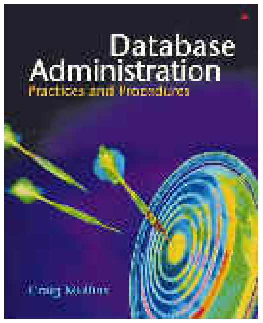
Craig S. Mullins

Return to Home Page

January 2006

DATABASE TRENDS AND APPLICATIONS



The DBA Corner

by Craig S. Mullins

Database Storage Management

Storage management can be quite complex, but when you factor in multiple different database platforms on a variety of operating systems, managing storage becomes extremely challenging.

Today's cross-platform DBA may be dealing with Sybase devices, databases, and segments in the morning and then have to switch gears to deal with DB2 containers, tablespaces, and nodegroups, in the afternoon. Simply remembering the storage syntax can be difficult.

Of course, when managing storage for databases, DBAs must not only master the differing syntax and methodologies for each DBMS, but must assure availability and performance of the databases being managed. Storage problems cannot be allowed to bring down an active database; similarly, storage must be defined and managed such that database performance is optimized.

But just what does this mean? At a high level, it means knowing the nature of each DBMS component that requires storage, and what its impact is on performance. Consider, for example, the impact of running out of storage capacity. It is a much bigger problem if your logs run out of space, than it is if a single tablespace runs out of space – because logs have a system-wide impact. The tablespace problem causes an outage for just that data, while the log problem causes an outage for every process that uses that log – and depending on the DBMS that might be the entire database environment. That does not mean that a DBA can ignore the "out of space" condition for a single tablespace; it just means that more resources should be placed on managing the storage for the logs.

Of course, autonomic computing facilities can help to minimize database storage management. DBAs can configure software to be ever vigilant for capacity problems and turn on "auto-extend" features that enable the underlying files to automatically grow to meet the demand of increasing data volumes. But such features do not yet create a "lights out" environment for managing database storage. What if there is no physical disk space that can be used to grow the files? So DBAs must keep a watchful eye on their storage resources.

Yet, if that was all that was needed to perform proper storage management for databases things would be quite simple for the DBA. No, storage issues can also cause performance problems, so the DBA has to have visibility to how his database uses storage. What types of issues need to be managed?

(1) Wasted space should be monitored and corrected. When a database object has more space allocated to it than it needs, performance degradation can occur. The more wasted space that exists, the more I/O that may be needed to retrieve data – especially for sequential scans.

(2) Fragmentation is a type of wasted space. It is the condition whereby there are many scattered areas of storage in a database that are too small to be used productively.

(3) Extent proximity is important, too. The DBMS engine's read-ahead capabilities can perform better when objects are contained in extents that are right next to, or in close proximity to one another.

(4) Data clustering and ordering should be matched to the most important processing needs. When indexes have a logical order that does not match the actual physical order of data stored in the database, performance can suffer. A tablespace can become unclustered when there is no room to maintain the physical order of the data in

sync with its clustering index. Why is clustering important? Well, consider a sequential read of customers whose last name starts with the letter "B". If the data is clustered, all of these rows will reside together on the same or contiguous pages. This reduces I/O operations which improves performance. If the data is unclustered, there is no guarantee where the rows will be – each row could even reside on a separate page.

(5) Chained and migrated rows can cause performance problems. When updated data does not fit in the space it currently occupies, the DBMS must find space for the row using techniques like row chaining and row migration. With row chaining, the DBMS moves a part of the new, larger row to a location within the tablespace where free space exists. With row migrations the full row is placed elsewhere in the segment. In each case a block-resident pointer is used to locate either the rest of the row or the full row. Both row chaining and row migration will result in multiple I/Os being issued to read a single row. This will cause performance to suffer because multiple I/Os are more expensive than a single I/O.

Database storage management can be a time-consuming task for a single-DBMS environment, but it can rise to gargantuan proportions in a heterogeneous environment. This is so because each DBMS product provides different storage techniques and uses different variations of the storage mechanisms previously described. So cross-platform DBAs need to take care when attempting to detect and diagnose database object storage problems. DBA tools can help – especially if you automate the problem detection with the analysis and problem resolution (which can require a series of administrative steps to be taken).

From Database Trends and Applications, January 2006.

Home.