Citus Setup and Configuration Guide

This document outlines the steps required to set up and run Citus in a distributed environment. Citus is an extension for PostgreSQL that enables horizontal scaling of database tables across multiple nodes.

Installation:

1. Execute the install_citus.sh script to download and install PostgreSQL and Citus.
2. Select the required number of servers for your distributed setup.
3. Run the init-citus-db.sh script to initialize the Citus database.
4. Ensure that there is at least one coordinator node in the cluster. The Citus coordinator stores metadata about the table shards and doesn't store any data.

Starting Citus:

1. Start Citus on the nodes using the following command:

```
/home/stuproj/cs4224b/pgsql/bin/pg_ctl -D /temp/teamb-data -l logfile start
```

2. Use the createdb <name> command on all nodes to create a database. The database name should be the same on all nodes.

Configuration:

1. All nodes in the distributed cluster must be able to communicate with each other. Edit the pg_hba.conf file to specify PostgreSQL access rules using the "host" method for the database. A sample access rule would look like this:

```
host project all 192.168.48.179/32 trust
```

   This rule grants unrestricted access to all roles/users connecting from the specified IP address, effectively bypassing authentication requirements.
2. Edit the postgresql.conf file and add the following line:

```
listen_addresses = '*'
```

   This configuration allows PostgreSQL to listen on all available network interfaces, including all IP addresses associated with the server.

Extension Setup:

To create the Citus extension, run the following command from the PostgreSQL interactive terminal:

```
CREATE EXTENSION citus;
```

Coordinator Node:

Set the coordinator node using the following command:

```
SELECT citus_set_coordinator_host('<ip address>');
```

This command enables horizontal scaling of database tables across multiple nodes and configures the coordinator node's IP address.

Worker Nodes:

Create worker nodes by running the following SQL command:

```
SELECT * FROM citus_add_node('<ip_address>', <port>);
```

This command adds the specified worker node to the Citus cluster, increasing the cluster's capacity and performance.

Table Creation:

Execute the Python script create_tables.py using the command python create_tables.py. This script takes care of creating and distributing tables.

Running Transaction Clients:

Run the following shell scripts to execute the transaction clients in parallel:

```
parallel_node0.sh
parallel_node1.sh
parallel_node2.sh
parallel_node3.sh
parallel_node4.sh
```

Each script runs four transaction clients in parallel, allowing all 20 transaction files to run concurrently.