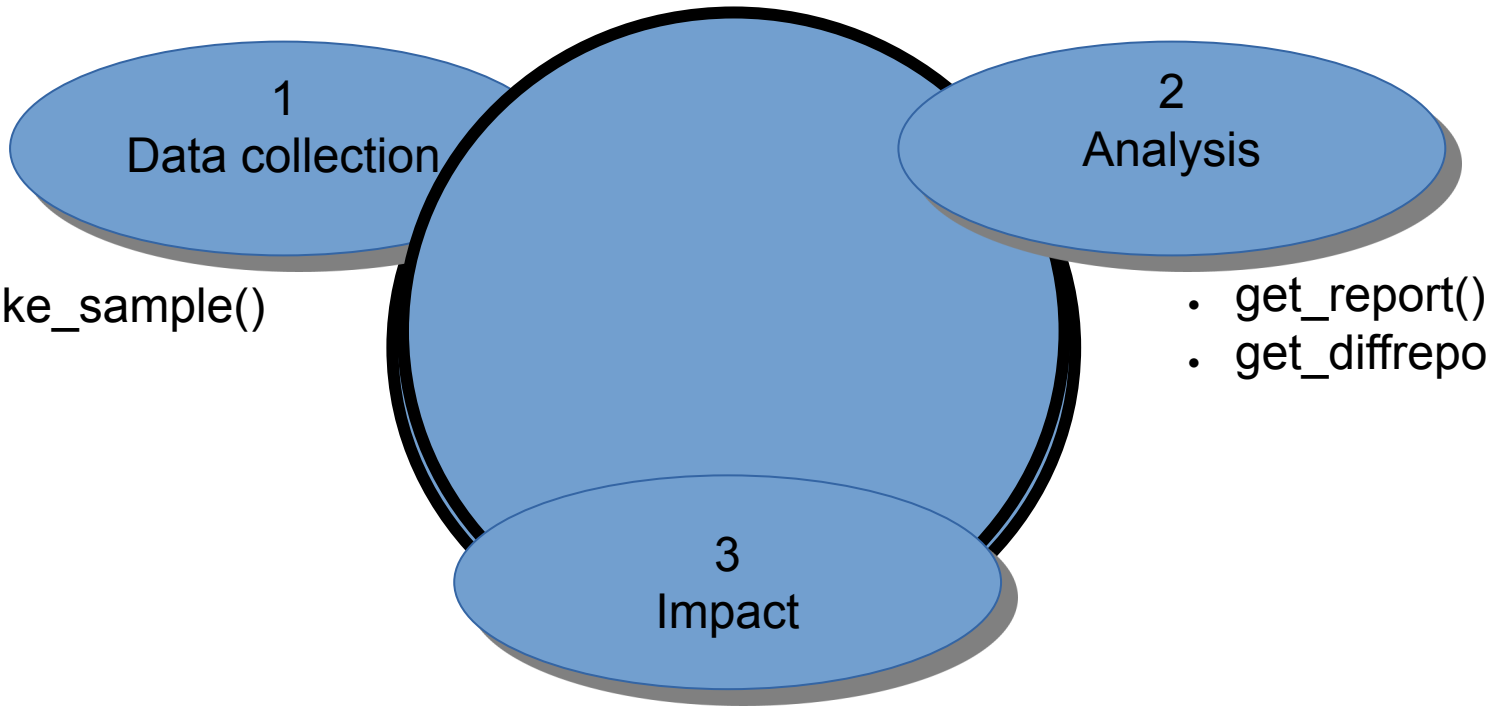
New features of pg_profile/pgpro_pwr – historical workload profiler

Andrey Zubkov,
Senior DBA (Postgres Professional)

PostgresPro

Performance Optimization



- take_sample()

- get_report()
- get_diffreport()

PostgresPro

Other Utilities

There are many great utilities out there:

- pgBadger
- pgcenter
- pgCluu
- postgres_checkup
- PoWA

PostgresPro

Flight Recorder

Expressions

- pg_stat_statements
- pg_stat_kcache
- pgrpo_stats

Tables

- pg_statio_all_tables
- pg_stat_all_tables

Indexes

- pg_statio_all_tables
- pg_stat_all_tables

Functions

- pg_stat_user_functions

Vacuum

- pg_stat_all_tables
- pg_stat_all_indexes

Cluster

- pg_stat_database
- pg_stat_tablespace
- pg_stat_bgwriter
- pg_settings
- adminisrative fuctions

PostgresPro

Mode of Operation

- Let's say we have a table...

```
postgres=# CREATE TABLE data (
    section_id     integer,
    row_data       text
);
```

PostgresPro

Mode of Operation

- Let's look at some of its statistics:

```
postgres=# SELECT
  relname,seq_scan,seq_tup_read,heap_blks_read,
  heap_blks_hit,toast_blks_read,toast_blks_hit
FROM pg_stat_all_tables join pg_statio_all_tables
  using (relid,schemaname,relname)
WHERE relname='data';

-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 87
seq_tup_read    | 292494
heap_blks_read  | 384
heap_blks_hit   | 16704
toast_blks_read | 1
toast_blks_hit  | 6
```

PostgresPro

Mode of Operation

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 87
seq_tup_read    | 292494
heap_blks_read  | 384
heap_blks_hit   | 16704
toast_blks_read | 1
toast_blks_hit  | 6
```

Mode of Operation

- We see changes in statistics.

Executing a query that requires a full scan:

```
postgres=# SELECT count(*) FROM data;
-[ RECORD 1 ]
count | 3362
```

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 88      (+1)
seq_tup_read    | 295856  (+3362)
heap_blks_read  | 384
heap_blks_hit   | 16896   (+192)
toast_blks_read | 1
toast_blks_hit  | 6
```

PostgresPro

Mode of Operation

- We see changes in statistics.

Executing a query that requires a full scan:

```
postgres=# SELECT count(*) FROM data;
-[ RECORD 1 ]
count | 3362
```

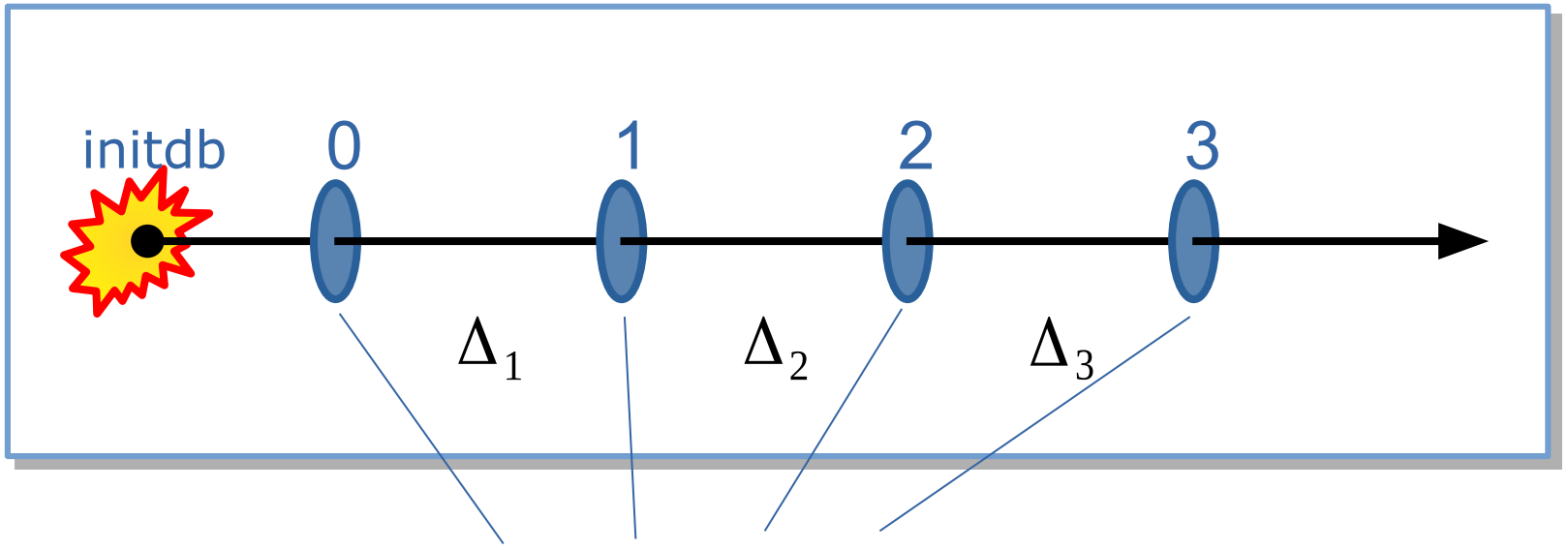
Query requiring a TOAST read:

```
postgres=# SELECT sum(length(row_data)) FROM data;
-[ RECORD 1 ]
sum | 1408526
```

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 89      (+1)
seq_tup_read    | 299218  (+3362)
heap_blks_read  | 384
heap_blks_hit   | 17088   (+192)
toast_blks_read | 2       (+1)
toast_blks_hit  | 6
```

PostgresPro

Mode of Operation

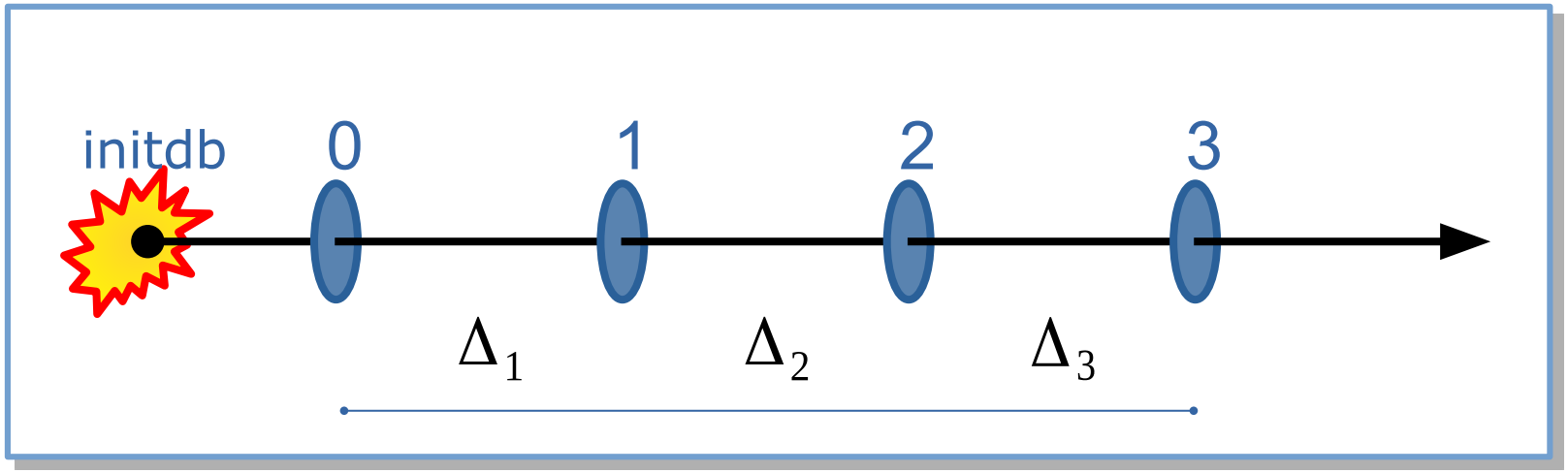


cron:

```
*/30 * * * *   psql -c 'SELECT take_sample()'
```

PostgresPro

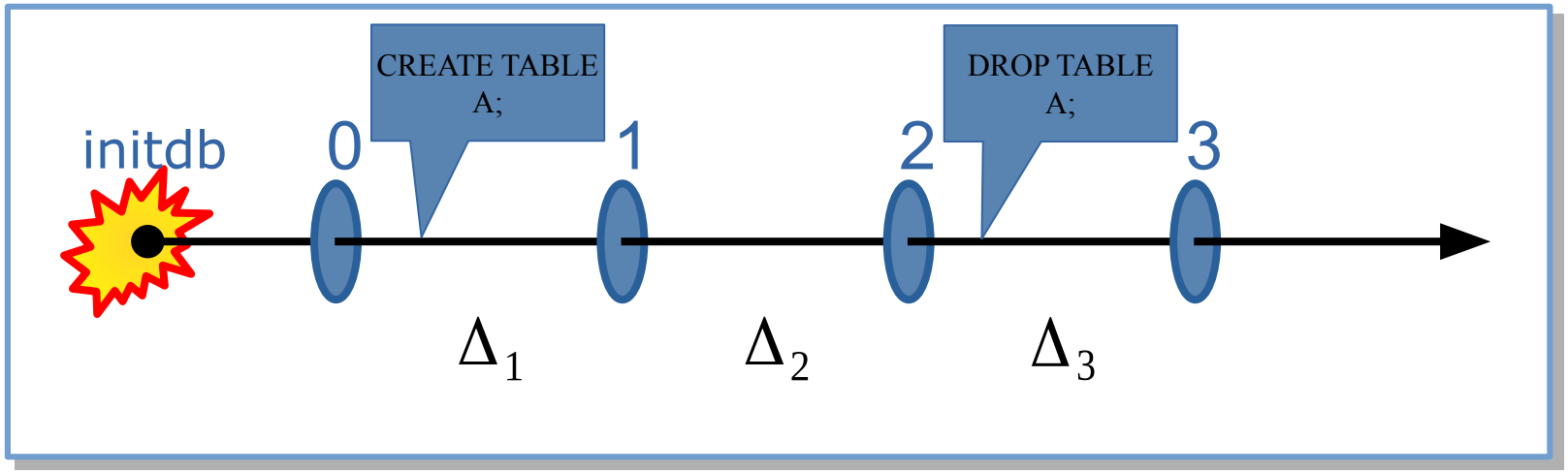
Mode of Operation



$$\Delta_{0-3} = \sum_{i=1}^{3} \Delta_i$$

PostgresPro

Mode of Operation



$$\Delta_{0-3} = \sum_{i=1}^{3} \Delta_i$$

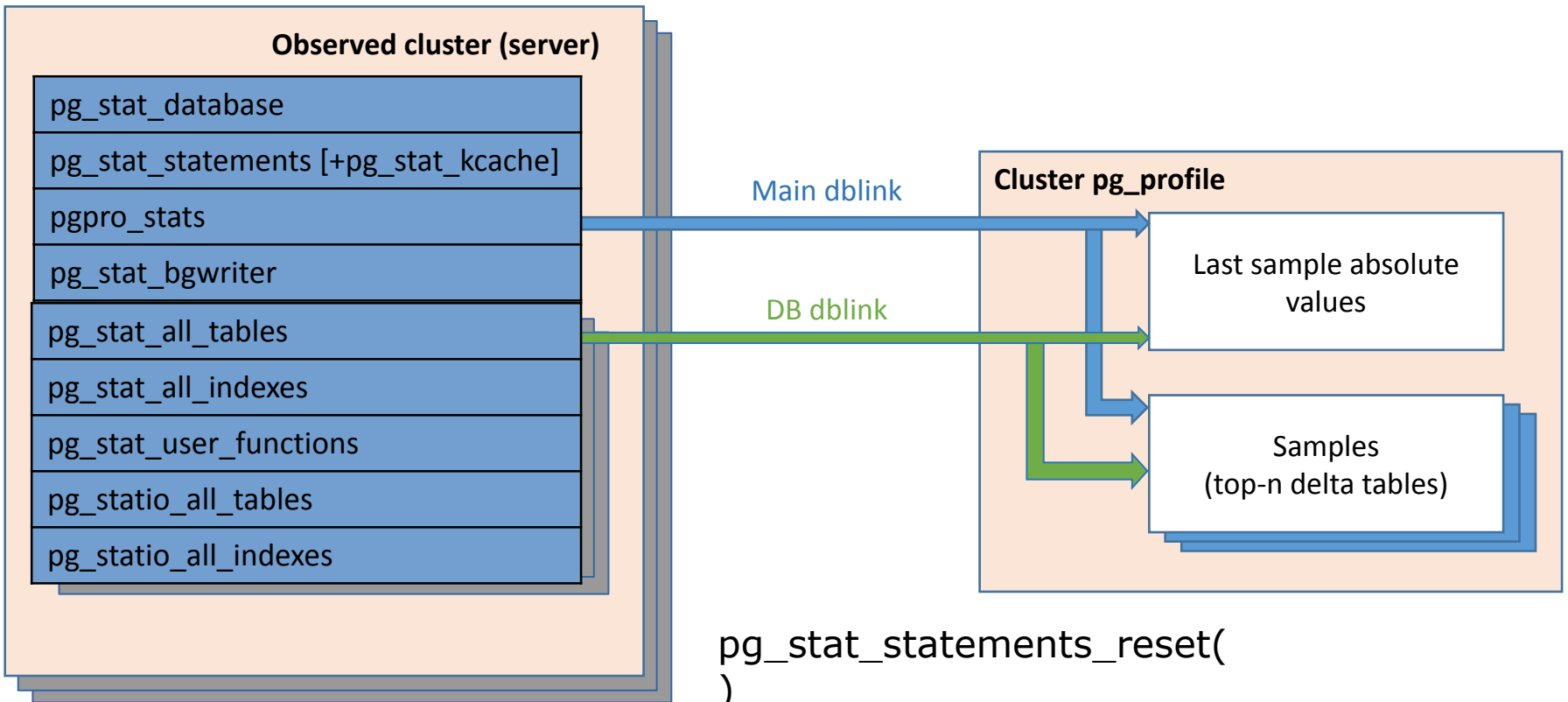
PostgresPro

Architecture

Pl/pgsql extension consisting of:

- Samples Storage Repositories
- Samples collection functions — *take_sample()*
- Report generation functions — *get_report()*
- Data transfer functions —export_data(), import_data()

Sample



pg_stat_statements_reset(
)

PostgresPro

Examples

Report generation:

```
$ psql -Aqtc "SELECT profile.get_report(480,482)" -o report_480_482.html
$ psql -Aqtc "select profile.get_report(
    tstzrange('2020-05-13 11:51:35+03','2020-05-13 11:52:18+03'))"
    -o report_range.html
```

- Troubleshooting of WAL generation WAL Report
- Report 0.3.1 Report Diff report

PostgresPro

Implicit Vacuum Load

Top indexes by estimated vacuum I/O load

DB	Tablespace	Schema	Table	Index	~Vacuum bytes	Vacuum cnt	Autovacuum cnt
demo	pg_default	tiger	small_table	idx1	191 GB		621
demo	pg_default	tiger	small_table	pk_small_table	131 GB		621
demo	pg_default	tiger	small_table	idx2	130 GB		621
demo	pg_default	tiger	small_table	idx3	127 GB		621

- Table size: approximately 700 KB
- Index sizes: 200 - 300 MB

PostgresPro

Daily Collection of Relation Sizes

- Relation size measuring fuction requires AccessShareExclusive lock.
- Collection of database-wide relation sizes may take considerable amount of time.
- Relation sizes may be not needed in every sample.

Daily Collection of Relation Sizes

```
set_server_size_sampling(
'local',                -- server
'23:00+03',             -- daily window start
interval '2 hour',      -- daily window duration
interval '8 hour'       -- minimal gap
);
```

PostgresPro

Daily Collection of Relation Sizes

```
set_server_size_sampling(
'local',               -- server
'23:00+03',            -- daily window start
interval '2 hour',     -- daily window duration
interval '8 hour'      -- minimal gap
);
```

PostgresPro

Data Export and Import

Exporting repository data:

```
export_data(
    server name,
    min_sample_id,
    max_sample_id,
    obfuscate_queries
)
RETURNS TABLE (section_id bigint, row_data json)
```

```
postgres=# \copy (select * from export_data()) to 'export.csv'
```

PostgresPro

Data Export and Import

Importing data to a repository:

```
import_data(data regclass)

postgres=# CREATE TABLE import (
    section_id bigint,
    row_data json
);
CREATE TABLE
postgres=# \copy import from 'export.csv'
COPY 6437
postgres=# SELECT * FROM import_data('import');
```

PostgresPro

Advanced PWR Features

- Expression statistics by plan
- Expression expectation statistics

- Report sample

PostgresPro

Complexities

- Measuring relation sizes
- Report encoding
- Common problems with Postgres statistics

PostgresPro

Thank You for Attention!

pg_profile:
https://github.com/zubkov-andrei/pg_profile
pgpro_pwr:
https://postgrespro.com/docs/postgrespro/13/pgpro-pwr

Andrey Zubkov,

Postgres Professional

a.zubkov@postgrespro.ru