

# Slow\_Query\_Different\_Replicas\_Secondary

Last updated by | Subbu Kandhaswamy | Nov 3, 2021 at 12:26 PM PDT

---

## Contents

- [Issue](#)
- [Investigation/Analysis](#)
  - [Configuration checks](#)
  - [Query plan capture:](#)
  - [Common issues](#)
- [Mitigation](#)
- [Root Cause Classification](#)

## Issue

A user may report issue that an identical query runs faster on one replica but slower on another. A typical example is that a query runs faster on primary but slower on secondary or vice versa. This TSG defines steps to collect information and basic analysis for this issue.

## Investigation/Analysis

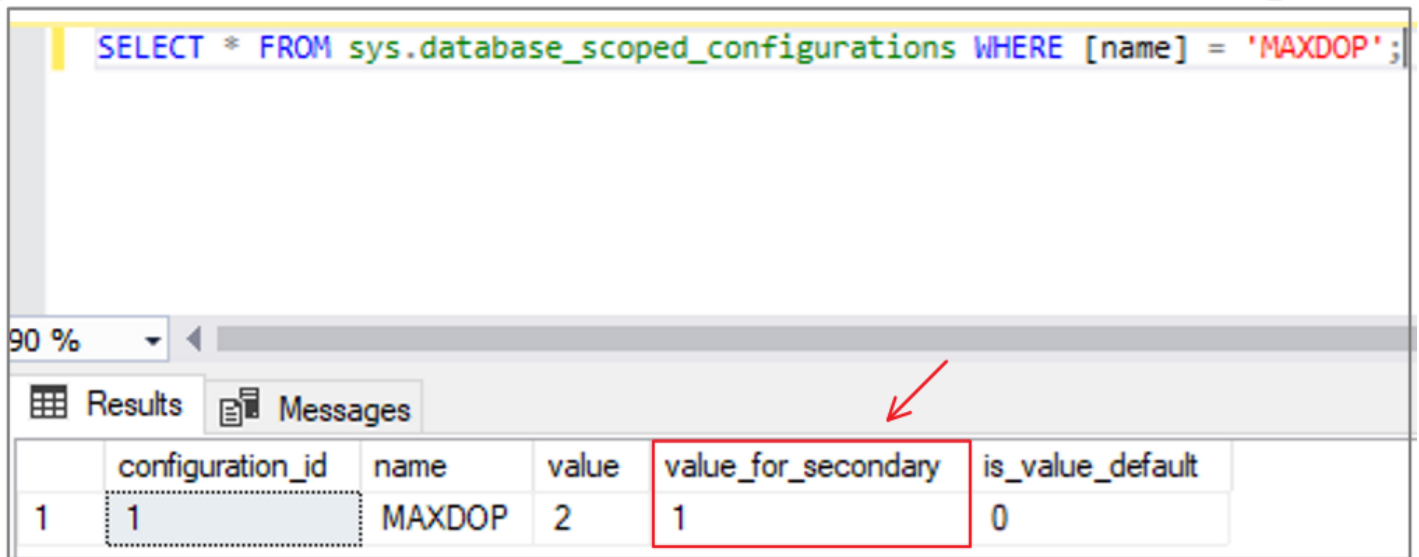
NOTE: When users say query never finishes, it may mean that query runs much longer. Wait for query to finish to get actual execution plan defined here, it is critical.

### Configuration checks

1. Verify MAXDOP are the same

```
SELECT * FROM sys.database_scoped_configurations WHERE [name] = 'MAXDOP'
```

- The "**value\_for\_secondary**" column represents the MaxDOP settings for Geo-Secondary. If this number is not the same as Primary MaxDOP then update the MaxDOP settings for the secondary.



	configuration_id	name	value	value_for_secondary	is_value_default
1	1	MAXDOP	2	1	0

- If customer has high end SLOs, have them configure MAXDOP to 8 for both primary and secondary.  
Example

- ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 8 ;
- ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET MAXDOP = 8;

## 2. For geo secondary, make sure primary and secondary SLO match.

- You can create ASC report for both the primary and the geo-secondary, and the SLO configuration is und
- Or you can use this kusto query to get the SLO information:

```

```
MonDmRealTimeResourceStats
| where AppName =~ "<>" and TIMESTAMP between (datetime()..datetime()) and NodeName=~""
| where LogicalServerName =~ '' and database_name =~ ''
| summarize StartTime=min(TIMESTAMP), EndTime=max(TIMESTAMP) by AppName, slo_name
| project StartTime, EndTime, AppName, slo_name
| order by StartTime asc
```
```

- \*To get the logical server name and database name for geo-secondary: open the XTS and use the view "Da

## Query plan capture:

- Capture query executions plans using the following. Actual execution plan is critical for analysis. Estimated plan from query store is useful but may not solve the problem.
  - scenario 1: If the slow query eventually finishes, Please follow [How to capture query execution plan and run time statistics](#). This is preferred over scenarios 2 and 3.
  - scenario 2: If query never finishes (taking hours or days), follow [Live Plan Statistics](#)
  - scenario 3: if user calls you in the middle of query execution (the query has started), use [How to capture partial execution plan for an in-flight](#)
- Use SSMS to compare the query plans to see if they are the same.
  - If the plans are different, engage QP experts for help if you cannot solve it yourself with your leads.
  - If the plans are the same and major difference is CPU, move to step 2.
  - how to know plan consumes CPU?
    - Statistics time output or query plan will tell you how much CPU is used. The XML text will have something like this: `<QueryTimeStats CpuTime="2498285" ElapsedTime="326209" />`

## Common issues

1. If issue is on Hyper-scale readable secondary, please ping Socrates Scrum [socscrum@microsoft.com](mailto:socscrum@microsoft.com), denzlr to see if there are any known issues before proceeding for an ICM.
2. Gen4 has physical cores while Gen5/6 has hyperV (double core counts). This may cause the difference in estimated available degree of parallelism. That is, the plan on Gen5/ Gen 6 machine has doubled core count, thus it may pick a plan with higher parallelism. To check the estimated available degree of parallelism, search for "EstimatedAvailableDegreeOfParallelism" in the XML plan. a. If the plan on Gen5/6 is better, upgrade the Gen4 to Gen5/6. If the plan on Gen4 is better, we have to decrease the MAXDOP in DB-scoped config for the Gen5/6, as the Gen4 is deprecated. A possible good value for the MAXDOP is the "EstimatedAvailableDegreeOfParallelism" in the XML plan on Gen4.

## Mitigation

At this point, it is assumed that both fast and slow plans are identical and main slowdown is caused by CPU. Please engage PG team: SQL Perf: Query Performance and attach all relevant findings to the incident.

## Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:

- Workload Performance/User-issue/error

## How good have you found this content?

