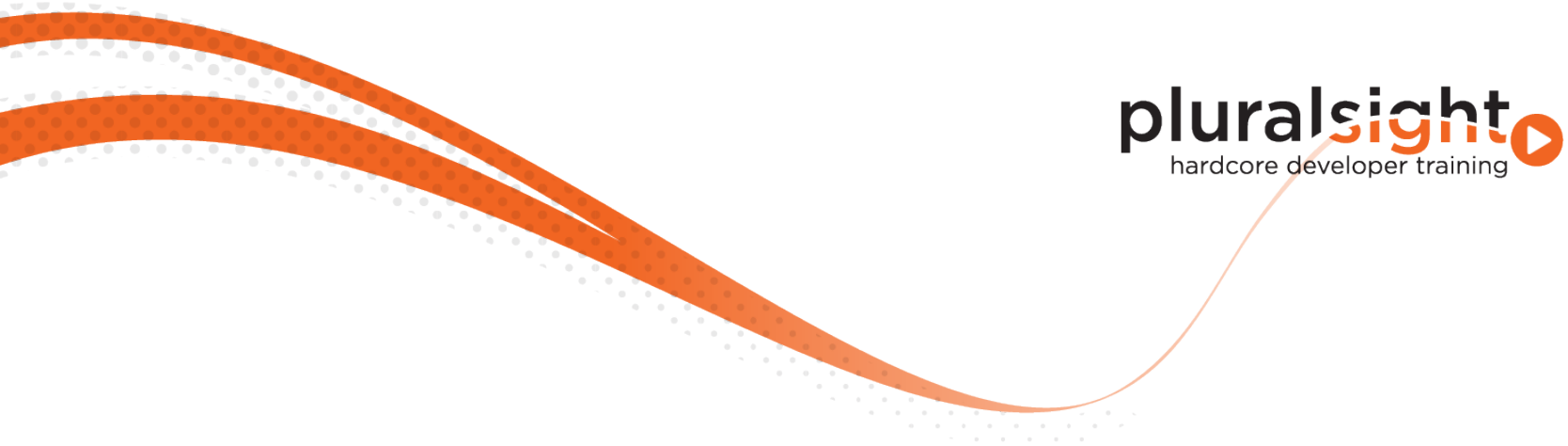
SQL Server: Benchmarking and Baselining

Module 4: Capturing Queries

Erin Stellato
Erin@SQLskills.com



pluralsight
hardcore developer training

Introduction

- **It is much easier to baseline performance in terms of numbers, rather than in terms of query performance**
- **However, the performance of individual queries is what ultimately uses the available resources we monitor so carefully**
- **It is possible to compare performance of a query or set of queries, but it is a two-part process**
  - Query capture
  - Query analysis
- **Query capture can be done using SQL Trace, Extended Events and DMVs**
  - DMVs will be covered in Module 5, Using DMVs
- **Query analysis involves SQL Server Profiler, XML and other third-party utilities**

SQL Trace

- **Low-level, server-side event implementation in SQL Server**
- **SQL Trace has existed since SQL Server 6.x**
  - Deprecated in SQL Server 2012
- **SQL Trace is captures information in real time, when enabled**
- **In order to capture queries, must set up a trace or use Profiler**
  - Traces can started created manually, or as part of a scheduled job
  - Profiler provides a graphical user interface for tracing and low-level analysis
- **Performance overhead can be introduced when capturing information, depending on trace configuration**
- **For more information, see the Pluralsight course, SQL Server: Collecting and Analyzing Trace Data**

Extended Events

- **Light-weight, server-side event implementation in SQL Server**
- **Extended Events are available in SQL Server 2008 and higher**
  - Graphical interface introduced in SQL Server 2012
- **Extended Events captures information in real-time, when enabled**
- **In order to capture queries, must set up an event session**
- **Event sessions can be started manually, or as part of a scheduled job**
- **Performance overhead can be introduced when capturing information, depending on event session configuration**
- **Analysis of Extended Events can be performed using the GUI or by programmatic interpretation of the event XML**
- **For more information, see the Pluralsight course, SQL Server: Introduction to Extended Events**

When to Use SQL Trace vs. Extended Events

- **SQL Trace and Extended Events can be both be used:**
  - If the information needed cannot be obtained through Dynamic Management Objects
  - As a proactive step when troubleshooting, rather than waiting for the issue to occur again
- **Use SQL Trace:**
  - When performance tuning code or during integration testing
  - If you need to capture a replay workload (Profiler, Distributed Replay)
  - If you want to use any third-party tool to analyze data
- **Use Extended Events:**
  - To capture performance counters previously only available in Performance Monitor (SQL Server 2012 only)

When to Use SQL Trace vs. Extended Events (2)

- **In general, Extended Events are more powerful than SQL Trace, but for benchmarking and baselining purposes, SQL Trace can suffice in many situations**
- **Prior to SQL Server 2012:**
  - Not all events from SQL Trace exist in Extended Events
  - The sql_text captured in Extended Events is not the same as Statement Text

Clear Trace

- **Free utility developed by Bill Graziano of ScaleSQL Consulting (http://bit.ly/bEUa0g)**
- **Ad hoc workloads are normalized**
  - Output includes total and average values for CPU, reads, writes, and duration for queries
- **Queries can be grouped by ApplicationName, LoginName, HostName, and/or TextData**
- **Allows you to:**
  - Determine what query, application, user, etc. is using the most resources
  - Determine how frequently a query or stored procedure is executed

ReadTrace

- **Free utility found within RML Utilities, which is developed by Microsoft and used by the SQL Server support team (http://bit.ly/99yHXZ)**
- **Ad hoc workloads are normalized**
  - Output includes total and average values for CPU, reads, writes, and duration for queries
- **Queries can be grouped by ApplicationName, LoginName, and/or TextData**
- **Use with Reporter tool to review data in graphical format**
- **Allows you to:**
  - Determine what query, application, user, etc. is using the most resources
  - Review individual queries and execution plans
  - Determine how frequently a query or stored procedure is executed
  - Compare two trace files

Benefits of Clear Trace vs. ReadTrace

- **ClearTrace**
  - No installation, configuration is straight-forward
  - GUI makes it easier to utilize and navigate initially
- **ReadTrace**
  - As part of RML Utilities, additional utilities are installed which may be of value (e.g. Ostress)
  - Command-line utility, usage is not as intuitive
  - Provides the ability to compare two traces
  - Gives more information overall
  - Provides graphical output which can be used for reporting

Replaying Workloads

- **Profiler**
  - Replay available for SQL Server 2005+
  - Replay can only be performed with one client
- **RML Ostress**
  - Replay available for SQL Server 2005+
  - Can replay a workload across multiple clients
- **Distributed Replay**
  - Added in SQL Server 2012
  - Replay available for SQL Server 2008+
  - Can replay a workload across multiple clients (maximum clients = 16 in Enterprise Edition)

Summary

- **Information about queries executing in SQL Server can be captured using Trace, Extended Events, or the DMVs**
  - For longer term trending, Trace and Extended Events provide the best method of collection
- **Analysis of query data often involves additional utilities**
- **The method to utilize depends on the problem you're trying to solve**
- **RML Utilities, Profiler Replay and Distributed Replay can be used to understand the impact of environment and code changes for a workload**
- **In the next module: Using DMVs**