Professional PostgreSQL scheduling made easy

Senior Database Consultant

pavlo.golub@cybertec.at

@PavloGolub

CYBERTEC
The PostgreSQL Database Company

About CYBERTEC

- Inhouse Development
- International team of developers
- Specialized in Data services
- Owner-managed since 2000

CYBERTEC Worldwide

Wiener Neustadt
AUSTRIA

Tallinn
ESTONIA

Zurich
SWITZERLAND

Montevideo
URUGUAY

CYBERTEC
The PostgreSQL Database Company

Client sectors

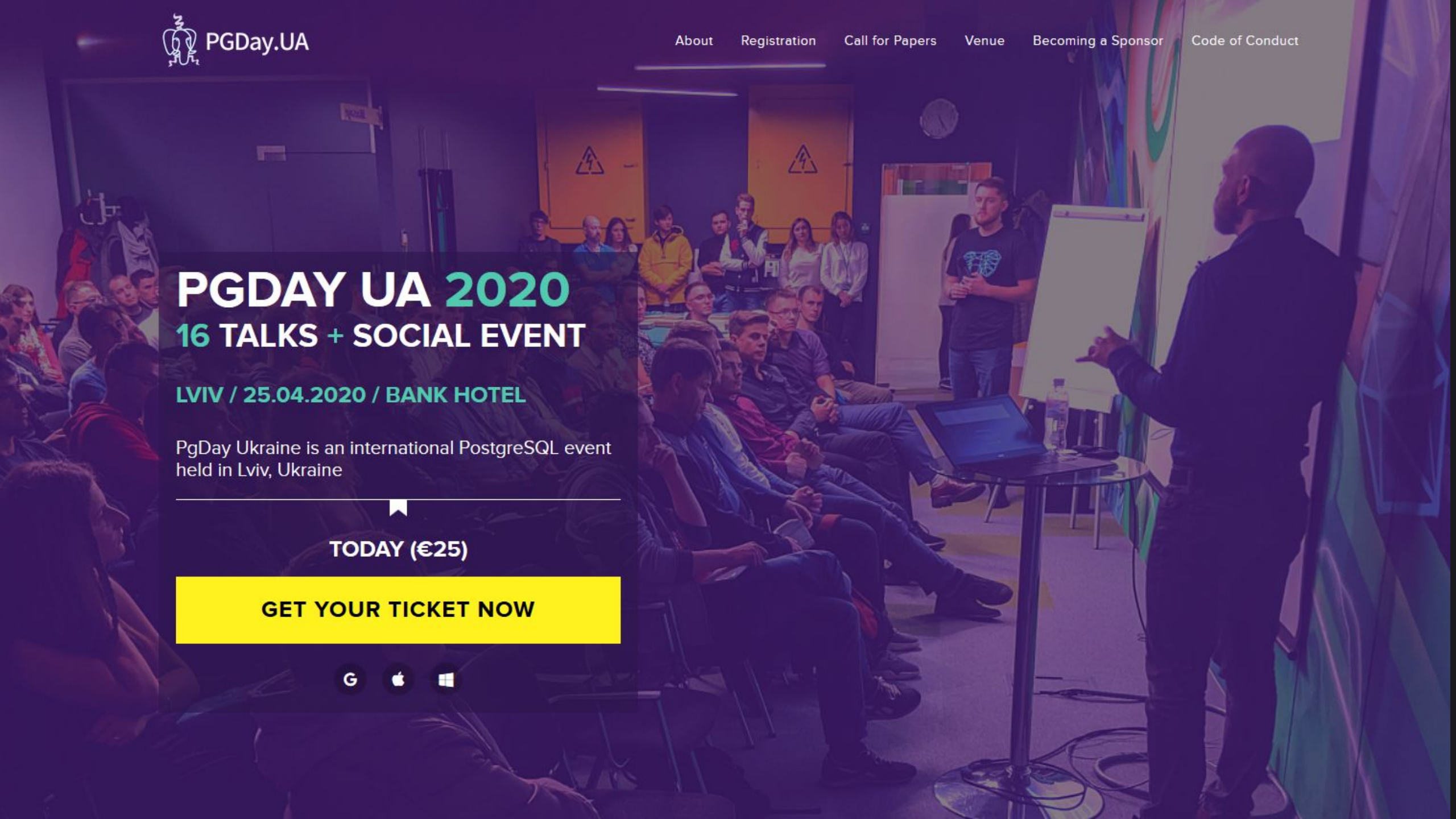
- University
- Automotive
- Government
- Industry
- Administration
- Finance
- Trade
- etc.

PostgreSQL Database Services

DATA Services

- Artificial Intelligence
- Machine learning
- BIG DATA
- Business Intelligence
- Data Mining


REPLICATION
MONITORING
TRAINING
node 1
node 2
node 3
type 6040
BenQ


PGDay.UA
About
Registration
Call for Papers
Venue
Becoming a Sponsor
Code of Conduct
PGDAY UA 2020
16 TALKS + SOCIAL EVENT
LVIV / 25.04.2020 / BANK HOTEL
PgDay Ukraine is an international PostgreSQL event held in Lviv, Ukraine
TODAY (€25)
GET YOUR TICKET NOW

Agenda

- Different levels of database scheduing

Agenda

- Different levels of database scheduing
- PostgreSQL scheduling approaches

Agenda

- Different levels of database scheduing
- PostgreSQL scheduling approaches
- PostgreSQL scheduling tools available

Agenda

- Different levels of database scheduing
- PostgreSQL scheduling approaches
- PostgreSQL scheduling tools available
- pg_timetable

Agenda

- Different levels of database scheduing
- PostgreSQL scheduling approaches
- PostgreSQL scheduling tools available
- pg_timetable
- TODO *

Why to schedule

- Maintenence
- Data Import / Export
- Backup / Restore
- Analytical Processing
- Monitoring
- External Actions

Different levels of scheduling

- Built-in Schedulers
- System Schedulers
- PostgreSQL land

Built-in Schedulers

- Microsoft SQL
- Oracle
- MySQL (MariaDB)
- DB2

System scheduling

Internal system tools, but they do know nothing about database.

Should be implemented as scripts or programs at the end.

- cron, anacron, etc.
- Windows Task Scheduler
- Google Cloud Tasks, Amazon Scheduled Tasks
- Kubernetes CronJob

PostgreSQL land

Many people say it's not necessary, and probably some hackers would oppose it; but mainly I think we just haven't agreed (or even discussed) what the design of such a scheduler would look like. For example, do we want it to be able to just connect and run queries and stuff, or do we want something more elaborate able to start programs such as running pg_dump? What if the program crashes -- should it cause the server to restart? And so on. It's not a trivial problem.

Alvaro Herrera

PostgreSQL land

- pgAgent
- jpgAgent
- pg_cron
- pgBucket (runseven)
- pgAutomator (discontinued?)

pgagent

- The oldest one!
- Was a part of pgAdmin, now is distributed independently
- Written in C++
- Stores configuration in the database
- SQL and SHELL tasks
- https://github.com/postgres/pgagent

jpgAgent

- pgAgent compatible
- written in Java
- minimizes the pain of switching for existing pgAgent users
- provides more stable and feature rich agent implementation
- SQL and SHELL tasks, with partial email task support
- parallel task execution
- can kill running jobs
- supports job and task timeout
- https://github.com/GoSimpleLLC/jpgAgent

pg_cron

- old enough
- implemented as PostgreSQL background worker
- written in C
- uses libpq to open a new connection to the databases
- SQL only tasks
- jobs are executed locally with permissions of the current user
- superusers may update sys table to allow remote execution
  - need to use .pgpass to authenticate with the remote server
- https://github.com/citusdata/pg_cron/

pgBucket (runseven)

- Under active development
- Written in C++
- Uses dedicated configuration file
- SQL and SHELL tasks
- Special cascaded/event tasks
- Auto job disable
- https://bitbucket.org/dineshopenscg/pgbucket/

HOW SCHEDULERS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION: THERE ARE 14 COMPETING SCHEDULERS

14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL SCHEDULER THAT COVERS EVERYONE'S USE CASES.

SOON:

SITUATION: THERE ARE 15 COMPETING SCHEDULERS

pg_timetable

- Main principles
- Architecture
- Features
- Demo

Main principles - why another tool?

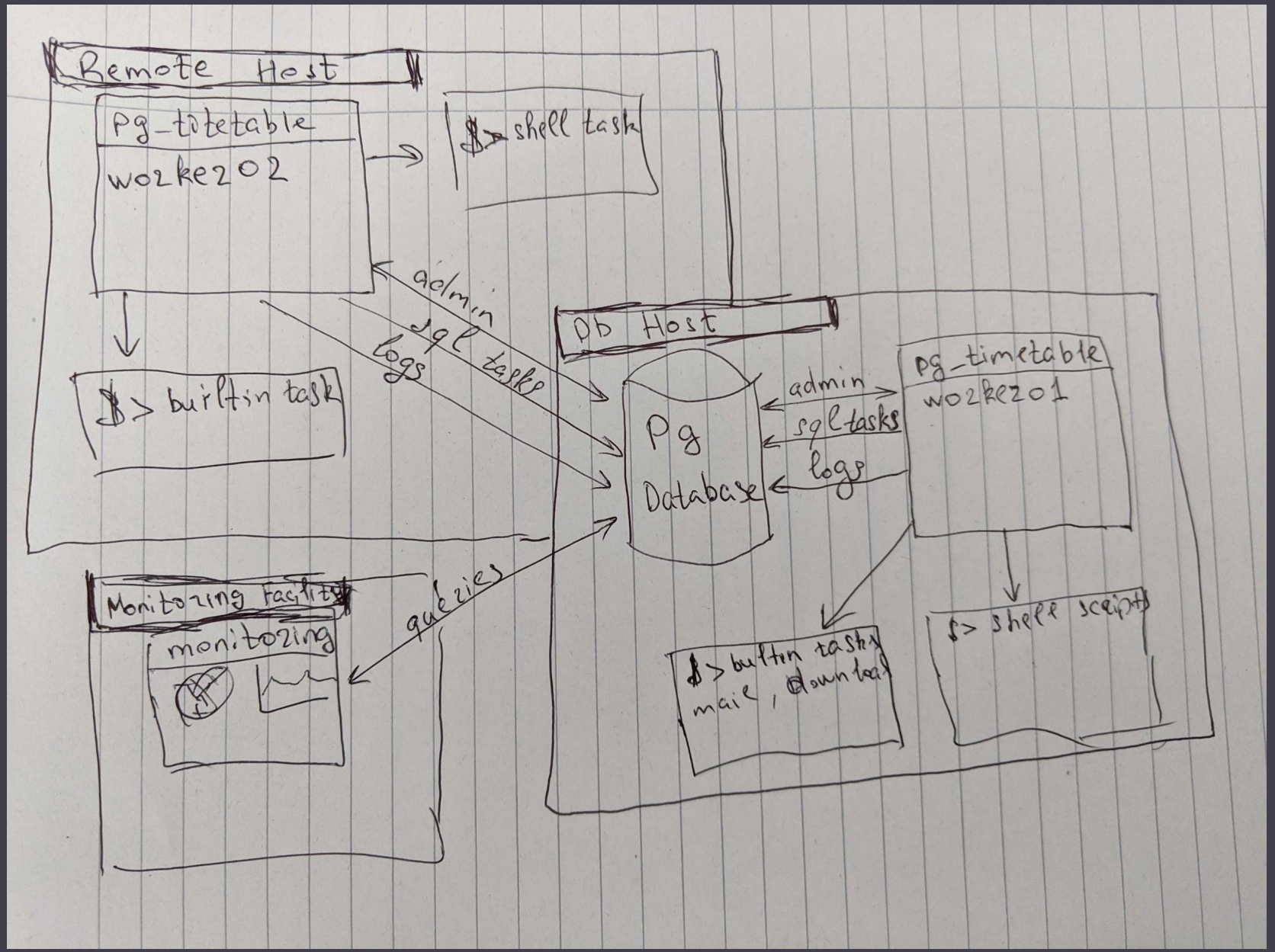
- 1-minute setup
  - docker image
  - one binary written in Go
- Non-invasive
  - No extensions or superuser needed for base functionality
  - Schema auto deployment
- Huge amount of jobs
- Cross platform support
- SQL, SHELL and BUILT-IN tasks available

Main principles - why another tool?

- Cron-style scheduling
- Enhanced logs
  - workflow log and task execution log
- Concurrency implemented using light weight goroutines
- Fully database driven configuration
- Concurrency protection
- Optional error ignoring
- Optional exclusive execution
- Self-destructive chains

Architecture components

- Workers (Golang)
- Config database
- Optional Target databases
- Optional monitoring
  - pgwatch2
  - psql



TODO

- Task / Chain abortion
- Asynchronous chain execution
- OnError Chain / Task
- Support interval scheduling, e.g. `'@interval(00:00:10)'`
- Collect client messages for SQL tasks, e.g. `'RAISE NOTICE foo'`
- Tool for debugging standalone tasks
- Graphical User Interface

Getting started

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable'
sslmode='disable' user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Getting started: start

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable'
sslmode='disable' user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Getting started: session

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable' sslmode='disable'
user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Getting started: new schema

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable'
sslmode='disable' user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Getting started: concurrency

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable'
sslmode='disable' user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Getting started: workflow

```
$ pg_timetable --clientname worker1 --verbose --dbname timetable --user scheduler
[2020-01-01 09:29:18.332 | worker1 | DEBUG ]:  Starting new session... Client:worker1
Verbose:true Host:localhost:5432 DB:timetable User:postgres
[2020-01-01 09:29:18.385 | worker1 | DEBUG ]:  Connection string:
application_name=pg_timetable host='localhost' port='5432' dbname='timetable'
sslmode='disable' user='postgres' password=''
[2020-01-01 09:29:18.449 | worker1 | LOG   ]:  Connection established...
[2020-01-01 09:29:18.475 | worker1 | LOG   ]:  Executing script: sql/ddl.sql
[2020-01-01 09:29:18.619 | worker1 | LOG   ]:  Schema file executed: sql/ddl.sql
[2020-01-01 09:29:18.624 | worker1 | LOG   ]:  Executing script: sql/json-schema.sql
[2020-01-01 09:29:18.640 | worker1 | LOG   ]:  Schema file executed: sql/json-schema.sql
[2020-01-01 09:29:18.641 | worker1 | LOG   ]:  Executing script: sql/tasks.sql
[2020-01-01 09:29:18.650 | worker1 | LOG   ]:  Schema file executed: sql/tasks.sql
[2020-01-01 09:29:18.651 | worker1 | LOG   ]:  Executing script: sql/job-functions.sql
[2020-01-01 09:29:18.672 | worker1 | LOG   ]:  Schema file executed: sql/job-functions.sql
[2020-01-01 09:29:18.673 | worker1 | LOG   ]:  Configuration schema created...
[2020-01-01 09:29:18.674 | worker1 | DEBUG ]:  Trying to get advisory lock for 'worker1' with
hash 0xc0c02cc
[2020-01-01 09:29:18.677 | worker1 | LOG   ]:  Checking for task chains...
[2020-01-01 09:29:18.679 | worker1 | LOG   ]:  Number of chains to be executed: 0
[2020-01-01 09:29:21.404 | worker1 | LOG   ]:  Ctrl+C pressed at terminal
[2020-01-01 09:29:21.410 | worker1 | LOG   ]:  Closing session
```

Schema: tables

```
$ psql -d timetable
psql (12.1)

timetable=# \dt timetable.*
                  List of relations
  Schema   |           Name            | Type  |   Owner
-----------+---------------------------+-------+-----------
 timetable | base_task                 | table | scheduler
 timetable | chain_execution_config    | table | scheduler
 timetable | chain_execution_parameters | table | scheduler
 timetable | database_connection       | table | scheduler
 timetable | execution_log             | table | scheduler
 timetable | log                       | table | scheduler
 timetable | run_status                | table | scheduler
 timetable | task_chain                | table | scheduler
(8 rows)
```

Schema: base tasks

```
timetable=# SELECT * FROM timetable.base_task;
 task_id |   name   |  kind   |  script
---------+----------+---------+----------
       1 | NoOp     | BUILTIN | NoOp
       2 | Sleep    | BUILTIN | Sleep
       3 | Log      | BUILTIN | Log
       4 | SendMail | BUILTIN | SendMail
       5 | Download | BUILTIN | Download
(5 rows)
```

Schema: log

```
timetable=# SELECT * FROM timetable.log;
 id |            ts               | client_name |  pid  | log_level |
message

----+-----------------------------+-------------+-------+-----------+-------------------------
-----------------------------------
  1 | 2020-01-16 12:48:07.107856+01 | worker1   | 15672 | LOG       | Schema file executed: sql/ddl.sql
  2 | 2020-01-16 12:48:07.128729+01 | worker1   | 15672 | LOG       | Schema file executed:
sql/json-schema.sql
  3 | 2020-01-16 12:48:07.139926+01 | worker1   | 15672 | LOG       | Schema file executed:
sql/tasks.sql
  4 | 2020-01-16 12:48:07.152016+01 | worker1   | 15672 | LOG       | Schema file executed:
sql/job-functions.sql
  5 | 2020-01-16 12:48:07.153597+01 | worker1   | 15672 | LOG       | Configuration schema created...
  6 | 2020-01-16 12:48:07.155027+01 | worker1   | 15672 | DEBUG     | Trying to get advisory lock for
'worker1' with hash 0xc0c02cc
  7 | 2020-01-16 12:48:07.159041+01 | worker1   | 15672 | LOG       | Checking for task chains...
  8 | 2020-01-16 12:48:07.163653+01 | worker1   | 15672 | LOG       | Number of chains to be executed: 0
  9 | 2020-01-16 12:48:10.697484+01 | worker1   | 15672 | LOG       | Ctrl+C pressed at terminal
 10 | 2020-01-16 12:48:10.705032+01 | worker1   | 15672 | LOG       | Closing session
(10 rows)
```

Adding a chain

- Download file from the internet
  - Ignore errors
- Remove accents
- Clean table
- Import new data from processed file

Adding a chain: variables

```
DO $$
DECLARE
    v_head_id bigint;
    v_chain_id bigint;
    v_chain_config_id bigint;
    v_task_id bigint;
BEGIN
   ...
```

Adding a chain: Step 1

```
-- Step 1. Download file from the server
-- Create the chain
 INSERT INTO timetable.task_chain (task_id, ignore_error)
    VALUES (timetable.get_task_id ('Download'), TRUE)
 RETURNING
    chain_id INTO v_head_id;

-- Create the chain configuration with default values executed every minute
 INSERT INTO timetable.chain_execution_config
  (chain_id, chain_name, live)
 VALUES
  (v_head_id, 'Download locations and aggregate', TRUE)
 RETURNING
    chain_execution_config INTO v_chain_config_id;
```

Adding a chain: Step 1

```
    -- Create the parameters for the step 1
     INSERT INTO timetable.chain_execution_parameters
        (chain_execution_config, chain_id, order_id, value)
     VALUES
        (v_chain_config_id, v_head_id, 1, '
            {
                "workersnum": 1,
                "fileurls":
["https://www.cybertec-postgresql.com/secret/orte.txt"],
                "destpath": "."
            }'::jsonb);

    RAISE NOTICE 'Step 1 completed. DownloadFile task added';
```

Adding a chain: Step 2

```
-- Step 2. Transform Unicode characters into ASCII
 -- Create the shell task to call 'uconv -x' and name it 'unaccent'
 INSERT INTO timetable.base_task(name, kind, script)
 VALUES ('unaccent', 'SHELL'::timetable.task_kind, 'uconv')
 RETURNING task_id INTO v_task_id;
```

Adding a chain: Step 2

```
-- Add shell task 'unaccent' to the chain
 INSERT INTO timetable.task_chain (parent_id, task_id, ignore_error)
 VALUES (v_head_id, v_task_id, TRUE)
 RETURNING
    chain_id INTO v_chain_id;
```

Adding a chain: Step 2

```
-- Create the parameters for the 'unaccent' base task.
-- Input and output files in this case
 INSERT INTO timetable.chain_execution_parameters
    (chain_execution_config, chain_id, order_id, value)
 VALUES
    (v_chain_config_id, v_chain_id, 2,
     '["-x", "Latin-ASCII", "-o", "orte_ansi.txt", "orte.txt"]'::jsonb);

RAISE NOTICE 'Step 2 completed. Unacent task added';
```

Adding a chain: Step 2

```
-- Create the parameters for the 'unaccent' base task.
-- Input and output files in this case
 INSERT INTO timetable.chain_execution_parameters
    (chain_execution_config, chain_id, order_id, value)
 VALUES
    (v_chain_config_id, v_chain_id, 2,
     '["-x", "Latin-ASCII", "-o", "orte_ansi.txt", "orte.txt"]'::jsonb);
```

Adding a chain: Step 3

```
-- Step 3. Import ASCII file to PostgreSQL table using "psql \copy"
 -- Create the shell task to cal 'psql' and name it 'psql'
 INSERT INTO timetable.base_task(name, kind, script)
 VALUES ('psql', 'SHELL'::timetable.task_kind, 'psql')
 RETURNING
 task_id INTO v_task_id;

 -- Add shell task 'psql' to the chain
 INSERT INTO timetable.task_chain (parent_id, task_id)
 VALUES (v_chain_id, v_task_id)
 RETURNING
    chain_id INTO v_chain_id;

-- Prepare the destination table 'location'
 CREATE TABLE IF NOT EXISTS location(name text);
```

Adding a chain: Step 3

```
    -- Add the parameters for the 'psql' base task.
    -- Execute client side \copy to 'location' from 'orte_ansi.txt'
     INSERT INTO timetable.chain_execution_parameters
        (chain_execution_config, chain_id, order_id, value)
     VALUES (v_chain_config_id, v_chain_id, 3, ('[
         "-h", "' || host(inet_server_addr()) || '",
         "-p", "' || inet_server_port() || '",
         "-d", "' || current_database() || '",
         "-U", "' || current_user || '",
         "-c", "TRUNCATE location",
         "-c", "\\copy location FROM orte_ansi.txt"
     ]')::jsonb);

     RAISE NOTICE 'Step 3 completed. Import task added';
END;
$$
LANGUAGE 'plpgsql';
```

Adding a chain

```
timetable=# \i samples/Download.sql
NOTICE:  Step 1 completed. DownloadFile task added
NOTICE:  Step 2 completed. Unacent task added
NOTICE:  relation "location" already exists, skipping
NOTICE:  Step 3 completed. Import task added
DO
timetable=#
```

Testing a chain

- Session start
- Check for tasks
- Check if chain can be executed
- Execute chain task by task
  - Ignore errors if needed
- Check if chain is finished
- Commit chain transaction

```
[2020-01-01 19:38:41.465 | worker01 | DEBUG ]:     Starting new session... Client:worker01 Verbose:true Host:localhost:5432
DB:timetable User:scheduler
[2020-01-01 19:38:41.465 | worker01 | DEBUG ]:     Connection string: application_name=pg_timetable host='localhost' port='5432'
dbname='timetable' sslmode='disable' user='scheduler' password=''
[2020-01-01 19:38:41.505 | worker01 | LOG   ]:     Connection established...
[2020-01-01 19:38:41.510 | worker01 | DEBUG ]:     Trying to get advisory lock for 'worker01' with hash 0xf0702fc
[2020-01-01 19:38:41.513 | worker01 | LOG   ]:     Checking for task chains...
[2020-01-01 19:38:41.521 | worker01 | LOG   ]:     Number of chains to be executed: 1
[2020-01-01 19:38:41.522 | worker01 | DEBUG ]:     Putting head chain
{"ChainExecutionConfigID":1,"ChainID":1,"ChainName":"Download locations and
aggregate","SelfDestruct":false,"ExclusiveExecution":false,"MaxInstances":16} to the execution channel
[2020-01-01 19:38:41.522 | worker01 | DEBUG ]:     Calling process chain for
{"ChainExecutionConfigID":1,"ChainID":1,"ChainName":"Download locations and
aggregate","SelfDestruct":false,"ExclusiveExecution":false,"MaxInstances":16}
[2020-01-01 19:38:41.523 | worker01 | DEBUG ]:     Checking if can proceed with chaing config ID: 1
[2020-01-01 19:38:41.525 | worker01 | LOG   ]:     Starting chain ID: 1; configuration ID: 1
[2020-01-01 19:38:41.571 | worker01 | DEBUG ]:     Executing task:
{"ChainConfig":1,"ChainID":1,"TaskID":5,"TaskName":"Download","Script":"Download","Kind":"BUILTIN","RunUID":{"String":"","Valid":f
alse},"IgnoreError":true,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}
[2020-01-01 19:38:41.573 | worker01 | DEBUG ]:     Executing builtin task Download with parameters [{"destpath": ".", "fileurls":
["https://www.cybertec-postgresql.com/secret/orte.txt"], "workersnum": 1}]
[2020-01-01 19:38:41.889 | worker01 | ERROR ]:     Task execution failed:
{"ChainConfig":1,"ChainID":1,"TaskID":5,"TaskName":"Download","Script":"Download","Kind":"BUILTIN","RunUID":{"String":"","Valid":f
alse},"IgnoreError":true,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}; Error:
download failed: [bad content length]
[2020-01-01 19:38:41.897 | worker01 | DEBUG ]:     Executing task:
{"ChainConfig":1,"ChainID":2,"TaskID":6,"TaskName":"unaccent","Script":"uconv","Kind":"SHELL","RunUID":{"String":"","Valid":false}
,"IgnoreError":true,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}
[2020-01-01 19:38:41.955 | worker01 | DEBUG ]:     Output for command uconv [-x Latin-ASCII -o orte_ansi.txt orte.txt]:

[2020-01-01 19:38:41.956 | worker01 | DEBUG ]:     Task executed successfully:
{"ChainConfig":1,"ChainID":2,"TaskID":6,"TaskName":"unaccent","Script":"uconv","Kind":"SHELL","RunUID":{"String":"","Valid":false}
,"IgnoreError":true,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}
```

```
[2020-01-01 19:38:41.958 | worker01 | DEBUG ]:     Executing task:
{"ChainConfig":1,"ChainID":3,"TaskID":7,"TaskName":"psql","Script":"psql","Kind":"SHELL","RunUID":{"String":"","Valid":false},"Ign
oreError":false,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}
[2020-01-01 19:38:42.162 | worker01 | DEBUG ]:     Output for command psql [-h 127.0.0.1 -p 5432 -d timetable -U pasha -c TRUNCATE
location -c \copy location FROM orte_ansi.txt]:
TRUNCATE TABLE
COPY 2354

[2020-01-01 19:38:42.163 | worker01 | DEBUG ]:     Task executed successfully:
{"ChainConfig":1,"ChainID":3,"TaskID":7,"TaskName":"psql","Script":"psql","Kind":"SHELL","RunUID":{"String":"","Valid":false},"Ign
oreError":false,"DatabaseConnection":{"String":"","Valid":false},"ConnectString":{"String":"","Valid":false}}
[2020-01-01 19:38:42.165 | worker01 | LOG   ]:     Chain ID: 1 executed successfully
[2020-01-01 19:38:42.166 | worker01 | DEBUG ]:     Commit transaction for successful chain execution
[2020-01-01 19:38:53.043 | worker01 | LOG   ]:     Ctrl+C pressed at terminal
[2020-01-01 19:38:53.048 | worker01 | LOG   ]:     Closing session
```

Improvement ideas?

User input very much appreciated!

github.com/cybertec-postgresql/pg_timetable

Thanks

Don't be a stranger:

https://www.cybertec-postgresql.com/en/blog/

CYBERTEC
The PostgreSQL Database Company

Professional PostgreSQL scheduling made easy

Senior Database Consultant

pavlo.golub@cybertec.at

@PavloGolub

CYBERTEC
The PostgreSQL Database Company