DeepGreen User's Guide

April 21, 2017

1 Introduction

DeepGreen DB is a MPP Database derived from the open source Greenplum Database project. DeepGreen DB implmeneted a LLVM based JIT query execution engine that dynamically generate machine code for each SQL query, dramatically increases the performance of Greenplum Database. DeepGreen maintain 100% compatibility with Greenplum Database. User should observe no difference between DeepGreen DB and Greenplum Database other than the increased performance. Therefore, we refer user to the Greenplum project and PostgreSQL project for documentation.

DeepGreen DB further extends Greenplum Database by adding new data types, functions, external database accesses and management utilitlies. This document covers these features.

2 GUCs

DeepGreen added the following GUCs,

deepgreen.enable (boolean, default true). Enable or disable JIT engine. When disabled, DeepGreen DB should behave exactly the same as Greenplum.

deepgreen.explain_verbosity (integer, 0, 1, or 2, default 0). Control the verbosity of JIT information in the output of explain analyze.

deepgreen.threshold (integer). Because JIT compilation cost some time, for very short queries, the compilation cost will overweight the query execution. This GUC turns off JIT if the query optimizer decides that query execution cost is less this threshold.

deepgreen.xdrive_host (string) Default XDrive host.

deepgreen.xdrive_port (int) Defaulit XDrive port.

deepgreen.version (string) Version string. It is readonly, cannot be set.

deepgreen.rev (string) Revision string. It is readonly, cannot be set.

3 Decimal64 and Decimal128

Decimal64 and Decimal128 implement the 64 and 128 bits decimal floating as defined in IEEE-754 2008. Unlike the the variable length numeric datatype, the 64/128 bits decimal format is much more efficient. Decimal64 can accurately represent 16 decimal digits and Decimal128 can accurately represent 34 decimal digits. All major arithemtic functions and aggreates are supported.

4 LZ4 and ZSTD Compression

User can specify two new compression algorithms 'lz4', or 'zstd' for table and/or column. LZ4 is an extremely fast compression algorithm that can be up to ten times faster than the zlib algorithm. ZSTD is slightly slower than LZ4 but offers better compression ratio.

5 High Performance Regular Expression Engine

DeepGreen added the following regular expression functions. In all the following functions, a pattern must be a constant string of regular expression pattern.

bool re_match(text,pattern) Regular expression match.

text re_match_extract(text,pattern) Regular expression match and return first matching result. If not matched, return NULL.

text[] re_match_extract_array(text,pattern) Regular expression match and return all matching results in an array. If not matched, return NULL.

text[] re_split(text,pattern) Split text using pattern.

text re_replace(text,pattern,replacement) Replace matching part with constant replacement string.

6 Approximate Count Distinct using HyperLogLog

Approximate_count_distinct An aggregate that returns an approximate distinct count on input, within 1% accuracy.

7 JSON

DeepGreen ported PostgreSQL 9.4 JSON type. All JSON functions and operators except table/set returning functions and populate_record, json_agg, are also ported. We expect JSON type will be ported to upstream Greenplum soon.

8 DG Setup

All the features mentioned in this Chapter are implemented as UDT and UDF (User Defined Type and User Defined Functions) for DeepGreen. To enable these features, user should run the dg setup utiltlity. For example,

```
dg setup -all template1
```

We recommend user to run dg setup -all template1 after installing and initiating DeepGreen cluster so that later created database will pickup these features automatically.

dg setup accepts the following options,

-all Setup all features

-decimal Enable decimal64/128 types and related functions.

-dg_utils Enable deepgreen utilities, dg_utils are required for using dg transfer.

-hll Enable approximate count distinct.

-host Database connection information, host.

-pgcryto Enable pgcrypto functions.

-port Database connection information, port.

-re Enable fast regular expresssion engine.

-user Database connection information, user.

-verbose Print verbose info.

9 DG Transfer

Dg transfer is a high performance, scalable data transfer uitility between deepgreen databases. To set up transfer, user should have started the source and destination deepgreen cluster. dg transfer requires an xdrive cluster, usually, an xdrive cluster started on the destination deepgreen cluster. Please refer to dg xdrive on how to initialize, start xdrive cluster. Dg transfer also requires dg_utilts module being properly setup in source database. Once setup properly user can transfer a database using

```
dg transfer -src.host= -src.port= -src.db= -dst.host= -dst.port= -dst.db=
```

dg transfer takes the following options,

-src.host Source deepgreen cluster connection information host.

-src.port Source deepgreen cluster connection information port.

-src.user Source deepgreen cluster connection user name.

-src.db Source database name.

-dst.host Destination deepgreen cluster host.

-dst.port Destination deepgreen cluster port.

-dst.user Destination deepgreen cluster user name.

-dst.db Destination database name.

-xdrive.hosts XDrive hosts to use for transfer.

-xdrive.port If using xdrive cluster running in destintion deepgreen cluster, user can just specify xdrive port number instead of using xdrive.hosts.

-schema The schema to transfer. If not specified, all schemas will be transfered.

-table The table to transfer. If not specified, all tables in the schema will be transfered.

-tablelist A file that each line contains schema.table to transfer.

-schemaonly Transfer database schema only, do not copy data over.

-ifexists=drop/keep/skip If destination table already exists, transfer should drop and create new table, or, keep and use the table, or, skip transfer data for the table.

-keepowner If keep destination table owner.

-index Rebuild index after transfer.

-analyze Run analyze after transfer.

-verify Verify the database, or schema, or table specified in the command contains same data. -verify is a readonly operation. It will not transfer data.

-verbose Print more information during transfer.

10 dg cluster

DG synchronous replication/HA tool. Install, start, stop, failover/failback HA cluster.

11 dg sync

Asynchronous replication for backup, restore, remote site standby.