Building Automatic Adviser and Performance Tuning Tools in PostgreSQL Q and A

PGConf.online 2021

Tatsuro Yamada
Julien Rouhaud

Q and A (1/3)

Q1: Do you have a feature to suggest Create statistics of columns?

A1: Tatsuro sent a pull-requested the feature to pg_qualstats.
We are now considering implementing the feature.

Q2: Are these production-ready?

A2: pg_plan_advsr is still in POC phase, but pg_qualstats is production-ready.

Q and A (2/3)

Q3: Could you explain how to optimize a plan?

A3: pg_plan_advsr uses a feedback loop. It fixes row estimation errors by ROWS hint on each query executions. Therefore further planning can be done using better calculated-cost and may find a better plan thanks to those. :

PGConf.Eu 2018

PGConf.Russia 2019

Q4: What can do pg_hint_plan?

A4: It can force join order, join method, scan method, and so on. You can see the manual:

https://github.com/ossc-db/pg_hint_plan/blob/master/doc/hint_list.html

Q and A (3/3)

Q5: how to force an execution plan found with pg_plan_advsr on your production without modifying client code?

A5: There are three kinds of methods:

- Use another extension
  - sr_plan by Postgres Professional is a good alternative
  - https://github.com/postgrespro/sr_plan
- Use the table hint (hint_plan.hints) of pg_hint_plan
  - get_hints function of pg_plan_advsr gives you hints to reproduce optimized plan.
  - Table hint is like a lookup table. You can add the hints with the query in the table of production. The hints will be used when the query executing.
  - Please see document more details: https://github.com/ossc-db/pg_hint_plan/blob/master/doc/pg_hint_plan.html
- Add hints as a comment, if you allow putting the comment on the client code
  - Put the comment on the top of your query to force a plan
  - This method is a preferable method than using table hint since overhead is smaller than it.