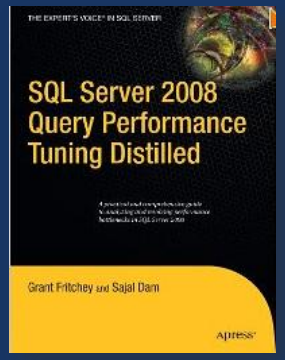
The World's Largest Community of SQL Server Professionals

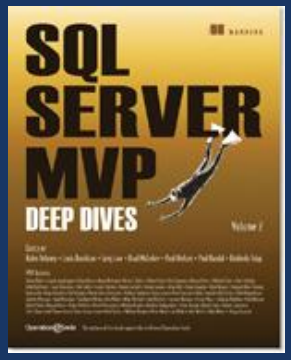
SQL Server Optimization Checklist

Grant Fritchey
Product Evangelist
Red Gate Software

Who?

- Product Evangelist for Red Gate Software
- Microsoft SQL Server MVP
- Author:
  - SQL Server Execution Plans
  - SQL Server 2008 Query Performance Tuning Distilled
  - Chapter in SQL Server MVP Deep Dives Volume 2





Goal

- Describe common and recognizable types of performance issues in order to provide best practices to avoid or mitigate them.

Test

Test

Test

Test

Test

Remember

- This is focused on performance tuning
- There are always exceptions
- Your situation may be different
- Did I mention test?
- Incremental changes
- Monitor your servers

Nested Table Valued User Defined Functions

- Problem
  - User defined functions that call user defined functions that call…
- Identification
  - Simplistic execution plans
  - Unrealistic costs in execution plans
  - Very slow performance
- Resolution
  - Use properly structured queries

Cost Threshold for Parallelism

- Problem
  - Simple queries execute in parallel
- Identification
  - Cost Threshold for Parallelism set to default (5)
  - Parallel plans for simple queries
  - Excessive cxpacket waits (not automatically bad)
  - Excessive queuing on CPU
- Resolution
  - Increase Cost Threshold
  - Simplify queries

Ad Hoc TSQL

- Problem
  - Memory problems caused by ad hoc tsql or bad ORM code
- Identification
  - Very low plan cache hit ratio
  - Excessive compile events
  - Very low page life expectancy
- Resolution
  - Work with ORM tool to resolve issues
  - Use stored procedures or parameterized queries
  - Enable ‘Optimize for Ad Hoc Workload’

Everything on One Disk

- Problem
  - Disk queuing due to all data and logs being on a single disk
- Identification
  - Excessive disk queues
  - Blocking from reads and writes
- Resolution
  - Increase the disks or LUNs on the system
  - Work with SAN team to ensure proper distribution

Blocking From Recompile

- Problem
  - Excessive, long-running, recompiles block system access
- Identification
  - Multiple blocked processes from single source
  - Wait on recompile
- Resolution
  - Reduce query complexity
  - Use table variables where applicable instead of temp tables
  - Use KEEP FIXED PLAN hint (but carefully)

Inappropriate Use of Query Hints

- Problem
  - FAST 1 query hint used everywhere
- Identification
  - Extremely simplistic execution plans
  - Execution plan shows estimated 1 row
- Resolution
  - Remove the hint
  - Deal with performance problem directly

Inappropriate Use Of Query Hints 2

- Problem
  - NO_LOCK query hint used everywhere
- Identification
  - Incorrect data returned
  - Extra rows
  - Missing rows
- Resolution
  - Remove the hint
  - Enable a type of snapshot isolation
  - Tune queries and indexes to reduce blocking

Removing Normalization

- Problem
  - Removing foreign key constraints to speed up inserts slows down reads
- Identification
  - Bad data due to missing constraints
  - Overly complex execution plans due to missing simplification step
- Resolution
  - Use foreign key constraints
  - Also use primary key and unique constraints

Indexes Don't Help Performance

- Problem
  - Adding an index doesn't speed up queries or it slows them down
- Identification
  - Execution plan ignores index
  - Execution plan changes to something unexpected
- Resolution
  - Verify index will help query
  - Validate that data supports selectivity

Applying Missing Indexes Doesn't Help

- Problem
  - Missing index recommendations are applied without knowledge or testing
- Identification
  - Indexes named [<Name of Missing Index, sysname,>]
  - Duplicate indexes
  - Excessive number of indexes
- Resolution
  - Test missing index suggestions
  - Validate need for all indexes

Deadlocks Hurt Performance

- Problem
  - Performance impact due to large number of deadlocks
- Identification
  - Deadlock errors in log
  - Complaints from users and errors in application
- Resolution
  - Reduce contention
  - Ensure processing order
  - Reduce execution time

Excessive Indexes Hurt Performance

- Problem
  - An index has been created on every column in the table
- Identification
  - Indexes never touched in sys.dm_db_index_usage_stats (but be careful)
- Resolution
  - Remove extraneous and unnecessary indexes
  - Do not add indexes without cause

Checklist

- Exercise in caution when nesting functions and views
- Change default settings for cost threshold for parallelism
- Enable ‘optimize for ad hoc workload’
- Distribute storage to multiple disk locations
- Write TSQL to avoid unnecessary recompiles
- Use query hints sparingly and appropriately
- Use normalization, primary key, foreign key and unique constraints
- Add indexes judiciously
- Test missing index suggestions carefully
- Deadlocks are a performance issue and must be dealt with

Goal

- Describe common and recognizable types of performance issues in order to provide best practices to avoid or mitigate them.

Resources

- Scarydba.com\presentation-resources

- Microsoft SQL Server 2008 Internals
  Kalen Delaney et al
- SQL Server MVP Deep Dives Vol I & II
  Kalen Delaney et al
- Inside SQL Server 2008 T-SQL Querying
  Itzik Ben Gan et al
- Brad's Sure DBA Checklist

Questions?

How would you… ?

What happens when… ?

Why does… ?

When do I… ?

SQL ServerCentral.com

The World's Largest Community of SQL Server Professionals

Thanks for Attending

Visit www.sqlservercentral.com for free SQL Server eBooks, articles, videos, blogs, news, and more.

Please Don't Forget to Turn in Your Evaluations