

High Memory Utilization

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


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


Memory Troubleshooting

High memory is not considered an issue if it did not hit 100%, though, 95% is considered concerning. If you started seeing out of memory errors in sandbox this is a problem and may result slow performance or failover (crash for the server). We usually need to correlate the memory usage with reboot, customer login and sandbox log to understand what contribute the high memory and if it is really impact the performance.

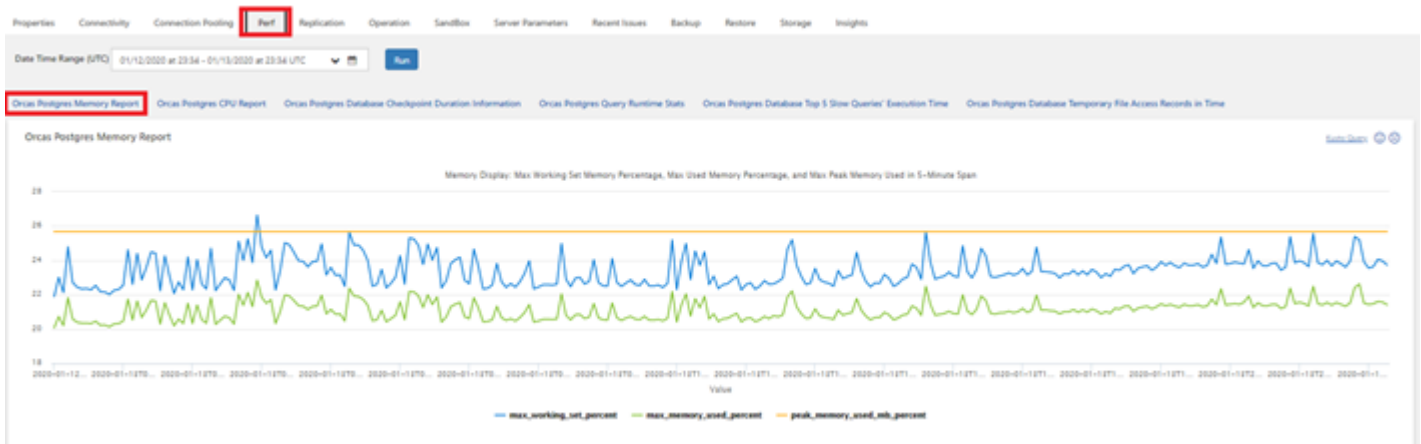
Customers can troubleshoot High memory utilization by:


1- Query Store: Turn it in and review top hitters that can contribute to high memory utilization

2- Customer can study how memory is used by PG shared_buffers by utilizing pg_buffercache extension:
[PostgreSQL: Documentation: 12: F.24. pg_buffercache](#) 

3- Customer can study how memory limits are affecting query performance by looking into shared_blks_hit and shared_blks_read data: [PostgreSQL: Documentation: 12: F.29. pg_stat_statements](#) 

from our side ASC shows memory used under the Perf tab



and you can refer to this [Workload TSG](#)  and correlate with the customer memory consumption , where you can check in ASC (Perf Tab) as in the above screenshot.

You may also check memory consumption in Kusto:

```
//memory
MonResourceMetricsProvider
| where LogicalServerName =~ "{ServerName}"
| where TIMESTAMP >= ago(7d)
//| where TIMESTAMP between ( datetime(2019-11-22 08:48:20) .. datetime(2019-11-22 10:48:20.2485224))
| where event == "dk_application_load"
| extend memory_percentage = working_set_percent
```

```
| project TIMESTAMP, memory_percentage
| render timechart
```

Solution for out of memory error in sandbox

Server crashed because of server out of memory. Usually, this usually happen for small SKU (1-2 vcores) when customer having load which occupies extra memory. When PostgreSQL cannot allocate memory in some code path, PostgreSQL engine will crash.

When customer realized memory almost full (> 90%), he can choose one of ways to prevent out of memory issue

- Change SKU to memory optimized
- Change to higher SKU with larger VCores with larger memory
- Avoid keeping transactions open too long
- Avoid idle connections: Idle connections are not free, they consume memory

You can check for this using the following query

```
//sandbox postgresql
MonRdmsPgSqlSandbox
| where LogicalServerName == "{ServerName}"
| where text contains "out of memory"
| project originalEventTimestamp, text
```

Please note that Azure database for PostgreSQL service consumes memory to achieve as many cache hit as possible, often customers can see memory utilization hovering around 90% and that is normal as long as the server is not throwing Out of Memory errors.

Customers should setup alerts if the memory reached 90%

Memory is assigned per vCore and its documented here: <https://docs.microsoft.com/en-us/azure/postgresql/concepts-pricing-tiers>

Basic Tier server running Out of Memory

Please note that out of memory errors are common in Basic Tier servers due to the low memory available in this SKU,

In order to see those errors, please enable Server loggings and you will start seeing errors as "Out Of memory", to enable server logging see: <https://docs.microsoft.com/en-us/azure/postgresql/concepts-server-logs>

You can also view the server memory utilization using the server metrics pane: <https://docs.microsoft.com/en-us/azure/service-health/resource-health-overview>

Few advises to avoid out of memory errors:

- 1- Avoid keeping transactions open too long
- 2- Avoid idle connections: Idle connections are not free, they consume memory.
- 3- implement connection pooling to better handle your connections
- 4- Use General Purpose SKU instead of Basic if none of the previously mentioned steps helped.

Note:

if you find that there is no customer workload correlated , and a slight increase in memory consumption in daily basis and the memory consumption did not reach the critical limit (<90%) , please share the below as RCA :

Most databases including SQL Server and PostgreSQL load as many data pages into memory as possible to optimize performance. The server does not optimize the reclamation of cached and temp memory that can be reused. As a result memory growth over time is common. This growth is more pronounced for intensive workloads and for long-running database connections; that is why we recommend recycling connections periodically, such as setting a max lifetime for pooled connections of no more than 2 hours. Single Server PG

uses a custom storage caching layer in addition to the PostgreSQL buffer-pool. This helps improve database read performance, giving the ability to access the pages directly from the filesystem in-memory cache before going to remote storage. This layer is using up to 50% of total memory; PG commits more memory on top.

We investigated all parts of our stack using a variety of techniques and have not been able to find any memory leaks. There is a small leak in Open Source PG which does not explain your memory situation; the fix is queued for the next deployment train. The cache layer and PG's page-loading behavior account for the memory growth you experienced; this is not a leak. Memory profiles inching up to 70-80% on customer servers are common, but if this results in 90-95% memory usage please consider upgrading to a higher sku to avail of more CPU, memory and connections

Created with Microsoft OneNote 2016.

How good have you found this content?

