

PostgreSQL 11 и далее: обзор новинок и тенденций

Иван Панченко



Чем хорош (интересен) Pg11

- 🐘 Параллелизм
- 🐘 JIT
- 🐘 Секционирование
- 🐘 Всякое-разное
- 🐘 Что не вошло?
 - 🐘 SQL/JSON
 - 🐘 Merge
 - 🐘

Общие обзоры Pg11

 http://sql-workbench.net/dbms_comparison.html

 [HP document](#)

 <https://habr.com/company/postgrespro/blog/353412/>

 [Русская дока на Pg11](#)

Параллелизм

- 🐘 Параллельное создание B-Tree индексов
- 🐘 Параллельный hash join
- 🐘 Ускорение параллельного seq scan
- 🐘 Параллельный Append (UNION ALL)
- 🐘 Параллельный SELECT при создании MAT VIEW

- 🐘 Ограниченная польза параллелизма при OLTP

<https://postgrespro.ru/docs/postgresql/11/parallel-plans>

Параллельный CREATE INDEX

🐘 B-Tree

🐘 Peter Geoghegan & Rushabh Lathia

`maintenance_work_mem` делится между воркерами

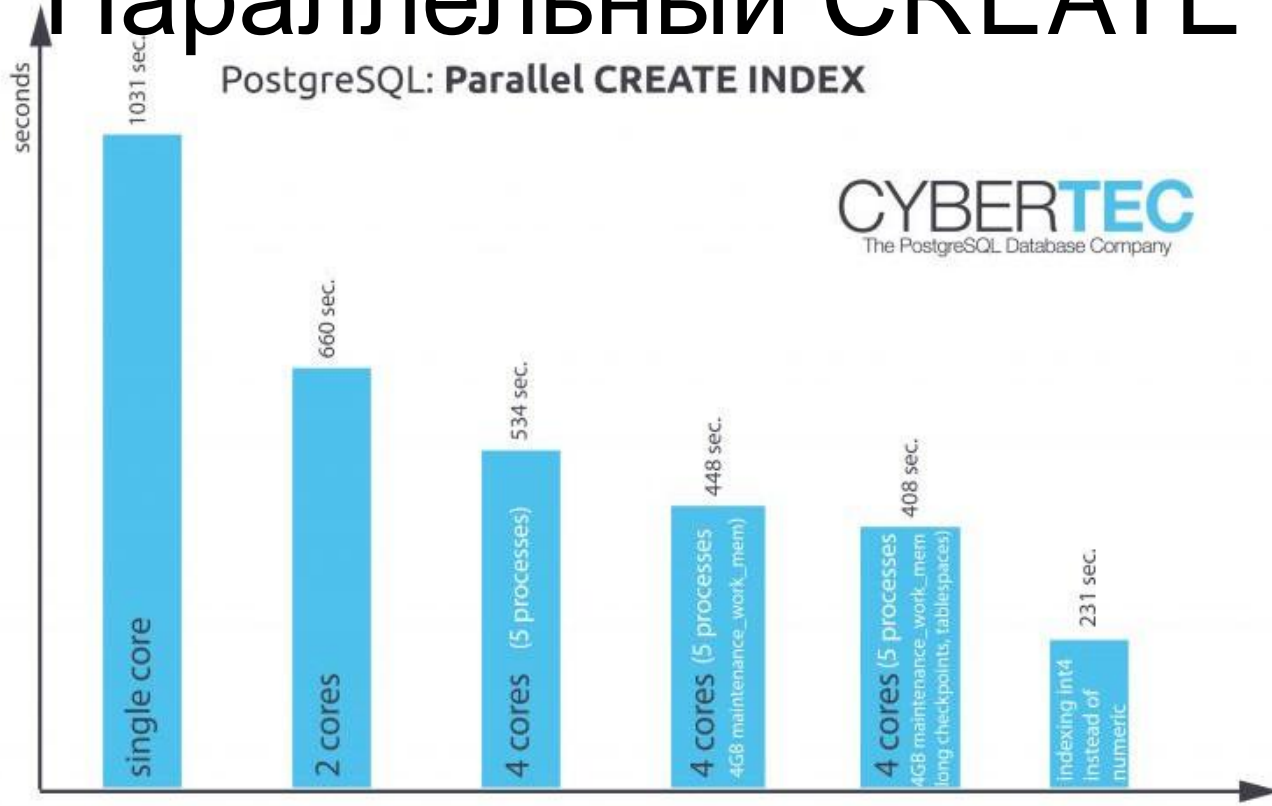
`max_parallel_maintenance_workers`

🐘 Количество воркеров – автоматическое
либо `ALTER TABLE set parallel_workers =`

...

🐘 Concurrently!

Параллельный CREATE INDEX



Параллельный CREATE INDEX

См. также:

[🔗 https://postgrespro.ru/docs/postgresql/11/sql-createindex](https://postgrespro.ru/docs/postgresql/11/sql-createindex)

[🔗 https://www.depesz.com/2018/02/12/waiting-for-postgresql-11-support-parallel-btree-index-builds/](https://www.depesz.com/2018/02/12/waiting-for-postgresql-11-support-parallel-btree-index-builds/)

Параллельный HASH JOIN

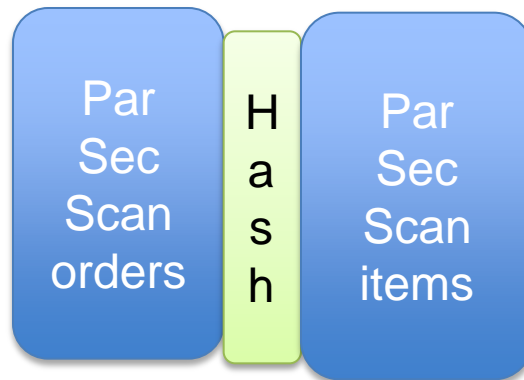
```
SELECT count(*)  
  FROM items  
  JOIN orders ON items."order" = orders.id
```

```
[ parallel] Hash Join  
  parallel Seq Scan items  
    [ parallel] Hash  
      [ parallel] Seq scan orders
```


Параллельный HASH JOIN



v.10

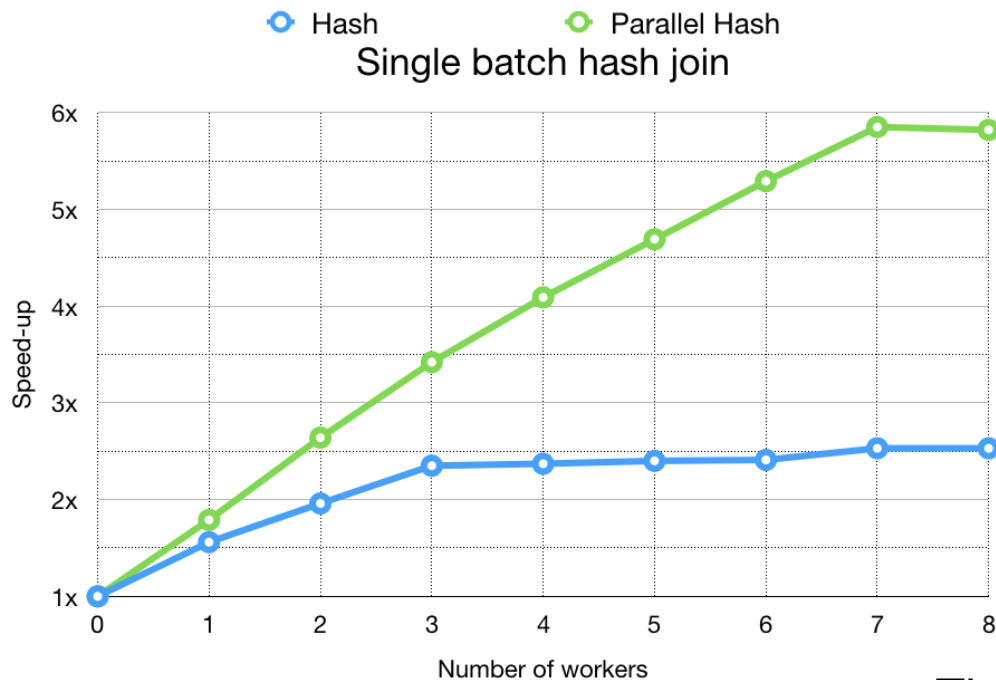


v.11

Thomas Munro, EDB

<https://write-skew.blogspot.com/2018/01/parallel-hash-for-postgresql.html>

Параллельный HASH JOIN



Thomas Munro, EDB

<https://write-skew.blogspot.com/2018/01/parallel-hash-for-postgresql.html>

Параллельный APPEND

 UNION ALL

 Amit Khandekar

JIT

JIT-ится (llvm):

👂 выражения внутри запроса

👂 парсинг дискового представления
записи

👂 `jit, jit_above_cost,
jit_inline_above_cost,
jit_optimize_above_cost`

<https://postgrespro.ru/docs/postgresql/11/jit>

JIT

Нужен clang-3.9 и llvm-3.9

```
./configure --enable-llvm
```

```
SET jit TO on;
```

```
EXPLAIN ...
```

```
JIT:
```

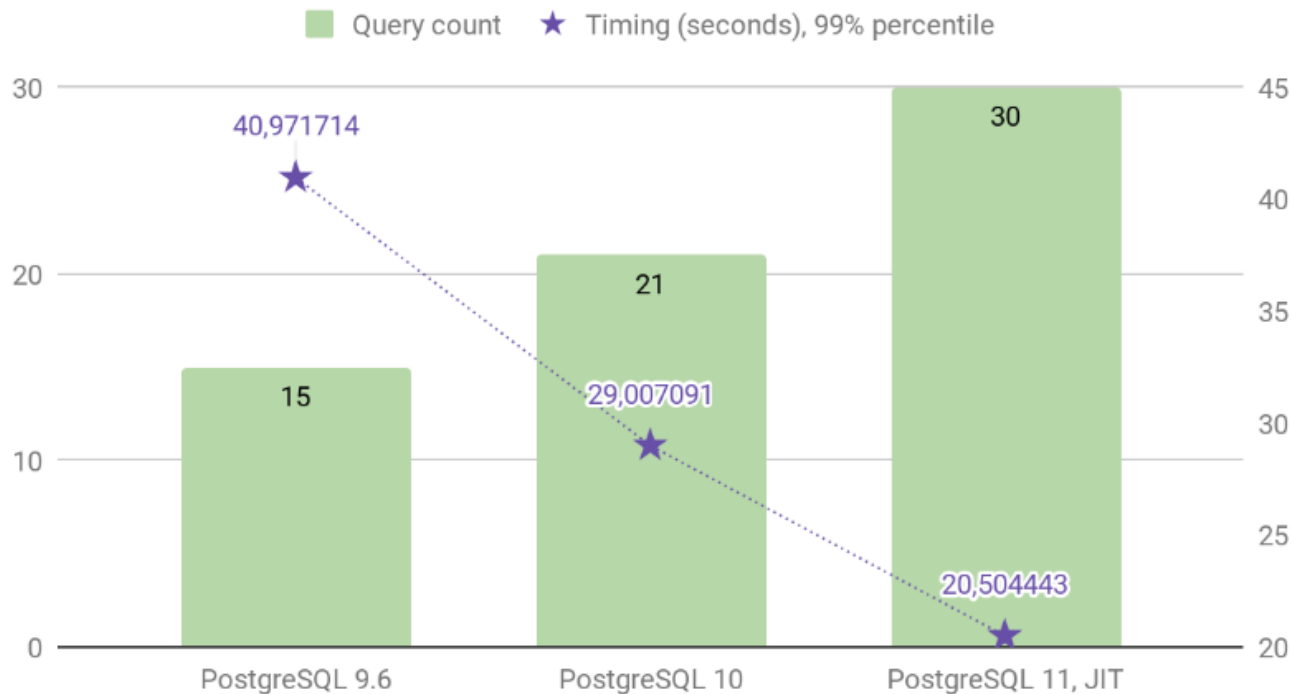
```
  Functions: 3
```

```
  Options: Inlining true, Optimization true, Expressions true,  
  Deforming true
```

```
  Timing: Generation 0.668 ms, Inlining 0.192 ms, Optimization  
  26.736 ms, Emission 14.394 ms, Total 41.990 ms
```

JIT

TPCH SF10 ; Q1 ; 10 mins ; c5.4xlarge



<https://www.citusdata.com/blog/2018/09/11/postgresql-11-just-in-time/>

Секционирование

- 🐘 Выбор партиций при JOIN (Ashutosh Bapat)
- 🐘 Cross-partition update (Amit Khandekar)
- 🐘 Hash partitioning (Amul Sul)
- 🐘 Constraint exclusion on partitioned tables (Ashutosh Bapat)
- 🐘 Local indexes for partitioned table (Álvaro Herrera, Maksim Milyutin)

Всякое-разное

 Процедуры

Всякое-разное

Процедуры

Польза не только для программистов:

Можно работать с большими данными
из хранимых процедур, мельча
транзакции

Всякое-разное


Include-индексы

```
CREATE INDEX ...ON ... () INCLUDE (flds)
```

<https://habr.com/company/postgrespro/blog/353126/>

(Анастасия Лубенникова)

Всякое-разное

 CREATE TRANSFORM для JSON(b)
PL/P{erl,ython}
(Антон Быков, Никита Глухов и др.)

Всякое-разное

 Быстрое добавление колонки с ненулевым дефолтом

```
ALTER TABLE ... ADD Col DEFAULT ....
```

дефолтное значение хранится в системном каталоге, а не вставляется физически

SQL/JSON (не вошло)



Элемент стандарта SQL/2016

4.46	JSON data handling in SQL.	174
4.46.1	Introduction.	174
4.46.2	Implied JSON data model.	175
4.46.3	SQL/JSON data model.	176
4.46.4	SQL/JSON functions.	177
4.46.5	Overview of SQL/JSON path language.	178
5	Lexical elements.	181
5.1	<SQL terminal character>.	181
5.2	<token> and <separator>.	185

SQL/JSON (не вошло)

- 🐘 Элемент стандарта SQL/2016
- 🐘 О.Бартунов, Н.Глухов
- 🐘 Операции с JSON (увы, без типов json, jsonb в стандарте)
- 🐘 Пока есть JQuery
<https://github.com/postgrespro/jquery>

MERGE (не вошло)

 Элемент стандарта SQL/2003

```
MERGE INTO table USING subquery  
  WHEN MATCHED THEN UPDATE ...  
  WHEN NOT MATCHED THEN INSERT ...
```

 Разработчик Simon Riggs

 С 9.5 есть

```
INSERT ON CONFLICT UPDATE
```

См. <https://habr.com/company/postgrespro/blog/412605/>

MERGE (не вошло)

В отличие от INSERT..UPDATE,

🐘 не требуется UNIQUE INDEX,

🐘 есть subquery

🐘 другая реализация

🐘 больше блокировок

См. <https://habr.com/company/postgrespro/blog/412605/>

И ССЫЛКИ ТАМ

Новые параллельные агрегаты (не вошло)

 string_agg, array_agg

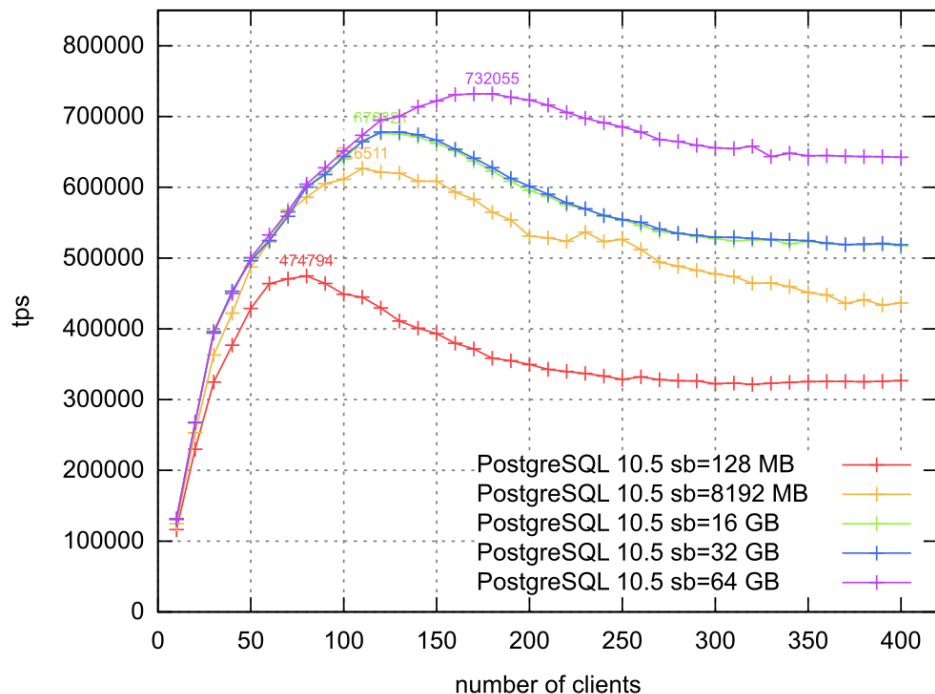
 David Rowley

 проблема в сортировке

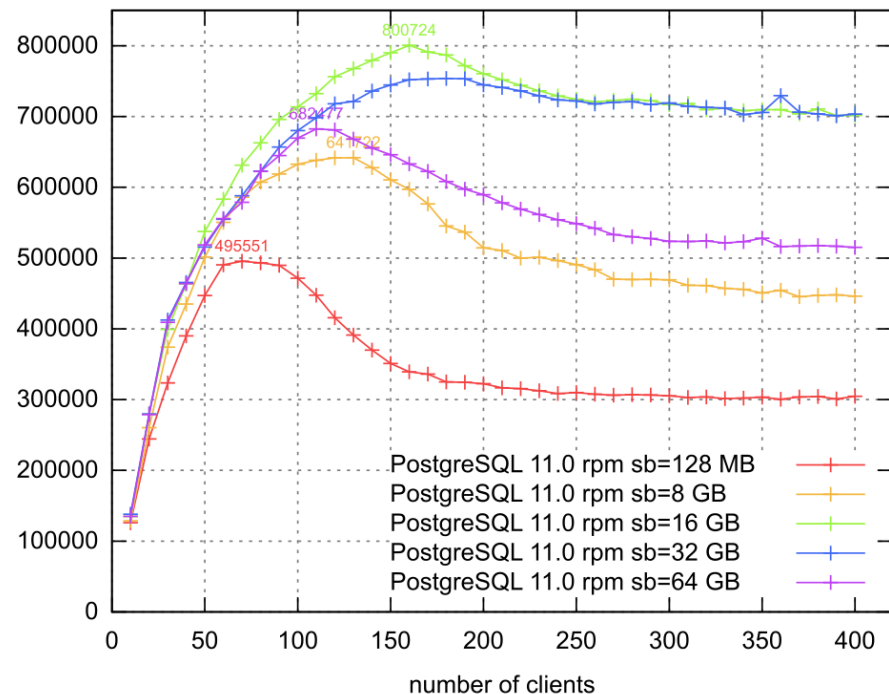
<https://commitfest.postgresql.org/18/1468/>

IBM Power 9

SMT=8 - transaction type: <builtin: select only>



SMT=8 <builtin: select only> tmpfs



Над чем идет работа в Postgres Pro

- 🔗 Обобщенная компрессия WAL
- 🔗 Предвычисление stable функций
<https://postgrespro.ru/list/id/ba261b9fc25dea4069d8ba9a8fcadf35@postgrespro.ru>
- 🔗 Ускорение логической репликации
- 🔗 Логическая репликация для 2PC
- 🔗 Кастомизируемые compression methods
- 🔗 Оптимизации JOIN
<https://commitfest.postgresql.org/19/1712/>
- 🔗 Глобальные (уникальные) индексы
- 🔗 Встроенный пул коннектов

Международная конференция по PostgreSQL

PgConf.Russia 2019
4-6 февраля



<https://pgconf.ru/>

Спасибо за внимание!

[Русская документация на Pg11](#)

Вопросы: info@postgrespro.ru

Вакансии: jobs@postgrespro.ru