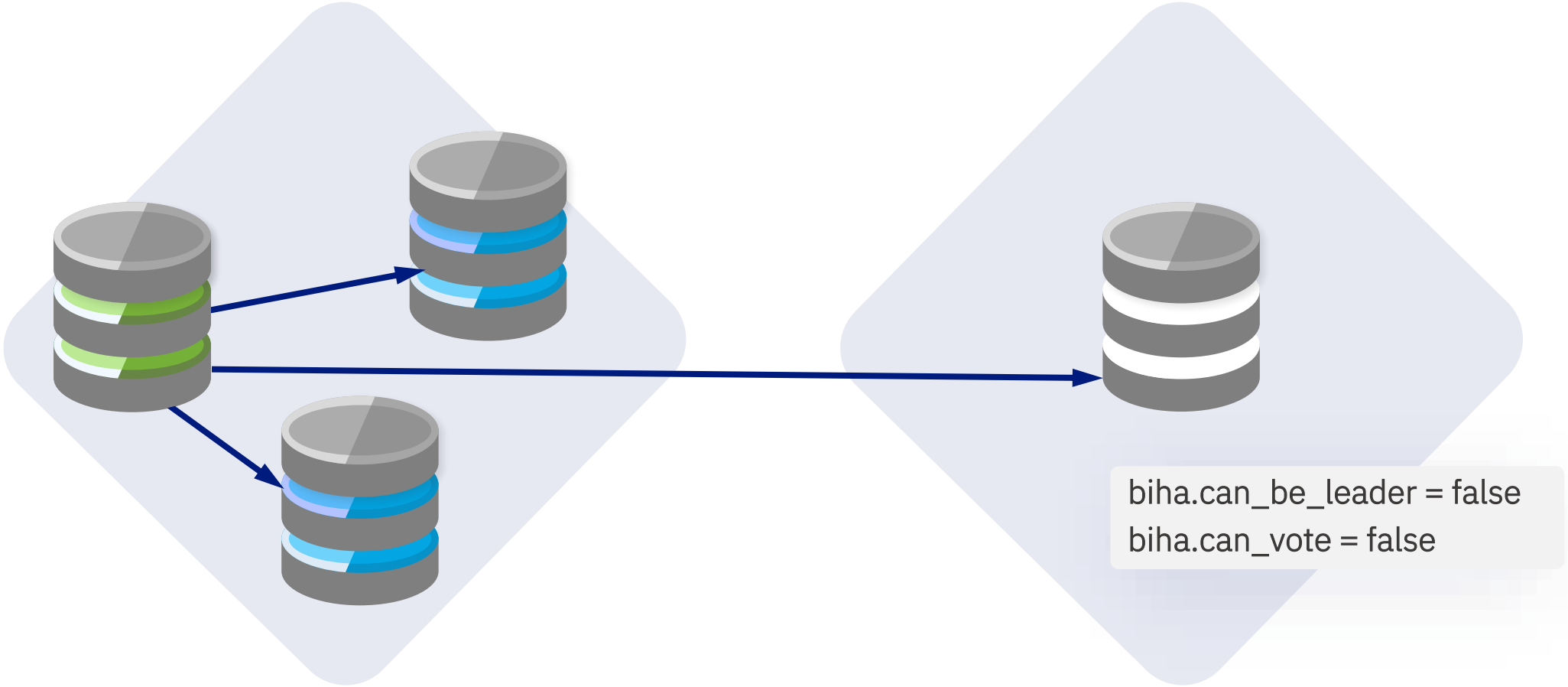
Если вы считаете, что один из серверов по какой-то причине не должен никогда стать лидером, например:

- недостаточно производительный,
- подключён к другой сети,
- включено отложенное применение WAL,

то можно запретить узлу становиться кандидатом и даже голосовать

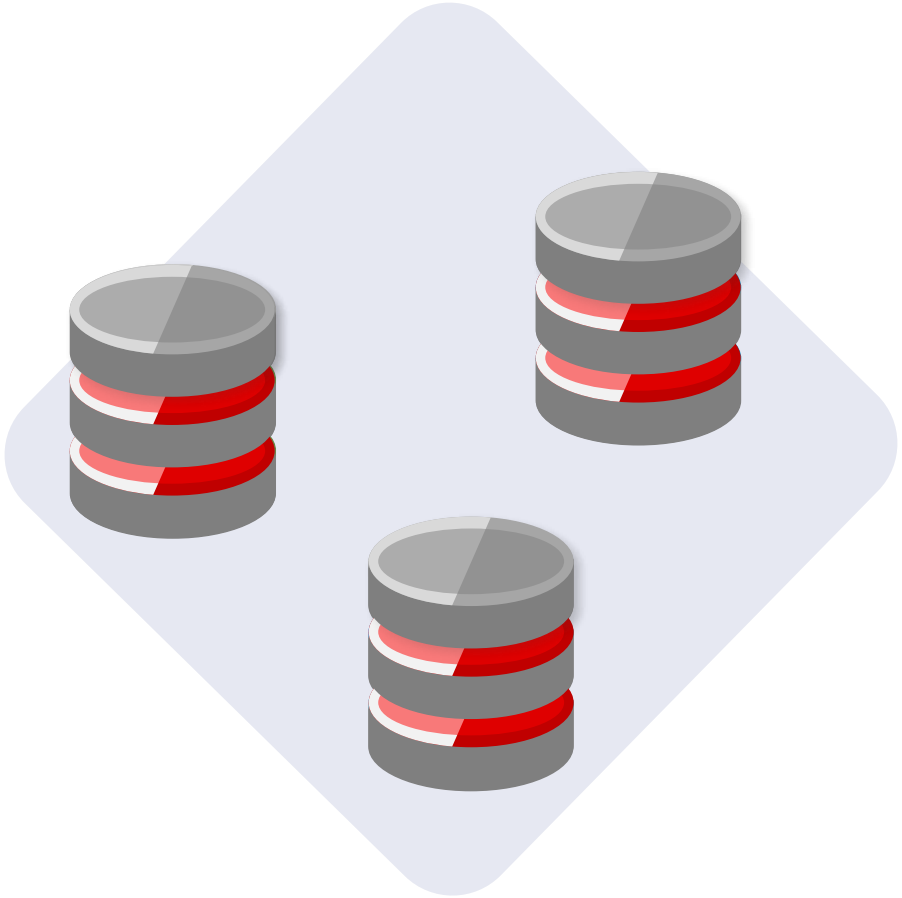
BiHA: узел в отдельном ЦОД

Лидером может стать только узел в основном ЦОД



BiHA: узел в отдельном ЦОД

Сбой всего ЦОД не приведёт к потери данных

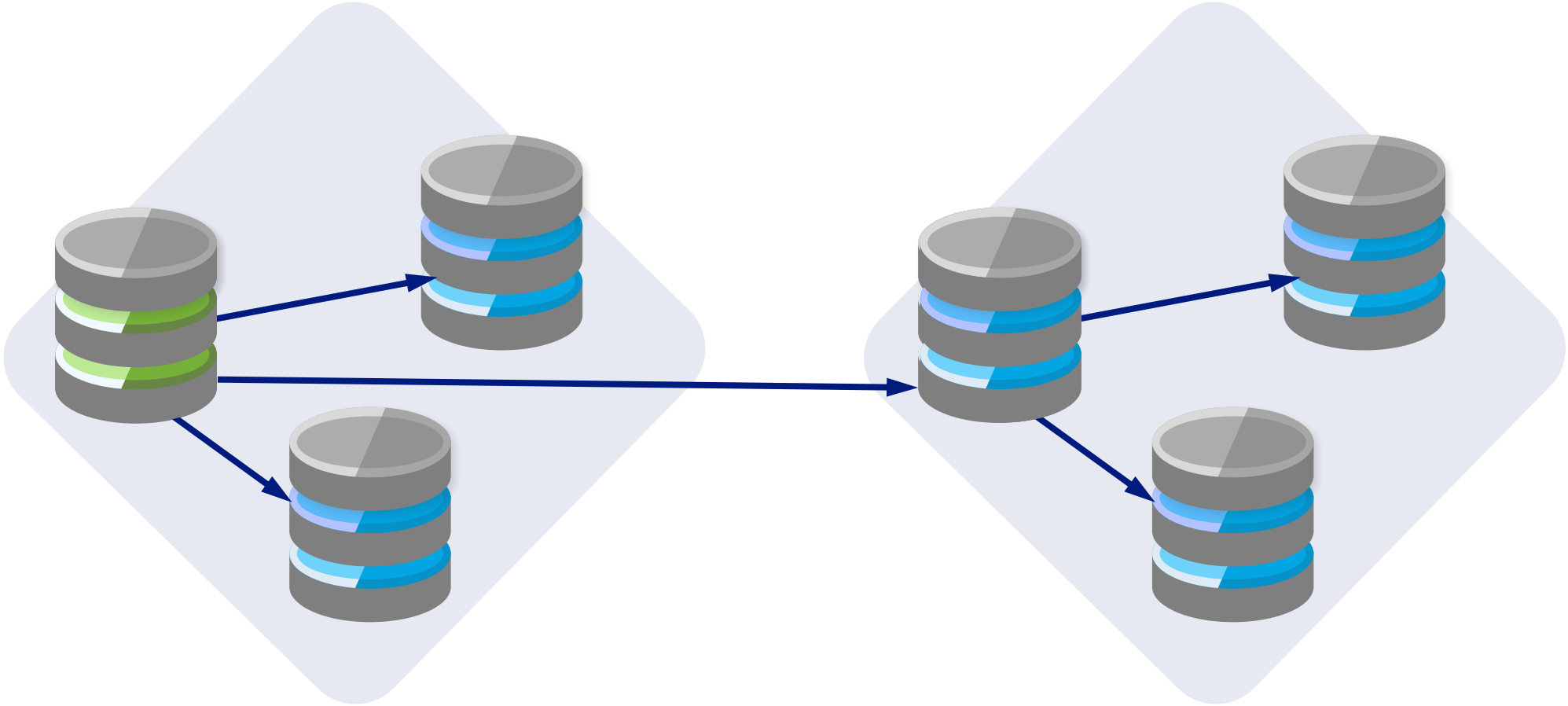


переключение на
резервный ЦОД
не автоматическое,
но автоматизированное

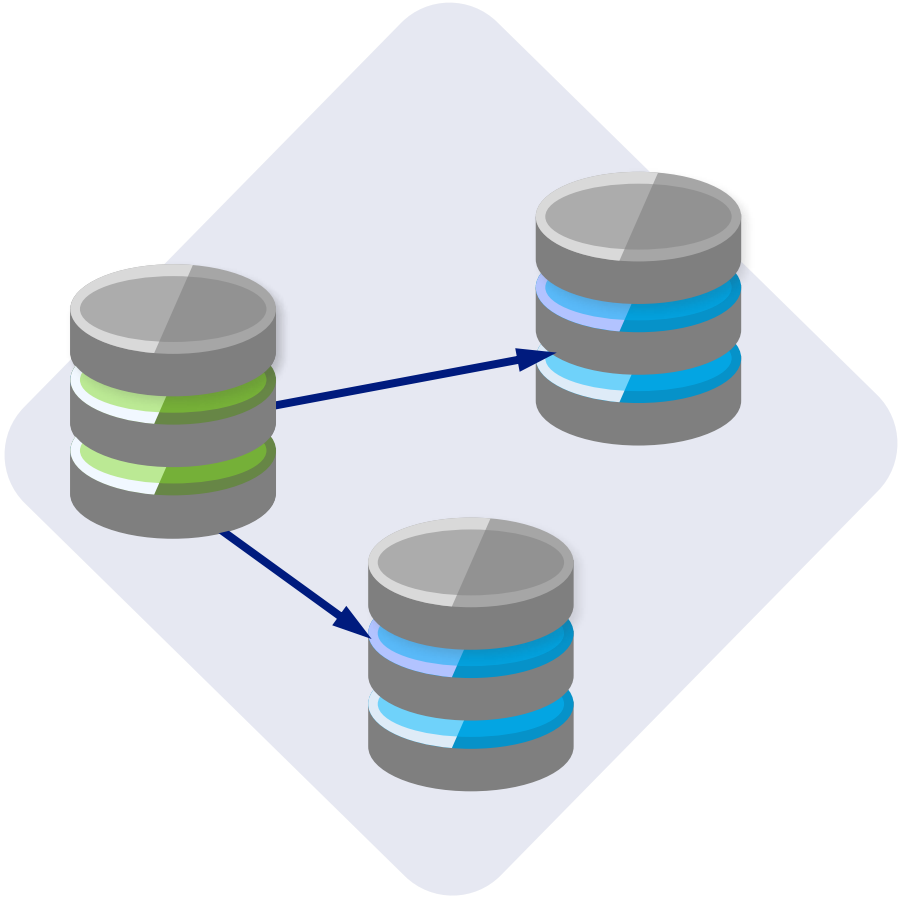
```
biha.can_be_leader = true
biha.can_vote = true
```

BiHA: катастрофоустойчивость

Катастрофоустойчивость – основное направление развития в 2025 году



BiHA : Функции-обработчики смены состояний



Вы создаёте SQL-функцию и регистрируете её в качестве обработчика. SQL-функция может уведомлять внешние сервисы о событиях в BiHA-кластере.

Например:

- CANDIDATE_TO_LEADER вызывается на узле, выбранном в качестве нового лидера.
- LEADER_CHANGED вызывается на каждом узле BiHA-кластера при смене лидера.

BiHA : Функции-обработчики смены состояний

Вы создаёте SQL-функцию и регистрируете её в качестве обработчика. SQL-функция может уведомлять внешние сервисы о событиях в BiHA-кластере.

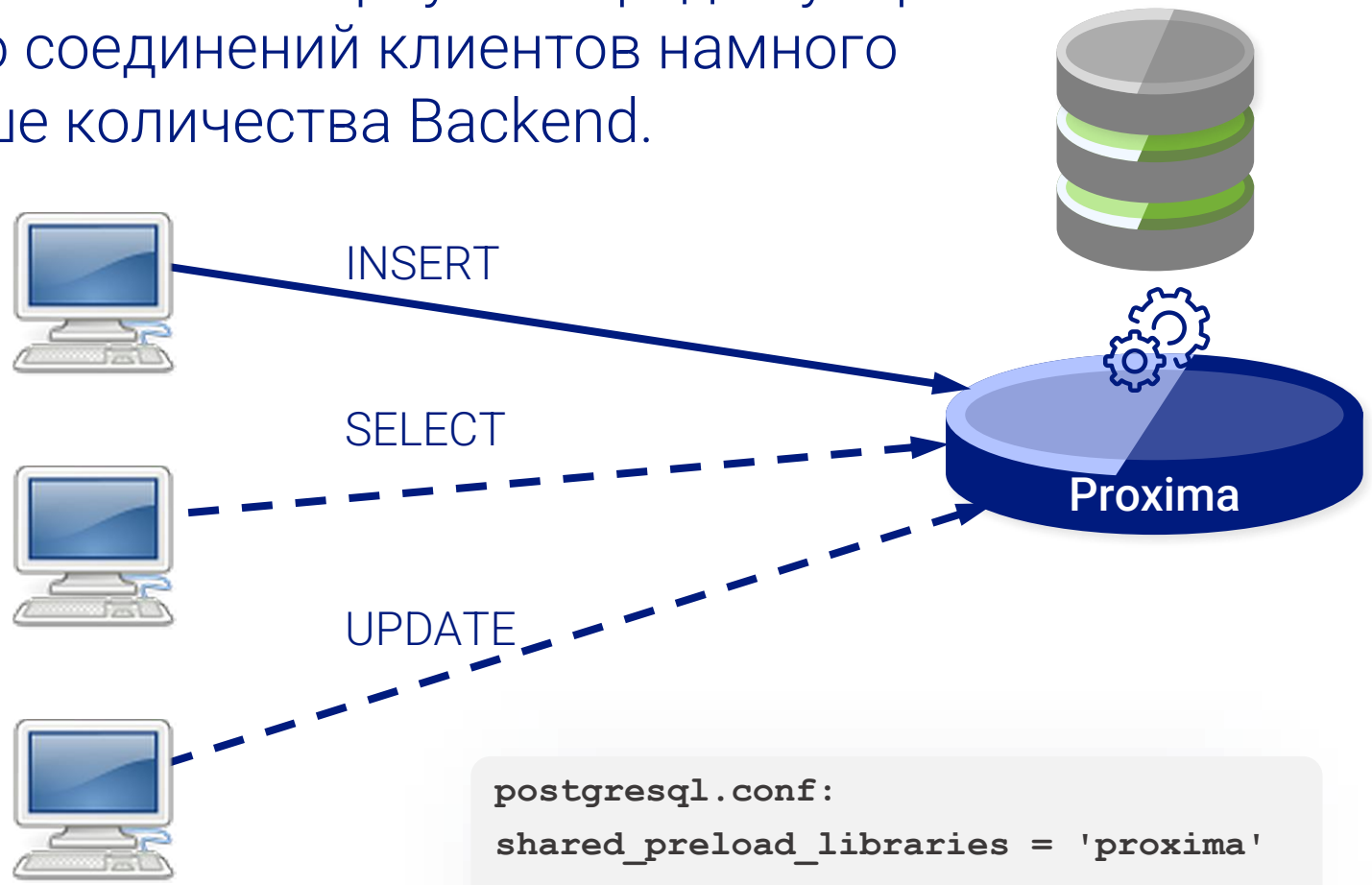
Например:

- CANDIDATE_TO_LEADER вызывается на узле, выбранном в качестве нового лидера.
- LEADER_CHANGED вызывается на каждом узле BiHA-кластера при смене лидера.

Если исполнение функции-обработчика длится дольше, чем biha.callbacks_timeout, BiHA останавливает исполнение и продолжает работать в обычном режиме.

Proxima

Proxima – это в первую очередь пулер. Число соединений клиентов намного больше количества Backend.



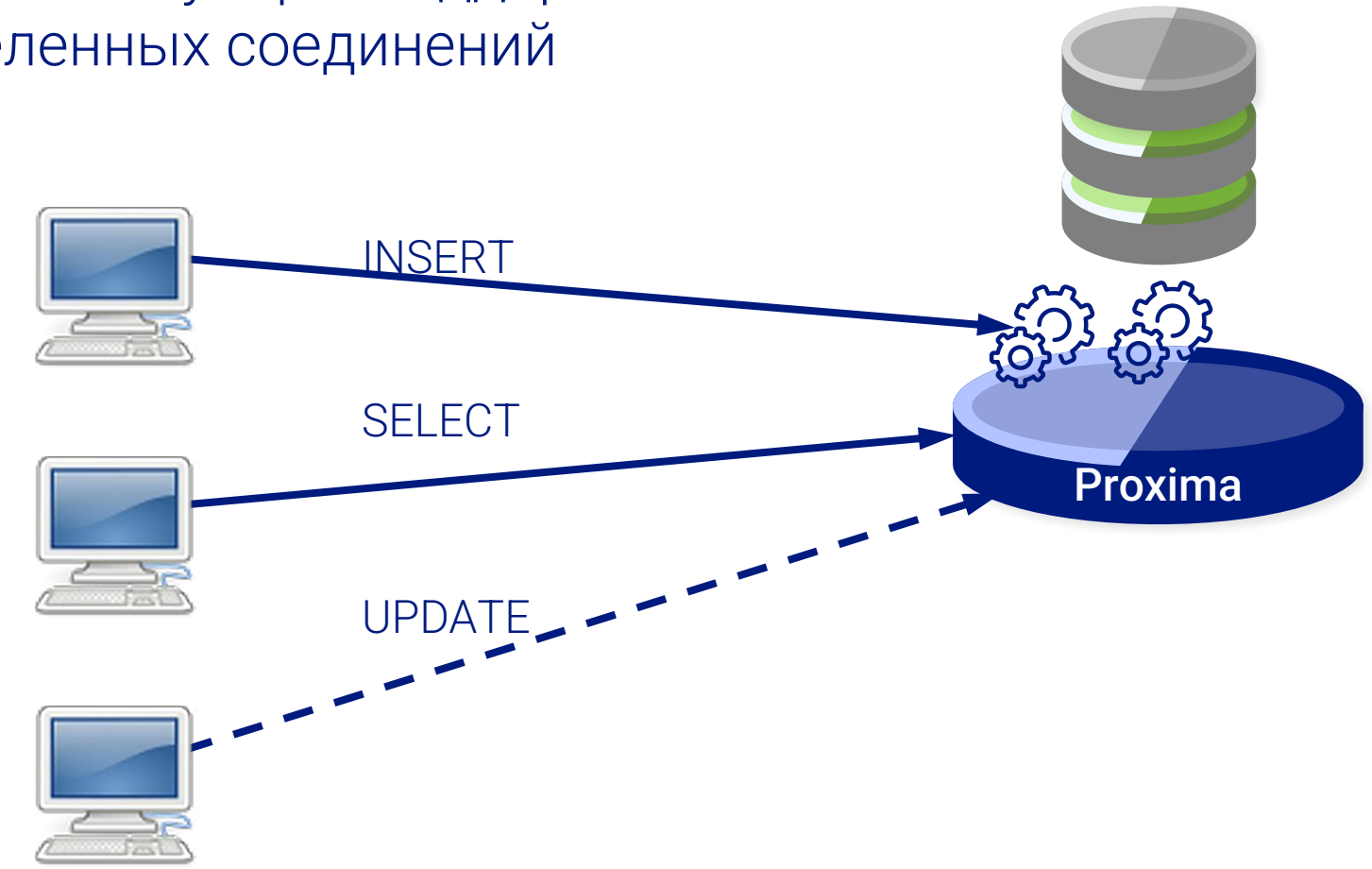

```
postgresql.conf:
shared_preload_libraries = 'proxima'
proxima.cluster_mode = 'standalone'
```

Backend пул - компонент Proxima - множество соединений с Backend процессами Postgres.

Каждый свободный Backend может быть использован для выполнения транзакции клиента,
после чего соединение возвращается в pool в качестве свободного.

Proxima

Proxima – пулер с поддержкой выделенных соединений



Dedicated режим: если клиент создает в процессе своей работы сессионный объект, то дальнейшую работу с ним он проводит через конкретный Backend процесс.

Proxima

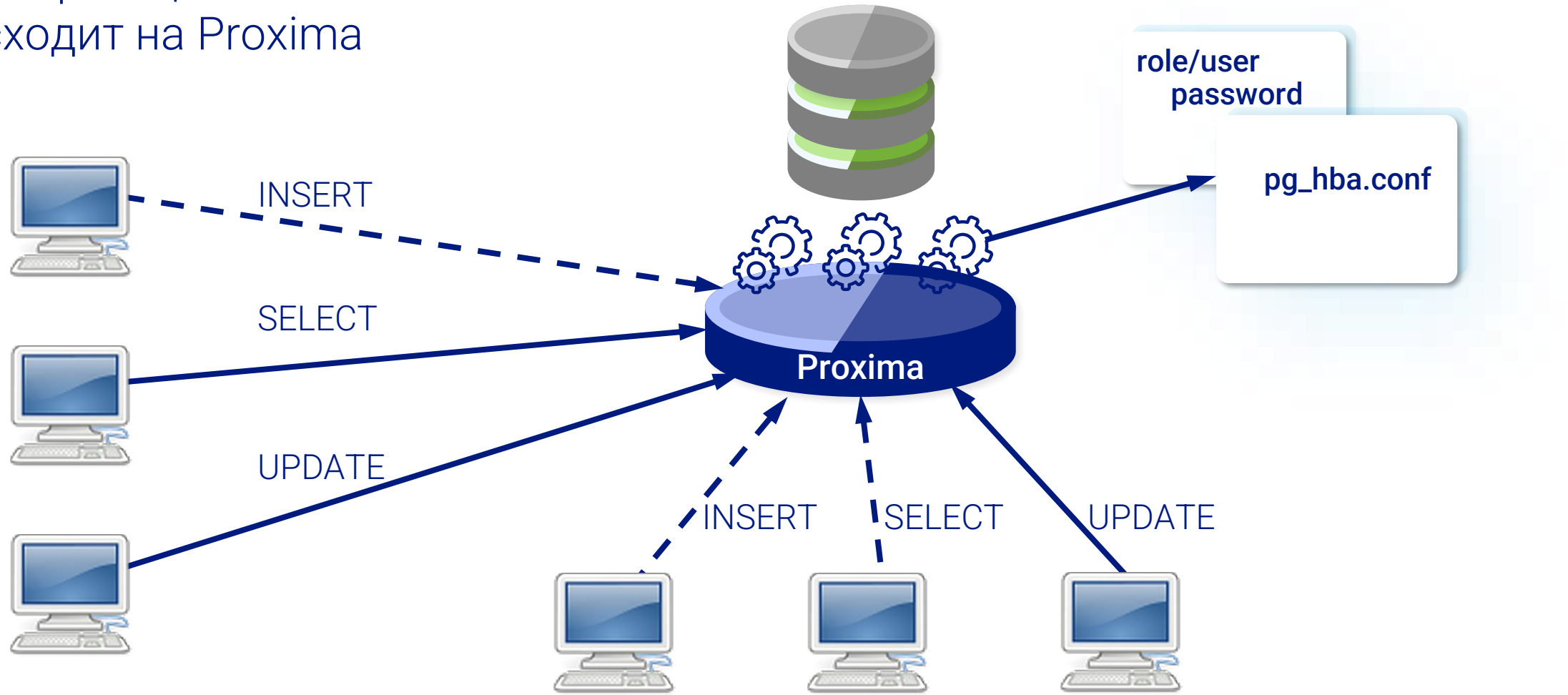
В рамках одной клиентской сессии может происходить переход в режим Dedicated и выход из него

Меняют сессионное состояние:

- временные таблицы,
- prepare statement ,
- SET [SESSION]
- set role,
- блокировки ,
- незавершённые транзакции и т.п.

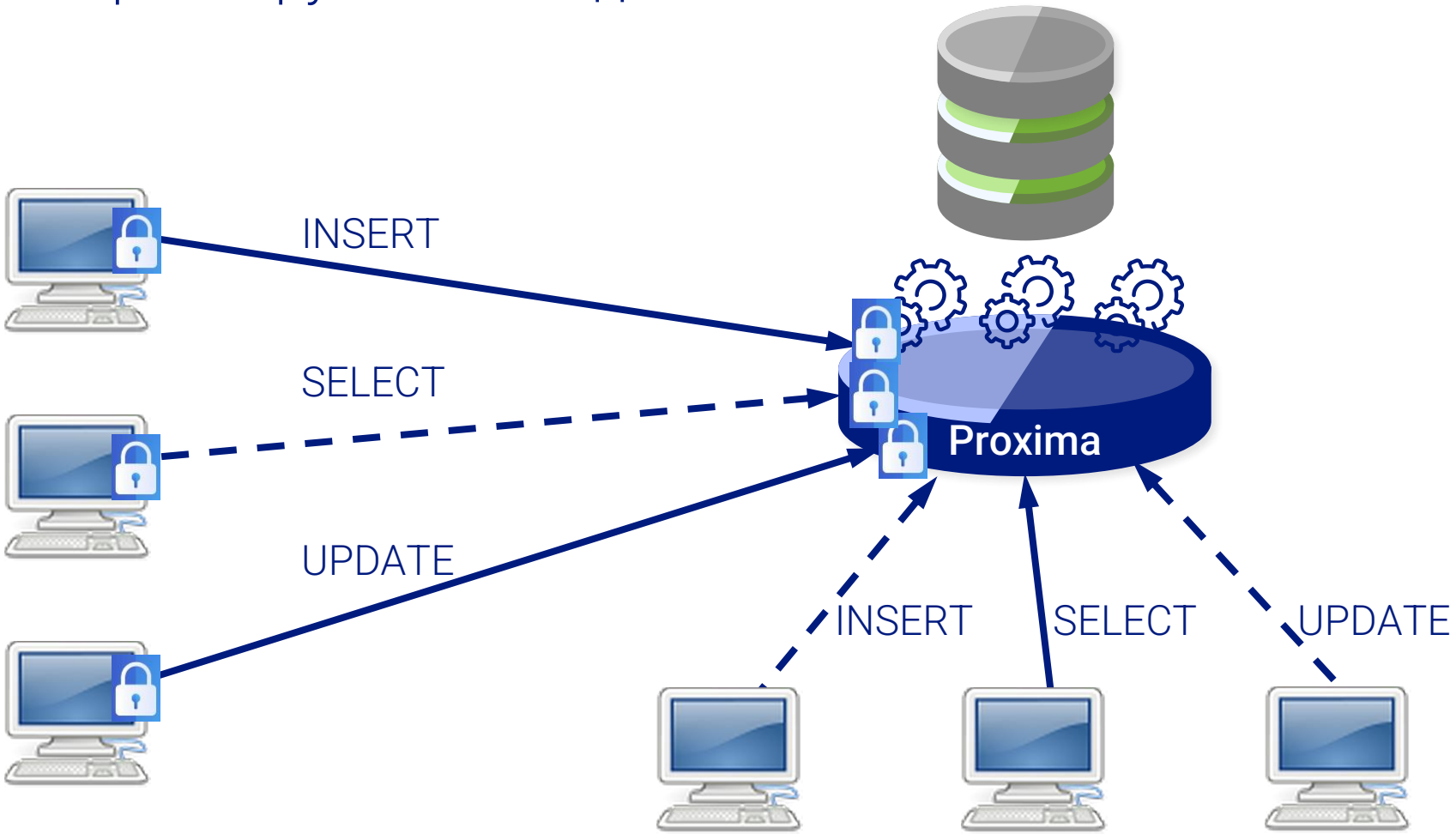
Proxima

Аутентификация пользователя происходит на Proxima



Proxima

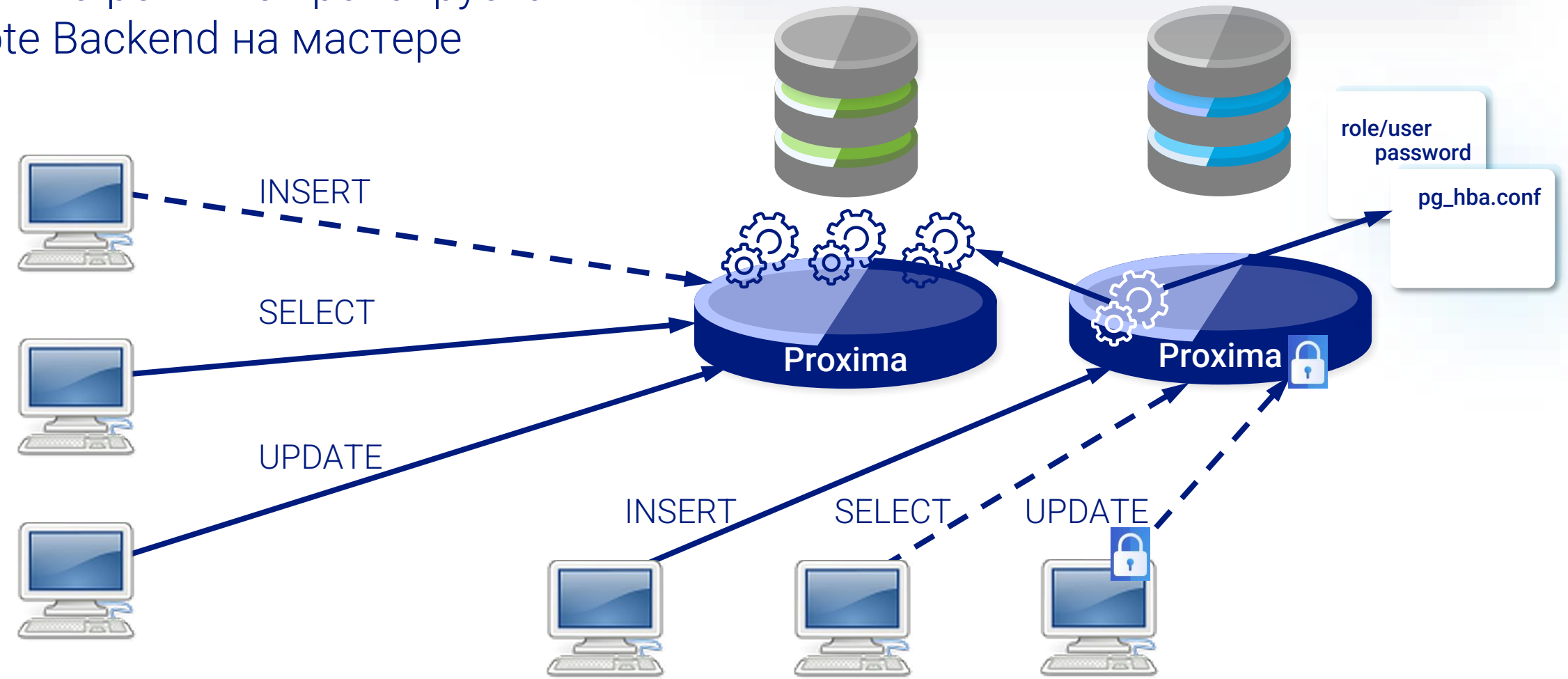
Proxima терминирует SSL соединения



Proxima: проксирование

Клиент на реплике проксируется к Remote Backend на мастере

```
proxima.cluster_mode='guc'
proxima.cluster_config='0,node1,4590,P;1,node2,4590,S;'
```



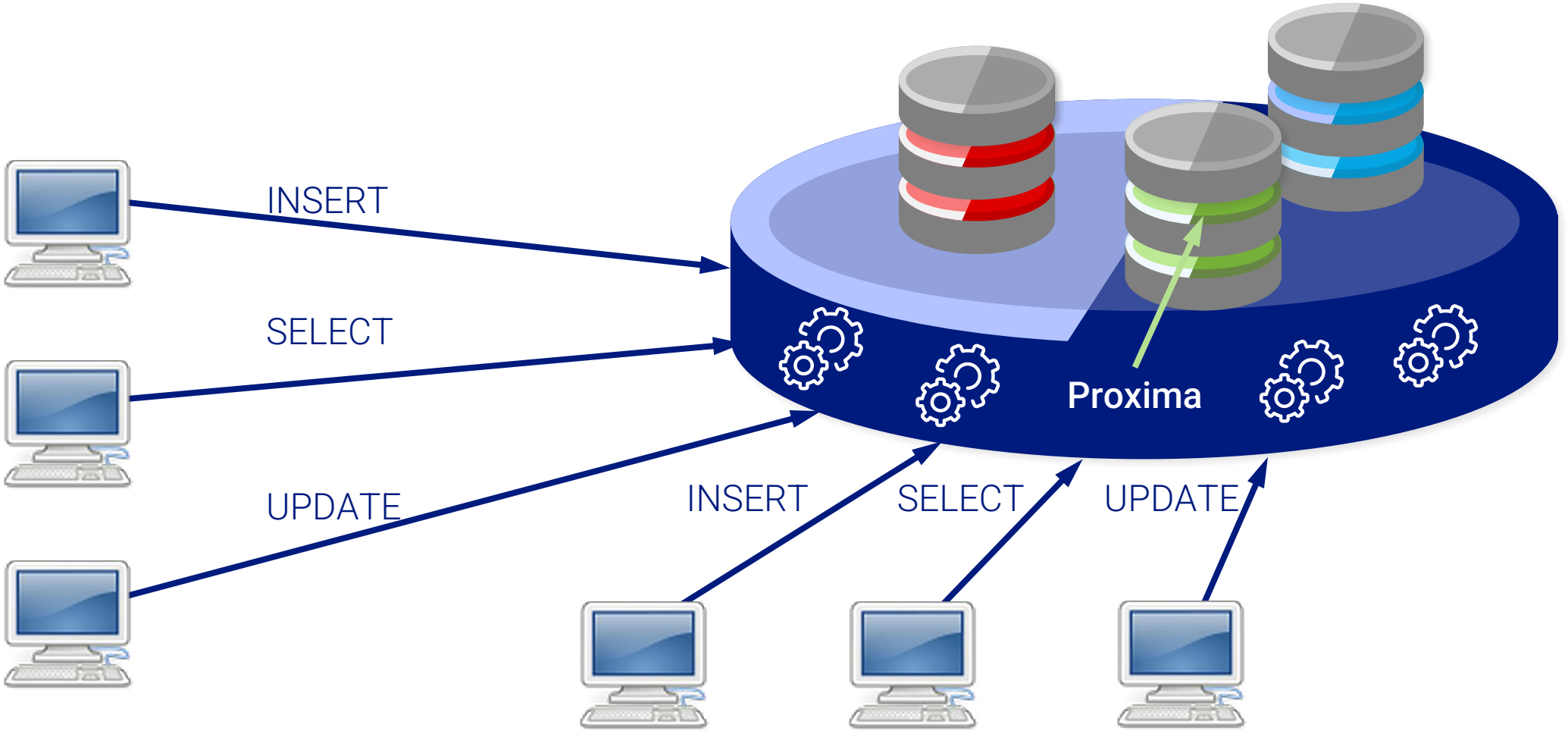
Proxima + BiHA

Любой узел может быть точкой входа в кластер
Определение лидера BiHA происходит на лету

```
postgresql.conf:
shared_preload_libraries = 'proxima'
proxima.cluster_mode = 'biha'
```

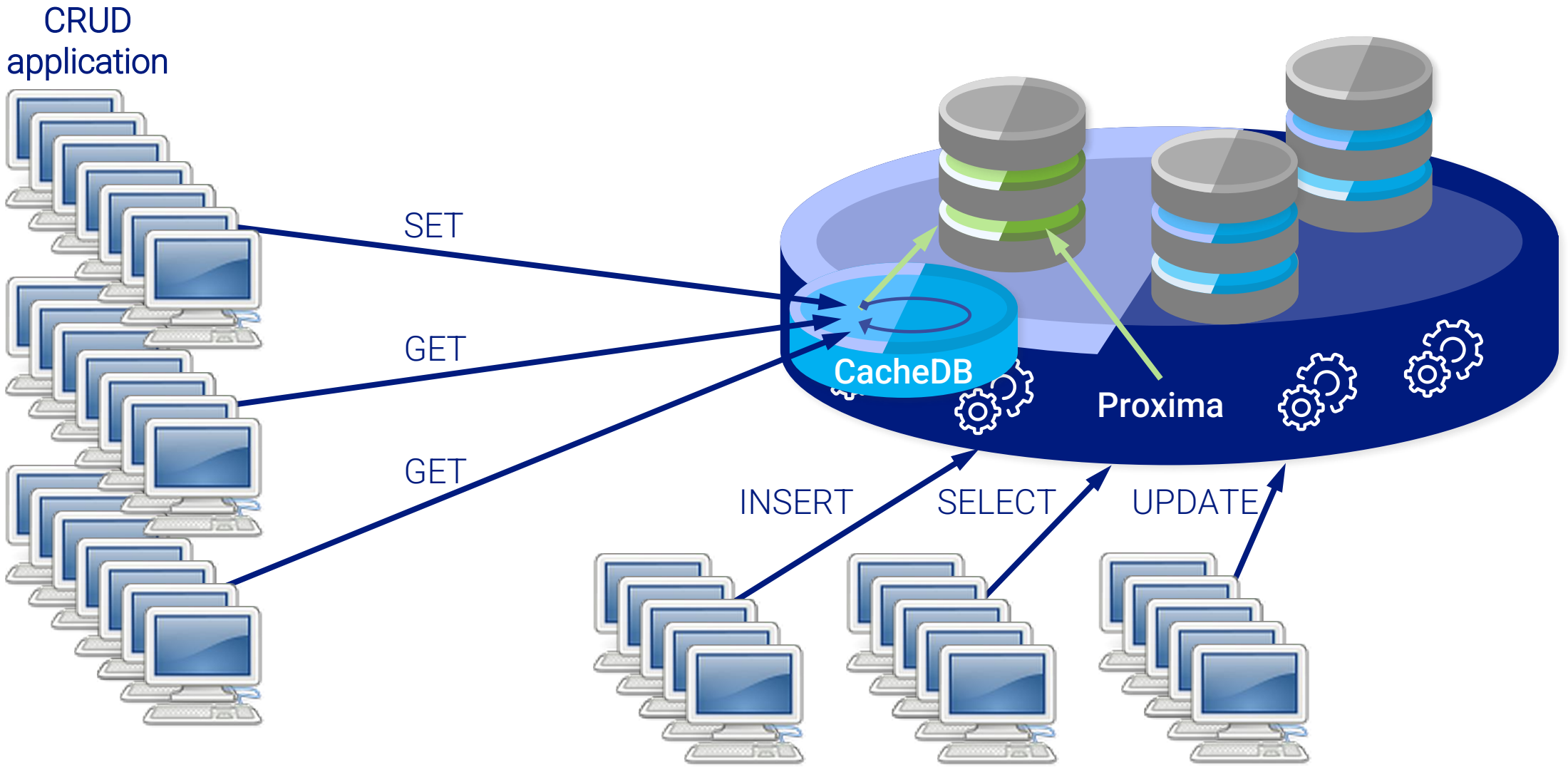
Proxima + BiHA

Proxima перенаправляет соединения на новый лидер BiHA



PostgresPro

Разрабатываем новую технологию - CacheDB



PostgresPro

Спасибо за внимание!