How to troubleshoot SQL Server performance issues

Article ID: 298475 - View products that this article applies to.

This article was previously published under Q298475

Summary

To troubleshoot performance issues, you must complete a series of steps to isolate and determine the cause of the problem. Possible causes include:

- Blocking
- System resource contention
- Application design problems
- Queries or stored procedures that have long execution times

Identifying these causes is typically very time consuming, and you may spend several days evaluating the collected information. The complexity of the performance analysis is not specific to a particular database product or operating system. All applications are subject to performance constraints because of resource, design, or usage factors.

For more information, click the following article number to view the article in the Microsoft Knowledge Base:
224587 (http://support.microsoft.com/kb/224587/) How to troubleshoot application performance with SQL Server

To help identify and troubleshoot the problem, collect the following information at the same time and have the output readily available before you contact Microsoft Product Support Services (PSS):

- Blocker script output
- SQL Profiler trace log
- SQL Server Performance Monitor log

Note If you do not collect this information when the performance problem occurs, you may have to gather all the information again. This can delay the troubleshooting process.

After the problem occurs, collect the following information and have it available:

- sqldiag report
- Microsoft Windows NT System and Application Event logs

In most scenarios, PSS requires this information to understand the environment and the nature of the performance issue. If any part of this information is not available, the troubleshooting process may be prolonged and identifying the performance issue may be delayed.

Even if you are currently not experiencing any performance issues, Microsoft recommends that you implement this process. If a performance issue occurs, you can capture the required information as soon as possible. Additionally, if you capture a baseline SQL Profiler log, SQL Server Performance Monitor log, and blocker script when the application is performing as expected, you can use that information for comparison when the application does not perform as expected.

If you gather this information in a high traffic SQL Server environment, you may experience some performance degradation. However, you must have this information to identify the cause of the problem and for troubleshooting purposes. The SQL Profiler trace has the most impact on performance. If the performance is severely degraded, you can customize the SQL Profiler trace by reducing the types of events that it captures. Limiting the SQL Profiler trace should provide some improvement. If you have questions or problems setting up and collecting the information, contact PSS.

Blocker script output

The blocker script is critical for identifying blocking scenarios. However, you can use the output from the script to troubleshoot performance problems even when blocking is not an issue. This output also helps to determine if queries are waiting on resources, such as file I/O, or if transactions are not being committed or rolled back as expected. For more information about implementing the blocker script, click the following article numbers to view the articles in the Microsoft Knowledge Base:

251004 (http://support.microsoft.com/kb/251004/) How to monitor SQL Server 7.0 blocking

271509 (http://support.microsoft.com/kb/271509/) How to monitor SQL Server 2000 blocking

SQL Profiler trace log

The SQL Profiler trace captures the activity on the computer running SQL Server. You can use this information to identify slow running queries and non-optimal execution plans. Additionally, SQL Profiler documents the series of events that occur before the performance problem and helps to identify its cause.

To create and implement a SQL Profiler trace by using the GUI, see the "What to Monitor" section of the following Microsoft Knowledge Base article:
224587 (http://support.microsoft.com/kb/224587/) How to troubleshoot application performance with SQL Server

For more information about how to create and execute a SQL Profiler trace by using Transact-SQL commands, click the following article numbers to view the articles in the Microsoft Knowledge Base:

289742 (http://support.microsoft.com/kb/289742/) How to create a SQL Server 7.0 trace

283790 (http://support.microsoft.com/kb/283790/) How to create a SQL Server 2000 trace

For more information about how to monitor a SQL Profiler trace by using Transact-SQL commands, click the following article numbers to view the articles in the Microsoft Knowledge Base:

289279 (http://support.microsoft.com/kb/289279/) INF: How to Monitor SQL Server 7.0 traces

283786 (http://support.microsoft.com/kb/283786/) How to monitor SQL Server 2000 traces

Note If you have a SQL Profiler trace of a large workload, you can use the Index Tuning Wizard. The Index Tuning Wizard uses the SQL Server query optimizer to determine the optimal set of indexes for the specified queries. The Index Tuning Wizard is a very efficient tool to determine if the correct indexes exist in your database. By implementing the indexes that the wizard suggests, you may be able to increase the performance of your application.

For more information about how to use the Index Tuning Wizard, see the "Index Tuning Wizard" topic in SQL Server Books Online.

SQL Server Performance Monitor log

SQL Server is typically affected by the following bottlenecks:

- CPU
- Memory
- File I/O
- Locking, blocking, or deadlocking

You can use SQL Server Performance Monitor to identify how these potential bottlenecks may affect SQL Server. Additionally, you can use this log to identify when an external process is heavily using the computer running SQL Server and negatively impacting SQL Server performance.

Before you start SQL Server Performance Monitor, make sure that the disk counters are on. To do so, run **diskperf** from a command prompt. If the disk counters are not on, run **diskperf -y** and then restart the computer.

When you create a SQL Server Performance Monitor log, collect the following information:

- Paging file
- Process
- Processor
- All SQL Server counters
- Memory
- Threads
- Logical disk
- Physical disk
- System

Note The default interval of 15 seconds should enough time to monitor the server; however, for some timing issues, you may have to reduce the time interval for collecting data.

For more information about how to set up a SQL Server Performance Monitor log, click the following article numbers to view the articles in the Microsoft Knowledge Base:

150934 (http://support.microsoft.com/kb/150934/) How to create a Performance Monitor log for NT troubleshooting

248345 (http://support.microsoft.com/kb/248345/) How to create a log using System Monitor in Windows 2000

Note For more information about monitoring performance in SQL Server 2005, see the "Monitoring and Tuning for Performance" topic in SQL Server 2005 Books Online.

sqldiag utility

The sqldiag utility is provided with SQL Server. It collects valuable information about the configuration of the computer running SQL Server, the operating system, and the information that is reported to the SQL Server error logs. For information about how to use the sqldiag utility, see the "sqldiag Utility " topic in SQL Server Books Online.

For more information about how to run sqldiag on a clustered SQL Server, click the following article number to view the article in the Microsoft Knowledge Base:

233332 (http://support.microsoft.com/kb/233332/) How to run SQLDIAG on a clustered/virtual SQL Server

Note In SQL Server 2005, the SQLdiag utility has changed significantly. The command line arguments for this utility are not compatible with SQL Server 2000. This utility may be changed, and applications or scripts that rely on its command line arguments or behavior may not work correctly in future releases. For more information, see the "SQLdiag Utility" topic in SQL Server 2005 Books Online.

Microsoft Windows NT System and Application Event logs

You can use the Windows NT system and application event logs to identify issues that you cannot see in other data. These logs help provide a complete view of server activity and provide a more complete understanding of the environment.

Where to save these files

The following Microsoft File Exchange FTP server permits you to send and receive files to and from PSS engineers:
ftp://ftppss.microsoft.com (ftp://ftppss.microsoft.com)

For more information, read the instructions that are provided on the Microsoft File Exchange FTP Web site.

Bookmark: 9

More information

Rule software	Rule title	Rule description	Product versions against which the rule is evaluated
System Center Advisor	SQL Server has multiple active traces which can impact performance	System Center Advisor checks the presence of active traces other than the default trace that is currently capturing information. This alert is generated if there are additional traces running, if they are being saved in UNC path and if they are capturing expensive e () vents. Review the information in this article and take corrective actions appropriately.	SQL Server 2008 SQL Server 2008 R2 SQL Server 2012

Properties

Article ID: 298475 - Last Review: September 18, 2012 - Revision: 7.0

Applies to

- Microsoft SQL Server 2000 Standard Edition
- Microsoft SQL Server 2000 64-bit Edition
- Microsoft SQL Server 7.0 Standard Edition
- Microsoft SQL Server 2005 Standard Edition
- Microsoft SQL Server 2005 Developer Edition
- Microsoft SQL Server 2005 Enterprise Edition
- Microsoft SQL Server 2005 Express Edition
- Microsoft SQL Server 2005 Workgroup Edition

Keywords: kbhowtomaster kbinfo KB298475