

SQLintersection

Session SQL01

SQL01: Solving Complex Problems with Extended Events

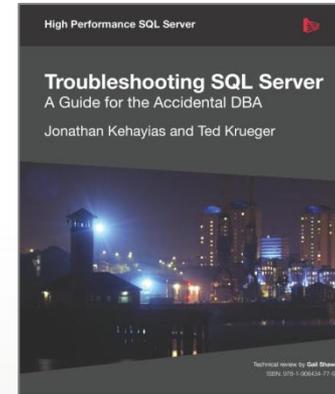
Jonathan Kehayias
Jonathan@SQLskills.com



Author/Instructor: Jonathan Kehayias



- **Consultant/Trainer/Speaker/Author**
- **Principal Consultant, [SQLskills.com](http://www.SQLskills.com)**
 - Email: Jonathan@SQLskills.com
 - Blog: <http://www.SQLskills.com/blogs/Jonathan>
 - Twitter: @SQLPoolBoy
- **Contributing Author: SQL Server 2008 Internals and Troubleshooting**
- **Microsoft Certified Master: SQL Server 2008**
- **SQL Server MVP since October 2008**
- **Author of Using Extended Events whitepaper on MSDN**
- **Developed Extended Events Manager SSMS Addin for SQL Server 2008 Open Source project on Codeplex**
- **Member of the Tampa SQL Server User Group and PASS**



- **Team of world-renowned SQL Server experts:**

- Erin Stellato (@ErinStellato)
- Glenn Berry (@GlennAlanBerry)
- Jonathan Kehayias (@SQLPoolBoy)
- Joe Sack (@JosephSack)
- Kimberly L. Tripp (@KimberlyLTripp)
- Paul S. Randal (@PaulRandal)

- **Instructor-led training: Immersion Events (US & UK)**

- **Online training:**  <http://pluralsight.com/>

- **Consulting: health checks, hardware, performance, upgrades, virtualization, design review, HA/DR, scalability**

- **Remote DBA: system monitoring and troubleshooting**

- **Conferences: SQLIntersection, PASS Summit**

- **Become a SQLskills Insider**

- <http://www.sqlskills.com/Insider>



SQLskills Immersion Events in Tampa

- **IE0: Accidental/Junior DBA, February 3-5**
- **IE1: Internals and Performance, February 3-7**
- **IE2: Performance Tuning, February 10-14**
- **IE3: High Availability and Disaster Recovery, February 17-21**
- **IEHW: SQL Server Hardware, February 6-7**
- **IEBI: Business Intelligence, February 10-14**
- **IEDEV: Developers, February 17-21**

SQLskills Pluralsight Courses

- 18 courses and counting...
- SQL Server 2012: Installation and Configuration
- SQL Server: Transact-SQL Basic Data Retrieval
- Developing and Deploying SQL Server ISV Applications
- SQL Server: Benchmarking and Baselineing
- SQL Server: Common Performance Issue Patterns
- SQL Server: Introduction to Extended Events
- SQL Server: Logging, Recovery, and the Transaction Log
- SQL Server: Performance Troubleshooting Using Wait Statistics
- SQL Server: Transactional Replication Fundamentals
- SQL Server: Virtualization
- SQL Server: Advanced Extended Events

Agenda

- **Extended Events basic concept review**
- **Example Scenarios**

What are Extended Events?

- Replacement for SQL Trace in SQL Server 2012 onwards
- Advanced event-collection infrastructure introduced in SQL Server 2008 and provided by SQLOS
- Highly-flexible implementation allows complex configurations for event collection that simplify problem identification with lower performance impacts to server workloads

Core Concepts: Events

- **Events correspond to well-known points in the code**
 - E.g. a T-SQL statement finished executing; a deadlock occurred
- **Events are stored as XML, as this is extensible**
- **Events deliver a basic payload of information**
 - Payload is defined by a schema of information
 - Events may contain optional (customizable) data elements that are only collected when specified
 - Events will always return all non-customizable data elements
- **Events are defined using the Event Tracing for Windows (ETW) model to allow integration with ETW**

Core Concepts: Actions

- **Actions only execute after predicate evaluation determines the event will fire**
- **Actions perform additional operations when an event fires**
 - E.g. collecting additional state data, or performing a memory dump
- **Actions execute synchronously on thread that fired the event.**
- **Any action may be used with any event, but state data may not be available at the point in code where an event fires**

Core Concepts: Targets

- **Event consumers**
 - Process single events or a buffer full of events
- **Both synchronous or asynchronous targets exist**
- **Basic targets collect raw event data**
 - Event file
 - Ring buffer
- **Aggregating targets aggregate data based on criteria**
 - Event bucketizer (histogram)
 - Event counter
 - Event pairing (matching events)
- **ETW target allows end-to-end tracing with Windows Kernel**

Core Concepts: Predicates

- **Predicates are Boolean expressions that define the conditions required for an event to actually fire**
- **Predicates support short-circuit evaluation**
 - First false evaluation prevents the event from firing
- **Predicates can use basic arithmetic operators, or textual comparators for more complex expressions**
 - E.g. Bitmask checking, greater than last max value, less than last min value, and divides evenly by int64 can perform event sampling not possible with basic Boolean expressions
- **Predicates can operate on global or event payload data**
- **Predicates can store state (this is pretty cool)**
 - E.g. count the number of times an event fires and only publish it every N times

Example Scenarios

- Tracking recompilations
- Execution timeouts
- Tempdb latch contention
- Lock escalation
- Problematic page splits
- Troubleshooting orphaned transactions
- Troubleshooting ASYNC_NETWORK_IO issues

Tracking Recompilations

Execution Timeouts

- Execution timeouts often occur when a client request exceeds the default .NET timeout of 30 seconds for the connection to SQL Server
- The statement level starting and completed events should both fire under normal query execution, but during a client timeout, the completed event will not be fired
- Pairing the events with the pair_matching target on the sqlserver.session_id and sqlserver.tsq_stack actions will simplify identification of timeout occurrences
 - The starting event for the timeout will not pair up to a completed event and will remain in the target when the timeout occurs
- Event session planning around recompiles must be made to filter the starting events fired under a recompile context to remove invalid unmatched pairs from the target

Tempdb Latch Contention

- **Latch contention on allocation bitmap pages in tempdb can significantly affect performance of SQL Server**
 - Page Free Space (PFS) and Shared Global Allocation Map (SGAM) are the bitmaps where contention can occur
 - Contention on these pages occurs when tracking page allocation and deallocation with many small temp tables
 - Increasing the number of files can reduce contention on these pages as round-robin allocation divides the allocations over the available files
- **The latch_suspend_end event tracks when latch waits end inside of SQL Server by database_id, file_id, and page_id**
 - Using a predicate with the divides_evenly_by_int64 predicator can track contention that occurs on tempdb allocation pages specifically
- **Bucketing the events produced with the histogram target simplifies identification of allocation bitmap contention inside of tempdb**

Lock Escalation

- Lock escalation inside the engine can result in excessive blocking or deadlocks when the object or partition locks are held for long durations
- Identifying the statement(s) that caused lock escalation to occur is one of the first steps in resolving problems that result from the higher granularity locking that results from the escalation
- In SQL Server 2008, the `sqlserver.lock_acquired` event can be used to identify object level locking (`resource_type = 5`) along with the `sqlserver.sp_statement_starting` or `sqlserver.sql_statement_starting` events and `TRACK_CAUSALITY` to identify the cause of escalated locks
- In SQL Server 2012, the `sqlserver.lock_escalation` event provides all necessary information for troubleshooting lock escalation occurrences without having to collect additional data

Problematic Page Splits

- **Mid-page splits cause data to be moved from one page to a newly allocated page inside of a database**
 - Results in fragmentation
 - Generates significantly more transaction log records for the split operation
- **The `sqlserver.page_split` event does not distinguish between normal page allocation 'splits' that occur for increasing keys, vs. mid-page splits that result in data movement and higher fragmentation**
- **The `sqlserver.transaction_log` event can be used to identify the occurrences of the `LOP_DELETE_SPLIT` log operation, to track problematic page splits**
 - The `database_id` and `alloc_unit_id` can be used to find the objects splitting the most frequently
 - Can be used with `fillfactor` adjustments to verify improvements after rebuilding problem indexes

Troubleshooting Orphaned Transactions

- **Identifying the root cause of a session with an orphaned transaction can be incredibly difficult since the problem manifests long after the statements that generated the transaction were actually executed**
 - Extended Events cannot prevent the problem from occurring, but can provide the necessary information to troubleshoot the problem after the next occurrence
- **The `database_transaction_begin` and `database_transaction_end` events can be used along with the `session_id` action and the `pair_matching` target to identify sessions that have begin events with no corresponding end event**
 - Leveraging the `tsql_frame` action in the data collection allows identifying the exact line in code where the orphaned transaction began

Troubleshooting ASYNC_NETWORK_IO issues

- **ASYNC_NETWORK_IO waits generally occur when a client application takes too long to process returned records before notifying SQL Server that the records have been received**
 - In Extended Events the wait_type is NETWORK_IO in the wait_types map
- **Tracking the wait_info event along with the client_app_name action using the histogram target can identify the application that is causing the majority of waits**
- **Tracking the wait_info event along with the host_name action using the histogram target can identify the server that is causing the majority of waits in distributed applications**
 - Could be an incorrect network configuration, for example duplex speed for the network adapter, or a remote server with a slow network link

Summary

- **More advanced troubleshooting can be accomplished using Extended Events than other methods available in SQL Server**
- **In some cases the ability to leverage Extended Events for advanced troubleshooting requires additional information about how SQL Server functions**
 - More information about the internals of SQL Server can be found in the additional courses on Pluralsight by SQLskills.com
- **Many more scenarios for using Extended Events exist than were covered in this module**
 - These are merely suggested scenarios where advanced troubleshooting can be performed using Extended Events

Questions?

*Don't forget to complete a session evaluation form
and drop it off at the conference registration desk.*

Session: SQL123

Thank you!

