Introduction

- This presentation is an introduction to the SQL Server Profiler tool.
  - Step-by-step guide to setting up a trace.
  - Identify server activities that can be monitored.
  - Learn to analyze traces.
Agenda

- SQL Profiler highlights
- SQL Profiler concepts
- Analyze slow-running queries
- Analyze deadlocks
- Use with Windows performance log data
- Use with Database Engine Tuning Advisor
- SQL Trace

SQL Profiler Highlights

- It is a graphical tool used to monitor a database or Analysis Service instance.
- It can be run from Start menu or SSMS.
- It can be used to
  - Debug database code to find a problem query.
  - Diagnose slow-running queries.
  - Monitor the performance of SQL Server.
  - Audit actions performed for security.
SQL Profiler Concepts

- Some basic Profiler terms are
  - Events: SQL Server actions captured, such as Start/End of a SQL statement.
  - Event Category: a group of related events, for e.g. Stored Procedures.
  - Event Class – Type of event together with Data Columns that can be traced, for e.g. SP:Completed.

- Data Columns: attributes that describe the event, for e.g. Duration.
- Filter: Criteria that defines the subset of event data to be collected, e.g. DBID = 4
- Template: a trace configuration that can be re-used. It can be user-defined and saved as .tdf
- Trace: Data captured for selected events, data columns and filters.
SQL Profiler Tasks

- First select an existing template or create a new one for future re-use.

SQL Profiler Tasks

- Create a trace after connecting to server with the option of saving results and time to stop trace at.
SQL Profiler Tasks

- Select Events and Data columns to trace. Click Show All to see columns.
  - Specify grouping of columns or the order in which you would like to display them.

SQL Profiler Tasks

- Filter events to capture only those that occur under specified circumstances like for a specific database if there are several on that server instance.
SQL Profiler Tasks

- Run the trace after it has been configured. You can pause, stop, clear the trace window, re-start the trace and search for a column value. You can open a saved trace and replay it too.

- Analyze the trace by stepping through the events. Select only those events and data columns that are relevant so that running the trace is not resource-intensive.
Analyze slow-running queries

- Run trace for a typical workload in a day. Check the queries with high duration.

Analyze Deadlocks

- SQL profiler can be used to identify deadlocks.
  - Events and data columns can be selected as below. Filter could be by Database Name.
Analyze Deadlocks

- Run trace for 24hrs, if server overwhelmed then just capture Deadlock Graph event or run trace programmatically. Once the query and objects are identified, future deadlocks can be prevented.

Other Useful Events

- The **Blocked Process Report** event helps identify long-running blocked process
  - It can be turned On/Off using the SP_CONFIGURE 'blocked process threshold' command.
Other Useful Events

- **Auto stats** event helps determine if statistics are not updated enough or updated too often.
- **SQL:StmtRecompiled** event identifies and troubleshoots excessive statement compilations.
- **Scan:Started** event fires for every table and index scan and helps determine excessive scanning.
- **Object and Security Audit** events are useful for investigating specific database access problems.

Correlate Perfmon Data

- SQL profiler trace can be used in conjunction with Performance Monitor to pinpoint specific queries bringing down server performance.
  - Create profiler trace with events and column as selected for the slow-running query.
Correlate Perfmon data

- Collect Performance Monitor activity data in a log file by creating a “Data Collector Set” and selecting the performance counters to track.

- Start and stop the Profiler trace and Perfmon Data Collector set at the same time and run them for a short interval during peak server loads.

---

Correlate Perfmon Data

- Open the Profiler .trc file and select File| Import Performance Data. Map the spikes in performance counters with SQL statements.
Profiler and DTA

- The Database Engine Tuning Advisor (DTA) can accept a SQL profiler trace output file or table.
- It analyses the statements that are run, their frequency and any indexes used.
- It provides indexing recommendations that it can implement or allow one to review and selectively apply at a later time.

SQL Profiler provides a default Tuning template that can be used directly or modified slightly to create the trace.
Profiler and DTA

- Open the DTA GUI from Start menu or from within SQL Profiler and select the database and the .trc file as workload. Click the Start Analysis button.
  - It gives us the recommendations and estimated percentage improvement.

SQL Trace

- System stored procs provided by TSQL to create and run a trace manually or from within an app without using SQL Profiler tool.
  - sp_trace_create
  - sp_trace_setevent
  - sp_trace_setfilter
  - sp_trace_setstatus
SQL Trace

- SQL trace is useful when there is a need to minimize overhead and have more control over the trace.
- One cannot organize columns or grouping with SQL Trace.
- The trace script can be generated from the Profiler GUI itself.
- Trace function, `fn_trace_gettable`, can be used to query the trace file.
SQL 2012 Extended Events

- Light weight performance monitoring system comprising of system stored procedures introduced in SQL2008.
- SQL2012 now offers a GUI to setup the extended event sessions easily.
  - Create a new Event Session
  - Select a Template
  - Pick Events to capture
  - Start Session. You can even watch Live Data.

SQL 2012 Extended Events

- Choose Global Fields or Actions common to all events.
- Specify the Event Filters.
- Select the Event Session Target file to store results in.
SQL 2012 Extended Events

Summary

- SQL Server Profiler is a powerful graphical tool to debug, troubleshoot and monitor database activities.
- Traces can be created, started, saved and replayed for selected events, data and filters.
- It can be used to analyze slow queries, deadlocks, blocking, server performance.
- It can be used in conjunction with Performance Monitor and Database engine Tuning Advisor.
Summary

- Server-side SQL traces can be created programmatically.
- SQL2012 offers a GUI to manage Extended Events that works as an alternative to SQL Profiler.
- Great resource – Mastering SQL Server Profiler by Brad McGehee

Questions?