

# Parallelism and CXPACKET waits

CXPACKET waits occur when a parallel query is executed and one or more threads have to wait on one of the other threads to complete. SQL Server parallelizes a query by dividing each query operation into equal sized sets and by splitting up these sets into multiple threads targeting specific processors for parallel processing. Because not all the threads complete at exactly the same time, the server experiences CXPACKET waits.



Parallel queries require extra overhead to coordinate the parallel processing along with the initialization, synchronization, and termination of the parallel plan. **If you have a parallel query execution**, you will have CXPACKET waits.

## High CXPACKET waits

CXPACKET waits are of concern only if they are in excess. High CXPACKET waits indicate that some performance tuning is necessary on a specific query or the whole SQL Server.

The causes for experiencing high CXPACKET in your SQL Server can be any of the following:

- Out of data statistics causing SQL Server to incorrectly divide the query into equal sized sets.
- Fragmented indexes causing slower I/O speeds that impact one thread over the others.
- Client applications processing result sets inefficiently.
- Hyper-threading that causes SQL Server to process threads on hyper-threaded cores instead of only physical cores.
- CPU pressure.
- Memory pressure.
- Incorrect configuration of the Max degree of parallelism server option.
- Incorrect configuration of the Cost threshold for parallelism server option.

If you experience high CXPACKET waits in your OLTP server, it is generally due to one or more of the following:

- Missing indexes
- Out of data statistics
- Missing search predicate
- Function used on a column in a search predicate

## Keep CXPACKET waits under control

Use the following process to keep regular maintenance of your index and statistics and to control CXPACKET waits:

1. Enable Auto Create Statistics and Auto Update Statistics database properties options in all databases.
2. Rebuild all indexes.
3. Perform an UPDATE STATISTICS on all stats objects with the FULLSCAN.
4. Disable Hyper-threading.
5. Set the Max Degree of Parallelism server configuration option to a number smaller than the number of physical processor cores.
6. Set the Cost Threshold for Parallelism server configuration option not too low.

## Identify the cause for experiencing CXPACKET waits

You can review the query plans from the plan cache to identify queries that can be parallelized. Use the following links to find scripts for viewing your query plans:

- [Can you dig it? - Parallelism in the plan cache](#)
- [Find query plans that may utilize parallelism](#)

SQL Doctor provides complete SQL performance tuning. [Learn more](#) > >

<a href="#">IDERA Website</a>	<a href="#">Products</a>	<a href="#">Purchase</a>	<a href="#">Support</a>	<a href="#">Community</a>	<a href="#">About Us</a>	<a href="#">Resources</a>	<a href="#">Legal</a>
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------