Yukun Ma
Week #4: Connection Pooling
Page #1

Problem Statement

This paper will show how the connection pooling service solves the huge expense caused by the database connection. Two potential problems brought by connection pools will also be evaluated.

References

[1] Wikipedia. Connection pool - wikipedia. `https://en.wikipedia.org/wiki/Connection_pool`. Accessed March 21, 2018.

[2] Brett Wooldridge. About pool sizing. `https://github.com/brettwooldridge/HikariCP/wiki/About-Pool-Sizing`. Accessed March 21, 2018.

Overview of Connection Pooling

Database connection is a key and expensive resource, which is particularly prominent in dynamic web applications. It's key because an application usually performs many requests to the database in a very short period of time. And database connection is expensive due to the fact that just creating the connection consumes a lot of memory, time and queries. For a typical dynamic web application, in most cases, it takes much more time to just create a connection than to make a query. So the management of database connections can significantly affect the scalability of the entire application.

Database connection pool is proposed for the issue that a database connection is much too expensive to create than just to perform the SQL query. In an application that uses a connection pool, a connection pool maintains a cache of database connections in which the connections can be reused for future requests.[1]

Critical Thinking

Although a connection pool sometimes can significantly increase the time efficiency of requests to a database, it may cause some problems.

When using a connection pool, you need to pay attention to the configuration of many parameters, such as the minimum number of connections, the maximum number of connections, the maximum idle time, the connection timeout time, and the number of timeout retry connections. These parameters are relevant to servers, applications and users and will certainly affect the efficiency of the connection pool. It's a rather complicated job for a newbie who is not familiar with connection pools.

Another shortcoming of connection pools is that even though users have no requests or few requests database connections are still alive and consume a lot of memory.

Page #2

Question

When we use a connection pool in a database, the first issue is how to set various parameters of the connection pool. Among the different parameters of the connection pool, a very important parameter is the number of database connections in the connection pool. So how to set the number of connections for a connection pool?

Method

Describe how you are going to answer your own question stated above.

Analysis and Discussion

The most straight-forward opinion is to build as many database connections as possible in a connection pool. When considering only the huge cost of database connection establishment, this idea seems very reasonable, because enough connections mean that when there are many clients requesting at the same time, the application will not spend too much time on the establishment of the database connection.

However, when we take the context switching costs between different operating system processes into account, we will find that this seems to be a bit wrong. Because when multiple programs run concurrently on a single core, the operating system actually schedules the time slices between different processes, and the processes switch when the time slices switch. If we have 4 cores in the server, then there are at most 4 processes are really parallel running. And this means there are at most 4 SQL queries are executing at the same time.

According to [2], if we use 10000 connections for 10000 users on a 4-core server, then we will not be able to improve operational efficiency, but will also reduce efficiency.

In conclusion, the best choice for the appropriate choice of the number of database connections in a connection pool is almost the number of CPU cores. I recommend setting the number of connections to the number of neighbors in the CPU core count and performing some stress tests to obtain the optimal number of connections.