



BRENT OZAR
UNLIMITED®

How to Measure Your SQL Server

1.1 p1

A performance tuner's job

Pick metrics that you want to improve

Measure those metrics before and after changes

Understand when you have the wrong hardware for your license

1.1 p2



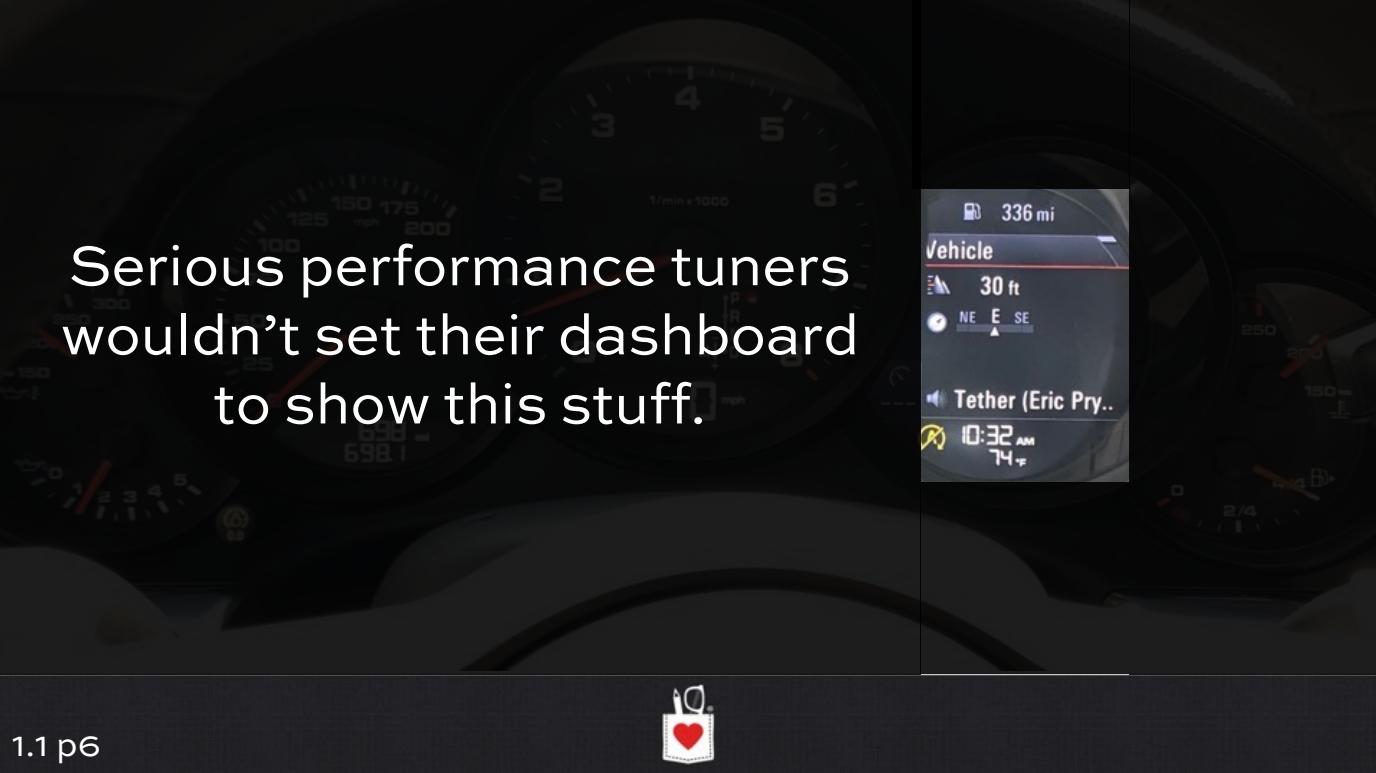
Pop quiz: how fast is a “fast” car?

1.1 p3









Serious performance tuners
wouldn't set their dashboard
to show this stuff.

1.1 p6



Best selling vehicle in the US: Ford F150



	0-60, seconds	HP	Weight, LBs	Base Price, USD
Ford F150 V6	7.6	290	5,265	\$28K
1.1 p8				



We could go faster by reducing weight.



	0-60, seconds	HP	Weight, LBs	Base Price, USD
Ford F150 V6	7.6	290	5,265	\$28K
Audi A3	6.8	200	2,800	\$34K
1.1 p1O				



We could also add power:
Ford sells an F150 with a bigger engine.



And an Audi RS3 for hooligans.



	0-60, seconds	HP	Weight, LBs	Base Price, USD
Ford F150 V6	7.6	290	5,265	\$28K
Ford F150 Raptor	5.5	450	5,673	\$65K
Audi A3	6.8	200	2,800	\$34K
Audi RS3	3.8	401	2,900	\$60K

1.1 p13



“No no. Even FASTER.”

Well, big gains may require
big money or big compromises.

1.1 p14



Like buying new technologies...



That might have management headaches...



	0-60, seconds	HP	Weight, LBs	Base Price, USD
Ford F150 V6	7.6	290	5,265	\$28K
Ford F150 Raptor	5.5	450	5,673	\$65K
Audi A3	6.8	200	2,800	\$34K
Audi RS3	3.8	401	2,900	\$60K
Tesla Model S Plaid	2.0	1,020	4,833	\$131K

1.1 p17





Or maybe we make
compromises.

	0-60, seconds	HP	Weight, LBs	Base Price, USD
Ford F150 V6	7.6	290	5,265	\$28K
Ford F150 Raptor	5.5	450	5,673	\$65K
Audi A3	6.8	200	2,800	\$34K
Audi RS3	3.8	401	2,900	\$60K
Tesla Model S Plaid	2.0	1,020	4,833	\$131K
Ariel Atom 4	2.8	320	1,300	\$102K

1.1 p19



Tuning is all about knowing and improving your metrics.

1.1 p20



The 3 performance metrics

How much weight are we carrying around?

How fast can we get to 60mph?

What bottleneck can we fix to go faster?

(Insufficient power, not enough grip, bad aerodynamics, etc.)

1.1 p21



Pop quiz: how fast is a “fast” SQL Server?

1.1 p22



Many of us have no idea.

1.1 p23



“We have a real fast server!”



It's time to get some answers.

1. How much weight are we carrying?
2. How fast are we going?
3. How hard are we working to achieve that speed?

1.1 p25



Total data size in GB/TB

Rough guess: size of MDF/NDF data files on the server

More accurate: used space in files. For each user database:

```
SELECT SUM(size)
FROM sys.database_files
WHERE type = 0;
```

Over time for the past: backup sizes
in msdb.dbo.backupfile

In SaaS environments, chart the
number of databases too

1.1 p26



When database size changes

Going up? Make sure it's related to user counts. If not, we started adding new features and storing new data, or storing backups inside the database.

Going down? Hopefully people are dropping scratch tables.

(Note: not specifying active vs logged data here)

1.1 p27



Database size seen in the field

1-150GB – easy to handle with SQL Server Standard Edition on commodity hardware.

150-500GB – easy to handle with Enterprise Edition

Over 500GB – it starts to matter if this is active data, and how it's accessed (OLTP vs analytical)

Over 1TB OLTP data – starts to get very challenging

Very uncomfortable high end:
~100TB per SQL Server, and
~10-12k databases per server.

1.1 p28



Virtualization



It's time to get some answers.

1. How much weight are we carrying?
**Total data size in GB/TB,
and number of DBs.**
2. How fast are we going?:
3. How hard are we working to achieve that speed?

1.1 p30



Transactional server speedometer

Perfmon - SQLServer: SQL Statistics – Batch Requests/sec

Not Transactions/sec

Trend on an hourly basis and break it out by:

- Weekday vs weekend
- Business hours vs after hours
- Maintenance windows
(backups, DB maintenance)

1.1 p31



When Batch Requests/Sec changes

Going up? More users, more API calls, new functionality in the application

Going down? Less users or API calls, or code has been streamlined so more work is done in less queries



Batch Requests/Sec seen in the field

0 – 1,000 – easy to handle with commodity hardware.

1,000-5,000 – be careful, because one bad change to a query can knock a commodity server over.

5,000-25,000 – if you're growing, you should be making a scale-out or caching plan.

Over 25,000 – doable, but attention.



Data warehouses are different.

Just one bad report
can take down the box.

You can't gauge by
reports/minute either:
some reports are easy,
other reports are hard.

So how does Microsoft do it?

1.1 p34

SQL Server Data Warehouse Fast Track Reference Architecture Results					
FTDW Certification #2014-002	HP ProLiant DL580 Gen8 with HP PCIe LE Workload Accelerator 90TB reference architecture for Microsoft SQL Server 2014 Data Warehouse Fast Track	Report Date: 9/3/2014			
FTDW Rev. 5.4					
System Provider	System Name	Processor Type	Memory		
	HP DL580 G8	Intel Xeon E7-4890 v2 2.8 GHz (4/60/120)	1536 GB		
	Operating System	SQL Server Edition			
	Windows Server 2012 R2	SQL Server 2014 Enterprise Edition			
Storage Provider	Storage Information				
	6 x 5.2TB HP Workload Accelerator PCIe Flash devices for data and tempdb 2 x 500GB 7200RPM HDD for OS (RAID 1) 2 x 1.3TB HP Workload Accelerator PCIe Flash devices for log (RAID 1)				
	Primary Metrics				
	Rated User Data Capacity ² (TB)	Row Store Relative Throughput ² (MB/Sec)	Column Store Relative Throughput ² (MB/Sec)	Maximum User Data Capacity ² (TB)	
	90	314	419	108	
	Row Store				
Relative Throughput ² (Queries/Hr/TB)	Measured Throughput (Queries/Hr/TB)	Measured Scan Rate Physical (MB/Sec)	Measured Scan Rate Logical (MB/Sec)	Measured I/O Throughput (MB/Sec)	Measured CPU (Avg.) (%)
314	366	8,351	10,139	9,245	92
	Column Store				
Relative Throughput ² (Queries/Hr/TB)	Measured Throughput (MB/Sec)	Measured Scan Rate Physical (MB/Sec)	Measured Scan Rate Logical (MB/Sec)	Measured I/O Throughput (MB/Sec)	Measured CPU (Avg.) (%)
419	2,721	3,392	N/A	N/A	96

The reference configuration is a 2 socket system rated for 25TB using the FTDW V4 methodology.

² Assumes a data compression ratio of 5:1.

³ Percent ratio of the throughput to the row store throughput of the reference configuration.

⁴ Percent ratio of the throughput to the column store throughput of the reference configuration.

⁵ Reported metrics are based on the qualification configuration which specifies database size and SQL Server memory.



Measured scan rate (MB/sec)

How much data are we scanning? (logical & physical reads, like we covered in Think Like the Engine)

Row Store					
Relative Throughput ²	Measured Throughput (Queries/Hr/TB)	Measured Scan Rate Physical (MB/Sec)	Measured Scan Rate Logical (MB/Sec)	Measured I/O Throughput (MB/Sec)	Measured CPU (Avg.) (%)
314	366	8,351	10,139	9,245	92

Column Store					
Relative Throughput ³	Measured Throughput (Queries/Hr/TB)	Measured Scan Rate Physical (MB/Sec)	Measured Scan Rate Logical (MB/Sec)	Measured I/O Throughput (MB/Sec)	Measured CPU (Avg.) (%)
419	2,721	3,392	N/A	N/A	96

1.1 p35

It's time to get some answers.

1. How much weight are we carrying?
**Total data size in GB/TB,
and number of DBs.**
2. How fast are we going?:
Batch Requests per second.
3. How hard are we working to achieve that speed?

1.1 p36



How SQL Server schedules CPU

What's Running Now

What's Waiting (Queue)

1.1 p37



Assume we have the world's smallest SQL Server – it's only got one core.

How SQL Server schedules CPU

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

1.1 p38



I run a query getting all of the restaurants. My Restaurants table happens to be mostly in cache, so it fires off and starts running. It will KEEP running – there's no concept of sharing CPU cycles in SQL Server. A query runs until it's done, or until it needs to wait on something like locks. More on that in a second.

How SQL Server schedules CPU

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

```
SELECT *  
FROM dbo.SoccerClubs  
(By Richie)
```

```
SELECT *  
FROM dbo.Resorts  
(By Erika)
```

1.1 p39



While my query is consuming CPU, other people's queries pile up behind me.

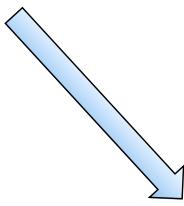
How SQL Server schedules CPU

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

```
SELECT *  
FROM dbo.SoccerClubs  
(By Richie)  
  
SELECT *  
FROM dbo.Resorts  
(By Erika)
```



1.1 p40



But the instant my query needs something that isn't in cache, like I need to wait for a locked page or I need to wait for something to come back from disk, I go to the back of the line.

How SQL Server schedules CPU

What's Running Now

What's Waiting (Queue)

```
←          SELECT *  
           FROM dbo.SoccerMatches  
        (By Richie)
```

```
SELECT *  
FROM dbo.Resorts  
(By Erika)
```

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

1.1 p41



Other people's queries can then jump in. While mine is waiting, SQL Server tracks the number of milliseconds that I'm waiting on stuff.

Stuff we wait for

Resources:

CPU, memory, storage, network, latches, locks

Stuff outside of SQL Server (Preemptive):

COM, OLEDB, CLR

System Tasks:

Lazywriter, trace, full text search

1.1 p42



Today, we're only focusing on that top list, Resources, because those are the most common problems for end user queries.

Is this SQL Server working hard?

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

1.1 p43



While my query is consuming CPU, other people's queries pile up behind me.

What about now?

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

```
SELECT *  
FROM dbo.SoccerClubs  
(By Richie)
```

```
SELECT *  
FROM dbo.Resorts  
(By Erika)
```

1.1 p44



While my query is consuming CPU, other people's queries pile up behind me.

Or now?

What's Running Now

```
SELECT *  
FROM dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

```
SELECT * FROM dbo.SoccerClubs  
(By Richie)
```

```
SELECT * FROM dbo.Resorts  
(By Erika)
```

```
EXEC sp_WhoIsActive  
(By Adam Machanic)
```

```
BACKUP DATABASE SQLbits  
(By Ola Hallengren)
```

```
DBCC CHECKDB  
(By Paul Randal)
```



1.1 p45

While my query is consuming CPU, other people's queries pile up behind me.

Some queries are simple

The screenshot shows the SQL Server Management Studio interface. A T-SQL query is entered in the query window:

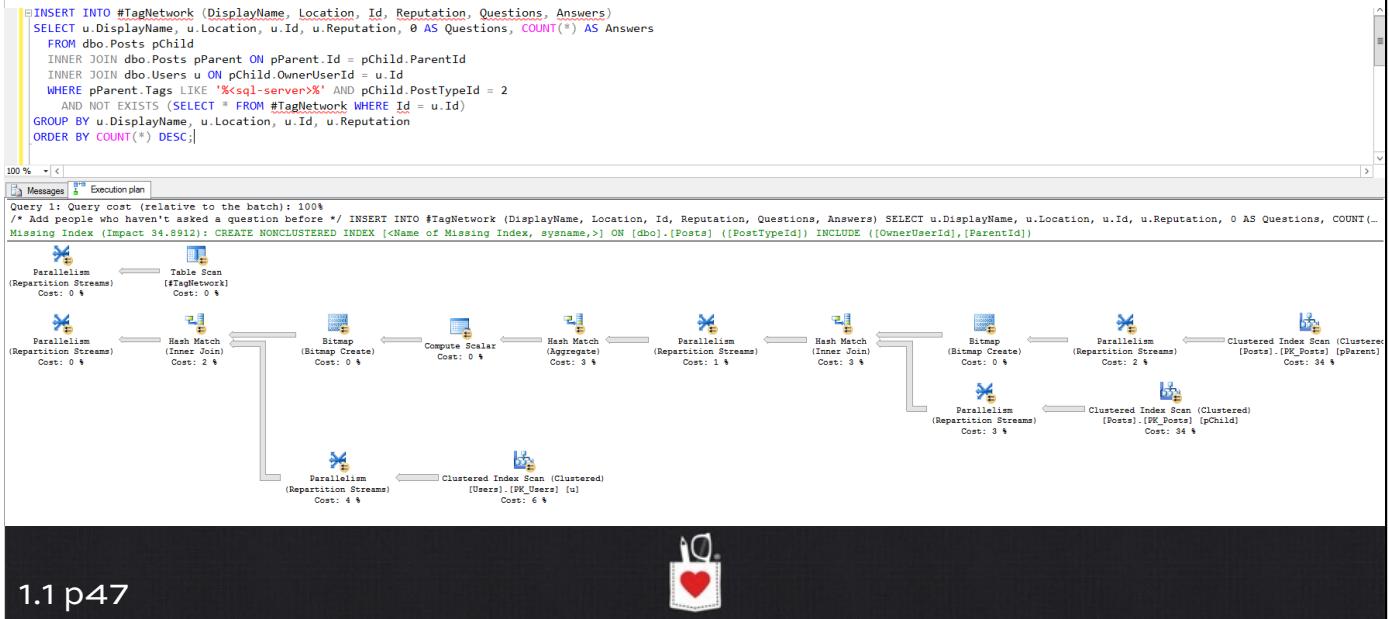
```
SELECT Id, ParentId, PostTypeId, OwnerUserId  
FROM dbo.Posts p  
WHERE p.Tags LIKE '%<sql-server>%';
```

The execution plan is displayed below the query results. It consists of three main components connected by arrows:

- Clustered Index Scan (Clustered)** [Posts].[PK_Posts] [p] (Cost: 95 %)
- Parallelism (Gather Streams)** (Cost: 5 %)
- SELECT Cost: 0 %**

Below the execution plan, the message "Query 1: Query cost (relative to the batch): 100%" is shown. The status bar at the bottom left indicates "1.1 p46".

Some queries have a lot going on



These were simple little queries.

What's Running Now

```
SELECT * FROM  
dbo.Restaurants  
(By Brent)
```

What's Waiting (Queue)

```
SELECT *  
FROM dbo.SoccerClubs  
(By Richie)
```

```
SELECT *  
FROM dbo.Resorts  
(By Erika)
```

1.1 p48



While my query is consuming CPU, other people's queries pile up behind me.

But one query looks more like this:

What's Running Now

Index scan of Table1

What's Waiting (Queue)

Index scan of Table2

Index scan of Table3

Index scan of Table4

1.1 p49



While my query is consuming CPU, other people's queries pile up behind me.

So for each core, you'll likely see:

What's Running Now

One task running,
using CPU

What's Waiting (Queue)

Several
(or several DOZEN)
tasks stacked up
waiting on stuff

1.1 p50



While my query is consuming CPU, other people's queries pile up behind me.

If we do this for one second, how many seconds of waits?

What's Running Now

One task running,
using CPU

What's Waiting (Queue)

15 tasks waiting on
storage

3 tasks waiting on
locks

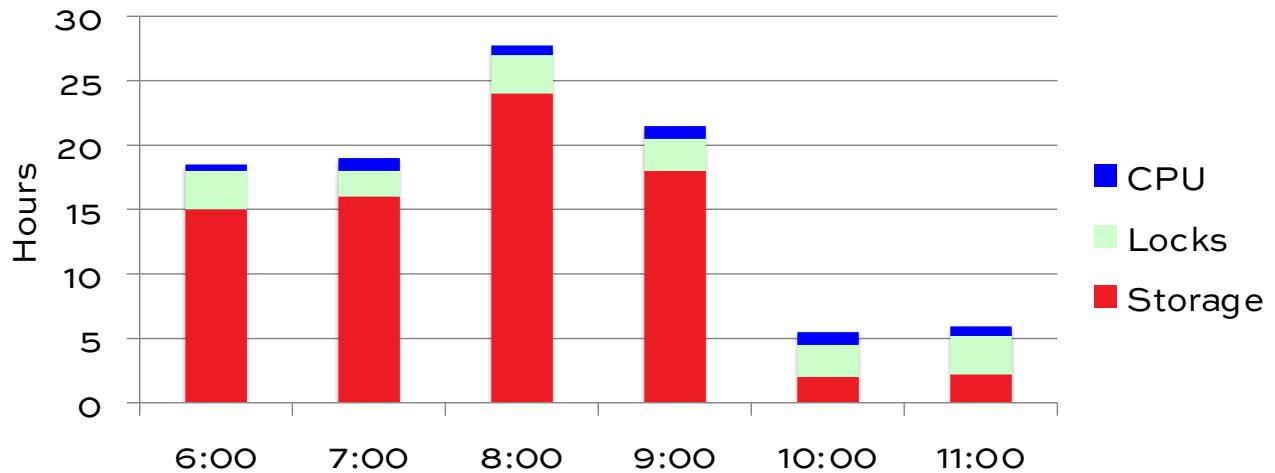
1 task waiting on CPU

1.1 p51



While my query is consuming CPU, other people's queries pile up behind me.

Wait Time Ratio: hours of wait time per hour



1.1 p52



How hard is your server working?

Dynamic Management View (DMV)

sys.dm_os_wait_stats

Tracked cumulatively over time

Trend on an hourly basis and break it out by:

- Weekday vs weekend
- Business hours vs after hours
- Maintenance windows
(backups, DB maintenance)

1.1 p53



When Wait time changes

If it went up:

- More batch requests/sec
- Queries were tuned (badly)
- Storage got slower
- Hardware is shared

If it went down:

- Less batch requests/sec
- Queries were tuned
- Indexes were tuned
- Memory was added

1.1 p54



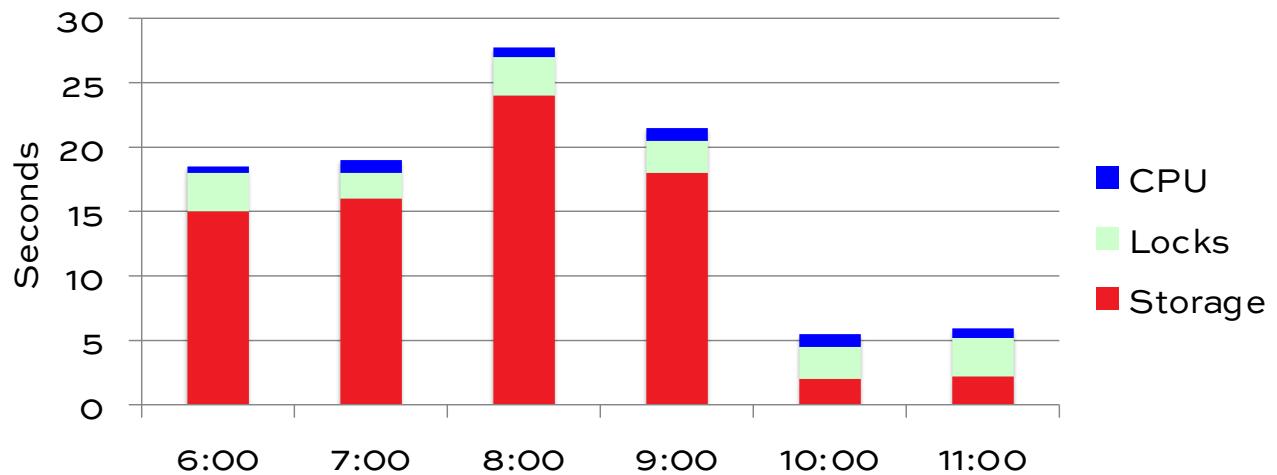
Wait time ratio: Wait time per core per second

- – Your server isn't doing anything.
- 1 second of waits – You're still not doing much.
- Multiple second per core – now we're working!
And we should probably be tuning.

1.1 p55



You want the same units of measure.



It's time to get some answers.

1. How much weight are we carrying?
**Total data size in GB/TB,
and number of DBs.**
2. How fast are we going?:
Batch Requests per second.
3. How hard are we working to achieve that speed?
**Wait time ratio:
Wait time per core per hour (or second, minute)**

1.1 p57



Quick snapshot: sp_BlitzFirst

1.1 p58



Quick snapshot: sp_BlitzFirst

Free at FirstResponderKit.org

Totally free diagnostic tool.

Installs in the master database.

Runs in 5 seconds ideally, but can be more under load.

By default, shows waits for a 5-second sample now.

@SinceStartup = 1 shows waits since, uh, startup.

1.1 p59



My favorite feature: “No Problems Found”

```
1 | sp_BlitzFirst
```

100 %

Results Messages

Priority	FindingsGroup	Finding	URL	Details
1	0	sp_BlitzFirst 2017-0...	From Your Community Volunteers	http://FirstResponderKit.org/ <?Click To See Details -- We hope
2	1	No Problems Found	From Your Community Volunteers	http://FirstResponderKit.org/ <?Click To See Details -- Try runnin
3	250	Server Info	Batch Requests per Sec	http://www.BrentOzar.com/go/measure <?Click To See Details -- 0 -- ?>
4	250	Server Info	CPU Utilization	http://www.BrentOzar.com/go/cpu <?Click To See Details -- 2%. Ring
5	250	Server Info	Wait Time per Core per Sec	http://www.BrentOzar.com/go/measure <?Click To See Details -- 0 -- ?>
6	251	Server Info	Database Count	http://www.BrentOzar.com/askbrent/ <?Click To See Details -- 22 -- ?>
7	251	Server Info	Database Size, Total GB	http://www.BrentOzar.com/askbrent/ <?Click To See Details -- 107.6475
8	255	Thanks!	From Your Community Volunteers	http://FirstResponderKit.org/ <?Click To See Details -- To get he

1.1 p60



Typical output under load

Priority	FindingsGroup	Finding	URL	Details	How
1	sp_BlitzFirst 2017-03-0...	From Your Community Volunteers	http://FirstResponderKit.org/	<?ClickToSeeDetails -- We hope you found this t...	NUL
2	10	Server Performance	Poison Wait Detected: THREADPOOL	http://www.brentozar.com/go/poi...	<?ClickToSeeDetails -- For 0 seconds over the l...
3	50	Server Performance	High CPU Utilization	http://www.BrentOzar.com/go/cpu...	<?ClickToSeeDetails -- 98%. Ring buffer details: ...
4	200	Wait Stats	CXPACKET	http://www.brentozar.com/sql/wai...	<?ClickToSeeDetails -- For 25750 seconds over ...
5	200	Wait Stats	SOS_SCHEDULER_YIELD	http://www.brentozar.com/sql/wai...	<?ClickToSeeDetails -- For 237 seconds over th...
6	200	Wait Stats	MEMORY_ALLOCATION_EXT	http://www.brentozar.com/sql/wai...	<?ClickToSeeDetails -- For 55 seconds over the ...
7	250	Server Info	Batch Requests per Sec	http://www.BrentOzar.com/go/me...	<?ClickToSeeDetails -- 45 -- ?>
8	250	Server Info	CPU Utilization	http://www.BrentOzar.com/go/cpu...	<?ClickToSeeDetails -- 98%. Ring buffer details: ...
9	250	Server Info	Wait Time per Core per Sec	http://www.BrentOzar.com/go/me...	<?ClickToSeeDetails -- 406 -- ?>
10	251	Server Info	Database Count	http://www.BrentOzar.com/askbre...	<?ClickToSeeDetails -- 22 -- ?>
11	251	Server Info	Database Size, Total GB	http://www.BrentOzar.com/askbre...	<?ClickToSeeDetails -- 107.64759063671 -- ?>
12	255	Thanks!	From Your Community Volunteers	http://FirstResponderKit.org/	<?ClickToSeeDetails -- To get help or add your o...

Also telling: what isn't there

Backups, restores	PLE low
DBCCs	Slow reads or writes
Data or log file growing or shrinking	Compilations/sec high
Long-running query blocking others	Plan cache erased recently

1.1 p62



Most useful parameters

@SinceStartup = 1

@ExpertMode = 1

@Seconds = 60

@OutputDatabaseName = 'DBAtools',

@OutputSchemaName = 'dbo',

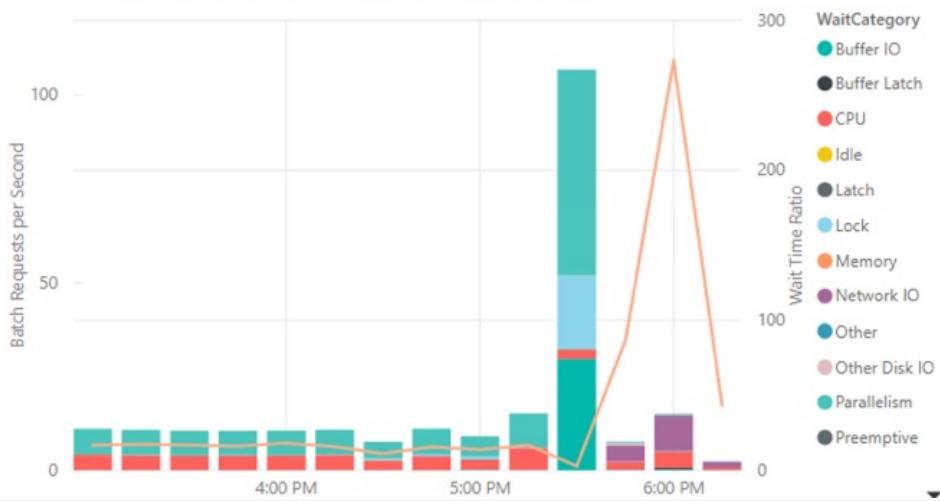
@OutputTableName = 'BlitzFirstResults'

1.1 p63



Monitoring tools overlay these.

Vital Stats: Batch Requests/sec and Wait Stats



1.1 p64





Tuning is all about 3 numbers.

1. Total database size (and quantity)
2. Batch Requests/Second
3. Wait Time Ratio:
Wait Time per Core per Hour (or per sec)

Capture these metrics with sp_BlitzFirst.

**Your top wait stat is
where to focus your tuning.**

1.1 p66

