Top PerfMon Counters for Analyzing SQL Server Performance Issues

Jeffry A. Schwartz

November 13, 2008

SQLRx® Webinar Series

jeffrys@isi85.com

Microsoft GOLD CERTIFIED Partner

2008 DATA MANAGEMENT SOLUTIONS PARTNER OF THE YEAR-WINNER Database Management

SQLRx.com

Introduction

- **Analysts often look in the wrong places to solve performance problems because**
  - Users provide vague descriptions regarding poor performance
  - Misleading or misinterpreted published information
  - Inadequate training involving system and database performance
  - Database automatically blamed regardless of actual cause
- **Analysts must possess techniques for accurately determining**
  - Causes of poor performance
    - Which hardware or software components are in actually in trouble
    - Using Windows Performance Monitor to identify major problem areas
  - Which queries are most troublesome, if **any**
  - How to develop appropriate solutions

Introduction

- **Today's is first of several performance-oriented sessions**
  - Provide performance metrics and analysis techniques to expedite analyses
  - Hardware-related and preliminary SQL Server performance analysis discussed today
  - Described information and techniques applicable to Windows 2000/2003/2008 and SQL Server 2000/2005/2008
  - More in-depth measurements and techniques presented in future sessions

Introduction

- **Many analysts misled by metrics that are useless, incomplete or whose importance is overly inflated**
  - Processor Queue Length
  - % Disk Time
  - Disk Queue Length
  - Buffer Cache Hit Ratio (SQL)
- **Unaware of invaluable metrics**
  - % User Time (Processor)
  - % Idle Time (Disk)
  - Avg. Disk sec/Transfer
  - Available Bytes (Memory)
  - Page Life Expectancy (SQL)
  - Page Reads/sec (Memory and SQL)

Query Side-Effects

- **Poorly designed SQL queries can cause a system to *appear* to be out of**
  - Processor
  - Memory
  - Disk
- **Because they can perform excessive**
  - Processor work while churning through memory-resident data buffers
  - Physical I/Os when data is not memory-resident that ultimately exhaust physical memory
- **Sorts often create all of the above**

Why Use Windows Performance Monitor?

- **Essential to focus any kind of analysis**
- **Remember SQL Server is only an application that runs under Windows**
  - If Windows does not perform well, neither will SQL Server
  - SQL Server can **make** Windows perform poorly if improperly configured
- **SQL Server metrics independent of hardware**
  - Most imply hardware performance issues, but cannot definitively isolate hardware components
- **Can gather performance data over time**

Analysis Methodology

- **Use Windows Performance Monitor, a.k.a. PerfMon, to determine**
  - Times when problems occur along with their durations
  - Which hardware components are involved
  - Many small queries or a few large resource-intensive queries?
- **Retain and analyze performance data using tool like Excel**
- **Correlate PerfMon data with other user-experience, business, or computer data**

PerfMon

- **Capture PerfMon data continuously to log file**
  - **All** bad periods will be captured, even the unreported ones
  - Dangerous to assume you **know** all the bad periods and their characteristics
- **Logging to a file**
  - **Extremely** low overhead unless process or thread objects captured
  - Process or thread objects generally unnecessary on SQL Server machines
- **Information logged primarily in two ways**
  - Binary
  - Comma-separated
- **On x64, must use version that matches SQL Server**
  - If using 32-bit SQL Server on x64, must use 32-bit collectors

PerfMon

- **Performance data buffers not locked**
  - Minimizes impact of performance data gathering on Windows performance
  - Values not always synchronized with each other, even for the same object
  - Reconciling multiple counters with each other may be difficult, if not impossible
- **Obtainable from local or remote machine**
  - Local collection can increase system overhead
    - Use **binary** format when captured on local machine to reduce overhead
  - Remote collection (preferred) can increase network overhead
    - Almost **never** an issue
    - Can use CSV format to simplify analyses

PerfMon

- **Data collection frequency**
  - **Do not** use GUI default of **one second** for **any** monitoring
    - Places undue pressure on system
    - Displays misleading values caused by volatility
    - Minimize GUI usage unless small # of counters and low collection frequency
  - 30 seconds usually sufficient for performance problems
  - 1 to 5 minutes usually sufficient for trending/capacity planning
- **GUI default settings can be changed and saved so proper behavior automatic**
  - Update frequency
  - Selected objects, instances, and counters

Performance Data Analysis

PerfMon

- Initially not intended as primary performance data analysis tool
- Evolved into just that
- Analysis typically **VERY** slow and laborious, especially with large files
- No really effective way to export substantial amounts of raw data via GUI

Most analysts familiar with Excel

- 2003 limited to 256 columns so smaller, i.e., 255 item counter sets must be used
- 2007 virtually unlimited, so full counter sets can be used

PerfMon Analysis Considerations

CSV format

- Easier to use outside of PerfMon, if # of counters < 256 or XL 2007
- Records can be impossibly long for spreadsheet programs
- Imposes **10x** more overhead on **collection** machine (**NOT** target machine) than if binary used

Binary format

- Usually **significantly** larger than CSV format
- Cumbersome because primarily usable only with PerfMon
- Required if process data gathered
- Convertible using Relog program shipped with 2003
  - Converted files **MUCH** smaller, but **NO** data lost in conversion

Relog

- **Converts NT 4 and Windows 2000/2003/2008 logs for easier manipulation outside PerfMon**
  - Any format → CSV, tab-delimited, SQL Server tables
- **Use 2003 version instead of 2000, XP, or Vista**
  - 2003 version has fewest problems
- **Runs properly on non-2003 OS when 2003 pdh.dll located in same directory as executable**
  - XP pdh.dll ≠ 2003 pdh.dll
  - Vista pdh.dll ≠ 2003 pdh.dll
- **http://www.mssqltips.com/tip.asp?tip=1607**

PerfMon Analysis Procedures

- If PerfMon data file in binary format, convert using relog
- Import CSV into appropriate version of Excel
- Add formulas for missing Disk data
  - Discussed later in this presentation
- Use Excel's conditional formatting to highlight warning, danger, and extreme danger conditions
- Graph important entities against each other to create visual correlations

Windows Performance Counters

- **Literally hundreds of unique counters**
  - Potentially thousands of instance-counter combinations
- **Imperative system categories to collect**
  - Processor
  - Memory
  - Physical Disk
  - Logical Disk (usually present)
  - Network I/O
  - SQL Server – all objects and instances

SQL Server Performance Counters

- **One set per SQL Server instance**
- **Sometimes will not appear in PerfMon when instance stopped**
- **Sometimes must be rebuilt because registry becomes corrupt**
  - Seems to most often occur on SQL Server 2000 clustered environments
  - Occurs far less often on SQL Server 2005

Which Performance Counters Useful?

- **Processor**
  - % Processor Time
  - % Privileged Time
  - % User Time
  - % Interrupt Time
  - % DPC Time (Deferred Procedure Calls)
- **System**
  - Context Switches/sec

Which Performance Counters Useful?

Physical Disk

- Avg. Disk sec/Transfer
  - Should be 0.020 seconds (20 ms) at **most** unless I/O size huge
- % Idle Time
  - Once this reaches zero, I/O rate cannot be increased
  - Performance usually degrades as it approaches zero
- Disk Transfers/sec, Disk Bytes/sec
  - Beware of disk specs because they usually cite **very large** I/Os
- Read and Write-specific counters also valuable, especially when read/write performance disparity exists or using RAID 5

Logical Disk

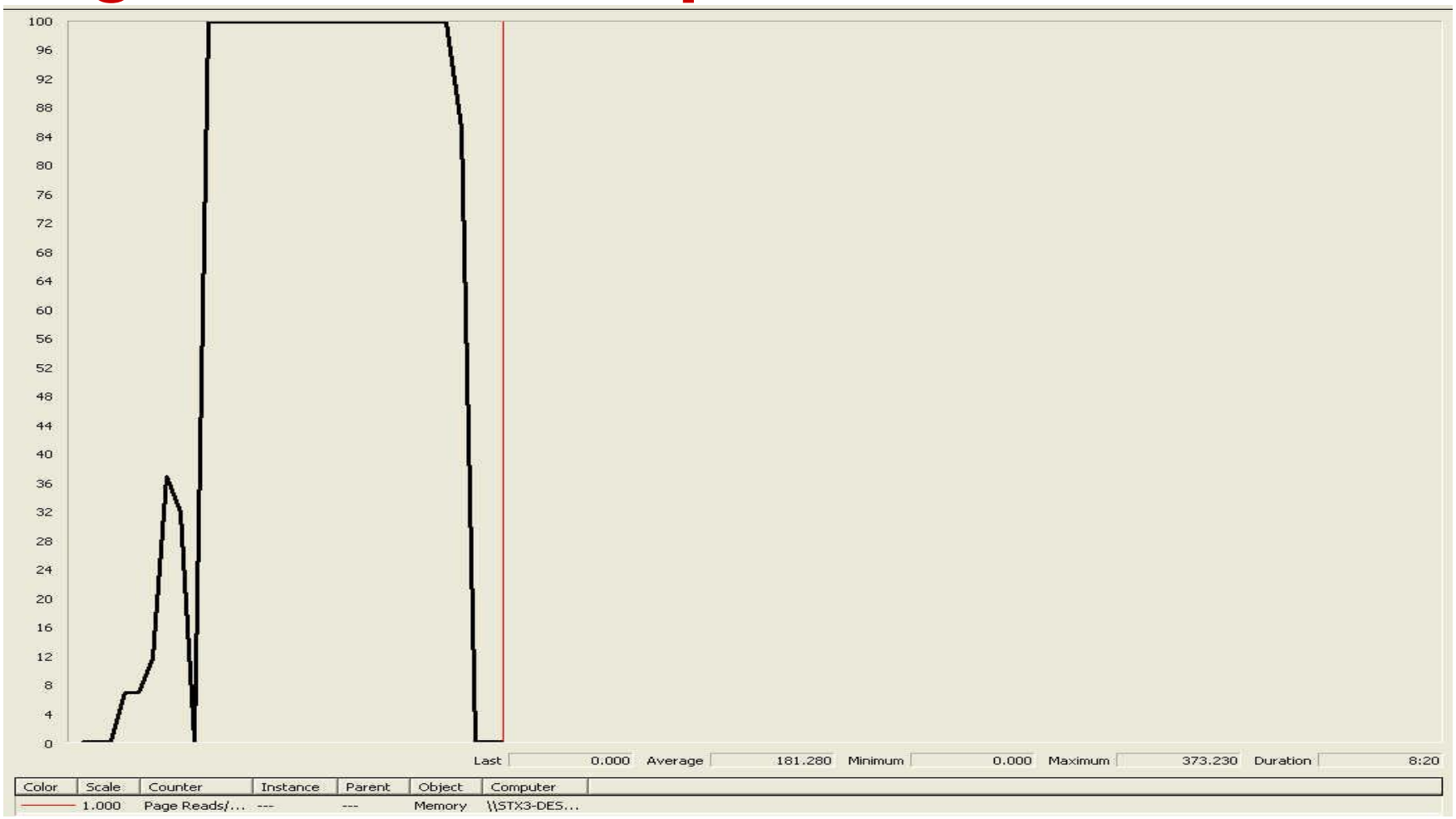
- Same counters available plus space-related ones
- Useful when multiple logical drives reside on one physical LUN

Which Performance Counters Useful?

Memory

- Page Reads/sec
  - **Not** just reads **from** paging file!
  - SQL Server I/Os **not** counted here
  - Should be almost zero on dedicated SQL Server machine **except** when
    - Reading flat files into the database
    - Working with backups
    - Recreating full text indices
- Available Bytes (Kbytes or Mbytes)
  - Should be at least 500 MB to allow for above activities
  - Some books suggest 4 MB ok – it is **NOT**
  - **System will stop responding long before this point**

Page Reads Example



Which Performance Counters Useful?

Network Interface

- Bytes Total/sec (for each NIC)
- Packets/sec (for each NIC)
  - Packets usually saturate NICs long before byte traffic does
  - Especially true if outboard optimizations disabled
    - Many have been disabled by default in the past
- Sometimes helpful to highlight application server problems that really are not database server problems
  - If very little data is arriving, problems may lie elsewhere

Minimal SQL Server Object List

SQL Server Objects	Category
SQL Server: Access Methods	Database access
SQL Server: Buffer Manager	Memory management
SQL Server: Memory Manager	Memory management

Interpreting Performance Counters

- **Processor queue length cannot be used reliably unless long spikes occur**
  - Most useful as relative measurement
- **Many counters misunderstood, e.g., % Disk Time**
- **Perpetuated by PerfMon explanation**
  - "% Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests."
- **Actually % Disk Time = 100 * Avg. Disk Queue Length**
  - **Frequently** referenced and interpreted as disk "busy" time
  - Completely **useless** metric
  - Artificially constrained to 100% by PerfMon
- **Actual busy = 100 - % Idle Time**

Physical I/O Measurements

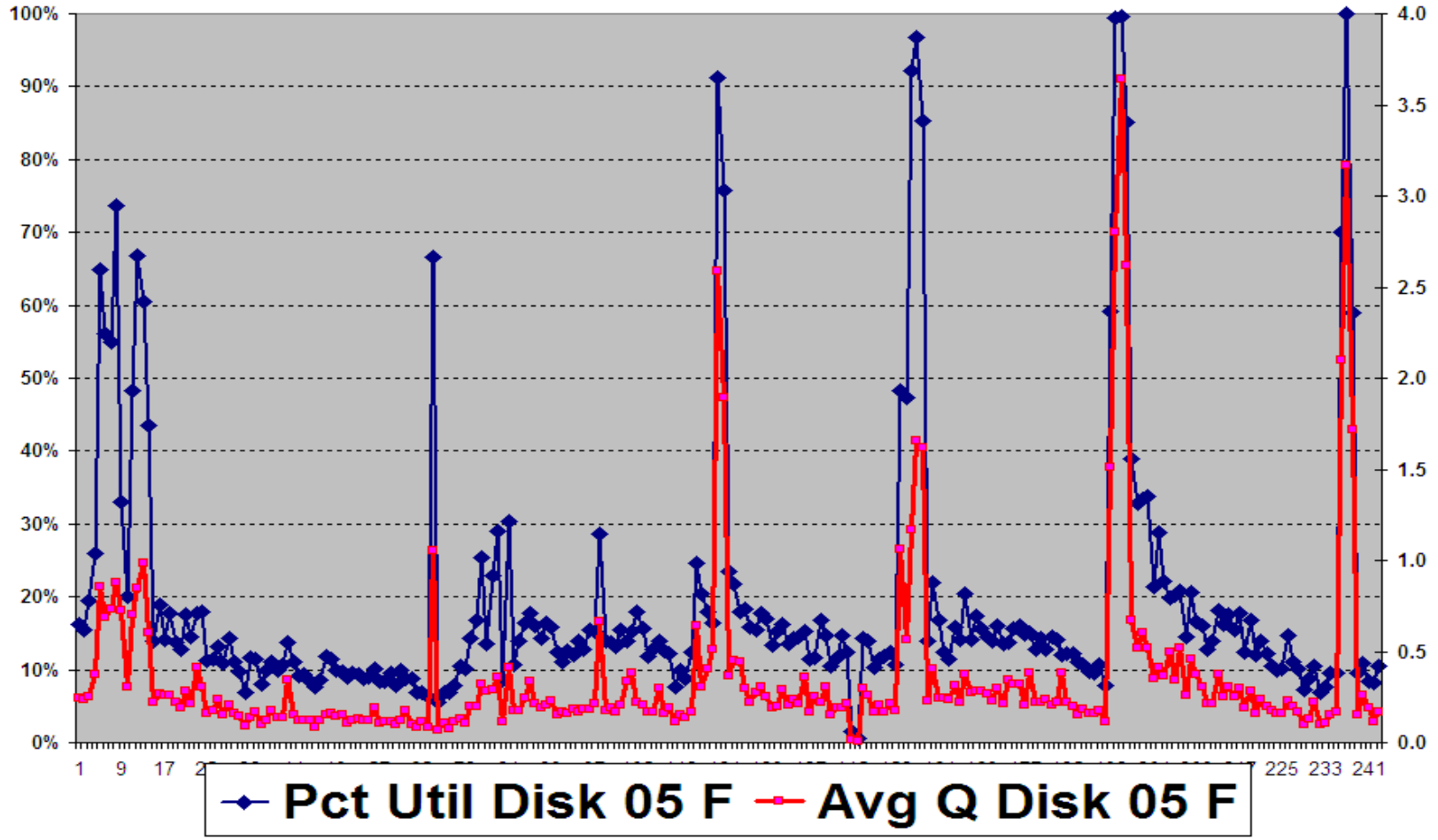
- **Critical for SQL Server systems because they are most frequently I/O constrained**
- **I/O time measured directly by disk driver**
  - Provides transfer times to Windows
- **I/O time = service time + queue time due to driver's location in I/O path**
  - Disk response time
- **Must know whether queuing causing large I/O times**
  - Reducing large service times usually requires additional hardware

Interpreting Performance Counters

Disk Queue lengths

- By far, **most commonly** quoted and used disk performance measurement
  - Actually **least** useful, except when outrageously high
- Use Avg. Disk sec/Transfer and % Idle Time instead

Utilization versus Queue Depth Graph



Performance Counters Incomplete

- **Two important metrics not measured or reported directly**
  - Avg. Disk Service Time per Transfer
  - Avg. Disk Queuing Time per Transfer
- **Missing values can be computed using the Utilization Law**

Using Utilization Law to Compute Missing I/O-Related Times

- **All calculations use PhysicalDisk counters**
  - LogicalDisk counters can be used, if necessary
- **Disk Utilization = 100 - *% Idle Time***
- **Disk service time = Disk Utilization / *Disk Transfers/sec***
- **Disk queue time = *Avg. Disk sec/Transfer* - Disk service time**

RAID Example Calculations #1 and #2

LUN #1	LUN #2
Disk Utilization 36.57%	**Disk Utilization** 77.67%
Disk Transfers/sec **0.65**	**Disk Transfers/sec** **30.89**
Avg. Disk sec/Transfer **2.0095 seconds!**	**Avg. Disk sec/Transfer** **2.4424 seconds!**
Disk service time 0.3657 / 0.65 = 0.563 seconds or **563 milliseconds**	**Disk service time** 0.7767 / 30.89 = 0.025 seconds or **25 milliseconds**
Disk queue time 2.0095 – 0.563 = **1.447 seconds or 1,447 milliseconds**	**Disk queue time** 2.4424 – 0.025 = **2.4174 seconds or 2,417 milliseconds**
Bytes/Transfer 1,307	**Bytes/Transfer** 22,437

RAID Example #1 vs. #2

- **I/O times (2.0095 vs. 2.4424) not that different despite being outrageously high**
- **Queuing occurred on both disks**
- **Low I/O rate of Disk #1 appears to contribute to high service times**
  - 1,307 bytes should not require 563 milliseconds

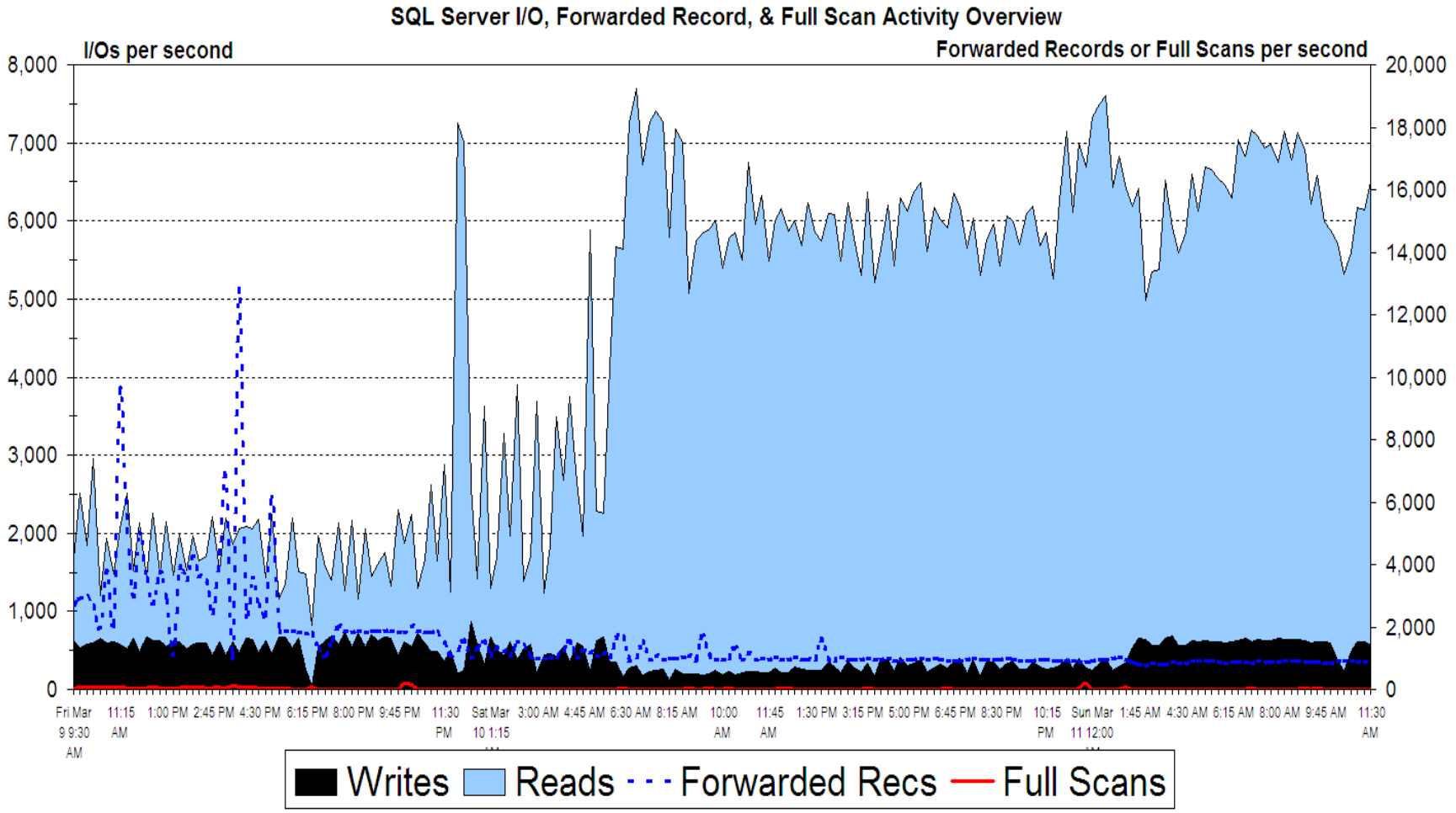
RAID Example #1 vs. #2

- **Disk #2 doing much more work**
  - Utilization double that of Disk #1
  - I/O size 17 times larger, but not huge
  - Service time much more reasonable @ 25 milliseconds
- **Problems began when faster processor complex attached**
  - Customer blamed new processor for poor performance
  - Wanted vendor to take it back because architecture was supposedly defective and slower than original
  - In reality, it was MUCH faster!
- **Solution was to reconfigure EMC drives**
  - Customer refused to state exactly what they changed
  - Probably multiple LUNs shared same physical drives

Database I/O Counters

- ***Page reads/sec*** **and** ***Page writes/sec*** **counters**
- **Measures physical I/Os, not logical I/Os**
- **May indicate**
  - Insufficient database memory
  - Applications improperly accessing database
  - Improper database table implementation
- **Useful to plot reads and writes together on same graph**
  - Highlights changes in workload behavior
  - Heavy write activity may coincide with periods of poor performance

SQL Server I/O Activity Graph



SQL Rx SQL Expertise

Microsoft GOLD CERTIFIED Partner

2008 DATA MANAGEMENT SOLUTIONS PARTNER OF THE YEAR-WINNER Database Management

SQLRx.com

Memory Very Important to SQL Server

- **Can reduce I/O subsystem load and improve performance**
  - Remember disk speed still in 10^{-3} seconds range, whereas memory speed in 10^{-9} seconds range
  - Also depends upon Read/Write ratio
    - Reads helped by memory
    - # of writes may be reduced slightly until a checkpoint

Buffer Cache Hit Ratio

- **Frequency database read requests satisfied from database cache memory instead of disk**
  - Often quoted exclusively as a measure of memory pressure
- **Higher values may result in lower disk usage**
  - Recommended value at least 90%
  - Raw performance data can sometimes exceed 100%
- **Not nearly as useful as many believe**
- **Observed numerous customer systems**
  - BCHR **never** dropped below 90%
  - However, Page Life Expectancy was seldom above 300

Page Life Expectancy

- **Measures amount of time non-locked buffer remains in memory**
- **Far more useful for identifying insufficient memory situations**
  - Values consistently under 300 seconds indicate SQL Server does not possess enough memory

Detecting Insufficient SQL Memory

- **Compare Memory Manager object's *Target Server Memory (KB)* with *Total Server Memory (KB)* counters**
  - If Total consistently less than Target, possibly insufficient memory
  - Procedure cache can be consuming remainder of memory
- **If *Page Life Expectancy* too low**
  - Allocate more memory to SQL Server or optimize queries
  - Malformed queries that read inappropriate amounts of data can cause low *Page Life Expectancy* and *Buffer Cache Hit Ratios*

SQL Statement Handling

Definition of a Batch

- Group of SQL statements
- Possibly hundreds or thousands of lines
- Must be parsed and compiled into optimized execution plan

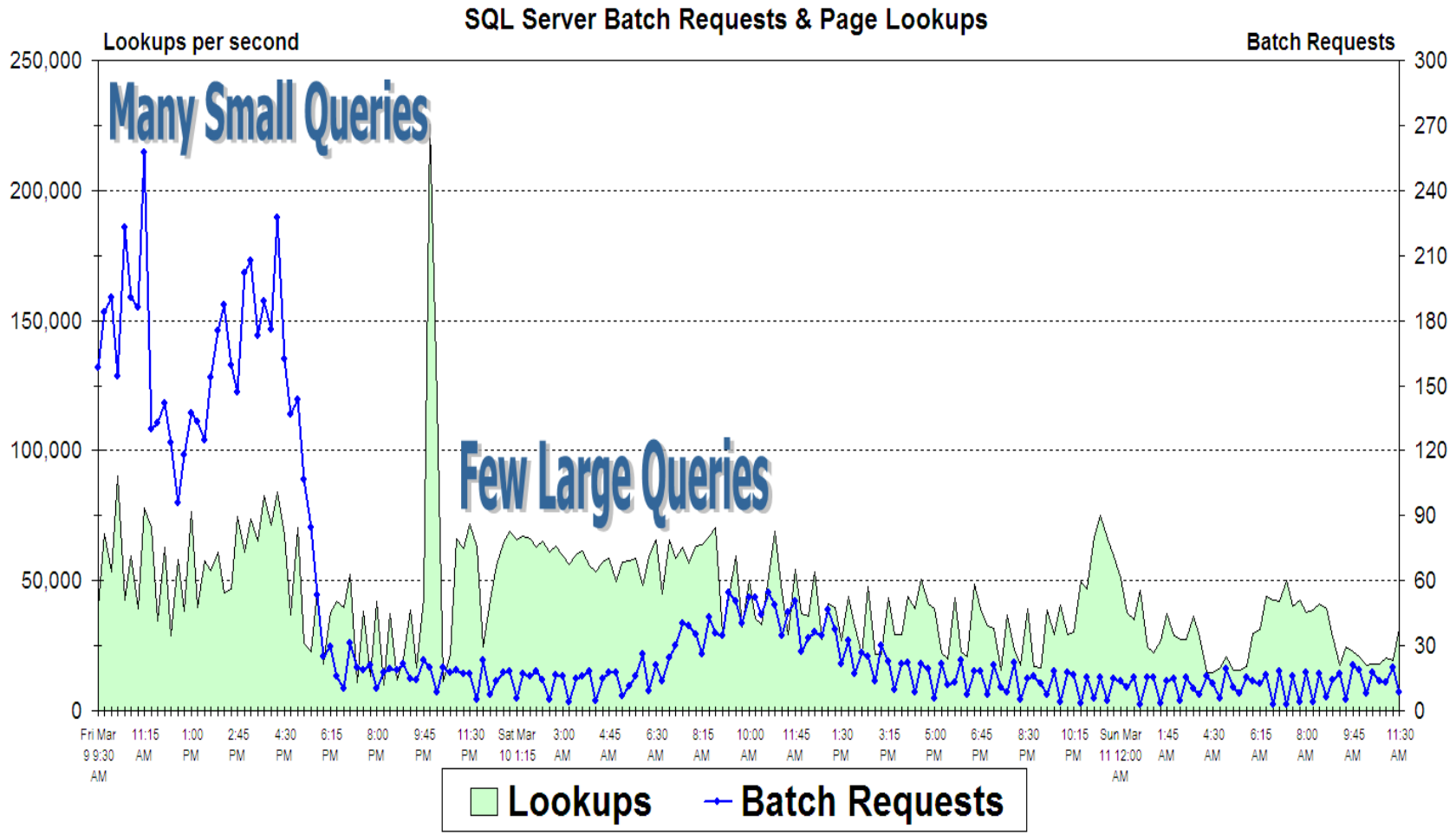
Batch Requests/sec

- **Does not adhere to actual batch definition**
- **Each select, insert, or delete statement triggers a batch event**
  - Causes counter to be incremented
  - **Note**: **Each** select, insert, or delete statement **within** a stored procedure counted

Page Lookups/sec Counter

- **Measures number of times database attempted to find page in buffer pool**
- **"Logical" read**
  - Corresponds to a read in SQL Profiler Trace
- **Compare**
  - *Batch Requests/sec with Page Lookups/sec*
  - *Page Life Expectancy* with *Page Lookups/sec*

Batch Requests vs. Page Lookups Graph



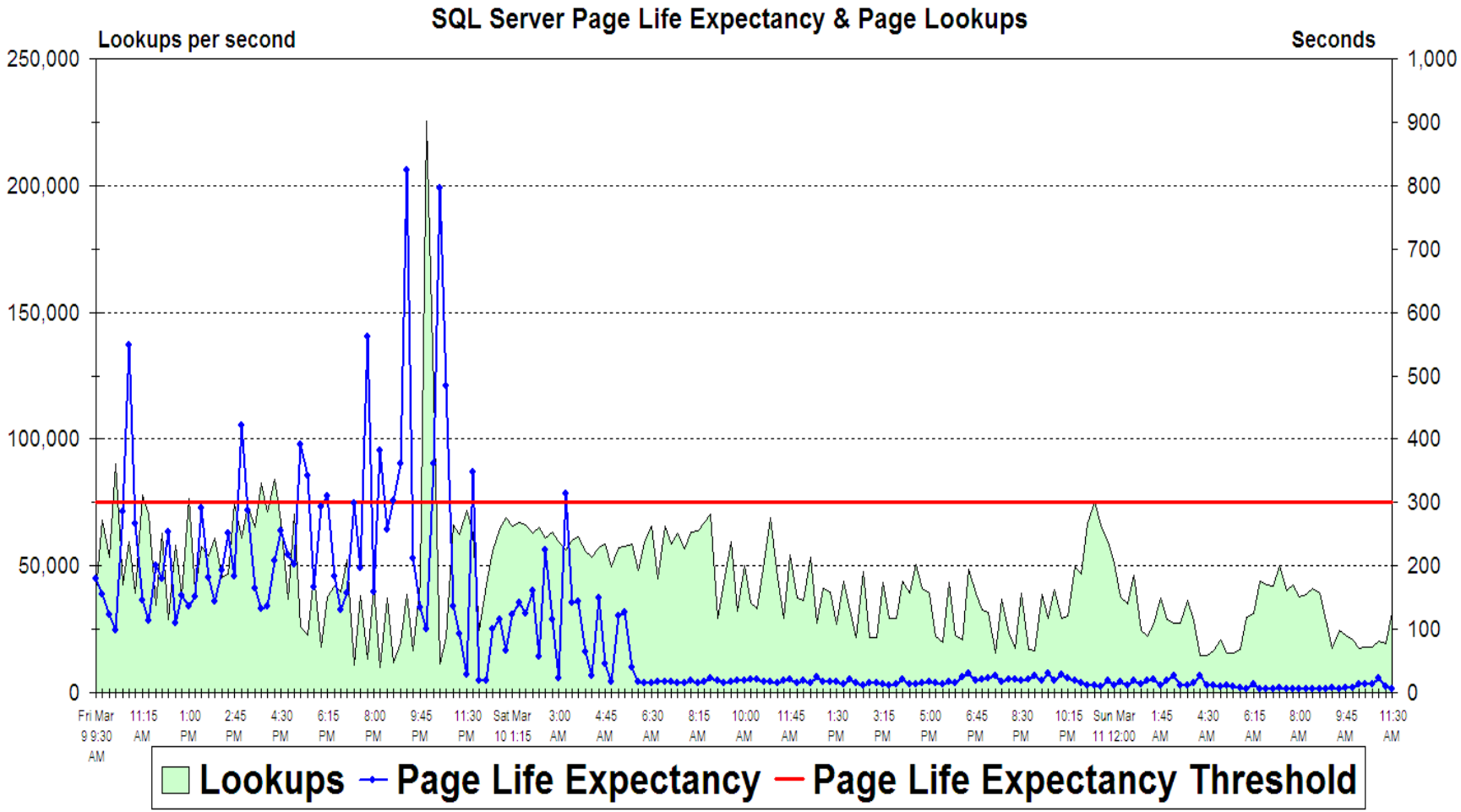
SQL Rx SQL Expertise

Microsoft GOLD CERTIFIED Partner

2008 DATA MANAGEMENT SOLUTIONS PARTNER OF THE YEAR-WINNER Database Management

SQLRx.com

Page Life Expectancy versus Page Lookups Graph



SQL Rx SQL Expertise

Microsoft GOLD CERTIFIED Partner

2008 DATA MANAGEMENT SOLUTIONS PARTNER OF THE YEAR-WINNER Database Management

SQLRx.com

Using SQLRx VitalSigns PerfMon Template

- Edit SQLRxVitalSignsV1.htm in Notepad
- Replace CUSTOMERSYSTEM with desired server name and save file
- Run PerfMon
- Select Counter Logs from Performance Logs and Alerts
- Right click in right pane and choose Select Log Settings From
- Navigate to SQLRxVitalSignsV1.htm (OK)
- If collecting remotely, change properties from binary to CSV
- Contact me for Vista or Server 2008 template

Using SQLRx (SQLRxVitalSignsV1_Sample.xls)

- Open created CSV (or converted binary) log file in Excel
- Determine column letters for PhysicalDisk % Idle and Disk Transfers/sec
- Copy formulas in columns BJ and BK from SQLRxVitalSignsV1_Sample.xls
- Change column values to reflect actual columns
- Copy for all LUNs

Using SQLRx (SQLRxVitalSignsV1_Sample.xls)

BJ2 =((100-W2)/100)/AE2

	A	BJ	BK
1	(PDH-CSV 4.0) (Eastern Daylight Time)(240)	PhysicalDisk(0 C:)\Disk Service Time	PhysicalDisk(0 C:)\Disk Queue Time
2	9:32:56 PM	0.008	0.013
3	9:33:26 PM	0.002	0.001
4	9:33:56 PM	0.003	0.003
5	9:34:26 PM	0.004	0.004
6	9:34:56 PM	0.003	0.001
7	9:35:26 PM	0.003	0.001
8	9:35:56 PM	0.003	0.001
9	9:36:26 PM	0.003	0.002
10	9:36:56 PM	0.003	0.001
11	9:37:26 PM	0.003	0.000

Assessing Your System

- **Potential problems exist if following counters consistently…**
  - % Processor Time > 70%
  - % Privileged Time > 30%
  - % Interrupt Time > 20%
  - % DPC Time > 25% (Processor)
  - Available Bytes < 500 MB (Memory)
  - % Idle Time < 40% for any Disk LUN and especially SQL LUNs
  - Avg. Disk sec/Transfer > 0.040 seconds (40 ms)
  - Avg. Disk sec/Write > 0.040 seconds (40 ms)
  - Page Life Expectancy < 300 seconds (SQLServer:Buffer Manager)

Conclusions

- **Windows Performance Monitor should always be used to focus tuning efforts**
- **Extremely important to combine Windows system performance and SQL Server information**
  - Especially for processor, memory, I/O, and network
- **Excel can be used to analyze PerfMon data**
- **Important missing Disk metrics can be computed**

Topics Covered in Future Sessions

- **Usage and interpretation of additional**
  - Windows and SQL Server Counters
  - Computed metrics from PerfMon counter data
- **Disk SAN and RAID performance issues**
- **Internal SQL Server performance data**
  - Usage of SQL Server 2005/2008 Dynamic Management Views
- **Using lean SQL Traces to identify performance issues quickly**
  - New SQL 2005/2008 SQL Trace event classes and their usage
- **Optimizing queries**
- **Your suggestions…**

Next Steps

1. Download *VitalSigns* tools zip file from www.sqlrx.com.
2. Collect data using *VitalSigns* PerfMon template.
3. Import data into Excel & add disk formulas using sample workbook.
4. Schedule a 15-minute *HealthCheck* (no charge) to review collected data. Email me, Dan Hooper, at dhooper@isi85.com or use "Contact Us" link on www.sqlrx.com .
5. Attend next in series and email suggested performance topics of interest to jeffrys@isi85.com.

SQL Rx SQL Expertise

Microsoft GOLD CERTIFIED Partner

2008 DATA MANAGEMENT SOLUTIONS PARTNER OF THE YEAR-WINNER Database Management

SQLRx.com