

Setting up for the lab

1. Restart the SQL Server service (clears stats)
2. Restore your StackOverflow database
3. Copy & run the setup script:
BrentOzar.com/go/serverlab5
4. Start SQLQueryStress:
 1. File Explorer, D:\Labs, run SQLQueryStress.exe
 2. Click File, Open, D:\Labs\ServerLab5.json
 3. Click Go



3.3 p1



BRENT OZAR
UNLIMITED®

Listening & Logging Waits

WRITELOG, HADR_SYNC_COMMIT, ASYNC_NETWORK_IO

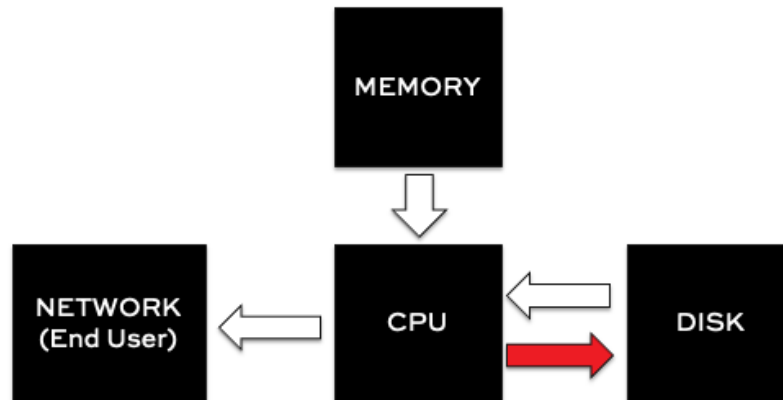
3.3 p2

WRITELOG



3.3 p3

WRITELOG: self-explanatory.




3.3 p4

Two wait stat numbers matter.

Wait Time: total amount of time you've waited

Avg ms Per Wait: when we do wait on something, how long do we have to wait?

Shown off to the right in sp_BlitzFirst:



Seconds Sample	wait_type	Wait Time (Seconds)	Per Core Per Second	Signal Wait Time (Seconds)	Percent Signal Waits	Number of Waits	Avg ms Per Wait
16	SOS_SCHEDULER_YIELD	3555.8	55.6	3555.8	100.0	15714	226.3
16	CXPACKET	1027.7	16.1	1.5	0.1	985	1043.4
16	THREADPOOL	3.1	0.0	0.0	0.0	57	54.9
16	LATCH_EX	1.0	0.0	0.3	30.0	204	4.8
16	LATCH_SH	0.0	0.0	0.0	0.0	7	4.7



3.3 p5

These two won't match.

WRITELOG average wait time, milliseconds:

when your query is waiting on the log file,
this is how long it's waiting.

Drive & file response time:

can be higher or lower, because you're not always
waiting for these files.

Your storage team only cares about the latter.



3.3 p6

Perfmon counters for more info

Performance Monitor counter:

Physical Disk: Avg Sec/Write (aka write latency)

Reported in whole seconds – use 3 decimal places for MS.

Microsoft says >3 milliseconds log writes are slow. Me: 20.

Good for both physical and virtual servers

Related counters: Physical Disk: Reads/sec, Writes/sec
(shows how much we're asking storage to work)

The more we ask storage to work, the slower it'll get.



3.3 p7

Solving WRITELOG is easier.

The transaction log is typically relatively small.
(Unless you're storing files in the database. Don't do that.)

Delayed Durability: new option in 2014 to consider transactions committed before they hit the log file.

Just one database involved? Use a dedicated pair of mirrored drives, ideally solid state.

Multiple databases involved? May need to stripe across many drives in a RAID 10, ideally solid state.



3.3 p8

HADR_SYNC _COMMIT



3.3 p9

Kinda like WRITELOG.

Only seen in Always On Availability Groups in synchronous commit mode.

Only affects data modification (not selects).

Wanna go faster?

- Switch to async (ha ha ho ho)
- Check waits on the sync secondaries (may be disk-bottlenecked)
- Check network latency between replicas
- Separate non-critical data (staging tables, scratch space) into separate, non-sync AG databases



3.3 p10

In-depth info from Microsoft

In case none of the following seem to apply to you:

https://blogs.msdn.microsoft.com/sql_server_team/troubleshooting-high-hadr_sync_commit-wait-type-with-always-on-availability-groups/



3.3 p11

ASYNC_NETWORK_IO



3.3 p12

Our routine

1. Start a workload
2. Check vital stats with `sp_BlitzFirst`
3. List possible root causes
4. List mitigation options
5. Apply one of them
6. Check vital stats again



3.3 p13

Preparation

Attach the StackOverflow database

Set your max memory up high, parallelism to defaults:

```
EXEC sys.sp_configure N'max server memory (MB)', N'55000'  
EXEC sys.sp_configure N'cost threshold for parallelism', N'5'  
EXEC sys.sp_configure N'max degree of parallelism', N'0'  
GO  
RECONFIGURE
```

Set up a new window for sp_BlitzFirst:

```
sp_BlitzFirst @ExpertMode = 1, @Seconds = 30
```



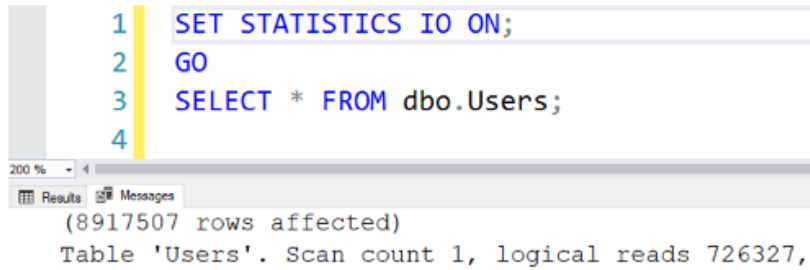
3.3 p14

Run this (ridiculous) query

```
1 SET STATISTICS IO ON;  
2 GO  
3 SELECT * FROM dbo.Users;  
4
```

200 %						
<div>ResultsMessages</div>						
	Id	AboutMe	Age	CreationDate	DisplayName	DownVote
1	-1	<p>Hi, I'm not really a person.</p> <p>I'm a backgrou...	NULL	2008-07-31 00:00:00.000	Community	956712
2	1	<p><a href="http://www.codinghorror.com/blog/archi...	NULL	2008-07-31 14:22:31.287	Jeff Atwood	1309
3	2	<p>Developer on the Stack Overflow team. Find me ...	NULL	2008-07-31 14:22:31.287	Geoff Dalgas	88
4	3	<p><a href="http://blog.stackoverflow.com/2009/01/...	NULL	2008-07-31 14:22:31.287	Jarrod Dixon	100
5	4	<p>I am:</p> the co-founder and CEO of <a ...	NULL	2008-07-31 14:22:31.317	Joel Spolsky	96
6	5	<p>Technical Evangelist at Microsoft, specializing in A...	NULL	2008-07-31 14:22:31.317	Jon Galloway	24

The query reads a lot of data



The screenshot shows a SQL Server query window with the following text:

```
1 SET STATISTICS IO ON;  
2 GO  
3 SELECT * FROM dbo.Users;  
4
```

Below the query window, the 'Results' tab is selected, displaying the following output:

```
(8917507 rows affected)  
Table 'Users'. Scan count 1, logical reads 726327,
```

But reading data isn't the bottleneck:
this table fits in RAM, and is completely cached.



3.3 p16

Top wait: ASYNC_NETWORK_IO.

```

1 SET STATISTICS IO ON;
2 GO
3 SELECT * FROM dbo.Users;
4

```

```

1 sp_BlitzFirst @ExpertMode = 1, @Seconds = 30

```

id	url	Age	CreationDate
1	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 00:00:00
2	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
3	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
4	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
5	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
6	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
7	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
8	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
9	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
10	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
11	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
12	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31
13	https://www.codinghorror.com/blog/2009/01/...	MULL	2009-07-31 14:22:31

run_date	elapsed_time	session_id	database_name	query_text	query_plan
2020-05-20 04:19:43.390	0:00:00.50:070	56	StackOverflow	SELECT * FROM dbo.Users;	...

Priority	FindingsGroup	Findings	URLs
1	sp_BlitzFirst	2020-05-20 05:05:00.00:00.00000000 +00:00	From Your Community Volunteers
2	50	Query Problems	Plan Cache Evicted Recently
3	50	Server Performance	Page Life Expectancy Low
4	100	Query Performance	Queues with 10000+ cardinality misestimations
5	100	Query Performance	Queues with 10000+ cardinality misestimations
6	200	Wait Stats	ASYNC_NETWORK_IO
7	250	Server Info	Batch Requests per Sec
8	250	Server Info	CPU Utilization

Pattern	Sample Ended	Seconds Sample	wait_type	wait_category	Wait Time (Seconds)
WAIT STATS	2020-05-20 04:20:19.4319347-07:00	29	ASYNC_NETWORK_IO	Network IO	16.3



3.3 p17

ASYNC_NETWORK_IO

Slow client machines, underpowered app server VMs

Application processing data row-by-row
instead of just getting it all from SQL Server first

Slow network connections, especially WANs or VPNs

Good news! It's not a database problem.*

* - It's always a database problem.



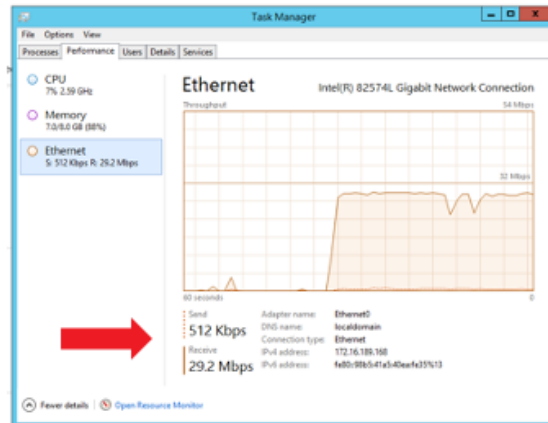
3.3 p18

Sanity check: is it network?

1Gb Ethernet:~100MB/sec

If you're actually saturating the network with query results, start asking tough questions.

(This screenshot is me downloading an ISO file.)



3.3 p19

Tracking down the apps involved

Run `sp_WhoIsActive` repeatedly and look for `ASYNC_NETWORK_IO` in the `wait_info`:

The screenshot displays two SQL Server query windows. The left window, titled 'SQLQuery1.sql', contains the following SQL code:

```
1 SET STATISTICS IO ON;
2 GO
3 SELECT * FROM dbo.Users;
4
```

The right window, titled 'SQLQuery2.sql', shows the results of the `sp_WhoIsActive` stored procedure. The results are displayed in a table with columns: `dd hh:mm:ss.mss`, `session_id`, `sql_text`, `login_name`, `wait_info`, `CPU`, and `temp`. A red arrow points to a row where the `wait_info` column contains the value `(2ms)ASYNC_NETWORK_IO`.

dd hh:mm:ss.mss	session_id	sql_text	login_name	wait_info	CPU	temp
00:00:00.644	56	<Query - SELECT * FROM dbo.Users ->	SQL2019\Brent	NULL	130	
00:00:00.627	56	<Query - SELECT * FROM dbo.Users ->	SQL2019\Brent	NULL	180	
00:00:01.627	56	<Query - SELECT * FROM dbo.Users ->	SQL2019\Brent	NULL	231	
00:00:01.234	56	<Query - SELECT * FROM dbo.Users ->	SQL2019\Brent	NULL	291	
00:00:01.457	56	<Query - SELECT * FROM dbo.Users ->	SQL2019\Brent	(2ms)ASYNC_NETWORK_IO		

Scroll across to the app, host

	host_name	database_name	program_name	st
	SQL2019	StackOverflow	Microsoft SQL Server Management Studio - Query	2

Go to the application owner with the query & host

Ask to see the source code for what's running that query to find out if it's doing row-by-row processing

Check the app server to see if it's maxed out on CPU or RAM



3.3 p21

Recap



3.3 p22

Hardware waits

WRITELOG: we need to write less to the log, or get lower-latency transaction log storage.

HADR_SYNC_COMMIT: the cost of sync replication.

ASYNC_NETWORK_IO: it's not our problem, but we have to help the developers and sysadmins find it.



3.3 p23

Setting up for the lab

1. Restart the SQL Server service (clears stats)
2. Restore your StackOverflow database
3. Copy & run the setup script:
BrentOzar.com/go/serverlab5
4. Start SQLQueryStress:
 1. File Explorer, D:\Labs, run SQLQueryStress.exe
 2. Click File, Open, D:\Labs\ServerLab5.json
 3. Click Go



3.3 p24