Writing efficient code for SQL Server Based Applications

Where to Start Looking for Your Performance Issues

About Me

- Security Related:
  - Infrastructure & Security Architect
  - Certified Information Systems Auditor (CISA)
  - Accredited CISA Trainer
  - Incident Handler / Penetration Tester

- SQL Server Related:
  - Data Architect
  - SQL Server security columnist / blogger
  - SQL Server and Security speaker & trainer

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Contact Information

K. Brian Kelley

Email: kbriankelley@acm.org
Twitter: @kbriankelley

Infrastructure/Security Blog: https://truthsolutions.wordpress.com
Personal Development Blog: https://gkdba.wordpress.com

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Agenda

- Understanding Execution Plans
- Indexes
- Row-By-Agonizing Row (RBAR)
- Cursors and Loops
- Functions
- Nested Views

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Understanding Execution Plans

The Key to Know What Is Going On

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Execution Plans

Some Keys:

- Production like amount and distribution of data
- Estimated and Actual may not match
- Pay attention to what you're asking for in the query
- You may not have permissions – check with your DBA

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Execution Plan Demo

Let’s Understand What We Are Looking at

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Execution Plans

- Scan vs. Seek
- Key Lookup
- Hash Match, Nested Loops, Merge Joins
- Cost

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Hash Match vs. Nested Loops vs. Merge Joins

- Merge Joins is additive (X table rows + Y table rows)
  - Requires sorted data
  - Tends to be cheapest unless sort is expensive
- Nested Loops is multiplicative (X rows * Y rows)
  - Take smaller resultset row-by-row
  - Compare each row of larger resultset
- Hash Match
  - When the other two can't be used
  - CPU and memory intensive (in comparison)

```
For I = 1 to 100
  For J = 1 to 100
  …
  Next J
Next I
```

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Understanding Indexes

How to Maximize for Performance

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Indexes

- Clustered = Physical sort (can only be one)
- Non-Clustered based on Clustered (if one exists)
- SQL Server can only use 1 per table for a query
- Bookmark Lookup (Key Lookup) = 2 Reads

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Indexes Demo

How Do They Work with Our Queries?

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Indexes Best Practices

- Use a **clustered index**.
- **Seeks** over **scans.**
- Too many indexes = performance issue.

(Exception: Read-only / slowly changing systems.)

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Why Can Too Many Indexes be Bad?

- Read / write database –
  - every time a row is inserted or deleted, every index has to be updated.
  - updated, it depends
    - Changing columns in the clustered index? **Yes**.
    - Changing columns otherwise? **Depends on the index.**

- Read only database –
  - Indexes are generally good. Cover what you can.
  - Think about storage requirements
  - Think about load times for warehousing

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Anti-Patterns

- SARGable vs. non-SARGable *(Search ARGument)*
  - Row-by-Agonzing-Row (RBAR)
- Implicit Conversions
- Unnecessary Complexity

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Anti-Patterns Demo

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Row-By-Agonizing-Row

Hidden Dangers in Your Queries

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

RBAR – Avoid it!

- Coined by SQL Server MVP Jeff Moden
- Stands for “Row by Agonizing Row”
- SQL Server designed for set-based operations
  - RBAR is iterative and procedural
  - Iterative is generally slow on SQL Server *(and all RDBMSes)*
- RBAR is “natural” because it fits with what we’re taught for other languages

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

RBAR vs. Tally Table

- Demo
  - Based on code by Jeff Moden
  - Updated to use SQL Server 2005+ DMV
  - Otherwise, it’s all his code

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

RBAR Demo

We Are Just Doing 5,000 rows. Imagine 5,000,000!

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Cursors and Loops

You've Guaranteed RBAR

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Cursors and Loops

- Cursors
  - Try to avoid these.
  - Typical with beginning developers and end users.
  - Iterative vs. Set-Based
- Any looping structures - WHILE in T-SQL
- Hidden RBAR
  - See Jeff's article on Triangular Joins
  - See Jeff's article on More RBAR and "Tuning" Updates
  - Both at http://www.sqlservercentral.com/

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Functions (Scalar)

And Why You Should Keep Up to Date with SQL Server Versions

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Problems with Scalar Functions

- Treated as a “Black Box”
- Executed as a series of independent statements
- Disables parallelism
- Unnecessary Complexity?

SQL Server 2019 introduces Scalar UDF Inlining – Provided You Have the Correct DB Compatibility Level

MSSQLTips

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Nested Views

Do You Really Know How Deep the Rabbit Hole Goes?

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Nested Views

- Unnecessary Complexity?
- How Many Tables Are Actually Involved via Joins?
- How Many Additional Predicate Operations?
- What Else Do You Not Know About?

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Recap

- Understanding Execution Plans
- Indexes
- Row-By-Agonizing Row (RBAR)
- Cursors and Loops
- Functions
- Nested Views

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved

Questions?

Copyright (c) 2006-2021 Edgewood Solutions, LLC All rights reserved