Common [**Wait Types**](https://blogs.msdn.microsoft.com/psssql/2009/11/02/the-sql-server-wait-type-repository/)(© 2017 Microsoft, available at [https://msdn.microsoft.com](https://msdn.microsoft.com/), obtained on February 24, 2017):

* [ASYNC\_NETWORK\_IO](http://logicalread.solarwinds.com/sql-server-async-network-io-wait-tl01/)—This is usually a sign of network latency between the client and the server.
* [CXPACKET](http://logicalread.solarwinds.com/sql-server-cxpacket-wait-type-tl01/)—Indicates that SQL is waiting on a Parallel process to complete. This can be a sign of resource issues on the SQL server (CPU, MEM, DiskIO) or the query itself is poorly written.
* [PAGEIOLATCH\_EX](http://logicalread.solarwinds.com/sql-server-pageiolatch_ex-wait-type/)—Buffer latches including the PAGEIOLATCH\_EX wait type are used to synchronize access to BUF structures and associated pages in the SQL Server database. This can be a sign of resource issues on the SQL server or a poorly written query. When seen in conjunction with the CXPACKET wait Index Fragmentation is often the cause.
* [WRITELOG](http://logicalread.solarwinds.com/sql-server-writelog-wait-dr01/)—When a SQL Server session waits on the WRITELOG wait type, it is waiting to write the contents of the log cache to disk where the transaction log is stored. This is almost always a sing of poor disk performance.
* [LCK\_(X)](http://blog.sqlauthority.com/2011/02/15/sql-server-lck_m_xxx-wait-type-day-15-of-28/) (©  2006 – 2017  [SQLAuthority.com](http://sqlauthority.com/), available at [https://blog.sqlauthority.com](https://blog.sqlauthority.com/), obtained on February 24th, 2017.)—This Occurs when a resource is in use by another query, usually the result of a UPDATE, INSERT, or DELETE statement. LCK waits are usually caused by resource contention but may also be the results of process blocking from any of the above wait types.

Query that returns the Latency for a given database :

SELECT

[ReadLatency] =

CASE WHEN [num\_of\_reads] = 0

THEN 0 ELSE ([io\_stall\_read\_ms] / [num\_of\_reads]) END,

[WriteLatency] =

CASE WHEN [num\_of\_writes] = 0

THEN 0 ELSE ([io\_stall\_write\_ms] / [num\_of\_writes]) END,

[Latency] =

CASE WHEN ([num\_of\_reads] = 0 AND [num\_of\_writes] = 0)

THEN 0 ELSE ([io\_stall] / ([num\_of\_reads] + [num\_of\_writes])) END,

[AvgBPerRead] =

CASE WHEN [num\_of\_reads] = 0

THEN 0 ELSE ([num\_of\_bytes\_read] / [num\_of\_reads]) END,

[AvgBPerWrite] =

CASE WHEN [num\_of\_writes] = 0

THEN 0 ELSE ([num\_of\_bytes\_written] / [num\_of\_writes]) END,

[AvgBPerTransfer] =

CASE WHEN ([num\_of\_reads] = 0 AND [num\_of\_writes] = 0)

THEN 0 ELSE

(([num\_of\_bytes\_read] + [num\_of\_bytes\_written]) /

([num\_of\_reads] + [num\_of\_writes])) END,

LEFT ([mf].[physical\_name], 2) AS [Drive],

DB\_NAME ([vfs].[database\_id]) AS [DB],

[mf].[physical\_name]

FROM

sys.dm\_io\_virtual\_file\_stats (NULL,NULL) AS [vfs]

JOIN sys.master\_files AS [mf]

ON [vfs].[database\_id] = [mf].[database\_id]

AND [vfs].[file\_id] = [mf].[file\_id]

-- WHERE [vfs].[file\_id] = 2 -- log files

-- ORDER BY [Latency] DESC

-- ORDER BY [ReadLatency] DESC

Where

DB\_NAME ([vfs].[database\_id]) ='AdventureWorksDW2012'

ORDER BY [WriteLatency] DESC;

GO

From the above you can check and make sense of the below points .

As a rule of thumb, here are the latencies you should expect if you are not overloaded on the I/O system

* Spinning media: Around 5-10ms
* SSD: <1ms

**ASYNC\_NETWORK\_IO :**

This type of latency happens between Client and Server. For example, when the client is requesting huge amount of data and the server is sending it in packets, the amount of data that the client can process causes this type of wait.

If the client side CPU usage is very high then its not able to consume all the data from the current packet and hence its not able to request the next packet from the server, which will cause the wait. Also, when the client is processing row by row then also this wait can happen.

The status usually shows as “**Suspended**”