SQL Server: Optimizing Ad Hoc Statement Performance

Module 1: Introduction

Kimberly L. Tripp
Kimberly@SQLskills.com
http://www.SQLskills.com/blogs/Kimberly

pluralsight
hardcore developer training

Background

- **My first Pluralsight course was *SQL Server: Why Physical Database Design Matters***
- **This course builds on the base concepts that:**
  - Performance tuning is not just about writing good code
  - Performance tuning is not just about knowing your data
  - Performance tuning is not just about knowing your workload/priorities
- **Performance tuning takes multiple things:**
  - Knowing your data
  - Knowing your workload
  - Knowing how SQL Server works
- **This last one is the reason for my courses**
  - Give you the sometimes-incredibly-unintuitive “missing link” to why things happen the way that they do…

This Course

- **There are lots of different ways in which applications can submit a request for data to SQL Server**
- **This course will show you that not all methods work for all data requests**
- **The course will show you the different ways to ask for data as well as how SQL Server determines how to get to that data**
- **A point that I make in all of my courses:**
  - Truly scalable applications don't happen by accident
    - There is no platform that's perfect for every possible use case
    - There is no single method for submitting data requests that works well ALL THE TIME and requires minimal to no work
  - Can you successfully use a power tool without reading the manual?
    - You might not get the full power of the tool
    - You might do something the hard way or end up breaking the tool
    - You might not be successful at all

What Does Optimizing Ad Hoc Statement Performance NOT Mean?

- **This is not a course on rewriting Transact-SQL statements in different ways to get better performance**
  - However, sometimes that does help
- **This is not a course on indexing**
  - However, that sometimes helps A LOT
  - And, we will look at execution plans and some of them will use indexes
  - And, we will see a couple of cases where indexes change the way a statement is handled by SQL Server
  - We’re not diving deeply into index creation techniques and strategies
- **This is not a deep course on statistics**
  - However, these can also help A LOT
  - And, we will dive into these a bit so that we can understand where some of the plans/estimates come from when our statements have different types of search arguments

What Does Optimizing Ad Hoc Statement Performance Mean?

- **There are different ways to submit data requests to SQL Server**
- **Each of these ways has pros and cons around:**
  - Cost
    - Do we have to compile a plan or is one already available?
    - How did SQL Server arrive at a plan? Was it a good one? Is it good for all executions?
      - This is all tied to statistics and data estimations
      - Can we “sniff” the value?
      - Or, is the value completely “unknown” during optimization?
  - Complexity
    - Different methods sometimes require additional strategies to reduce exposure (e.g. SQL Injection)
    - Different methods sometimes require explicit type conversion to reduce plan cache pollution
- **All of these things affect the performance of the statement executed as well as future performance of the statement (*lots more to discuss here*)**

Why Is This Course Relevant

- **There are some very common scenarios that are not well understood**
- **I've watched arguments on web sites, blog posts, in-person…often based on assumptions and misunderstandings about what SQL Server is doing**
- **Microsoft SQL Server is a multi-purpose, relational database engine**
  - It can do anything
  - It can house any data
  - But, that doesn't mean that the defaults for EVERYTHING are good for EVERYONE and EVERY database
- **Most mistakes have relatively simple solutions that would have been trivial to implement early-on in the development process**
  - But once that application is in production, the changes might be anywhere from challenging to impossible (without great concessions such as downtime)

Course Focus and Structure (1)

- **This course expects basic knowledge of database terminology**
  - E.g. database, transaction log, backup
- **This course applies to all Editions of SQL Server 2005 onward, except where noted**
- **This course talks about the different ways that statements can be executed and how that affects caching, reuse, complexity, and ultimately performance**
  - This course is not a general course about writing Transact-SQL constructs
  - But, these techniques apply to ALL Transact-SQL statements (it's critical for your best overall performance and understanding of SQL Server)
- **All content and demos are shown on SQL Server 2012 but the behaviors and options are applicable to SQL Server 2005 and higher (unless, for example, a feature is listed as "new in SQL Server 2008")**

Course Focus and Structure (2)

- **Module 2: Statement execution methods**
- **Module 3: Estimates and selectivity**
- **Module 4: Statement caching**
- **Module 5: Plan cache pollution**
- **Module 6: Statement execution summary**

- **Consider watching all components of this course – in order**
  - It might not seem like some of the components are necessary but almost every module has cross references
- **I limited topics such as estimates, cost-based optimization, and statistics to only the relevant content, but they will help you understand later modules**