Performance Killers

* Poor indexing
* Inaccurate statistics
* Poor query design
* Poor execution plans, usually caused by bad parameter sniffing
* Excessive blocking and deadlocks
* Non-set-based operations, usually T-SQL cursors
* Poor database design
* Excessive fragmentation
* Nonreusable execution plans
* Frequent recompilation of queries
* Improper use of cursors
* Improper configuration of the database log
* Excessive use or improper configuration of tempdb

Parameter Sniffing

* The OPTIMIZE FOR (@parameter = value) query hint builds a reusable plan based on a specific value
* OPTIMIZE FOR (@parameter UNKNOWN) uses average distribution statistics for a particular parameter
* OPTIMIZE FOR UNKNOWN uses average distribution for all parameters (same effect as trace flag 4136)
* The WITH RECOMPILE stored procedure option compiles a fresh procedure plan for every execution
* The OPTION (RECOMPILE) query hint compiles a fresh plan for an individual statement

Investigate parameter sniffing

1. Look at execution plan

Parameter Information

1. Run below query to find plan cache

SELECT

DEQS.plan\_generation\_num,

DEQS.execution\_count,

DEQS.last\_logical\_reads,

DEQS.last\_elapsed\_time,

DEQS.last\_rows,

DEQP.query\_plan

FROM sys.dm\_exec\_query\_stats AS DEQS

CROSS APPLY sys.dm\_exec\_sql\_text(DEQS.plan\_handle) AS DEST

CROSS APPLY sys.dm\_exec\_query\_plan(DEQS.plan\_handle) AS DEQP

WHERE

DEST.objectid = OBJECT\_ID(N'dbo.F', N'TF');---changes

Events to Analyze Query Recompilation

sql\_batch\_completed or module\_end

sql\_statement\_recompile

sql\_batch\_starting or module\_start

sp\_statement\_completed or sql\_statement\_completed (Optional)

sp\_statement\_starting or sql\_statement\_starting (Optional)

Recompile Cause Data Column Reflecting Causes of Recompilation

Schema or bindings to regular table or view changed

Statistics changed

Object did not exist in the stored procedure plan but was created during execution

SET options changed

Schema or bindings to temporary table changed

Schema or bindings of remote rowset changed

FOR BROWSE permissions changed

Query notification environment changed

MPI view changed

Cursor options changed

WITH RECOMPILE option invoked

Avoiding Recompilations

• Don’t interleave DDL and DML statements.

• Avoid recompilation caused by statistics changes.

• Use the KEEPFIXED PLAN option.

• Disable the auto update statistics feature on the table.

• Use table variables.

• Avoid changing SET options within the stored procedure.

• Use the OPTIMIZE FOR query hint.

• Use plan guides.

Avoiding Changing SET Options Within a Stored Procedure

Following SET options ON:

• ARITHABORT

• CONCAT\_NULL\_YIELDS\_NULL

• 0U0TED\_IDENTIFIER

• ANSI\_NULLS

• ANSI\_PADDINC

• ANSI\_WARNINGS

• And NUMERIC\_R0UNDAB0RT should be OFF

* !< Condition vs. >= Conditione

>= comparison operator allows the optimizer to use an index on the column referred

1. Avoid Arithmetic Operators on the WHERE Clause Column

* Avoid Functions on the WHERE Clause Column
* SUBSTRING vs. LIKE

SUBSTRING function prevented the optimizer from using the index on the column. This function on the column made the optimizer use a clustered index scan.

* Date Part Comparison

Avoid data type conversion.

• Use EXISTS over COUNT(\*) to verify data existence.

• Use UNION ALL over UNION.

• Use indexes for aggregate and sort operations.

• Avoid local variables in a batch query.

• Be careful naming stored procedures