Introduction

I decided to write this booklet after going through upgrading two major trading systems to Sybase Adaptive Server Enterprise (ASE) 15. One of these systems was Equities [1] and the other one was a Fixed Income application. I thought that there was a need to write such a booklet that focused on the practical aspects of upgrading to ASE 15, as opposed to the widely available literature published by Sybase Inc.

This is thus an aid. It is not meant to replace what Sybase officially provides nor your familiarity with your own application. However, what it provides is in line with the tried and tested steps on how to upgrade typically complex trading systems.

The target audience of this book is the Sybase upgrade team. This is the team that has hundred percent responsibilities for the successful upgrade and future performance of ASE 15. This team includes the upgrade project manager, infra-structure manager, the developers, testers, UNIX administrators and the DBA staff. This is contrary to the popular belief that the DBA is responsible for every aspect of ASE 15 upgrade and has complete responsibility over it. Since I am a fan of motor racing and Ferrari, I will use the analogy of a Formula 1 race. The DBA is the mechanic/pit-stop guy who makes sure that the fuel tank is full, the tires are replaced, the visor is cleaned and Ferrari functions as expected and ready for the remaining laps. Of course the DBA role is very crucial. A pit stop is never as easy as it looks. For seven to eight seconds (if all goes well), the drivers' chances in the race are in the hands of his pit crew/DBAs. All the racer can do is sit and wait. However, there is very little the pit-stop guy can do once the car has left the pit. It is up to the development team and testers to ensure that the car (in this case ASE 15) performs as it should and does handle those bumpy corners smoothly. In short, this ASE 15 upgrade is not just the responsibility of DBA but rather a team effort.

Let us backtrack and examine what distinguishes ASE 15 from the other previous releases of ASE. Most of us work in environments with applications that often operate under Extreme Transaction Processing (XTP), mixed Online Transaction Processing (OLTP) and Decision Support System (DSS) loads. These mixed loads are usually referred to as Operational Decision Support Systems (ODSS) these days. These applications typically have to provide a very quick response. Although we are required to answer very fast, we have to perform complex SQL statements on our databases that can be dozens of Gigabytes in size. For a long time, ASE (known earlier as Sybase SQL Server) employed nested loops joins and later on it added merge joins. The general rule of thumb, going back many years was that the nested loop join is good if at least one join's input is small or at least when indexes cover the join and the join cardinality is fairly small (i.e. 1:5). On the other hand merge join is good for two large inputs with high cardinality join (>1:10), especially where the tables are already sorted in join order (i.e. clustered on join keys) or where a predicate on the inner table can reduce the table projection to a quantity easily sorted in available memory. Hash joins work best on large tables for which no other predicates other than the join keys exist or the projections have large join cardinality, but where the input tables are too large to easily sort.

Some database systems, in particular those that targeted online transaction processing applications, did not even use merge join, never mind other join methods like hash join. This is certainly true in the early days of ASE. All join operations were executed by the nested loop algorithm. The most sophisticated join algorithm sorted the outer input and built an index on the inner input on the fly - called "reformatting" or store_index in optimization parlance. As time went by, systems had to adapt to handle both OLTP and DSS. As a result, ASE 15 introduced the hash join (HJ) algorithm. Hash join was introduced in ASE to deal with the requirements of ODSS.

Starting with ASE 15, Sybase decided to rewrite the optimizer of ASE from new. It introduced novel concepts such as n-ary Nested Loop Join (NLJs involving 3 or more tables with a close query graph) and based the new Optimizer on Volcano model with the new execution engine called Lava Execution Engine.

What all these mean to us the practitioners is that ASE 15 uses more sophisticated query-processing techniques compared to the previous versions, both in terms of the set of relational operators and algorithms implementing them. As a consequence, it is fair to say that the task of the optimizer has become harder - and thus takes longer. This is important as a full query execution involves parsing, normalization, optimization and execution - and in some cases, optimization can exceed actual execution time. Generally, optimizers handle models of query execution – and there is a conceptual difference between models and practice. The best model in the world can potentially leave parts of the reality uncovered. That is, for any real life optimizer, no matter how good it is, there will always be a query, a database state and a system state where the optimizer will take the wrong decision - or will simply spend too long optimizing as compared to actual execution time. There is not such a thing as the perfect optimizer. If we agree that making occasional errors is in the nature of query optimizers, then we the practitioners need to find a way to understand the behaviour of the new ASE 15 optimizer and to cope with that.

Understanding ASE 15 new optimizer behaviour is thus important in upgrade processes. However, that is not the end to itself. In the words of a good colleague of mine, upgrading ASE 15 is not just about loading the database from an earlier version, online it and we are home and dry. As we will see later there are other factors that can make your upgrade a success or failure.

The organisation of this booklet

When I tried to push for ASE 15 in a client site in late 2007, I came across various questions and justifications from the top management that were not technical by nature. I decided then to present a cost benefit analysis of deploying ASE 15. Something that I thought could go well with less technical minds that were more interested with the value add to business (in short what business gains from this upgrade). With the benefit of hindsight that was the correct approach. So this booklet is not all about bits and bytes. It involves some business considerations that you may find applicable to your environments or you can adapt them for your own use, in order, hopefully, to facilitate your migration to ASE 15.

To keep this booklet concise and focused I have decided to detail subjects that I have actually deployed myself in ASE 15 upgrade and there are definitely certain utilities that I have not used. For example I made extensive use of Monitoring Data Access (MDA) tables readings but did not deploy any Query processing (QP) metrics. I therefore do not cover QP tuning. I also do not provide the configuration parameters for MDA settings. I therefore assume that the reader is either familiar with the MDA settings or can get some information either from myself (via email) or another resource. Of course there are others that have their own home grown monitoring scripts using Perl and other stuff. So there is a choice.

Going forward I cover various upgrade methodologies as you may care to consider. I then touch on ASE configuration settings of importance to ASE 15. We then take a route on loading a database into ASE 15 and walk through the load process. We consider such aspects as upgrading from ASE 32-bit to ASE 64-bit. Post database load we look at reorg rebuild, various aspects of statistics and consider the new ASE utility datachange() function, followed by missing statistics and histogramming steps. No upgrade is complete without detailed consideration of the ASE 15 optimizer including the join methods and optimizer switches. Since Hash Joins are new to ASE 15, we look at this join method in more details. We then cover the auxiliary utilities that you need to consider with ASE 15 upgrade, such as the correct version of Perl libraries, the SDK for UNIX and Windows and Jconnect for Websphere etc. In a complex environment there are often various types of replicated data from ASE or other sources (simple or bidirectional). We will consider replicated data from ASE 12.5.n to ASE 15 and vice versa.

No upgrade is a success without testing the code in ASE 15. It is equally important to consider how to go about testing the code as well as doing it. We will then look at planning test cycles, how to capture performance matrix, zooming on problematic SQL and so forth. ASE 15 optimizer will play a major role in the upgrade process and thus familiarity and understanding it is a must for DBAs and developers. As a DBA you may end up explaining the plan output to the developers so you will need to be sufficiently familiar with the optimizer concepts.

In the course of evaluating ASE 15 semantic partitioning we found it to be useful as a substitute to the traditional archiving. Thus partitioning besides performance benefits can potentially have other usage. This is also covered.

Why Upgrade to ASE15

This is a question that inevitably comes up in any conversation with the management. Is ASE 15 upgrade a high risk process with little return? Do we have to upgrade just because the end of life for ASE 12.5.4 is sometime in 2009? What are the technical and business reasons to upgrade to ASE 15, given the resources and testing time needed?

Technical Considerations

Operational Advantages of deploying ASE 15

- Eliminate/ remove specific performance bottlenecks
- Make Sybase look and feel the same across the board by allowing all the applications to be upgraded to the same version of ASE, thus getting rid of the old versions across the board
- Support for latest generation operating systems
- The new ASE 15 will make DBA support work more streamlined as it will facilitate DBAs to:
  - Consolidate to one version of ASE
  - Implement new Sybase installation standards.
  - Implement uniform housekeeping scripts across different applications
  - Retire Legacy scripts
  - Utilize new and better performance monitoring tools
  - Reduce maintenance windows over the week-end
- Maintenance jobs timing will be further reduced with semantic partitioning because the process can be done using parallelism and on the individual partitions only
- End of life in 2013.
- ASE 15 allows for data warehouse type databases to be built

Pre ASE 15 limitations

- Query optimizer costing model needed improvement
- Existing versions of ASE were 'hard-wired' for OLTP
  - Nested-loop joins and merge joins only
  - Poor or inadequate handling of DSS queries
- 256 Device limit per server and 32GB max device size limitations
- Issues with catalog contention for DDL operations, especially in temporary databases
- 30 character limit on object names
- Limitations with statement cache and ad-hoc queries

Improvements in ASE 15

- ASE 15 execution engine based on the Volcano model
- More flexibility to add new access methods

- ASE 15 execution is internally Stream-oriented rather than step-by-step oriented. This makes the result set order more efficient.
- Query optimization has been improved using new techniques such as pipelined parallelism, cost-based pruning, and timeout mechanisms as part of Volcano model
- Text replication is faster
- Key Catalogs are row-locked in all databases
- Reduced catalog contention for DDL operations
- Higher throughput for applications
- Deadlocks due to system catalog contention reduced considerably
- Improved application availability
- Increased concurrency for maintenance tasks
- Contention in the temporary work database tempdb reduced significantly
- Relieve log/cache contention
- Large identifiers, removes 30 character limit on object names
- Allows for Unsigned and 64-bit integer types
- Default network packet size = 2048 Bytes. Previous versions used 512 Bytes due to legacy network protocols
- 256 Device limit per server and 32GB max device size limitations gone
- Devices = 2^31; maximum device size = 4Tb
- Blocking due to stored procedure renormalization largely eliminated
- Renormalization caused by temp tables created outside proc, etc
- ASE 15 handles OLTP more efficiently

New functionalities in ASE 15

- Semantic partitioning:
  - Tables can be divided into small chunks of data called “partitions” that can be individually managed. Queries run faster because the “smart” query optimizer bypasses partitions that don’t contain relevant data. In addition, one can run maintenance tasks on a subset of selected partitions which speeds up overall performance.
- Computed Columns:
  - The results of calculations can be stored in columns, saving time when applications repeat the same calculation over and over.
  - Optionally, the user or application can store the formula instead of the result, creating a virtual column that will be recalculated when it is accessed. This saves the effort of re-specifying the definition, and allows other columns to be created based on actual and virtual ones.
- Function Indexes:
  - The server can build indexes on a table based on the result of a function. When repeated searches use the function, the results do not need to be computed from scratch and the index can be used to do fast lookup based on function result.

- ASE 15.0 supports three new/improved join methods all useful for many typical applications:
  - N-ary Nested loop Join (NLJ),
    - A variant of NLJ patented by Sybase
  - Merge Join (MJ)
    - Improved from ASE 12.0 implementation
  - Hash Join (HJ)
  - Queries hitting many rows
  - Queries against star/snowflake schema
  - Queries with many aggregates
  - queries joining many tables

    - Queries that are not-so-well written
- ▪ SQL User Defined Functions added
- ▪ The new *instead of triggers* function improves bulk loading of data
- ▪ XML enhancements

Business considerations

Cost/Benefit analysis of migrating to ASE 15

- **Costs: (**The investment needed to realize the value of upgrading to ASE 15)

    - ➢ Implementation costs
    - ➢ Ongoing costs
    - ➢ External consultancy costs if any

- **Benefits** (The value delivered to the IT and business units by this upgrade)

    - ✓ Improved functionality
    - ✓ Reduced development cycle
    - ✓ Improved performance and query response time
    - ✓ More efficient DBA support
    - ✓ Reduced maintenance times

Other Factors

- **Data growth trends**

    - ➢ Anticipated data growth of 50-100% annually in most sites
    - ➢ Higher demands on space and performance

- **Provide value add to the development teams**

    - ✓ Enable developers to take advantage of new functionalities
    - ✓ More efficient code, less workarounds, thus reduced maintenance cost

- **Manage and control the support resources**

    - ➢ Current freeze on resources at times of rising demands with the technology dollars at premium

- **Maintain high level of availability**

    - ✓ Sybase is well known for its dynamic configuration, thus allowing online changes. ASE 15 takes this further making the impact of changes to the configuration of Sybase servers on users and the applications minimal.

- **Maintain high level of scalability**

    - ✓ Every single application we use in Sybase requires high availability in practice

  - ✓ Some systems are 24x5 or more
  - ✓ ASE 15 Cluster Edition builds on Sybase’s reputation for reliability by delivering a database framework that reduces costs, improves application availability and facilitates creating a flexible data infrastructure for future business growth. An ideal solution for *mission critical* applications

- **Better server consolidation**

  - ✓ ASE 15 caters for both OLTP and DSS workloads. This allows applications to use the same Sybase serves on the same hardware, thus sharing computing resources among applications. Instead of maintaining multiple copies of ASE and the OS, you only need to manage a single copy.

Some Math

- **Cost Assumptions**

  - ➢ Working days in a Month = 20
  - ➢ Uniform cost per day = £840 (£105 per hour). This is based on the total cost of a resource per typical company averaged over all resource type (DBA, developer, tester etc)
  - ➢ Resource cost per week = 5 * £840 = £4,200

- **Typical Implementation costs**

Project	**DBA + Data Architect resource (week)**	**Tester or developer Resource (week)**	**Batch cycle run (week)**	**Resource total (week)**	**Duration (week)**	**Total cost (£)**
Small	1	2	1	3	4	3 * 4,200 = 12,600
Medium	3	7	2	10	11	10 * 4,200 = £42,000
Large	8-12	20-30	8	28-42	20-30	28 * 4,200 = 117,600 to 42 * 4,200 = 176,400

In addition to the above cost you are advised to add 10% contingency on the top. Consideration should also be given to indirect costs such as:

  - ➢ Licensing cost
  - ➢ New hardware cost

- **Benefits, tangible**

  - ✓ Improved functionality
    - – 30% functionality improvement

- ✓ Reduced development cycle
  - 20 % reduction in development cycle
    = 20 days/month*12 months*£840/day*0.2 (20% reduction) = £40,320 annual savings per developer resource
- ✓ Improved performance and query response time
  - 25% reduction in incomplete queries
  - 20% reduction in lost transactions (non recoverable)
  - 5% reduction in transactions not completed due to database not being available.

  Assume 10,000 transactions per month at £100 per transaction. This leads to **10,000 * 12 * 100 *0.25*0.05* 0.2 = £30,000 annual value regained**

- ✓ More efficient DBA support
  - 30% reduction in the annual staff growth

  = 20 days/month*12 months*£840/day*0.3 (30% reduction)
  = £60,480 annual savings per resource

- ✓ Reduced maintenance times
  - Our test on our larger databases showed on average four times faster update index statistics compared to the previous versions of ASE. Even on a conservative basis one can actually schedule the whole of the maintenance window on one weekend day, thus providing 6x24 as opposed to the usual 5x24 availability. That is 20% extra business capacity/improvement in productivity. Assume 5,000 batch transactions per month at £100 per transaction. This leads to **5,000 * 12 * 0.2 = £12,000 annual value increase**

- Summary Benefits (annual)
  - Reduced development cycle = £40,320
  - Improved performance and query response time = £30,000
  - More efficient DBA support = £60,480
  - Reduced maintenance times = £12,000
  - **Total benefits = £142,800**

Summary Cost/benefit of upgrade to ASE 15

We take an example of a typical cost for a large project (28 days) with an estimated cost of £117,600 as calculated from the above table.

Total cost (present value)	= £129,360
Total benefits (present value)	= £142,800
Total (net present value)	= £ 13,440
Payback period (years)	**= 0.9**
Total cost (risk adjusted[†])	= £181,104
Total benefits (risk adjusted)	= £ 85,680
Payback period (years, risk adjusted)	**= 2.1**

What the payback periods indicate is that the cost of deploying ASE 15 will be recovered at 0.9 years or at worst scenario in 2.1 years taking into account the benefits vs. the capital expenditure.

[+] Risk adjustment is worked out at 40% above the present value of Total Cost and 40% below Total Benefits