Article ID: 2023536 - Last Review: April 3, 2012 - Revision: 3.0

Recommendations and Guidelines for 'max degree of parallelism' configuration option

Summary

The SQL Server configuration option 'max degree of parallelism' controls the number of processors used for the execution of a query with a parallel plan. This option determines the computing and thread resources used for the query plan operators that can perform the work in parallel. Depending upon whether SQL Server is setup on a Symmetric MultiProcessor Computer [SMP], Non-Uniform Memory Access computer [NUMA] or HyperThreading enabled processors, you will need to configure the 'max degree of parallelism' option appropriately. Also various other configuration options like 'affinity mask', MAXDOP query hints and parallel index option, Soft NUMA and MAXDOP Resource Governor configuration option can affect or override the 'max degree of parallelism' configuration option.

For SQL Server 2008 R2, SQL Server 2008 and SQL Server 2005 servers, use the following guideline:
a. For servers that have eight or less processors, use the following configuration where N equals the number of processors: max degree of parallelism = 0 to N .
b. For servers that use more than eight processors, use the following configuration: max degree of parallelism = 8.
c. For servers that have NUMA configured, max degree of parallelism should not exceed the number of CPUs that are assigned to each NUMA node with the max value capped to 8. This will increase the likelyhood of all parallel threads of a query to be located within a NUMA Node and avoid costly remote node data look ups.
d. For servers that have hyper-threading enabled, the max degree of parallelism value should not exceed the number of physical processors.

Use these same guidelines when setting the MAXDOP option for Resource Governor workload groups.

The maximum value of 8 provided in the above guideline is applicable for typical SQL Server activity and the overhead for the exchange operators used in parallel query plans. You could vary this maximum value depending upon your specific application patterns and concurrent activity on the SQL Server. For example:
- If you have very small number of concurrently executing queries relative to the number of processors, then you can set maxdop to a higher value like 16.
- If you have very large number of concurrently executing queries relative to the number of processors, then you can set maxdop to a lower value such as 4.
Any value you propose to use should be thoroughly tested against the specific application activity or pattern of queries.

More Information

For SQL Server 2005 and SQL Server 2008 recommendations and guidelines for t