simacan

B-Tree Index Bloat

2019-10-03 | Jeroen Soesbergen

Let’s introduce myself...

- Jeroen Soesbergen
- Scrum Master / Software Engineer
- Regular guest at the PostgreSQL User Group NL
- Learned a lot here
- Really nice to contribute back today

simacan

simacan

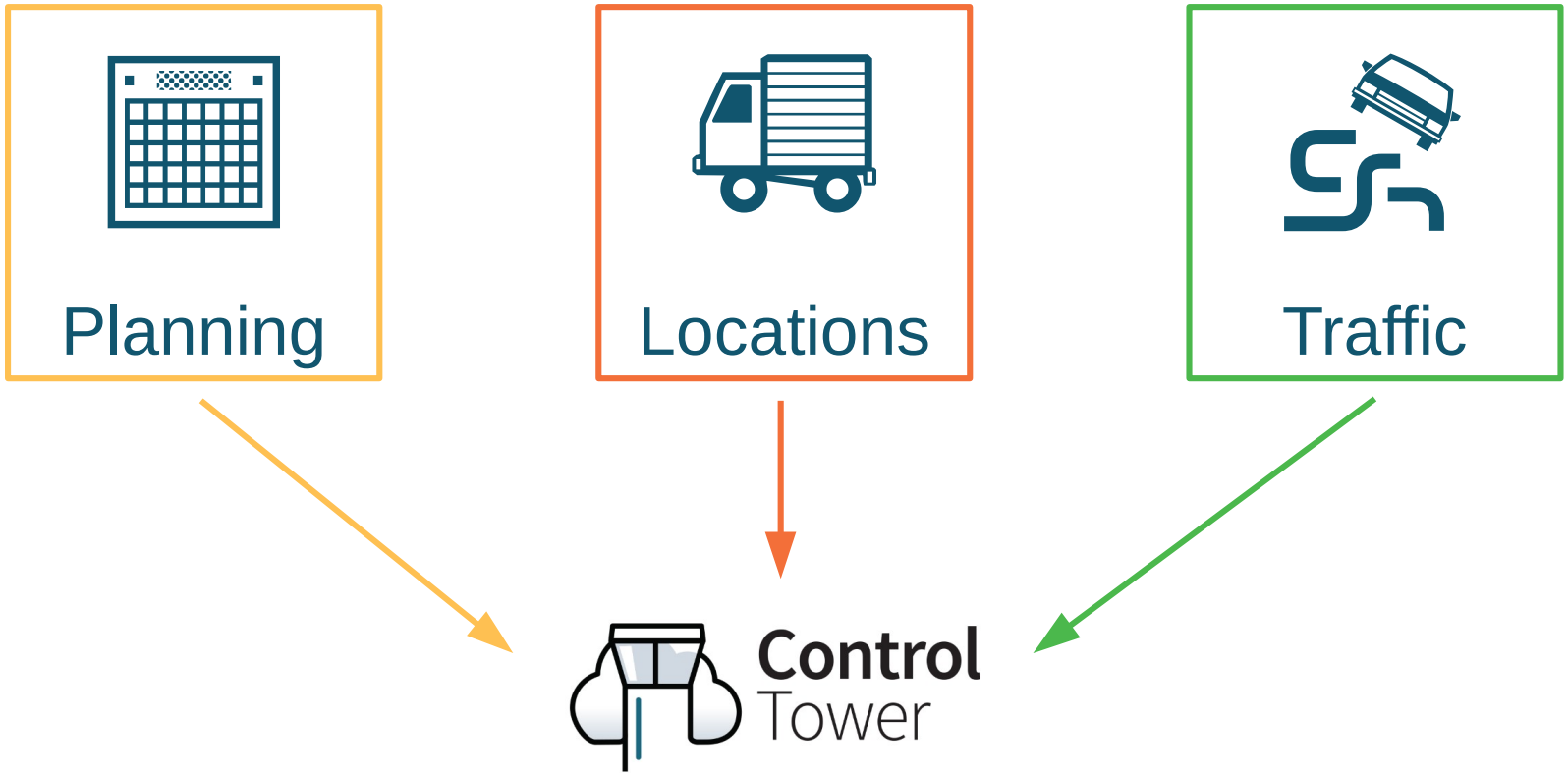
About Simacan

- Software for Traffic & Transport
- Our product:

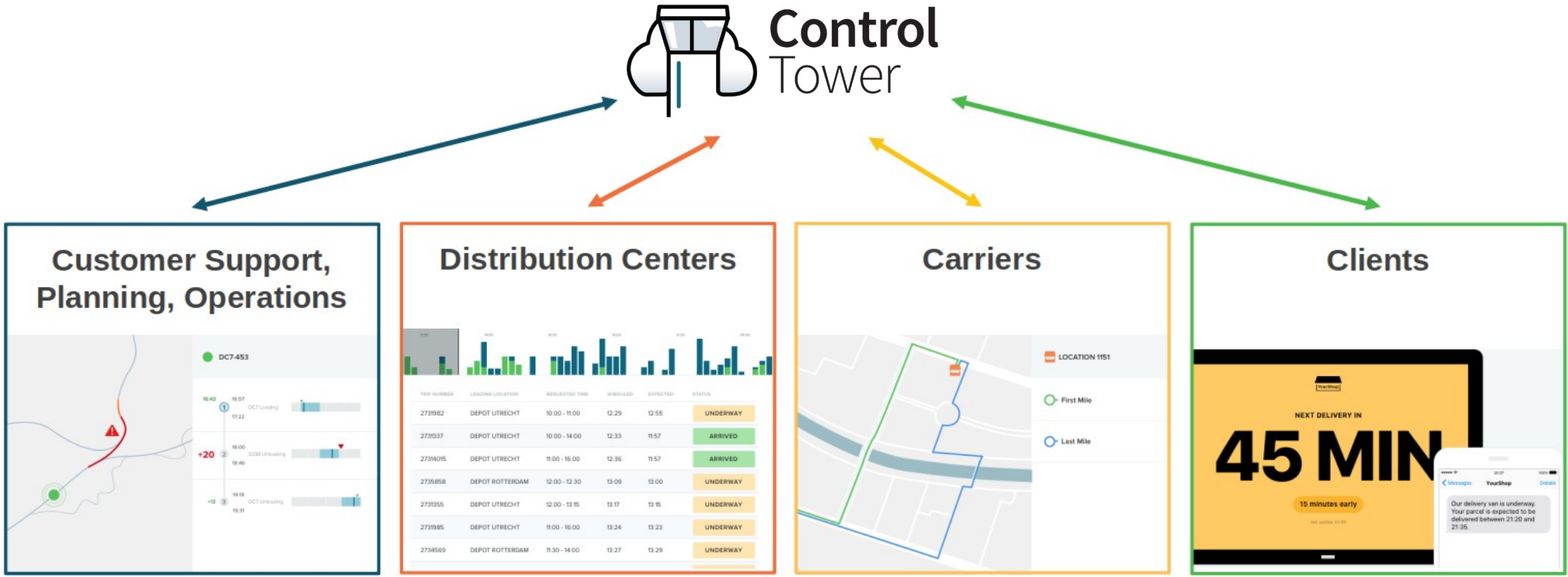
Control Tower

simacan

Simacan Control Tower (1)



Simacan Control Tower (2)



Technology @ Simacan

- Microservice Cluster in AWS
- Relational database: PostgreSQL
  - Open Source
  - Spatial extension: PostGIS
  - Proven to be a reliable choice

simacan

B-Tree Index Bloat

- What is index bloat?
- Demo
- Where does it come from?
- How to deal with index bloat?

simacan

What is B-Tree Index Bloat?

- What is this, exactly?
  - B-Tree Index = default type of index
  - Bloat = an object on disk contains empty space
- Why is this a problem?
  - Waste of disk space and I/O
  - Performance impact
- How bad can it get?

simacan

Demo: How bad can it get?

- Take a simple table/index
- Perform update/delete/insert/vacuum
  - In a specific pattern
  - Exact pattern will be revealed later!
- Monitor table and index bloat

  https://github.com/pgexperts/pgx_scripts

simacan

Where does index bloat come from?

- Index fragmentation
- Multi Version Concurrency Control (MVCC)
- Vacuum characteristics

Index fragmentation

- Indexes consist of 8kB pages
- When an index page is full:
  - It will be split
- Pages will never be merged

- Lower fill factor can prevent full pages

Multi Version Concurrency Control

- After a tuple is updated:
  - The old version still exists
  - A copy is created with the new values
- Both copies must be in the index!
- The index grows...

simacan

Vacuum characteristics

- VACUUM: Cleans up dead tuples
- Makes space in tables and index pages available for re-use
- Autovacuum daemon:
  - Triggered by vacuum threshold
    - Percentage of dead tuples in table
    - Works fine for tables, not always for indexes
    - Table tuples do not have to be ordered, but indexes do
  - Can be tweaked, when necessary

```
vacuum threshold = vacuum base threshold +
                   vacuum scale factor * number of tuples
```

Demo: Revisit

- Let’s check the demo!

simacan

Demo: Observations

- Very little table bloat
- The index goes crazy

- Interesting:
  - Index bloat can exist without table bloat
  - You need to monitor both separately

simacan

Demo: How does it work?

- Bloat 1% of the id range at a time
  - First: 10 updates on every tuple
    - Autovacuum not triggered by default
    - All copies must be indexed => Index bloat
  - Delete 80% of tuples
    - Bloats index even more
  - Vacuum
    - To ‘clear’ the updates and deletes
  - Insert new tuples, to fill table again
- This 1% of the index gets bloated badly
- Repeat on the next 1%

How to deal with index bloat?

- Prevent:
  - Tweak index fillfactor
  - Tweak autovacuum

- Fix:
  - REINDEX
  - Create replacement index

REINDEX

- Pretty simple
- Index contents are rebuilt from scratch
- Replaces the old one
- Table lock: SHARE
  - No inserts/updates/deletes
- Index lock: ACCESS EXCLUSIVE
  - Blocks all access (also reads) to the index

- This probably means downtime...

simacan

Create a replacement index

- Create an exact copy of the bloated index
- CREATE INDEX CONCURRENTLY
- Swap bloated and new index
- Extension: pg_repack
  - Does something like this

CREATE INDEX CONCURRENTLY

- Table lock: SHARE UPDATE EXCLUSIVE
  - No schema changes
  - No vacuum runs
  - No other create index concurrently
- Takes longer to complete
- May fail
  - Results in invalid index
  - Try again

Fix index bloat: Summary

- There are some ways to fix index bloat
- In common:
  - Do not repair a bloated index
  - Create new / replace old
  - Require manual action
- Differences:
  - Locking
  - Level of convenience
- Is this really what we were looking for?

REINDEX, like autovacuum?

- We like to automate everything
- We like to have a tool that:
  - Monitors index bloat
  - Automatically fixes index bloat

- Should we build one, ourselves?
- How do you deal with index bloat?

Conclusions

- Index bloat can get pretty bad
  - Demo: Almost 8000%
- Ways to prevent index bloat
  - Tweak fillfactor / autovacuum
- Ways to fix index bloat
  - REINDEX, CREATE INDEX CONCURRENTLY

- Index bloat should be monitored

simacan

Thank you!