

Afternoon lab setup

In the afternoon lab, you'll have 3 choices:

1. Tune a single query, or
2. Run a load test against your lab SQL Server, then use `sp_BlitzCache` to figure out which queries to tune, then tune 1, or
3. Run `sp_BlitzCache` against your live production server to figure out which queries to tune



1.5 p1

If you want to do #1 or #2:

1. Restart your SQL Server service (clears all stats)
2. Restore your StackOverflow database (Agent job)
3. Copy & run the setup script for Lab 2

And if you want to do #2, the load test,
start SQLQueryStress
with QueryLab2.json



1.5 p2



BRENT OZAR
UNLIMITED®

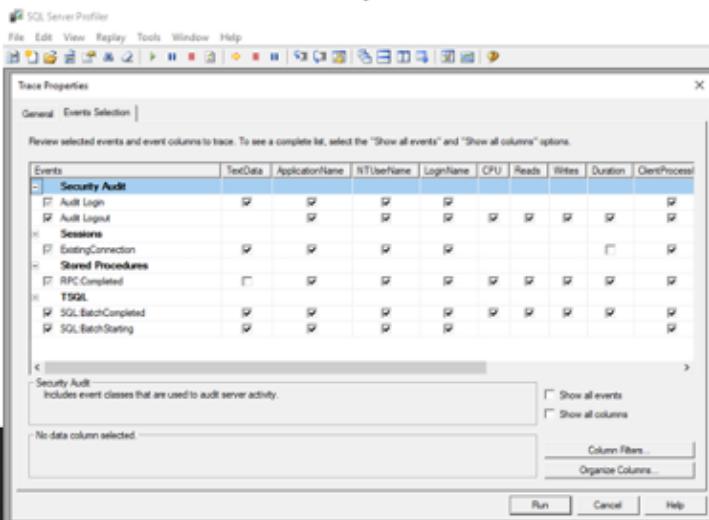
Find the Right Queries to Tune with `sp_BlitzCache`

Cache me outside, howbow dah

1.5 p3

When I was your age...

If I wanted to know what queries were slow,
I had to start a SQL Profiler trace in advance.



1.5 p4

Profiler has a lot of problems.

Must be started in advance

Can have a huge performance overhead

**Doesn't aggregate data together
(hides the death-by-a-thousand-cuts scenario)**



1.5 p5

Over time, we got more tools.

SQL 2008: Extended Events

SQL 2016: Query Store

SQL 2017/2019: Lightweight Profiling

All have to be configured in advance

All can have an ugly performance impact



1.5 p6

So I use the plan cache instead.

SQL Server already caches plans for queries

It's on by default (and you can't turn it off)

It has no additional overhead

The big drawbacks:

- It doesn't cache every query**
- It clears out over time**
- Stored in XML, which sucks to query**



1.5 p7

Enter `sp_BlitzCache`

Main authors: Erik Darling, Jeremiah Peschka

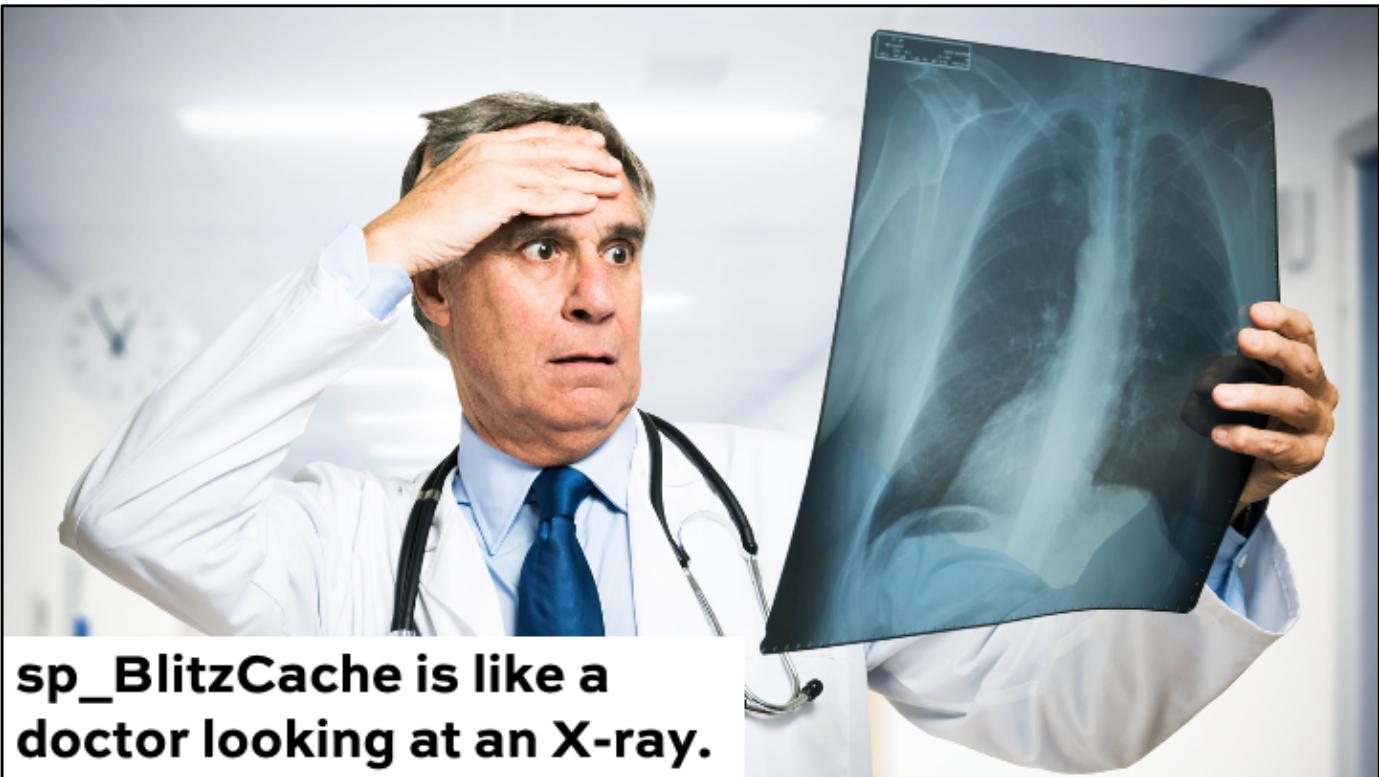
Totally free, open source: FirstResponderKit.org

**Shows your top 10 most
resource-consuming queries**

**Gives you advice on why they're
consuming resources**



1.5 p8



**sp_BlitzCache is like a
doctor looking at an X-ray.**

X-rays are taken after the fact

The damage has already happened

We can't tell exactly what happened,
like exactly *how* you broke that leg

We can't tell exactly how to prevent it,
but we can give you some guidelines



1.5 p10

Abra-macabre

| Use it when | Not when |
|---|---|
| Investigating query-related performance issues after the fact | Server is falling over right now |
| Tying queries to missing index requests | Experiencing locking, blocking, deadlocks |
| Posting questions about performance tuning | You need to know what queries are executing right now |
| Getting to know a server better | Wondering if your indexes are fragmented |



1.5 p11

```

DBCC FREEPROCCACHE;
GO
CREATE OR ALTER PROC dbo.usp_UsersByLocation @Location sql_variant AS
    SELECT *
    FROM dbo.Users
    WHERE Location = @Location
    ORDER BY DisplayName;
GO
EXEC usp_UsersByLocation @Location = 'Telluride'
GO
sp_BlitzCache

```

150 % ▾

Results Messages

| Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|---------------|---------|---|--|---|--|
| StackOverflow | 108.348 | CREATE PROC dbo.usp_UsersByLocation @Location ... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML.xmle="http://brentozar.com/blitzcache/showplanxml.aspx?...">ShowPlanXML.xmle="http://brentozar.com/blitzcache/showplanxml.aspx?..." |
| StackOverflow | 108.348 | SELECT * FROM dbo.Users WHERE Location = @L... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML.xmle="http://brentozar.com/blitzcache/showplanxml.aspx?...">ShowPlanXML.xmle="http://brentozar.com/blitzcache/showplanxml.aspx?..." |

<

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|---------------------------------|--|---|--|------------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/sql20d... | If these percentages are high, it may be a sign of ... | 999 |
| 2 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 5 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 6 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plan-detect... | Parallel plans detected. These warrant investigatio... | 6 |
| 7 | Need more help? | Paste your plan on the internet! | http://pastetheplan.com | This makes it easy to share plans and post them t... | 2147483646 |
| 8 | Thanks for using sp_BlitzCache! | From Your Community Volunteers | http://FirstResponderKit.org | We hope you found this tool useful. Current versio... | 2147483647 |

What you're looking at

Top: your top 10 most resource-intensive queries

| Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|---------------|---------|---|--|---|---|
| StackOverflow | 108,348 | CREATE PROC dbo.usp_UsersByLocation @Location ... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML.xmlnse="Http" |
| StackOverflow | 108,348 | SELECT * FROM dbo.Users WHERE Location = @L... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML.xmlnse="Http" |

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|---------------------------------|--|---|--|------------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/sql2ed... | If these percentages are high, it may be a sign of ... | 999 |
| 50 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 50 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 100 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 200 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 200 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plans-detect... | Parallel plans detected. These warrant investigatio... | 6 |
| 255 | Need more help? | Please visit our website for more information. | http://pastetheplan.com | This makes it easy to share plans and post them t... | 2147483646 |
| 255 | Thanks for using sp_BlitzCache! | From Your Community Volunteers | http://FirstResponderKit.org | We hope you found this tool useful. Current versio... | 2147483647 |

Bottom: decoder ring for the top's Warnings



1.5 p13

How stored procedures show up

This proc shows up twice in sp_BlitzCache:

- “Procedure or Function” = proc’s total impact
- “Statement (parent)” = this one line’s impact

The screenshot shows a SQL query window with the following content:

```
CREATE OR ALTER PROC dbo.usp_UsersByLocation @Location sql_variant AS
    SELECT *
    FROM dbo.Users
    WHERE Location = @Location
    ORDER BY DisplayName;
GO
EXEC usp_UsersByLocation @Location = 'Telluride'
GO
sp_BlitzCache
```

Below the query window is a results grid from sp_BlitzCache:

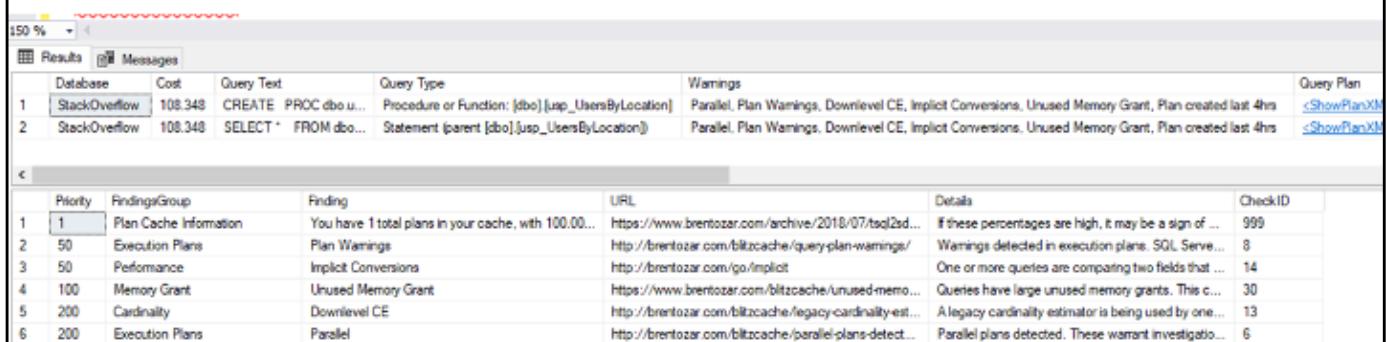
| | Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|---|---------------|---------|---|--|---|-----------------------------|
| 1 | StackOverflow | 108.348 | CREATE PROC dbo.usp_UsersByLocation @Location ... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML |
| 2 | StackOverflow | 108.348 | SELECT * FROM dbo.Users WHERE Location = @L... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit ... | ShowPlanXML |

You will be tempted

You'll wanna jump into the query text

You'll wanna click on the query plan

Hold that thought, speedy:
read the warnings first



| | Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|---|---------------|---------|----------------------|--|--|-----------------------------|
| 1 | StackOverflow | 108.348 | CREATE PROC dbo.u... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |
| 2 | StackOverflow | 108.348 | SELECT * FROM dbo... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|------------------------|--|---|---|---------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/sql20d... | If these percentages are high, it may be a sign of ... | 999 |
| 2 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 5 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 6 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plans-detec... | Parallel plans detected. These warrant investigation... | 6 |

Dissecting this one

Parallel = the plan used multiple cores

Plan Warnings = there will be a yellow bang when you open the query plan

Implicit Conversions, Unused Grant, Downlevel CE: this is where the second result set helps

| Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|-----------------|---------|----------------------|--|--|-----------------------------|
| 1 StackOverflow | 108.348 | CREATE PROC dbo.u... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |
| 2 StackOverflow | 108.348 | SELECT * FROM dbo... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|------------------------|--|---|--|---------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/tag2id... | If these percentages are high, it may be a sign of ... | 999 |
| 2 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 5 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 6 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plans-detec... | Parallel plans detected. These warrant investigation. | 6 |

The warnings are just that

We're trying to warn you

It's not that you need to *fix* all of them, necessarily:
sometimes they're just a heads up, like "Parallel"

For this class, you can ignore "Downlevel CE" and
"Plan created last 4 hrs" (but maybe not in real life)

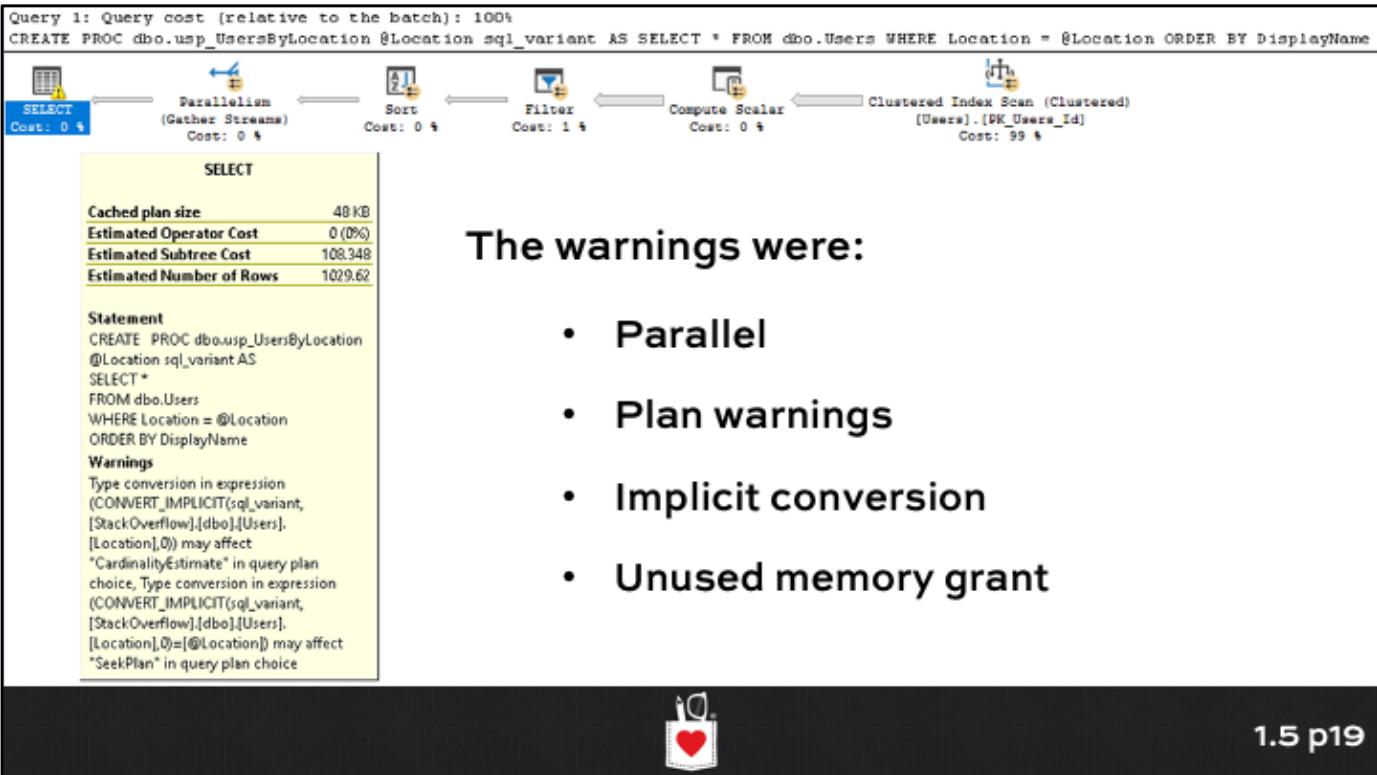
| Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|-----------------|------------------------|--|---|--|-----------------------------|
| 1 StackOverflow | 108.348 | CREATE PROC dbo.u... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |
| 2 StackOverflow | 108.348 | SELECT * FROM dbo... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |
| < | | | | | |
| Priority | FindingsGroup | Finding | URL | Details | CheckID |
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/tag2id... | If these percentages are high, it may be a sign of ... | 999 |
| 2 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 5 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 6 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plans-detec... | Parallel plans detected. These warrant investigation. | 6 |

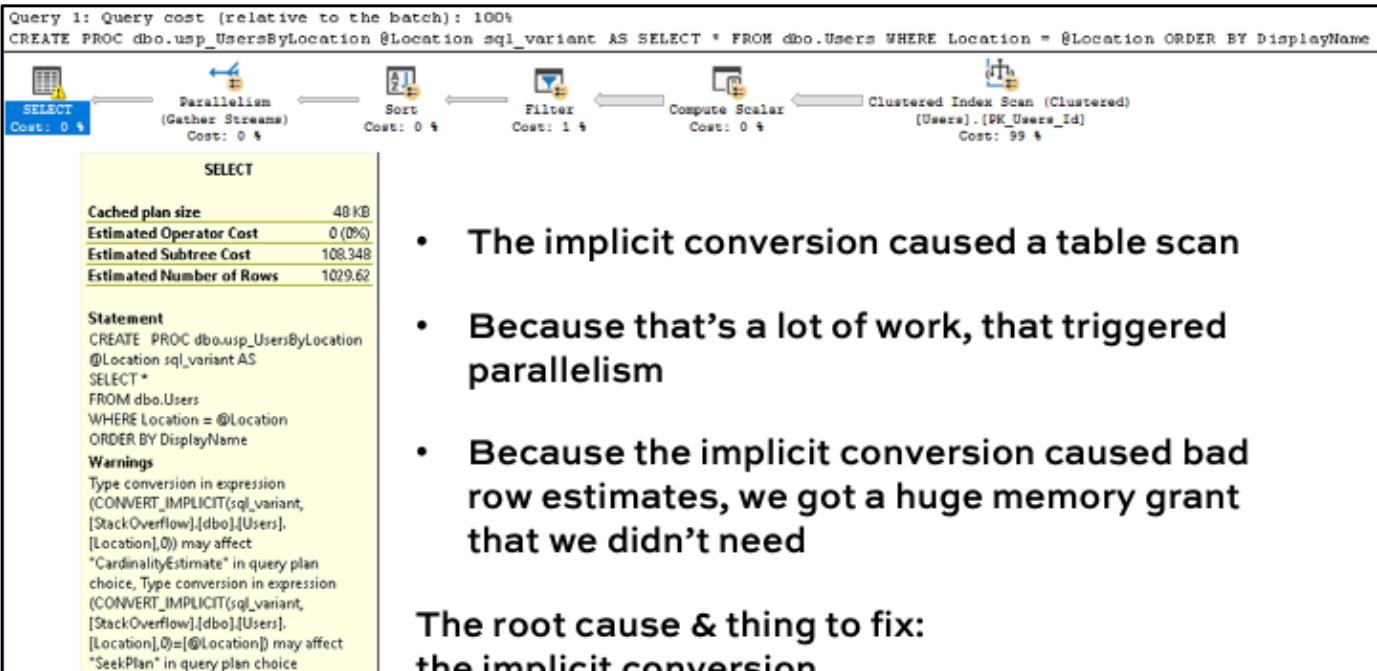
My sp_BlitzCache workflow

1. Make sure I don't have that Priority 1 warning about all of my plans being new in the last 4-24 hours (more on that in Mastering Server Tuning)
2. Look at the first row's Warnings, make notes
3. Open the first query plan and start analyzing

| Database | Cost | Query Text | Query Type | Warnings | Query Plan |
|-----------------|---------|----------------------|--|--|-----------------------------|
| 1 StackOverflow | 108.348 | CREATE PROC dbo.u... | Procedure or Function: [dbo].[usp_UsersByLocation] | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |
| 2 StackOverflow | 108.348 | SELECT * FROM dbo... | Statement (parent [dbo].[usp_UsersByLocation]) | Parallel, Plan Warnings, Downlevel CE, Implicit Conversions, Unused Memory Grant, Plan created last 4hrs | ShowPlanXML |

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|------------------------|--|---|--|---------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/tag2id... | If these percentages are high, it may be a sign of ... | 999 |
| 2 | Execution Plans | Plan Warnings | http://brentozar.com/blitzcache/query-plan-warnings/ | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | Memory Grant | Unused Memory Grant | https://www.brentozar.com/blitzcache/unused-memo... | Queries have large unused memory grants. This c... | 30 |
| 5 | Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-est... | A legacy cardinality estimator is being used by one... | 13 |
| 6 | Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plans-detec... | Parallel plans detected. These warrant investigation. | 6 |





1.5 p20

Fix the parameter, and run it again

```
CREATE OR ALTER PROC dbo.usp_UsersByLocation @Location NVARCHAR(40) AS
    SELECT *
    FROM dbo.Users
    WHERE Location = @Location
    ORDER BY DisplayName;
GO
EXEC usp_UsersByLocation @Location = 'Telluride'
GO
sp_BlitzCache
CC
```

150 %

| | Results | Messages | | | | |
|---|---------------|----------|---|--|---|---------------------------|
| 1 | StackOverflow | 108.064 | CREATE PROC dbo.usp_UsersByLocation @Location ... | Procedure or Function: [dbo].[usp_UsersByLocation] | Missing Indexes (1). Parallel, Downlevel CE, Plan created last 4hrs | Show Plan |
| 2 | StackOverflow | 108.064 | SELECT * FROM dbo.Users WHERE Location = @Lo... | Statement [parent [dbo].[usp_UsersByLocation]] | Missing Indexes (1). Parallel, Downlevel CE, Plan created last 4hrs | Show Plan |

| Priority | FindingsGroup | Finding | URL | Details | CheckID |
|----------|-------------------------------------|--|---|---|------------|
| 1 | Plan Cache Information | You have 1 total plans in your cache, with 100.00... | https://www.brentozar.com/archive/2018/07/tsql2s... | If these percentages are high, it may be a sign of... | 999 |
| 2 | 50 Performance | Missing Indexes | http://brentozar.com/blitzcache/missing-index-request/ | Queries found with missing indexes. | 10 |
| 3 | 200 Cardinality | Downlevel CE | http://brentozar.com/blitzcache/legacy-cardinality-es... | A legacy cardinality estimator is being used by on... | 13 |
| 4 | 200 Execution Plans | Parallel | http://brentozar.com/blitzcache/parallel-plan-detect... | Parallel plans detected. These warrant investigat... | 6 |
| 5 | 255 Need more help? | Paste your plan on the internet! | http://pastetheplan.com | This makes it easy to share plans and post them ... | 2147483646 |
| 6 | 255 Thanks for using sp_BlitzCache! | From Your Community Volunteers | http://FirstResponderKit.org | We hope you found this tool useful. Current versi... | 2147483647 |

Because now Clippy has an idea.

```
Query 1: Query cost (relative to the batch): 1000
CREATE PROC dbo.usp_UsersByLocation @Location NVARCHAR(40) AS SELECT * FROM dbo.Users WHERE Location = @Location ORDER BY DisplayName
Missing Index (Impact 99.4902): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[Users] ([Location]) INCLUDE ([Ab
```



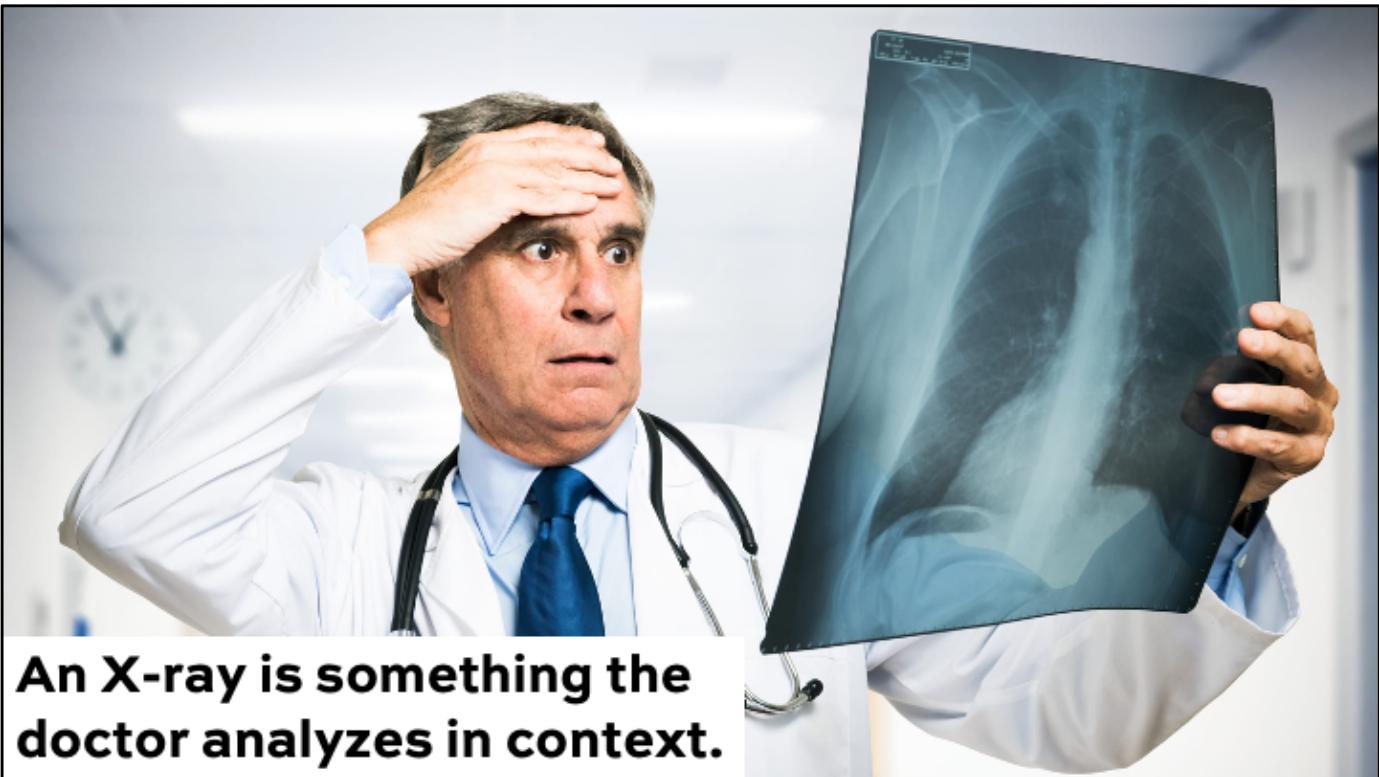
We didn't get a missing index recommendation before because of the implicit conversion.

There's always going to be something new to tune.

But this query would probably drop off the top 10.



1.5 p22



An X-ray is something the doctor analyzes in context.

X-rays have some hard truths.

For example, there's a strange object inside you.

They just can't tell how the object got there,
or how the doctor's going to remove it.



1.5 p24

sp_BlitzCache's hard truths

Query level metrics by total and average:

- Reads, writes
- Executions, executions per minute
- CPU
- Duration
- Newer SQL Server versions add memory grants, spills

XML plan analysis down to the operator level

Rolled up and individual plan warnings

Links to more information, additional details



1.5 p25

Chloroform: stuff that disappears

Per-execution statistics like:

- Variances between estimated & actual rows
- Spills per execution
- Parameters used to execute (not compile) a query

Queries that hide:

- OPTION (RECOMPILE)
- Dynamic SQL
- Unparameterized SQL
- Servers with memory pressure (or resource_semaphore)



1.5 p26

Helpful parameters

@SortOrder: metric you want to order results by

@DatabaseName: focus on one database

@StoredProcName: focus on a proc (no schema prefix)

@QueryFilter: Procedures, statements, functions



1.5 p27

Default @Top = 10

By default, we only look at the top 10 plans.

Shredding XML is expensive and can be time consuming.

**The more queries you want,
and the larger those queries are,
the longer execution will take.**

**Only set @Top to larger than 10 if you've
already tuned the top 10 as much as you can.
(You haven't.)**



1.5 p28

Performance Decoder Ring

Get your top wait type with `sp_BlitzFirst @SinceStartup =1`

| If your top wait type is... | Then use @SortOrder = |
|-----------------------------|-----------------------|
| CXPACKET | CPU, Reads |
| SOS_SCHEDULER_YIELD | CPU |
| THREADPOOL | CPU |
| LCK_** | Duration |
| RESOURCE_SEMAPHORE | Memory grant |
| PAGEIOLATCH_** | Reads |
| WRITELOG | Writes |



1.5 p29

Common Findings



1.5 p30

Implicit conversion

One of the most documented query performance problems around – yet we still find it constantly

When comparing two different data types, SQL has to convert one

This chews up CPU and prevents the efficient use of indexes

You'll also get pretty bad cardinality estimates

Get your barf bag



1.5 p31

| From | binary | varbinary | char | varchar | nchar | nvarchar | datetime | smalldatetime | date | time | datetimeoffset | datetime2 | decimal | numeric | float | real | begin | int(N)4 | smallint(N)2 | tinyint(N)1 | money | smallmoney | bit | timestamp | uniqueidentifier | image | text | sql_variant | xml | CLR_UDT | hierarchyid | | |
|------------------|--------|-----------|------|---------|-------|----------|----------|---------------|------|------|----------------|-----------|---------|---------|-------|------|-------|---------|--------------|-------------|-------|------------|-----|-----------|------------------|-------|------|-------------|-----|---------|-------------|--|--|
| To | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| binary | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| varbinary | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| char | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| varchar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nchar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| nvarchar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| datetime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| smalldatetime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| datetimeoffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| datetime2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| decimal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| numeric | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| float | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| real | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| begin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| int(N)4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| smallint(N)2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| tinyint(N)1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| money | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| smallmoney | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| timestamp | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| uniqueidentifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| image | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| text | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| sql_variant | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| xml | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CLR_UDT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| hierarchyid | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Legend:

- Explicit conversion
- Implicit conversion
- ✗ Conversion not allowed
- ◆ Requires explicit CAST to prevent the loss of precision or scale that might occur in an implicit conversion.
- Implicit conversions between XML data types are supported only if the source or target is untyped XML. Otherwise, the conversion must be explicit.

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-type-conversion-database-engine>



1.5 p32

Missing indexes

Missing indexes are SQL Server's cry for help

- They're a good place to *start* performance tuning

But when they come up, you need to ask some questions

- Are they the *only* missing index request in the query?
- Do you have any indexes that are close in definition?
- Is this really the best long-term index?
- Is this even the best index *right now*?



1.5 p33

Forced serialization

MAXDOP set to 1, or:

- Scalar/MSTVFs
 - Computed columns, check constraints
- TOP
- Global aggregates
- Sequence operators
- Backwards scans
- Recursive CTEs
- Some system tables and functions
- CLR that performs data access
- Dynamic cursors
- OUTPUT clause

Far and away the worst,
additional overhead

Can cause serial zones in
plans. Some different rules
with APPLY

Far less common. System
functions and views are less
likely to cause overall query
performance issues.



1.5 p34

Fixing forced serialization

Focus on the function

- Many other constructs mentioned can be fixed
 - Even backwards scans can be indexed for

Like crack and heroin

- No matter how many cautionary tales are told
 - People keep using them



1.5 p35

Table variables

Modifications are serialized

Only 1 row estimates, unless you recompile

Low estimates can lead to bad plan choices

There are some upsides:

- Survive errors and rollbacks
- Can be good for high frequency execution



1.5 p36

Expensive sorts

Now look, most sorting should be done in the application

- There are times when it's unavoidable

Sometimes SQL will inject a Sort you didn't ask for

- Merge Joins and Stream Aggregates require ordered data

Not every Sort is a problem

- Some are ticking time bombs

Some sorts are bigger than others

- Sorts can be sensitive to parameter sniffing issues



1.5 p37

Expensive key lookups

Like sorts, Key Lookups are sensitive to parameter sniffing

- For a predictably small number of rows, they're not an issue
- For a growing or varying amounts of rows, they can lead to larger performance issues

Be careful of what you fix, and how you fix it

- Output vs Predicate
- Long lists of columns
- Large columns (Xooo/MAX)



1.5 p38

Not-so-common findings



Columnstore indexes in row mode

ColumnStore indexes work best in Batch mode

- Executing in Row Mode inhibits many optimizations
- As of 2016, code that forces Row Mode is fairly limited

Why Batch Mode?

- Works on batches of rows rather than per-row
- Allows for row group elimination, local aggregations

Things that sabotage batch mode:

- Trivial plans (pre-2017)
- Data types that don't support Aggregate Pushdown
- Operators that don't support Aggregate Pushdown



1.5 p40

Unused memory grants

Look, some queries *need* memory grants

Some queries need *big* memory grants

But sometimes, they just *think* they need big memory grants

When they're wrong, they still get all that memory

They just don't *use* it

This can be bad on servers of *any* kind



1.5 p41

RAMifications

A short list of things that can be awful

- RESOURCE_SEMAPHORE* waits
 - Queries can't get enough memory to start running/compile
- Lots of memory taken from the buffer pool
 - Pages are forced out, queries read from disk
- Adding memory doesn't always fix it
 - They may just ask for *more memory* (percentage of max)
 - By default, queries can ask for 25% of server memory to run



1.5 p42

Expensive spools

We focus on Eager Index Spools only

Often, an Index Spool is a sign that you need an index

The funny thing is that when SQL Server chooses to spool data into an index, it doesn't register a missing index request

We dislike passive aggressive behavior,
like SQL Server sorting the whole thing
without asking for an index request



1.5 p43

Unmatched indexes

Means a filtered index was present, but it couldn't be used

They have some clever limitations in cached plans

- If a variable is used as the predicate to match the index definition, the index can't be used

This makes plenty of sense

- If your filtered index is on a BIT column
- A plan searching on 1 or 0 in a variable will have to work in either scenario



1.5 p44

Computed column UDFs

| Schema bound? | Persisted? | Indexed? | Shows Up In sp_Blitz Cache? | Once per row? | Forced Serial? |
|---------------|------------|----------|-----------------------------|---------------|----------------|
| No | No | No | Yes | Yes | Yes |
| Yes | No | No | Yes | Yes | Yes |
| Yes | No | Yes | No | No | Yes |
| Yes | Yes | No | No | No | Yes |
| Yes | Yes | Yes | No | No | Yes |



1.5 p45

dba.stackexchange.com/questions/187342/

Is there a way to prevent Scalar UDFs in computed columns from inhibiting parallelism?



Much has been written about the [perils of Scalar UDFs](#) in SQL Server. A casual search will return oodles of results.



21 There are some places where a Scalar UDF is the only option, though.

As an example: when dealing with XML: XQuery can't be used as a computed column definition. One option documented by Microsoft is to use a [Scalar UDF](#) to encapsulate your XQuery in a Scalar UDF, and then use it in a computed column.



6 This has various effects, and some workarounds.

- Executes row by row when the table is queried
- Forces all queries against the table to run serially

You can get around the row-by-row execution by schema binding the function, and either persisting the computed column, or indexing it. Neither of those methods can prevent the forced serialization of queries hitting the table, even when the scalar UDF isn't referenced.

Is there a known way to do that?

[sql-server](#) [functions](#) [parallelism](#) [computed-column](#)

asked 4 months ago

viewed 1,436 times

active 3 months ago

BLOG

[Updating Navigation fo Enterprise, and Stack](#)

FEATURED ON META

[Affiliate Ads Are Comin](#)

HOT META POSTS

11 [Community Promotion](#)

Linked



1.5 p46



sp_BlitzCache: plan cache X-ray

Huge number of parameters:
we've only scratched the surface

The Warnings column catches all
kinds of anti-patterns

You can't be expected to know
how to fix every anti-pattern, but
that's where the second result
set will help you back at the office



Related info from sniffing classes

`sp_BlitzCache` isn't the only way to analyze queries:

- Logging `sp_BlitzWho`, `sp_WhoIsActive` to table
- Extended Events with Erik Darling's `sp_HumanEvents`
- Query Store
- 2019's new default lightweight profiling, `air_quote_actual plans`



1.5 p49



**Questions?
Bio break?**