

# Blocking (Managed Instance)

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:42 PM PDT

```
// Long running transactions
MonDmTranActiveTransactions
| where AppName contains "<AppName>"
| where TIMESTAMP >= {startTime} and TIMESTAMP < {endTime}
| where session_id != -1
| extend duration_hour = (end_utc_date - transaction_begin_time) / time(1h)
| summarize max_duration_hour = arg_max(duration_hour, NodeName, session_id, transaction_id,
transaction_begin_time, transaction_type, transaction_state, program_name, status
) by user_db_name, database_id, accessed_tempdb
| sort by max_duration_hour desc

//*****
// blocking
//*****
// B.01
//blocking task vs active task count
MonGovernorWorkloadGroups
| where AppName == "{AppName}" and NodeName == "{NodeName}"
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and AppName =~ "{AppName}" and NodeName =~ "{NodeName}"
| where name startswith "SloPriDGroup.DBId" or name startswith "SloPriS" or name startswith
"UserPrimaryGroup"
| project TIMESTAMP, active_request_count, blocked_task_count
| render timechart
// B.02
// Identify lead blocker
let blockingchain=MonBlockedProcessReportFiltered
| where LogicalServerName =~ "{LogicalServerName}"// and database_name =~ "{LogicalDatabaseName}"
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| extend monitorLoop = extract("monitorLoop=\"([0-9]+)\"", 1, blocked_process_filtered, typeof(int))
| parse blocked_process_filtered with anystr3:string "<blocked-process>" blockee "</blocked-process>"
discard1
| parse blocked_process_filtered with anystr4:string "<blocking-process>" blocker "</blocking-process>"
discard2
| extend blockee_session_id = extract("spid=\"([0-9]+)\"", 1, blockee, typeof(int))
| extend blockee_status = extract("status=\"(.*)\"", 1, blockee, typeof(string))
| extend blockee_waittime = extract("waittime=\"([0-9]+)\"", 1, blockee, typeof(int))
| extend blockee_trancount = extract("trancount=\"([0-9]+)\"", 1, blockee, typeof(int))
| extend blockee_lasttranstarted = extract("lasttranstarted=\"(.*)\"", 1, blockee, typeof(datetime))
| extend blockee_queryhash = extract("queryhash=\"(.*)\"", 1, blockee, typeof(string))
| extend blockee_isolationlevel = extract("isolationlevel=\"(.*)\"", 1, blockee, typeof(string))
| extend blockee_lastattention = extract("lastattention=\"(.*)\"", 1, blockee, typeof(datetime))
| extend blockee_lastattention = extract("lastattention=\"(.*)\"", 1, blockee, typeof(datetime))
```

```

| extend blockee_lastbatchstarted = extract("lastbatchstarted=\"(.*)\"", 1, blockee, typeof(datetime))
| extend blockee_waitresource = extract("waitresource=\"(.*)\"", 1, blockee, typeof(string))
| extend blocker_session_id = extract("spid=\"([0-9]+)\"", 1, blocker, typeof(int))
| extend blocker_status = extract("status=\"(.*)\"", 1, blocker, typeof(string))
| extend blocker_waittime = extract("waittime=\"([0-9]+)\"", 1, blocker, typeof(int))
| extend blocker_trancount = extract("trancount=\"([0-9]+)\"", 1, blocker, typeof(int))
| extend blocker_lasttranstarted = extract("lasttranstarted=\"(.*)\"", 1, blocker, typeof(datetime))
| extend blocker_queryhash = extract("queryhash=\"(.*)\"", 1, blocker, typeof(string))
| extend blocker_isolationlevel = extract("isolationlevel=\"(.*)\"", 1, blocker, typeof(string))
| extend blocker_lastattention = extract("lastattention=\"(.*)\"", 1, blocker, typeof(datetime))
| extend blocker_lastattention = extract("lastattention=\"(.*)\"", 1, blocker, typeof(datetime))
| extend blocker_lastbatchstarted = extract("lastbatchstarted=\"(.*)\"", 1, blocker, typeof(datetime))
//| project PreciseTimeStamp, monitorLoop, blockee_session_id, blocker_session_id
| order by monitorLoop asc nulls last;
let leadblockers=
blockingchain
| join kind= rightanti (
    blockingchain
) on $left.monitorLoop == $right.monitorLoop and $left.blockee_session_id==$right.blocker_session_id
| extend lead_blocker_session_id = blocker_session_id
| distinct monitorLoop, lead_blocker_session_id;
leadblockers
| join kind= inner (
    blockingchain
) on $left.monitorLoop == $right.monitorLoop and $left.lead_blocker_session_id==$right.blocker_session_id
| order by TIMESTAMP asc nulls last
| project PreciseTimeStamp, lead_blocker_session_id, blocker_session_id, blocker_trancount,
blocker_lastattention, blocker_queryhash=toupper(blocker_queryhash), blocker_status,
blockee_session_id, blockee_waitresource, blockee_queryhash=toupper(blockee_queryhash),
blockee_waittime, blockee_status, blocked_process_filtered=toupper(blocked_process_filtered)
//| top 20 by blockee_waittime desc
| order by PreciseTimeStamp asc nulls last

// B.02
// Top lock waits by query
MonWiQdsWaitStats
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" // and database_name =~ "{LogicalDatabaseName}"
| where is_primary == 1
| where wait_category == "LOCK"
| summarize arg_max(max_query_wait_time_ms, *) by database_name, query_hash
| top 10 by max_query_wait_time_ms desc nulls last

MonWiQdsExecStats
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" // and database_name =~ "{LogicalDatabaseName}"
| where is_primary == 1
| where query_hash =~ "0x6919447CA6760071"
| extend cpu_time_ms=round(cpu_time/1024.0,0), elapsed_time_ms=round(elapsed_time/1024.0,0)

```

```
| project TIMESTAMP, database_name, query_hash, cpu_time_ms, elapsed_time_ms , rowcount, exec_type ,
statement_type
| top 100 by cpu_time_ms desc nulls last
```

#### MonWiQdsExecStats

```
//| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where TIMESTAMP > datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}"// and database_name =~ "{LogicalDatabaseName}"
| where is_primary == 1
| where query_hash =~ "0x6919447CA6760071"
| extend cpu_time_ms=round(cpu_time/1024.0,0), elapsed_time_ms=round(elapsed_time/1024.0,0)
| summarize exec_count=count(), avg_row_count=round(avg(rowcount),0),
avg_cpu_ms=round(avg(cpu_time_ms),0), max_cpu_ms=max(cpu_time_ms) by exec_type
| project TIMESTAMP, database_name, query_hash, cpu_time_ms, elapsed_time_ms , rowcount, exec_type ,
statement_type
| top 100 by cpu_time_ms desc nulls last
```

#### MonSQLSystemHealth

```
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and AppName =~ "{AppName}" and NodeName =~ "{NodeName}"
| where event=="errorlog_written"
| where message contains "KILL" or message contains "kill"
```

**How good have you found this content?**



-