<https://cloud.google.com/sql/docs/sqlserver/optimize-high-memory-usage>

This document explains how to review and optimize a Cloud SQL for SQL Server instance if that instance is identified by the underprovisioned instance recommender as having high memory consumption.

## SQL Server memory

SQL Server memory can be divided into the following:

* [Caches](https://cloud.google.com/sql/docs/sqlserver/optimize-high-memory-usage#cache)
* [Fixed memory](https://cloud.google.com/sql/docs/sqlserver/optimize-high-memory-usage#fixed-memory)
* [SQL Server overhead](https://cloud.google.com/sql/docs/sqlserver/optimize-high-memory-usage#sql-server-overhead)
* [In-Memory online transactional processing (OLTP)](https://cloud.google.com/sql/docs/sqlserver/optimize-high-memory-usage#oltp)

### Caches

These are objects on a disk that can be reloaded, such as database pages and stored procedures. As a result, the SQL Server can grow and shrink these objects based on memory utilization. Caches include buffer pools and plan caches.

### Fixed memory

Fixed memory can grow and shrink. It only shrinks when not in use; for example, when the number of connections drops or the number of queries executing decreases. It's different from caches. If there is not enough fixed memory, SQL Server can run out of memory. Fixed memory includes connection memory and memory grants.

### SQL Server overhead

SQL Server overhead includes threads and stacks.

### In-Memory OLTP

In-Memory OLTP includes In-Memory tables and In-Memory filegroups.

The memory consumption by SQL Server is controlled by setting maximum server memory and memory.memory.limitmb. The memory.memory.limitmb parameter is set by Cloud SQL automatically.

To learn more about memory.memory.limitmb, see the [Microsoft documentation](https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-configure-mssql-conf?view=sql-server-ver16#memorylimit).

You must set the max server memory limit to a suitable value. In SQL Server, if the max server memory is not set, the database pages can consume most of the memory close to and up to 100%. This can sometimes be misleading.

## Memory optimization options

To determine if an instance needs more memory tuning, do the following:

* [Set the value of max server memory](https://cloud.google.com/sql/docs/sqlserver/flags). The general recommendation is to set the max server memory to about 80% to prevent SQL Server from consuming all the available memory. For instances that have large amounts of memory, 80% might be too low and could lead to wasted memory.
* Monitor Page life expectancy

Page life expectancy indicates the amount of time, in seconds, that the oldest page stays in the buffer pool. This value should be more than 300 as recommended by Microsoft. If it consistently falls below 300, it could be an indication that the instance is facing high memory utilization. Run the following query to monitor Page life expectancy.

SELECT   
  [object\_name],  
  [counter\_name],  
  [cntr\_value]   
FROM   
  sys.dm\_os\_performance\_counters  
WHERE   
  [object\_name]   
LIKE   
  '%Manager%'  
AND   
  [counter\_name] = 'Page life expectancy'

* Check Memory Grants Pending

Memory Grants Pending specifies the total number of processes waiting for a workspace memory grant. Run the following query to check Memory Grants Pending. If this query consistently shows grants pending, then it indicates high memory utilization. You can reduce it by querying the database waits and tuning any statement that's waiting on memory.

SELECT  
  @@SERVERNAME AS [Server Name],  
  RTRIM([object\_name]) AS [Object Name],  
  cntr\_value AS [Memory Grants Pending]  
FROM   
  sys.dm\_os\_performance\_counters WITH(NOLOCK)  
WHERE  
  [object\_name]   
LIKE   
  N'%Memory Manager%'  -- Handles named instances  
AND   
  counter\_name = N'Memory Grants Pending'

Was this helpful?

Send feedback