POSTGRES UNIVERSAL DATABASE

Major Postgres contributor since 1995

Олег Бартунов,
Moscow University,
Postgres Professional

PostgresPro

Как выбрать правильную СУБД?

Чего обычно хотят от СУБД?

- Функциональность, производительность
- Доступность (лицензия, цена)
- Собственная и локальная экспертиза
- Совместимость с привычной средой
- Наличие техподдержки

И часто забывают про:

- Гибкость, расширяемость
- Долговечность, наличие предпосылок для устойчивого развития

PostgresPro

Почему выбирают Postgres

Это универсальная СУБД

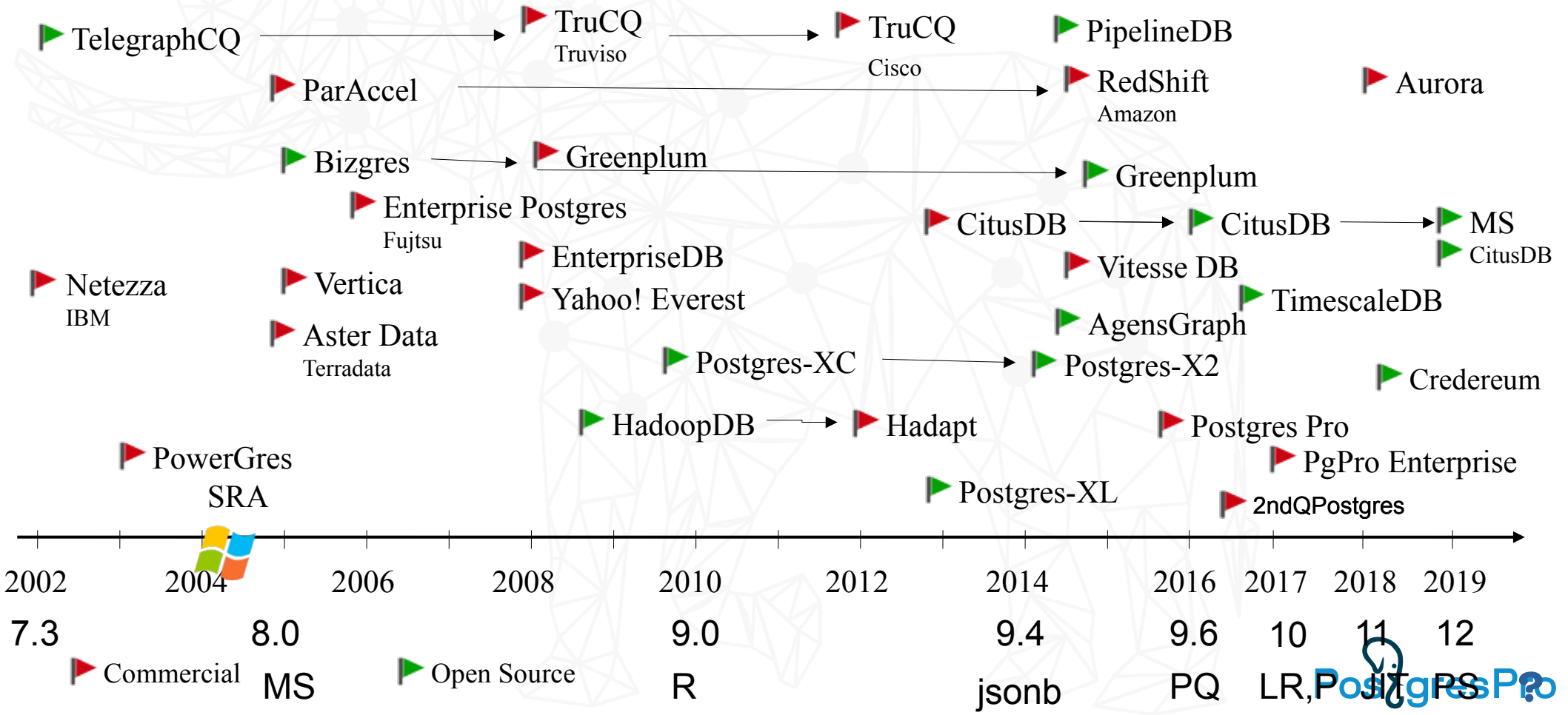
- На старте подойдёт для любого проекта
- Можно работать с Open Source, можно получить коммерческую ТП, можно работать с производным коммерческим продуктом
- Благодаря гибкости сможет адаптироваться к росту вашего проекта.

Это СУБД, имеющая наиболее четкие предпосылки для устойчивого развития

- Устойчивость обеспечивается структурой сообщества: балансом между миром Open Source и интересами компаний-участников сообщества.

PostgresPro

Мир Postgres: OLTP, MPP, OLAP, CLOUD, GIS, STREAM, TIMESERIES, GPU, NoSQL



Краткая история Postgres

1996 — Старт проекта
1997 (6.1) — Интернациализация + **World**
2005 (8) — Поддержка Microsoft Windows + **Windows users**
2010 (9) — Встроеная репликация + **Enterprise users**
2014 (9.4) — JSONB + **NoSQL users**
2016 (9.6) — Параллельная обработка + **OLAP users**
2017 (10) — Логическая репликация, Секционирование
2018 (11) — JIT-компиляция
2019 (12) — Pluggable storage API, SQL/JSON
202X (?) — Адаптация к облачной среде + **ALL**

PostgresPro

Эволюция сообщества Postgres

- 198X – Университетский проект (x10)
- 1995 – Проект сообщества (<400)
- Сообщество разрабатывает для себя (Developer-driven period)
- 200X — Первые компании, профессионально работающие над Postgres (GreatBridge, 2ndQuadrant, EDB...)

PostgresPro

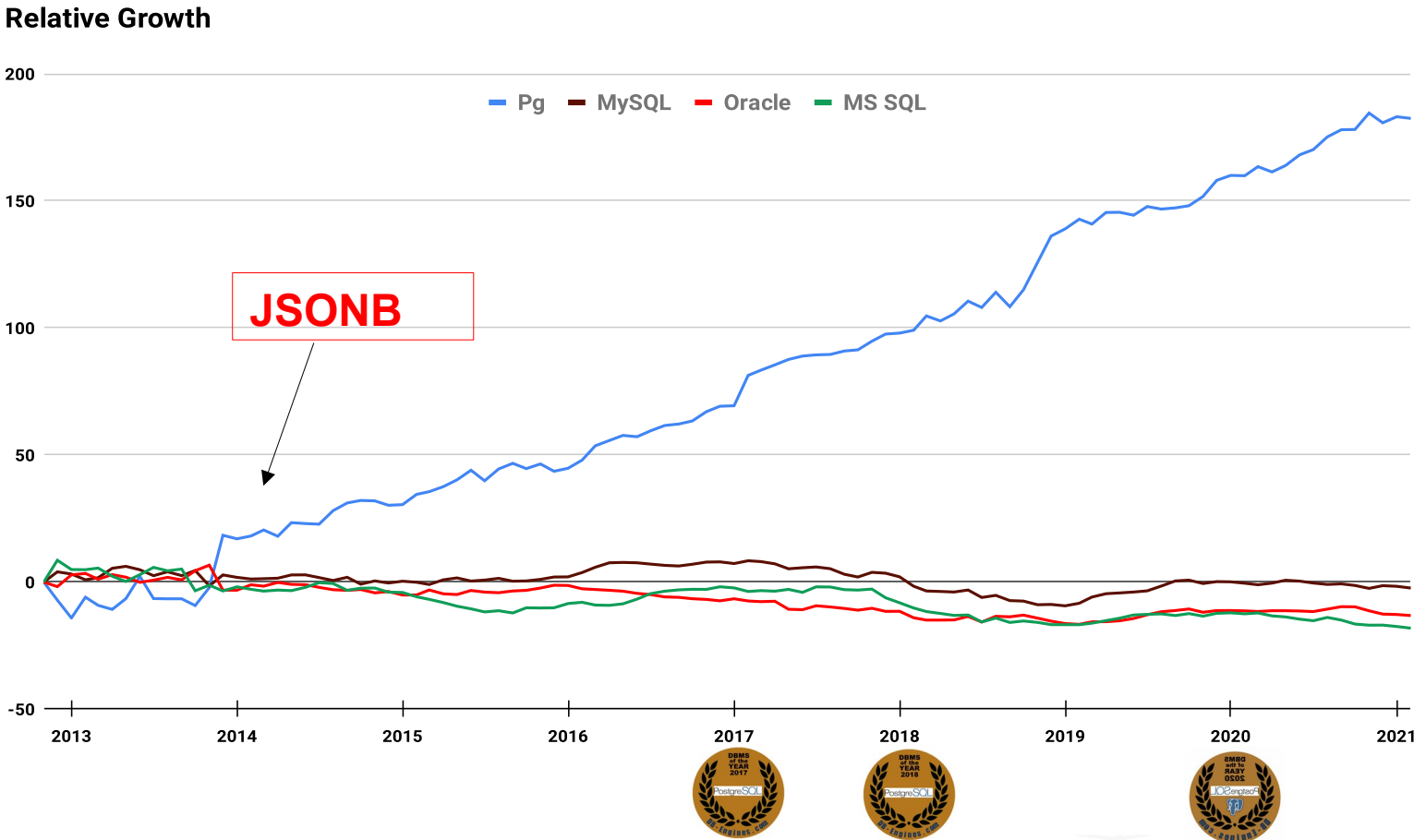
Postgres в профессиональную эпоху

- 2010 — Postgres проникает в Enterprise
- 2015 — Больше компаний (+Citus Data, +Postgres Professional), company-driven period
  - Основная разработка ведется в компаниях
  - Но они не отрываются от сообщества
  - Постгрес становится более зрелым
  - И более професиональным

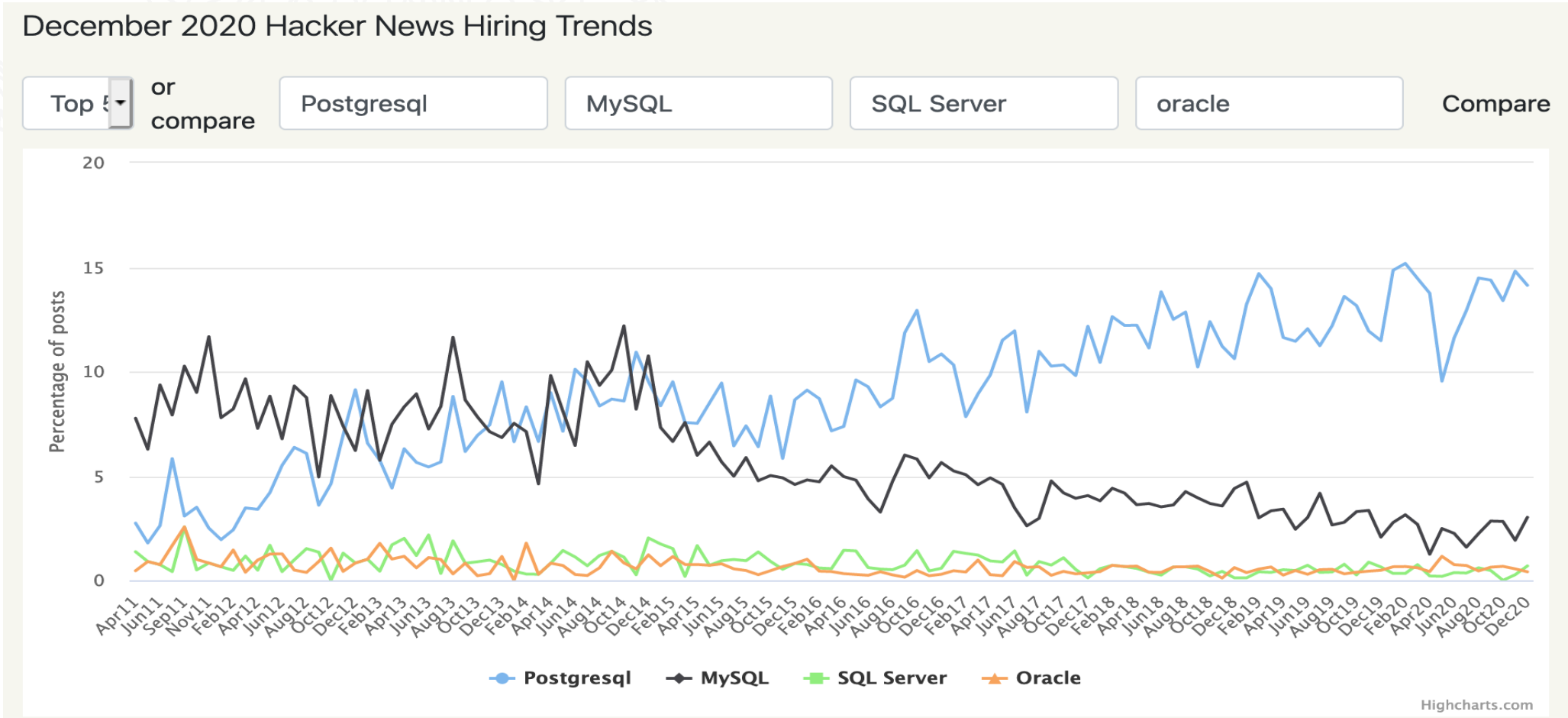
PostgresPro

Популярность Postgres растет

DB-Engines



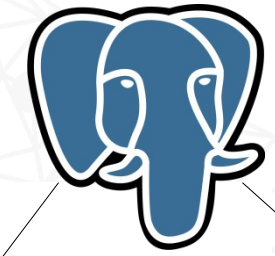
Hacker News Hiring Trends - 2020



Stack Overflow developer survey 2019

MOST USED

MOST LOVED



Databases

All Respondents | Professional Developers

MySQL	52.0%
PostgreSQL	36.3%
Microsoft SQL Server	34.4%
SQLite	30.9%
MongoDB	26.0%
Redis	20.6%
MariaDB	16.5%
Oracle	16.4%
Elasticsearch	15.9%
Firebase	12.3%
DynamoDB	6.8%
Cassandra	3.7%
Couchbase	2.2%

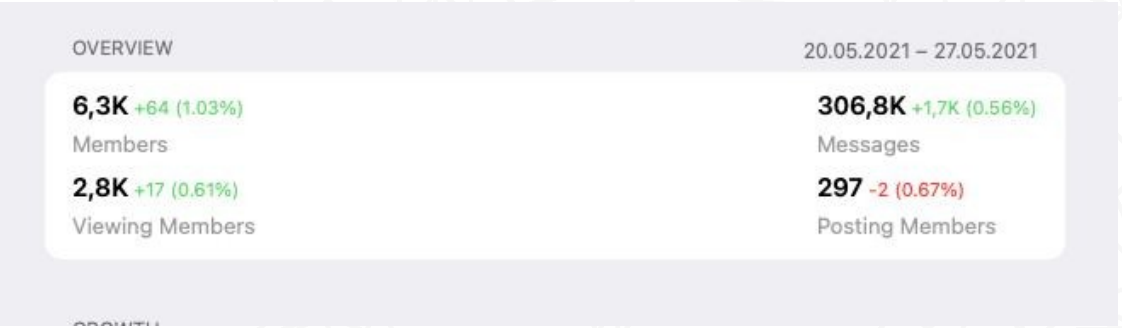
Most Loved, Dreaded, and Wanted Databases

Loved | Dreaded | Wanted

Redis	71.3%
PostgreSQL	69.6%
Elasticsearch	63.6%
Firebase	61.4%
MongoDB	59.5%
MariaDB	59.1%
Microsoft SQL Server	57.5%
SQLite	55.6%
DynamoDB	54.7%
MySQL	53.9%
Cassandra	47.2%
Oracle	38.2%
Couchbase	37.3%

Postgres в России

- Россия активно переходит на Postgres
  - Большие системы федерального уровня (ФНС, Минфин, Сбербанк, Роснефть...) системы с >100k tps, 100+ТБ
  - Крупнейшие интернет-компании (Yandex, Mail.ru, Avito)
- Очень активный телеграм-канал:
- https://t.me/pgsql - 6303 members



PostgresPro

Postgres в России

- Группа в Facebook "PostgreSQL в России"
- https://www.facebook.com/groups/postgresql/
  - 3,859 участников

PostgresPro

Postgres в России

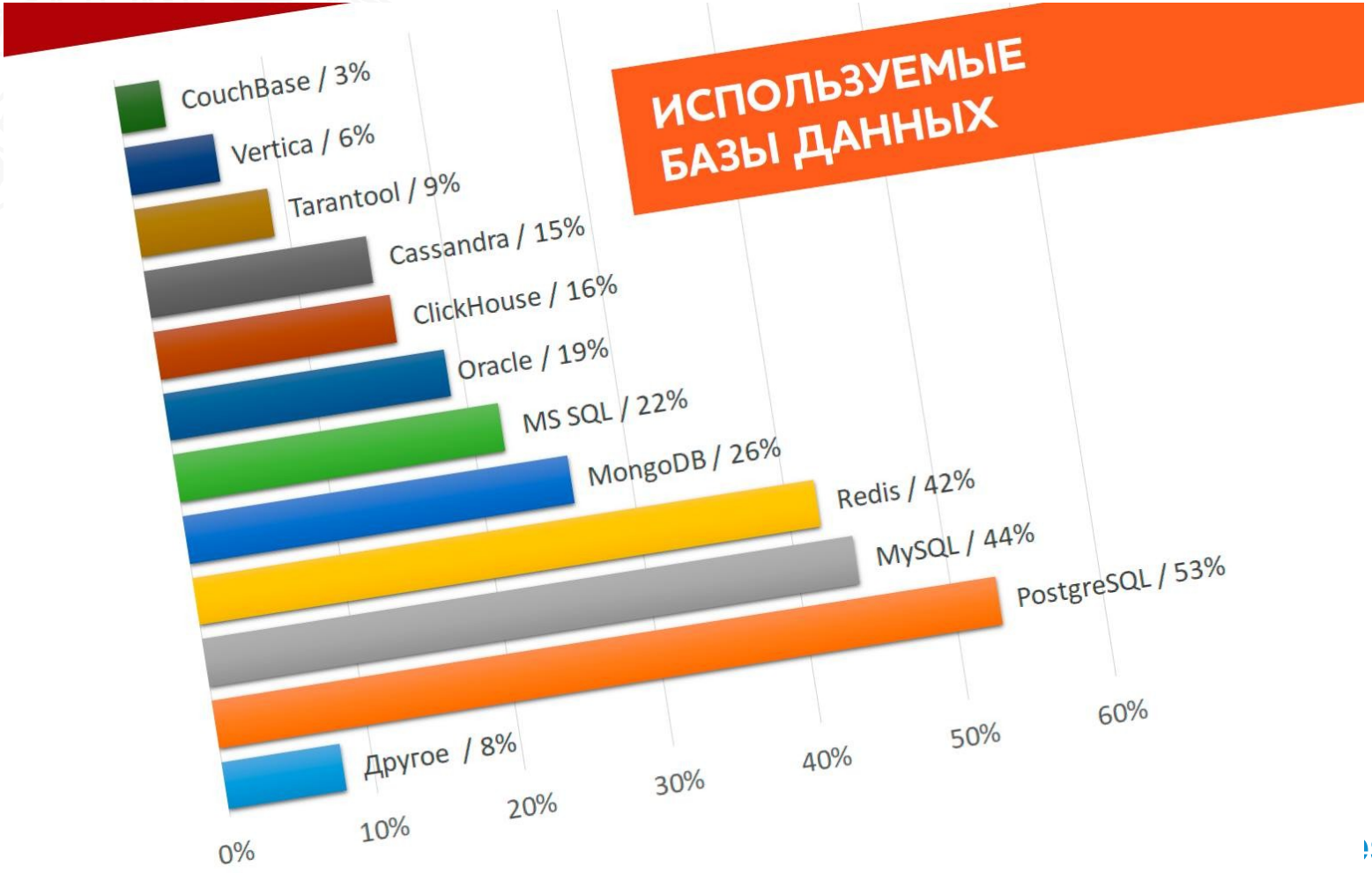
- Крупнейшие конференции в мире по Postgres (PGCONF.RU, региональные) + Митапы
- Российское сообщество представлено на всех больших международных конференциях

PostgresPro

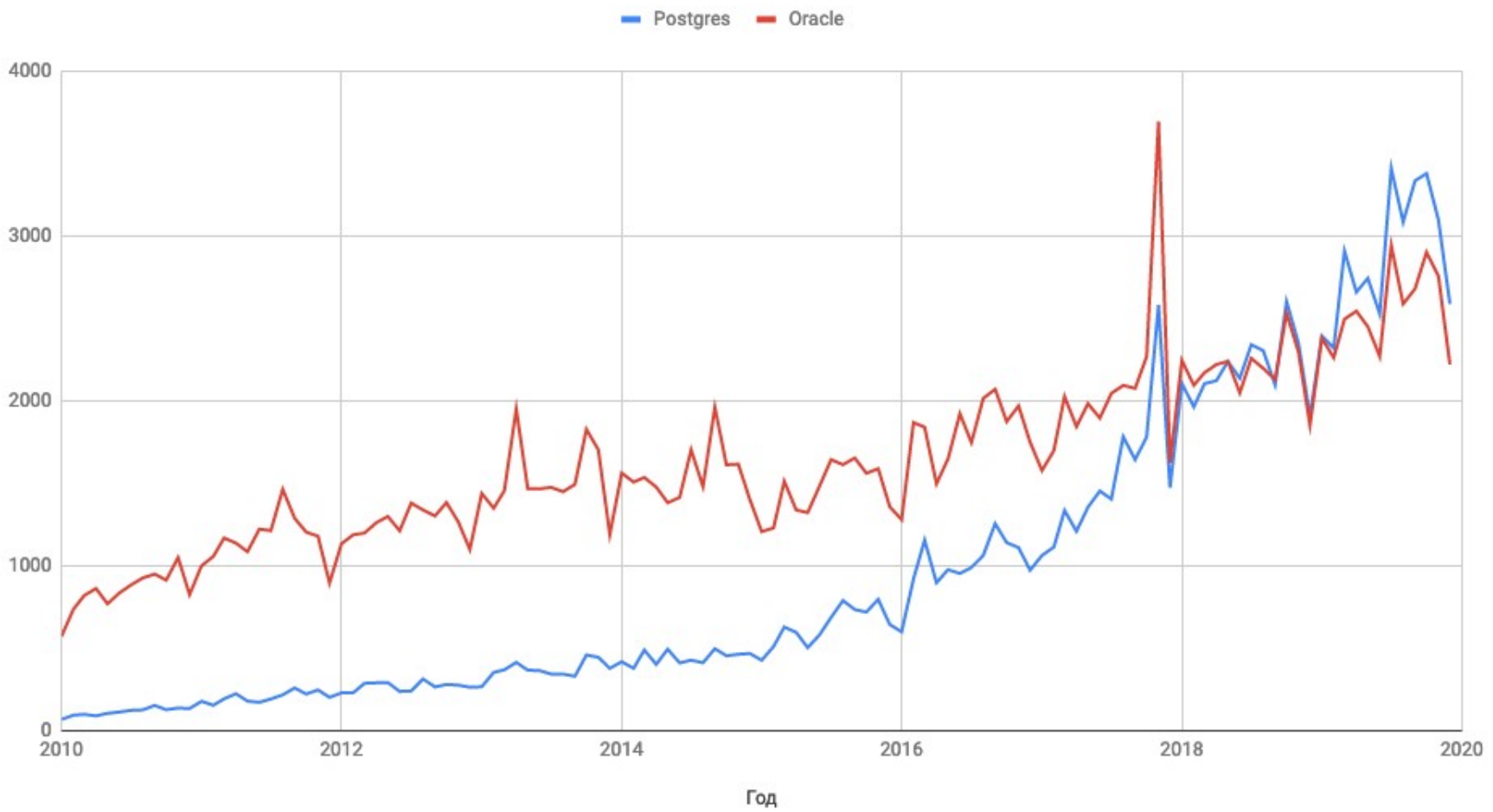
Postgres в России

- Выпущено два университетских курса по постгресу
- Учебные курсы DBA{1,2,3}, DEV, QPT
  https://postgrespro.ru/education
- Профессиональная сертификация специалистов
  https://postgrespro.ru/education/cert

По данным HighLoad:



Вакансии в России



Postgres Professional – российский разработчик СУБД

6 лет
на рынке с 2015 г.

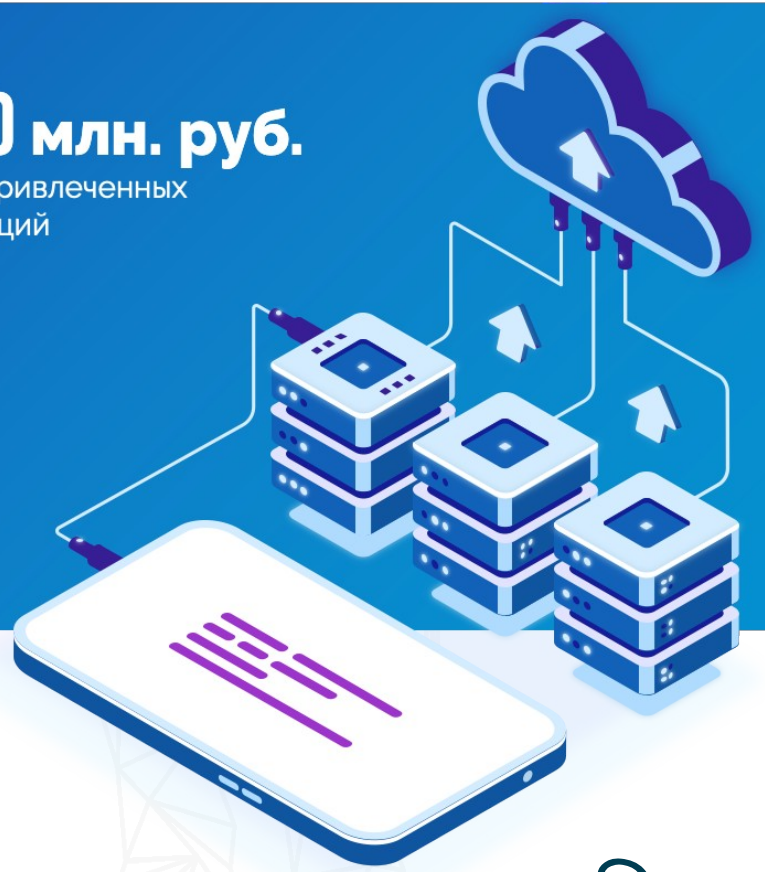
>20 лет
опыта в разработке PostgreSQL

>300 млн. руб.
объем привлеченных инвестиций

>100 чел.
штат компании

включая ведущих разработчиков (major contributor) PostgreSQL и **2х коммитеров** (committer), имеющих право вносить изменения в ядро PostgreSQL.

>100 патчей
вносят сотрудники компании в каждый релиз PostgreSQL



PostgresPro

СУБД Postgres Pro

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

PostgresPro

Postgres Pro Enterprise

Postgres Pro Enterprise

Система управления базами данных Postgres Pro для высоконагруженных систем крупных предприятий.

Подходит для систем:

до 10 000 | одновременно работающих пользователей

до 150 ТБ | размер базы данных

до 5 сек. | максимальная задержка при восстановлении работы кластера

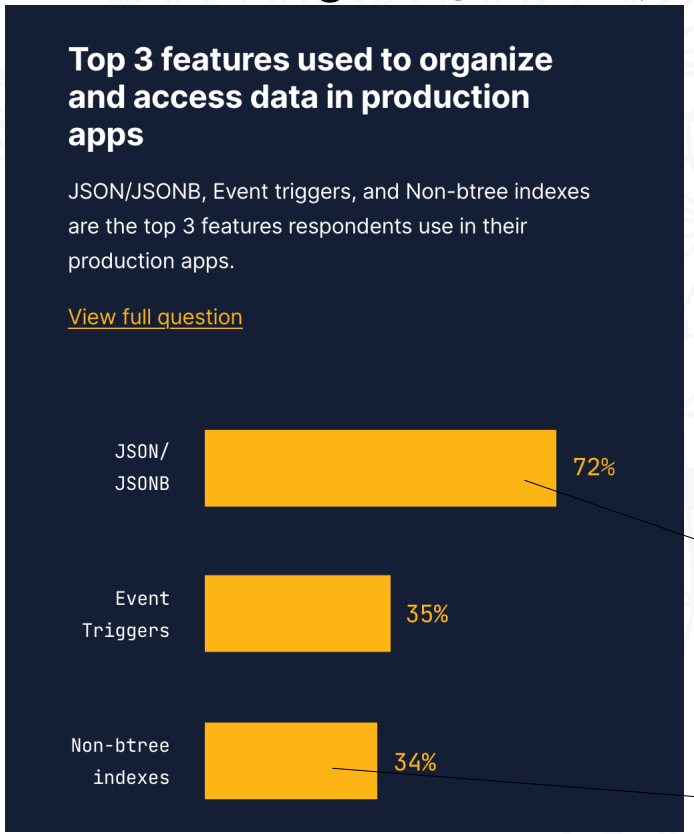
Пользователям СУБД Postgres Pro Enterprise 13 доступны следующие возможности:

- Обновление версии без остановки СУБД
- Встроенный пулер запросов
- Сжатие данных на уровне блоков
- Полнотекстовый индекс с ранжированием по дополнительному критерию
- Адаптивное планирование запросов
- Встроенный планировщик заданий
- Оптимизированное секционирование таблиц
- Автоматическая компиляция и планирование запросов
- Инкрементальный бэкап
- Модуль in-memory

PostgresPro

JSONB Popularity - CREATE TABLE qq (js JSONB)

State of PostgreSQL 2021 (Survey)



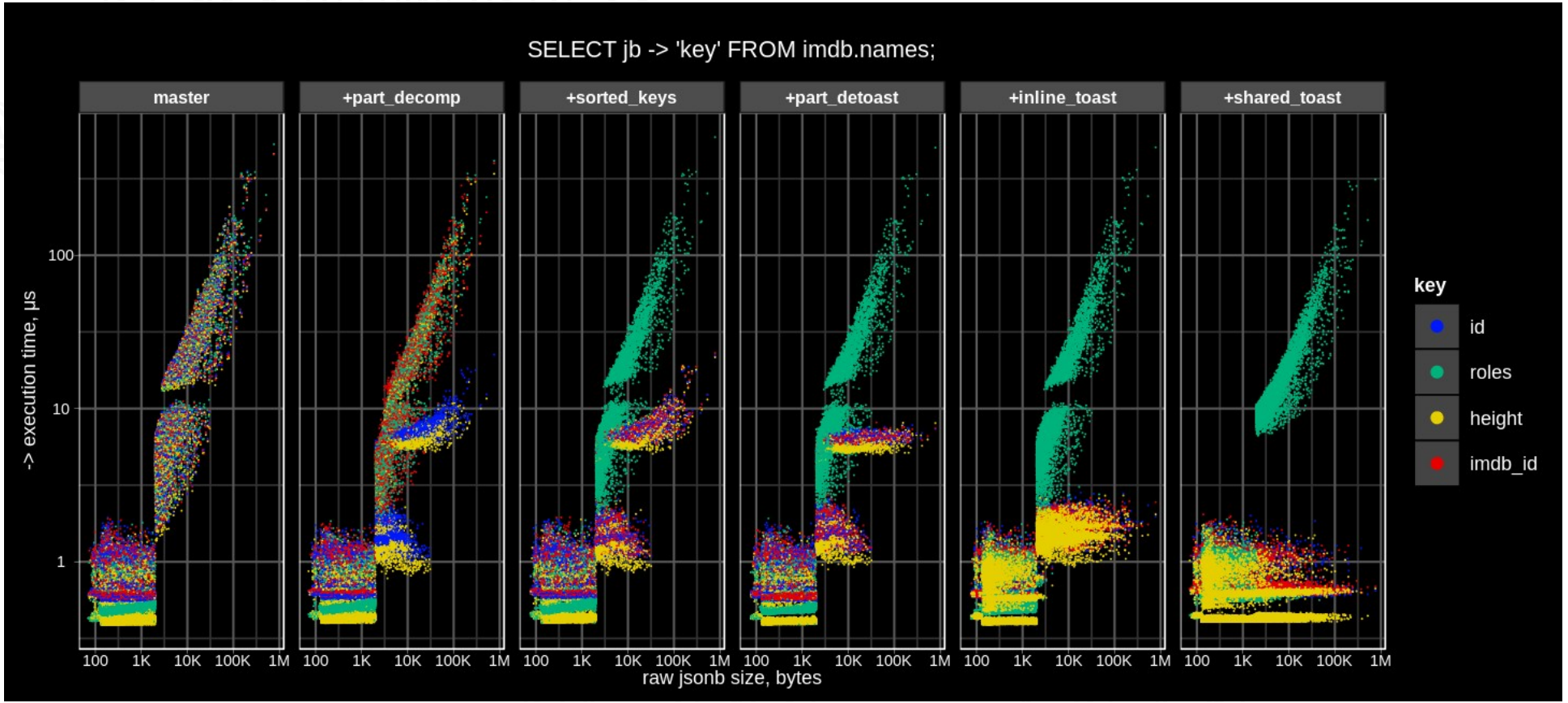
Pgsql telegram (6170) — 26.02.2021

- SELECT 8061/312083
- SQL 4473/144789
- JSON[B] 3116/88234
- TABLE 2997/129936
- JOIN 2345/108860
- INDEX 1519/74327
- BACKUP 1484/42618
- VACUUM 1470/53919
- REPLICA 707/31036

НАШ ВКЛАД !

PostgresPro

Step-by-step Jsonb improvements



PostgresPro

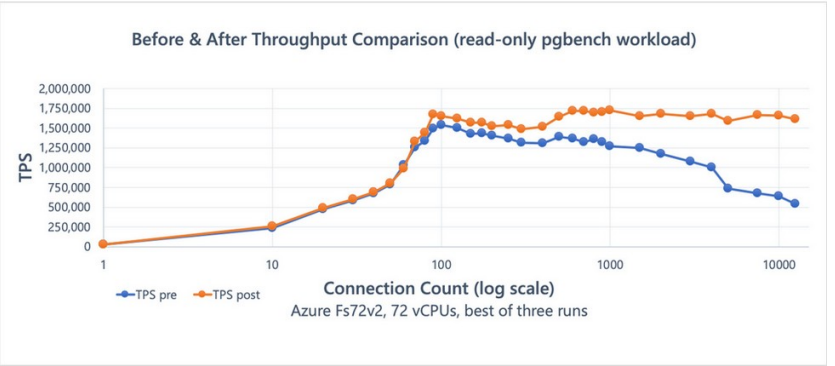
Сегодня: Кратко о Postgres 13

- Сжатие (дедупликация) B-Tree.
- Инкрементальная сортировка: sorted(k1,k2) → sorted(k1,k2,k3)
- PL/PgSQL не ходит в планировщик за простыми операторами
- Вычисление immutable функций на этапе планирования.
- Ускорение **`TRUNCATE`** (один скан shared_buffers вместо трех).
- Частичная декомпрессия TOAST.
- Вакуум индексов параллельно.
- Автовакуум при вставке (visibility map для append-only таблиц)

Подробнее: http://www.sai.msu.su/~megera/postgres/talks/pg13-tyumen.pdf
Хабр: Много ли нового в Чёртовой Дюжине?

PostgresPro

Повышение масштабируемости по коннектам



Облегчение GetSnapshotData()

Andres Freund:

[Analyzing the Limits of Connection Scalability in Postgres]

[Improving Postgres Connection Scalability: Snapshots]

PostgresPro

Восходящее удаление

(Bottom-up deletion)

Прежде чем делать split индексной страницы B-Tree при UPDATE, не меняющем ключ индекса, попытаться зачистить старые версии на ней.

Важная оптимизация при большом количестве UPDATE.

Асинхронный APPEND

Секционирование + FDW = Шардинг?

```
SELECT * FROM partitioned_table;


Append
    ->  Async Foreign Scan on partition1
    ........
```

Непонятно, когда планировщик изберёт такой scan.

PostgresPro

Покрывающие индексы SP-GiST

```
CREATE TABLE t (p point, z text);
CREATE INDEX i USING spgist ON (p)INCLUDE (z);
…
EXPLAIN SELECT * FROM t
    WHERE p <@ box(point(5,5),point(6,6));

Index Only Scan using i on t
    Index Cond: (p <@ '(6,6),(5,5)'::box)
```

PostgresPro

VACUUM

- Параметры по умолчанию:
  `vacuum_cost_page_miss = 10` → **2** **#диски стали быстрее**
- Умеет выполняться параллельно с **`CREATE INDEX | REINDEX CONCURRENTLY`**
- Умеет не заходить в индексы, если сканирование таблицы нашло мало bloat'a.
- **`VACUUM (PROCESS_TOAST OFF)`**
- Воркер autovacuum падает вслед за постмастером.

PostgresPro

Реплики и репликация (1)

- Изменилась реакция реплики на изменение критических параметров
- `max_connections`
- `max_prepared_transactions`
- `max_locks_per_transaction`
- `max_wal_senders`
- `max_worker_processes`

- Вместо остановки всей реплики теперь — остановка репликации с предупреждением.

PostgresPro

Реплики и репликация (2)

- Изменение `restore_command` не требует перезапуска.
- Для `pg_rewind` можно использовать реплику в качестве source.
- Логическая репликация:
  Передача больших транзакций постепенно.
  `CREATE PUBLICATION … WITH (streaming = on)`
- Логическая репликация в двоичном формате (так быстрее)
  `CREATE|ALTER SUBSCRIPTION … WITH (binary = on)`

PostgresPro

pg_stat_*

```
SET compute_query_id = 'on';
```

Заполняется поле `pg_stat_activity.query_id`
чем же, чем `pg_stat_statement.queryid`

- Новое представление:

```
pg_stat_statements_info (
    dealloc bigint, -- сколько раз переполнялось
    stats_reset timestamptz -- когда сбрасывали
)
```

PostgresPro

pg_stat_statements

- Различие верхнеуровневых и вложенных запросов:
- `SET pg_stat_statements.track = 'all'`
- ```
  SELECT toplevel, query
    FROM pg_stat_statements
  ```
- Подсчет обработанных строк в поле `rows` для **`CREATE TABLE AS`**, **`SELECT INTO`**, **`CREATE MATERIALIZED VIEW`**, **`FETCH`** (раньше не считалось)

PostgresPro
  ```

Новое представление pg_stat_wal

```
SET track_wal_io_timing = 'on';
SELECT * FROM pg_stat_wal;
wal_records       | 21049276
wal_fpi           | 79
wal_bytes         | 1242346187
wal_buffers_full  | 129358 раз
wal_write         | 129555 раз
wal_sync          | 239    раз
wal_write_time    | 337.535
wal_sync_time     | 409.48
stats_reset       | 2021-05-14 17:04:36.006022+03
```

full page images

Есть в pg_stat_statements

```
SELECT pg_stat_reset_shared('wal')
```

PostgresPro

ResultCache

Новый узел плана запроса. Кеширует результат внутреннего цикла в Nested Loop.

```
SET enable_resultcache = ON;

EXPLAIN (costs off)
    SELECT name FROM customer
    JOIN issue ON issue.customer = customer.id;

 Nested Loop
   ->  Seq Scan on issue
   ->  Result Cache
         Cache Key: issue.customer
         ->  Index Scan using customer_id_idx on customer
               Index Cond: (id = issue.customer)
```

PostgresPro

Перестройка индекса в другой tablespace

```
REINDEX (TABLESPACE space_to_move)
    TABLE CONCURRENTLY tablename;
```

Новый индекс создаётся в указанном tablespace.

Нужно, когда кончается место. Альтернативы:

ALTER INDEX … SET TABLESPACE …

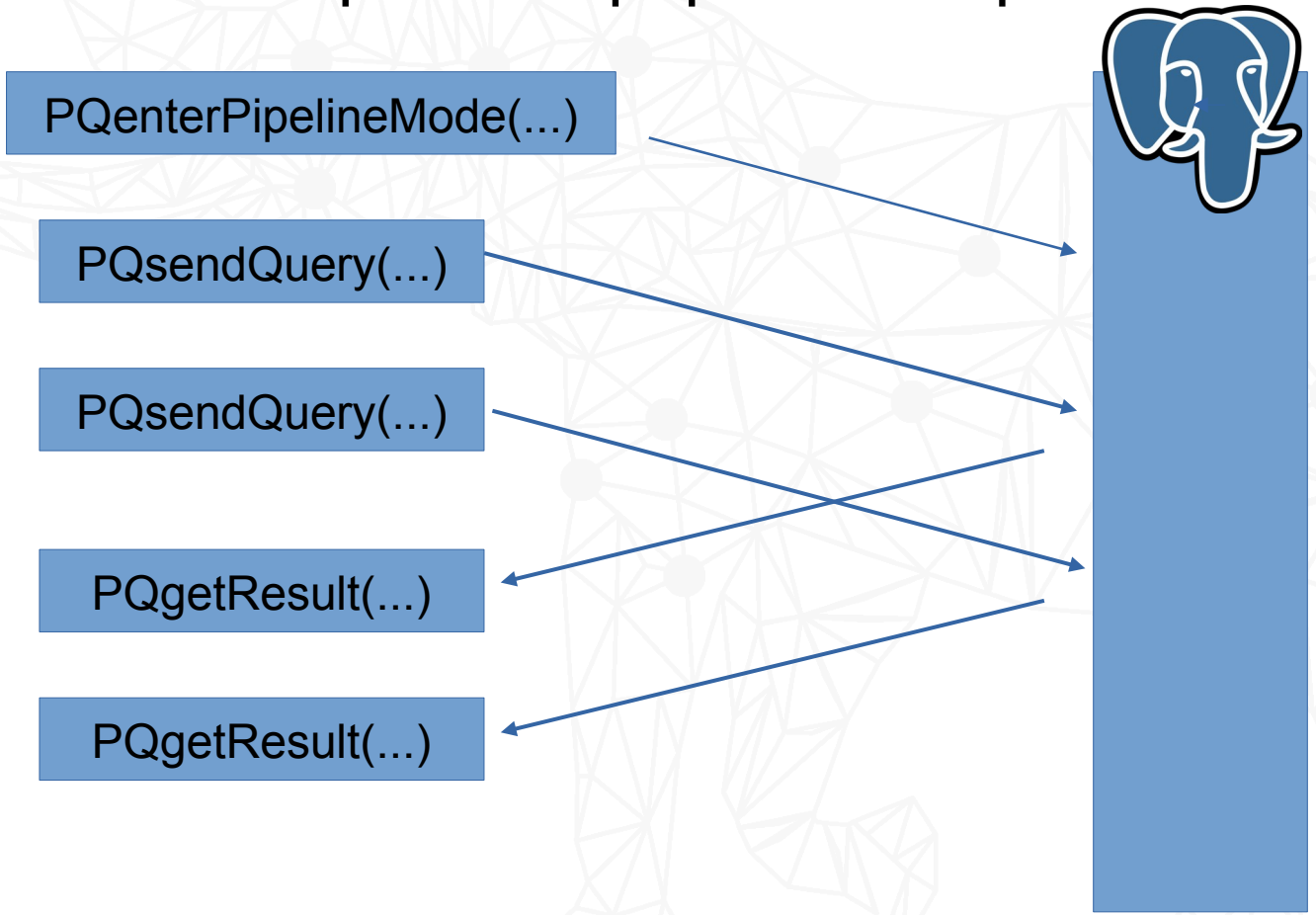
ALTER TABLE … SET TABLESPACE …

Расширение [pg_squeeze]

CYBERTEC

PostgresPro

Конвейерный (pipeline) режим в libpq



\+ помогает, если:

Много коротких команд, большие задержки в сети.

\–

Ждём поддержки в библиотеках более высокого уровня.

PostgresPro

FDW

- Новые параметры postgres_fdw:

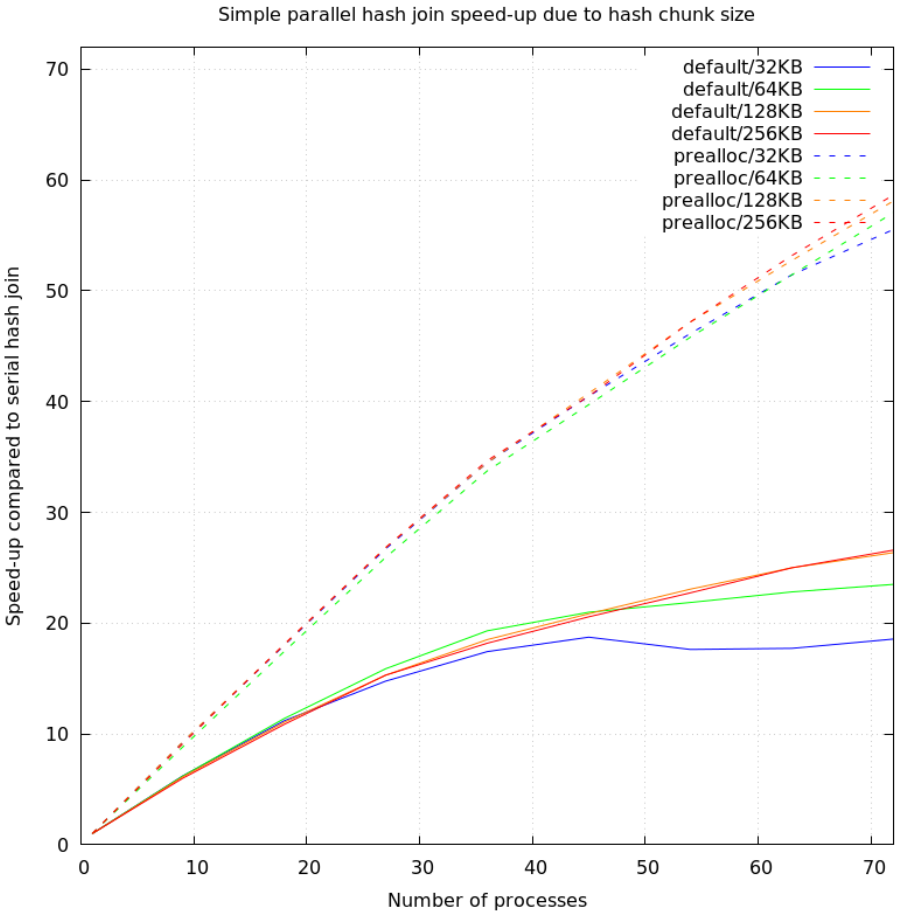
```
CREATE SERVER … FOREIGN DATA WRAPPER postgres_fdw
    OPTIONS (…  keep_connections 'off',
                batch_size '100');
```

- Автоматический реконнект к удаленному серверу (не в транзакции)

PostgresPro

Преаллокация shm для параллельного исполнения запросов

- GUC
  `min_dynamic_shared_memory`



PostgresPro

Статистика

- Статистика по выражениям на таблицах:
-
```
CREATE STATISTICS s
    ON some_func(r.field1, const1) FROM table r;
```
- Расширенная статистика лучше используется в OR-ах
- В пустой таблице 0 записей

CONCURRENTLY...

- **`ALTER TABLE … DETACH PARTITION … CONCURRENTLY;`**
- старые транзакции завершаются, видя эту секцию;
  новые уже не видят её, **но работают**.
- Можно одновременно выполнять
- **`CREATE INDEX CONCURRENTLY`** или **`REINDEX CONCURRENTLY`**
  по нескольким различным таблицам.

PostgresPro

Разное (1)

- Выбор метода компрессии для TOAST (pglz , lz4)

```
./configure –with-lz4
SET default_toast_compression = 'lz4';
ALTER TABLE tab ALTER data SET COMPRESSION lz4;
```

- **TRUNCATE** удалённую секцию таблицы (т. е. удаленную таблицу)
- **SET idle_session_timeout** = '1000ms';
- Ускорение зачистки shared buffers при **TRUNCATE, DROP TABLE** и т. п. стараемся обходиться без полного перебора
- now() кешируется в рамках транзакции.

PostgresPro

Разное (2)

- Использование атрибута компилятора `cold` в ветках с обработкой ошибок.
- Инкрементальная сортировка применяется для оконных функций
- Ускорение построения GiST индекса с помощью пространственного упорядочивания.
- REINDEX применим к секционированным таблицам.
- btree_gist стал PARALLEL_SAFE
- Parallel sequence scan работает пачками блоков, а не по одному.

Завтра: Ссылки о Postgres 14

На Хабре:

П. Лузанов

- PostgreSQL 14: Часть 1 или «июльский разогрев» (Коммитфест 2020-07)
- PostgreSQL 14: Часть 2 или «в тени тринадцатой» (Коммитфест 2020-09)
- PostgreSQL 14: Часть 3 или «ноябрьское затишье» (Коммитфест 2020-11)
- PostgreSQL 14: Часть 4 или «январское наступление» (Коммитфест 2021-01)
- PostgreSQL 14: Часть 5 или «весенние заморозки» (Коммитфест 2021-03)

Подробно: http://www.sai.msu.su/~megera/postgres/talks/Postgres14.pdf

PostgresPro

Слон идёт в облака
The future of Postgres

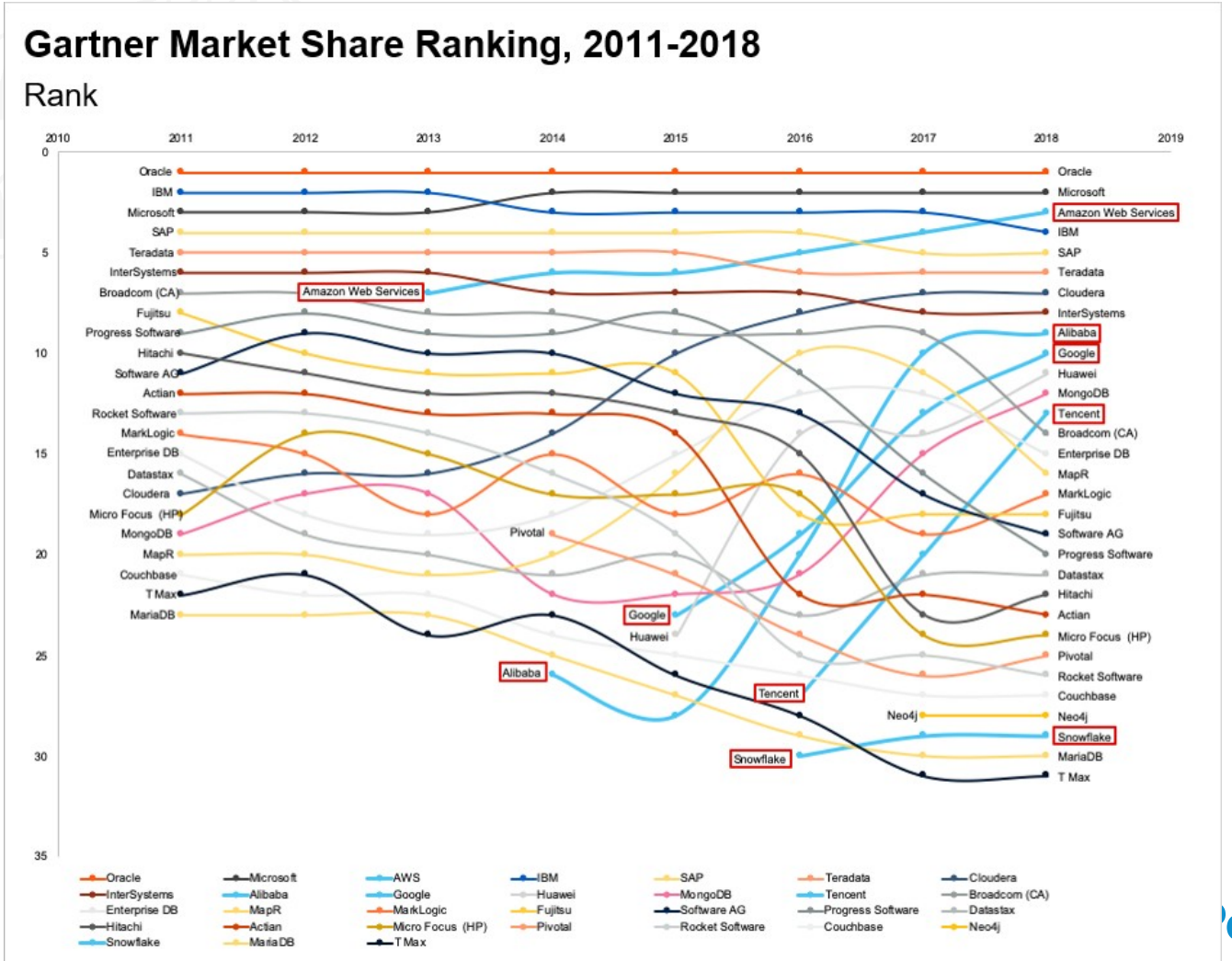
Gartner: The Future of Database Management Systems is Cloud!

- Облака — платформа по умолчанию для баз данных.
- К 2022 году 75% всех баз данных будут в облаках.
  - Облачные DBMS и DBaaS рынок вырастет с $12 млрд в 2020 до $24,8 млрд в 2025.
- 30% рост on-premise СУБД - следствие "DBMS Stockholm Syndrome".
  - Повышение цен, покупка дополнительных лицензий — много компаний привязаны лицензиями.
- Большинство инноваций делается сначала в облачных СУБД, а некоторые только для них (cloud-only, cloud first).
- Рост популярности облачной платформы уменьшит число производителей СУБД.

https://blogs.gartner.com/adam-ronthal/2019/06/23/future-database-management-systems-cloud/

PostgresPro

Выживут облака



Передний край и ближайшее будущее

- СУБД в виртуальных машинах
  - Amazon RDS (relational database service) — mysql, oracle, sql server, postgresql
  - Amazon Redshift (MPP, analytic workload, column-storage), based on Postgres 8.0.2
- Интеграция с облаком
  - Облачные СУБД (Amazon Aurora, AliCloud PolardDB)
  - Программируемые SSD
- Горячая тема:
  - Zero-administration - автономные, адаптивные СУБД
- World Wide Cloud — World Wide Database ?
- DBA не нужны ?

PostgresPro

Облачные вызовы

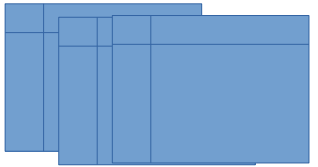
- Нулевое администрирование
  Адаптивный планировщик
  Бесшовный апгрейд
  Масштабируемость: шардинг, NUMA, многоядерность
- Multitenancy
- Разнообразные хранилища, Direct IO, больше NoSQL, поддержка Blockchain

PostgresPro

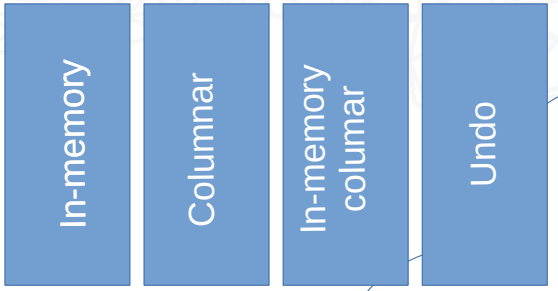
Будущий Postgres [Pro]

Builtin HA Sharded Cluster

Advanced GUI



Pluggable storages

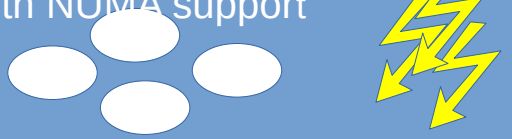


Cloud version

Postgres Adaptive Cloud

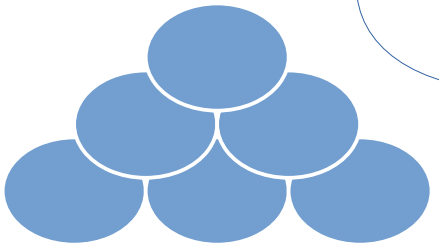
Multicore parallelism with NUMA support

JIT



RAW devices

Backup: full, incremental, partial



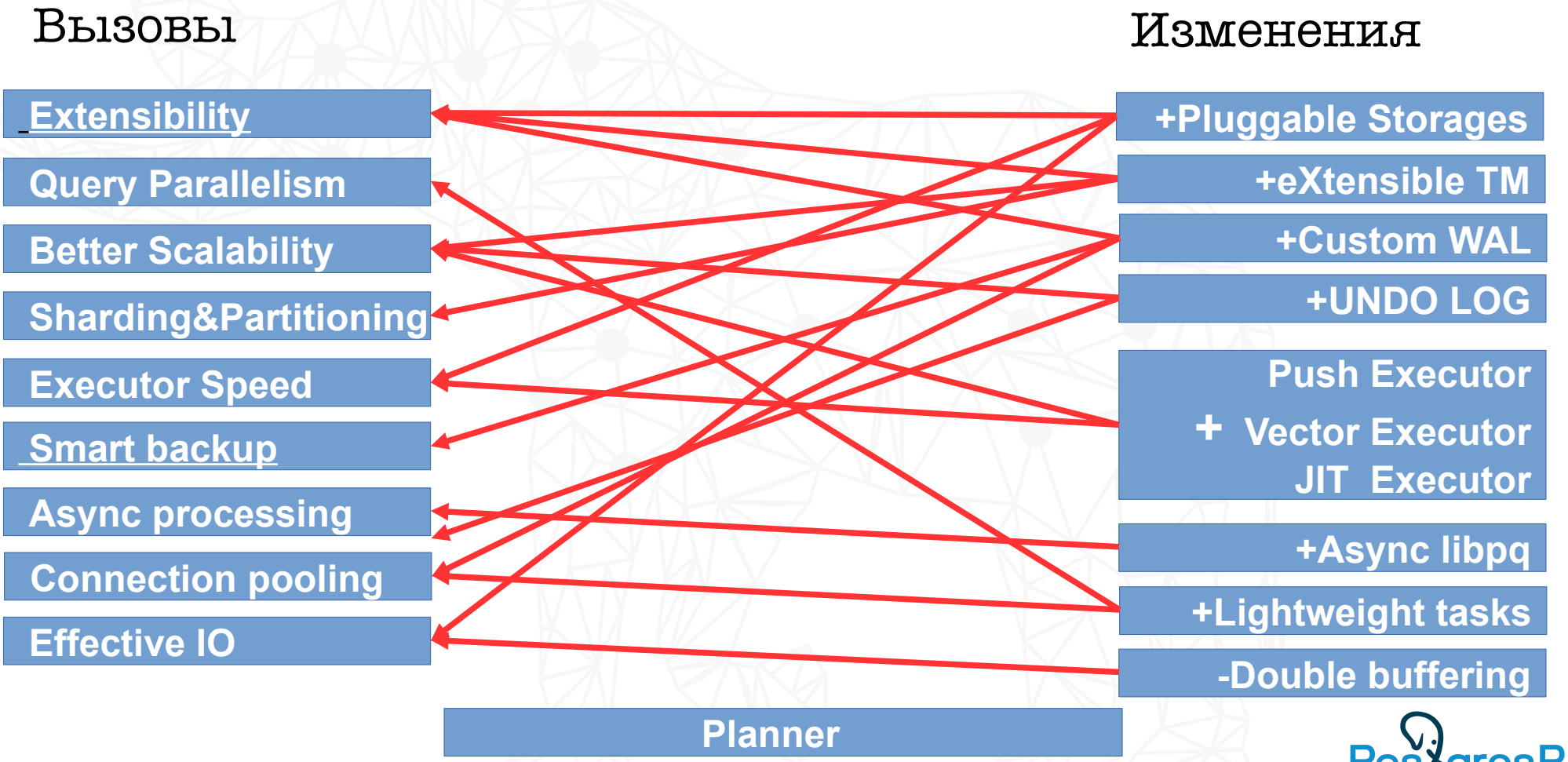
PostgresPro

Будущее в облаках!

- Вызовы разработчикам
- Вызовы опенсорсному сообществу
- Вызовы бизнесу

PostgresPro

Ветер перемен...



PostgresPro


ALL
YOU
POSTGRES
NEED
IS