

Identifying SQL Server Performance Problems

Asked 16 years, 4 months ago Modified 10 years ago Viewed 4k times



9



We're having sporadic, random query timeouts on our SQL Server 2005 cluster. I own a few apps that use it, so I'm helping out in the investigation. When watching the % CPU time in regular ol' Perfmon, you can certainly see it pegging out. However, SQL activity monitor only gives cumulative CPU and IO time used by a process, not what it's using right then, or over a specific timeframe. Perhaps I could use the profiler and run a trace, but this cluster is very heavily used and I'm afraid I'd be looking for a needle in a haystack. Am I barking up the wrong tree?

Does anyone have some good methods for tracking down expensive queries/processes in this environment?

[sql-server](#)

[performance](#)

[sql-server-2005](#)

Share

Improve this question

Follow

edited Nov 25, 2014 at 20:41



[Matt](#)

15.1k ● 27 ● 107 ● 157

asked Aug 18, 2008 at 14:39



Chris

4,070 ● 4 ● 48 ● 76

7 Answers

Sorted by:

Highest score (default)



12



This will give you the top 50 statements by average CPU time, check here for other scripts:

<http://www.microsoft.com/technet/scriptcenter/scripts/sql/sql2005/default.mspix?mfr=true>

```
SELECT TOP 50
    qs.total_worker_time/qs.execution_count as [Av
SUBSTRING(qt.text,qs.statement_start_offset/2,
    (case when qs.statement_end_offset = -1
    then len(convert(nvarchar(max), qt.text))
    else qs.statement_end_offset end -qs.state
as query_text,
    qt.dbid, dbname=db_name(qt.dbid),
    qt.objectid
FROM sys.dm_exec_query_stats qs
cross apply sys.dm_exec_sql_text(qs.sql_handle) as qt
ORDER BY
    [Avg CPU Time] DESC
```

Share Improve this answer

answered Aug 18, 2008 at 15:09

Follow



SQLMenace

135k ● 25 ● 211 ● 225



4

I've found the [Performance Dashboard Reports](#) to be very helpful. They are a set of custom RS reports supplied by Microsoft. You just have to run the installer on your client



PC and then run the setup.sql on the SQL Server instance.



After that, right click on a database (does not matter which one) in SSMS and goto Reports -> Custom Reports. Navigate to and select the performance_dashboard_main.rdl which is located at in the \Program Files\Microsoft SQL Server\90\Tools\PerformanceDashboard folder by default. You only need to do this once. After the first time, it will show up in the reports list.

The main dashboard view will show CPU utilization over time, among other things. You can refresh it occasionally. When you see a spike, just click on the bar in the graph to get the detail data behind it.

Share Improve this answer

Follow

answered Aug 18, 2008 at 16:34



Paul G

2,742 ● 4 ● 37 ● 54



3

We use Quest's [Spotlight](#) product. Obviously it's an investment in time and money so it might not help you out in the short term but if you are have a large SQL environment it's pretty useful.



Share Improve this answer

Follow

answered Aug 18, 2008 at 15:01



brendan

29.9k ● 20 ● 69 ● 109



3

As Yaakov says, run profiler for a few minutes under typical load and save the results to a table which will allow you to run queries against the results making it much easier to spot any resource hogging queries.



Share Improve this answer

answered Aug 18, 2008 at 15:13



Follow



[ChrisAnnODell](#) ♦

1,350 ● 15 ● 24



2

Profiler may seem like a "needle in a haystack" approach, but it may turn up something useful. Try running it for a couple of minutes while the databases are under typical load, and see if any queries stand out as taking way too much time or hogging resources in some way. While a situation like this could point to some general issue, it could also be related to some specific issue with one or two sites, which mess things up enough in certain circumstances to cause very poor performance across the board.



Share Improve this answer

answered Aug 18, 2008 at 14:47

Follow



[Yaakov Ellis](#)

41.4k ● 27 ● 134 ● 184



2

Run Profiler and filter for queries that take more than a certain number of reads. For the application I worked on, any non-reporting query that took more than 5000 reads deserved a second look. Your app may have a different threshold, but the idea is the same.



Share Improve this answer

answered Aug 18, 2008 at 15:24

Follow



[Curt Hagenlocher](#)

20.9k ● 8 ● 62 ● 50



2

[This utility](#) by Erland Sommarskog is awesomely useful.

It's a stored procedure you add to your database. Run it whenever you want to see what queries are active and get a good picture of locks, blocks, etc. I use it regularly when things seem gummed up.



Share Improve this answer

answered Aug 18, 2008 at 16:27

Follow



[Michael Haren](#)

108k ● 41 ● 171 ● 206