



Database Servers

Microsoft® SQL Server - IBM® DB2® - Oracle® - Sybase Server®

Database management servers are in the top tier of servers placing the heaviest demands on the storage subsystem. They are the subject of much discussion regarding performance optimizations. Even after optimizing database, index, query, and application design, DBMS servers often under-perform, and input/output bottlenecks are often the reason.

The intelligent allocation of CPU and memory resources is critical to optimizing system throughput. When CPU utilization is less than 99%, system resources are underutilized; the system is not performing at its full potential. Although most of system memory may be allocated, its underutilization may be implied by low processor utilization and very low memory or pagefile input/output rates.

Use processor and memory resources to enhance performance

Database servers contain built-in caches, yet these frequently do not achieve optimal system performance. High-performance disk caches and RAM disks can prove invaluable in putting underutilized system resources to their most advantageous use.

tempdb file access

Most database servers execute intermediate operations in temporary database files, typically referred to as "tempdb" files. Frequently, the server's overall performance is limited by access to the tempdb files. System monitor can be used to determine if the system is bottlenecked by tempdb access.

Because the tempdb files are, in fact, temporary, their contents do not require persistent storage. In view of this, RAM disks are the ideal candidate to accelerate systems constrained by tempdb access. Placing the tempdb files on a RAM disk eliminates disk access latencies, and puts CPU and memory resources to greater use.

Pagefile access

When a pagefile's input/output rates climb to levels which push against its I/O path bandwidth, system performance suffers. Windows servers executing memory-intensive operations are frequently bottlenecked by pagefile access. A block-level cache in the pagefile path often is able to improve pagefile performance. This proves true despite the fact that the physical memory for the cache is taken from system memory.

Additionally, because of the dynamic nature of the pagefile's contents, write-caching can be safely enabled with no negative consequences. The result is a noticeable boost in system throughput.

Table file access

Very large, intensely-accessed databases stress the Windows file-system cache, frequently revealing another bottleneck: table file access. Reallocating system memory to a block-level cache in the table file path can alleviate this problem. Again, when processor and memory resources are underutilized, intelligent re-allocation of those resources will result in greater system performance.