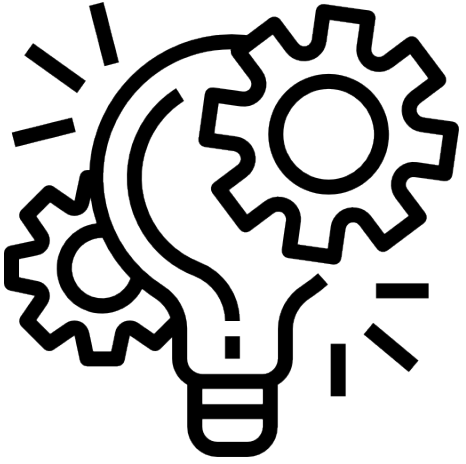
ZABBIX '25
CONFERENCE
GERMANY

Migrate Zabbix DB from MariaDB/MySQL to PostgreSQL with TimescaleDB

Wolfgang Alper

CEO
IntelliTrend GmbH

What is TimeScaleDB?



What is TimeScaleDB?

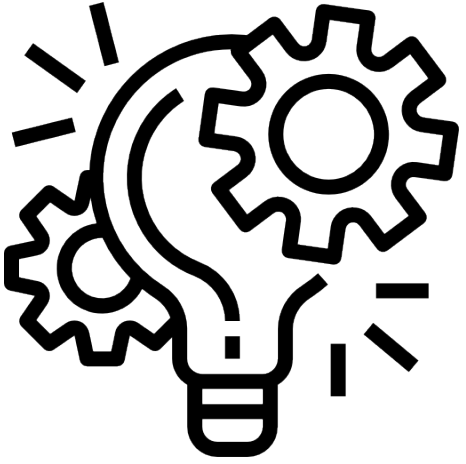
TimescaleDB is an open-source time series database developed by Timescale Inc., built as an **extension** on top of **PostgreSQL**.

TimescaleDB extends PostgreSQL with custom data structures and optimizations for **time-series data**, such as **hypertables** and **chunks,** providing **native partitioning**.

One of the key features are is its read- and write **performance** when working with **time-series data** and its ability to **compress** this type of data.

TimescaleDB is offered under the **Apache 2.0 license**. Additional features are offered in a community edition under the **Timescale License Agreement**.

Why TimeScaleDB?



Migrate to PostgreSQL with TimeScaleDB

Motivation

- **Smaller data volume**: History* and Trends* tables usually take up more than 90% of the database size. So compression on these tables is really beneficial.
- **Improved read performance**: Any frontend operation using historical data like graphs, benefits from optimized time-series handling provided by TimeScaleDB.
- **Improved write performance**: Any write operations during metrics collection benefit from optimized time-series handling provided by TimeScaleDB.
- **Improved Housekeeping**: Like with any partitioning, housekeeping for historical data is much faster.

ZABBIX
PREMIUM PARTNER

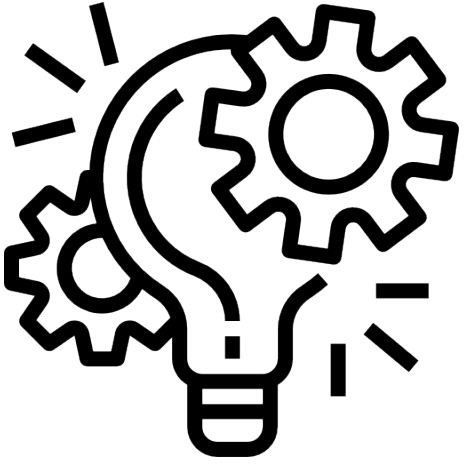
IntelliTrend
IT-Services

Migrate to PostgreSQL with TimeScaleDB

Limitations

- **No individual history/trends item periods**: Partitioning ignores individual period settings per item.
- **No manual deletion of item history/trends**: Manuel deletion will not work due to the housekeeping changes.
- **Limited data from the past**: When using compression, incoming values for items are limited by the “compress records older than“ setting in Zabbix.

Planning the migration



Migrate to PostgreSQL with TimeScaleDB

Before migration:

- Zabbix 7.0.10
- Ubuntu 22.04.5 LTS
- MariaDB 10.6.18 with partitioning, managed via a script
- 16 weeks retention for history, 57 weeks retention for trends
- 218/300 GB volume on /var/lib/mysql
- New 250 GB volume on /var/lib/postgresql

Migrate to PostgreSQL with TimeScaleDB

After migration:

- PostgreSQL 17
- TimescaleDB 2.17
- Compression support
- Extended retention of 1 year for history, 3 years of trends

Migrate to PostgreSQL with TimeScaleDB

Challenges:

- Data migration is required, lots of data needs to be moved
- SQL isn't SQL – MariaDB and PostgreSQL have their own "dialects"
  ➡ no simple SQL dump/import
- Server has HDD storage
  ➡ slow I/O, high latency
- Downtime should be as short as possible

Migrate to PostgreSQL with TimeScaleDB

Solutions:

- Use official PostgreSQL Zabbix schema as base
- Migrate data using specialized tools like “pgloader”
- Only migrate what is absolutely necessary in the first step

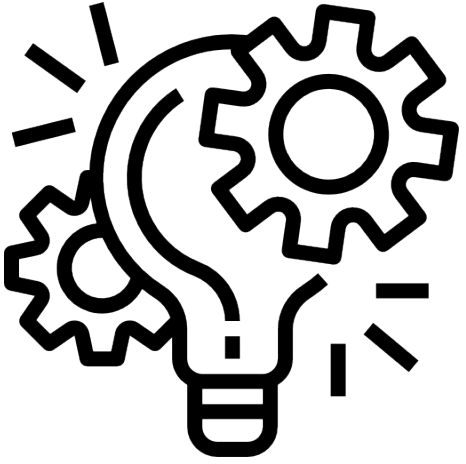
  ➡ Skip history/trends
- Migrate the history/trends later when the Zabbix server is running again

Migrate to PostgreSQL with TimeScaleDB

Required steps overview:

- Install PostgreSQL and TimescaleDB repository and packages
- Create PostgreSQL database and user
- Split up Zabbix PostgreSQL schema
- Create Zabbix tables from schema file
- Migrate database to PostgreSQL with pgloader (without history/trends)
- Apply rest of the Zabbix schema file
- Enable TimescaleDB extension and apply database patch
- Update Zabbix server/frontend configuration
- Migrate history/trends data in the background from old DB Server

Installation and setup



Install PostgreSQL and TimescaleDB

Install PostgreSQL repo:

```
# apt install -y postgresql-common
# /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
```

Install TimescaleDB repo:

```
$ curl -s https://packagecloud.io/install/repositories/timescale/timescaledb/script.deb.sh | bash
```

Install TimescaleDB 2.17 and PostgreSQL 17:

```
# apt install timescaledb-2-postgresql-17
```

Note: These TimescaleDB packages are Apache license variants and do NOT have compression support due to license restrictions!

- timescaledb-2-oss-postgresql-17 (TimescaleDB repo)
  - postgresql-17-timescaledb (PostgreSQL repo)

Enable PostgreSQL, install pgloader

Start and enable PostgreSQL:

```
# systemctl enable --now postgresql
```

Install pgloader:

```
# apt install pgloader
```

What is pgloader?

pgloader loads data from various sources into PostgreSQL. It can transform the data it reads on the fly and submit raw SQL before and after the loading.

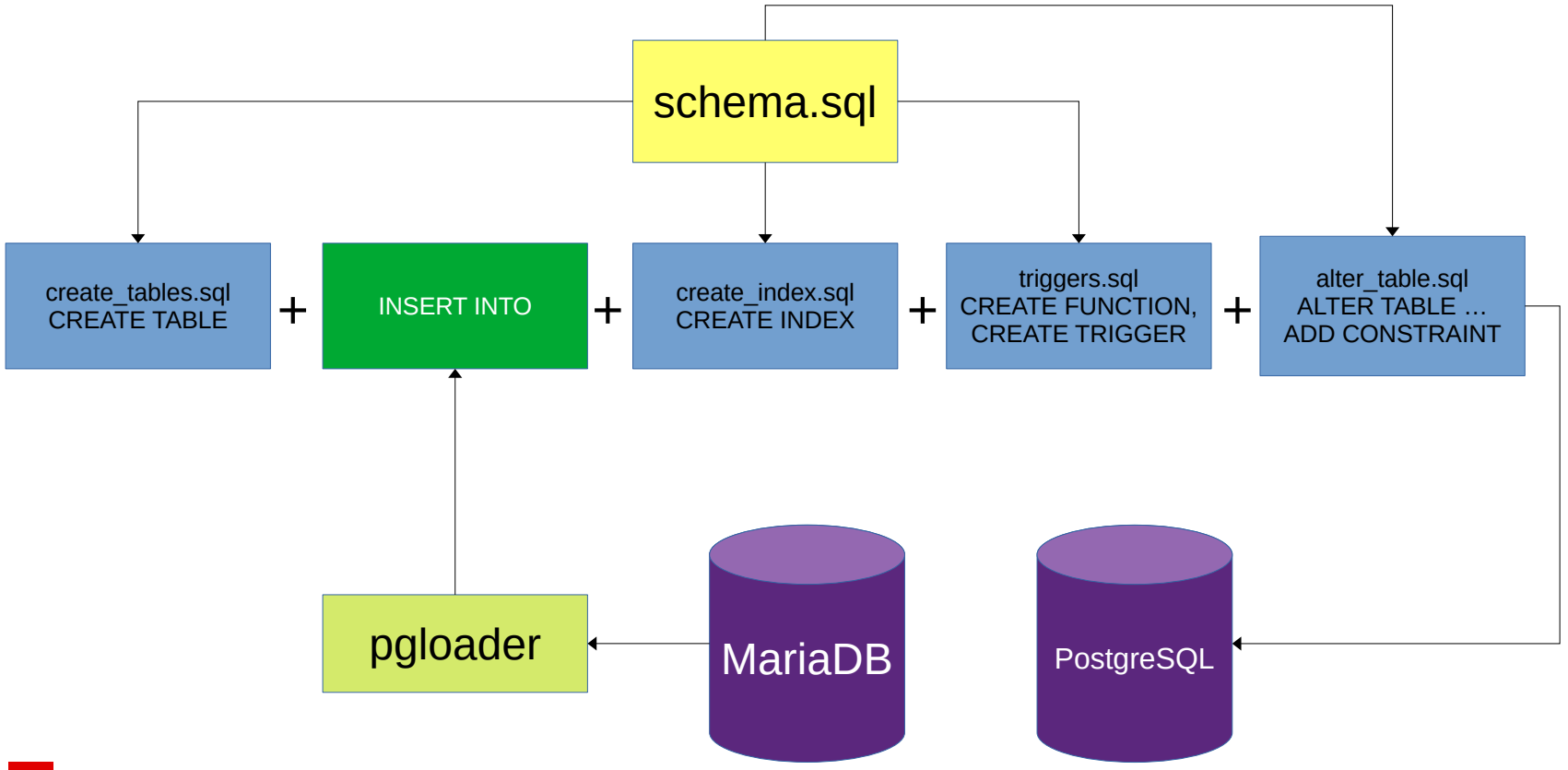
Prepare DB schema

- Zabbix has a **schema.sql** for the initial PostgreSQL database schema.
  ➡ we can use it as a starting point
- Problem: It also contains triggers, constraints and indices.
  Inserts may cause conflicts or primary key violations.

Required steps:

1. Create raw tables without indices, constrains or triggers.
2. Copy data from MySQL/MariaDB using pgloader.
3. Create index, constrains and triggers with data in place.

Prepare DB schema



Prepare DB schema

Create work directory for the schema and download Zabbix 7.10.0 source code:

```
$ mkdir /tmp/zabbix-db-migration && cd /tmp/zabbix-db-migration
$ wget https://cdn.zabbix.com/zabbix/sources/stable/7.0/zabbix-7.0.10.tar.gz
$ tar xfvz zabbix-7.0.10.tar.gz
$ cd zabbix-7.0.10/database/postgresql/
```

Delete all lines for altering tables (constraints), data inserts or creating indices:

```
$ grep -vE 'ALTER TABLE ONLY|INSERT|CREATE( UNIQUE)? INDEX' schema.sql > /tmp/zabbix-db-
migration/create_tables.sql
```

Search for first “create or replace function” call and delete everything until the end of the file – save as **create_tables.sql**:

```
$ sed -i '/create\ or\ replace\ function/,$d' /tmp/zabbix-db-migration/create_tables.sql
```

Prepare DB schema

Extract all trigger creation query – save as **triggers.sql**:

```
$ awk '/INSERT INTO dbversion/{p=1;next} /ALTER TABLE/{p=0} p' schema.sql > /tmp/zabbix-db-migration/triggers.sql
```

Extract all index create queries – save as **create_index.sql**:

```
$ grep -E 'CREATE INDEX|CREATE UNIQUE INDEX' schema.sql > /tmp/zabbix-db-migration/create_index.sql
```

Extract all constraint queries – save as **alter_table.sql**:

```
$ grep 'ALTER TABLE ONLY' schema.sql > /tmp/zabbix-db-migration/alter_table.sql
```

Prepare PostgreSQL

Create database user and database:

```
# su postgres
$ createuser --pwprompt zabbix
Enter password for new role:
Enter it again:
$ createdb -O zabbix zabbix
```

Create basic Zabbix database schema:

```
$ psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f
/tmp/zabbix-db-migration/create_tables.sql
Password for user zabbix:
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
...
```

ZABBIX
PREMIUM PARTNER

IntelliTrend
IT-Services

pgloader.conf

Prepare pgloader configuration

```
LOAD DATABASE
FROM mysql://zabbix:<password>@127.0.0.1/zabbix
INTO postgresql://zabbix:<password>@127.0.0.1/zabbix
WITH include no drop,
     truncate,
     create no tables,
     create no indexes,
     no foreign keys,
     reset sequences,
     data only,
     prefetch rows = 1000,
     batch rows = 1000,
     batch concurrency = 1
INCLUDING ONLY TABLE NAMES MATCHING ~/.*/
EXCLUDING TABLE NAMES MATCHING ~/history.*/, ~/trends.*/
ALTER SCHEMA 'zabbix' RENAME TO 'public';
```

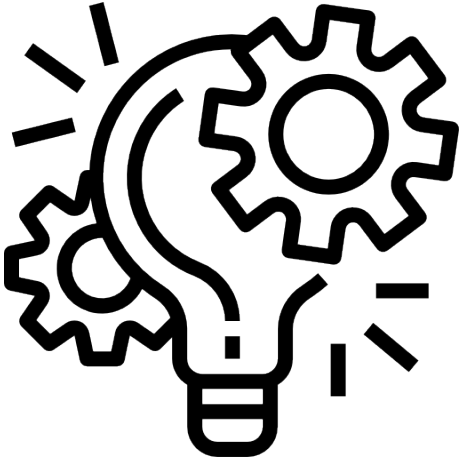
Keep existing tables

Copy data only, skip any table/index/foreign key creation

Include all tables, then filter out history/trends

Migrate to PostgreSQL with TimeScaleDB

Migrate the data



Run pgloader

Stop Zabbix server and frontend:

```
# systemctl stop zabbix-server apache2x
```

Run pgloader:

```
# pgloader pgloader.conf /tmp/zabbix-db-migration/create_tables.sql

2025-03-12T16:20:05.020000Z LOG pgloader version "3.6.10~devel"
2025-03-12T16:20:05.540005Z LOG Migrating from #<MYSQL-CONNECTION
mysql://zabbix@127.0.0.1:3306/zabbix {1006B07733}>
2025-03-12T16:20:05.540005Z LOG Migrating into #<PGSQL-CONNECTION
pgsql://zabbix@127.0.0.1:5432/zabbix {1006B078C3}>
2025-03-12T16:20:12.664063Z WARNING Source column "public"."acknowledges"."acknowledgeid" is casted
to type "numeric" which is not the same as "bigint", the type of current target database column
"public"."acknowledges"."acknowledgeid".
2025-03-12T16:20:12.664063Z WARNING Source column "public"."acknowledges"."userid" is casted to type
"numeric" which is not the same as "bigint", the type of current target database column
"public"."acknowledges"."userid".
[...]
```

There will be some warnings about different datatypes – this is expected

Run pgloader

```
                      table name     errors       rows      bytes      total time
---------------------------------  ---------  ---------  ---------  --------------
                fetch meta data          0        191                       0.664s
                       Truncate          0        191                       0.224s
---------------------------------  ---------  ---------  ---------  --------------
                public.auditlog          0     644034   264.8 MB         1m8.393s
              public.housekeeper         0     427025    13.4 MB          9.128s
                public.event_tag         0     265018     8.8 MB         1m4.276s
                   public.events         0     144299    15.0 MB          5.400s
                   public.alerts         0     189008   266.1 MB         1m3.672s
                public.item_tag          0     116038     3.6 MB          2.004s
             public.item_preproc         0      63551     3.6 MB          1.644s
              public.item_rtdata         0      48315   827.0 kB          1.036s
              public.item_rtname         0      47376     4.3 MB          1.872s
                public.functions         0      29172   903.9 kB          0.952s
                   public.graphs         0      11798     1.2 MB          1.084s
              public.trigger_tag         0      10571   296.2 kB          0.536s
                public.hostmacro         0       7174   660.5 kB          0.648s
           public.graph_discovery        0       5312   138.7 kB          0.380s
           public.trigger_depends        0       4023    69.6 kB          0.384s
            public.item_condition        0       3649   193.9 kB          0.492s
              public.acknowledges        0       2757   180.1 kB          0.656s
                  public.triggers        0      16825     6.0 MB          1.980s
                                      [...]
```

Run pgloader

```
                                          [...]
    public.lld_override_opperiod          0          0                      4.504s
                      public.mfa          0          0                      4.580s
           public.opcommand_grp           0          0                      4.612s
             public.opinventory           0          0                      4.696s
        public.proxy_autoreg_host         0          0                      4.780s
        public.proxy_group_rtdata         0          0                      4.804s
              public.report_param         0          0                      4.900s
             public.report_usrgrp         0          0                      4.988s
       public.service_status_rule         0          0                      4.992s
               public.sysmap_url          0          0                      5.092s
               public.tag_filter          0          1      0.0 kB          5.196s
                public.task_data          0          1      0.4 kB          5.176s
public.task_remote_command_result         0          0                      5.280s
            public.trigger_queue          0          0                      5.372s
    public.userdirectory_idpgroup         0          0                      5.396s
       public.userdirectory_media         0          0                      5.468s
      public.userdirectory_usrgrp         0          0                      5.584s
              public.widget_field         0      14867    897.0 kB          5.864s
---------------------------------  ---------  ---------  ---------  --------------
          COPY Threads Completion         0          4                  2m34.877s
                  Reset Sequences         0          1                     0.308s
                 Install Comments         0          0                     0.000s
---------------------------------  ---------  ---------  ---------  --------------
                Total import time         ✓    2354116    635.5 MB       2m35.185s
```

ZABBIX PREMIUM PARTNER

IntelliTrend IT-Services

Apply remaining schema

Apply indices:

```
$ psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f
/tmp/zabbix-db-migration/create_index.sql
```

Apply constrains:

```
$ psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f
/tmp/zabbix-db-migration/alter_table.sql
```

Apply triggers:

```
$ psql --host=127.0.0.1 --dbname=zabbix --username=zabbix -f
/tmp/zabbix-db-migration/triggers.sql
```

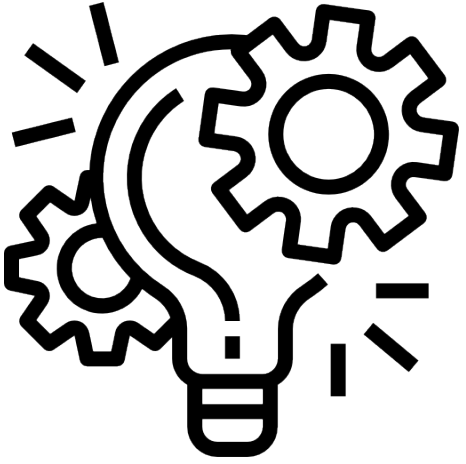
Optimize tables:

```
$ vacuumdb --dbname=zabbix --analyze --username=postgres --jobs=$(grep -c processor
/proc/cpuinfo)
```

ZABBIX
PREMIUM PARTNER

We now have a working PostgreSQL database.

Activate TimeScaleDB



Intermediate result

A few more steps are required:

- Activate TimescaleDB extension and schema
- Reconfigure server and frontend
- Migrate history and trends data

timescaledb-tune

Tune PostgreSQL configuration for TimescaleDB – also activates library:

```
# timescaledb-tune
Using postgresql.conf at this path:
/etc/postgresql/17/main/postgresql.conf

Is this correct? [(y)es/(n)o]: y
Writing backup to:
/tmp/timescaledb_tune.backup202503051145

shared_preload_libraries needs to be updated
Current:
#shared_preload_libraries = ''
Recommended:
shared_preload_libraries = 'timescaledb'
Is this okay? [(y)es/(n)o]: y
success: shared_preload_libraries will be updated

Tune memory/parallelism/WAL and other settings? [(y)es/(n)o]: y
Recommendations based on 15.82 GB of available memory and 4 CPUs for PostgreSQL 17
[...]
```

Enable TimescaleDB

Activate TimescaleDB extension:

```
$ echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | psql zabbix
CREATE EXTENSION
```

Enable TimescaleDB Zabbix schema and migrate tables:

```
$ cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/schema.sql | psql zabbix
```

Migration will take a few minutes only.

Enable TimescaleDB

```
CREATE FUNCTION
NOTICE:  function base36_decode(pg_catalog.varchar) does not exist, skipping
DROP FUNCTION
NOTICE:  PostgreSQL version 17.4 (Debian 17.4-1.pgdg120+2) is valid
NOTICE:  TimescaleDB extension is detected
NOTICE:  TimescaleDB version 2.18.2 is valid
WARNING:  column type "character varying" used for "source" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "value" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "auditid" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "username" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "ip" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "resource_cuid" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "resourcename" does not follow best practices
HINT:  Use datatype TEXT instead.
WARNING:  column type "character varying" used for "recordsetid" does not follow best practices
HINT:  Use datatype TEXT instead.
NOTICE:  migrating data to chunks
DETAIL:  Migration might take a while depending on the amount of data.
NOTICE:  TimescaleDB is configured successfully
DO
```

There will be some warnings about datatypes – this is expected

Reconfigure Zabbix for TimescaleDB

Install Zabbix Server PosgreSQL variant (uninstalls zabbix-server-mysql) and Postgres PHP module:

```
# apt install zabbix-server-pgsql php8.1-pgsql
```

Update frontend config - edit /etc/zabbix/web/zabbix.conf.php:

- Change **$DB['TYPE']** to **POSTGRESQL**
- Change **$DB['SCHEMA']** to **public**
- Change **$DB['SERVER']** to **localhost**

Then update database settings in /etc/zabbix/zabbix.conf(.d/) if required (socket, host name).

Start new PostgresSQL enabled Zabbix server and frontend:

```
# apt start zabbix-server-pgsql apache2
```

Intermediate result

We now have a working Zabbix server with TimescaleDB!
But we're not quite finished yet… graphs are still empty.

(...remember we skipped history and trends data)

History and Trends - Migration strategies

Migrating history and trends using a “small application” in go:

- Select items one by one
- Select all values for that item, ordered by timestamp
- Insert the values in batches
- Save last timestamp after each batch
- Add delay between batches to lower DB load
- Skip duplicates

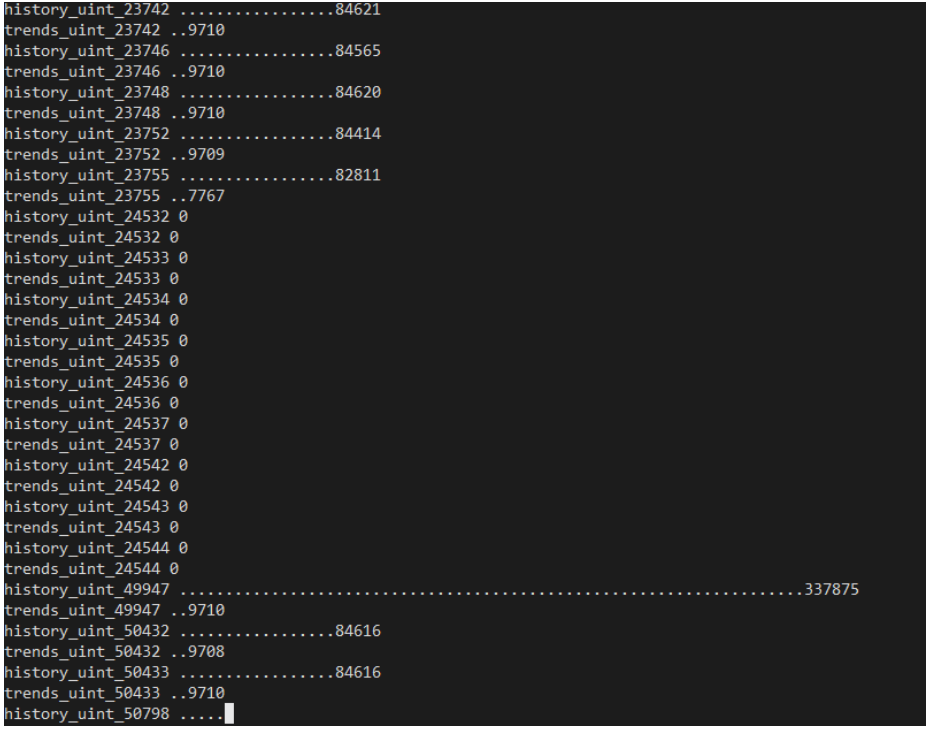
History and Trends - Migration strategies

Observations during the migration:

- Low load on TimescaleDB, moderate load on MariaDB
- Restarts after interruption are very quick:
  ➡ skip items with existing timestamp file
  ➡ continue with timestamp from last timestamp file

History and Trends - Migration strategies

Screenshot of migration in progress:



ZABBIX PREMIUM PARTNER

IntelliTrend IT-Services

Conclusion

Lessons learned:

- Disable compression during migration of history/trends (and enable after migration)
- Compression reduced the total data size from ~218 GB to ~75 GB.
- “Compress records older than” is also the limit for historical data received by Zabbix trappers (**older data is silently discarded**)
- `timescaledb-tune` picks settings that are optimal for ssds but not for hard drive storage (according to https://pgtune.leopard.in.ua/):
  - random_page_cost: 4 for hdd instead of 1.1 for ssd
  - effective_io_concurrency: 2 for hdd instead of 256 for ssd

ZABBIX '25
CONFERENCE
GERMANY

Thank you

Wolfgang Alper

CEO
IntelliTrend GmbH