Three Ways to Identify Long Running Queries

Grant Fritchey, Product Evangelist, Red Gate Software
Goals

• Learn three methods for identifying long running queries
• Recognize the strengths and weaknesses of each method
• Know how to interpret the data gathered by each method
Grant Fritchey

Product Evangelist for Red Gate Software
Email: grant@scarydba.com
Twitter: @gfritchey
Blogs: scarydba.com
simple-talk.com/community/blogs/scary
Agenda

- Why Tune Queries
- Which Queries To Tune
- Gathering Data About Queries
- Dynamic Management Objects
- Server-side Trace
- Extended Events
## Why Tune Queries?

<table>
<thead>
<tr>
<th>Reason</th>
<th>%</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU power saving</td>
<td>2%</td>
<td>6</td>
</tr>
<tr>
<td>Other hardware or OS issue</td>
<td>2%</td>
<td>7</td>
</tr>
<tr>
<td>Virtualization</td>
<td>2%</td>
<td>7</td>
</tr>
<tr>
<td>SQL Server/database configuration</td>
<td>3%</td>
<td>10</td>
</tr>
<tr>
<td>Out-of-date/missing statistics</td>
<td>9%</td>
<td>31</td>
</tr>
<tr>
<td>Database/table structure/schema design</td>
<td>10%</td>
<td>38</td>
</tr>
<tr>
<td>Application code</td>
<td>12%</td>
<td>43</td>
</tr>
<tr>
<td>I/O subsystem problem</td>
<td>16%</td>
<td>60</td>
</tr>
<tr>
<td>Poor indexing strategy</td>
<td>19%</td>
<td>68</td>
</tr>
<tr>
<td>T-SQL code</td>
<td>26%</td>
<td>94</td>
</tr>
</tbody>
</table>

Which Queries To Tune?

- The one people complain about
- Whichever runs longest
- One using the most resources
- The bosses favorite report query
- ?
Which Queries To Tune?

Tune from knowledge

- Knowledge of the business
- Knowledge of the application
- Knowledge of the queries

To understand queries, you must gather data
Gathering Data About Queries

Important Data
- Frequency of execution
- Run duration
- CPU used
- I/O
- Date and time

Secondary Data
- Parameters passed
- Application calling
- User calling
- Execution plan

Tools used
- Dynamic Management Objects
- Server-side Trace
- Extended Events
Dynamic Management Objects

Strengths

- Immediate
- It’s T-SQL!
- Powerful because of the unique combinations of information that can be put together all at once

Weaknesses

- Dependent on the cache
- Or, dependent on currently executing sessions
- No parameters
- No user information
- No runtime information
- It’s T-SQL
Dynamic Management Objects

Reading the results

- It’s T-SQL
- Some understanding of the various hooks is required
  - Get a copy of “Performance Tuning with SQL Server Dynamic Management Views” by Louis Davidson & Tim Ford
Dynamic Management Objects

Demo
Server-side Trace

NOTE: Not Profiler

Strengths

- Ubiquitous
- GUI for setting up the trace
- GUI for reading output
- Low cost
- Works with Distributed Replay

Weaknesses

- Difficult to set up
- Troublesome to maintain
- Low cost != no cost
Server-side Trace

Reading the results

- Open results in Profiler
  - Do not run Profiler against production
- Load results into table
- Use RML Utilities
- Use 3rd Party products
Server-Side Trace

Demo
Extended Events

Strengths
- GUI for setting up (in 2012)
- GUI for reading output (in 2012)
- Easy to maintain
- Very easy to create
- Extremely low cost

Weaknesses
- Only in 2008 or better
- No GUI until 2012
- Does not work with Distributed Replay
Extended Events

Reading the results
- Open in Data Explorer
- Load into a table
Extended Events

Demo
Goals

• Learn three methods for identifying long running queries
• Recognize the strengths and weaknesses of each method
• Know how to interpret the data gathered by each method
Resources
Scarydba.com/resources
Query Performance Tuning: Start to Finish
  - Full day pre-con at PASS Summit 2012
SQL Server 2012 Query Performance Tuning
Grant Fritchey

Product Evangelist for Red Gate Software
Email: grant@scarydba.com
Twitter: @gfritchey
Blogs: scarydba.com
simple-talk.com/community/blogs/scary
Questions?

How would you...?

What happens when...?

Why does...?

When do I...?
Thank You for Attending