<http://www.confio.com/English/Tips/Async_Network_IO.php>

**ASYNC Network IO**

The “async network io” (in SQL 2005/2008) and “networkio” (in SQL 2000) wait types can point to network related issues, but most often are caused by a client application that is not processing results from the SQL Server quickly enough. This will result in filling the network buffers so that SQL Server cannot send more data to the client. Therefore, the process executing the batch will need to wait for the ability to continue sending results to the client.

**Reducing Waits / Wait times:**

If there are significant wait times on “async network io’, review the client applications. Most often, client applications will process rows one at a time using fetches. This may cause the server process to wait on “async network io” when serving up many rows. If this is the issue, there is nothing you can do to improve the SQL Server process performance. Instead, the client application (or middle tier if a web application) may need to be modified to allow for more efficient retrieval of data. Review the following list for client application issues:

* Some applications, such as Microsoft Access, will ask for large result sets (typically identified by select statements with no where clause or full table scans), and then further filter the data on the client. If this is causing significant wait time, see if it’s possible to create a view for the client application to use instead. This will significantly reduce the amount of data being pushed to client since all of the filtering will done on SQL Server. Another fix could be to add a ‘where clause’ or further restrict the query so that less data is being sent to the client.
* Identify large result sets and verify with the application or developer team how the data is being consumed.  If the application is querying large result sets but using only a few rows, consider only querying the rows that are needed or use ‘TOP n’ to reduce the number of rows returned.
* If you are encountering high “async network io” wait times when performing data loads on the server, make sure the shared memory protocol is enabled for the SQL Server instance and the session is connected using net\_transport = ‘Shared memory’. You can determine the net\_transport for the connection by looking at the DMV – sys.dm\_exec\_connections.

If the above tuning tips are reviewed and applied, but the server is still is encountering high “async network io” times, then ensure there aren’t any network related issues:

* Review counters such as ‘Batch Requests per second’. Any values over 3000 for a 100MB network card are getting excessive. ‘Current Bandwidth’ on the Network Interface with values over 0.6 are getting excessive.
* Check network adapter bandwidth - 1 Gigabit is better than 100 megabits and 100 megabits is better than 10 megabits.
* Look at your NIC configuration on the server to make sure there are no issues with the physical card. Also, check if autodetect is picking the fastest speed.
* Validate all of the network components between the client application and the SQL Server instance (e.g. switches / routers).

**DMV query:**

As a side note, reviewing the resources that SQL Server sessions are waiting on (or using up) can be quite helpful when tuning SQL Server for performance. There are several DMVs (Dynamic Management Views) that report the wait types (or resources) each the session is waiting on while running specific SQLs. To view the sql text, the wait type and time waited for a session, run the query listed below:

|  |  |  |
| --- | --- | --- |
| http://www.confio.com/English/Images/Misc/ulcornertan.gif |  | http://www.confio.com/English/Images/Misc/urcornertan.gif |
|  | SELECT st.text AS [SQL Text], w.session\_id,  w.wait\_duration\_ms, w.wait\_type, w.resource\_address,  w.blocking\_session\_id,  w.resource\_description FROM sys.dm\_os\_waiting\_tasks AS w INNER JOIN sys.dm\_exec\_connections AS c ON w.session\_id = c.session\_id  CROSS APPLY (SELECT \* FROM sys.dm\_exec\_sql\_text(c.most\_recent\_sql\_handle)) AS st WHERE w.session\_id > 50 AND w.wait\_duration\_ms > 0 |  |
| http://www.confio.com/English/Images/Misc/llcornertan.gif |  | http://www.confio.com/English/Images/Misc/lrcornertan.gif |

The ‘wait\_duration\_ms’ value will increase for the session\_id as the session is waiting for the ‘wait\_type’ (for a specific SQL). The statistics in the DMV query are cumulative since last Instance restart or since the statistics were last cleared. In order to get a clear picture of all the SQLs and what resources (or wait\_types) they are waiting on perform deltas of this query and record them over a period of time. This will give you a better understanding of what sqls to work on first and how to tune them. To reset the wait statistics in the DMVs, enter the following command in SSMS: DBCC SQLPERF('sys.dm\_os\_wait\_stats', CLEAR).

**Conclusion**

When a session waits on the "async network io” event, it may be encountering network issues. More likely, however, it may be an issue with the client application not processing the data quickly enough. If the wait times for “async network io” are high, review the client application to see if large results sets are being sent to the client. If they are, work with the developers to understand if all the data is needed and reduce the size of result set if possible. Ensure that any data filtering is performed in SQL Server instead of the client by utilizing views or more specific where clauses. Use the ‘TOP n’ clause so that only the rows that the client needs are returned to the client. Investigate network issues, if client application tuning tips do not apply.