High IOPs Consumption

Last updated by | Hamza Aqel | Feb 22, 2022 at 8:23 AM PST

This TSG is part of GT for any change please contact haaqel@microsoft.com

IOPs:

The storage you provision is the amount of storage capacity available to your Azure Database for PostgreSQL server. The storage is used for the database files, temporary files, transaction logs, and the PostgreSQL server logs. The total amount of storage you provision also defines the I/O capacity available to your server.

Storage attributes	Basic	General Purpose	Memory Optimized
Storage type	Basic Storage	General Purpose Storage	General Purpose Storage
Storage size	5 GB to 1 TB	5 GB to 16 TB	5 GB to 16 TB
Storage increment size	1 GB	1 GB	1 GB
IOPS	Variable	3 IOPS/GB Min 100 IOPS Max 20,000 IOPS	3 IOPS/GB Min 100 IOPS Max 20,000 IOPS

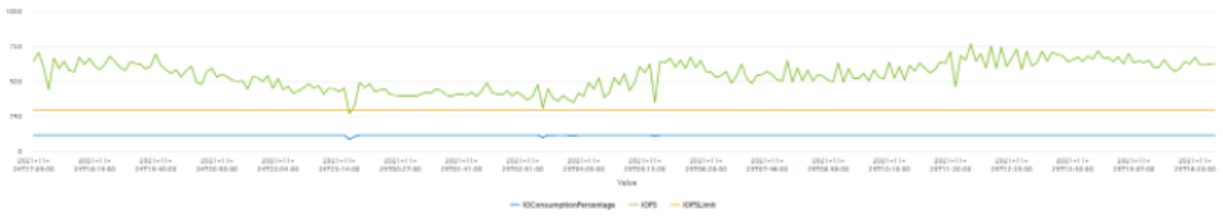
Storage up to 16TB and 20,000 IOPS is supported in the following regions: East US, East US 2, Central US, West US, North Central US, South Central US, North Europe, West Europe, UK South, UK West, Southeast Asia, East Asia, Japan East, Japan West, Korea Central, Korea South, Australia East, Australia South East, West US 2 and West Central US.

All other regions support up to 4TB of storage and 6000 IOPS.
The Basic tier does not provide an IOPS guarantee. In the General Purpose and Memory Optimized pricing tiers, the IOPS scale with the provisioned storage size in a 3:1 ratio.

You can monitor the IOPs in ASC → storage tab

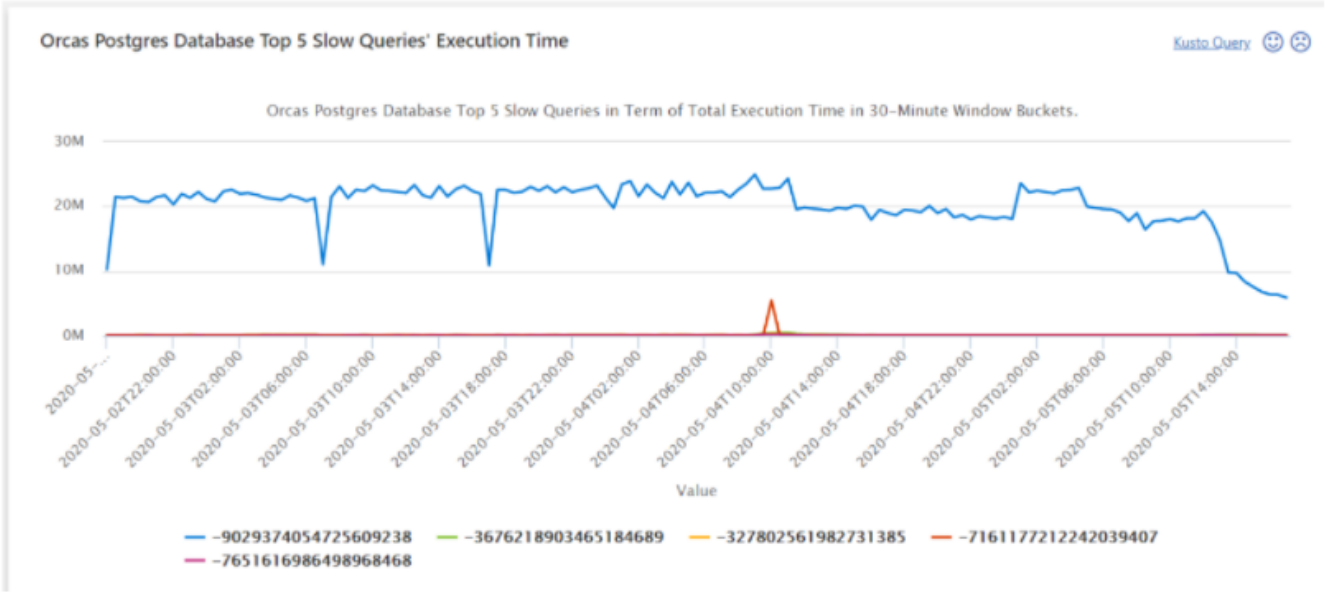
You can check the IOPs consumption from our ASC (storage tab):

for PFS servers:



and for XIO :

if you can see the customer reaching the limit , please ask the customer to increase the storage size to get more IOPs , and to check his workload to get more details about the reason the customer can leverage the Query Store to determine which queries correlating with the IOPs consumption:



and you can share these queries (Query IDs) with the customer, if the customer did not enable the query store , you can use the below queries from pg_stat_statements from the customer side :

Top IO in one call:

select userid::regrole, dbid, query from pg_stat_statements order by (blk_read_time+blk_write_time)/calls desc limit 5;

and if the issue still occurring , it is worth to run pg_stat_activity from customer side too to check what are the current active sessions running to get more details :

SELECT pid, user, client_addr, pg_stat_activity.query_start, now() - pg_stat_activity.query_start AS query_time, query, state, wait_event_type, wait_event FROM pg_stat_activity WHERE state='active'

How good have you found this content?