Waiting for PG13

Олег Бартунов
МГУ, Postgres Professional

Postgres developer/contributor since 1995

When I started using Postgres

- No Slonik
- No UTF-8, even no 8-bit
- No WAL
- No MVCC
- No replication
- No usable non-scalar data types
- No subselects, no window functions, no CTE

- It was Postgres95

OLEG BARTUNOV

Postgres developer/contributor since 1995

OLEG BARTUNOV

PROFESSIONAL Postgres

Waiting for PG13

- Только фичи
- Багфиксов не будет
- Закомиченные фичи вероятно (99%) войдут в PG13
- Несбывшиеся надежды :(
- Возможные (50%) фичи из https://commitfest.postgresql.org/

Performance

- Ускорение PL/PGSQL (даже для r/w)
  "Avoid taking a new snapshot for an immutable simple expression in pl/pgsql."
  73b06cf893c9d3bb38c11878a12cc29407e78b6c

Implementation	time (ms)
PL/v8	41550
PL/Lua	32220
PL/pgSQL	19808
C/SPI	9406
SQL	7399
SQL (JIT)	5532
C/coreAPI	2873

Committed:

45% improvements:

```
while i < 10000000
    loop
      i := i + 1;
    end loop;
```

Предвычисление immutable functions

7266d0997dd2a0632da38a594c78e25ff21df67e

```
EXPLAIN (costs off) SELECT subject, ts_rank_cd(fts,q) AS rank FROM pglist,
to_tsquery('english','tuple') q  WHERE  fts @@ q ORDER BY rank LIMIT 10;
```

Всегда указывайте конфигурацию !

PG12

```
Limit
  ->  Sort
        Sort Key: (ts_rank_cd(pglist.fts, q.q))
        ->  Nested Loop
              ->  Function Scan on q
              ->  Bitmap Heap Scan on pglist
                    Recheck Cond: (fts @@ q.q)
                    ->  Bitmap Index Scan on pglist_fts_idx
                          Index Cond: (fts @@ q.q)
(9 rows)
```

PG13

```
Limit
  ->  Sort
        Sort Key: (ts_rank_cd(pglist.fts, '''tupl'''::tsquery))
        ->  Bitmap Heap Scan on pglist
              Recheck Cond: (fts @@ '''tupl'''::tsquery)
              ->  Bitmap Index Scan on pglist_fts_idx
                    Index Cond: (fts @@ '''tupl'''::tsquery)
(7 rows)
```

Предвычисление immutable functions

```
EXPLAIN (costs off) SELECT subject, ts_rank_cd(fts,q) AS rank FROM pglist,
to_tsquery('tuple') q  WHERE  fts @@ q ORDER BY rank LIMIT 10;
```

Для non-immutable функций оптимизация не работает

PG13

```
Limit
  ->  Sort
        Sort Key: (ts_rank_cd(pglist.fts, q.q))
        ->  Nested Loop
              ->  Function Scan on q
              ->  Bitmap Heap Scan on pglist
                    Recheck Cond: (fts @@ q.q)
                    ->  Bitmap Index Scan on pglist_fts_idx
                          Index Cond: (fts @@ q.q)
(9 rows)
```

PG13

```
Limit
  ->  Sort
        Sort Key: (ts_rank_cd(pglist.fts, '''tupl'''::tsquery))
        ->  Bitmap Heap Scan on pglist
              Recheck Cond: (fts @@ '''tupl'''::tsquery)
              ->  Bitmap Index Scan on pglist_fts_idx
                    Index Cond: (fts @@ '''tupl'''::tsquery)
(7 rows)
```

Avoid full index scan for GIN ('w' & '!star')

4b754d6c16e16cc1a1adf12ab0f48603069a0efd

ENTRY TREE

Report Index

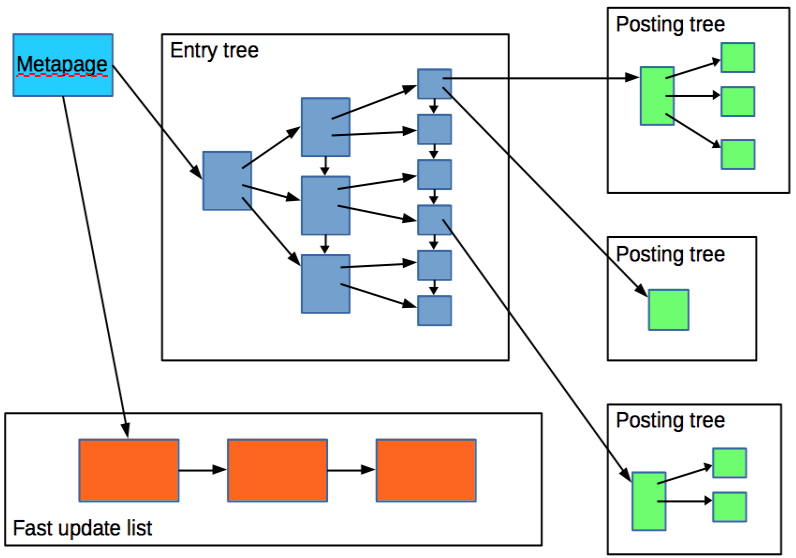
A

abrasives, 27
acceleration measurement, 58
accelerometers, 5, 10, 25, 28, 30, 36, 58, 59, 61, 73, 74
actuators, 4, 37, 46, 49
adaptive Kalman filters, 60, 61
adhesion, 63, 64
adhesive bonding, 15
adsorption, 44
aerodynamics, 29
aerospace instrumentation, 61
aerospace propulsion, 52
aerospace robotics, 68
aluminium, 17
amorphous state, 67
angular velocity measurement, 58
antenna phased arrays, 41, 46, 66
argon, 21
assembling, 22
atomic force microscopy, 13, 27, 35
atomic layer deposition, 15
attitude control, 60, 61
attitude measurement, 59, 61
automatic test equipment, 71
automatic testing, 24

B

backward wave oscillators, 45

compensation, 30, 68
compressive strength, 54
compressors, 29
computational fluid dynamics, 23, 29
computer games, 56
concurrent engineering, 14
contact resistance, 47, 66
convertors, 22
coplanar waveguide components, 40
Couette flow, 21
creep, 17
crystallisation, 64



Performance

- Ускорение jsonb функций
  abb014a63106653f2872a3b1662a79ab80d4dcbb
- Ускорение [AUTO]VACUUM, TRUNCATE - удаление буферов (MAIN, FSM, VM) за один проход
  6d05086c0a79e50d8e91ed953626ec7280cd2481)

```
128MB -      48s → 42s
  8GB —  5m 30s →  2m 26s
 24GB — 14m 13s →  5m 35s
```

Performance

- Масштабирование LISTEN/NOTIFY

  51004c7172b5c35afac050f4d5849839a06e8d3b

  Reduce signal traffic if listeners are spread among **many databases.**

  - Раньше NOTIFY рассылался (будил) на все бэкенды (возможно разных баз), которые анализировали его и все кроме одного удаляли сообщение, что приводило к значительным оверхедам.
  - Теперь NOTIFY шлется на бэкенды текущей базы и только на "протухшие" бэкенды из других баз.

Postgres PROFESSIONAL

Server

- Random UUID generation in core
  (gen_random_uuid ()from contrib/pgcrypto)
- Corruption error code
  (ERRCODE_DATA_CORRUPTED,ERRCODE_INDEX_CORRUPTED)
- Корректный подсчет транзакций
  (без parallel workers, backpatched)
- min/max для pg_lsn
- FTS для греческого языка (snowball stemmer), now 23 languages
- CREATE DATABASE .. LOCALE=.. LC_COLLATE= .. LC_CTYPE=..

Server

- ignore_invalid_pages GUC
- log_min_duration_sample/log_statement_sample_rate (log_min_duration_statement has higher priority)
- CREATE OR REPLACE VIEW — переименование колонок
- Parallel vacuum (indexes) - vacuum parallel #workers
  - One worker per index
  - index size must be > min_parallel_index_scan_size
  - #workers <= max_parallel_maintenance_workers

https://habr.com/ru/company/postgrespro/blog/486264/

.datetime in JSONPATH

- jsonb_path_*() - immutable and throw error for non-immutable comparisons, jsonb_path_*_tz() - stable and support operarions involving timezones.

```
SELECT  jsonb_path_query(
      '["12:34:00 +01", "12:35:00 +01", "12:36:00 +01", "12:35:00 +02", "12:35:00 -02", "10:35:00", "11:35:00",
       "12:35:00", "2017-03-10", "2017-03-10 12:35:00", "2017-03-10 12:35:00 +1"]',
      '$[*].datetime() ? (@ == "12:35 +1".datetime("HH24:MI TZH"))');
ERROR:  cannot convert value from time to timetz without timezone usage
HINT:  Use *_tz() function for timezone support.

SELECT  jsonb_path_query_tz(

 jsonb_path_query_tz
--------------------
 "12:35:00+01:00"
(1 row)
```

JSONPATH in PG13

`.datetime()` item method (T832) not supported in PG12

SQL/JSON feature	PG 12	PG 13	Oracle 18c	MySQL 8.0.4	SQL Server 2017
JSON PATH: 15	14/15	15/15	11/15	5/15	2/15

PostgreSQL 13 has complete **implementation** of JSON Path

Server

- gcd(), lmc()
- ANALYZE progress report
- ALTER STATISTICS [IF EXISTS] ... SET STATISTICS for EXTENDED STATISTICS
- DROP DATABASE … WITH (FORCE)
- jsonb_set_lax() - extra argument: 'use_json_null', 'raise_exceptions', 'return_target', 'delete_key'
- ALTER TABLE … ALTER COLUMN … DROP EXPRESSION *generated* column make *not generated*

psql etc

- \d pg_toast* - показ индексов и родительской таблицы (-E)
- Автодополнение CREATE TYPE, CREATE OR REPLACE
- Allow invisible PROMPT2 in psql - \set PROMPT2 '%w'

```
[local]:6666 postgres@postgres=# SELECT 1,
                                 2,
                                 3;
```

- Pgbench — partitioned account
  --partitions and --partition-method options
- Vacuumdb —parallel (parallel workers to process indexes)
- Add support for –jobs in reindexdb

Политкорректность

- Удаление master/slave терминологии. Теперь и в тестах
  replication: "slave" → "standby"
  partitioning: "master" → "root"
  "slave" → "leaf"
- "Virgin" → "pristine"

Несбывшиеся надежды :(

- Жизнь оказалась сложнее, надеемся на PG14
- Не будет Zheap — no-overwrite storage

  MVCC implementation:

  Oracle, MySQL, SQL Server: old versions are in other place

  MVCC in Postgres: all row versions are in table → table bloat, write amplification

- Не будет Zedstore — columnar storage with compression
  https://github.com/greenplum-db/postgres/tree/zedstore

To be commited… Server

- SQL/JSON functions — complete implementation of SQL/JSON standard (JSON_TABLE)
- Incremental Materialized View Maintenance
- Anycompatible (coalesce, greatest etc)
- INSERT .. SET ..
- REINDEX .. [TABLESPACE]
- CREATE OR **REPLACE** TRIGGER

PROFESSIONAL Postgres

To be commited… про скорость

- Incremental sort (a,b,c,d)
- GROUP BY optimization
- Remove self join
- Autoprepare
- KNN-Btree
- Btree deduplication

To be commited… про скорость

- Built-in connection pooler?
- Index Skip Scan ?
- Block-level parallel vacuum (relation)
- Ускорение drop/truncate table (replica)
- Hash group on disk

To be commited
Партицирование

- Join двух партиционированных таблиц
- Join партицированной и не партицированной таблицы
- Логическая репликация партиционированной таблицы

To be commited… интересно

- Row filtering for logical replication
- Generic type subscription
- Global temporary table
- Multi multiple extended statistics
- BRIN multi-range indexes (was bloom filter) ?
- Opclass parameters

To be committed
Важные мелочи

- Pgbench — pseudo-random permutation
- pgrename_temp()
- Shared-memory stats collector
- Page number in error context of vacuum


ALL
YOU
POSTGRES
NEED
IS