

Why You'll Want to Upgrade to ASE 15.7

BY JEFFREY GARBUS, SOARING EAGLE CONSULTING

INTRODUCTION

It's that time again; Sybase[®] (now an SAP[®] company!) has added a suite of enhancements as an intermediary release (an "IR" rather than a full release), and it is decision time. Do you upgrade?

For most, an upgrade to an IR is not automatic; you either need the features that are coming out in the IR, or you wait for a major release. After all, the time needed for testing is not insignificant in today's world of having too many tasks to perform in too little time. Or is it?

The nice thing about an IR is that you are dealing mostly with code enhancements, rather than rewrites, so you're not dealing with dramatic changes to the optimizer that affect 90% of your queries; you are looking at new features that affect only areas of code in which you are hoping to see improvements anyway. So, while you're still going to want to run a thorough regression test, there's a realistic expectation that the testing will go smoothly.

I think that for many shops, many of the individual features in ASE 15.7 will be in and of themselves compelling enough reasons to consider the upgrade.

The features fit into a few broad categories:

1. Dealing with large data sets. This includes better management of large objects, as well as enhancements to operational, storage, and data management tasks. With the costs of today's high-end SANs, data compression may pay for the whole project.
2. Performance and scalability. New features here include better management of dynamic queries (I know that one is going to make friends) as well as a variety of ways to compress data, and better kernel integration for parallel processing performance.
3. More ease of maintenance. ASE 15.7 supports fully recoverable DDL commands such as select into, alter table, and reorg rebuild. In addition, you can now shrink log space, diagnostic support has taken a major step forward, and alter table statements can now be performed without copying data (how about that for speed of internal application upgrades?).
4. Application and developer productivity, including a variety of transact-sql enhancements.
5. Security, including object ownership changing, login profiles, and single sign-on.

There's not enough room here for a deep dive into all of the, so we're going to hit some highlights.

OPTIMIZING STORAGE

2.8 terabytes of data in one database isn't a lot today. Well, ok, it's a lot but it's not an unmanageable or unusual amount of data. I do though have a customer who manages 36 copies of the same database. I asked for the storage concession, and was laughed at.

This does add up. Sybase has added a few storage enhancements which will help this customer out, as well as a variety of others with large-dataset issues.

Row data (as well as LOB data) can be compressed

Compression was first introduced by Sybase for use with backups. One of the things we immediately discovered was that we can sacrifice a little CPU (used for compression) in order to gain significant storage benefits. This reduces both overall elapsed time (less writing, the worst offender for elapsed time) and the amount of required storage.

Compression can rescue applications that are straining storage requirements by compressing data on the pages, either at the page level or at the column level.

First, the server creates the ability to compress empty space in fixed-length columns. This reduces row width, and increases the number of potential rows per page, reducing the number of pages in the table, improving scan time. (Note: Improving scan time is not only valuable for queries that might scan tables, but also to improve times for reorgs, update statistics, dbcc checkstorage, etc.).

Next, there are two types of page-level compression available.

The first is page dictionary compression. In page dictionary compression, repeatedly referenced data is replaced by a symbol, and a symbol dictionary is created for reference at display time.

The second is page index compression. In page index compression a set of columns with duplicate data is stored in (only) one place, and is referenced via a marker in a row.

Finally, large object datatypes (LOBs) can be compressed using the same algorithm as the dumps.

Alter table statements no longer need to make data copies

I predict this enhancement makes a lot of friends. In particular, I have a lot of clients with large database that live in abject terror of updating certain table schemas, and occasionally need to plan extended (as in 3-day weekend) downtime in order to apply changes. Rollback plans for applying releases can be... nontrivial. The ability to modify tables without the tens of hours required for copying of tables that are hundreds of millions of rows deep helps us spend more weekend times with family.

Specifically, without data copy, you can now add a non-null column. Two desirable capabilities for a follow-on release would be the ability to drop a column and modify column type and nullability.

Benefits here are substantial. Release planning will become simplified (i.e. database teams will be happy) and data availability is improved (making end users happy). Add to this that temporary space and disk requirements are reduced or on a good day, removed.

In-row storage of LOBs

Prior to ASE 15.7, large object types were stored on separate data pages. Specifically, a 16-byte pointer on a data row would point to the starting page for the object referenced, enabling the server to use up to a 2 gig column size.

The catch is, if you are storing the characters, "Hi" you are going to store those 2 characters on a separate (4k) page, with the 16-byte pointer.

Here's the benefit now with ASE 15.7: If you have data (descriptions, notes, copies of emails, etc.) which have an extremely diverse range of data widths, you no longer have to worry about wasting extreme amounts of space. Note also that if you have any schemas with specific design flaws (perhaps somebody used text for columns that are frequently far less than a page in size), you may regain a lot of space.

DIAGNOSTICS AND MONITORING

The more complex your application code is, the more likely you are going to want to measure, maintain, and improve performance. This has been relatively easy if you are using constants (you turn on showplan, noexec and then submit the query to the optimizer; it returns a plan without running the query).

More often, though, you are running stored procedures (this remains a "best practice"), or have parameters that you are passing for analysis, which become local variables.

With 15.7, you can analyze dynamic parameters without running the query, which dramatically simplifies performance troubleshooting. In addition, you can identify which parameters are causing the specific problems when examining the execution plan or the optimizer cost estimate.

In addition, rather than running optdiag at the O/S level (or writing queries directly against sysstatistics, there is a new stored procedure (sp_showoptstats) to allow you to work with them within T-SQL.

Finally, when you are absolutely stuck, and want to involve Sybase technical support, rather than having to collect logs, system parameters, and other associated documentation, ASE can collect comprehensive configuration, monitoring, O/S and platform environment information in external files, generate a .zip output file with html files (viewable in browser), and have a single ready-to-send file to FTP up to Sybase Technical Support.

APPLICATIONS DEVELOPMENT

For those of you who script a lot, there are new scripting language drivers created for PHP, Perl, and Python.

For those of you who engage in data acquisition for data warehousing, ASE 15.7 has added the merge (*upsert*) statement, which combines the insert and update. For example, if you have a source dataset, you can “merge” it into a target data set by instructing the server to insert new rows, and update existing rows. This can save a lot of T-SQL code.

Sub-select syntax has improved. My favorite allows you now to have multiple columns in the column list of an exists statement (to test foreign key existence in a target table with a compound index, we have had in the past to jump through a few hoops).

Wider DOL rows now permit DOL columns up to 32767 bytes in width (up from 8192).

PERFORMANCE

With the advent of ASE 15, Sybase taught the optimizer a lot of new tricks (in fact, it was completely rewritten). The downside of new tricks is that the optimizer needs a bit more time to decide when to use them. This was somewhat mitigated at migration time by increasing the procedure cache, and creating or increasing the size of the statement cache (if you are on ASE 15 or higher and haven’t done this yet, I strongly recommend it).

With ASE 15.7, Query Processor (QP) latency has been reduced, especially for dynamic sql. The code path has been optimized, and query plans may now be shared across sessions, further reducing dbms workload in high-volume systems.

Query execution engine performance has improved, and client-ASE communication overhead has been reduced, creating a combined performance improvement of up to 2 times on a standard TPCC workload.

In addition, there have been a variety of kernel enhancements, enabling better utilize parallel cores, better integrate I/O event processing, and reduced I/O overhead.

This has led to better performance scaling on parallel hardware.

SECURITY

A simple enhancement which until this point we’ve done through either the tedious process of dropping and recreating objects or the risky process of directly modifying system tables is object ownership change, now built-in.

Sybase has also introduced login profiles.

Managing a large number of logins can be difficult when the CSO identifies a change that needs to be made to your security policies, whether because of an identified hole or due to Sarbanes-Oxley or HIPAA or new SEC regulations. Login profiles are SQL-defined containers of login attributes and their values (describe the three bullets under login profiles).

PCS/DSS requirements are also now met allowing split knowledge / dual control of encryption keys, so that the sudden departure of a security officer doesn’t cause data access problems.

In addition, ASE 15.7 provides stronger password and hidden text encryption using one-way secure hash (SHA-256), meeting FIPS 140-2 standards compliance.

Finally with ASE 15.7, Sybase also offers single sign-on and E-2-E Kerberos credential forwarding, passing user credentials between client mid-tier, and server levels without having to reauthenticate.

LOB ENHANCEMENTS

This last set of enhancements is very specialized, but amongst a certain group will make friends very fast. In general, large objects are best stored outside the database, with pointers inside the database for reference (a database is simply not a good way to store large objects; the header on each row, all by itself, creates storage overhead).

For those of you with a business need to do this though, or have purchased a 3rd-party system, or are dealing with legacy applications, help is at hand.

ASE 15.7 introduces a new feature called the “LOB Locator”, a handle to a LOB that can be referenced in t-sql statements passable between server & client.

More information is available in the reference manual, but in short, this reduces memory requirements, memory (you can reference the handle or value rather than the full object), and network requirements (again, you can pass the handle rather than the whole LOB).

ODBC, JDBC, and T-SQL enhancements all exist to take better advantage of the new features, including the ability to pass a large object to a stored procedure.

SUMMARY

Not everybody jumps to be an early acceptor to a new software release. Testing can be a lot of work, and few people have the time it takes to perform regression testing without a business driver.

In this paper, we’ve listed a host of potential individual business drivers for upgrading to ASE 15.7; as a collection, they’re pretty convincing.

Jeff Garbus has 20 years of expertise in architecture, tuning and administration of Sybase ASE, Oracle, and Microsoft SQL Server databases with an emphasis on assisting clients in migrating from existing systems to pilot and enterprise projects. He has co-authored 15 books and has published dozens of articles on the subject. Mr. Garbus is the CEO of Soaring Eagle Consulting, an organization that specializes in assisting businesses maximize database performance. www.soaringeagle.biz.