Sys.dm_io_file_stats (Transact-SQL)

This dynamic management view (DMV) returns I/O statistics for data and log files. It can get all the information of any file in any database.

sys.dm_io_virtual_stats(
{ database_id | NULL} , { file_id | NULL}
)

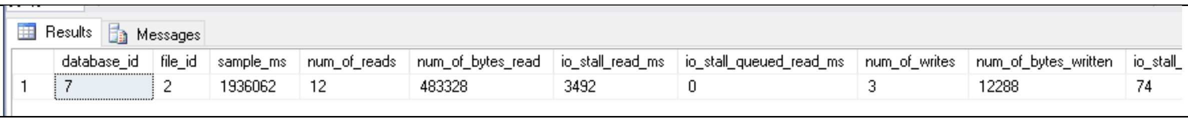
- database_id | NULL - ID of the database (database_id is int, with no default). Valid inputs:ID number of a database or NULL. When NULL - all databases in the instance of SQL Server are returned. The built-in function DB_ID can be specified. When using DB_ID without specifying a database name, the compatibility level of the current database must be 90.
- file_id | NULL - ID of the file (file_id is int, with no default). Valid inputs: ID number of a file or NULL. When NULL is specified, all files on the database are returned. The built-in function FILE_IDEX can be specified, and refers to a file in the current database.

Table Returned

Column name	Data type	Description
Database_name	**Sysname**	Database name.
database_id	Smallint	ID of database.
file_id	Smallint	ID of file.
sample_ms	Int	Number of milliseconds since the computer was started. This column can be used to compare different outputs from this function.
num_of_reads	Bigint	Number of reads issued on the file.
num_of_bytes_read	Bigint	Total number of bytes read on this file.
io_stall_read_ms	Bigint	Total time, in milliseconds, that the users waited for reads issued on the file.
num_of_writes	Bigint	Number of writes made on this file.
num_of_bytes_written	Bigint	Total number of bytes written to the file.
io_stall_write_ms	Bigint	Total time, in milliseconds, that users waited for writes to be completed on the file.
io_stall	Bigint	Total time, in milliseconds, that users waited for I/O to be completed on the file.
size_on_disk_bytes	Bigint	Number of bytes used on the disk for this file. For sparse files, this number is the actual number of bytes on the disk that are used for database snapshots.
file_handle	Varbinary	Windows file handle for this file.
io_stall_queued_write_ms	Bigint	Total IO latency introduced by IO resource governance for writes. Is not nullable.
io_stall_queued_read_ms	Bigint	Total IO latency introduced by IO resource governance for reads. Is not nullable.

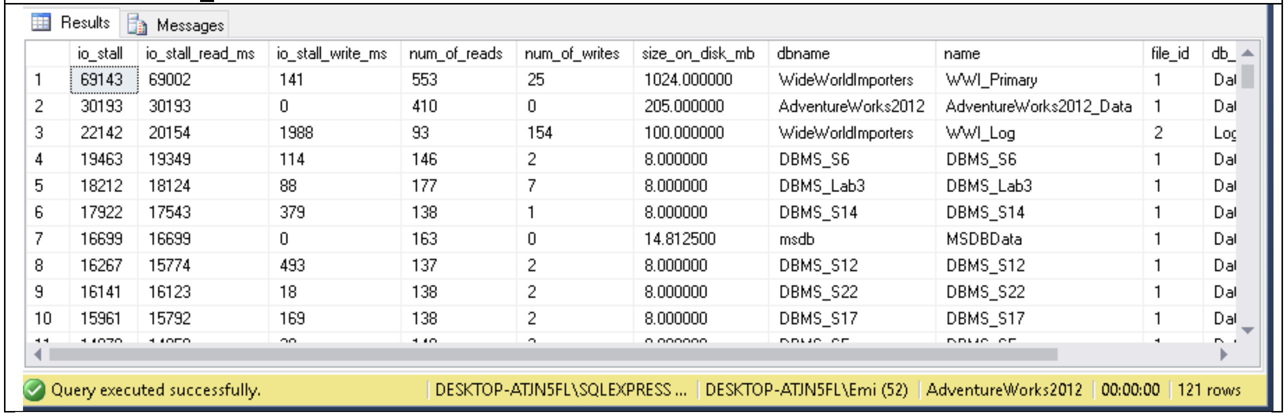
Requires VIEW SERVER STATE permission.

```
-- returns statistics for the log file in the AdventureWorks2012 database
SELECT * FROM sys.dm_io_virtual_file_stats(DB_ID(N'AdventureWorks2012'), 2);
GO
```



```
   database_id  file_id  sample_ms  num_of_reads  num_of_bytes_read  io_stall_read_ms  io_stall_queued_read_ms  num_of_writes  num_of_bytes_written  io_stall_
1  7            2        1936062    12            483328             3492              0                        3              12288                 74
```

```sql
-- This analysis can help make decisions around table partitioning and potentially file and
index placement. Of course, this will all depend on the customer's SAN and other constraints.
SELECT a.io_stall, a.io_stall_read_ms, a.io_stall_write_ms, a.num_of_reads,
a.num_of_writes,
--a.sample_ms, a.num_of_bytes_read, a.num_of_bytes_written, a.io_stall_write_ms,
( ( a.size_on_disk_bytes / 1024 ) / 1024.0 ) AS size_on_disk_mb,
db_name(a.database_id) AS dbname,
b.name, a.file_id,
db_file_type = CASE
                   WHEN a.file_id = 2 THEN 'Log'
                   ELSE 'Data'
                   END,
UPPER(SUBSTRING(b.physical_name, 1, 2)) AS disk_location
FROM sys.dm_io_virtual_file_stats (NULL, NULL) a
JOIN sys.master_files b ON a.file_id = b.file_id
AND a.database_id = b.database_id
ORDER BY a.io_stall DESC
```


Results | Messages

	io_stall	io_stall_read_ms	io_stall_write_ms	num_of_reads	num_of_writes	size_on_disk_mb	dbname	name	file_id	db_
1	69143	69002	141	553	25	1024.000000	WideWorldImporters	WWI_Primary	1	Dat
2	30193	30193	0	410	0	205.000000	AdventureWorks2012	AdventureWorks2012_Data	1	Dat
3	22142	20154	1988	93	154	100.000000	WideWorldImporters	WWI_Log	2	Log
4	19463	19349	114	146	2	8.000000	DBMS_S6	DBMS_S6	1	Dat
5	18212	18124	88	177	7	8.000000	DBMS_Lab3	DBMS_Lab3	1	Dat
6	17922	17543	379	138	1	8.000000	DBMS_S14	DBMS_S14	1	Dat
7	16699	16699	0	163	0	14.812500	msdb	MSDBData	1	Dat
8	16267	15774	493	137	2	8.000000	DBMS_S12	DBMS_S12	1	Dat
9	16141	16123	18	138	2	8.000000	DBMS_S22	DBMS_S22	1	Dat
10	15961	15792	169	138	2	8.000000	DBMS_S17	DBMS_S17	1	Dat

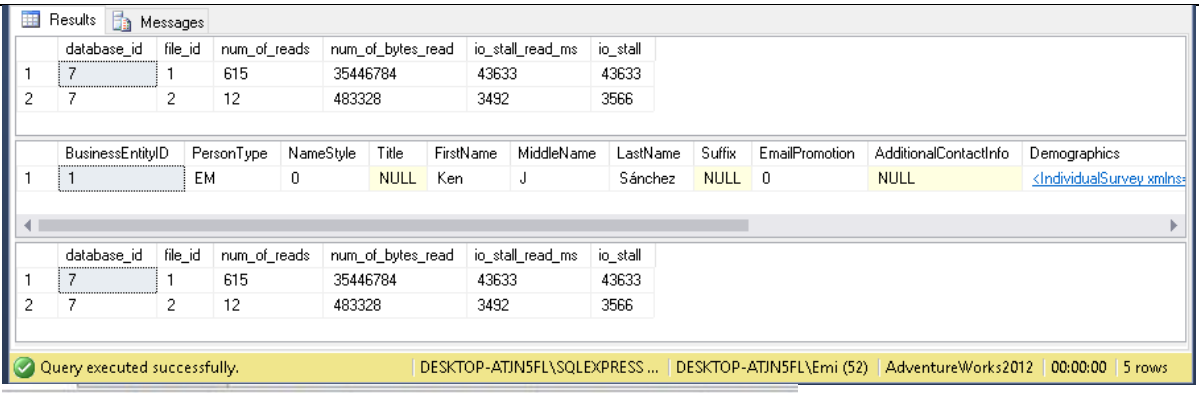
Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | AdventureWorks2012 | 00:00:00 | 121 rows

SQL Server performance depends on the I/O subsystem. Unless the database fits into physical memory, SQL Server brings database pages in and out of the buffer pool. This generates substantial I/O traffic. The log records need to be flushed to the disk before a transaction can be declared committed. SQL Server uses TempDB for various purposes (i.e. store intermediate results, to sort, to keep row versions). So, a good I/O subsystem is critical to the performance of SQL Server.

Access to log files is sequential except when a transaction needs to be rolled back while access to data files, including TempDB, is randomly accessed. So, one should have log files on a separate physical disk than data files for better performance. Once an I/O bottleneck is identified, one may need to reconfigure the I/O subsystem.

The information from this DMV is not of reads/writes, not of bytes read/write, wait times for reads/writes/both to complete.

```sql
-- collect the DMV data before and after running a select on a big table and one will see the
stats increase for read related counter.
SELECT database_id, file_id, num_of_reads, um_of_bytes_read, io_stall_read_ms, io_stall
FROM sys.dm_io_virtual_file_stats(DB_ID(), NULL)
GO
SELECT top 1 * FROM Person.Person
GO
SELECT database_id, file_id, num_of_reads, num_of_bytes_read, io_stall_read_ms, io_stall
FROM sys.dm_io_virtual_file_stats(DB_ID(), NULL)
```

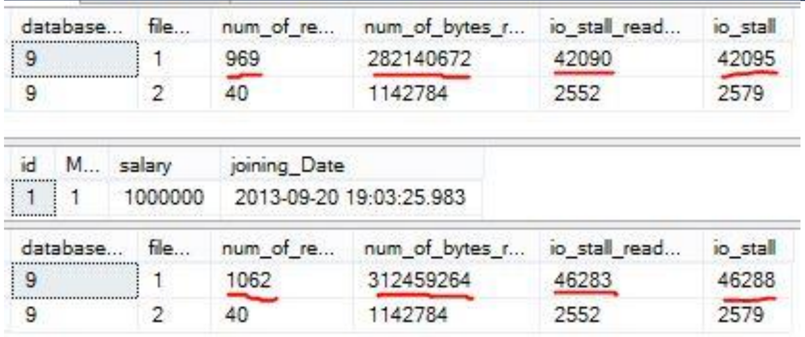

Results | Messages

	database_id	file_id	num_of_reads	num_of_bytes_read	io_stall_read_ms	io_stall
1	7	1	615	35446784	43633	43633
2	7	2	12	483328	3492	3566

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics
1	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	NULL	<IndividualSurvey xmlns=

	database_id	file_id	num_of_reads	num_of_bytes_read	io_stall_read_ms	io_stall
1	7	1	615	35446784	43633	43633
2	7	2	12	483328	3492	3566

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | AdventureWorks2012 | 00:00:00 | 5 rows



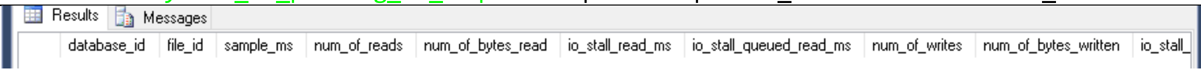
database...	file...	num_of_re...	num_of_bytes_r...	io_stall_read...	io_stall
9	1	969	282140672	42090	42095
9	2	40	1142784	2552	2579

id	M...	salary	joining_Date
1	1	1000000	2013-09-20 19:03:25.983

database...	file...	num_of_re...	num_of_bytes_r...	io_stall_read...	io_stall
9	1	1062	312459264	46283	46288
9	2	40	1142784	2552	2579

```
-- the select statement on person table has done reads. If are seeing IO issues live on
environment, for get the sense of which file is hitting more IO, one can use the next query to
run with a gap of 1 minute and get the difference to troubleshoot IO related issues.
```

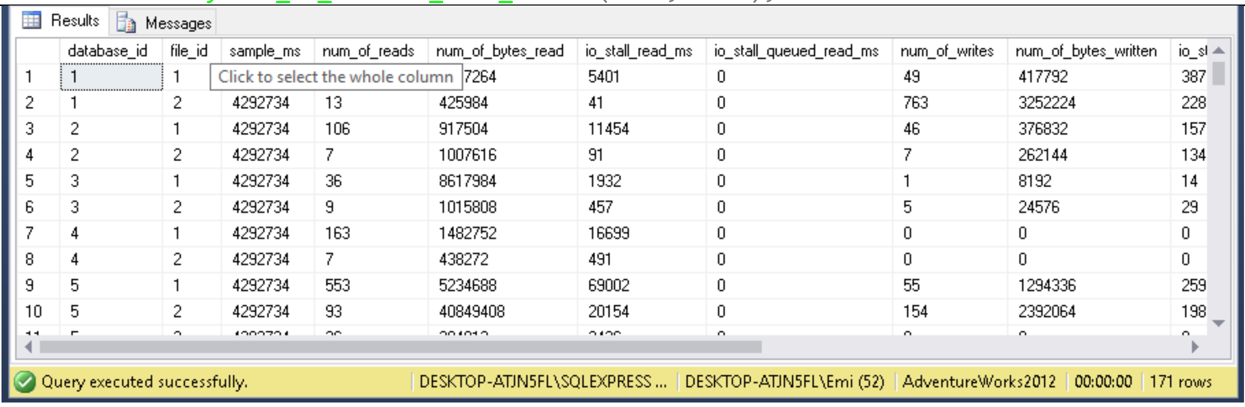
```
SELECT * FROM sys.dm_io_virtual_file_stats(DB_ID(), NULL) ivfs
INNER JOIN sys.dm_pending_io_requests ipir ON ipir.io_handle = ivfs.file_handle
```


Results | Messages

	database_id	file_id	sample_ms	num_of_reads	num_of_bytes_read	io_stall_read_ms	io_stall_queued_read_ms	num_of_writes	num_of_bytes_written	io_stall_

```
-- shows how many I/Os have occurred, with latencies for all files.
```

```
-- default output (with some column names changed slightly to make things fit nicely)
SELECT * FROM sys.dm_io_virtual_file_stats (NULL, NULL); GO
```


Results | Messages

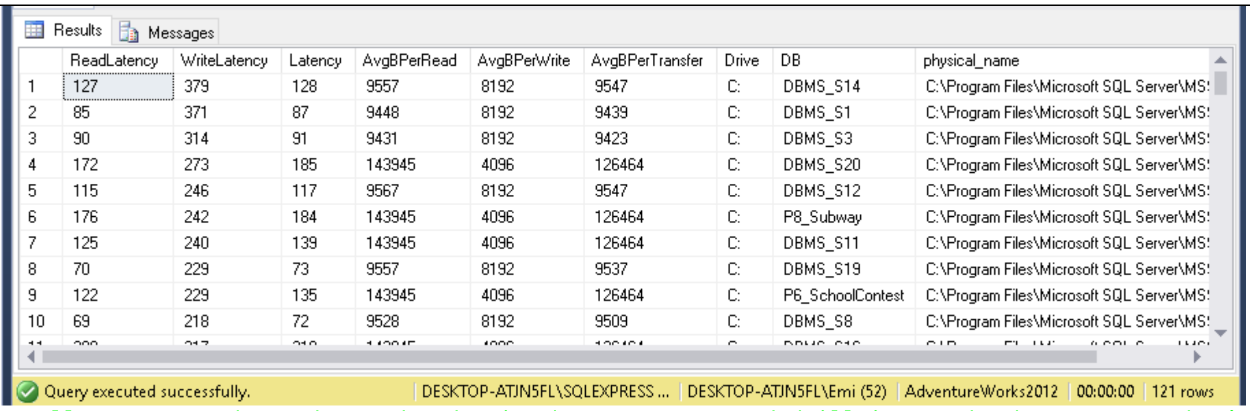
	database_id	file_id	sample_ms	num_of_reads	num_of_bytes_read	io_stall_read_ms	io_stall_queued_read_ms	num_of_writes	num_of_bytes_written	io_st
1	1	1	Click to select the whole column		7264	5401	0	49	417792	387
2	1	2	4292734	13	425984	41	0	763	3252224	228
3	2	1	4292734	106	917504	11454	0	46	376832	157
4	2	2	4292734	7	1007616	91	0	7	262144	134
5	3	1	4292734	36	8617984	1932	0	1	8192	14
6	3	2	4292734	9	1015808	457	0	5	24576	29
7	4	1	4292734	163	1482752	16699	0	0	0	0
8	4	2	4292734	7	438272	491	0	0	0	0
9	5	1	4292734	553	5234688	69002	0	55	1294336	259
10	5	2	4292734	93	40849408	20154	0	154	2392064	198

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | AdventureWorks2012 | 00:00:00 | 171 rows

```
-- This isn't that useful because there is no database IDs and file paths memorized, and it
gives aggregate latencies (io_stall_read_msand io_stall_write_ms).
```

Viewing Aggregate Information - part of script used for doing a server health check for a client, that allows one to filter on read or write latencies and it joins with sys.master_files to get database names and file paths.

```
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
ORDER BY [WriteLatency] DESC;
GO
```



Results | Messages

	ReadLatency	WriteLatency	Latency	AvgBPerRead	AvgBPerWrite	AvgBPerTransfer	Drive	DB	physical_name
1	127	379	128	9557	8192	9547	C:	DBMS_S14	C:\Program Files\Microsoft SQL Server\MS
2	85	371	87	9448	8192	9439	C:	DBMS_S1	C:\Program Files\Microsoft SQL Server\MS
3	90	314	91	9431	8192	9423	C:	DBMS_S3	C:\Program Files\Microsoft SQL Server\MS
4	172	273	185	143945	4096	126464	C:	DBMS_S20	C:\Program Files\Microsoft SQL Server\MS
5	115	246	117	9567	8192	9547	C:	DBMS_S12	C:\Program Files\Microsoft SQL Server\MS
6	176	242	184	143945	4096	126464	C:	P8_Subway	C:\Program Files\Microsoft SQL Server\MS
7	125	240	139	143945	4096	126464	C:	DBMS_S11	C:\Program Files\Microsoft SQL Server\MS
8	70	229	73	9557	8192	9537	C:	DBMS_S19	C:\Program Files\Microsoft SQL Server\MS
9	122	229	135	143945	4096	126464	C:	P6_SchoolContest	C:\Program Files\Microsoft SQL Server\MS
10	69	218	72	9528	8192	9509	C:	DBMS_S8	C:\Program Files\Microsoft SQL Server\MS

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | AdventureWorks2012 | 00:00:00 | 121 rows

```
-- Allows to see where the read and write hot spots are and drill into a database to see what’s going on.
```

References:
https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2012/ms190326(v=sql.110)
https://blogs.msdn.microsoft.com/dpless/2010/12/01/leveraging-sys-dm_io_virtual_file_stats/
http://www.sqlservergeeks.com/sys-dm_io_virtual_file_stats/
https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-io-virtual-file-stats-transact-sql?view=sql-server-2017
https://www.sqlskills.com/blogs/paul/how-to-examine-io-subsystem-latencies-from-within-sql-server/