

# Serverová část the damn project

2021-04-15T22:16:52+02:00

# Serverová část the damn project

# Serverová část the damn project

Jiří Vlasák

Co vlastně řešíme?

# Problém

- Velká oblast k mapování.
- Hodně mapperů.

# Související otázka

- Jak uspořádat práci, aby si mapperi neměnili data pod rukama?

Řešení už máme

# Řešení už máme

(Řazeno podle data prvního příspěvku do repozitáře se zdrojovým kódem.)

- 9. 7. 2011 – HOT Tasking Manager
- 2. 2. 2013 – HOT Tasking Manager 2
- 2. 2. 2017 – HOT Tasking Manager 3
- 22. 4. 2019 – Mapping North Korea
- 12. 6. 2019 – HOT Tasking Manager 4 (pouze změna frontend)
- 25. 8. 2019 – Damn server
- 26. 3. 2020 – Simple Task Manager

Ostatní, co jsem našel, jsou *deprecated*.



Divide and map. Now.

# Divide and map. Now.

*The damn project helps mappers by dividing some big area into smaller squares that a human can map.*

# Architektura

Architektura the damn project

# Oblast k mapování

# Oblast k mapování

- Informace o oblasti.
- Instrukce pro mappery.
- Rozdělení na menší části.

# Architektura databáze 1/2

## current\_areas

- AID (primary key)
- TAGS (changeset comment)
- PRIORITY
- DESCRIPTION (JSON {"lang code": "description", ...})
- INSTRUCTIONS (JSON {"what": "link to wiki", ...})
- FEATURECOLLECTION (GeoJSON)
- CREATED

## current\_squares

- SID (dohromady s AID primary key)
- AID (foreign key)
- BORDER (geometry)



Map -> Review -> Done

# Map -> Review -> Done

- Zamykání a odemykání squares.
- Zaznamenávání změn stavů squares.
- Statistiky.

# Úprava oblastí k mapování

# Úprava oblastí k mapování

- Zaznamenávání změn areas.

# Architektura databáze 2/2

## current\_commits

- CID (primary key)
- SID (spolu s AID foreign key)
- AID (foreign key)
- DATE (automaticky, UTC now())
- AUTHOR (foreign key users.display\_name)
- TYPE (to map, to review, done, locked)
- MESSAGE

SQL pro *Map* -> *Review* -> *Done*

# Budu mapovat

- Náhodný square
- Nejnovější square
- Nejbližší square



## Provedu review

- Jako u mapování
- square nezkušeného mappera

```
WITH last AS ( SELECT DISTINCT ON (sid) cid, sid, type FROM
current_commits WHERE aid=$1 ORDER BY sid, cid DESC ),
toreview AS ( SELECT * FROM last WHERE type='to review'
ORDER BY cid DESC LIMIT 1 ) INSERT INTO current_commits (sid,
aid, author, type, message) SELECT sid, $1, $2, 'locked', 'Working on
review' FROM toreview RETURNING sid
```

Nejbližší square:

```
toreview AS (
  SELECT sid, cid, type
  FROM last, mine
  WHERE type='to review'
  ORDER BY ST_Distance(mine.border, last.border)
```

SQL pro *Map* -> *Review* -> *Done*

# Po mapování rozhoduji, jestli

- square potřebuje více mapování
- square je připravený k review
- Rozdělím square

# A po review

- square je hotový
- square potřebuje více mapování
- Necítím se na to rozhodnout

```
SELECT * FROM current_commits WHERE aid=$1 AND sid=$2  
ORDER BY cid DESC LIMIT 1
```

Je poslední commit locked? Je autor posledního commitu autor právě přidávaného commitu?

```
INSERT INTO current_commits(sid, aid, author, type, message)  
VALUES ($1, $2, $3, 'done', $4)
```

# Kde použít PostGIS

## *Map -> Review -> Done*

- Budu mapovat nejbližší square
- Rozdělím square
- Ze všech mnou zamknutých squares jednoho area vytvořím nový square

# Vytváření area

- Rozdělení na squares

Rozdělím area nebo square



# Rozdělím area nebo square

- Potřebuji ST\_XMax, ST\_XMin, ST\_YMax, ST\_YMin
- Potřebuji ST\_GeomFromText( ... POLYGON ...)
- Potřebuji ST\_Translate
- Potřebuji ST\_Intersection, ST\_Intersect

Funkce `st_divide` napsaná v *pgsql*.

...

```

RETURN QUERY
WITH foo AS (
    SELECT
        ST_Translate(
            ss_def,
            ST_XMin(ar) + i * dx,
            ST_YMin(ar) + j * dy
        ) AS ss
FROM

```

Nový square ze všech mnou zamknutých

# Nový square ze všech mnou zamknutých

- Potřebuji mnou zamknuté squares
- Potřebuji ST\_Union

```
... to_be_merged AS ( SELECT border FROM current_squares cs,
locked lo WHERE cs.sid=lo.sid AND cs.aid=lo.aid ), merged AS (
SELECT ST_UNION(border) as mb FROM to_be_merged ) INSERT
INTO current_squares (sid, aid, border) SELECT f.msid +
row_number() OVER () as sid, f.maid as aid, mb FROM ( SELECT
MAX(cs.sid) as msid, MAX(cs.aid) as maid FROM current_squares cs,
locked lo WHERE cs.aid=lo.aid ) as f, merged RETURNING sid, aid
```

## Shrnutí the damn project

# Shrnutí the damn project

- Distribuovaná architektura projektu.
- Jednoduchá a funkční architektura databáze.
- Jádrem jsou PostGIS SQL dotazy, pgsql funkce. (Python server je jen API do databáze.)
- [www.damn-project.org](http://www.damn-project.org)

Demo

# Demo

- <https://manager.damn-project.org/>
- <https://client.damn-project.org/>
- <https://mappy.damn-project.org/>

# Otázky