MUSTANG POSTGRESQL DATABASE QUERY OPTIMIZATION, PERFORMANCE ISSUES, AND UPGRADE RFQ

9/3/2019

Amendment #1: Questions and Answers & Extend the Due Date

Due Date Extension

Please note that the referenced RFQ has been extended until Friday, September 6, 2019 at 5:00 p.m.

Questions & Answers

Below, please find questions and answers received for the referenced solicitation.

Question #1: How many more-or-less unique queries need analysis?

Answer: As a rough approximation, there are generally three or four primary patterns for metrics queries although the code may have additional select constructs.

Question #2: What is the typical number of tables referenced in each query?

Answer: There are typically three tables that are referenced in each metrics query. The primary data table has a FK ID reference to a normalized source reference table. This reference table will join 1:1 on a string name resolver table. In many cases, the match to records in the main data table is carried out via a filter containing an array of IDs.

Question #3: Index, partitioning, and design issues diagnosis and conceptual solutions: how many tables with how many significant columns each? As an example, or best guess estimate with 20 tables and = 20 * 8 * 30 minutes = 40 hours.

Answer: The quality metrics schema has 64 tables, one for each metric. The large majority of these are identically constructed tables with 6 columns: value, target, start, end, id, and load date. They are currently indexed by value, target, start, end, load date. A few metrics have additional columns for multiple values. The spectra schema

has 8 relevant tables that contain spectra information aggregated by timespans year, month, week, day along with instrument response information. These have between 4 and 15 columns.

Question #4: Index, partitioning, and design issues diagnosis: How many new indexes and time for partitioning diagnosis is dependent on the number tables identified above? Partitioning diagnosis: 4 hours per table. Example: two tables = 8 hours.

Answer: IRIS currently needs partitioning on one table in the spectra schema, the spectra.psd_day table. This table is over 6TB in size and IRIS wants to divide this between two table spaces to balance filesystem I/O.

Question #5: Diagnose slow queries, with conceptual solution. Does requestor have a general idea of how many queries are longrunning, number of tables per query and approximate number of WHERE operators?

Answer: The notable slow query pattern listed in the RFQ is a query for a long list of IDs, typically presented by a large virtual network. Virtual networks are a concept that combines stations across multiple real networks and they can be large. For example, the _GSN virtual network resolves to more than 9100 ID numbers. The query pattern then consists of a where clause with four operators. Three of them are for temporal bracketing (start time, end time). The final is an IN(SELECT()) clause matching to any ID numbers in that array that can be found in the primary metrics table having over 100 million rows. These are simple queries structurally but the arbitrary ID matching against a large table seems to cause slowness.

Question #6: Is there a budget/more information that can be provided regarding this requirement?

Answer:

Major/Minor Issues

* Top issues: _Query optimization and performance issues; long-running queries_
* Additional comments/concerns: _Need version upgrade and recommendations for features in 11 that will benefit performance, scaling, and availability_

General System Details

* Use case: _Seismic quality metrics and spectra measurements that is accessed by the public 24/7_
* End users (internal/external): _Public_
* PostgreSQL Version (production; test): _9.5 (target 11.x)_
* OS type and version: _Centos 7 Linux_
* Server(s) setup: Postgres 9.5 on CentOS 7 OS w/VMWare hypervisor (8 core / 24 Gig RAM)

NFS mounted cluster tablespaces on 2xNetApp
Streaming physical replication to identical remote server in standby mode.

* Latest (major) work done by client on server(s): Upgrade from CentOS 6 to CentOS 7.
Increased level of logging.

System Details

* Approximate size of DB and tables: 11TB database; largest table size/# of rows = 172 GB with additional 6.7 TB in toast table/105557118 rows
* Any partitioning: Not currently, but pending for single large table (pg_partman installed)
* Any dependencies/integrations: streaming replication
JDBC clients w/DBCP connection pooling
pg_hba configuration for multi-subnet access
FK and UNIQUE constraints
* Any extensions: btree_gist, plpgsql

Business Requirements

* Maintenance window: weekdays 9-4
* Access restrictions: no access to root privileges
* Budgetary restrictions (specific number of $$ or hours; per month/quarter/year): $10,000 (this is only an estimate)
* Internal deadlines: _Spend and prefer completion by Fall 2019