

Performance Counters

BY SRIKANTH BONTHALA

Performance Counters

- ▶ Performance Counters allows us a method to measure current performance, as well as performance over time.
- ▶ Identifying the metrics we like to use to measure SQL Server performance and collecting them over time gives us a quick and easy way to identify SQL Server problems, as well as to graph performance trend over time.
- ▶ There are many performance Counters available related to SQLServer and had listed some of important Counters among them.

1. SQL Server: Buffer Manager: Buffer cache hit ratio

- ▶ The buffer cache hit ratio counter represents how often SQL Server is able to find data pages in its buffer cache when a query needs a data page.
- ▶ The higher this number the better, because it means SQL Server was able to get data for queries out of memory instead of reading from disk. You want this number to be as close to 100 as possible.
- ▶ Having this counter at 100 means that 100% of the time SQL Server has found the needed data pages in memory. A low buffer cache hit ratio could indicate a memory problem.

2. SQL Server: Buffer Manager: Page life expectancy

- ▶ The page life expectancy counter measures how long pages stay in the buffer cache in seconds. The longer a page stays in memory, the more likely SQL Server will not need to read from disk to resolve a query.
- ▶ We should watch this counter over time to determine a baseline for what is normal in your database environment. Some say anything below 300 (or 5 minutes) means you might need additional memory.

3. SQL Server: SQL Statistics: Batch Requests/Sec

- ▶ Batch Requests/Sec measures the number of batches SQL Server is receiving per second.
- ▶ This counter is a good indicator of how much activity is being processed by SQL Server box. The higher the number, the more queries are being executed on box.
- ▶ We can review this counter over time to determine a baseline number for our environment.

4. SQL Server: SQL Statistics: SQL Compilations/Sec

- ▶ The SQL Compilations/Sec measure the number of times SQL Server compiles an execution plan per second.
- ▶ Compiling an execution plan is a resource-intensive operation. Compilations/Sec should be compared with the number of Batch Requests/Sec to get an indication of whether or not complications might be hurting your performance.
- ▶ To do that, divide the number of batch requests by the number of compiles per second to give us a ratio of the number of batches executed per compile.
- ▶ Ideally we want to have one compile per every 10 batch requests.

5. SQL Server: SQL Statistics: SQL Re-Compilations/Sec

- ▶ When the execution plan is invalidated due to some significant event, SQL Server will re-compile it.
- ▶ The Re-compilations/Sec counter measures the number of time a re-compile event was triggered per second.
- ▶ Re-compiles, like compiles, are expensive operations so we want to minimize the number of re-compiles. Ideally we want to keep this counter less than 10% of the number of Compilations/Sec.

6. SQL Server: General Statistics: User Connections

- ▶ The user connections counter identifies the number of different users that are connected to SQL Server at the time the sample was taken. We need to watch this counter over time to understand our baseline user connection numbers.
- ▶ Once we have some idea of high and low water marks during normal usage of our system, we can then look for times when this counter exceeds the high and low marks.

7. SQL Server: Locks: Lock Waits / Sec: _Total

- ▶ In order for SQL Server to manage concurrent users on the system, SQL Server needs to lock resources from time to time.
- ▶ The lock waits per second counter tracks the number of times per second that SQL Server is not able to retain a lock right away for a resource.
- ▶ Ideally we don't want any request to wait for a lock.
- ▶ Therefore we want to keep this counter at zero, or close to zero at all times.

8. SQL Server: Access Methods: Page Splits / Sec

- ▶ This counter measures the number of times SQL Server had to split a page when updating or inserting data per second.
- ▶ Page splits are expensive, and cause our table to perform more poorly due to fragmentation.
- ▶ Therefore, the fewer page splits we have the better our system will perform.
- ▶ Ideally this counter should be less than 20% of the batch requests per second.

9. SQL Server: General Statistic: Processes Block

- ▶ The processes blocked counter identifies the number of blocked processes.
- ▶ When one process is blocking another process, the blocked process cannot move forward with its execution plan until the resource that is causing it to wait is freed up.
- ▶ Ideally we don't want to see any blocked processes. When processes are being blocked we should investigate.

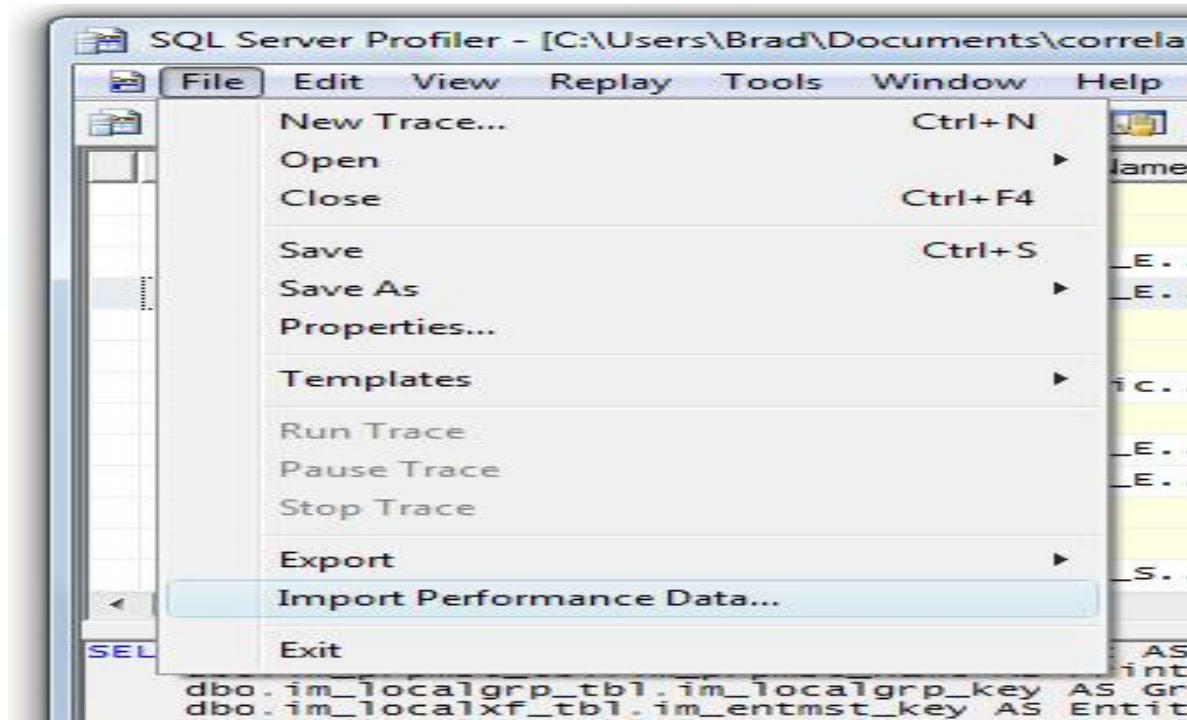
10. SQL Server: Buffer Manager: Checkpoint Pages /Sec

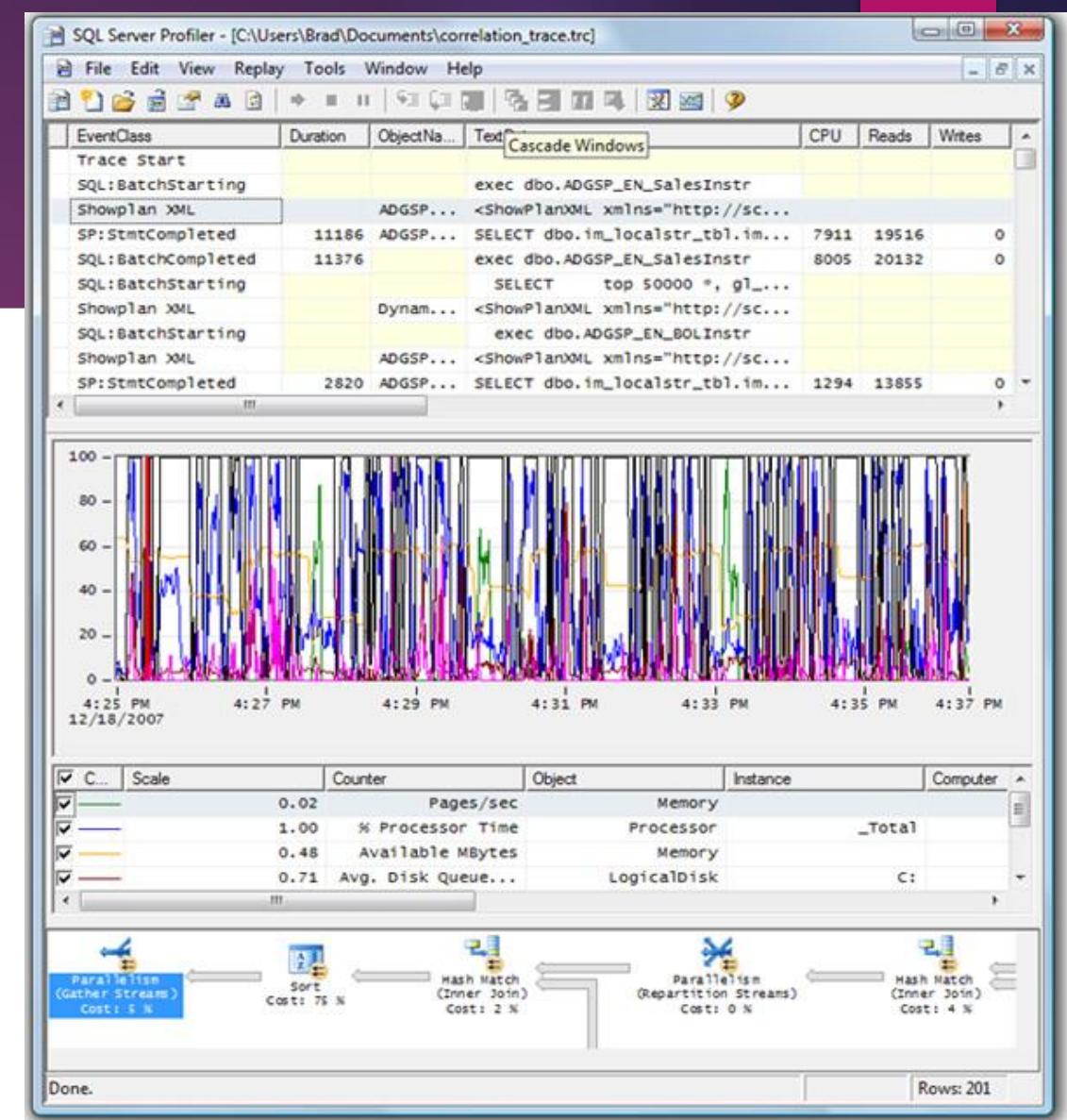
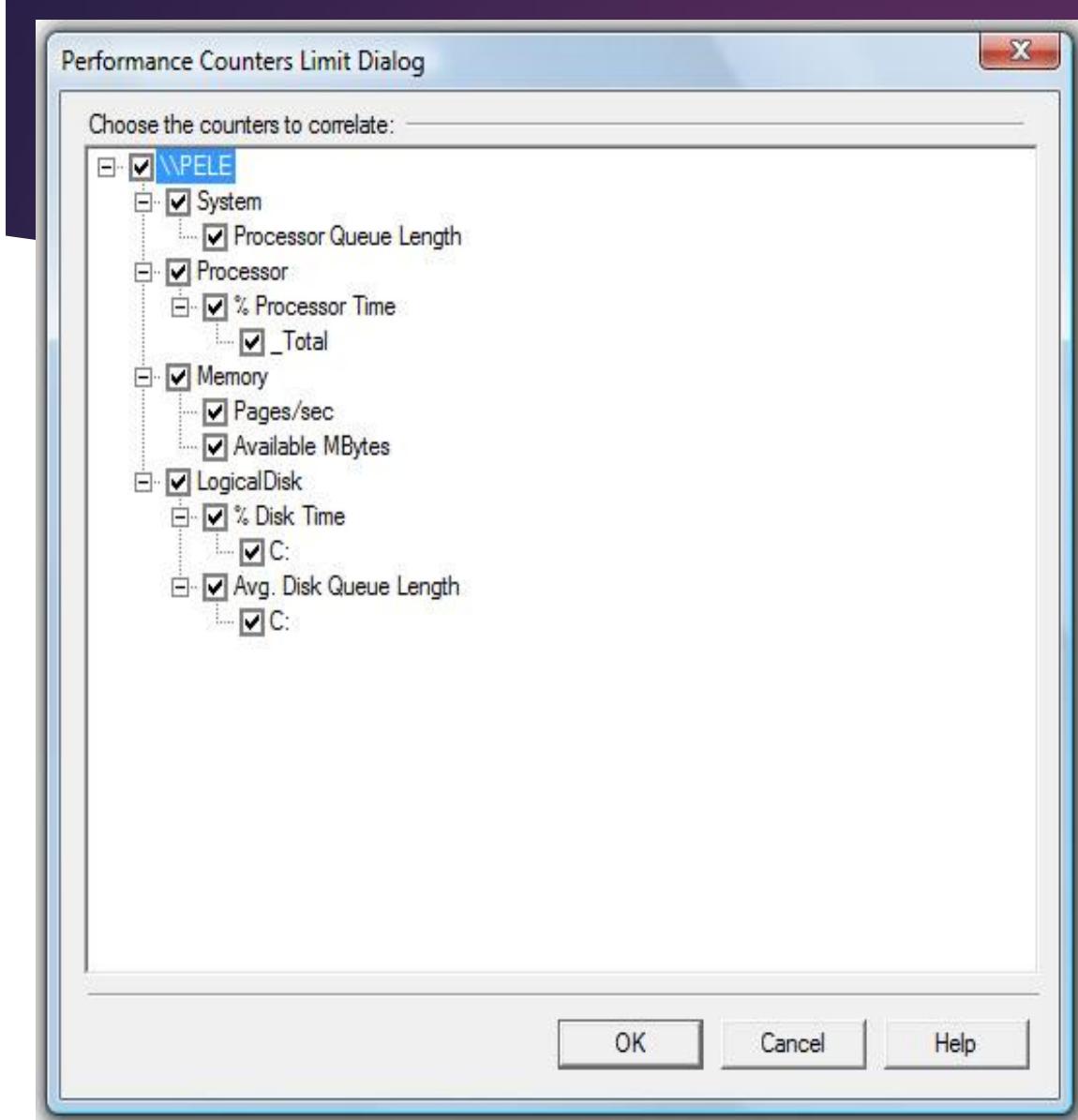
- ▶ The checkpoint pages per second counter measures the number of pages written to disk by a checkpoint operation.
- ▶ We should watch this counter over time to establish a baseline for our systems.
- ▶ Once a baseline value has been established we can watch this value to see if it is climbing.
- ▶ If this counter is climbing, it might mean we are running into memory pressures that are causing dirty pages to be flushed to disk more frequently than normal.

Correlating SQL Server Profiler with Performance Monitor

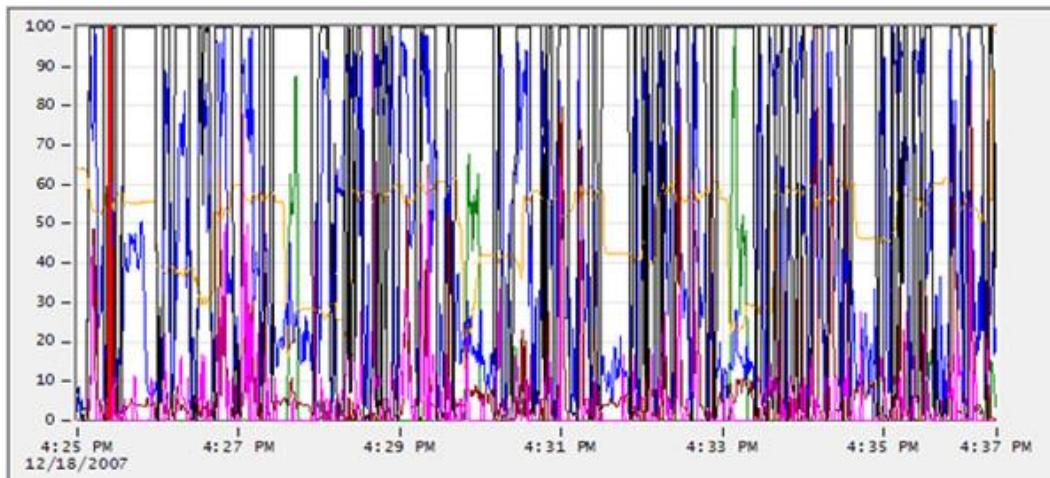
- ▶ Both Performance Monitor and SQL Server Profiler provide valuable information about servers.
- ▶ However, if we combine the data from the two sources, then this correlated information can help find some of the noticeable bugs.
- ▶ With SQL Server 2005 Profiler, we can import Performance Monitor log data and compare it directly with Profiler activity.
- ▶ For example, If we see a spike in CPU utilization, we can identify which statement or statements were running at the same time, and diagnose potential problems.

- ▶ From the main menu of Profiler, select File | Import Performance Data

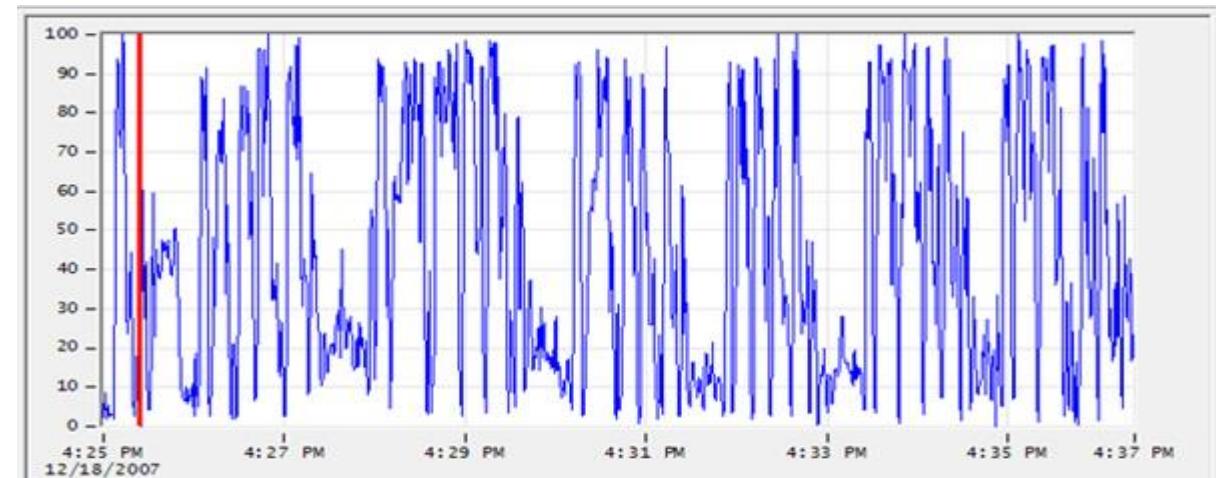




How to Analyze Correlated Data



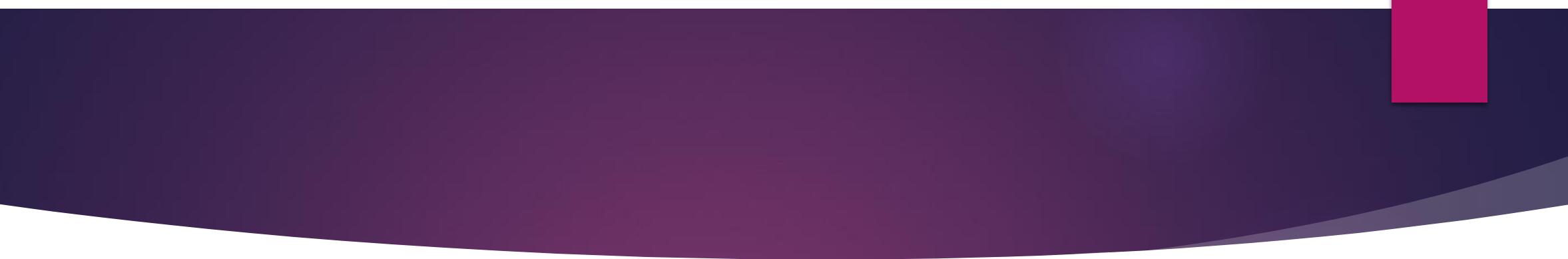
C...	Scale	Counter	Object	Instance	Computer	Min Value	Max Value	Avg Value
<input checked="" type="checkbox"/>	0.02	Pages/sec	Memory		\PELE	0.00	4902.84	
<input checked="" type="checkbox"/>	1.00	% Processor Time	Processor	_Total	\PELE	0.00	100.00	
<input checked="" type="checkbox"/>	0.48	Available MBytes	Memory		\PELE	27.00	207.00	
<input checked="" type="checkbox"/>	0.71	Avg. Disk Queue...	LogicalDisk	C:	\PELE	0.00	141.56	
<input checked="" type="checkbox"/>	5.56	Processor Queue...	System		\PELE	0.00	18.00	
<input checked="" type="checkbox"/>	1.00	% Disk Time	LogicalDisk	C:	\PELE	0.00	100.00	



C...	Scale	Counter	Object	Instance	Computer	Min Value	Max Value	Avg Value
<input checked="" type="checkbox"/>	0.02	Pages/sec	Memory		\PELE	0.00	4902.84	
<input checked="" type="checkbox"/>	1.00	% Processor Time	Processor	_Total	\PELE	0.00	100.00	
<input type="checkbox"/>	0.48	Available MBytes	Memory		\PELE	27.00	207.00	
<input type="checkbox"/>	0.71	Avg. Disk Queue...	LogicalDisk	C:	\PELE	0.00	141.56	
<input type="checkbox"/>	5.56	Processor Queue...	System		\PELE	0.00	18.00	
<input type="checkbox"/>	1.00	% Disk Time	LogicalDisk	C:	\PELE	0.00	100.00	

References

- ▶ <https://www.simple-talk.com/sql/database-administration/correlating-sql-server-profiler-with-performance-monitor/>



Thank You