



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

1С на PostgreSQL: клиент-сервер, тюнинг и бэкап без потерь

Наш стандарт: сервер 1С 8.3.27 + PostgreSQL 17, тюнинг
СУБД, pgBackRest, рас и мониторинг

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Файловая база 1С на 15–50 рабочих мест упирается в блокировки: документы проводятся по минуте, закрытие месяца растягивается на ночь, а «бэкапом» служит копия папки или выгрузка .dt. Один сбой диска — и учёт откатывается на сутки или теряется целиком. Мы переводим такие базы на клиент-сервер с PostgreSQL, тюним СУБД под нагрузку 1С и строим проверяемый контур восстановления.

Почему это важно бизнесу

- Час простоя 1С останавливает продажи, отгрузки и бухгалтерию всей компании сразу
- Потеря базы без WAL-архива — это потеря дней учёта и ручное восстановление документов
- Тормоза 1С — это оплаченные часы сотрудников, потраченные на ожидание отчётов
- Штрафы за срыв сроков отчётности, если база упала в отчётный период
- Лицензии MS SQL дороже: PostgreSQL закрывает ту же задачу без лицензионных платежей

Ключевые параметры реализации

8.3.27

Актуальная платформа: сервер x64, кластер с лимитами памяти rphost, RAS для управления

по докам 1C (v8.1c.ru)

17.x

PostgreSQL только в сборке 1C или Postgres Pro — с патчами mchar/mvarchar, ванильный не ставим

по докам 1C / postgrespro.ru

25%

shared_buffers от RAM выделенного сервера СУБД, effective_cache_size — 75%

наш стандарт по докам PostgreSQL

0

max_parallel_workers_per_gather: параллелизм на OLTP-запросах 1C даёт регрессии планов

по докам 1C

2.59

pgBackRest: full/diff/incr, сжатие zst, непрерывный архив WAL — восстановление на момент време...

pgbackrest.org

300 с

archive_timeout: сегмент WAL уходит в репозиторий не реже 5 минут — это наш фактический RPO

наш стандарт



Перевод 1С с файловой базы на клиент-сервер PostgreSQL

Что настраиваем

Сервер 1С + узел СУБД (одна или две VM), офис до 50 рабочих мест

Как мы это делаем

- 1 Разворачиваем сервер 1С 8.3.27 x64: агент ragent на 1540, менеджер кластера rmngr на 1541, rphost на 1560-1591, служба RAS на 1545 для управления через ras
- 2 Ставим PostgreSQL 17 из сборки 1С: initdb с локалью ru_RU.UTF-8, каталог данных и pg_wal — на отдельные диски
- 3 Тюним postgresql.conf: shared_buffers 25% RAM, work_mem 64-128 МБ, maintenance_work_mem 1 ГБ, max_parallel_workers_per_gather=0
- 4 Перенастраиваем автовакуум: autovacuum_max_workers=4, autovacuum_naptime=20s, autovacuum_vacuum_scale_factor 0.01-0.05 — иначе пухнут регистры итогов
- 5 Переносим базу через .dt в нерабочее окно, замеряем APDEX и время ключевых операций до и после

РЕЗУЛЬТАТ

Уходят блокировки файлового движка: документы проводятся за секунды, отчёты не валят базу, число одновременных пользователей больше не ограничено файловым режимом. СУБД получает предсказуемый профиль памяти и диска.

КЛЮЧЕВОЙ НЮАНС

Работа 1С поддерживается только на сборках PostgreSQL с патчами 1С (типы mchar/mvarchar) или на Postgres Pro — сверяем версию по таблице совместимости системных требований платформы.



Резервное копирование и PITR на pgBackRest

Что настраиваем

Узел СУБД + выделенный репозиторий бэкапов вне сервера

Как мы это делаем

- 1 Включаем непрерывный архив: `archive_mode=on`,
`archive_command='pgbackrest --stanza=db1c archive-push %p'`,
`archive_timeout=300`
- 2 Создаём stanza и расписание: full по воскресеньям, diff ежедневно,
incr каждые 4 часа; `repo1-retention-full=2`, `compress-type=zst`,
`process-max=4`
- 3 Репозиторий выносим на отдельный хост или S3 (`repo1-type=s3`),
включаем шифрование `repo1-cipher-type=aes-256-cbc`
- 4 Ежемесячный тестовый restore на стенд: `pgbackrest --delta restore`
с recovery до метки времени, контроль — открытие базы клиентом
1C
- 5 `pgbackrest check` и `verify` в cron, падение любого задания — алерт в
Telegram

РЕЗУЛЬТАТ

RPO не хуже 5 минут и восстановление на произвольный момент: ошибочное групповое перепроведение или падение диска откатываются к точке до инцидента. Время восстановления известно заранее, потому что замерено на стенде.

КЛЮЧЕВОЙ НЮАНС

Выгрузка .dt — не резервная копия: на повреждённой базе она просто не выполнится. Бэкапим средствами СУБД, а .dt используем только для переноса; критерий живости бэкапа — успешный restore, а не зелёный лог.



Эксплуатация кластера: рас, техножурнал, регламенты СУБД

Что настраиваем

Кластер серверов 1C + PostgreSQL, регулярное обслуживание и мониторинг

Как мы это делаем

- 1 Поднимаем RAS (ras cluster --daemon, порт 1545); rac session list и rac process list — в мониторинг: сессии, память rphost, время вызовов
- 2 В консоли кластера задаём предел памяти рабочих процессов и ночной перезапуск rphost — гасим утечки памяти длинных сеансов
- 3 Включаем техножурнал logcfg.xml: события EXCP, TIMEOUT, TDEADLOCK, DBPOSTGRS с отбором по duration — ловим долгие запросы и блокировки
- 4 Регламенты СУБД: еженедельный vacuumdb --analyze, REINDEX CONCURRENTLY по топ-таблицам, контроль bloat по n_dead_tup/n_live_tup в pg_stat_user_tables
- 5 Пороги в Zabbix: число сессий, память rphost, lag архива WAL, время отклика тестовой операции

РЕЗУЛЬТАТ

Проблемы видны до звонка клиента: рост памяти rphost, долгие запросы и невыгруженные WAL ловятся алертами. Разбор «почему вчера тормозило» занимает минуты — по техножурналу, а не по ощущениям пользователей.

КЛЮЧЕВОЙ НЮАНС

Без лимита памяти rphost сервер деградирует незаметно: процесс распухает неделями и уводит хост в своп. Лимит плюс плановый перезапуск — дешёвая страховка, встроенная в саму платформу.



Подводные камни

✗ Ванильный PostgreSQL без патчей 1C

Платформе нужны патчи 1C (mchar/mvarchar, fasttrun). Ставим сборку с releases.1c.ru или Postgres Pro — иначе ошибки типов и неподдерживаемая связка.

✗ Выгрузка .dt как резервная копия

На битой базе выгрузка падает, а делается она только эксклюзивно. Бэкап — это pgBackRest/pg_dump плюс регулярный тестовый restore на стенде.

✗ Автовакуум с настройками по умолчанию

autovacuum_vacuum_scale_factor 0.2 запускает vacuum слишком поздно — регистры итогов пухнут, запросы деградируют. Ужимаем до 0.01–0.05 и добавляем wo...

✗ synchronous_commit=off на боеу

Даёт прирост записи, но при сбое питания теряет последние транзакции. Отключаем только на время загрузки .dt, на проде держим on.

✗ rphost без лимитов памяти

Утечки соединений и сеансов раздувают процесс до свопа. Задаём предел памяти в кластере и ночной перезапуск рабочих процессов.

✗ Включённый параллелизм запросов

Параллельные планы на коротких OLTP-запросах 1C дают нестабильное время. Ставим max_parallel_workers_per_gather=0 по рекомендации 1C.

✗ Данные, WAL и бэкап на одном томе

Смерть одного диска забирает и базу, и копии. Разносим pg_wal и данные, репозиторий pgBackRest — только на другой хост или S3.

✗ Тяжёлые отчёты по боевой базе

Аналитика за годы блокирует OLTP-контур в рабочее время. Отчёты 1C — на копию базы (механизм копий БД, КОРП) или в ночное окно; SQL/BI-выгрузки — на...

Как правильно

МИНИМУМ

- Клиент-сервер: 1С 8.3.27 x64 + PostgreSQL из сборки 1С вместо файловой базы
- Ночной pg_dump с копией на другой хост, хранение минимум 14 дней
- Лимит памяти rghost в консоли кластера и ночной перезапуск процессов

НОРМАЛЬНО

- pgBackRest: WAL-архив и расписание full/diff/incr, RPO не хуже 5 минут
- Тюнинг postgresql.conf по профилю: буферы, work_mem, автовакуум под 1С
- Мониторинг rac/RAS + Zabbix: сессии, память rghost, время отклика
- Ежеквартальный тестовый restore на стенде с замером времени

ХОРОШО

- Patroni 4.x: streaming-реплика и автоматический failover СУБД
- Отдельный контур под отчёты: механизм копий БД (КОРП) или streaming-реплика для SQL/...
- Техножурнал EXCP/TIMEOUT/TDEADLOCK и разбор долгих запросов
- Шифрованный офсайт-репозиторий бэкапов (S3, aes-256-cbc)



Чек-лист самопроверки

- | | |
|--|--|
| ☐ База 1С работает в клиент-серверном варианте, а не в файловом при 10+ пользователях? | ☐ Заданы лимиты памяти ghost и плановый перезапуск рабочих процессов кластера? |
| ☐ СУБД — сборка PostgreSQL с патчами 1С или Postgres Pro, версия сверена с таблицей совместимости? | ☐ Есть мониторинг сессий и времени отклика кластера 1С (rac/RAS + Zabbix) с алертами? |
| ☐ Настроен непрерывный архив WAL и известен фактический RPO в минутах? | ☐ Техножурнал включён хотя бы на ошибки и долгие запросы (EXCP, TIMEOUT)? |
| ☐ Восстановление проверялось на стенде за последний квартал, время restore замерено? | ☐ Репозиторий бэкапов лежит вне сервера СУБД и зашифрован? |
| ☐ Автовакуум перенастроен под 1С (autovacuum_vacuum_scale_factor 0.01-0.05, workers), bloat контролируется? | ☐ Тяжёлые отчёты вынесены с боевой базы — на копию БД/реплику или в ночное окно? |

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).

- Переводим 1С с файловой базы на клиент-сервер PostgreSQL под ключ: перенос, тюнинг, замеры до/после
- Внедряем pgBackRest: WAL-архив, расписание копий, шифрованный репозиторий, регламент тестовых restore
- Строим мониторинг кластера 1С: RAS/rac, Zabbix, техножурнал, алерты в Telegram
- Настраиваем отказоустойчивость: Patroni, streaming-реплики, реплика под отчёты
- Сопровождаем СУБД по регламенту: автовакуум, реиндексация, контроль bloat и деградации дисков

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** Системные требования «1С:Предприятия 8»: СУБД PostgreSQL (v8.1c.ru — 8.3.27)
- 02** Клиент-серверный вариант. Руководство администратора (rac/RAS, кластер) (its.1c.ru — 8.3.27)
- 03** PostgreSQL Documentation: Server Configuration, Routine Vacuuming, WAL (postgresql.org — 17)
- 04** pgBackRest User Guide (pgbackrest.org — 2.59)
- 05** Patroni Documentation (patroni.readthedocs.io — 4.1)
- 06** Шаблон ITfresh: postgresql.conf для 1С (профили 32/64/128 ГБ RAM) (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

