



Local Executor With PostgreSQL

Pseudo Distributed Mode



What is PostgreSQL

- PostgreSQL is an object-relational database and ACID compliant.
- PostgreSQL is a client-server database, so there is a server process (managing database files and connections, performing actions etc.) and a client which is used to perform database operations.
- PostgreSQL can handle multiple concurrent connections from client in writing as well as in reading mode.
- It implements SQL standard as well as advanced SQL stuff like Window functions.
- Scalable.



What is a Local Executor

- Local Executor executes tasks locally in parallel. It uses the multiprocessing Python library and queues to parallelize the execution of tasks.
- It can run multiple tasks at a time.
- It run tasks by spawning processes in a controlled fashion in different modes on the same machine.
- You can tune the number of processes to spawn by using the parallelism parameter.



Local Executor Strategies

Two strategies depending on the parallelism value:

- `parallelism == 0` which means unlimited parallelism. Every task submitted to the Local Executor will be executed in its own process. Once the task is executed and the result stored in the “result_queue”, the process terminates.
- `parallelism > 0` which means the Local Executor spawn the number of processes equal to the value of `parallelism` at start time using a “task_queue” to coordinate the ingestion of tasks and the work distribution among the workers. During the lifecycle of the Local Executor in this mode, the worker processes are running waiting for tasks, once the Local Executor receives the call to shutdown the executor, a poison token is sent to the workers to terminate them.



Local Executor with PostgreSQL

- If you remember the last lesson, we have seen that SQLite does not accept more than one writer which makes multiple tasks to run in parallel impossible.
- PostgreSQL is a perfect fit for Local Executor since it does accept multiple connections in both ways, writing and reading allowing for task parallelism.

Sidenote: A Sequential Executor could be thought (thought because the implementation is not exactly the same) as a Local Executor with limited parallelism sets to 1.



Let's Try This!

Let's try this in a concrete example