Connection Performance in PostgreSQL

Last updated by | Lisa Liu | Nov 6, 2020 at 10:35 AM PST

Connection Performance in PostgreSQL

Friday, September 6, 2019
11:09 AM

Happy to help you dig into the performance you're seeing. First a bit on the initial time to establish a connection. There are two factors in the initial process of establishing a connection.

The first is that we have a central gateway that allows us to provide built in high availability. You can see more about this architecture at https://docs.microsoft.com/en-us/azure/postgresql/concepts-connectivity-architecture. The second, is that because we ensure everything is encrypted in transit Postgres has to do an initial TLS negotiation. This is a recommended practice anytime running within the cloud. The combination of these two items means the cost of establishing a connection is often between 50 and 100ms. To reduce this time of establishing an initial connection we recommend using a connection pooler. Most application frameworks have an option for this. But, in addition to a pooler within your application it is good to have a server side pooler to minimize the overhead of each connection. Pgbouncer is our recommended approach. You can find some follow-on on reading for these topics at:

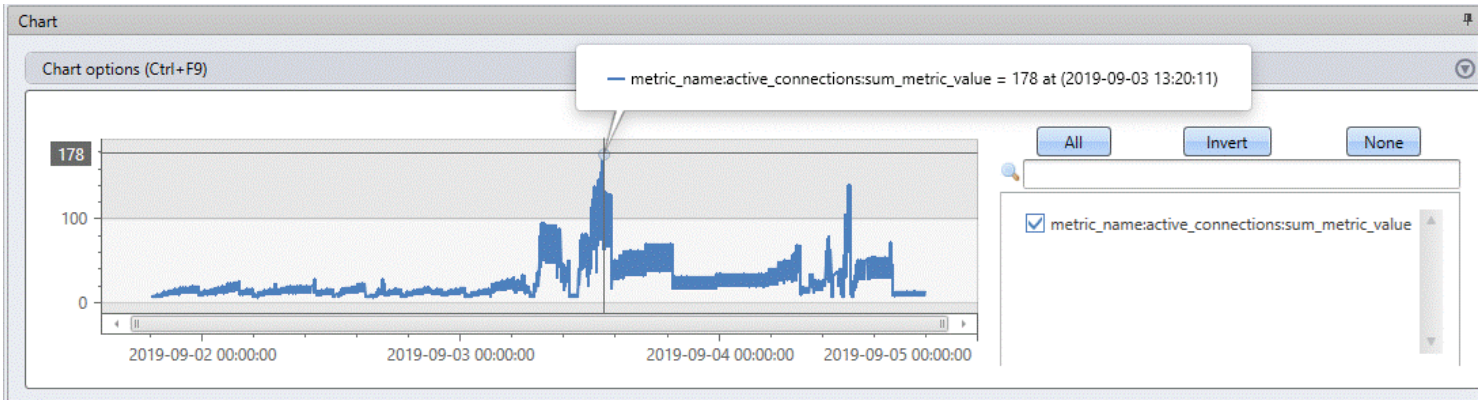
- https://techcommunity.microsoft.com/t5/Azure-Database-for-PostgreSQL/Not-all-Postgres-connection-pooling-is-equal/ba-p/825717
- https://techcommunity.microsoft.com/t5/Azure-Database-for-PostgreSQL/Connection-handling-best-practice-with-PostgreSQL/ba-p/790883
- https://techcommunity.microsoft.com/t5/Azure-Database-for-PostgreSQL/Steps-to-install-and-setup-PgBouncer-connection-pooling-proxy/ba-p/730555
- https://techcommunity.microsoft.com/t5/Azure-Database-for-PostgreSQL/Performance-Troubleshooting-Basics-on-Azure-Database-for/ba-p/819227

This query which is covered in the second blog post above will highlight your idle connections, and if you see a double digit number there then connection pooling with pgbouncer could help:

```
SELECT count(*),state
FROM pg_stat_activity GROUP BY 2;
```

Beyond that, it would be helpful if you could provide the query plan for the query you are running. You can do this by adding explain analyze before the query and then providing the output of the query plan.

I have anoticed a high number of active connections against the DB server over the past 7 days which exceeds the limits of your DB SKU. The maximum connections allowed for a General Purpose 2 Vcore server is 150 connections and you have crossed that limit several times. Please find screenshot below confirming the same. The limits for the postgres DB can be found in this link: https://docs.microsoft.com/en-us/azure/postgresql/concepts-limits



If you workload requires these number of high connections against your database, We would recommend you to please scale up your resources to the next higher SKU.

If you can send the query plan we can help dig in further.

Created with Microsoft OneNote 2016.

How good have you found this content?