**Identify Active Long running Queries:**

Get the list of current long running queries, run either one of the following:

Sp\_whoisactive or SP\_Blitzwho

Table

Description automatically generated with medium confidence

Get the list of blocked queries:

EXEC sp\_WhoIsActive @find\_block\_leaders = 1,@sort\_order = '[blocked\_session\_count] DESC'

**Identify Long running Queries history:**

View log running queries in last 2 hours (Enabling querystore is a prerequisite for this script)

SELECT TOP 50 rs.avg\_duration/1000000 as [avg\_duration\_Sec], rs.max\_duration/1000000 as maxDurationSec,qt.query\_sql\_text,q.query\_id,p.plan\_id,count\_executions,  
object\_name(q.object\_id) as [objectname],rs.avg\_logical\_io\_reads,  
rs.avg\_physical\_io\_reads,rs.avg\_tempdb\_space\_used/128 as [avg\_tempdb\_space\_used\_MB],

rs.avg\_query\_max\_used\_memory/128 as [avg\_query\_max\_used\_memory\_MB],avg\_cpu\_time

FROM sys.query\_store\_query\_text AS qt

JOIN sys.query\_store\_query AS q

ON qt.query\_text\_id = q.query\_text\_id

JOIN sys.query\_store\_plan AS p

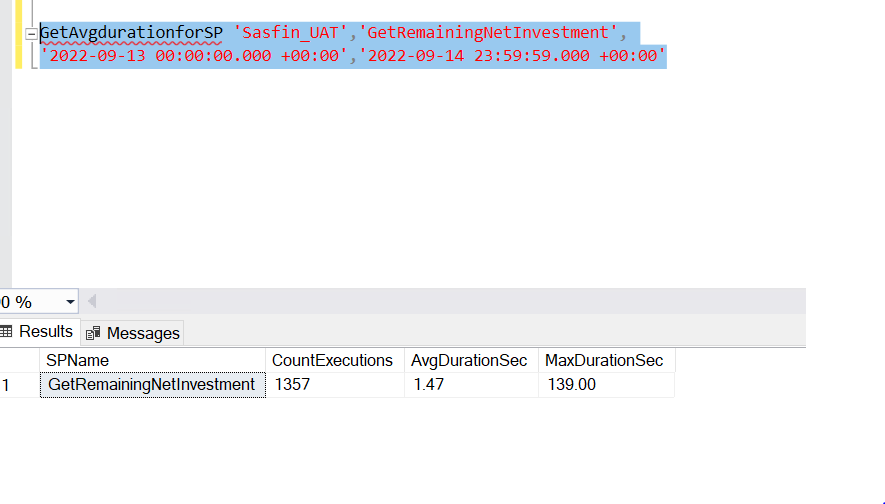
ON q.query\_id = p.query\_id

JOIN sys.query\_store\_runtime\_stats AS rs

ON p.plan\_id = rs.plan\_id

WHERE rs.last\_execution\_time > DATEADD(MINUTE, -120, GETUTCDATE())

ORDER BY rs.max\_duration DESC;  
  
Get the average duration for an SP in specific timeframe:  
GetAvgdurationforSP 'DatabaseName','SPName',  
'FromTime-UTC','ToTime-UTC'



**Identify IOPS/Throughput Bottleneck (Only for SQL Managed Instance):**

Get IOPS/throughput for all the databases, run the below SP. (If any of the databases are above MaxIOPS or MaxthoughputLimitMBPS, then there’s a bottleneck with IOPS)

Getthroughputstats

**Identify Bottleneck from SQL Server waittypes:**

To get the SQLServer waittypes, please run  
  
Getdbperformance 'waits'

If any of the below waittypes with very high percentage is observed, it indicates the following:  
  
**RESOURCE\_SEMAPHORE/RESOURCE\_SEMAPHORE\_QUERY\_COMPILE:** There’s a bottleneck with CPU/Memory on SQL Server. Run sp\_whoisactive to identify the long running query. Run Getdbperformance 'memoryqueries' to get queries using high memory.

**ASYNC\_NETWORK\_IO** : Bottleneck with Client/Application side. Check Application server CPU/Memory usage.  
 **LCK\_M\_U or LCK\_M\_IX or LCK\_M\_X:** There’s Locking/Blocking at SQLServer level. Run sp\_whoisactive to identify root blocker.

**PageIOLatch\_\***: SQL server is waiting on disk. Very high percentage of this waittype indicates disk IO pressure.  
  
**PageLatch\_\***: Check for the database id related to this waittype. If the tempdb is under pressure, look for queries using tempdb/make sure tempdb is setup asper the best practices. Consider increasing the number of files.  
exec Gettempdbusage  
**Identify Bottleneck from Database server with Performance Counters:  
(Applicable only for Azure SQL VMs/Onpremise)**

Setup and run Windows Performance Counters for a fixed timeframe to monitor disk/cpu/memory usage on DB server/App server.

**Get list of Deadlocks in last 24 hrs:**

For Onprem/SQL VMs,  
  
OC\_GetDeadlockinfo 24

For SQL MIs,

OC\_GetDeadlockinfoAzure 24

**Get tuning recommendations in Azure SQL MI:**

To get the list of queries which were automatically pinned and get the list of queries which needs to be pinned, run below sp.

GetTuningRecommendations 'DatabaseName'