Citus with Postgresql 13

Overview

- What is Citus?
- Advantages
- Hardware requirements
- Install as extension in Postgresql
- Import sample data
- Citus vs Postgres w/o Citus
- Summary

What is Citus?

Citus is an open source extension to Postgres that distributes data and queries across multiple nodes in a cluster. Because Citus is an extension (not a fork) to Postgres, when you use Citus, you are also using Postgres. You can leverage the latest Postgres features, tooling, and ecosystem.

Citus transforms Postgres into a distributed database with features like sharding, a distributed SQL engine, reference tables, and distributed tables. The Citus combination of parallelism, keeping more data in memory, and higher I/O bandwidth can lead to significant performance improvements for multi-tenant SaaS applications, customer-facing real-time analytics dashboards, and time series workloads. Citus scales horizontally by adding worker nodes, and vertically by upgrading workers/coordinator.

Advantages

Some advantages of Citus for multi-tenant applications:\

- Fast queries for all tenants
- Sharding logic in the database, not the application
- Hold more data than possible in single-node PostgreSQL
- Scale out without giving up SQL
- Maintain performance under high concurrency
- Fast metrics analysis across customer base
- Easily scale to handle new customer signups
- Isolate resource usage of large and small customers

Hardware requirements

- OS: Ubuntu 18.04
- Disk: 500GB (depends on your data) - Memory: 16GB CPU: 8Core
- Install PostgreSQL 13 and the Citus extension.

❖ Install PostgreSQL 13 and the Citus extension

Citus with Postgresql 13

- Add Citus repository for package manager
- curl https://install.citusdata.com/community/deb.sh | sudo bash

```
root@psj          :/home/ubuntu# curl https://install.citusdata.com/community/deb.sh | sudo bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7577  100  7577    0     0   9150      0 --:--:-- --:--:-- --:--:--  9150
Detected operating system as Ubuntu/bionic.
Running apt-get update... done.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Checking for postgresql-13...
Detected postgresql-13...
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/citusdata_community.list... done.
Importing Citus Data gpg key... done.
Running apt-get update... done.

The repository is set up! You can now install packages.
```

- Install the server and initialize db
- sudo apt-get -y install postgresql-13-citus-10.1

```
root@p          /home/ubuntu# sudo apt-get -y install postgresql-13-citus-10.1
Reading package lists... Done
Building dependency tree
Reading state information... Done
postgresql-13-citus-10.1 is already the newest version (10.1.0.citus-1).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

  - As I already installed it earlier, we're all set here.
- Initialize the Cluster. Let's create a new database on disk. For convenience in using PostgreSQL Unix domain socket connections, we'll use the postgres user. This user has access to sockets in /var/run/postgresql
  - sudo su - postgres
- include path to postgres binaries
- export PATH=$PATH:/usr/lib/postgresql/13/bin

```
root@p          /home/ubuntu# sudo su - postgres
postgres@psj-myv-uv-ps01:~$ export PATH=$PATH:/usr/lib/postgresql/13/bin
```

-

```
postgres@p          :~$ cd ~
postgres@p          :~$ mkdir citus
mkdir: cannot create directory ‘citus’: File exists
postgres@p          :~$ initdb -D citus
```

  - Already being done earlier.
- Citus is a Postgres extension. To tell Postgres to use this extension you'll need to add it to a configuration variable called shared_preload_libraries:
  echo "shared_preload_libraries = 'citus'" >> citus/postgresql.conf
- Citus is a Postgres extension. To tell Postgres to use this extension you'll need to add it to a configuration variable called shared_preload_libraries:
-

```
postgres@p          :~$ echo "shared_preload_libraries = 'citus'" >> citus/postgresql.conf
```

- Start the database server. Finally, we'll start an instance of PostgreSQL for the new directory:

Citus with Postgresql 13

- pg_ctl -D citus -o "-p 9700" -l citus_logfile start

```
postgres@[redacted]:~$ cd /usr/lib/postgresql/13/bin
postgres@[redacted]:/usr/lib/postgresql/13/bin$ ./pg_ctl -D /var/lib/postgresql/citus -o "-p 9700" restart
waiting for server to shut down.... done
server stopped
waiting for server to start....2021-08-01 11:20:52.914 +03 [23682] LOG:  number of prepared transactions has not been configured, overriding
2021-08-01 11:20:52.914 +03 [23682] DETAIL:  max_prepared_transactions is now set to 200
2021-08-01 11:20:52.946 +03 [23682] LOG:  starting PostgreSQL 13.3 (Ubuntu 13.3-1.pgdg18.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 7.5.0-
3ubuntu1~18.04) 7.5.0, 64-bit
2021-08-01 11:20:52.946 +03 [23682] LOG:  listening on IPv4 address "10.10.50.47", port 9700
2021-08-01 11:20:52.948 +03 [23682] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.9700"
2021-08-01 11:20:52.958 +03 [23684] LOG:  database system was shut down at 2021-08-01 11:20:52 +03
2021-08-01 11:20:52.972 +03 [23682] LOG:  database system is ready to accept connections
 done
server started
```

- Here, we load the user-facing side of Citus (such as the functions you'll soon call):

```
nocitus=# \c citus_test
You are now connected to database "citus_test" as user "postgres".
citus_test=# psql -p 9700 -c "CREATE EXTENSION citus;"
```

- Verify that installation has succeeded:

```
postgres@[redacted]:/usr/lib/postgresql/13/bin$ psql -p 9700 -c "select citus_version();"
                                              citus_version
-----------------------------------------------------------------------------------------------------
 Citus 10.1.0 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0, 64-bit
(1 row)
```

- At this step, you have completed the installation process and are ready to use your Citus cluster.

- Create databases on **node1** and **node2**:
  - We will create two databases, one with Citus added (**node1**) and another without Citus (**node2**) in order to compare performance: (See script attached).
  - *Create database **citus_test;*** **-- (**check if Citus is included**)**

```
citus_test=# select citus_version();
                                              citus_version
-----------------------------------------------------------------------------------------------------
 Citus 10.1.0 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0, 64-bit
(1 row)
```

  - *Create database **nocitus;*** **-- (**check if Citus is **NOT** included**)**

```
nocitus=# select citus_version();
2021-08-01 13:35:20.459 +03 [24965] ERROR:  function citus_version() does not exist at character 8
2021-08-01 13:35:20.459 +03 [24965] HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
2021-08-01 13:35:20.459 +03 [24965] STATEMENT:  select citus_version();
ERROR:  function citus_version() does not exist
```

  - Before distributing tables, enable some extra features
    - *SET citus.replication_model = 'streaming';*
  - If it doesn't accept then you need to set the following:
    - *SET citus.shard.replication_factor = 'streaming';*
  - Then you need to add Citus extension in the database **citus_test**:

```
citus_test=# CREATE EXTENSION citus;
2021-08-04 13:28:10.534 +03 [26161] ERROR:  extension "citus" already exists
```

    - As you can see, it is already being created.

Citus with Postgresql 13

Import sample data:

The below script will create:

- t_demo with 200.000.000 million records.
- Create a primary key **t_demo**.
- Create an **id** column in t_demo.

```
CREATE TABLE t_demo (data numeric);
CREATE OR REPLACE PROCEDURE insert_data(buckets integer)
LANGUAGE plpgsql
AS $$
   DECLARE
      i int;
   BEGIN
      i := 0;
      WHILE i < buckets
      LOOP
         INSERT INTO t_demo SELECT random()
            FROM generate_series(1, 1000000);
         i := i + 1;
         RAISE NOTICE 'inserted % buckets', i;
         COMMIT;
      END LOOP;
      RETURN;
   END;
$$;
CALL insert_data(200);
# add primary key column id :
ALTER TABLE t_demo ADD COLUMN id SERIAL PRIMARY KEY;
# add id coumnd:
ALTER TABLE t_demo ADD COLUMN id SERIAL PRIMARY KEY;
```

* **Note**: the above is as well created in a normal PostgreSQL installation without Citus being added for comparison purposes.

- Now note the default number of shards (distrbuted_tables) is set to 32. If you plan to change it set the following to the number desired:
  - *set citus.shard_count= '3';*
  - SELECT create_distributed_table(t_demo, 'id');

Citus vs PostgreSQL without Citus:

Citus	Nocitus

`citus_test=# select count(id) from t_demo;`
`   count`
`----------`
`200000000`
`(1 row)`
 	
`Time: 3015.321 ms (00:03.015)`
	`nocitus=# select count(id) from t_demo;`
	`   count`
	`----------`
	`200000000`
	`(1 row)`
	
	`Time: 7328.628 ms`

Time spent: 3.0 seconds	Time spent: 7.3 seconds

Citus with Postgresql 13

```
Timing is on.
citus_test=# select*from t_demo where id>5 and id<=7;
 0.143666494022749 |  7
 0.282534445881453 |  6

Time: 3.690 ms
```

Time spent: 0.0028 seconds

```
citus_test=#  select count(id) from t_demo where id<50000;
 count
-------
 49999
(1 row)

Time: 16.938 ms
```

Time spent: 0.01693 seconds

```
nocitus=# select*from t_demo where id>5 and id<=7;
       data        | id
-------------------+----
 0.184924051631242 |  6
 0.283110986929387 |  7
(2 rows)

Time: 8.623 ms
```

Time spent: 0.0086 seconds

```
nocitus=# select count(id) from t_demo where id<50000;
 count
-------
 49999
(1 row)

Time: 4893.389 ms
```

Time spent: 0.0028 seconds

```
Time: 16.938 ms
citus_test=# update t_demo set "data" = '111' where id < 1000;
UPDATE 999
Time: 29.452 ms
```

Time spent: 0.029 seconds

```
Timing is on.
nocitus=# update t_demo set "data" = '111' where id < 1000;
UPDATE 999
Time: 13448.998 ms
```

Time spent: 13.444 seconds

Summary:

Citus adds superpowers to Postgres as it produces not only faster execution time queries but as well as a high availability of queries across either a single node or multiple clusters. You could always add more distributed tables as much as you need. It also gives the chance to add more nodes (cluster) to your coordinator. As shown above, performance of queries are x10 times faster than a regular Postgres.

Citus with Postgresql 13

Reference:

1. https://www.cybertec-postgresql.com/en/postgresql-parallel-create-index-for-better-performance
2. http://docs.citusdata.com/en/v10.1/installation/multi_node_debian.html#steps-to-be-executed-on-all-nodes
3. https://www.cybertec-postgresql.com/en/how-to-interpret-postgresql-explain-analyze-output/