



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Резерв подключений PostgreSQL 18 для мониторинга без superuser

Как мы отделяем агента мониторинга от лимита
`max_connections` без выдачи ему `superuser`

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Приложение под нагрузкой занимает все слоты `max_connections`, и в этот момент мониторинг (Zabbix, скрипты, дашборды) не может подключиться и слепнет — видимость пропадает именно тогда, когда нужнее всего. Выдать агенту `SUPERUSER` — плохое решение: он получит доступ к данным, DDL и настройкам кластера. Нужен резерв слотов и роль с правами только на чтение статистики.

Почему это важно бизнесу

- В момент перегрузки БД пропадает телеметрия — инженеры чинят инцидент вслепую, время простоя растёт
- `SUPERUSER` у сервисного агента — лишняя точка компрометации: утечка пароля мониторинга даёт полный доступ к данным
- Каждое лишнее открытое подключение — это `shared memory` и накладные расходы процесса, а не бесплатный ресурс
- Без раннего триггера по проценту занятых подключений авария по коннектам обнаруживается только когда приложение уже легло

Ключевые параметры реализации

100

max_connections по умолчанию — порог, который одно тяжёлое приложение выдает без пулера

PostgreSQL 18, default

PG 16+

с этой версии есть reserved_connections и роль pg_use_reserved_connections без superuser

PostgreSQL 16, docs

3

слота superuser_reserved_connections по умолчанию — финальный резерв для аварийного входа

PostgreSQL docs, default

5

слотов reserved_connections (по умолчанию 0) закладываем под роль pg_use_reserved_connections

наш стандарт

5.0 с

reserve_pool_timeout в PgBouncer — простой клиента дольше этого времени открывает резервный пул

pgbouncer.org, default

5 мин

idle_in_transaction_session_timeout для сервисных ролей — рвём забытые в транзакции подключения

наш стандарт



Резерв подключений на уровне PostgreSQL

Что настраиваем

Кластер PostgreSQL 18, роль мониторинга без SUPERUSER, конфиг `postgresql.conf`

Как мы это делаем

- 1 В `postgresql.conf` при `max_connections=200` ставим `reserved_connections=5` и `superuser_reserved_connections=3` (по умолчанию 0 и 3)
- 2 Оба параметра ставятся только при старте сервера (PGC_POSTMASTER): нужен `restart`, `ALTER SYSTEM + reload` их не применит
- 3 Создаём роль: `CREATE ROLE zbx_monitor LOGIN CONNECTION LIMIT 5`, затем `GRANT pg_use_reserved_connections, pg_monitor TO zbx_monitor`
- 4 Проверяем членство: `SELECT pg_has_role('zbx_monitor','pg_use_reserved_connections','member')`
- 5 `log_connections='all'` (в PG 18 это список аспектов, а не boolean); отказ по слотам всегда виден как FATAL «sorry, too many clients already»

РЕЗУЛЬТАТ

Мониторинг подключается даже когда приложение выело все обычные слоты — он садится в зарезервированный пул, а не в пул суперпользователя. Резерв `superuser` остаётся неприкосновенным для аварийного входа инженера, а у мониторинга нет прав менять роли, данные или конфигурацию кластера.

КЛЮЧЕВОЙ НЮАНС

`reserved_connections` и `superuser_reserved_connections` применяются только рестартом, а их сумма должна быть меньше `max_connections`. Задаём заранее, а не когда слоты уже кончились.



Пул подключений перед приложением

Что настраиваем

PgBouncer перед PostgreSQL 18, transaction pooling, отдельный пул мониторинга

Как мы это делаем

- 1 pool_mode=transaction, default_pool_size=20 на папу user/db, max_client_conn=100 — сотни клиентов, десятки backend-коннектов
- 2 reserve_pool_size=5 (по умолчанию 0, резерв выключен), reserve_pool_timeout=5.0 — ждущий слот дольше 5 с клиент берёт коннект из резерва
- 3 Мониторинг подключаем отдельной записью в databases со своим pool_size — не делит очередь с трафиком приложения
- 4 На стороне самого PostgreSQL держим reserved_connections как последний барьер, если PgBouncer перегружен или лёг

РЕЗУЛЬТАТ

Даже когда приложение упирается в default_pool_size и клиенты ждут в очереди, у мониторинга свой пул и свои слоты — он не стоит в общей очереди с бизнес-трафиком и продолжает опрашивать pg_stat_activity.

КЛЮЧЕВОЙ НЮАНС

Пулер снижает число реальных backend-подключений к PostgreSQL, но сам может стать точкой отказа: резерв на уровне СУБД нужен даже при живом и настроенном PgBouncer.



Мониторинг без лишних прав: pg_monitor + Zabbix agent 2

Что настраиваем

Шаблон «PostgreSQL by Zabbix agent 2», роль `zbx_monitor`, метрики `pg_stat_activity`

Как мы это делаем

- 1 GRANT pg_monitor TO zbx_monitor — читаем pg_stat_activity, статистику и конфигурацию, без DDL/DML на пользовательские схемы
- 2 Макрос `{$PG.CONN_TOTAL_PCT.MAX.WARN}=90` и триггер `min(/PostgreSQL by Zabbix agent 2/pgsql.connections.sum.total_pct,5m)>90`
- 3 Ключи `pgsql.connections.sum.idle_in_transaction` и `pgsql.connections.sum.waiting` — видим не только число, но и природу занятости
- 4 Ограничиваем `zbx_monitor` через `CONNECTION LIMIT 5`, чтобы сам агент не мог случайно выесть свой же резерв

РЕЗУЛЬТАТ

Один шаблон Zabbix закрывает и раннее предупреждение (90% от `max_connections`), и разбор причины (`idle_in_transaction`, `waiting`) — без выдачи мониторингу прав выше `pg_monitor`.

КЛЮЧЕВОЙ НЮАНС

`pg_monitor` не даёт доступ к данным пользовательских таблиц — это осознанно: мониторинг видит метрики и активность, но не бизнес-данные и не может их менять.



Подводные камни

✗ SUPERUSER мониторингу

Утечка пароля superuser-агента = полный доступ к данным и DDL; заменяем на pg_monitor (pg_read_all_stats/settings) + pg_use_reserved_connections

✗ reload вместо restart

reserved_connections и superuser_reserved_connections — параметры PGC_POSTMASTER: ALTER SYSTEM и reload их не применяют, нужен полный restart сервера

✗ Резерв без роли

Выставили reserved_connections, но не выдали GRANT pg_use_reserved_connections — слоты есть, а подключиться в них кроме superuser никто не может

✗ CONNECTION LIMIT не защищает от чужих

Лимит на роль ограничивает только её саму, а не спасает от голода по коннектам — без reserved_connections он бесполезен при полном max_connections

✗ idle_in_transaction копит слоты

Забытые открытые транзакции держат подключение занятым бесконечно; ставим idle_in_transaction_session_timeout, иначе резерв съедают зависшие сессии

✗ Пулер как единственная защита

PgBouncer снижает число backend-подключений, но при его перегрузке или падении нужен резерв именно в PostgreSQL — закладываем оба уровня, а не один

✗ Нарращивание max_connections вместо анализа

Поднимают лимит вместо разбора, кто держит слоты; каждое лишнее подключение — это shared memory и накладные расходы процесса, а не бесплатный ресурс

✗ Забыли про лимит самой базы

Кроме лимита роли есть datconlimit у базы (ALTER DATABASE ... CONNECTION LIMIT) — не проверив его, легко упереться в чужой потолок

Как правильно

МИНИМУМ

- Роль мониторинга через GRANT pg_monitor, без superuser
- superuser_reserved_connections не трогаем — оставляем 3 (default)
- CONNECTION LIMIT на роль мониторинга, например 5

НОРМАЛЬНО

- + reserved_connections=5 и GRANT pg_use_reserved_connections роли мониторинга
- + idle_in_transaction_session_timeout для сервисных и прикладных ролей
- + триггер Zabbix на `{$PG.CONN_TOTAL_PCT.MAX.WARN}=90` по total_pct

ХОРОШО

- + PgBouncer с отдельным пулом мониторинга и reserve_pool_size > 0
- + отдельные алерты на idle_in_transaction и waiting, а не только на total
- + проверка PGC_POSTMASTER-параметров резерва после каждого мажорного апгрейда



Чек-лист самопроверки

☐ reserved_connections и superuser_reserved_connections выставлены и применены рестартом, а не reload?

☐ Роли мониторинга выдан pg_use_reserved_connections, а не superuser?

☐ У роли мониторинга задан CONNECTION LIMIT?

☐ idle_in_transaction_session_timeout настроен для сервисных ролей приложения?

☐ В Zabbix настроен триггер на pgsql.connections.sum.total_pct с порогом 90%?

☐ Проверена datconnlimit целевой базы, а не только лимит роли?

☐ Если используется PgBouncer — задан reserve_pool_size больше нуля?

☐ Мониторингу выдан pg_monitor, а не прямой доступ к пользовательским таблицам?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Настраиваем резерв подключений (reserved_connections) и роль мониторинга без superuser под ключ
- Разворачиваем и тюним PgBouncer с отдельным пулом под мониторинг и резервным пулом на пиковую нагрузку
- Подключаем шаблон «PostgreSQL by Zabbix agent 2» с триггерами по проценту занятых подключений и idle_in_transaction
- Аудит прав ролей PostgreSQL: убираем избыточный superuser, переводим на pg_monitor и pg_use_reserved_connections
- Планируем окна restart для применения PGC_POSTMASTER-параметров без внеплановых простоев

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** Connections and Authentication — reserved_connections (postgresql.org — 18)
- 02** Release 16 — reserved_connections, pg_use_reserved_connections (postgresql.org — 16)
- 03** Predefined Roles — pg_monitor (postgresql.org — 18)
- 04** CREATE ROLE — CONNECTION LIMIT (postgresql.org — 18)
- 05** The Cumulative Statistics System — pg_stat_activity (postgresql.org — 18)
- 06** Client Connection Defaults — idle_in_transaction_session_timeout (postgresql.org — 18)
- 07** PgBouncer configuration — reserve_pool_size, reserve_pool_timeout (pgbouncer.org — 1.24)
- 08** PostgreSQL by Zabbix agent 2 — макросы и триггеры (zabbix.com — 7.0)

