**Slow running query because of memory issue**

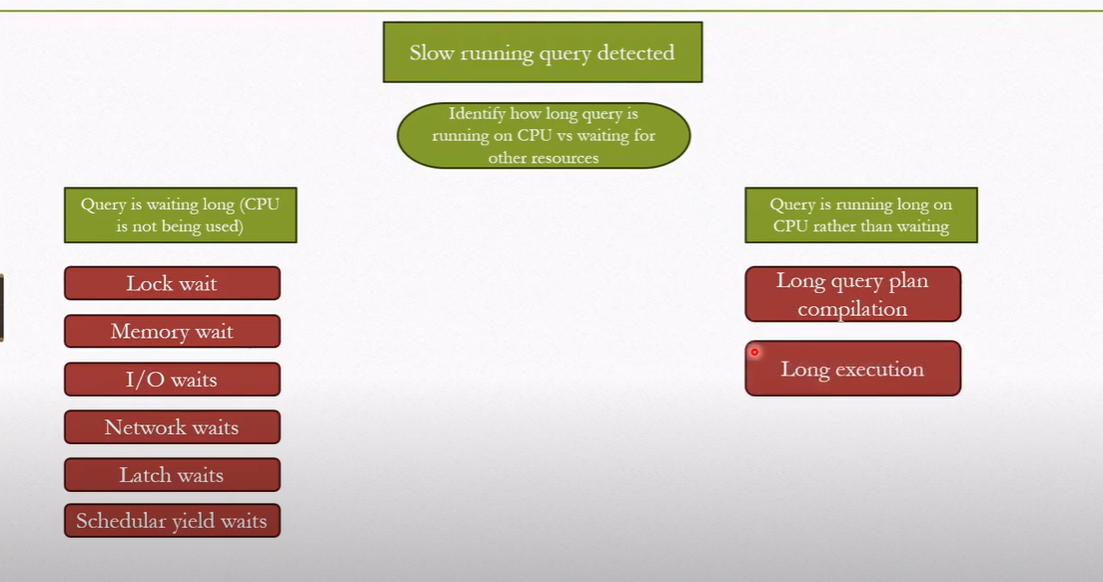
**Memory Grant wait**

When a slow running query is detected: identify how long query is running on CPU vs waiting for other resources.

Query is waiting long (but CPU is not being used) – then find the wait type.

Lock-wait, memory wait, IO wait, Network waits, latch waits, scheduler yield waits.

Lets focus on Memory wait.



CPU related

Non-CPU related

The process in assigning memory.

A diagram of a computer

Description automatically generated

Check the query plan – how much memory is granted

Select \* from sys.messages

Order by message\_id

Right click on select see memory

A screenshot of a computer

Description automatically generated

The below query lists all wait types along with the duration, CPU time and statement

Select

req.session\_id

,req.wait\_type

,req.total\_elapsed\_time as duration\_ms

,req.cpu\_time as cpu\_time\_ms

,req.total\_elapsed\_time - req.cpu\_time as wait\_time

,substring (replace (replace (substring (st.text, (req.statement\_start\_offset/2) + 1,

((case statement\_end\_offset

When -1

Then datalength(st.text)

Else req.statement\_end\_offset

End - req.statement\_start\_offset)/2) + 1), char(10), ' '), char(13), ' '),

1, 512) as statement\_text

From sys.dm\_exec\_requests as req

Cross apply sys.dm\_exec\_sql\_text(req.sql\_handle) AS ST

Order by req.total\_elapsed\_time Desc;

The below query shows how many sessions the number of granted (grantee\_count) and waiters (waiter\_count) of a wait-type

Select

Pool\_id,

Total\_memory\_kb

, available\_memory\_kb

,granted\_memory\_kb

,grantee\_count, waiter\_count

, resource\_semaphore\_id

From sys.dm\_exec\_query\_resource\_semaphores rs

How much memory by each session: session\_id, requested memory, granted, used, wait time, sql text

Select

Session\_id, requested\_memory\_kb, granted\_memory\_kb, used\_memory\_kb,

Queue\_id, wait\_order, wait\_time\_ms, is\_next\_candidate,pool\_id, text, query\_plan

From sys.dm\_exec\_query\_memory\_grants

Cross apply sys.dm\_exec\_sql\_text(sql\_handle)

Cross apply sys.dm\_exec\_query\_plan(plan\_handle)

If the query runs in the past, no longer now

Use the plan cache history to analyze which query had high memory request

select top 20

SUBSTRING(st.text, (QS.statement\_start\_offset/2) + 1,

((case statement\_end\_offset

when -1 then datalength(st.text)

Else QS.statement\_end\_offset END

- QS.statement\_start\_offset)/2) + 1) as statement\_text

,Convert(decimal (10,2), max\_grant\_kb /1024.0) as max\_grant\_mb

,Convert(decimal (10,2), min\_grant\_kb /1024.0) as min\_grant\_mb

,Convert(decimal (10,2), (total\_grant\_kb / execution\_count) /1024.0) AS avg\_grant\_mb

,Convert(decimal (10,2), max\_used\_grant\_kb /1024.0) as max\_grant\_used\_mb

,Convert(decimal (10,2), min\_used\_grant\_kb /1024.0) as min\_grant\_used\_mb

,Convert(decimal (10,2), (total\_used\_grant\_kb / execution\_count) /1024.0) AS avg\_grant\_used\_mb

,Convert(decimal (10,2), (total\_ideal\_grant\_kb/ execution\_count) /1024.0) AS avg\_ideal\_grant\_mb

,Convert(decimal (10,2), (total\_ideal\_grant\_kb/ 1024.0)) AS total\_grant\_for\_all\_executions\_mb

,execution\_count

from sys.dm\_exec\_query\_stats QS

cross apply sys.dm\_exec\_sql\_text(QS.sql\_handle) as ST

Order By max\_grant\_kb DESC