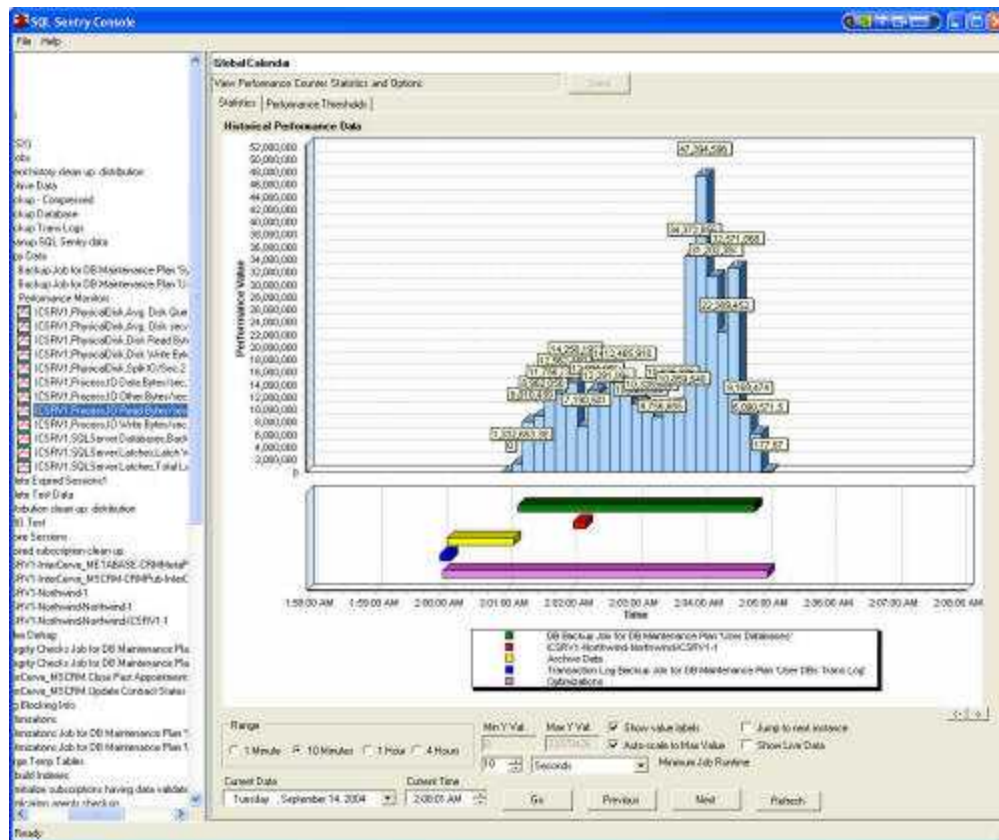


Monitoring Disk Contention

Unusually heavy contention for disk resources can dramatically impact SQL Server performance, and Event Manager makes it a simple matter to see exactly what role jobs are playing in this issue. Many types of jobs are some of the biggest offenders since they incur disk IO's outside of the SQL Server process space, meaning SQL Server isn't in direct control of prioritizing all of the reads and writes. This includes database and transaction log backups to disk, scheduled import/export (ETL) processes using DTS or BCP, or any other jobs that read and write large amounts of data to or from disk. If those jobs are hitting the same physical disk resources used by your database files (data/index or trans log files), it can cause performance problems for end users accessing those databases. Without warning application queries may take 2-3 times longer than normal, or even worse buffer latch timeout errors can occur. You can also experience a dramatically shortened lifespan for your disk resources.

To quickly get at the root of disk-related performance issues, using the Event Manager console you can link some disk-related performance counters to a job with a few mouse clicks. Event Manager will then automatically start and stop the counters according to the job's schedule, and generate wonderful graphs which visually correlate the job's run lifetime with performance counter activity.

Below are some recommended counters to use to determine how a database or transaction log backup job (to disk) is affecting disk performance.



SQL Server:Databases: Backup/Restore Throughput/sec

The total bytes transferred to disk by the backup operation.

Physical Disk: Avg. Disk Queue Length

The avg number of queued read and write requests during the interval. If you see this counter spike to over 2 per disk spindle while the backup is running, it's a strong indicator that the backup and/or other activity may be overloading the disk subsystem, causing incoming requests to be queued. There are individual queue length counters for reads and writes which can also be used.

Process: IO Write Bytes/sec [sqlserver]

The total bytes being written to the disk for the SQL Server process (sqlserver.exe). This counter should correlate roughly to the Backup/Restore Throughput/sec counter values while the backup is running. If it is considerably higher than the Backup/Restore Throughput/sec counter and you are seeing queued write requests during the same time period, it may indicate that the backup process is contending with other SQL Server database-related operations (data, index and/or transaction log).

Physical Disk: Disk Write Bytes/sec

The total bytes that are being written to the disk per second. This counter should correlate roughly to the Backup/Restore Throughput/sec counter values. If it is considerably higher than the Backup/Restore Throughput/sec counter and you are seeing queued write requests during the same time period, it may indicate that the backup process is contending for disk resources either with database-related operations, write activity from SQL Server import/export jobs, or write activity from other processes outside of SQL Server.

Here are some quick formulas which can help you gauge the percentage of write activity related to SQL Server and other processes:

The percentage of write activity related to the backup process =

Backup/Restore Throughput/sec / Disk Write Bytes/sec

The percentage of write activity from SQL Server database-related write operations =

(Process: IO Write Bytes/sec [sqlserver] - Backup/Restore Throughput/sec) / Disk Write Bytes/sec

The percentage of write activity related to all other processes (including SQL Server import/export jobs) =

(Disk Write Bytes/sec - Process: IO Write Bytes/sec [sqlserver]) / Disk Write Bytes/sec

Keep in mind that although we are focusing on the "write" counters here, controllers and disks have a limited throughput, so heavy write activity can affect read performance and vice versa. Some of the read activity you'll see during a backup is incurred by SQL Server reading data pages from disk that aren't in cache and loading it into the backup buffer. In other words backups don't always just write data to disk. That said, you may want to include some of the "read" counters as well.

Some other counters which can provide valuable insight into disk-related performance issues are:

Physical Disk: Avg. Disk sec/Transfer

The average seconds it takes for each read and write to disk. This counter can be a good measure of how much slow disk performance is manifesting itself in slow performance for end users. Since the counter is typically a fraction of a second, it's best to use it as a



relative measure, i.e., look at how much it is increasing during the backup job. If it's tripling from .05 to .15 during the backup and you see a high percentage of activity related to SQL Server database operations using the above formula, it may mean end user queries are taking 3 times as long.

Physical Disk: Split IO/Sec

The number of times per second a disk IO was split into multiple IO's. High readings for this counter often indicate that the disk is fragmented, which can directly affect the rate at which data is written to and read from disk, and lead to queued requests.

After the backup job runs, take a look at the Event Manager performance graphs for each counter and you'll see how much the job is contending for disk resources with other jobs, database operations, or non-SQL Server activity. If you observe that other jobs are running concurrently with the backup job and causing disk queuing, using the Event Manager console you can easily level the schedule by using drag-and-drop to stagger the jobs. If you find that other database-related operations are contending with the backup job, the best option is typically to use a separate physical partition for the backups to disk, one that doesn't include any data/index files, transaction logs, tempdb, etc. Ideally this partition should be comprised of a completely separate controller and disk array. You may also be able to use this same partition for import/export activities.

By minimizing disk contention using Event Manager you'll often find that the net result is faster backups, faster databases, and happier end users!

About SQL Sentry Event Manager

SQL Sentry Event Manager is the ultimate scheduling, alerting and response system for optimizing schedule performance of database servers and related IT resources. Event Manager provides DBAs with "Outlook-style" visibility and functionality for managing SQL Agent jobs, Windows Tasks, and Oracle jobs in increasingly complex cross-platform environments.

Download a free trial at www.sqlsentry.net and contact our team at sales@sqlsentry.net to find out how SQL Sentry Event Manager can optimize your server schedule performance.