Refining Simulated Annealing approach for query optimization

Grzegorz Parka

Organization: PostgreSQL Project

Abstract: Around 2010, Jan Urbański proposed SAIO - extension using Simulated Annealing approach as a possible replacement for Genetic Query Optimizer (GEQO). It proved to generate valid query plans, often comparable or better than GEQO plans. SAIO was not yet merged into source tree because it did not meet Postgres code quality and portability standards. The aim of this project is to start from where Jan left off, verify SAIO approach and push it forward to become Postgres-quality module.

Name: Grzegorz Parka
e-mail: grzegorz.parka@gmail.com

Project: *Refining Simulated Annealing approach for query optimization*

In 2009-2010, as a part of his master thesis, Jan Urbański prototyped SAIO – join order optimizer using Simulated Annealing algorithm as a possible replacement for GEQO. It proved to generate valid query plans which were comparable to GEQO plans for small queries and better than GEQO output for larger queries. This alternative JOIN order optimizer was not yet merged as a feature because it did not meet Postgres code quality and portability standards. The project itself may be viewed on github: https://github.com/wulczer/saio

The purpose of my project would be to start from where Jan left off, ensure Simulated Annealing is a better approach than GEQO for join order optimization and then make SAIO a Postgres-quality module.

Benefits to the PostgreSQL Community:

There is a long lasting Todo item on Postgres wiki - *Consider compressed annealing to search for query plans.*

This project should be a step forward to create a better and simpler join order optimizer than the current GEQO.

As the end user I would get a possibility to choose a different join order optimizing algorithm to tune my heavy queries.

Deliverables:

- set of benchmarks to ensure that Simulated Annealing approach is really better than current GEQO; detailed report on its results
- improved SAIO module with tests and documentation that conform to Postgres standards.

After the Summer I expect the module to be ready to be merged to /contrib or released on PGXN with prospects to be moved to the main source tree in the future. However this would have to be yet discussed with the Postgres community.

During the project, the module will be developed in a separate repository available on Github.

Expected project schedule:

Before the official coding period (till 25 May):	
	• Getting familiar with GEQO, SAIO and optimizer (note: this is partially done, both in theory and practice) • Getting to know the community, establishing contact with the mentor • Preparing appropriate test cases – a lot of sample queries and datasets, to have an evidence if SAIO is indeed better than GEQO • it would be best to find some live cases (like open source projects) where queries with enough join relations are produced and extract the heavy parts • Gathering and initially evaluating possible improvements for SAIO, for example: • Research on different variants of simulated annealing that could be used • Different cooling routines • Reducing number of failed join plans generated by SAIO
During the coding period:	
Before the midterm evaluation (25 May - 29 June):	
25 May - 8 June	creating a set of benchmarks to compare SAIO against GEQO (2 weeks)
8 June - 15 June	preparing report on the benchmark results (1 week)
15 June - 22 June	testing and debugging on alternative platforms supported by Postgres (at least Linux, Windows, Mac OS and FreeBSD) (1 week)
22 June - 29 June	extra time for delays (1 week)
Before the final evaluation (29 June – 21 August):	
29 June - 27 July	cleaning up the existing SAIO module to meet postgres standards (4 weeks)
27 July - 10 August	writing documentation on the improved SAIO module (2 weeks)
10 August - 17 August	extra time for delays
optional	regular (weekly) blogging about the current work and status
21 August	Final evaluation deadline

I plan to work on the SAIO module during whole coding period of Google Summer of Code 2015.

If I manage to finish the core part of the project earlier than expected, the remaining time will be spent on research how to improve the SAIO even further.

If somehow my benchmarks do not confirm that Simulated Annealing is better than Genetic Algorithm approach, I will work on tuning the SAIO to become more competitive.

Even if for some reason the community does not accept the module, the minimal outcome would be improved knowledge about the algorithm and improved SAIO extension available on PGXN.

About myself:

I am Grzegorz Parka, Bsc Eng of Technical Physics and current student of master studies in Computer Science at Warsaw University of Technology.

As a student of Physics and Computer Science I have the required level of understanding of evolutionary algorithms, Postgres database and C programming to take up the project.

Last year I successfully completed Google Summer of Code 2014, where I developed pykCSD - scientific Python library for the International Neuroinformatics Coordinating Facility. The project enables to reconstruct current source density (a local measure of electrical activity) in neural tissue using kernel methods for 1D, 2D and 3D data sets. It is a Python implementation of Kernel Current Source Density method which was proposed in the PhD thesis of Jan Potworowski.

The project may be viewed on its Github repository (https://github.com/INCF/pykCSD) and the corresponding blog (http://parkag.github.io/pykcsd-blog/).