

The logo for Postgres Pro, featuring a stylized white bird-like icon above the text.

PostgresPro

Обеспечение эффективной работы 1С на Postgres Pro

Андрей Забелин

Postgres Professional

Что такое Postgres Pro Enterprise

- **Отечественная коммерческая объектно-реляционная СУБД** для высоконагруженных систем крупных предприятий
- **Разработчики и техническая поддержка в России**
- **Сертификат ФСТЭК, уровень доверия УД 4** (защита конфиденциальной информации)
- **В реестре** российских программных продуктов
- **Регулярное слияние** с ванильной СУБД PostgreSQL

СУБД Postgres Pro – первый в России коммерческий продукт на основе PostgreSQL. Входит в Единый реестр отечественных программ и баз данных Минкомсвязи.

Standard

Современная СУБД, включает все новые функции PostgreSQL и полезные доработки от компании

Enterprise

Наиболее полнофункциональная СУБД с высокой производительностью и масштабируемостью

Certified

Сертифицированные ФСТЭК версии Standard и Enterprise



* Специальная версия для 1c

Версии Postgres Pro для 1С



На сайте <https://v8.1c.ru/tekhnologii/postgrespro/> опубликованы версии Postgres Pro и операционные системы, на которых поддерживается 1С:Предприятие.

Postgres Pro 1C - это сборка PostgreSQL с нашими патчами для совместимости с 1С, которую можно получить на сайте <https://1c.postgres.ru/> . Вы можете использовать это ПО бесплатно для любых целей, кроме предоставления СУБД как услуги (DBaaS) и встраивания в свои продукты. Если вам понадобится техническая поддержка в режиме 24x7, вы всегда можете приобрести ее для этой бесплатной версии в компании Postgres Professional. Запрос на приобретение техподдержки по адресу 1c@postgrespro.ru .

Postgres Pro Standard и Postgres Pro Enterprise – это платные версии Postgres Pro.

В случае использования Postgres Pro Enterprise есть лицензии со специальной ценой, но в этом случае использовать наше ПО можно только для 1С.

1С:Предприятие 8

Система программ

Москва

Программы 1С

Платформа и технологии

Поддержка и обучение

Приобретение и внедрение

Главная > Поддерживаемые версии PostgresPro платформой «1С:Предприятие 8»

Поддерживаемые версии PostgresPro платформой «1С:Предприятие 8»

Список новых поддерживаемых версий СУБД PostgresPro с указанием поддерживаемых ОС.

Содержание

- PostgresPro 1C
- PostgresPro Enterprise
- PostgresPro Standard

PostgresPro 1C

PostgresPro 1C*	15.2.1	14.7.1	13.10.1	12.14.1	11.19.1
Altros 7	-	-	-	-	-
Альт Linux 8		-	-	-	-
Альт Linux 9, 10	-	-	-	-	-
Альт Linux-spt 7		-	-	-	-
Альт Linux-spt 8		-	-	-	-
Альт Linux-spt 8.2	-	-	-	-	-
Astra Open 2.12	-	-	-	-	-
Astra Смоленск 1.6	-	-	-	-	-
Astra Смоленск 1.7	-	-	-	-	-
Debian 9		-	-	-	-
Debian 10,11	-	-	-	-	-
Fed OC 7		-	-	-	-
Fed OC 7.3	-	-	-		
Red Hat Enterprise Linux 7		-	-	-	-
Red Hat Enterprise Linux 8	-	-	-	-	-
Red Hat Enterprise Linux 9	-	-	-	-	
RDS Enterprise Linux 7	-	-	-	-	-
RDS SX Linux 7	-	-	-	-	-
SUSE Linux 12, 15	-	-	-	-	-
Ubuntu 18.04		-	-	-	-
Ubuntu 20.04	-	-	-	-	-
Ubuntu 22.04	-	-	-	-	
Ubuntu 22.10	-	-	-		
Windows	-	-	-	-	-

*PostgreSQL для 1C от Postgres Pro

Ознакомиться со списком более ранних поддерживаемых версий Postgres Pro 1C можно [здесь](#) (pdf).

PostgresPro Enterprise

PostgresPro Enterprise	15.2.1	14.7.1, 14.7.1C**	13.10.2, 13.10.2C	12.14.1	11.19.1
------------------------	--------	-------------------	-------------------	---------	---------

Приобретая лицензию на Postgres Pro Enterprise для 1С, заказчик получает право использовать СУБД **только с платформой 1С**, нельзя организовывать в СУБД базы данных других приложений.

- **Лицензии на ядра процессоров без ограничения срока действия** (постоянные лицензии). Каждая лицензия включает 1 год стандартной технической поддержки. Минимально для 1 установки СУБД необходимо приобрести лицензию на 2 ядра.
- **Лицензии на серверы и пользователей.** Это лицензии без ограничения срока действия (постоянные лицензии), включающие 1 год стандартной технической поддержки. Необходимо приобрести лицензии на то количество серверов СУБД, которое используется в конфигурации 1С. Для каждого сервера БД необходимо приобрести столько пользовательских лицензий, сколько **пользовательских лицензий на всех серверах 1С.**

Client



Master

Пример 3-х узлового кластера

sync

async



реплика 1



реплика 2

Client



Master

Пример автоматического переключения



Referee

sync



реплика

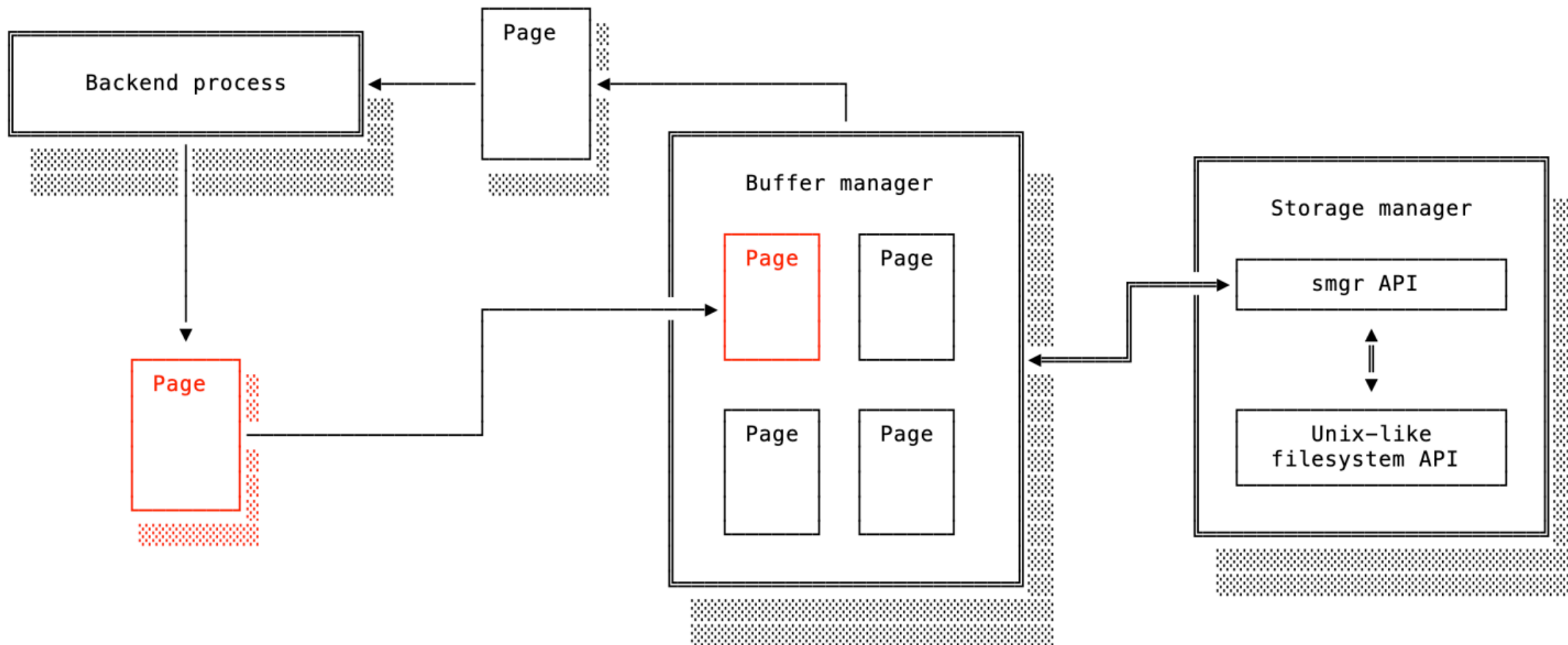
- Физическая репликация синхронная/асинхронная
- Восстановление битых блоков из реплики, восстановление битых записей WAL
- Реплика может быть географически удалена
- Referee – Patroni, Stolon, Corosync ...

Инкрементальные копии создаются на уровне страниц и включают только те данные, которые изменились со времени последнего копирования.

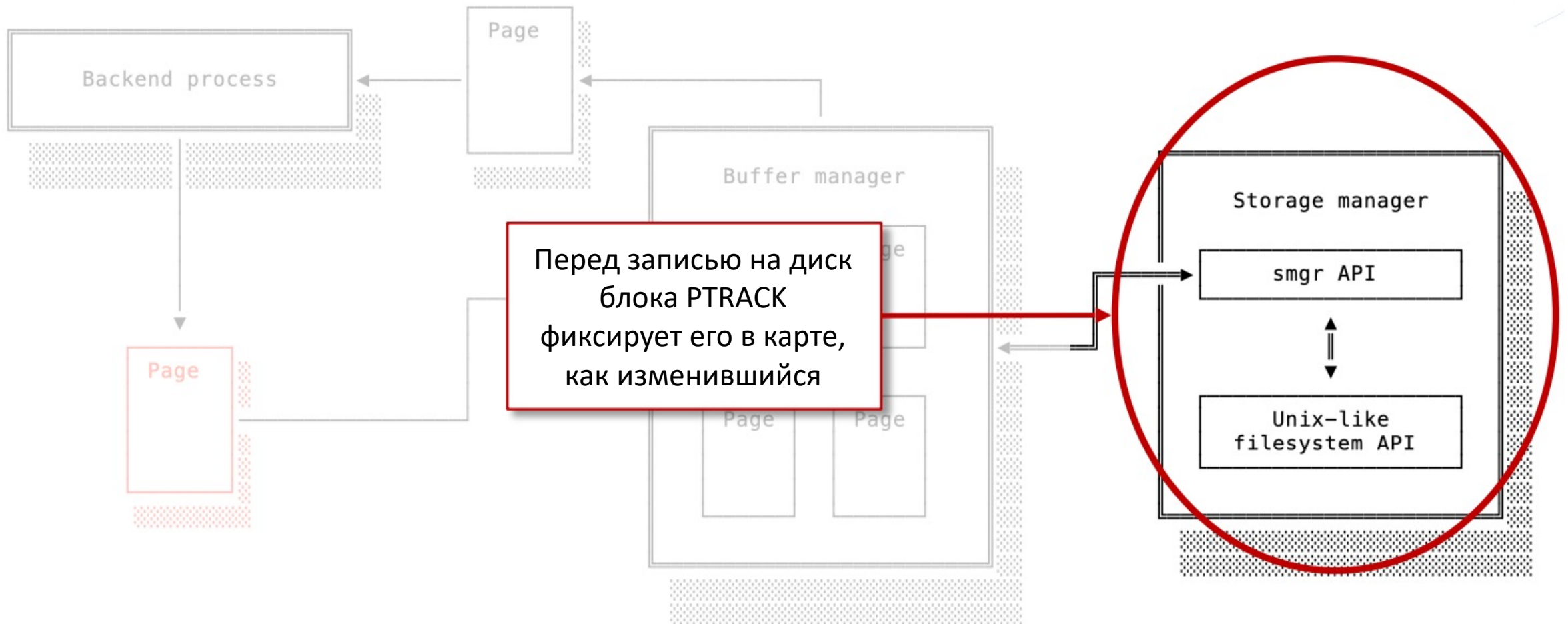
pg_probackup поддерживает следующие режимы :

- В режиме DELTA pg_probackup считывает все файлы баз данных и копирует только те страницы, которые изменились со времени предыдущего копирования.
объём ввода/вывода может равняться объёму при полном резервном копировании.
- В режиме PAGE pg_probackup сканирует все файлы WAL в архиве с момента создания предыдущей полной или инкрементальной копии и копирует страницы, фигурирующие в записях WAL.
если размер WAL файлов сравним с размером базы данных, ускорение будет менее значительным, но размер копии будет меньше.
- В режиме PTRACK Postgres Pro отслеживает изменения страниц на лету. При каждом изменении страницы она помечается в специальной карте PTRACK.
отслеживание приносит небольшие издержки в работу сервера, но значительно ускоряет инкрементальное копирование.

PTRACK — это механизм, предназначенный для инкрементального резервного копирования базы данных Postgres Pro на уровне блоков.



Измененные блоки фиксируются при их записи на диск (используется storage manager API).
Карты блоков PTRACK из оперативной памяти сбрасываются на диск во время контрольных точек.



pg_probackup merge объединяет полную и инкрементальные копии, относящиеся к одной цепочке.

```
pg_probackup merge -B каталог_копий --instance имя_экземпляра \  
-i идентификатор_резервной_копии
```

Если выбрана полная копия, она будет объединена с первой инкрементальной копией после неё.

Если выбрана инкрементальная копия, она будет объединена с родительской полной копией, включая все инкрементальные копии между ними.

После завершения объединения результирующая полная копия будет вмещать в себя все данные, а инкрементальные копии будут удалены как избыточные.

Поддерживается сжатие, если объединяемые копии выполнялись с одинаковой степенью сжатия.

Резервное копирование: проверка целостности резервных копий

Автоматический контроль целостности данных и возможность проверки резервных копий без восстановления.

`pg_probackup` вычисляет контрольные суммы для всех файлов копии во время резервного копирования.

По умолчанию проверка выполняется сразу после создания резервной копии и непосредственно перед восстановлением для выявления возможных повреждений резервных копий.

Проверку целостности можно проводить периодически:

```
pg_probackup validate --recovery-target=immediate -i Incr2
```

Восстановление в таком случае можно ускорить пропустив процедуру проверки :

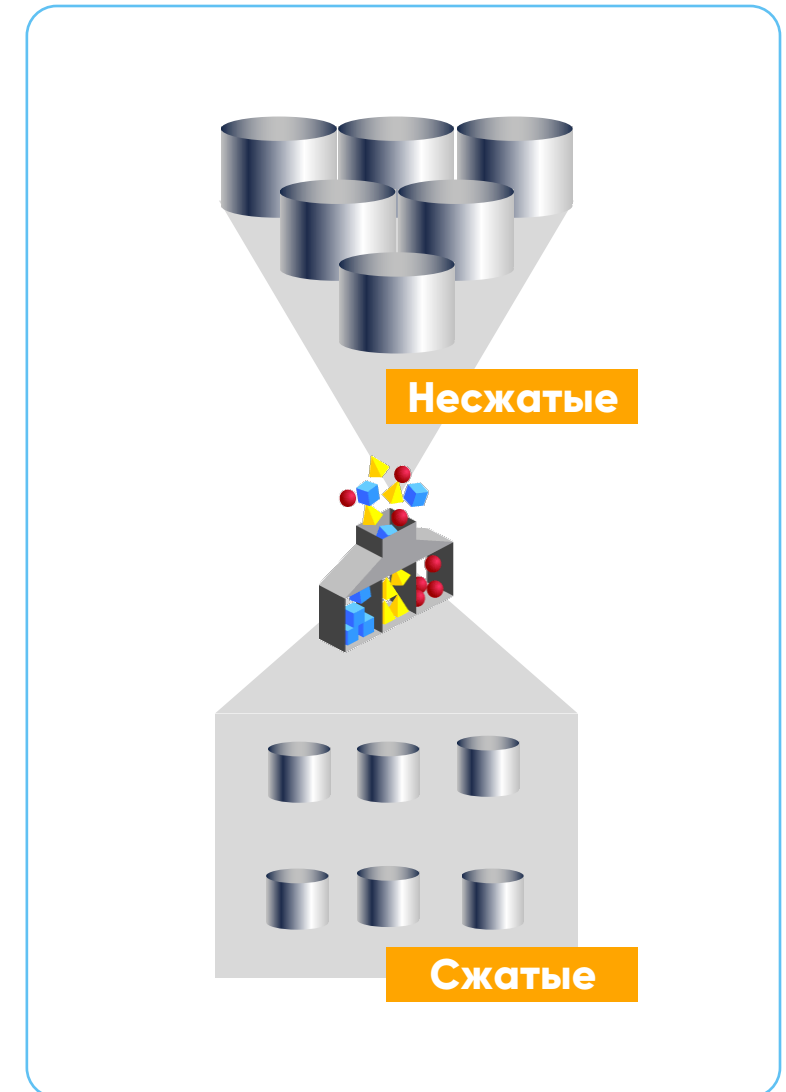
```
pg_probackup restore --no-validate --recovery-target=immediate -i Incr2
```

- Параллельное выполнение: выполнение внутренних процессов команд backup, restore, merge, delete, validate и checkdb в несколько параллельных потоков.
- Сжатие: хранение копируемых данных в сжатом состоянии для экономии дискового пространства.
- Политика хранения: управление архивами WAL и резервными копиями в соответствии с установленными правилами их хранения. Вы можете ограничить хранение резервных копий по времени или их количеству, а также переопределить время жизни (TTL) для избранных копий. Потерявшие актуальность резервные копии могут объединяться или удаляться.
- Архивирование внешних каталогов: резервное копирование файлов и каталогов, расположенных вне каталога данных Postgres Pro (PGDATA), например скриптов, файлов конфигурации, журналов или SQL-дампов.

CFS (Compression File System) встроена в Postgres Pro Enterprise .

Механизм CFS позволяет реализовать сжатие на уровне страниц в СУБД Postgres Pro Enterprise. Сжатие помогает уменьшить объём данных, который приходится записывать или считывать с диска. Сжатие можно включить только для отдельных табличных пространств. CFS создаёт для каждого файла БД отдельную карту отображения сжатых страниц.

Помимо очевидного плюса, заключающегося в экономии места (до 5 раз), сжатие может также увеличить быстродействие системы.



Мониторинг Postgres Pro Enterprise

Расширение `pgpro_pwr` позволяет получать отчёты по нагрузке для выявления наиболее ресурсоёмких операций в базе данных.

Расширение `pgpro_stats` для сбора статистики планирования и выполнения SQL-операторов.

Расширение `pg_wait_sampling` для периодического сбора статистики по событиям ожидания во всех обычных и фоновых рабочих процессах.

Мониторинг запросов в реальном времени `pg_query_state` для актуальной статистики запроса

- [Top tables by blocks vacuum fetched](#)
- [Top indexes by blocks vacuum fetched](#)
- [Top tables by blocks vacuum read](#)
- [Top indexes by blocks vacuum read](#)
- [Top tables by WAL size generated by vacuum](#)
- [Top tables by vacuum operations](#)
- [Top tables by analyze operations](#)
- [Cluster settings during the report interval](#)

Server statistics

Database statistics

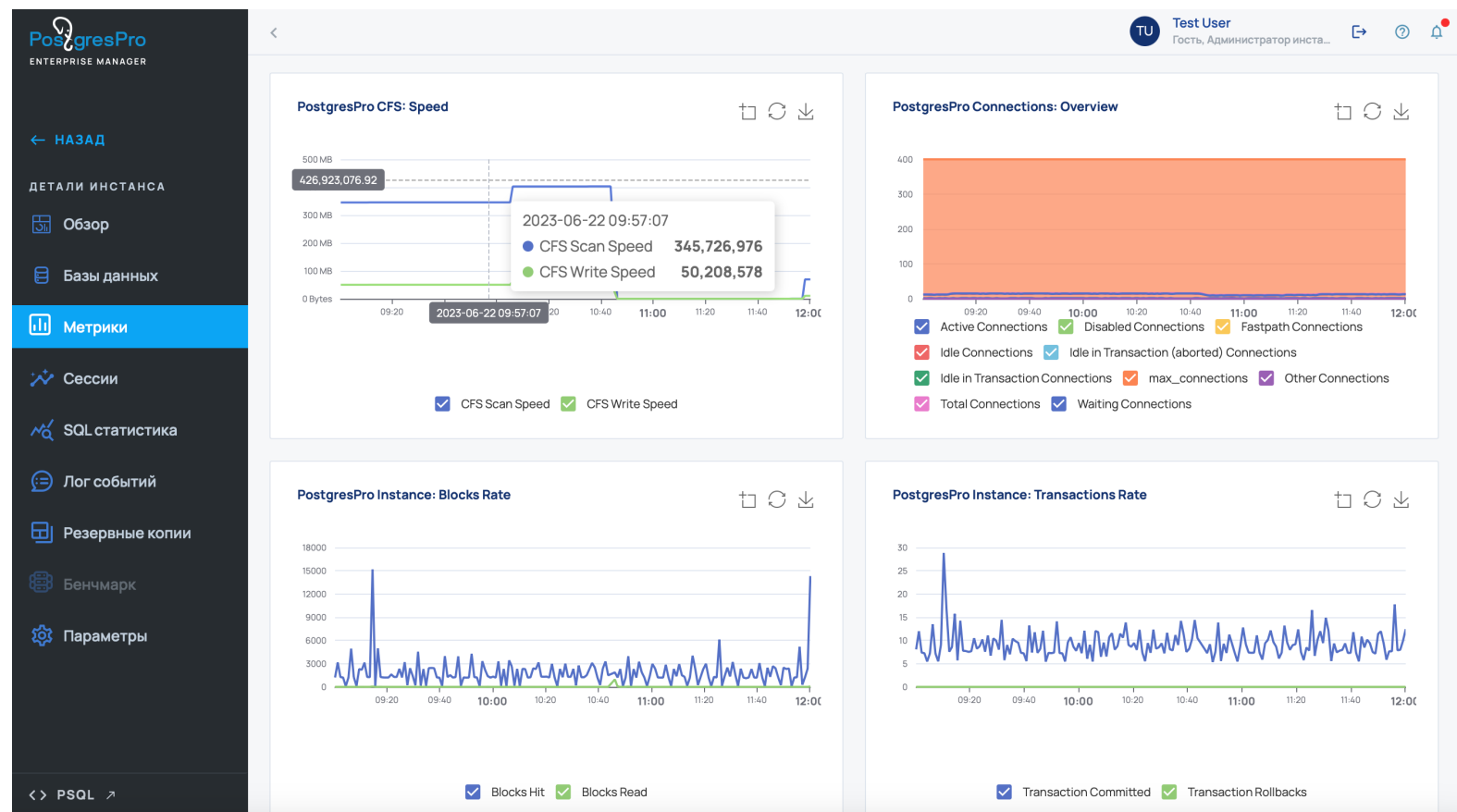
Database	Transactions			Block statistics			Block I/O times		Tuples					Temp files		Size	Growth
	Commits	Rollbacks	Deadlocks	Hit(%)	Read	Hit	Read	Write	Ret	Fet	Ins	Upd	Del	Size	Files		
db5	3377			100.00		185864			2129743	39349						9317 kB	
demo	17183			100.00		193323			2298257	43985						707 MB	
mamonsu_database	61593			100.00		359089			6689577	48937		1416		6998 MB	1337	9517 kB	
postgres	3764	5		99.93	1096	1554516	0.10		12364299	647192	21928	12584	14362	2306 kB	9	86 MB	4744 kB
Total	85917	5		99.95	1096	2292792	0.10		23481876	779463	21928	14000	14362	7000 MB	1346	811 MB	4744 kB

Session statistics by database

Database	Timings (s)			Sessions			
	Total	Active	Idle(T)	Established	Abandoned	Fatal	Killed
db5	80463.05	4.36	0.36	8			
demo	123699.00	225.80	0.33	11	3		
mamonsu_database	80411.25	192.43	0.34	8			
postgres	126511.75	10.96	4.11	25	1		

Мониторинг: Postgres Pro Enterprise Manager

- Графическая система мониторинга и управления БД Postgres
- Концепция «единого окна» - все БД в одной консоли
- Вывод сводной информации – информационная панель Dashboard
- Визуализация метрик производительности - графики с выбором интервала наблюдения
- Активность по сессиям
- Статистика по SQL-запросам
- Запуск/останов БД
- Выполнение backup/restore
- Изменение параметров
- Лог событий



Производительность Postgres Pro Enterprise

64-битные идентификаторы транзакций.

Оптимизация работы с временными объектами.

Улучшенная производительность при использовании множества временных таблиц в отдельных обслуживающих процессах и при большом количестве одновременных подключений.

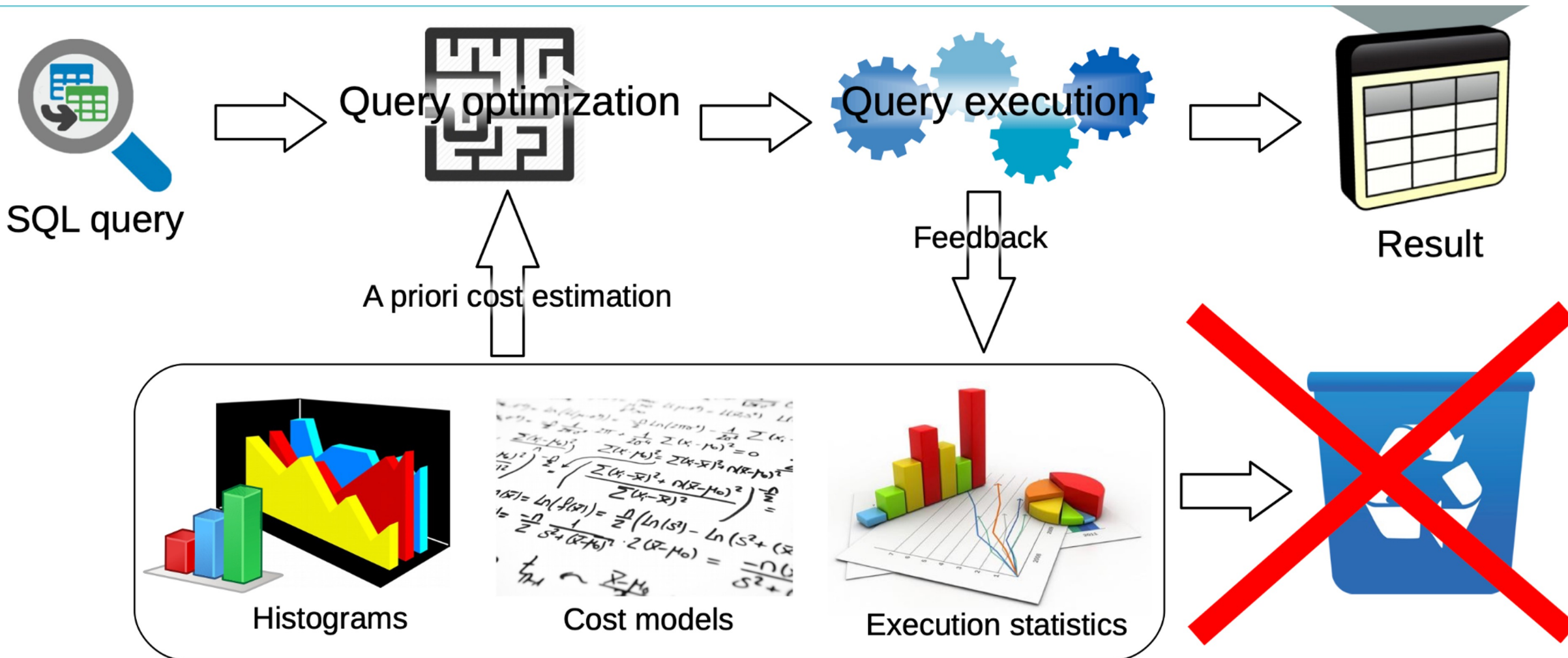
Уменьшенное потребление памяти при сложных запросах со множеством таблиц.

Точная оценка селективности для правильного планирования запросов, поступающих из 1С

Увеличение производительности благодаря уменьшению количества отправляемых серверными процессами друг другу сообщений о событиях аннулирования, касавшихся временных таблиц.



Адаптивная оптимизация запросов (AQO)



```
postgres=# set aqo.mode = disabled;  
SET
```

```
postgres=# EXPLAIN ANALYZE
```

```
SELECT T1.IDRRef, T1.Date_Time_, T1.Executed_, T1.Fld35378_, T1.Fld35374_, T1.Fld35371_, T1.Fld35397RRef, T1.MainAddrPropRRef, T1.Fld35373_, T1.Fld35372_TYPE, T1.Fld35372_RTRe  
...
```

QUERY PLAN

Sort (cost=1196921.31..1198203.54 rows=512890 width=372) (actual time=28179.756..28179.848 rows=153 loops=1)

Sort Key: "**SELECT* 1".date_time_ DESC, "**SELECT* 1".idrref DESC

Sort Method: quicksort Memory: 103kB

-> **Hash Left Join** (cost=588432.44..765992.94 rows=**512890** width=372) (actual time=27913.408..28179.378 rows=**153** loops=1)

...

Planning Time: 13.213 ms

Execution Time: **28180.300** ms

```
postgres=# set aqo.mode = forced;  
SET
```

```
postgres=# EXPLAIN ANALYZE
```

```
SELECT T1.IDRRef, T1.Date_Time_, T1.Executed_, T1.Fld35378_, T1.Fld35374_, T1.Fld35371_, T1.Fld35397RRef, T1.MainAddrPropRRef, T1.Fld35373_, T1.Fld35372_TYPE, T1.Fld35372_RTRe  
...
```

QUERY PLAN

Sort (cost=1196921.31..1198203.54 rows=512890 width=372) (actual time=22908.467..22908.535 rows=153 loops=1)

Sort Key: "SELECT 1".date_time_ DESC, "SELECT 1".idrref DESC

Sort Method: quicksort Memory: 103kB

-> Hash Left Join (cost=588432.44..765992.94 rows=512890 width=372) (actual time=22727.346..22907.965 rows=153 loops=1)

...

Planning Time: 21.105 ms

Execution Time: 22912.668 ms

Using aqo: true

AQO mode: FORCED

Query hash: -7281839219166836624

JOINS: 4

postgres=# EXPLAIN ANALYZE

```
SELECT T1.IDRRef, T1.Date_Time_, T1.Executed_, T1.Fld35378_, T1.Fld35374_, T1.Fld35371_, T1.Fld35397RRef, T1.MainAddrPropRRef, T1.Fld35373_, T1.Fld35372_TYPE, T1.Fld35372_RTRe
...
```

QUERY PLAN

Sort (cost=86185.02..86185.40 rows=153 width=372) (actual time=30.147..30.163 rows=153 loops=1)

Sort Key: **"*SELECT* 1".date_time_ DESC, "*SELECT* 1".idrref DESC**

Sort Method: quicksort Memory: 103kB

-> **Nested Loop Left Join** (cost=1.67..86174.44 rows=**153** width=372) (actual time=21.113..30.013 rows=**153** loops=1)

...

Planning Time: 21.298 ms

Execution Time: **33.825** ms

Using aqo: true

AQO mode: **FORCED**

Query hash: -7281839219166836624

JOINS: 4

Возможность скачать или заказать Trial лицензии нашей СУБД

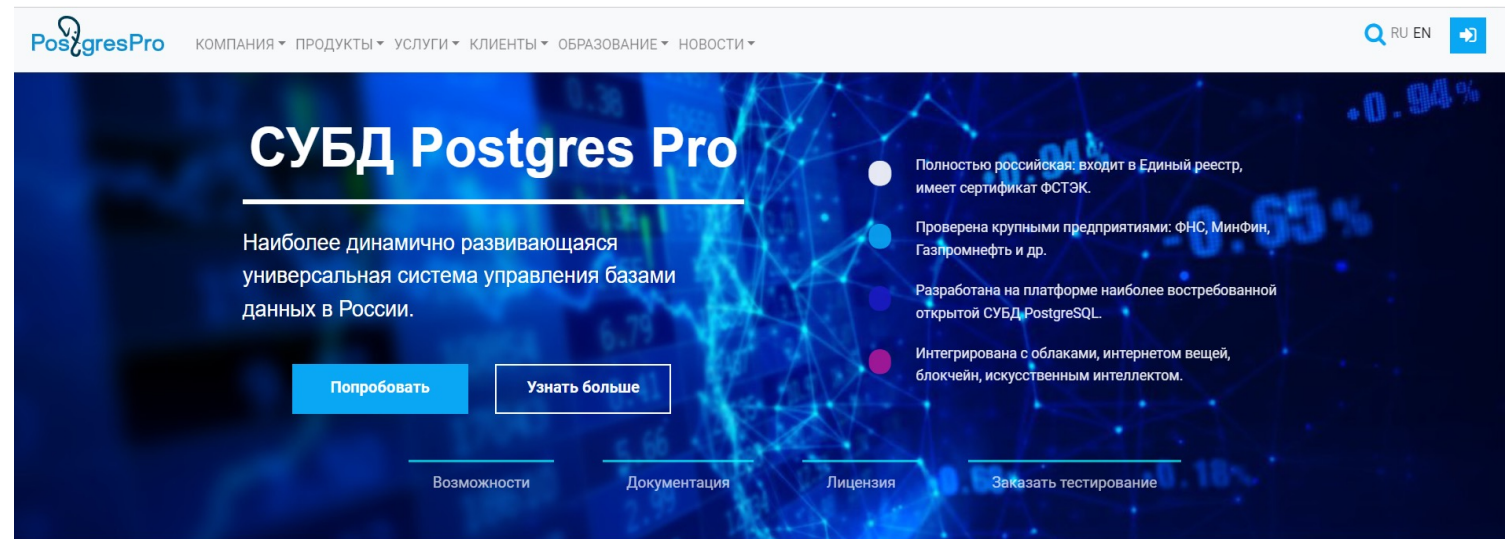
Свежая документация на русском языке

Учебные курсы, материалы, видео

How To видео

Новости

Мероприятия



Postgres Pro доверяют



PostgresPro

Q & A

