<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Data migration engineering at HP</td>
<td>2</td>
</tr>
<tr>
<td>Migration circumstances</td>
<td>3</td>
</tr>
<tr>
<td>When should you migrate?</td>
<td>3</td>
</tr>
<tr>
<td>Transition and your business practice</td>
<td>4</td>
</tr>
<tr>
<td>Database content and use (static and dynamic)</td>
<td>4</td>
</tr>
<tr>
<td>What transition system environments will you need?</td>
<td>6</td>
</tr>
<tr>
<td>Migration steps overview</td>
<td>7</td>
</tr>
<tr>
<td>Step 1: Setting up the system, storage, and network</td>
<td>8</td>
</tr>
<tr>
<td>Step 2: Create new environment and database</td>
<td>8</td>
</tr>
<tr>
<td>Step 3: Static or dynamic-specific tasks</td>
<td>8</td>
</tr>
<tr>
<td>Sybase utilities</td>
<td>9</td>
</tr>
<tr>
<td>Sybase Bulk Copy Program</td>
<td>9</td>
</tr>
<tr>
<td>Sybase Component Integration Services</td>
<td>11</td>
</tr>
<tr>
<td>Sybase migration utility</td>
<td>12</td>
</tr>
<tr>
<td>Transaction replication</td>
<td>13</td>
</tr>
<tr>
<td>Backups, snapshots, and clones</td>
<td>16</td>
</tr>
<tr>
<td>Verification</td>
<td>16</td>
</tr>
<tr>
<td>Sybase database consistency checker tool – dbcc</td>
<td>16</td>
</tr>
<tr>
<td>Sybase optimizer statistics diagnostic tool – optdiag</td>
<td>17</td>
</tr>
<tr>
<td>Sybase Central</td>
<td>17</td>
</tr>
<tr>
<td>Your application</td>
<td>17</td>
</tr>
<tr>
<td>Summary</td>
<td>17</td>
</tr>
<tr>
<td>Further information</td>
<td>19</td>
</tr>
</tbody>
</table>
Introduction

In the coming years, HP will be phasing out Tru64 UNIX and the AlphaServer enterprise system line in favor of the new HP Integrity server line based on the Intel® Itanium® 2 processor architecture. HP is also integrating leading technology from Tru64 UNIX, such as TruCluster Server software and Advanced File System (AdvFS), into HP-UX 11i. In light of this evolution, your IT organizations should begin planning when and how to migrate your Tru64 UNIX AlphaServer-based Sybase databases to HP-UX 11i on the HP Integrity server platform.

HP is investing significant resources to help you transition easily and reliably to HP-UX 11i on HP Integrity servers. To offer you reliable and valuable advice and assistance in the database migration process, HP is:

- Investigating optimization strategies and tactics, performance and quality issues, and best practices
- Delivering tools and white papers related to migration planning, design, implementation, and management

This document is a high-level, semi-technical description of strategies, tactics, techniques, and factors to consider when undertaking a Sybase database migration from a Tru64 UNIX system to an HP-UX 11i system. It addresses business considerations that might dictate how and when your enterprise should take on migration, and it explains several practical approaches regarding the migration of a Sybase database from one operating system to another.

Adaptive Server Enterprise 12.5

The database migration engineering work described in this document is centered on cases in which Adaptive Server Enterprise (ASE) 12.5 is the RDBMS in use on both the source and target platforms. If you are using an older version of Sybase’s RDBMS, HP recommends that you upgrade your environment to ASE 12.5 before conducting a migration of your database to another platform.

HP Services can help you with migrations from earlier versions if you require it. Future releases of Sybase products may change some of HP’s recommendations for certain migration scenarios.

While this document presents the current advice from HP on Sybase database migration, it is not the final work. Continuing studies will refine and add to knowledge in this field. As new information becomes available, HP will update this and other supporting transition documents to help you smooth your transition to future system environments.

Data migration engineering at HP

HP has invested, and is continuing to invest, significant resources in developing and verifying innovative techniques that will help you reliably migrate your Sybase database from a Tru64 UNIX environment to an HP-UX 11i environment. This investment includes the people, time, systems, and funds necessary to create and deliver tools, documentation, and recommendations that will help remove or reduce difficult or risky migration elements. The goal is to provide you with processes and methods that are efficient, safe, flexible, and successful.

1 Concentration is entirely on Sybase databases. For convenience, the word database alone means a database managed by one of Sybase’s Relational Database Management System (RDBMS) products (for example, ASE 12.5).
Migration methods and best practices are tested using representative databases as test rigs and workloads. The databases are described in the following table.

<table>
<thead>
<tr>
<th>Database</th>
<th>Representative environment</th>
<th>Example use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Processing Council TPCC</td>
<td>Environment with heavy on-line transaction processing</td>
<td>Online transaction processing such as eCommerce, telecommunications, financial</td>
</tr>
<tr>
<td>database</td>
<td>----------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Transaction Processing Council TPC-H</td>
<td>Environment in which most activity is batch update or query</td>
<td>Decision support or data warehouse work</td>
</tr>
<tr>
<td>database</td>
<td>----------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>“Mixed workload” database</td>
<td>Designed for testing RDBMS performance in “real-time enterprise” IT systems</td>
<td>Telecom business or retail store chain</td>
</tr>
</tbody>
</table>

The migration engineering activities follow normal engineering practice: research, investigative lab work, development and automation of tools, end-to-end lab testing, and development of documentation and training for customers, HP field personnel, professional services, and system integrators.

To help detect and avoid configuration-specific problems, HP conducts migration tests in more than one environment. Tests will be conducted both internally and with pilot customers, as appropriate.

As a result of its data migration investment, HP is in the best position to assist you by providing tools and reliable advice on migrating Sybase databases to the new generation of HP enterprise platforms. HP migration engineering intends to roll out additional materials, updated tools, and best practices as Sybase database products evolve over time. The Tru64 UNIX to HP-UX 11i transition tools and methodologies website provides up-to-date information and advice on database migration and other transition aspects:

http://www.hp.com/go/transition-modules/

The website resources include a series of Planning Assessment Documents (referred to as PADs) that discuss practical aspects of database migration.

**Migration circumstances**

The circumstances under which database migrations might be conducted vary from company to company (indeed, even within a company). Each company has its own particular environment, business model, and database characteristics which, taken together, influence when and how it should take on the migration of its enterprise databases to newer IT platforms. This section explores these factors and their impact on migration decisions and technique selection.

**When should you migrate?**

The easy (and correct) answer is “Migrate when, and only when, it fits your business needs.” There is no need to rush into a database migration, nor is there any reason to consider anything less than the big picture as it relates to your enterprise.

Recognize that the product roadmaps for Tru64 UNIX, AlphaServers, and HP Integrity servers allow great flexibility. Tru64 UNIX upgrades and new AlphaServer models are still being introduced and full support will be available through 2011. To see the latest roadmap and other transition information, go to:

http://www.hp.com/go/alpha-retaintrust
You might consider a database migration as part of a specific business need, such as moving to a new application, IT consolidation efforts, or other changes in your business practices.

**Transition and your business practice**

Remember that migration is not only about IT platforms. Look for economies and synergies that might come from combining migration with compatible improvements to your IT systems. Consider the applications in your shop, how you use them, their roadmaps, and whether you should make changes centered on your applications, such as upgrading or changing to new applications.

---

**Best business strategy**

It is possible that the best business strategy for migrating is one in which you combine database migration with a change in your applications that use databases. An application/database upgrade driven by changing business needs might be the best time to migrate.

---

Some of the steps in such a change might fit well with otherwise separate database migration operations. For example, are you considering moving to a different CRM or ERP product that uses a different database version than the one you are now using? Perhaps such a change would allow you to “retire” your current database to an archive after you begin using the new application.

Employee expense management is an example. If you are converting to a new application product, bring it up in parallel with the former product, but require its use for new expense reports. At some point, the prior application and its database will no longer be required online. Enforce a time beyond which all users must have adopted the new product by using it to enter current expenses, and have all outstanding expense reports on the old system approved and paid. Beyond that point, your use of the old system will be only for archival reference.

---

**Database content and use (static and dynamic)**

Database content (how the data is organized and typed) and how the data is used will affect your choice of migration tools. This is one reason why it is important that you know your database and how it is used. For example, an uninformed choice of migration tools might not accommodate all datatypes. Binary large objects and complex user-defined datatypes are examples of datatypes that are not handled well by all methods.

An important issue you will face in a transition is the downtime window. This is the time when data is being copied to the target system and cannot be modified; a time during which the database is not available to your users.

**Static data**

The term static refers to the case when the data doesn’t change, at least for extended periods. Examples:

- An On-Line Transaction Processing (OLTP) situation that can tolerate an ample downtime window
- A Decision Support System (DSS) database
- A web storefront catalog
- An archive database

---

*You might still want to migrate the original database, but it now fits the static data case, which is easier to deal with. Furthermore, you can usually conduct archival migrations at your convenience.*
If under normal circumstances, your data changes frequently, try to do the following:

- Find a window of natural or enforced idleness in which the migration can be accomplished
- Restrict write access to the database until the migration has been completed

Accomplishing the above two tasks gives you the benefit of the most important characteristic of a static database—the ability to avoid problems that might arise from processing transactions during the data copy.

The methods HP suggests for migrating static databases do allow read-only access during the time the data is being copied to the target system. This makes it possible for your users to continue to access the database during the transfer process, with the caveats that they will be unable to change data and performance might suffer due to the additional transfer activity.

**Dynamic data**

*Dynamic data* is the other major case where users need access to the database and expect or depend upon any changes they make to be reflected in the database on a timely basis. In other words, only minor downtime is permitted; downtimes on the order of tens of minutes.

It is critical to realistically and stringently re-consider requirements for database availability, so that you choose the right set of tools and methods to meet your requirements. Note that even in the strictest of 24x7 environments, it is possible, even likely, that some elements of your data are static. These can usually be moved separately, using static methods, thus reducing your potential downtime window.

Is it practical to shut down write access to your database long enough to conduct the migration? If so, you can move yourself to the easier static situation. If not, you have one of two situations: a batch update at a later time or a transactional migration.

**Two ways to do dynamic migration**

- **Transactional update of dynamic data**
  
  This situation requires that changes be reflected in a timely manner in both the source and target databases—the database is in active use during the data move and users need access to up-to-date information while the migration is going on.

- **Deferred update of dynamic data**
  
  This is the situation where you can collect and apply updates to the target database at a later time.

To achieve deferred update of dynamic data, astute programming of web application front ends can sometimes mimic the effect of transaction completion when the transactions are actually being deferred and applied at a later time. Storefronts are an example. Actual order processing can be deferred even though the shopping cart process appears to have been completed because the products would not have been shipped until a later time in any case. In that scenario, deferral (so long as it is not for too long a time) is transparent and can be used to provide time to conduct a migration.

Transactional update of dynamic data is very much like the *business continuity* situation (sometimes called *disaster tolerance/disaster recovery*), in which database changes are reflected in two separate systems, usually remotely located from one another.

If you have set up your IT environment for disaster tolerance/disaster recovery and you have a remotely-located system with your main IT operation by a data replication arrangement, a transactional update might be the best vehicle for performing the migration. On the other hand, this is a fairly complex situation—unless you already have your enterprise set up with the necessary *replication infrastructure*, HP does not recommend this approach.
In a disaster tolerance/disaster recovery situation, the two systems are usually of the same platform type, composed of similar hardware, all from the same manufacturer, and all running the same operating system. In the database migration situation however, you have your new target platform and operating system at the other end.

Not all disaster tolerance/disaster recovery products permit heterogeneous platform elements, and network arrangements become particularly important. A product which does allow different arrangements at each site is Replication Server (see the Transaction replication section for details of the dynamic data update case). Technically, this is the most complicated case and setting up such an arrangement can be a challenge. HP strongly suggests that you consult with HP and Sybase, so as to have the benefit of the latest information on issues and best practices.

What transition system environments will you need?

Among the tools HP is developing is a set of planning, assessment, and re-hosting tools to help analyze and suggest migration timeframes, baseline scenarios, impact assessments, and provisioning specifications. These will help you determine the equipment you need for your chosen migration process. You might want more than just your new target system.

Minimally, you should conduct some trial runs and tests before committing to a production data transfer. In virtually every case, such a prototyping activity turns out to be critical to a successful transition. Ideally, you should have a functional duplicate of your production system to use for prototyping. If a duplicate or a suitable subset of your production system is not available, then the production source will have to be utilized in the prototype environment.

Of course, the most irreducible element is storage. You need enough for both your target and original source databases, plus any intermediate space that your selected process might require. If speed of data transfer is critical, you will want a fast network link and perhaps also a clever storage-based maneuver (see the Backups, snapshots, and clones section) to help move data quickly.

For the actual transfer, you will most likely require a target system environment equivalent to, or a superset of, your source Tru64 UNIX system that is running the target OS environment on the platform to which you are migrating:

- HP-UX 11i systems
- Storage subsystem
- Network
- HP-UX 11i operating system
- Tools such as compilers, management, and so on

Limitations to your equipment environment might require you to consider migration processes that trade off time for storage space or network capacity.

Software elements are also important. Absent other considerations (version availability, for instance), your best chance for a trouble-free migration is to ensure that both your source and target platforms are at the same version level.

You might also find it prudent to keep your original source system available for awhile. Miscues and unexpected results might occur during your migration. Without proper verification testing, it might be some time before missed items in your migrated database are noticed. To help prevent this from happening, HP plans to publish procedures, best practices, and implementation tools targeted at minimizing these occurrences.

---

3 Future access to archives, as may be required by recent legislation (for example, Sarbanes-Oxley), may impose this.
Finally, provide time to conduct performance and tuning work on the target environment to help ensure that final operation will be efficient and scalable.

Migration steps overview

Latest software
HP strongly recommends that you ensure that the Sybase ASE RDBMS on both your source and target platforms is up-to-date with respect to the latest revision level and that you install the latest EBF's and one-off patches. Also, check the Sybase website often for new information that might affect the migration.

Current best practice is to divide the migration into several steps:
1. Set up the system storage and network.
2. Create new target environment and create database.
3. Perform the steps appropriate to your circumstance:
   - If your circumstance is static:
     a. Conduct the bulk data move
     b. Validate the bulk data move database
     c. Deal with the database completion items for production use (for example, index creation and stored procedures)
   - If your circumstance is dynamic:
     a. Materialize the replication database (that is, conduct a bulk data move and verify it)
     b. Validate the bulk data copy database
     c. Apply transactions generated while bulk data move was in progress
     d. Deal with the database completion items for production use (for example, index creation and stored procedures)

Running a prototype project to test out a subset of your migration is also useful, even critical. The experience might help you carry out the production environment migration with more certainty and confidence.

HP is preparing tools targeted to assist in each of the major migration scenarios (static and dynamic). These include technical guides, provisioning tools, scripts, and so on, as appropriate to the nature of each scenario.

The main steps of the process are sketched below, to convey the overall sequence and flow. Note however, that specific situations may differ from what is presented.

---

4 Emergency Bug Fix.
5 Do not take this to imply a requirement to use the Sybase BCP command. There are several methods of conducting a bulk data move, as will be explained later.
Step 1: Setting up the system, storage, and network

Consider the following before setting up your storage system and network:

- Will your new database (post migration) be on the same storage system or will you be setting up new storage hardware?
- Do you want to consider revising the organization of your storage subsystem to improve performance, perhaps to apply a change that you have wanted to make and for which this is the first opportunity?
- Consider how your system elements are connected—are there any bottlenecks between them that might unnecessarily increase the time needed to transfer your data?

Some migration methods employ intermediate storage. If your method does, you might need a temporary expansion of your storage system. You might decide to copy your database to storage on a separate prototype platform where you can prototype your work on the migration while your production system remains fully available to users.

On the network side, you might have selected a method that works best if a lot of network bandwidth is available. In this case, you will want to provide as much bandwidth as possible by using Gigabit Ethernet. You might also want to reserve all channel capacity for the migration by restricting access to your local area network during the transfer. You might also add another parallel network to connect just the elements used in the migration.

Some of the methods can run with multiple execution streams, allowing more utilization of CPU, memory, and I/O capacity. In some cases, you can transfer database partitions in their own execution streams—individually matched up with separate extraction and insertion processes.

If the objective is to transfer data as quickly as possible, the underlying method of arranging this is to balance the workload among available resources (CPU, memory, I/O, storage, and network). Speeding the transfer is important if you are logging updates and inserts for later application to the target database. You can’t log forever. If you have chosen a static method and asked your users to wait while you migrate the database, try to keep this window short. Develop a good estimate of the time it will take and be sure it is an acceptable time—otherwise, look into a different method.

Step 2: Create new environment and database

First, prepare the target system to receive the data. In preparation for moving data, you need to create the database on the target system:

1. Create named segments (if any).
2. Create tables.
3. Create the objects and definitions that will be required by the bulk data move methods you have chosen (for example, stored procedures).

After the target database has been created, conduct the bulk data move and then conduct exercises to verify that the database has been migrated successfully (see the Verification section).

Step 3: Static or dynamic-specific tasks

Static case

If your database does not include complex datatypes and you can fit your migration into an available downtime window, you might be able to use one of the Sybase data move utilities for the whole process. That is, after appropriate preparation, you conduct your bulk data move using one of the utilities and verify the move.
Additional steps might be needed to complete operations that are sometimes better left until after the move. For example, in most cases, application of constraints and creation of indexes should be deferred until after the bulk move. Details of which fits where and what is recommended for each situation are described in following sections.

There are several Sybase tools that you can use to conduct the bulk data move:

- BCP (Sybase Bulk Copy Program)
- CIS (Sybase Component Integration Services)
- DDLGEN (Sybase DDL Generator Utility)
- Sybmigrate (Sybase Migration Utility)

See the Sybase utilities section for more details on each of the utilities.

**Dynamic case**

If your situation is the dynamic case, you will need to divide the process into several steps. Essentially, this consists of preparation followed by a bulk data move, verification that the bulk data move was successful, and finally, application of transactions that took place while the move and verification were being done.

There are two main situations that must be addressed as part of your preparations for a dynamic migration:

- **Transactions can be deferred**—that is, during the bulk data move, transactions can be collected in a log file or other buffer and then applied to the database at a later time. Before that time, the effects of transactions do not appear in the database.

- **Changes must appear promptly**—it is necessary that the effects of transactions appear quickly. As explained later on, this case gets particularly difficult to handle. In this case, use of transaction replication is called for (see the Transaction replication section).

**Sybase utilities**

Sybase offers three different utility programs which are applicable to the bulk data copy phase of a database migration.

**Sybase Bulk Copy Program**

The Sybase Bulk Copy Program (BCP) is used to extract data from one instance of Advanced Server (your source Tru64 UNIX environment) to another instance of Advanced Server (your target environment), that is, from one server to another.

Because BCP records to and reads from a flat file, sufficient platform independence is naturally provided for BCP to be used across different platforms for most types of data. Because data is converted to and from ASCII in the most common flat file data types, most cross-platform conversions (Endian conversions, for example) are dealt with naturally by the platform-neutral properties of flat file data representations.

---

6 The most familiar flat file examples are ASCII text files consisting of SQL statements, or files with data in a fixed format or delimited by some punctuation character (for example, the familiar “csv” or comma-separated variable file). A fine point, but worth mentioning, is that a “flat file” does not necessarily have to be stored on a disk. The term may also be used to describe certain data being sent via a pipe over a LAN. It is the nature of the “content” that is the essential element.

7 “Endian” refers to the way in which multi-byte data types are stored. Tru64 UNIX, for example, stores the least-significant byte (“little end”) at the lowest address, whereas HP-UX stores the most significant byte (“big end”) at the lowest address.
Note that a flat file does not necessarily have to be composed of only ASCII text—flat files containing binary data are also possible. In these cases though, BCP needs to be informed about details of how the file is formatted. With information about column formatting and row termination characters, BCP can distinguish and convert most Sybase datatypes. This formatting information is provided to BCP using format files. This information can be collected for you by BCP and used to help manage the bulk data copy (see Reference 5).

In addition to assuring the correct conversion of datatypes, the amount of time required is also of interest—your users are waiting for the migration to be finished in order to regain access to their data. There are several ways in which a BCP transfer can be arranged—it varies according to which server BCP is run on and in which OS file system the flat file is located. The file can even be moved via named pipes. These have differing degrees of transfer speed and convenience. Three arrangements are illustrated in Figures 1 through 3.

The named pipes method was about 20% faster than the other two methods in HP’s preliminary tests. This was in a Symmetric Multi-Processor situation in which BCP was executed as a client of the source database server. Actual results will depend on many factors like platform processor speed and memory, network type, storage system characteristics, and the data types used in the database (some require more conversion computation than others).

The BCP utility has two speed modes over which you have only some control. In the slow mode, each row insert is logged; in the fast mode, only page allocations are logged. The fast mode can only be used when handling target tables that do not have indexes or triggers. Slow mode is used automatically if the target table being imported contains indexes or enabled triggers (you can control this by means of alter table disable trigger commands; indexes are another issue). Fast mode is much better than slow for migration, so much so that it is better to leave indexes behind and rebuild them after the bulk data copy phase is done. Furthermore, slow mode logging may fill up your storage too quickly and cause additional complications.

Although Sybase does not have a direct method for initiating parallel export operations on a table, this can be accomplished indirectly to good effect by exporting from views. Assuming that the data can be divided into two or more separate parts using ranges of a clustered index, a view of each part can be created. Separate executions of BCP from each view can then be initiated.

Other optimizations of BCP are possible (such as adjustments to cache behavior and specifying larger network packet sizes). They may be worth consideration if your database is very large and you absolutely need to minimize the time required for the bulk data copy.

![Figure 1: Export of BCP files to source platform and FTP transfer to target platform](image)
Sybase Component Integration Services

Component Integration Services (CIS) provides for remote object query through a proxy structure on the local server. This is the kind of thing you could do using Remote Procedure Calls if you wanted to establish customized access to database tables located on a different server (in order to move them, for example). The CIS utility makes it easier. Basically, it makes it possible to access remote tables as if they were local. From the target server, you can query database tables on the source Tru64 UNIX server and copy them to the migrated database on the target server.
Figure 4 illustrates the arrangement. When a proxy database is set up on the target server, proxy table metadata is imported from the source server and used to create the proxy tables, which can be referenced locally. It works cross-platform, and administration and implementation are relatively easy. There is also less need for external temporary storage I/O than with BCP. CIS is an integral part of Adaptive Server Enterprise and is therefore available to all customers. CIS is described fully in Reference 9, the Component Integration Services User Guide.

HP found CIS to be a fairly fast method of migration.

Sybase migration utility

As stated in Reference 8 (Chapter 49), sybmigrate for Adaptive Server 12.5.0.3 and later provides for cross-platform migration. Originally, this migration utility was used to migrate to the new page sizes9 supported by ASE 12.5. Sybmigrate combines transparently with some other Sybase utilities to carry out much of the work of cross-platform migration. For example, the DDLGEN tool and CIS are used within sybmigrate to capture and recreate the source database schema on the target platform and then to transfer the data. In addition to this functionality, it is equipped with a graphical user interface10.

sybmigrate
This utility is now the recommended, preferred tool for cross-platform static migrations like the present case.

Some cautions: Make sure your database is sound—don’t try to migrate a “broken” database. Make backups, conduct dbcc full checks, and fix any problems before you launch sybmigrate. Also bear in mind that sybmigrate is for the strictly static migration case.

---

9 This reference adds new chapters at the beginning of the document as new ASE releases are made. As a result, chapter numbers tend to “migrate.” See, for example, chapter 29, which is the original documentation for sybmigrate and which is also valuable for background information on its purpose and operation.

10 Earlier releases supported only a 2K page size. ASE 12.5 supports up to 16K page sizes, which allow for longer rows with potential performance improvements.

You could use DDLGEN and BCP to carry out these operations manually, but this can be complex and tricky to manage. Sybmigrate relieves most of these issues.
If something goes wrong during the sybmigrate run and you find yourself obliged to restart the process, it is critical that you remove partially migrated tables from the target platform before you resume. Sybmigrate will fail if a table already contains data.

Chapter 29 of Reference 8 contains comprehensive lists of what sybmigrate does and does not do. You should review this reference to see what cautions apply to your particular databases and also to see what items will need to be addressed after the “automatic” parts of the migration process have been conducted. It also discusses a number of optimizations and configuration items which you must deal with in order for sybmigrate to operate properly. For example, worker processes and user connections must be balanced when you are paralleling transfer of partitioned tables—worker processes send and user connections receive.

Several options for optimization are available. For example, the parameter CIS bulk insert batch size controls how many rows are transferred before the transaction is committed. This can have a significant effect on data transfer rate. A caution however, is that increasing this results in increased use of memory on the source platform. This may affect other operations. Finally, tables with text or image columns are not helped by this optimization tactic because rows from such tables are always handled one row at a time.

Cross Platform Dump and Load
Starting with ASE 12.5.3, Sybase expects to offer a version of the Cross Platform Dump and Load utility that you can use with platforms of different architectures, such as those having endian differences. Details will be provided in a future update to this paper.

Transaction replication

Replication methods are the outgrowth of earlier methods which provide standby databases to protect against business interruption. Because replication methods can create and maintain databases on heterogeneous platforms, they can be adapted to serve as database migration tools. However, there are several considerations to keep in mind.

The practice of database replication recognizes two major types: physical and logical. How they differ is important to understanding some of the necessary characteristics that make replication systems practical as migration tools.

During operation, as application transactions change the source database, the changes are recorded in transaction logs, which are then employed to deliver and apply change information to the standby database.

- For physical standby databases, changes are applied in ways which assume particular physical similarities between the source and target systems (same endian type, for example).
- For logical replica databases, transaction logs are used to generate SQL statements, which are then used to apply the changes.

Physical standby databases must be identical to their source databases, block-for-block, because recovery operations use physical information to apply changes. Dump and load methods are an example. While they can often be used for backup or copying purposes, dump files cannot be used in an environment that is not exactly like the one in which they were created (that is, both platforms must use the same kind of hardware and operating system).
Logical standby databases are relieved of the physical identity requirement because transaction log records are converted into T-SQL transactions before applying them to the standby database. The database can be physically different from the source database, with different indexes and physical characteristics, while remaining logically consistent with the source database and with applications that use the standby database. However, this is achieved at the cost of added processing.

The purpose of replication is to create and maintain (on-the-fly, in real-time, more or less) a functional duplicate of a database in parallel with transaction activity; while the database is in use, inserts, deletes, and updates enabled. When such a functional duplicate can be created practically on a different platform, the net effect is to have migrated the original database to a new system. This is the situation in which write access to your database must be essentially uninterrupted.

This process will need a sort of jump start though, because it would not be practical to try to build the target database one transaction at a time going back to the beginning of time. It is necessary to conduct a bulk data move at some early point. In the case of Replication Server, this is known as materialization and is accomplished with one of four materialization methods available under Replication Server (see Chapter 10 of Reference 15).

Replication databases
Starting from functionally duplicate databases on separate systems, replication arrangements ensure that the effects of all transactions are reproduced in replication databases, so that they are synchronized. This is what you need to provide if you must keep your database available during the migration—this is how you ensure that changes to your source database are accurately replicated in the target database when the migration has been completed and your users have been switched over.

Although it may not work out for every environment, replication may sometimes simplify switching your applications over to the new database and platform environment. Once you have materialized your target platform database, and brought it up to synchronization after testing it to verify that it was properly migrated, you can perform the (often) straightforward operations needed to point the application at the new database. After that, you can start up your applications on their new target platform servers, point them at the new target platform database, and move users to the new platform application servers.

Replication arrangements can be driven in several ways. For example, the application using the database might direct its transactions at both databases, achieving the result of functional identity. Some middleware tools can also provide this functionality—the HP Reliable Transaction Router is an example. And finally, replication can be provided as a function of the RDBMS and one or more of its accessory or utility products.

ASE Replicator is not an option for dynamic migration methods. ASE Replicator is a limited function tool for simple situations in low traffic environments and does not provide the comprehensive coverage or transaction handling speed needed in an enterprise environment. Replication Server is more likely what is needed in a dynamic migration situation.
Replication Server has different functions to satisfy a variety of replication objectives, such as distribution to distributed databases. For the migration situation, the most applicable is a mode of use called primary copy. Reference 13 provides a useful overview of Replication Server’s use in the context of upgrading ASE databases, a context which corresponds in many respects to a migration. This reference includes a step-by-step discussion of preparing and using Replication Server in a warm standby situation. (Pertinent and more specific details for the primary copy method are found in References 14 through 16. Reference 13 conveys a summary understanding of the replication process, but note that the dump method depends on binary identity; for migration, the primary copy model must be used instead.) Figure 5 illustrates the general organization of elements and data flow in a Replication Server arrangement.

A caution: in a very high transaction rate environment, Replication Server may have difficulty keeping up with transactions on both the active and standby (target platform) databases. This depends a great deal on the capacity of your network, the speeds and numbers of processors on your source and target platforms, the relative performance capabilities of the processors, and the performance of your storage systems. You may want to conduct some throughput tests beforehand and balance them against the transaction rates you experience in your production environments to see if this issue will be of concern. Although it is unlikely in any but the most severe environments, it is good to know whether your environment is relatively benign.

1. Subscription is activated with suspension to target DB. Active transactions are queued in the outbound stable queue.
2. Data is “copied” into target DB via Sybase load tool (CIS, BCP, etc.).
3. Set auto correction on table to resolve any inconsistencies.
4. Resume connection to target DB. Queued transactions are applied to target DB.

Figure 5. ASE Replication Server data flow

1 For example, the context of the reference assumes one can use dump and load operations to materialize the standby database. In the case of a platform migration, however, materialization must be conducted using methods that do not assume binary identity and that manage the necessary conversions (deal with endian issues, for example). That is, BCP, CIS or sybmigrate (which is the recommended method).
Backups, snapshots, and clones

Although backup, snapshot, and clone processes can’t fulfill critical requirements for a complete inter-platform migration, they can still serve a useful role. The critical point here is that migration processes and equipment must be able to work in a practical way between disparate platforms and operating systems.

For example, by design, backup systems are supposed to restore data that is an exact copy or duplicate of the original. That is not exactly what is called for when performing a database migration. While the database should look the same to applications on either the source or target systems, the database on the source system cannot be physically identical to the one on the target system. Simple endian differences are enough to make the databases differ physically, even though data extracted from one would look the same as data extracted from the other.

And while most restore processes will let you restore to some point other than the original backup point, they will not get inside a database and adjust paths and files to account for their new locations. They will not make compensating adjustments for database block and extent coding, nor will they make appropriate endian adjustments.

So are backups, snapshots, or clones of any use during a database migration? The answer is yes. If you have a storage subsystem with adequate capacity and functionality, you might use them to create an intermediate copy of the database. You can then use it to reduce contention for storage resources and CPU resources.

A backup, snapshot, or clone can be useful in database migration because you can use them for any of the following, depending on the overall migration process you have chosen:

- To capture a copy of your database with which to work as you conduct your prototyping
- To quickly move files that do not require conversion for use on the target platform

Verification

There are many aspects of a database that you can use to determine confidence in the viability and integrity of a database following its move to a new platform. This section summarizes some useful verification techniques.

Note that it is generally useful to capture as much information as you can about your present database before the migration. This information can be useful in verifying the completeness and accuracy of the migration. For example, post-transfer, the BCP utility will tell you how many rows were copied successfully and how many were not.

Sybase database consistency checker tool – dbcc

This tool provides several useful operations to check confidence in the integrity and consistency of a database which has been migrated. (It is a good idea to check the consistency of the pre-migration database as well because damaged databases rarely migrate successfully. It is better to find out beforehand.)

The following are several functions of dbcc that may prove useful in verifying a migration:

- dbcc checkdb: Checks data and index page integrity
- dbcc checkalloc: Checks allocations across a database
- dbcc checkcatalog: Checks system catalog consistency
- dbcc checkstorage: Checks database object consistency (combines checks above)
Chapter 25 of Reference 4 provides descriptions of how to use dbcc, with explanations of the underlying structures.

**Sybase optimizer statistics diagnostic tool – optdiag**

The Sybase utility optdiag (see Reference 17) provides several methods of analyzing database component parts and elements based on statistics. Note that these help to ensure accurate replication of structure and quantity of database elements, but do not ensure element accuracy. The results of analyses on both the source and target database can be compared to check for substantive differences that may not have been disclosed by an integrity check such as dbcc. Analyses should be run against the pre-migration database and the results saved for comparison to the results of analyses against the post-bulk copy database.

Various points to check are whether all source database users are present on the target, whether columns of numbers sum to the same values, whether the number of rows in tables are the same, that data entry constraints are operating, and other such easily tested characteristics that should be preserved following the migration.

Different parameter settings on source and target systems may induce differences in these reports, so comparison may require some interpretation.

**Sybase Central**

You can use the tool Sybase Central to examine various characteristics of the migrated database and compare them to the pre-migration database. This is a plug-in GUI tool which provides, among many other things, for straightforward examination of object property sheets and dependency lists. This is useful for such verification activities as checking how user-defined datatypes were handled.

**Your application**

The reason you are using a database in the first place is because it supports some application. Gather application output from the source system for comparison to results from the target system. Set up an instance of your application and point it at the newly migrated database. Use it to perform some task that is part of its normal functionality and that uses data from the database. Does it behave normally? Are the results what you would expect to have received on the pre-migration system? Exercise all of the application’s user types’ assigned functions to be sure roles, grants, and privileges were migrated properly. After due consideration for performance differences that you would expect between your source and your target systems, did the queries or reports seem to take an appropriate amount of time (this is an indication that your indexes are good)?

**Summary**

The following tables summarize static and dynamic situations and the tools to use in variations of those situations. Note that they do not portray all factors in detail. As work continues, this information will be updated with new findings. In the meantime, go to the Alpha RetainTrust website for the latest information:

http://www.hp.com/go/alpha-retaintrust

For the latest transition planning information, go to:

http://www.hp.com/go/transition-modules/
### Figure 6. Characteristics of migration scenarios

<table>
<thead>
<tr>
<th>Static situation</th>
<th>BCP</th>
<th>CIS</th>
<th>Sybmigrate</th>
<th>Dump/Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower version to 12.5.0.2</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version 12.5.0.3 to 12.5.2</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Version 12.5.3 and above</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️ ✔️</td>
<td>✔️ ✔️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic situation</th>
<th>Replication Server with BCP</th>
<th>Replication Server with CIS</th>
<th>Replication Server with Dump/Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower version to 12.5.2</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Version 12.5.3 and above</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Further information

1. Tru64 UNIX to HP-UX 11i transition modules: http://www.hp.com/go/transition-modules/

2. From Tru64 UNIX on Alpha to HP-UX on the Itanium Architecture, HP white paper: http://www.tru64unix.compaq.com/transition/pdf/unixmigration_wp_902.pdf


6. Replication Server, An Architecture for Distributing and Sharing Corporate Information, Sybase white paper

7. Sybase Adaptive Server Enterprise 12.5, Sybase white paper


10. Sybase Replication Server Performance and Tuning, Sybase white paper, September 2001


13. Upgrading Adaptive Server Enterprise (ASE) with Replication Server Warm Standby, Sybase white paper


© 2004 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Itanium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Sybase, Adaptive Server, Replication Server and Transact-SQL are registered trademarks of Sybase, Inc. UNIX is a registered trademark of The Open Group.