

## Re: URGENT deadlock question

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.server/2004-12/2149.html>

---

**From:** ChrisR (bla\_at\_noemail.com)

**Date:** 12/17/04

Date: Fri, 17 Dec 2004 11:38:30 -0800

Thanks for the quick response. However, now Im really confused.

When you use

- > *Profiler to view a trace, SQL Server dumps rowsets across the network to*
- > *the Profiler client, but there is no guarantee you'll see everything if*
- > *SQL Server doesn't feel like it can keep up with all the activity. A*
- > *server-side trace to a trace file guarantees you'll see everything. Just*
- > *make sure the trace file is created on a local server drive, not on a*
- > *network drive or you'll throttle server SQL Server and kill overall*
- > *performance.*

Will I get everything if I save to a table as well, or only a file?

Also, Ive always read that it would hurt performance to run Profiler on the local box. But if Im reading correctly, you are saying thats the only way to gauantee seeing all activities?

Do you know if I can get the needed info from the messages in the Error Log?

"David Gugick" <davidg-nospam@imceda.com> wrote in message  
news:ezkZQwG5EHA.3368@TK2MSFTNGP10.phx.gbl...

> *ChrisR wrote:*

- >> *I received 4 deadlocks in about 15 minutes. 3 of them occured at the*
- >> *same time. Im pretty sure that one was due to a large program that*
- >> *manipulates/ bcp's data in. That one is dealable. So I turned on*
- >> *Profiler. I have a deadlock template that I use with the following*
- >> *settings:*

>>

>> *Locks:*

>> *DeadLock*

>> *Deadlock Chain*

>>

>> *SP's:*

>> *RPC Completed*

>> *-- Starting*

>> *SP Completed*

>> *-- Starting*

>>

> > *TSQL:*  
> > *SQL: BatchCompleted*  
> > *-- BatchStarting*  
> > *SQL: Statement Completed*  
> > *SQL: Statement Starting*  
> >  
> > *According to BOL for DBCC TraceON(1204):*  
> >  
> > *"Trace flags remain enabled in the server until disabled by executing*  
> > *a DBCC TRACEOFF statement."*  
> > *I set DBCC TraceON(1204) a month or so ago so I assumed it would*  
> > *still be on. No reboots since then.*  
> >  
> > *So, Profiler is running and another Deadlock occurs 10 minutes later.*  
> > *I stop Profiler but there is no Deadlock that I can find. Throughgtout*  
> > *the Trace, there is a message that says:*  
> >  
> > *"Some trace events have not been reported to SQL Profiler because the*  
> > *server has reached its maximum amount of available memory for the*  
> > *process."*  
> >  
> > *So, one of two things is happening:*  
> >  
> > *1. The above message kept Profiler from reporting my Deadlock.*  
> > *2. 1204 wasnt on.*  
> >  
> > *More info:*  
> >  
> > *In my SQL Error Log, there are about 30 lines of info per deadlock.*  
> > *They have various Stored Procedure and Select statements for the*  
> > *viewing. Ive been told that Profiler MUST be running when a deadlock*  
> > *occurs in order to track it down.*  
> >  
> > *So my questions are:*  
> >  
> > *1. Does 1204 need to be set every time Profiler is started?*  
> > *2. Cant I figure out the cause of my deadlocks based on the info from*  
> > *the Error Log? Is it correct that the info from the Error Log Cant be*  
> > *used to track down a deadlock?*  
> >  
> > *Any other pointers on the matter would be great.*  
>  
> *You don't need the trace flag to see deadlock information in Profiler.*  
> *But it sounds like you really should be using a server-side trace and*  
> *not a Profiler started trace. Even using Profiler to start a server-side*  
> *trace actually creates two traces (on server-side and one for Profiler).*  
> *So start the server-side trace using T-SQL code you can generate from*  
> *Profiler using the Script Trace option on the File menu. When you use*  
> *Profiler to view a trace, SQL Server dumps rowsets across the network to*  
> *the Profiler client, but there is no guarantee you'll see everything if*  
> *SQL Server doesn't feel like it can keep up with all the activity. A*

microsoft.public.sqlserver.server: Re: URGENT deadlock question

- > *server-side trace to a trace file guarantees you'll see everything. Just*
- > *make sure the trace file is created on a local server drive, not on a*
- > *network drive or you'll throttle server SQL Server and kill overall*
- > *performance.*
- >
- > *Instead of the Completed events, use the Starting ones for deadlock*
- > *detection, since a statement may not complete if its terminated. Use SQL*
- > *StmtStarting, RPC Starting, and SP StmtStarting if you know these*
- > *problems are occurring from within stored procedures. Obviously add the*
- > *deadlock and deadlock chain events. If you can, filter the information*
- > *to limit data collection (if you know the users or applications*
- > *involved, for example).*
- >
- > *If you look for the SQL Statement that was executed immediately before*
- > *the deadlock chain event for that SPID, you should see the SQL Statement*
- > *that was involved in the deadlock. It's a good idea to gather all*
- > *statements, not just the deadlock victim to see all the SQL involved. SQL*
- > *Server 2000 does not report SPID information correctly in the deadlock*
- > *events, so always check the TextData column for these events for the*
- > *"real" SPID.*
- >
- >
- > --
- > *David Gugick*
- > *Imceda Software*
- > *www.imceda.com*
- >