

Performance best practices

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

Performance best practices

Wednesday, February 20, 2019

11:40 AM

- [Performance best practices for using Azure Database for PostgreSQL – Connection Pooling](#)
- [Performance best practices for using Azure Database for PostgreSQL](#)
- [Performance updates and tuning best practices for using Azure Database for PostgreSQL](#)
- [Performance troubleshooting best practices using Azure Database for PostgreSQL features](#)
- [Client Query Caching for applications with less frequent changes](#)
- **For latency between Compute and Storage** – This latency is minimal but can get amplified if compute and storage end up in separate datacenters in Azure regions. Generally, the recommendation for customers writing new cloud native applications is to batch the transactions and minimize chatty workload. Chatty workload is when you do a lot of singleton commits which causes log flush operations frequently. In addition, it is also important to ensure you choose the storage tier based on the IOPs required and not the absolute size of the database. For instance if you have a 100GB but if the IOPs requirement is 5K IOPs, it makes sense to choose a 2TB storage tier which provides around 6K IOPs (some additional buffer for backup and other backend operations). Overall we are also working behind the scenes with Compute and storage team to minimize this latency by collocating the compute and storage in the same datacenter for a given server which can make significant difference in highly latency sensitive applications.
- **Connection pooling:** <https://argonsys.com/microsoft-cloud/library/performance-best-practices-for-using-azure-database-for-postgresql-connection-pooling/>

Created with Microsoft OneNote 2016.

How good have you found this content?

