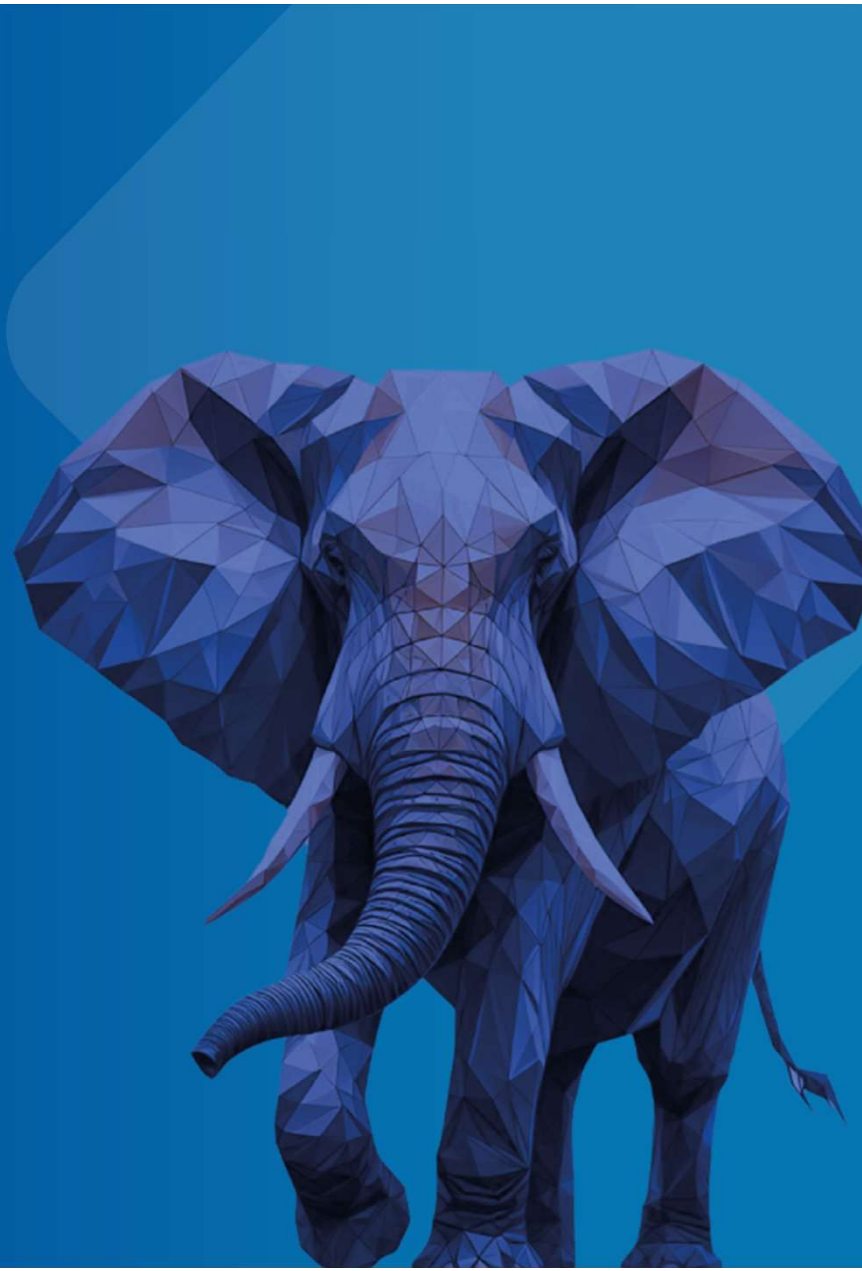
PostgresPro

Новые возможности СУБД Postgres Pro Enterprise 17



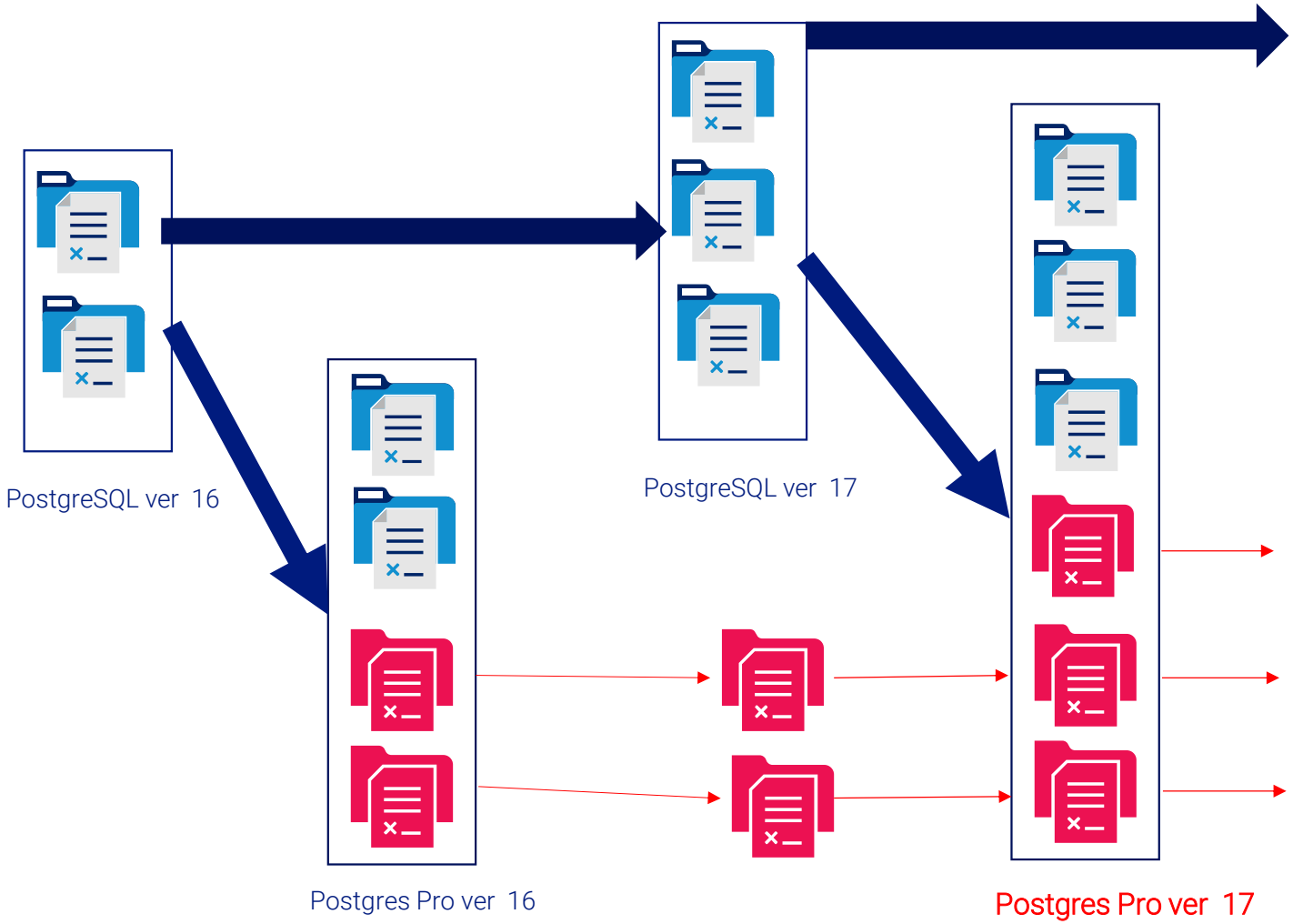
Марк Ривкин,

28 января 2025 года

m.rivkin@postgrespro.ru

PGProDay

Слияние версий PostgreSQL и Postgres Pro Enterprise (версии 15, 16, 17 ….)



Источники новых возможностей

- PostgreSQL 17

- Postgres Pro Enterprise 16+
  - Суперфайлы
  - Планы управления ресурсами в RM
  - Ora2PgPro – видимость функций пакета, AST — абстрактное синтаксическое дерево разбора
  - Probackup 3.0
- Разработки 2024 года

- Плашка 17.2

Продолжается улучшение (MPP)

- CITUS
  - Для OLAP
  - Бесплатное чужое расширение
  - Поколоночное хранение
  - Шардирование с координатором и без

- Shardman
  - Для OLTP
  - Документация
  - Размер БД = 130+ Tb , количество шардов ~ 20 + реплики, 56K users
  - Переход на 17 Enterprise в 2 этапа (2025)
  - BiHA, PPEM (2025)

Новые возможности СУБД Postgres Pro Enterprise 17 (30+)

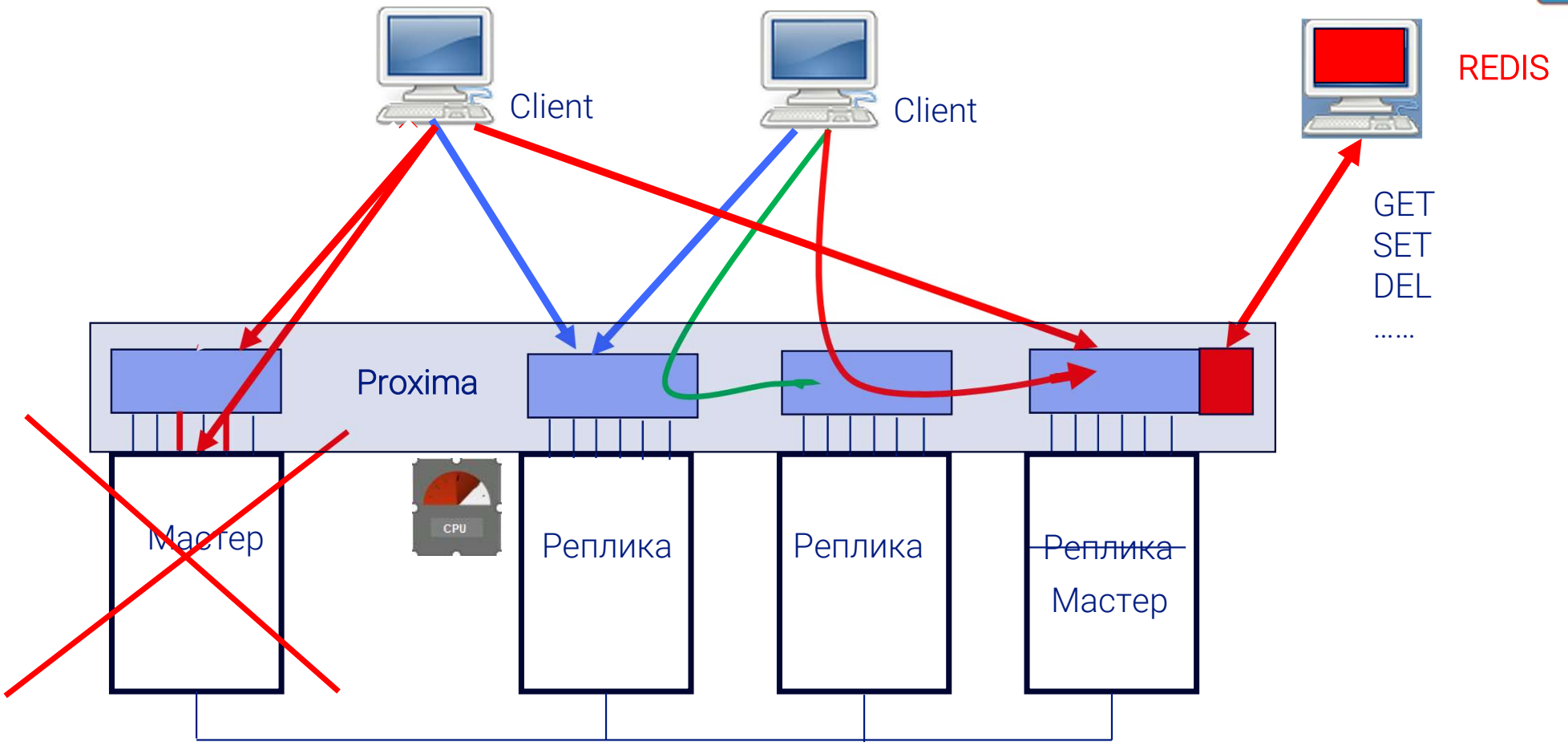
Масштабируемость и производительность
Proxima (pooler, proxy, LB) + REDIS
План в виде Hints — переносим на новые версии
Sr_plan (multiplan (outline)) для выбранного SQL (Ассистент), Шаблоны (Wild Card)
AQE (триггеры), AQO (перенос на реплику, непрерывное обучение)
Шардирование — Shardman, CITUS
High Availability
Referee в BiHA, удаленная реплика
Pg_probackup 3
Упрощенный Restore отдельной БД в новый кластер
Безопасность
Поиск избыточных привилегий
Защитное преобразование (TDE)
pg_proaudit 2.0
Разведка данных
Разработка
Улучшение автоматического Interval partitioning к PostgreSQL, reference partitioning
Очереди сообщений
Глобальные индексы
Векторная БД
Миграция с Oracle
Утилита pgpro_validate, ProGate (proSync+proCopy)
Управляемость и администрирование
PPEM — новые версии
Information Lifecycle management (ILM)
DB для OLTP и 1C — пресеты
Автонастройка параметров при установке, утилита pgpro_tune

PostgresPro

Производительность и масштабируемость

PROXIMA = Pooler + Proxy + Load Balancer

17.2



Эмуляция REDIS

- REDIS — Remote DIctionary Server
- СУБД ключ-значение в памяти. Доступ по ключу
- Быстрая выборка значений по ключу
- Изменение/удаление строк
- Инвалидация при изменении в БД (когерентный кэш)
- Подмножество команд REDIS
  - SET — создать, изменить объект
  - GET — прочитать объект по ключу
  - GETSET — изменить и вернуть старое значение
  - DEL — удалить поле
  - .....

Преимущества Proxima

- Нет дублирования
  - Единый разбор протокола
  - Единая аутентификация
  - Работа с SSL
- Proxy
  - Любой узел может быть точкой входа в кластер
  - Простое проксирование 1к1, без анализа трафика
  - Поддержка определения лидера на лету
- Pooler 2025
  - Для некоторых сессий и транзакций может потребоваться работа в dedicated режиме (через 1 backend)
  - Функциональности Postgres, которые меняют сессионное состояние:

    временные таблицы, prepared statement, SET [SESSION], set role, блокировки, незавершённые транзакции и т.п.
  - В рамках одной клиентской сессии может автоматически происходить переход в режим dedicated и выход из него
- Load balancer, Redis (cacheDB) — 2025 год

PostgresPro

Высокая надежность

Встроенный отказоустойчивый кластер BiHA

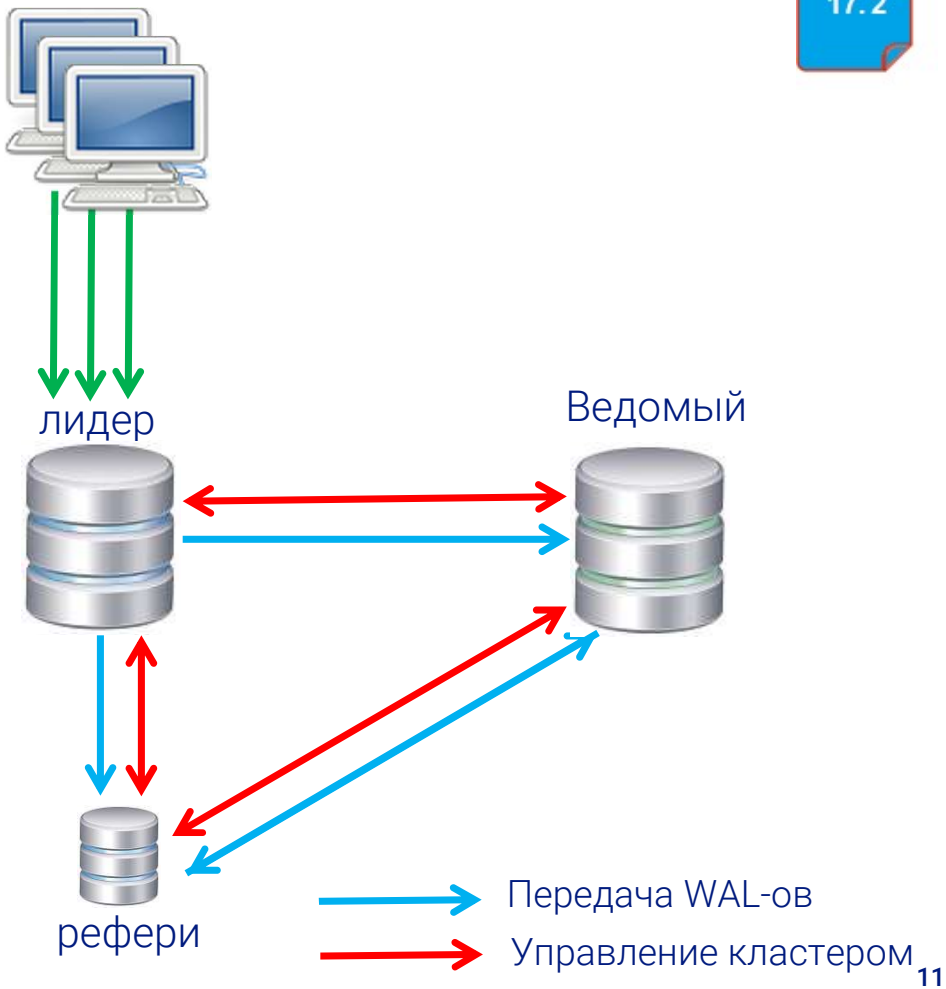
17.2

Рефери – конфигурация 2+1

Если больше 2 узлов — каждый узел может быть и рефери

Если 2 узла — рефери узел — легковесный экземпляр, не содержит пользовательских данных, но член кластера BiHA: сам кандидатом на лидера не выступает, но участвует в голосовании

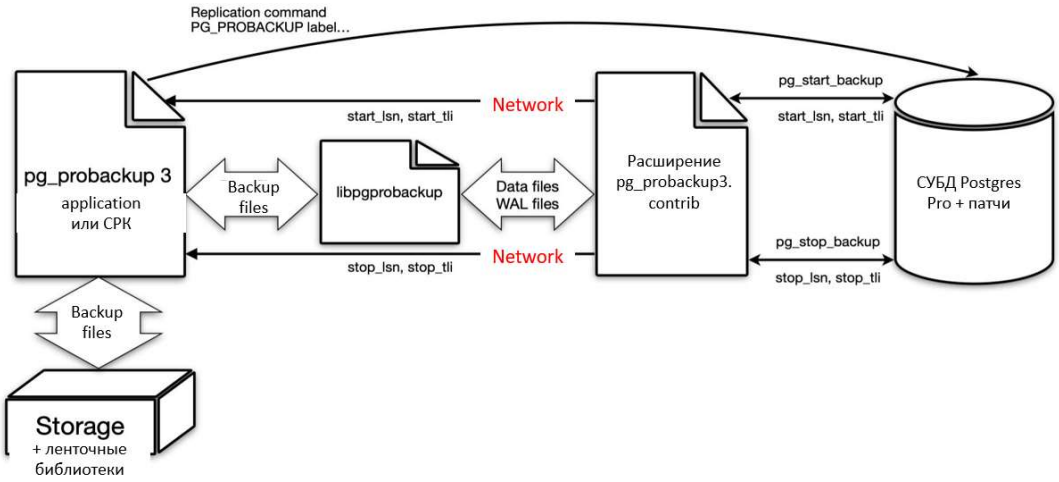
Защита от потери основного ЦОД — вынос резервного узла в резервный ЦОД (не лидер, не голосует, нет split brain, перевод в лидер вручную)



Новая версия pg_probackup — v3 (ОПЭ)

- В весии v2 для каждой версии/редакции Postgres Pro – свой pg_probackup (т.к. структура данных меняется)
- Из каждого backup надо восстанавливаться pg_probackup той же версии в БД той же версии
- В новой версии расширение для конкретной версии/редакции входит в состав ПО (архитектура)
- Не нужен SSH протокол (свой репликационный протокол)
- Новый формат хранения резервных копий (а не блоков)
    - Архив в больших файлах, а не россыпь файлов,

  в них идет параллельная запись кусков файликов
    - Смесь кусков файлов и метаданных
    - Параллелизм
- Сжатие, расширенная retention policy, лента, S3, СРК
- Community (с ограничениями) и Enterprise версии
- Доступ из приложения, СРК, Python API
- Fuse — запуск инстанса на бэкапе

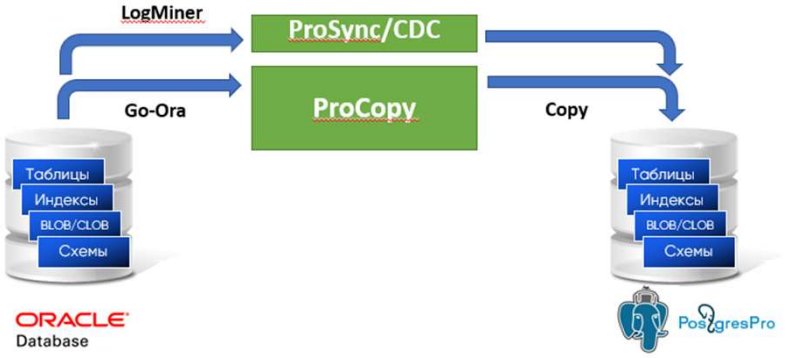


PostgresPro

Миграция с Oracle

Перенос данных при миграции

- ProGate (ProCopy, ProSync (CDC) + DataQuality) + Утилита pgpro_validate
- V1 — Oracle –> Postgres Pro Enterprise
- Мигрирует схемы, таблицы, файлы, результат SELECT
- Возобновление после сбоя/остановки
- BLOB, CLOB, BFiles
- Mapping имен схем, таблиц, колонок



- Параллелизм копирования таблиц (ProCopy), отключение индексов и ограничений целостности
- ProSync
  - Запоминает SCN
  - Переносит схемы целиком с фильтрацией по таблицам или таблицы с фильтрацией по колонкам
- Скорость:
  - На 10% быстрее Датафлот
  - Соизмерима с Ora2pgcopy от Форс

Утилита pgpro_Validate

- Утилита для проверки корректности и ссылочной целостности Postgres Pro
  - Поиск лишних/потерянных файлов и проверка прав доступа в PGDATA
  - Поиск испорченных индексов
  - Проверка CHECKSUM блоков
  - Проверка доступности всех записей таблицы на чтение
  - Проверка корректности pg_catalog
- Можно указать, какие БД/таблицы проверяем
- Часть тестов можно выполнять без работающего Postgres
- Умеет давать ответ в 2 режимах: на экран и в JSON

PostgresPro

Безопасность

Поиск избыточных привилегий – примеры отчетов

PostgresPro

17.2

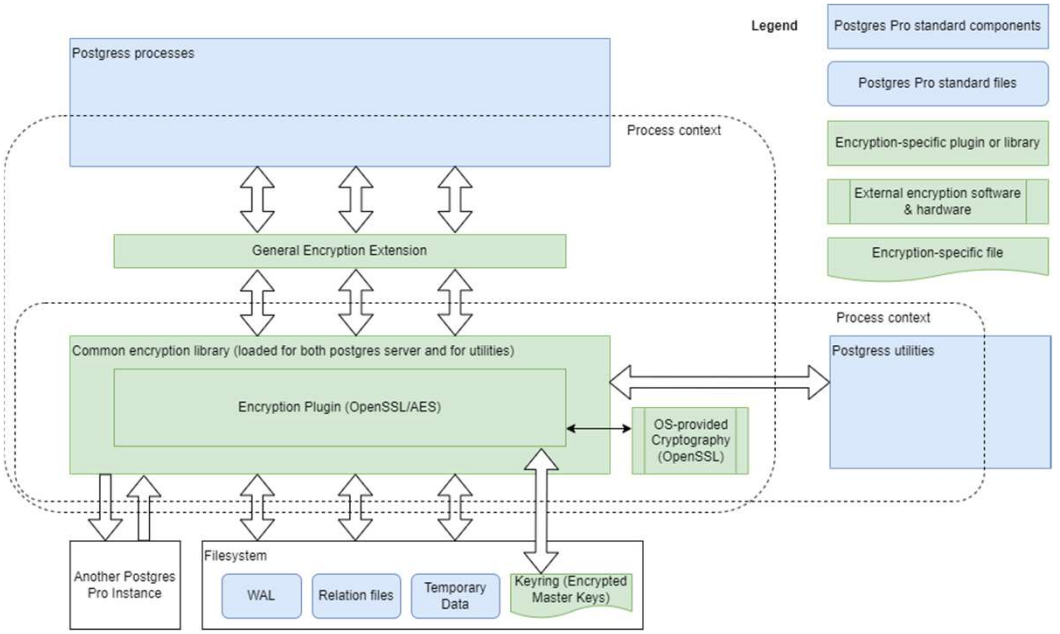
- Статистика выполнения функций или обращений к таблице в разрезе пользователей
- Выполняется путём сравнения всех выданных прав (включая доступ, полученный через другие роли) с реально использованными
- Учитывается вызов системных и пользовательских функций
- Использование прав определяется по статистике *pg_stat_all_tables_per_user* и *pg_stat_all_functions_per_user*
- Отчёт показывает, какие права пользователя использовались, какие не использовались, напрямую ли были получены эти права либо же через какую-то обобщающую роль

User	Object	Access	Access Provided	Provided via Role	Used or not
MARIA	Accounts	READ	TRUE	ACCOUNTANT	FALSE
MARIA	Accounts	READ	TRUE	CHIEF_ACCOUNTANT	FALSE
ANNA	Accounts	READ	TRUE	ACCOUNTANT	TRUE
ANNA	Accounts	READ	TRUE	CHIEF_ACCOUNTANT	TRUE
ANNA	Accounts	UPDATE	TRUE	ANNA	TRUE
ADMIN	Accounts	GRANT	TRUE	ADMIN	TRUE
ACCOUNTANT	Accounts	READ	TRUE	ACCOUNTANT	TRUE
CHIEF_ACCOUNTANT	Accounts	READ	TRUE	CHIEF_ACCOUNTANT	TRUE

Защитное преобразование (TDE)

17.2

- Защита от воровства БД, дисков, несанкционированного доступа
- Расширение с плагинами. Для зарубежных пользователей — AES256
- Работаем с партнерами, которые умеют сертифицировать в ФСБ
- Защита отдельных Tablespace, временных данных и WAL, данные незащищенных Tablespace остаются прежними
- Утилиты могут работать с шифрованными TS
- **CREATE TABLESPACE** ts1 LOCATION '.....' WITH (ENCRYPTED = true)



Audit 2.0

PostgresPro

17.2

- Объединение пользователей в группы
- В СУБД добавлены фиксированные группы для аудита группы действий (ALL_DDL, ALL_DML, ALL_PROC, ALL ROLE ….)
- Назначение группы контролируемых действий группе пользователей

DB	event_type	object_type	object_name	role_name	comment
	AUTHORIZE				все новые авторизации (соединения)
	DISCONNECT				все окончания соединений
	ALL	ROLE			все действия над пользователями и ролями - create/alter/drop user/role/group
	ALTER SYSTEM				все изменения системной конфигурации
	ALL_DDL				все создания и модификации баз данных, таблиц, представлений, хранимых функция и процедур, ...
	GRANT				все разрешения доступов
HR_DB	ALL_DML	SCHEMA	HR_VAULT		весь доступ к данным внутри безопасной схемы
HR_DB	ALL_PROC	SCHEMA	HR_VAULT		все вызовы хранимок из безопасной схемы
	ALL			SUPPORT	все действия инженеров поддержки
	ALL			PGPRO_DBMS_ADMIN	все действия Администраторов СУБД

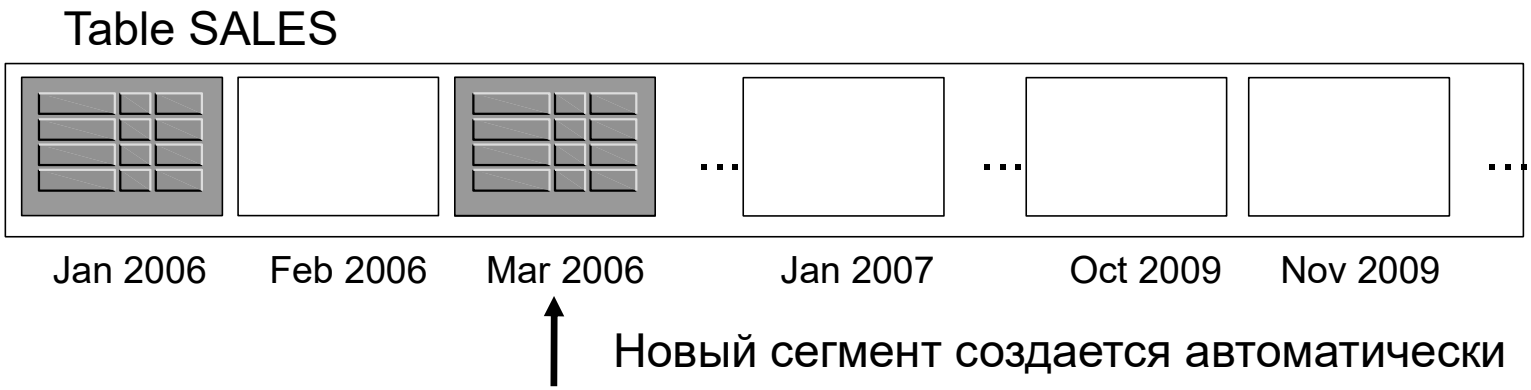
PostgresPro

Разработка

Новое интервальное секционирование

17.2

- Pg_pathman (большое расширение) и секционирование PostgreSQL конфликтуют, трудно поддерживать
- Разработано новое расширение к механизму секционирования PostgreSQL — pgpro_autopart
- Таблица автоматически заменяется на view с триггерами (insert, update)
- Интервал: год, месяц, день, интервал чисел
- Ap_enable_automatic_partition_creation_view('sales', 'month')

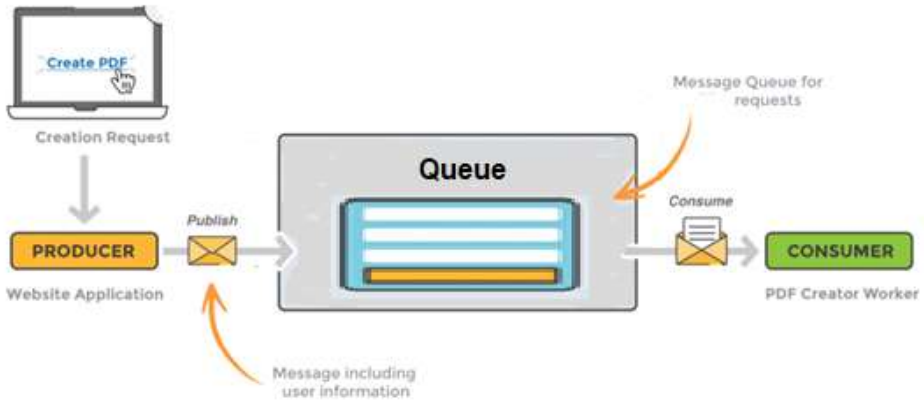



```
INSERT INTO sales (order_date DATE, ...)   VALUES ('04-MAR-2006',...);
```

Транзакционные очереди (Advanced Queuing)

17.2

- Механизм управления очередями сообщений в БД для реализации сложной бизнес-логики
- Подписчик должен подписаться, пока все подписчики не прочтут, сообщение хранится
- Есть хорошее расширение PgQ (скорость, надежность, Enterprise-уровень (в Skype))
- Недостатки существующих реализаций:
  - Только последовательная обработка блоков сообщений
  - Нет приоритетов сообщений
  - Нет фильтрации по атрибутам
  - Нет Exception queue (пока сделали по времени)
  - Нет транзакционности !!! (автоматический возврат сообщений в очередь с отсрочкой видимости)

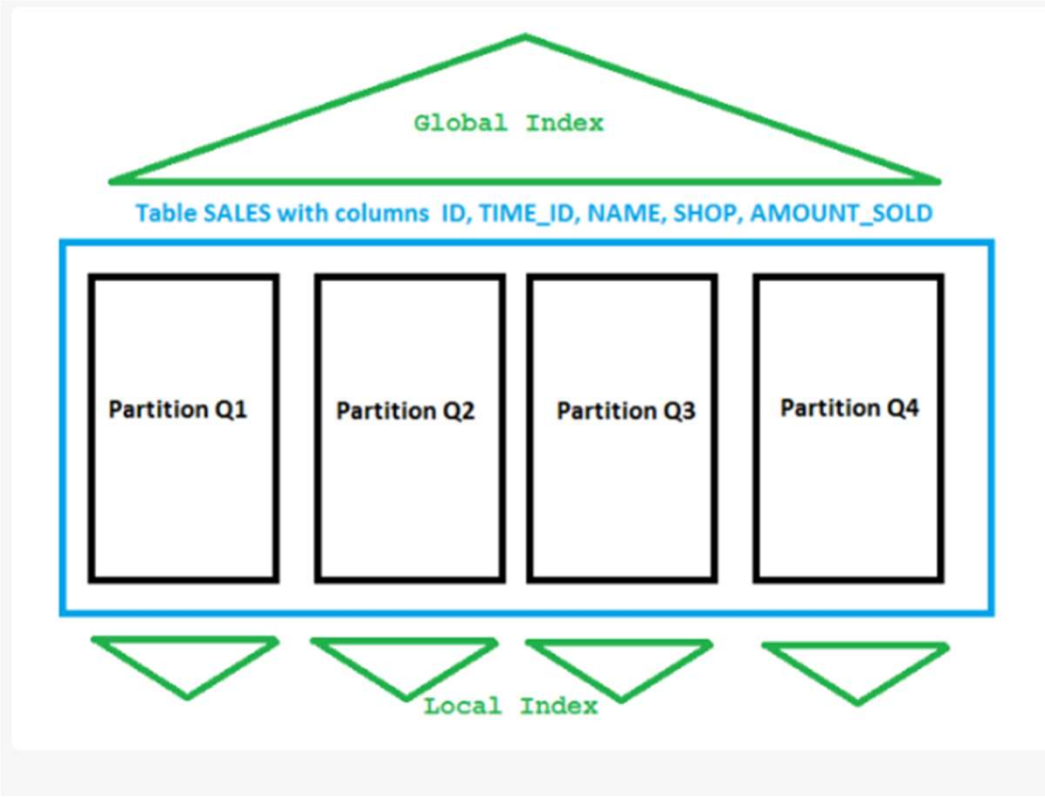


Транзакционные очереди сообщений

- Скорость — десятки тысяч сообщений в секунду
- Поддержка передачи в сообщении объектов самого разного типа (сообщения, события, заказ активности)
- Автоматическое удаление обработанного сообщения по Commit на стороне получателя (как только 1 подписчик его использовал)
- Сохранение информации в очереди при перезапуске сервера
- Сохранение информации в очереди при переключении на Standby
- Автоматический Retry with delay сообщения по Rollback на стороне получателя
- Пример: DB транзакции (чтение из очереди, построение отчета, неудачная посылка, откат транзакции)

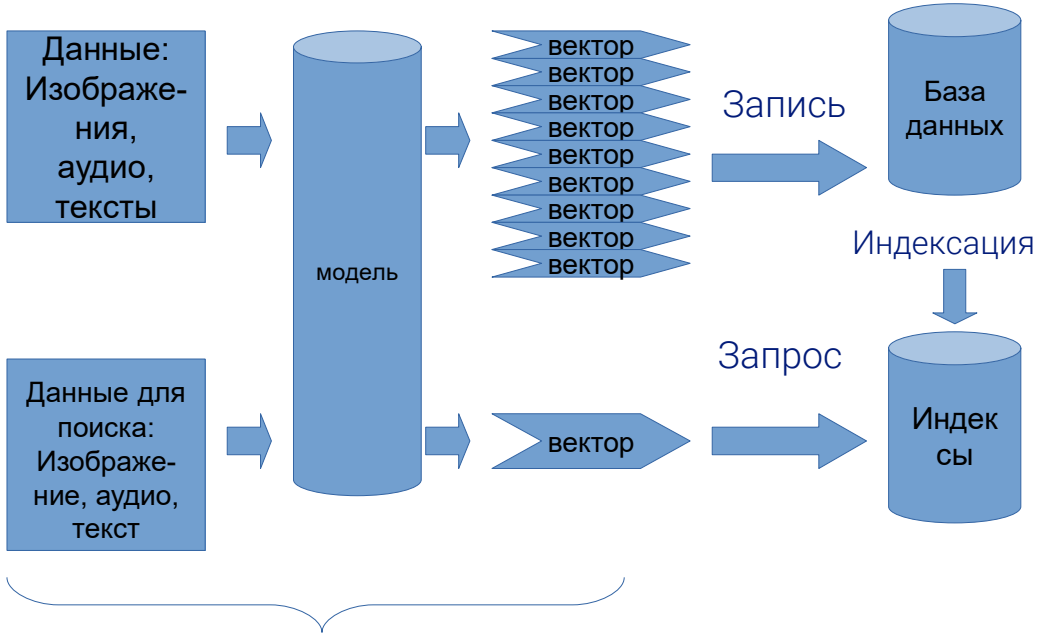
Глобальные индексы

- Индексы ускоряют выборку
- При секционировании локальные индексы строятся для каждой секции и указывают только на ее строки и включают ключ секционирования
- Но часто существуют запросы, требующие индексации по колонкам, не совпадающим с ключом секционирования
- То есть индекс должен указывать на строки разных секций — глобальный индекс
- Глобальный индекс может быть уникальным



Векторная БД

- Для AI используется преобразование неструктурированных данных в вектор(изображения, видео, тексты, аудио, временные ряды/датчики)
- БД осуществляет хранение векторов, создание индексов, поиск ближайших
- Поиск по близости и по направлению (углу)
- Существует расширение pgvector (есть недостатки)
- Колонка типа Vector
- Индекс содержит текущий вектор и группу ближайших точек для раскрутки



Векторная БД – инженерный режим

- Недостатки Pgvector
  - Только 2 алгоритма индексации (hnsw, invflat) — недостаточно
  - Нет условия WHERE на колонки (ищем текст + язык)
  - Нет составного индекса (вектор + значение полей)
  - Может не найти значений, хотя они есть, т.к. ef_search достигнут, но WHERE отсекло часть записей
  - Скорость индексации и поиска
- Создали свое расширение pgpro_vector, снимающее эти ограничения
  - Подключение своих алгоритмов индексации
  - Продолжение поиска за ef_search
  - Условие WHERE

- **`CREATE TABLE primer (id INT4, language iNT4, v vector(5));`**
- **`CREATE INDEX primer_idx ON primer USING hnsw(v vector_l2_ops) WITH (m=20,ef_construction=100);`**
- **`SELECT id FROM primer WHERE language = 2 ORDER BY v <-> '[2,1,4,3,5]' LIMIT 10;`**
- **CREATE INDEX primer_idx1 ON primer USING gann (language gann_int4, v mc_gann) WITH (m=20,ef_construction=100);**

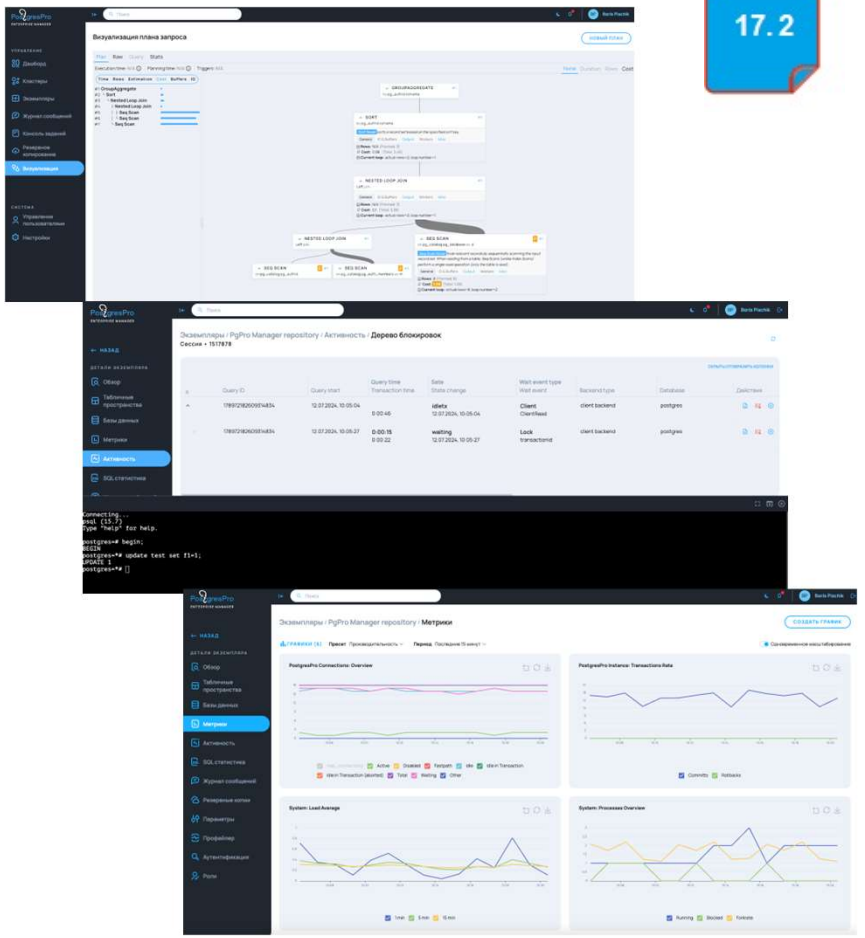
PostgresPro

Управление и администрирование

PPEM 2.0

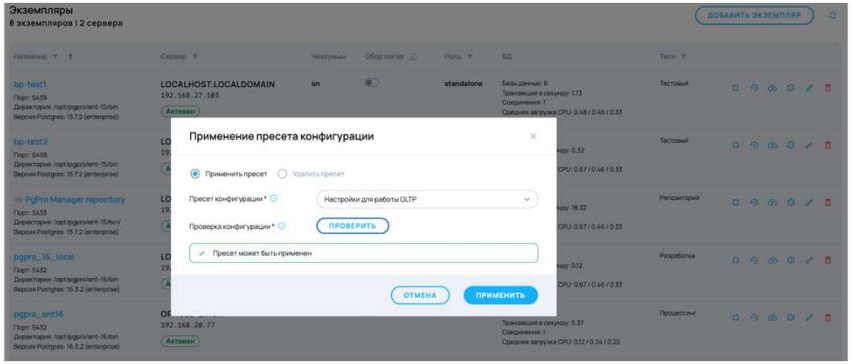
17.2

- Новая архитектура и платформа (Golang)
- Обновленный UI, планы запросов GUI, PWR
- Отображение состояния кластеров репликации
- Показ дерева блокировок
- Показ профиля ожиданий для сессии
- Восстановление в режиме Point-In-Time-Recovery
- Управление объектами схемы БД
- Автоматическая настройка экземпляра под 1C/OLTP



Автонастройка СУБД Postgres Pro

- Автоматическое формирование значений параметров конфигурации
- При создании инстанса (**pg-setup initdb**) и позже
- Утилита pgpro_tune получает данные о конфигурации оборудования: CPU, Memory и формирует набор оптимальных настроек (40+ параметров)
- Специальные алгоритмы для 1с
- По формулам и из опыта
- Графический интерфейс в PPEM



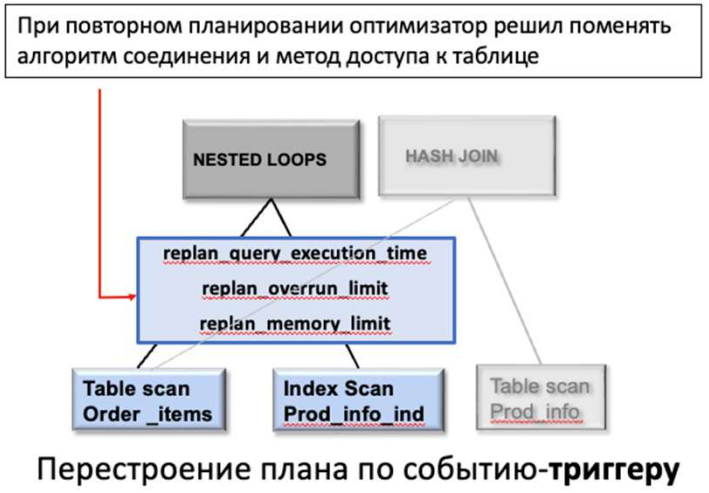
Adaptive Query Optimizer (AQO) и Adaptive Query Executor (Replan)

PostgresPro

17.2

- AQO Enterprise: Новый механизм оценки кардинальности — Delta Learning, повышает качество планов
- Автоматическое включение/отключение режима обучения, когда он не нужен (Enterprise)
- Автоматическая передача базы знаний AQO на реплику через WAL

- AQE: 3 различных события-триггера, которые вызывают перепланирование
  - Время
  - Память
  - Кардинальность
  - Количество попыток
- «Умные» триггеры перепланирования — проверяем, прежде чем прервать запрос
- Поддержка extended-протокола/prepared statements

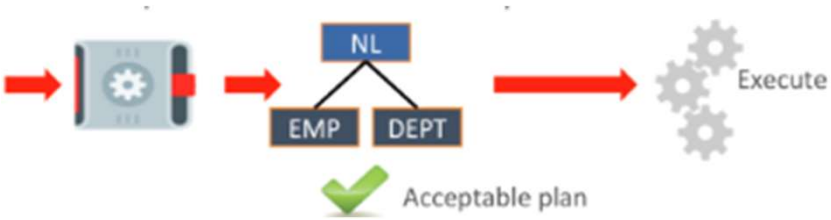


Перестроение плана по событию-**триггеру**

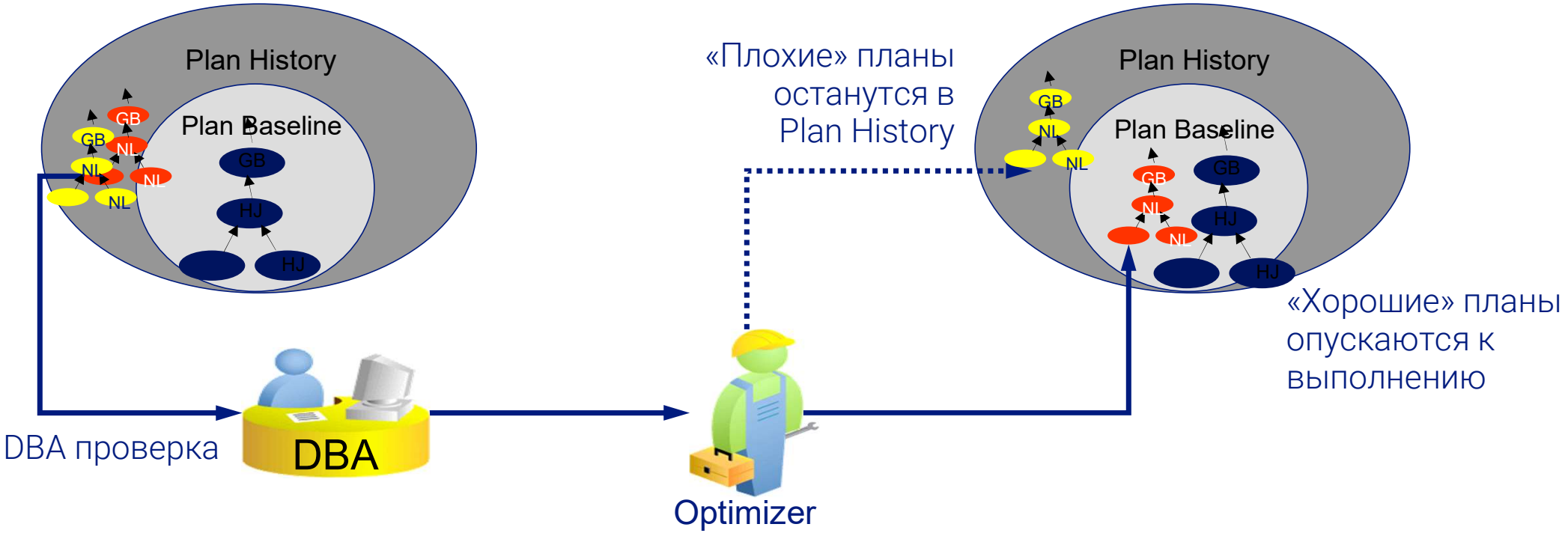
SR_PLAN ➔ Мультиплан

17.2

- «Фиксация» планов запросов — стабилизация производительности
- Новый тип зафиксированных планов — Hintset, привязан не к OID объектов, а к их именам. И его можно перенести на другую БД
- Автоматическая передача набора сохраненных планов на реплику через WAL
- Шаблоны (Wild Card) на имена таблиц в сохраненных планах — полезно для временных таблиц 1С (temp_tab_*)
- Ассистент регистрации запросов — упрощенный механизм «захвата» планов для последующей фиксации (фиксируются ID + планы) и потом можно заморозить нужный план (а не текущий)



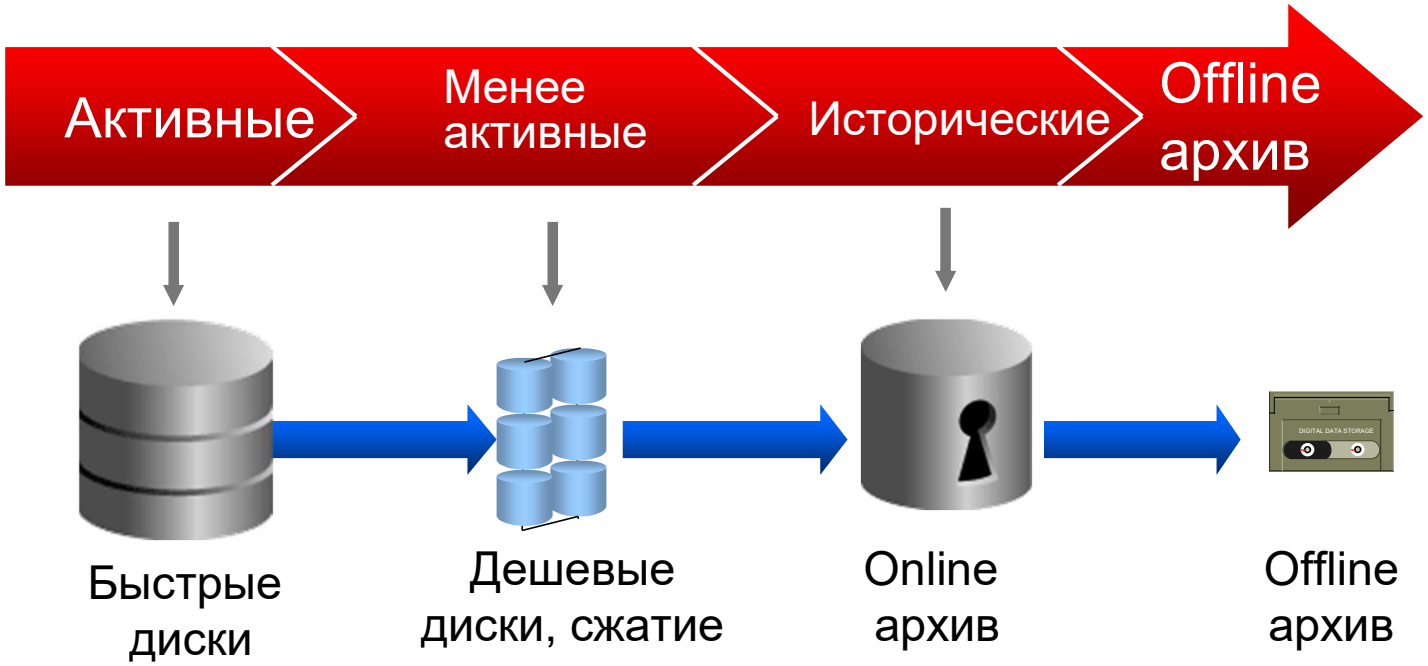
Управление стабилизацией планов (мультиплан)



- Управление планами запросов на основе списка «хороших» планов — Baselines
- DBA может допускать/запрещать определенные планы к выполнению
- Объединяет работу технологий sr_plan и pg_hint_plan под общим интерфейсом и настройками

Цикл жизни данных

(ILM – Information Lifecycle management)



ILM – автоматическое перемещение/сжатие

- Информация о времени последнего чтения и изменения таблицы/секции пользователем хранится в pg_stat_all_tables_per_user
- Работа системных пользователей (vacuum, backup) не фиксируется
- Политика ILM задается функцией pg_ilm_add_rule

  Pg_ilm_add_rule('policy1', 'T1', 'NO_ACCESS', '3 month', 'ALTER_TS', 'TS2')
- Перемещение, сжатие, [read only]
- Тип политики — NO_ACCESS, NO_MODIFICATION
- Выполнение политики — через job_scheduler или вручную *pg_ilm_process_all_rules()*

Температурная карта: бит- секция

PostgresPro

Из PostgreSQL 17

Полезные механизмы из PostgreSQL 17

- Наши ~~MERGE/SPLIT~~ partition , On Login trigger, transaction_timeout параметр (ограничение времени транзакции)
- Оператор MERGE:
  - Не только вставка новых и модификация совпадающих строк, НО и обработка несовпадающих строк целевой таблицы (WHEN NOT MATCHED)
  - Возврат всех измененных/добавленных строк и типа операции
- COPY FROM … (on_error ‘ignore’ ….) — при групповой загрузке не остановится при ошибках формата
- Привилегия maintain и роль pg_maintain на конкретную таблицу или пользователю
  - vacuum, vacuum full, analyze, cluster, refresh MV, lock table, reindex
- Быстрое преобразование физической репликации в логическую для VLDB (pg_createsubscriber)

PostgresPro

Q&A

PostgresPro

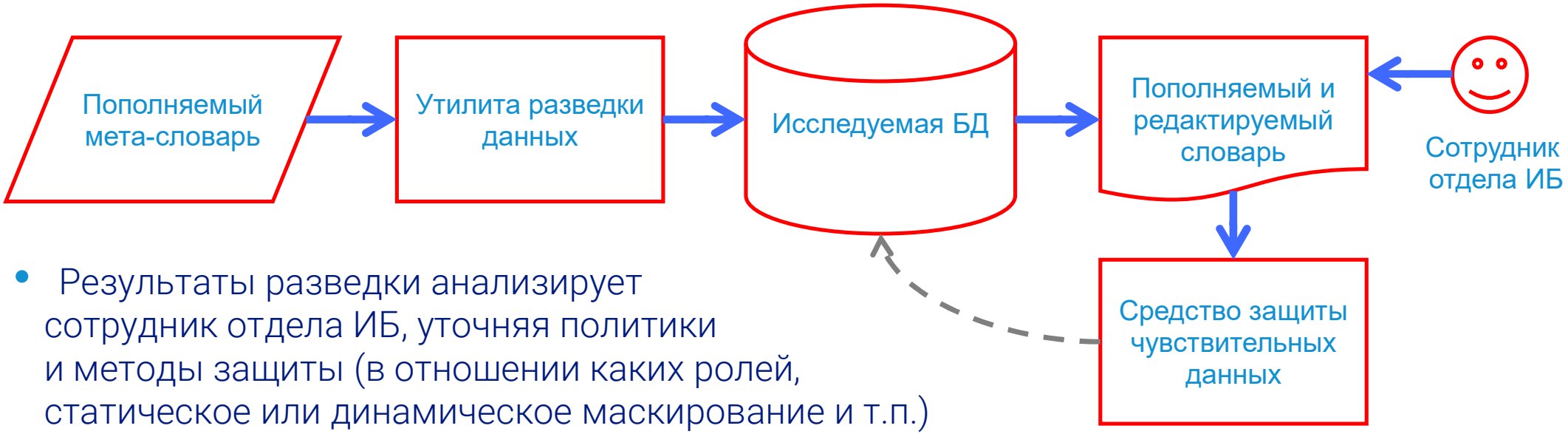
Спасибо за внимание!

Поиск чувствительной информации

PostgresPro

17.1

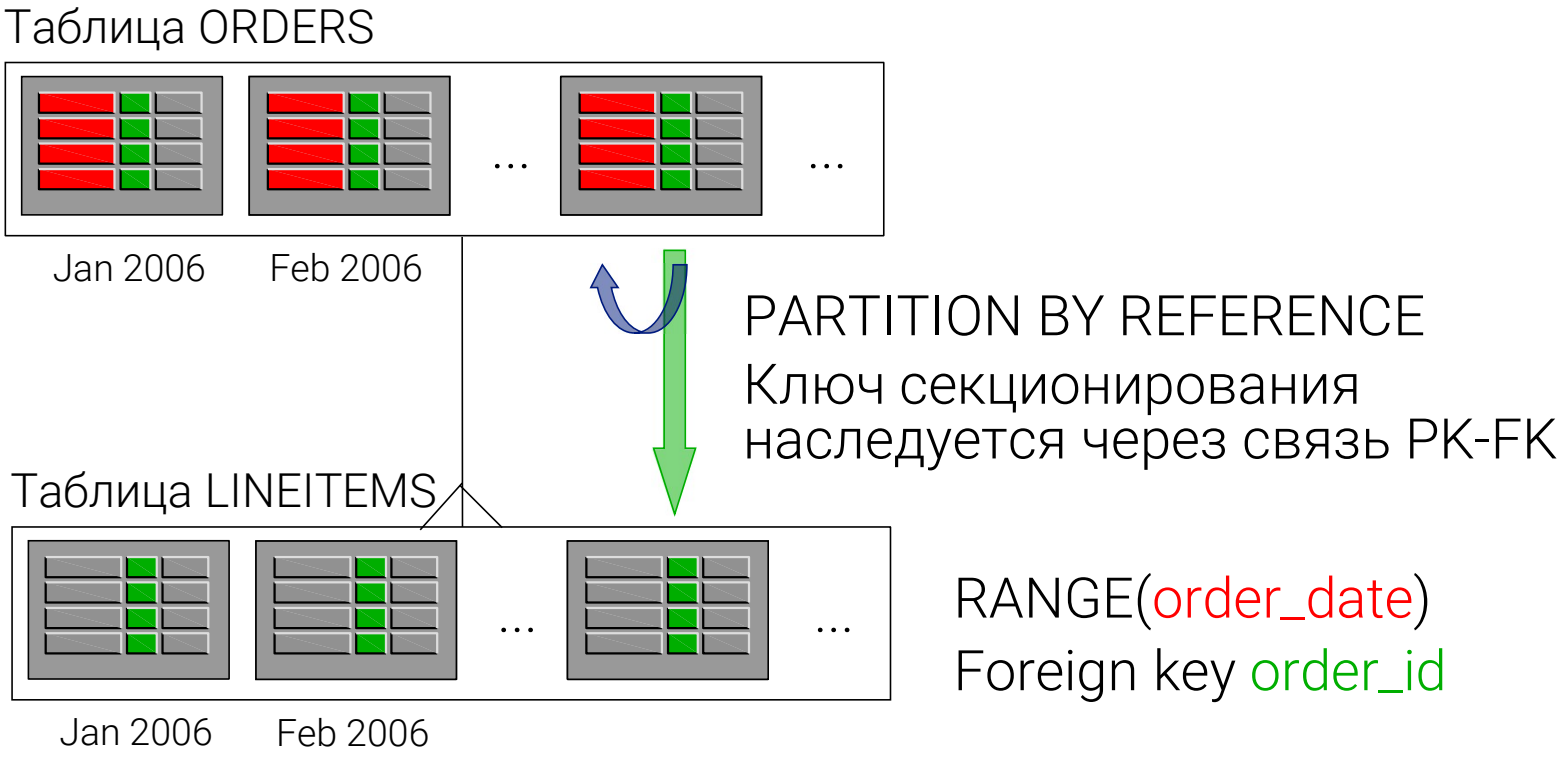
- Поиск колонок таблиц — кандидатов на маскирование (пароль, credit card, зарплата, названия организаций, фамилии и т д)
- Для разведки используется мета-словарь, содержащий маски для проверки имен и содержимого полей по набору регулярных выражений и константным значениям



- Результаты разведки анализирует сотрудник отдела ИБ, уточняя политики и методы защиты (в отношении каких ролей, статическое или динамическое маскирование и т.п.)
- Обновленные параметры из словаря применяются при использовании средства защиты

Reference Partitioning

RANGE(order_date)
Primary key order_id



TDE – реализация – управление ключами

- Применяется трёхуровневая схема:
  - MMK – главный мастер-ключ, хранится в защищённом месте, шифрует мастер-ключи сервера
  - MMK создает и хранит чужое ПО (например HashiCorp Vault)
  - SMK/WMK – мастер-ключи сервера (TS и WAL), шифруют сессионные ключи (для страниц таблиц, индексов, журналов) Они зашифрованы MMK
  - У шифрованных TS в папке пустой файл pg_encryption. И файл ключей keyset.bin в подпапке БД
  - Уникальный ключ шифрования страницы формируется через OpenSSL на основе SMK/WMK, страница шифруется, а инфо для восстановления этого ключа шифрования (для расшифровки) хранится в доп файле keyring для этой таблицы/индекса. Для WAL другой механизм (WMK, LSN, таймлайн)

- Разработаны процедуры:
  - смены MMK с перешифровыванием серверных SMK/WMK
  - смены SMK/WMK с перешифровыванием или ротацией (хранится связка ключей, т к старые нужны для старых страниц) сессионных SSK/WSK
  - Резервного копирования с дополнительным сохранением файла keyring и информацией, какой версией (датой генерации) MMK пользовались при этом