

SQL Server: Deadlock Analysis and Prevention

Module 4: Collecting Deadlock Information

Jonathan M. Kehayias
Jonathan@SQLskills.com



Introduction

- **The first step to successfully troubleshoot a deadlock is to collect information about the deadlock from the Database Engine**
- **Multiple methods exist for collecting deadlock graphs in SQL Server 2005 and onwards**
 - The correct method depends on the specific requirements of the environment that you are working in
 - All methods will provide you with the necessary information to determine the root cause of the deadlock
- **In this module we'll cover:**
 - Trace flags
 - Trace and Profiler
 - Event Notifications/WMI
 - Extended Events

Trace Flags

- Trace Flags enable alternate "code paths" at key points inside the Database Engine, allowing additional code to execute
- Prior to SQL Server 2005, trace flags were the only method of collecting the necessary information for deadlock troubleshooting
- Trace flags must be explicitly enabled using DBCC TRACEON or through startup parameters
 - DBCC TRACEON requires the -1 trace flag option so all sessions are affected
 - E.g. DBCC TRACEON(1205, -1);
- Trace flags 1205 and 1222 provide process-level information about the tasks that participate in the deadlock
 - This deadlock information is written to the ERRORLOG file for the instance
- Trace flag 1204 provides deadlock graph node-level information and was the only method of getting deadlock information in SQL Server 2000

SQL Trace and Profiler

- Starting in SQL Server 2005, the Deadlock Graph trace event can be used in a server-side trace or with SQL Server Profiler to capture a full graph in XML format for deadlocks that occur
- The Deadlock Graph event XML contains all the information necessary to troubleshoot the cause of a deadlock
- Deadlock Graph events can be extracted from trace files, or Profiler captured data, into individual XDL files for analysis
 - The XDL format of the deadlock graph allows the graphical representation of the deadlock in SQL Server Profiler as well as in SQL Server Management Studio
- Reading the deadlock graph in XML form can often be faster for analysis than trying to interpret the graphical representation

Event Notifications

- Event notifications were added in SQL Server 2005 and allow specific Trace events to be captured using Service Broker for automated processing of the event data when the event occurs
- The DEADLOCK_GRAPH event provides the same information as the SQL Trace Deadlock Graph event
 - The event is entered in a Service Broker queue instead of being output to SQL Trace for consumption
- **Configuring Event Notifications requires:**
 - A queue to capture the event messages
 - A service to route the messages to the queue
 - A server-level Event Notification for the DEADLOCK_GRAPH event to capture the data and send it to the service
- An optional 'Activation Stored Procedure' can be created to automatically process the events as they are queued

Windows Management Instrumentation

- **Starting in SQL Server 2005, the Database Engine was instrumented to integrate with Windows Management Instrumentation (WMI) for specific events**
 - WMI events in SQL Server rely on Event Notifications through the msdb database natively
- **SQL Server Agent alerts were rewritten to be able to monitor for, and capture data about, WMI events being raised by the Database Engine**
 - Server names exceeding 14 characters do not work unless SQL Server 2005 Service Pack 2 with Cumulative Update 5 has been applied
- **Alerts can be created using the WMI query language (WQL) to query the specific WMI namespace for the event to be monitored**
- **A full example of WMI Alerts is available in Books Online (<http://bit.ly/SQLWMIAlert>)**

Extended Events

- Extended Events were introduced in SQL Server 2008 as a light-weight diagnostic data collection mechanism
- The `xml_deadlock_report` event fires when the Lock Monitor in SQL Server identifies a deadlock and raises error 1205
- **New XML format in Extended Events**
 - Supports multi-victim deadlock analysis
 - Incompatible with graphical display of deadlock graph in SSMS
 - Reduces redundant information that existed in the previous XML format
- **The RTM releases of SQL Server 2008 and 2008R2 contain a bug which causes the new XML format to be incorrectly formed**
 - SQL Server 2008 SP1+CU6 or higher, and SQL Server 2008R2 RTM+CU1 or higher, fix this bug (<http://support.microsoft.com/kb/978629>)
- The event is collected by default in the `system_health` event session from SQL Server 2008 onwards

Summary

- **Collecting deadlock information is the key to understanding the causes of deadlocks in SQL Server**
- **Multiple methods exist for capturing deadlock information and the appropriate method is determined by the environment requirements**
 - Event Notifications and WMI provide methods to automate a task in response to the event occurring
 - Trace Flags write to the ERRORLOG and Windows Application log and may be better for enterprise monitoring solutions that monitor this already
 - Third-party monitoring solutions may use Trace to capture deadlock graphs
- **Starting in SQL Server 2008, deadlock graph information is captured automatically by the system_health event session**
- **The next module will look at:**
 - Deadlock analysis