

Module 11

Benchmarking and Baselining



Overview

- Benchmark vs. baseline
- Data collection methods and tools
 - This module will focus on the “how”
 - *The “what” (counters, statistics) has been woven in to the IO, CPU and Memory modules*
 - Performance Monitor
 - Collector Sets and Counter Logs
 - PAL Tool (Performance Analysis of Logs)
 - SQL Trace data
 - Interoperability with Performance Monitor
 - Distributed Replay Utility (DRU) introduction

Purpose of Benchmarking

What are the capabilities?

- Measure performance using a specific set of indicators to determine the performance level in a manner that can be compared to other systems, business requirements or previous benchmarks
- Determine the capacity limits of a system
 - Maximum number of concurrent connections
 - Maximum number of transactions/batches per second
 - Be able to forecast replacement/upgrade requirements before exceeding limits

Purpose of a Baseline

What's does "normal" look like?

- Provide a starting point for comparison of additional benchmarks over time
- Measure the impact of changes
 - Code and design changes
 - Increased workload
 - Hardware configuration changes
 - Upgrades to OS or SQL Server
 - Test the changes on another server before going to production and measure the impact
- Past baseline data is invaluable during a performance "crisis" and is also helpful in capacity planning
- *The tools and data collected for benchmark and baseline activities often overlap*

External Non-SQL Native Tools

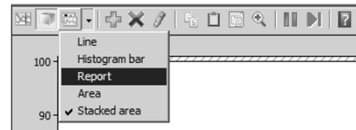
- SQLIO
- Iometer
- Quest Benchmark Factory
- Compuware
 - QA Run, QA Director, QA Load, Test Partner, TrackRecord
- Mercury
 - WinRunner, LoadRunner, TestDirector
- RML Utilities
 - Distributed Replay Utility in SQL Server 2012 inherits some of the features

Data Collection

- Performance Monitor
 - Single collection of performance counters for hardware, the Windows OS and SQL Server
 - Easy to use for trending over time
- DMV output
 - Wait statistics
 - File statistics
 - Buffer and Plan Cache usage
 - Index statistics (query optimization and storage engine)
- Trace Data
 - CPU, reads, writes, duration
 - Extended Events moved more into this space for SQL Server 2012

Using Performance Monitor

- Performance Monitor offers multiple views of the live performance data being collected
 - Line graph
 - Histogram bar graph
 - Report
 - Area, Stacked Area
- Live monitoring using the Performance Monitor user interface only provides the last 100 seconds of data
 - Useful for “right now” but not for historical trends
- Data Collector Sets provide the ability to collect performance counter data to files for long durations



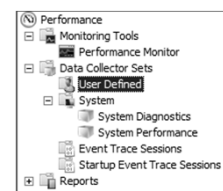
©SQLskills.com
SQLskills Immersion Event

7

SQL

Data Collector Sets

- User Defined
 - Custom data collections that can be created manually or from an existing template that collect user specified information from the system
- System
 - System Diagnostics
 - Built in template for diagnostic data collection
 - System Performance
 - Built in template for performance data collection



©SQLskills.com
SQLskills Immersion Event

8

SQL

Data Collector Sets

- Once created, data collector sets exist until they are deleted from the system
 - Define once and export to a template for reuse
 - Import a standard configuration of collector sets as a part of the server build process to facilitate baseline collection, diagnostic troubleshooting, and benchmarking
- Collector sets can be started from Performance Monitor manually or through the command line with logman
 - Logman exists in Windows Server 2003, 2008 and 2008 R2 and can be used to automate data collection in response to events and alerts

General OS Counters to Collect

- | | |
|--|---|
| <ul style="list-style-type: none"> ➤ Processor <ul style="list-style-type: none"> ➤ % Processor Time ➤ % Privileged Time ➤ System <ul style="list-style-type: none"> ➤ Processor Queue Length ➤ Memory <ul style="list-style-type: none"> ➤ Available Mbytes ➤ Pages/sec ➤ Paging File <ul style="list-style-type: none"> ➤ %Usage | <ul style="list-style-type: none"> ➤ PhysicalDisk <ul style="list-style-type: none"> ➤ Avg. Disk sec/Read ➤ Avg. Disk sec/Write ➤ Disk Reads/sec ➤ Disk Writes/sec ➤ Process (sqlservr.exe) <ul style="list-style-type: none"> ➤ % Processor Time ➤ % Privileged Time |
|--|---|

General SQL Counters to Collect

- SQL Server:Access Methods
 - Forwarded Records/sec
 - Full Scans/sec
 - Index Searches/sec
- SQL Server:Buffer Manager
 - *Buffer cache hit ratio?*
 - Free List Stalls/sec
 - Free Pages
 - Lazy Writes/sec
 - *Page Life Expectancy?*
 - Page Reads/sec
 - Page Writes/sec
- SQL Server:General Statistics
 - User Connections
- SQL Server:Locks
 - Lock Waits/sec
 - Number of Deadlocks/sec
- SQL Server:Memory Manager
 - Total Server Memory (KB)
 - Target Server Memory (KB)
- SQL Server:SQL Statistics
 - Batch Requests/sec
 - SQL Compilations/sec
 - SQL Re-Compilations/sec
- SQL Server:SQL Statistics
 - Batch Requests/sec
 - SQL Compilations/sec
 - SQL Re-Compilations/sec
- SQL Server:Latches
 - Latch Waits/sec

What does all this actually tell us?

Performance Analysis of Logs Tool

- Free tool on Codeplex designed to analyze Performance Monitor counter logs using industry standard thresholds
- Provides a built in template for SQL Server created by David Pless, a Premier Field Engineer at Microsoft
 - Template can be used to create a Data Collector Set template that can be imported into PerfMon
 - Details of the individual performance counters, what they mean, how they relate to each other, and the thresholds being tested are available in the user interface
 - The template can be customized to add additional counters or change thresholds if necessary

Demo

Data Collector Sets and Performance Analysis of Logs



SQL Trace

- Real-time insight into SQL Server activity
- Understand duration, frequency, and resource utilization of queries
- Gather a baseline or benchmark of system activity for consolidation or load projections
- Troubleshoot application errors or performance problems
- Auditing user activity
- Watch out for “observer overhead”

When to use SQL Trace

- It depends on the specific goals and requirements for the environment
 - Can you achieve the same objectives through Dynamic Management Views and Functions?
- In response to a error or alert
- Performance tuning development code or during integration testing
- Proactive tracing can be used to prevent having to wait for a problem to reoccur to see what caused it

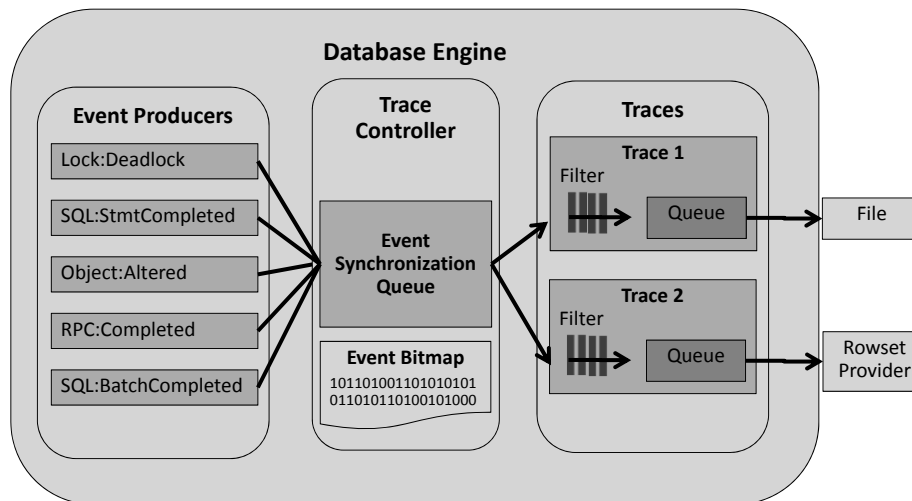
How SQL Trace Works

- The trace controller inside the database engine maintains a bitmap of events that are being collected by an active trace
- Event providers check if their event is active in the bitmap and if it is, provides one copy of the event data to the trace controller
- The trace controller queues the event data and provides the event data to all active traces collecting the event

How SQL Trace Works (2)

- The individual traces filter the event data removing any columns that are not needed, and discarding events not matching the trace filters
- The remaining event data is written to a file locally on the server, or buffered to the row-set provider for consumption by external applications like SMO and SQL Profiler

SQL Trace Architecture



Creating SQL Trace Sessions

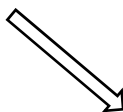
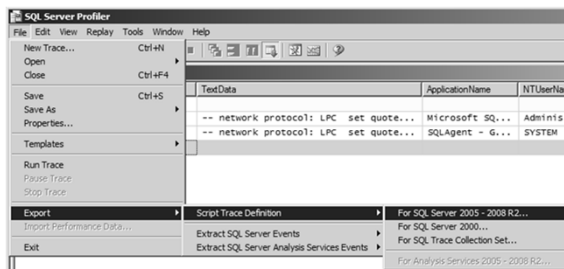
- **sp_trace_create**
 - Creates a trace with the provided configuration and returns the trace_id for the new trace
- **sp_trace_setevent**
 - Adds/removes an event or column to an existing trace
- **sp_trace_setfilter**
 - Applies a filter to an existing trace
- **sp_trace_setstatus**
 - Modifies the status of a trace (0-stop, 1-start, 2-delete)
- Changing a traces definition requires that the trace be in a stopped status (status=0)

©SQLskills.com
SQLskills Immersion Event

19

SQL

Building Trace Scripts with Profiler



```

7 1-- Create a Queue
8 declare @rc int
9 declare @TraceID int
10 declare @maxfilesize bigint
11 set @maxfilesize = 5
12
13 -- Please replace the text InsertFileNameHere, with an
14 -- appropriate
15 -- filename prefixed by a path, e.g., c:\MyFolder\MyTrace.
16 -- The .trc extension
17 -- will be appended to the filename automatically. If you are
18 -- writing from
19 -- remote server to local drive, please use UNC path and make
20 -- sure server has
21 -- write access to your network share
22
23 exec @rc = sp_trace_create @TraceID output, 0, N
24 'InsertFileNameHere', @maxfilesize, NULL
25 if (@rc != 0) goto error
26
27 -- Client side File and Table cannot be scripted
28
29 -- Set the events

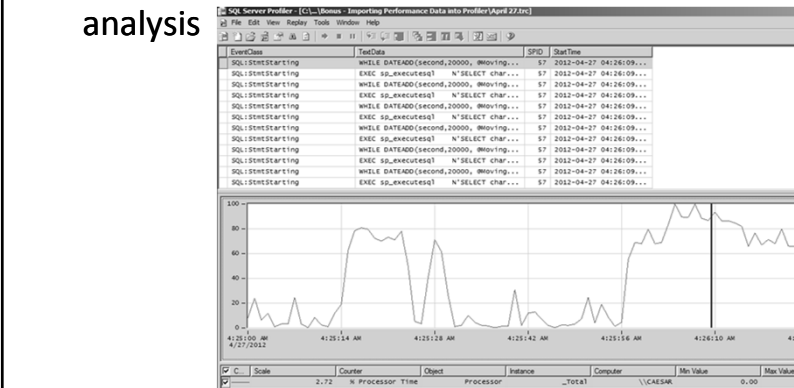
```

©SQLskills.com
SQLskills Immersion Event

SQL

Using Trace and PerfMon Data Together

- SQL Server Profiler can read trace file data and Import Performance Data captured during the same time interval as the trace to provide a single view for analysis



©SQLskills.com
SQLskills Immersion Event

21

Automating Trace Capture

- Create a trace script as a stored procedure on the server
 - If created in the master database the procedure can be marked for automatic execution at startup with `sp_procoption`
 - Another option is a SQL Server Agent Job with a schedule type of "start automatically when SQL Server Agent starts"
 - Or you can automate based on an event and then shut down after a specific period of time
- *Ensure that the filename is generated dynamically and maintains uniqueness of the trace will fail to start*

©SQLskills.com
SQLskills Immersion Event

22

Demo

Automated Profiling on Increased IO Waits



Tool Use-Cases

- Perfmon and PAL
 - Appropriate for use in both benchmarking and baselining
- SQL Trace \ Profiler
 - Useful if needing to capture a replay workload for benchmarking or testing before/after baselines prior to changes
- DRU, RML Utilities for Load Testing
 - More options than Profiler Replay
 - Can be used for before/after baselines and benchmarking
- SQLIO / IOMeter
 - Benchmarking and evaluating the effectiveness of varying configurations (for example, HBA queue depth, RAID configs, offsets)

SQL Server 2012

Distributed Replay Utility

- Capture a trace and replay from multiple computers
 - Profiler limited to replay from one computer
- Topology:
 - Distributed Replay controller
 - Orchestrates actions of clients (can be only one)
 - Distributed Replay clients
 - Multiple clients (<=16) work together to simulate workload
 - Distributed Replay administration tool
 - DReplay.exe used to talk to controller
 - Target server
- Settings
 - Synchronization mode (replay in order of original event – synchronized across clients)
 - Stress mode (drive load, and no synchronization across clients)
 - Can decrease “think time” and “connect time”

Review

- Benchmark vs. Baseline
- Data collection methods and tools
 - Focus on the “how”
 - Performance Monitor
 - Collector Sets and Counter Logs
 - PAL Tool
 - SQL Trace data
 - Interoperability with Performance Monitor
 - Replay Markup Language Utilities