# Max_DOP_Issues

Last updated by | Amie Coleman | Nov 23, 2022 at 7:39 AM PST

Contents

## Issue

Customer is experiencing High CPU Utilisation potentially caused by misconfigured MAXDOP.

### Background

MAXDOP (max degree of parallelism) controls intra-query parallelism in the database engine. In **general**, a higher MAXDOP value results in more parallel threads per query, and faster query execution.

In Azure SQL Database, the default MAXDOP setting (database-scoped configuration) for each new single database and elastic pool database is 8. Any database created prior to September 2020 may still have the older default of 0. The default MAXDOP change reduces the frequency and severity of incidents caused by excessive query parallelism and improves workload performance by reducing unnecessary resource utilisation. The recommendation is to avoid a MAXDOP setting of 0 even if it doesn't appear to currently cause problems. If the database MAXDOP setting is set to 0, consider updating the MAXDOP setting.

The customer can control the MAXDOP values at the database and query level:

- At the database level, using the MAXDOP database scoped configuration
- At the query level, using the MAXDOP query hint

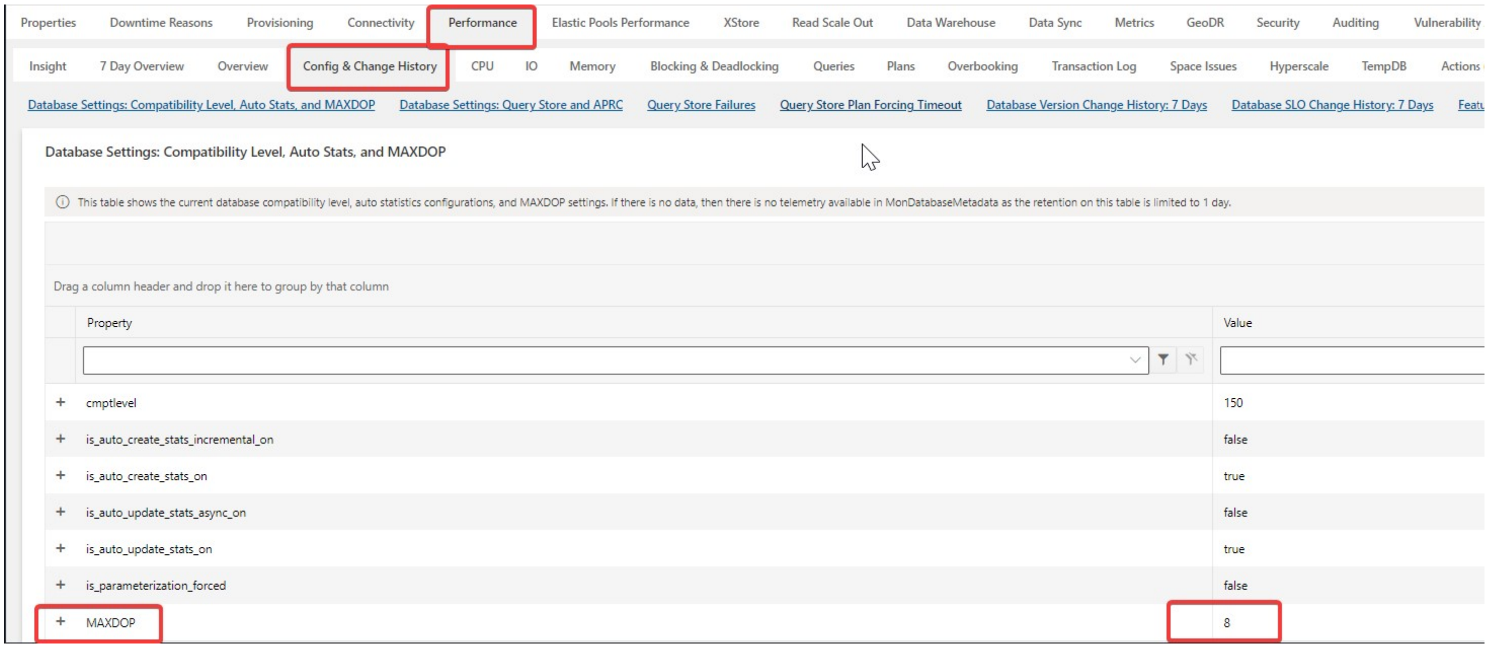
## Investigation/Analysis

When the customer is reporting high CPU, one of the causes can be excessive parallelism due to misconfigured MAXDOP. Query parallelism is not always an indication of an issue and is normal to see, however, if you're observing CXPACKET waits (above other waits) and high CPU utilisation then it is recommended that parallelism is investigated.

To start the investigation, check the database-scoped MAXDOP configuration and Query Level MAXDOP results for the top CPU consuming queries using these steps:

## Database-Scoped MAXDOP

Review the database-scoped MAXDOP value for the resource in ASC:
SQL Troubleshooter >> Performance >> Config & Change History



Database Settings: Compatibility Level, Auto Stats, and MAXDOP

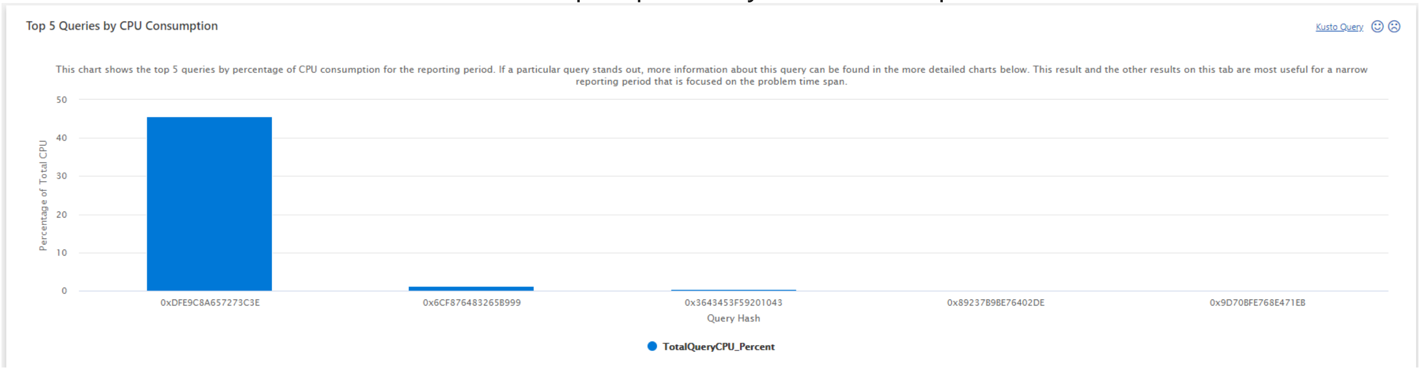
This table shows the current database compatibility level, auto statistics configurations, and MAXDOP settings. If there is no data, then there is no telemetry available in MonDatabaseMetadata as the retention on this table is limited to 1 day.

Drag a column header and drop it here to group by that column

| Property | Value |
|---|---|
| cmptlevel | 150 |
| is_auto_create_stats_incremental_on | false |
| is_auto_create_stats_on | true |
| is_auto_update_stats_async_on | false |
| is_auto_update_stats_on | true |
| is_parameterization_forced | false |
| MAXDOP | 8 |

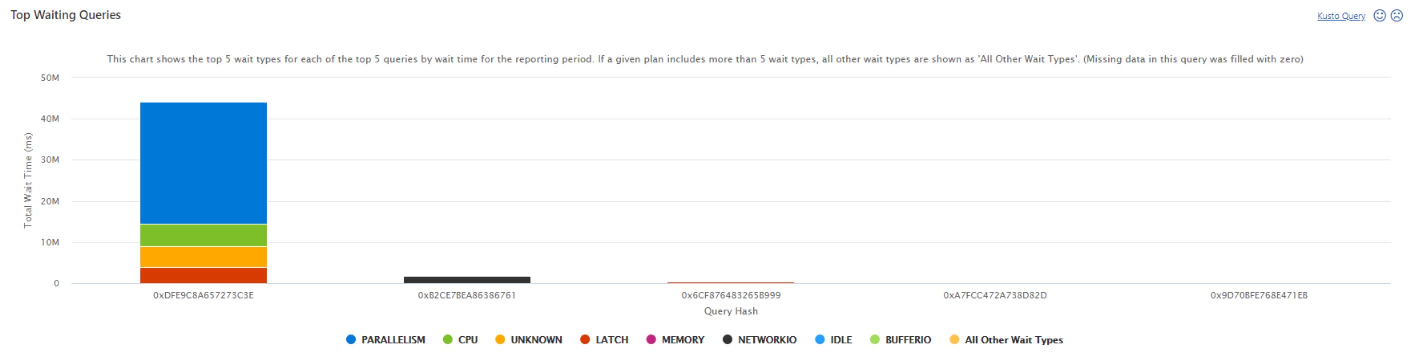
Review the Top CPU consuming queries and corresponding wait types to understand if CPU utilisation is being driven by parallelism:

1. SQL Troubleshooter >> Performance >> Top 5 queries by CPU consumption



This chart shows the top 5 queries by percentage of CPU consumption for the reporting period. If a particular query stands out, more information about this query can be found in the more detailed charts below. This result and the other results on this tab are most useful for a narrow reporting period that is focused on the problem time span.

2. Check the top waiting queries, if the top CPU consumer has excessive parallelism waits, the issue could be due to MAXDOP misconfiguration



## Query level MAXDOP

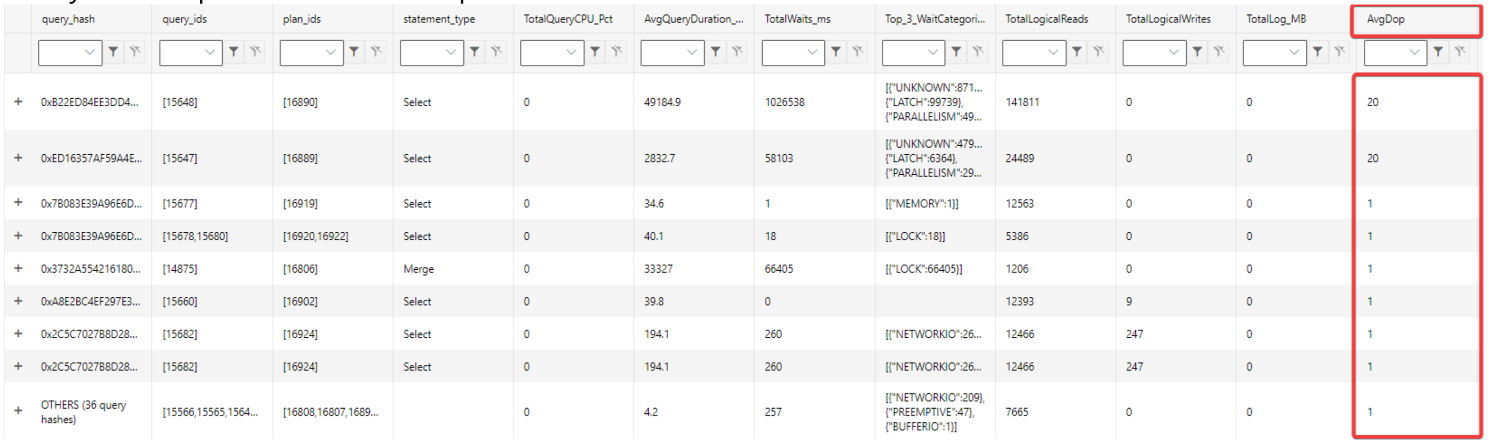
Customers can specfiy a MAXDOP value at the query level using a Query Hint. For example, the below hint specifies that the query should be run with a MAXDOP of 2 (OPTION (MAXDOP 2)):

```
SELECT *
FROM Sales.SalesOrderDetail
OPTION (MAXDOP 2);
GO
```

Note that if the customer has a MAXDOP query hint then this will override the MAXDOP option set as a database-scoped configuration. And, the value specified in the Query Hint can always exceed that of the database-scoped configuration.

To review individual query execution and see what MAXDOP value they are executing with, utilise the AvgDop column in the below ASC report:
Performance >> Queries >> Top 5 Queries for each CPU Time, Logical Reads and Logical Writes

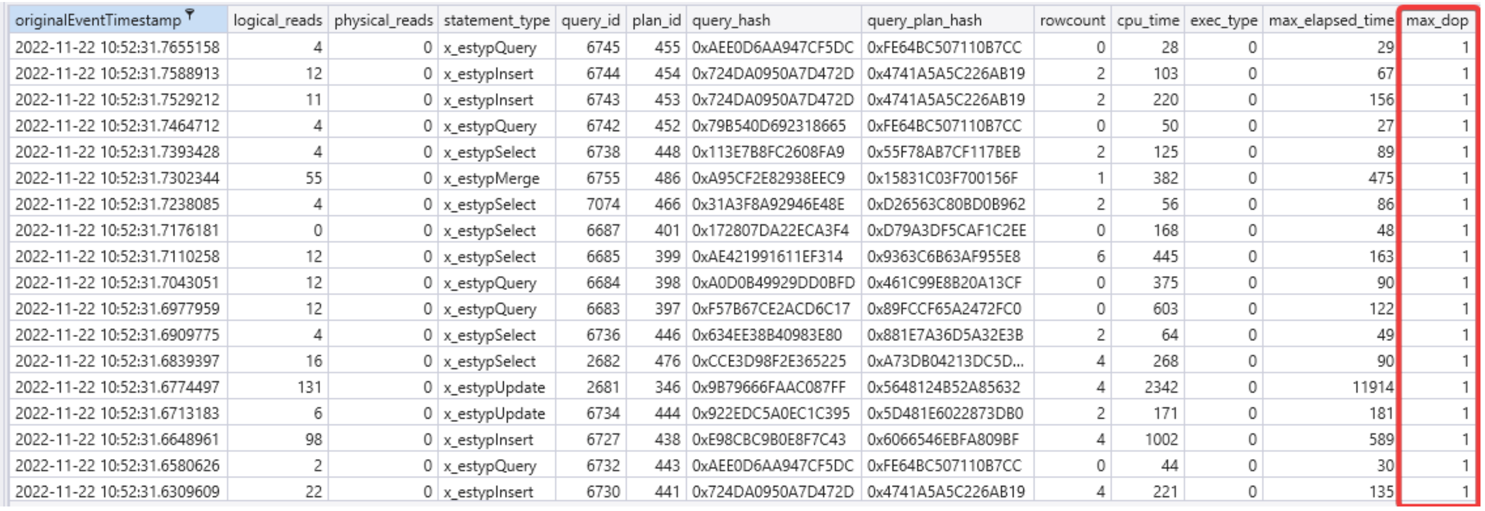
Query Hints explanation and examples



| | query_hash | query_ids | plan_ids | statement_type | TotalQueryCPU_Pct | AvgQueryDuration_... | TotalWaits_ms | Top_3_WaitCategori... | TotalLogicalReads | TotalLogicalWrites | TotalLog_MB | AvgDop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | 0xB22ED84EE3DD4... | [15648] | [16890] | Select | 0 | 49184.9 | 1026538 | [{"UNKNOWN":871... {"LATCH":99739}, {"PARALLELISM":49... | 141811 | 0 | 0 | 20 |
| + | 0xED16357AF59A4E... | [15647] | [16889] | Select | 0 | 2832.7 | 58103 | [{"UNKNOWN":479... {"LATCH":6364}, {"PARALLELISM":29... | 24489 | 0 | 0 | 20 |
| + | 0x7B083E39A96E6D... | [15677] | [16919] | Select | 0 | 34.6 | 1 | [{"MEMORY":1}] | 12563 | 0 | 0 | 1 |
| + | 0x7B083E39A96E6D... | [15678,15680] | [16920,16922] | Select | 0 | 40.1 | 18 | [{"LOCK":18}] | 5386 | 0 | 0 | 1 |
| + | 0x3732A554216180... | [14875] | [16806] | Merge | 0 | 33327 | 66405 | [{"LOCK":66405}] | 1206 | 0 | 0 | 1 |
| + | 0xA8E2BC4EF297E3... | [15660] | [16902] | Select | 0 | 39.8 | 0 | | 12393 | 9 | 0 | 1 |
| + | 0x2C5C7027B8D28... | [15682] | [16924] | Select | 0 | 194.1 | 260 | [{"NETWORKIO":26... | 12466 | 247 | 0 | 1 |
| + | 0x2C5C7027B8D28... | [15682] | [16924] | Select | 0 | 194.1 | 260 | [{"NETWORKIO":26... | 12466 | 247 | 0 | 1 |
| + | OTHERS (36 query hashes) | [15566,15565,1564... | [16808,16807,1689... | | 0 | 4.2 | 257 | [{"NETWORKIO":209}, {"PREEMPTIVE":47}, {"BUFFERIO":1}] | 7665 | 0 | 0 | 1 |

Alternatively, you can query in Kusto using MonWiQdsExecStats:

```
MonWiQdsExecStats
//| where TIMESTAMP >= datetime(2022-11-09 01:00:00Z)
//| where TIMESTAMP <= datetime(2022-11-10 08:00:00Z)
| where TIMESTAMP >= ago(5d)
| where LogicalServerName =~""
//| where database_name == ""
//| where query_hash =~ ""
//| where query_id ==
//| where plan_id ==
//| where statement_type contains ""
| project originalEventTimestamp, logical_reads, physical_reads, statement_type, query_id, plan_id, query_hash
| order by originalEventTimestamp desc
```



| originalEventTimestamp | logical_reads | physical_reads | statement_type | query_id | plan_id | query_hash | query_plan_hash | rowcount | cpu_time | exec_type | max_elapsed_time | max_dop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2022-11-22 10:52:31.7655158 | 4 | 0 | x_estypQuery | 6745 | 455 | 0xAEE0D6AA947CF5DC | 0xFE64BC507110B7CC | 0 | 28 | 0 | 29 | 1 |
| 2022-11-22 10:52:31.7588913 | 12 | 0 | x_estypInsert | 6744 | 454 | 0x724DA0950A7D472D | 0x4741A5A5C226AB19 | 2 | 103 | 0 | 67 | 1 |
| 2022-11-22 10:52:31.7529212 | 11 | 0 | x_estypInsert | 6743 | 453 | 0x724DA0950A7D472D | 0x4741A5A5C226AB19 | 2 | 220 | 0 | 156 | 1 |
| 2022-11-22 10:52:31.7464712 | 4 | 0 | x_estypQuery | 6742 | 452 | 0x79B540D692318665 | 0xFE64BC507110B7CC | 0 | 50 | 0 | 27 | 1 |
| 2022-11-22 10:52:31.7393428 | 4 | 0 | x_estypSelect | 6738 | 448 | 0x113E7B8FC2608FA9 | 0x55F78AB7CF117BEB | 2 | 125 | 0 | 89 | 1 |
| 2022-11-22 10:52:31.7302344 | 55 | 0 | x_estypMerge | 6755 | 486 | 0xA95CF2E82938EEC9 | 0x15831C03F700156F | 1 | 382 | 0 | 475 | 1 |
| 2022-11-22 10:52:31.7238085 | 4 | 0 | x_estypSelect | 7074 | 466 | 0x31A3F8A92946E48E | 0xD26563C80BD0B962 | 2 | 56 | 0 | 86 | 1 |
| 2022-11-22 10:52:31.7176181 | 0 | 0 | x_estypSelect | 6687 | 401 | 0x172807DA22ECA3F4 | 0xD79A3DF5CAF1C2EE | 0 | 168 | 0 | 48 | 1 |
| 2022-11-22 10:52:31.7110258 | 12 | 0 | x_estypSelect | 6685 | 399 | 0xAE421991611EF314 | 0x9363C6B63AF955E8 | 6 | 445 | 0 | 163 | 1 |
| 2022-11-22 10:52:31.7043051 | 12 | 0 | x_estypQuery | 6684 | 398 | 0xA0D0B49929DD0BFD | 0x461C99E8B20A13CF | 0 | 375 | 0 | 90 | 1 |
| 2022-11-22 10:52:31.6977959 | 12 | 0 | x_estypQuery | 6683 | 397 | 0xF57B67CE2ACD6C17 | 0x89FCCF65A2472FC0 | 0 | 603 | 0 | 122 | 1 |
| 2022-11-22 10:52:31.6909775 | 4 | 0 | x_estypSelect | 6736 | 446 | 0x634EE38B40983E80 | 0x881E7A36D5A32E3B | 2 | 64 | 0 | 49 | 1 |
| 2022-11-22 10:52:31.6839397 | 16 | 0 | x_estypSelect | 2682 | 476 | 0xCCE3D98F2E365225 | 0xA73DB04213DC5D... | 4 | 268 | 0 | 90 | 1 |
| 2022-11-22 10:52:31.6774497 | 131 | 0 | x_estypUpdate | 2681 | 346 | 0x9B79666FAAC087FF | 0x5648124B52A85632 | 4 | 2342 | 0 | 11914 | 1 |
| 2022-11-22 10:52:31.6713183 | 6 | 0 | x_estypUpdate | 6734 | 444 | 0x922EDC5A0EC1C395 | 0x5D481E6022873DB0 | 2 | 171 | 0 | 181 | 1 |
| 2022-11-22 10:52:31.6648961 | 98 | 0 | x_estypInsert | 6727 | 438 | 0xE98CBC9B0E8F7C43 | 0x6066546EBFA809BF | 4 | 1002 | 0 | 589 | 1 |
| 2022-11-22 10:52:31.6580626 | 2 | 0 | x_estypQuery | 6732 | 443 | 0xAEE0D6AA947CF5DC | 0xFE64BC507110B7CC | 0 | 44 | 0 | 30 | 1 |
| 2022-11-22 10:52:31.6309609 | 22 | 0 | x_estypInsert | 6730 | 441 | 0x724DA0950A7D472D | 0x4741A5A5C226AB19 | 4 | 221 | 0 | 135 | 1 |

## Mitigation

Based on our telemetry data and experience with Azure SQL Database, the default and recommended MAXDOP (database-scoped) is 8. Whilst this is the optimal value for a variety of customer workloads, there are workloads out there that will benefit from other MAXDOP values.

Customers need to experiment with different MAXDOP values to determine what is most optimal for their workload.

- If you find that MAXDOP is set to 0 (unlimited), consider updating the value to 8 in the first instance.
- If the customer has a value other than 0 already configured and there is evidence of excessive parallelism (CXPACKET waits) and high CPU with customer impact, consider a lower MAXDOP value. **Note** that the choice for MAXDOP is highly influenced by the nature of the customers workload and there is not a single MAXDOP value that is optimal for all workloads. As any changes to MAXDOP can impact query performance, thorough testing is advised prior to making the change.

### How to change MAXDOP

MAXDOP can be changed using the ALTER DATABASE SCOPED CONFIGURATION statement in the scope of the database. For example, the below statement changes MAXDOP to 8 for a database:

```
ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 8;
```

### Query Hints

- If you find that the customer is using Query Hints in their queries and those queries are the top CPU consumers/driving up the relative waits, the customer should consider changing the MAXDOP query hint value.

## Public Doc Reference

Configure MAXDOP in Azure SQL Database
Modifying MAXDOP Considerations
Changes to default MAXDOP article

## Root Cause Classification

Root Cause: Azure SQL v3\Performance\Specific Query Slow

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