supervisor_thread_count

Parameters Affecting the Supervisor Thread Pool Size

The supervisor thread count is meant to increase to match transient bursts in demand, then decrease again. The 3 parameters which directly influence the number of supervisor threads are:

- supervior_max_threads
- supervisor_idle_threads
- supervisor_max_clients

Supervisor Thread Pool Size Behavior

The **supervisor_max_threads** and **supervisor_idle_threads** set the upper and lower bounds on the size of the supervisor process pool, while **supervisor_max_clients** affects the lifespan length of a single supervisor process, which indirectly affects how quickly the process pool shrinks after the burst in demand.

When a request comes in, the main supervisor process looks for an idle process in the thread pool. If there is an idle thread its assigned the request, but if there are no idle process, another supervisor process is spawned, added to the supervisor thread pool, and assigned the request.

The **supervisor_idle_clients** indirectly affects the number of threads in existence by setting the number of transactions a single thread will handle before quitting. Once a supervisor process has handled a request, it will check its **supervisor_max_clients** value; if it has not yet processed that many requests it will go into an idle state awaiting another request, otherwise it will exit.

Setting the **supervisor_max_clients** to a low value causes the thread count to decrease a bit quicker after a transient demand burst, but at the expense of possibly excessive thread creation - it's a balance between available memory and process creation overhead.

Spawning a new Supervisor Thread

When all supervisor processes are tied up doing something (which may be as simple as being blocked behind the DB performing a long operation) and another request comes in, the main supervisor thread will spawn another thread up to the **supervisor_max_threads** value.

When the Thread Pool Remains at the Upper Limit

When the process count hits the upper bound and stays there for some time is indicative of a performance issue in the supervisor host configuration.

Increasing the **supervisor_max_threads** count when you see this behavior is like increasing the limit on a credit card; it may be appropriate to increase it in small increments to see if the host is capable of handling the additional processes, but at some point the process count reaches a tipping point where more supervisor processes make the problem even worse, so if you were to graph the count you'd see it gradually increase, then the curve goes almost straight up.

At this point you may want to open a support case with PipelineFX to further investigate any performance bottlenecks.