Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Оптимизация PostgreSQL под 1С:Предприятие 8.3: наша методика

Эталонный postgresql.conf, autovacuum под регистры, диагностика через pg_stat_statements

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

База 1С на PostgreSQL с дефолтным конфигом деградирует по мере роста: проведение документов растягивается до секунд, отчёты СКД висят минутами, к закрытию месяца сервер «встаёт». Профиль нагрузки 1С (временные таблицы, write-load регистров, длинные транзакции) не совпадает с дефолтами СУБД. Мы приводим связку к эталонному профилю: сборка, память, WAL, autovacuum, диагностика.

Почему это важно бизнесу

- Проведение документа 3–5 с вместо 0,3 с — минус часы работы операторов ежедневно; при 30 РМ это заметные потери ФОТ
- Отчёт за период, который висит 10 минут, блокирует и бухгалтерию, и склад именно в момент закрытия месяца
- Деградация незаметна: dead tuples и bloat копятся месяцами, а «встаёт» база в самый пиковый день
- Апгрейд железа без тюнинга СУБД не помогает: узкое место — конфигурация; деньги на RAM/CPU уходят впустую

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

14–17

ветки актуальных сборок PostgreSQL с патчами 1С для платформы 8.3; в прод ставим ветку 16

по докам v8.1c.ru

25%

shared_buffers от RAM сервера (при 32 ГБ — 8GB), под этот объём включаем huge_pages

наш стандарт / доки PostgreSQL

1000

max_connections: кластер 1С держит пул соединений на каждый rphost; рекомендация Postgres Pro…

по докам Postgres Pro

256

max_locks_per_transaction — иначе реструктуризация падает с «out of shared memory»

по докам Postgres Pro

2%

autovacuum_vacuum_scale_factor вместо дефолтных 20% — регистры 1С копят dead tuples быстрее

наш стандарт

1.1

random_page_cost на SSD/NVMe вместо 4.0 — планировщик перестаёт избегать индексного доступа

наш стандарт для SSD

РЕАЛИЗАЦИЯ 1 ИЗ 3

Эталонный postgresql.conf под профиль нагрузки 1С

Что настраиваем

Кластер 1С 8.3 + отдельная ВМ PostgreSQL 16.x (сборка 1С), 32–64 ГБ RAM, NVMe; клиенты 20–50 РМ

Как мы это делаем

1. Память: shared_buffers=25% RAM, effective_cache_size=75%, temp_buffers=256MB (временные таблицы 1С), work_mem=64–256MB с расчётом на пиковое число сессий
2. Обязательные для 1С: standard_conforming_strings=off, escape_string_warning=off, max_locks_per_transaction=256, max_connections=1000, локаль ru_RU.UTF-8
3. WAL: max_wal_size=4–8GB, checkpoint_completion_target=0.9, checkpoint_timeout=15min, wal_compression=on; synchronous_commit не отключаем
4. Планировщик: random_page_cost=1.1 (NVMe), default_statistics_target=200; модули сборки 1С — online_analyze (table_type=temporary) и plantuner
5. ОС: vm.swappiness=1, vm.dirty_background_ratio=3, huge pages по размеру shared_buffers, I/O-scheduler none для NVMe

РЕЗУЛЬТАТ

Проведение документов ускоряется в 3–10 раз против дефолтного конфига, пики I/O на checkpoint сглаживаются, отчёты СКД перестают ронять сервер по памяти. Профиль тиражируем: три шаблона под 32/64/128 ГБ RAM, развёртывание на новом сервере — 15 минут.

КЛЮЧЕВОЙ НЮАНС

work_mem выделяется на каждую операцию сортировки/хеша, а не на сессию: 100 сессий × 256MB кладут сервер в OOM. Считаем от пиковых сессий, для тяжёлых регламентов поднимаем work_mem точечно в сеансе.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Autovacuum и регламентное обслуживание регистров

Что настраиваем

Горячие таблицы _accumrg*/_inforg* (регистры накопления и сведений) продуктивной базы

Как мы это делаем

1. Глобально: autovacuum_max_workers=6, vacuum_threshold=50, vacuum_scale_factor=0.02, analyze_scale_factor=0.01, cost_delay=2ms, cost_limit=1000
2. Горячие регистры находим по pg_stat_user_tables (n_dead_tup, dead_pct) и задаём персонально: ALTER TABLE ... SET (autovacuum_vacuum_scale_factor=0.005)
3. Еженедельно в окне: REINDEX DATABASE CONCURRENTLY (PostgreSQL 12+ — без блокировки записи) + ANALYZE; bloat индексов контролируем pgstattuple, порог 30%
4. После загрузки .dt и реструктуризации — обязательный REINDEX + ANALYZE, плюс «Тестирование и исправление» в конфигураторе на остановленной базе

РЕЗУЛЬТАТ

Bloat регистров держится ниже 30% вместо 200–300% на дефолте, планы запросов стабильны за счёт свежей статистики, база не «встаёт» в день закрытия месяца. Размер базы на диске перестаёт расти в 2–3 раза быстрее полезных данных.

КЛЮЧЕВОЙ НЮАНС

Дефолтный scale_factor 20% означает: регистр на 10 млн строк чистится после 2 млн мёртвых версий. Для 1С это катастрофа — порог снижаем на порядок и никогда не выключаем autovacuum «ради скорости».

РЕАЛИЗАЦИЯ 3 ИЗ 3

Диагностика: pg_stat_statements + техжурнал 1С

Что настраиваем

Связка СУБД-метрик и технологического журнала на рабочем сервере 1С (rphost); вывод в Zabbix

Как мы это делаем

1. shared_preload_libraries='pg_stat_statements', track=all, max=10000; log_min_duration_statement=5000 — все запросы дольше 5 с попадают в лог
2. На сервере 1С кладём logcfg.xml: событие DBPOSTGRS с фильтром duration≥5000000 мкс — видим, какой вызов кода 1С породил медленный SQL
3. Топ тяжёлых запросов снимаем по total_exec_time из pg_stat_statements, планы разбираем через EXPLAIN (ANALYZE, BUFFERS)
4. Блокировки ловим готовым запросом по pg_locks + pg_stat_activity (blocked/blocking pid); метрики и dead_pct отдаём в Zabbix с триггерами

РЕЗУЛЬТАТ

Разбор жалобы «1С тормозит» занимает минуты, а не дни: сразу видно, СУБД это, код конфигурации или блокировки. Проблемные отчёты чиним адресно — индексом CREATE INDEX CONCURRENTLY по фильтруемому полю, а не «перезагрузкой сервера».

КЛЮЧЕВОЙ НЮАНС

Смотреть только со стороны СУБД мало: pg_stat_statements покажет SQL, но не место в конфигурации. Связка с техжурналом (DBPOSTGRS + контекст) даёт строку модуля 1С — это и есть точка исправления.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Ванильный PostgreSQL вместо сборки 1С**
  Без патчей 1С деградируют временные таблицы и LIKE по кириллице. Ставим только сборку с releases.1c.ru или Postgres Pro (сертиф. под 1С).

- ✕ **work_mem без расчёта на сессии**
  Параметр выделяется на каждую сортировку/хеш: 100 сессий × 256MB = OOM. Считаем от пиковых сессий, поднимаем точечно в сеансе регламента.

- ✕ **Дефолтный autovacuum на регистрах**
  scale_factor 20% для больших таблиц — чистка «раз в никогда». Снижаем до 2%, горячим регистрам — 0.5% через ALTER TABLE ... SET.

- ✕ **synchronous_commit=off «для скорости»**
  При сбое питания теряются уже проведённые документы — для учётной системы недопустимо. Ускоряем WAL быстрым диском, а не отключением гарантий.

- ✕ **random_page_cost=4.0 на SSD**
  Планировщик считает индексный доступ дорогим и уходит в seq scan. На SSD/NVMe ставим 1.1 — планы отчётов СКД меняются кардинально.

- ✕ **Нет REINDEX/ANALYZE после загрузки .dt**
  После миграции с MS SQL статистики нет — планировщик слеп, база «тормозит с первого дня». Перенос всегда завершаем REINDEX + ANALYZE.

- ✕ **max_locks_per_transaction = 64**
  Реструктуризация и обновление конфигурации падают с «out of shared memory». По докам Postgres Pro для 1С ставим 256.

- ✕ **WAL и данные на одном медленном диске**
  Checkpoint-пики I/O бьют по проведению документов. Разносим WAL на отдельный NVMe и растягиваем checkpoint (completion_target=0.9).

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Сборка PostgreSQL от 1С / Postgres Pro ветки 16, а не ванильная
- shared_buffers=25% RAM, temp_buffers=256MB, effective_cache_size=75%
- standard_conforming_strings=off, max_locks_per_transaction=256
- Ночной pg_dump + тестовое восстановление раз в квартал

НОРМАЛЬНО

- Autovacuum: scale_factor 2%/1%, 6 воркеров, персонально для горячих регистров
- NVMe под данные, random_page_cost=1.1, checkpoint_completion_target=0.9
- pg_stat_statements + log_min_duration_statement=5000 включены постоянно
- Еженедельный REINDEX CONCURRENTLY + ANALYZE в окне обслуживания

ХОРОШО

- Huge pages под shared_buffers, WAL на отдельном NVMe, тюнинг sysctl
- Техжурнал 1С (DBPOSTGRS) + метрики СУБД в Zabbix с триггерами
- Стриминг-реплика + pg_probackup с PITR вместо голого pg_dump
- Профили конфига под 32/64/128 ГБ RAM, ревизия параметров раз в квартал

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Стоит ли сборка PostgreSQL с патчами 1С (releases.1c.ru / Postgres Pro), а не ванильная?
- [ ] Рассчитан ли work_mem от пикового числа сессий, а не скопирован из чужого конфига?
- [ ] Снижен ли autovacuum_vacuum_scale_factor до 2% и заданы ли параметры горячим регистрам?
- [ ] Включены ли pg_stat_statements и логирование запросов дольше 5 секунд?
- [ ] Настроен ли техжурнал 1С (logcfg.xml, событие DBPOSTGRS) для связи SQL с кодом конфигурации?
- [ ] Есть ли регламент REINDEX CONCURRENTLY + ANALYZE и контроль bloat с порогом 30%?
- [ ] Лежат ли данные и WAL на SSD/NVMe и снижен ли random_page_cost до 1.1?
- [ ] Не отключены ли synchronous_commit и fsync ради скорости?
- [ ] Выполняются ли REINDEX и ANALYZE после каждой загрузки .dt и реструктуризации?
- [ ] Проверяется ли восстановление бэкапа на тестовом сервере, а не только его наличие?

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

ЧТО МЫ ДЕЛАЕМ

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).

- Аудит связки 1С+PostgreSQL: конфиг, bloat, топ медленных запросов — отчёт с планом работ
- Внедрение эталонного профиля postgresql.conf + sysctl под ваш объём RAM и диски
- Миграция базы 1С с MS SQL на PostgreSQL через .dt с пост-обработкой и сверкой
- Регламентное обслуживание: autovacuum, REINDEX, бэкапы pg_probackup, мониторинг Zabbix
- Разбор «тормозов» по техжурналу и pg_stat_statements с точечными индексами

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

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Системные требования «1С:Предприятия 8» — СУБД PostgreSQL (v8.1c.ru — 2026)

02 Приложение «Настройка Postgres Pro для решений 1С» (postgrespro.ru — верс. 16)

03 Server Configuration: Resource Consumption, WAL (postgresql.org — верс. 16)

04 Routine Vacuuming (autovacuum) (postgresql.org — верс. 16)

05 Технологический журнал — руководство администратора (its.1c.ru — верс. 8.3)

06 Наш шаблон postgresql.conf: профили 32/64/128 ГБ RAM (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.