Migration Management Methodologies

Suzanne T. Perry *
Statistics of Income Division
Internal Revenue Service

Abstract

An important task in managing any computer system is the adoption of upgraded versions of application software. Such upgrades usually require the migration of user application programmes from one version to another. This paper specifically addresses a migration from ORACLE version 5.1.22 to version 6.0 in Digital Equipment's Unix-based operating system, Ultrix.

To describe this specific task, a general framework for accomplishing such migrations has been developed. Resolving hardware/software compatibility issues are addressed along with determining the degree of transparency to the user. Evaluating the extent of modification to current user applications with regard to reprogramming, to fully utilize all the new features, will also be discussed.

Introduction

In any computing environment, hardware and operating system combined, the migration of a software package from one version to another, should be undertaken very cautiously. There are a myriad of details which must be researched prior to the installation, or if at all possible, prior to the purchase of the new software. Indeed, a number of questions should be asked, such as: Will the cost benefit ratio be significant enough to warrant the purchase of the new software version? What will the impact be on the other software currently running on the system? What will the anticipated expenditures be for retraining and reprogramming to accommodate the upgrade? The aforementioned list represents issues that need attention and must be resolved prior to purchase and installation.

In the situation described in this paper, the decision to migrate from ORACLE version 5 to version 6 was based solely on the need for greater performance from the database on the existing hardware/operating system platform. Due to table locking, the size of the tables, the amount of user exits needed for consistency tests, and the operating system I/O limitations, version 6 with TPO appeared to be the best solution. Of course, the acquisition of version 6 with TPO unquestionably impacted the whole computing environment, from the operating system through the major software packages on the system. Also, any problems that would be encountered were compounded two-fold, because there are three duplicate computer systems in use. Hence, there was the need for a complete and methodical approach to the migration.

Background Information

The Statistics of Income Division (SOI) of the Internal Revenue Service is concerned with producing statistical tables which identify trends in income reporting from the individual taxpayer to giant corporate accounts. Up until 1987 these studies were processed in a mainframe environment. When SOI purchased three Digital Equipment Vax 11/780's in 1985, the Division started to move the processing of some studies to the new minicomputers. Ultrix 2.2 was the operating system, ORACLE version 5.1.17 the relational database software and SPSSX which was used for statistical table formatting.

The first projects to migrate to the new computing environment were relatively small scale. Each study averaged 20 tables with 5000 rows of data in 35 data columns. The one exception in the above group was the Sales of Capital Assets (SOCA) study. In this particular study, there were approximately 30,000 returns utilizing 25 tables with an average of 50 data columns per table. Ordinarily this would not be a source of great concern, but, in fact, it was during the actual data entry/update phase that systemic problems were initially encountered. The data that are edited from the tax forms are not merely inserted/updated into the database tables, but rather Oracle's SQL*Forms is used to perform data validation

and consistency testing before the data are committed to the database. An example of a consistency test may include the addition and/or subtraction of five data fields, the result of which is compared to the values present in three other data fields (that may or may not necessarily be present in the same table). Incorrect values must immediately be identified to the editor (user) so that keying errors can be corrected or accept-coded (in which case a flag must be set to indicate further evaluation of the return is needed). Consistency tests vary in level of complexity and frequency with respect to the data fields. Since the editors/users process tax returns in the same manner and at about the same rate, it was found that commits were being performed at the same time. Hence, an unmanageable dead-lock situation was produced. Some of the locking problems were alleviated with the increased use of "C" program user exits, but the response time of the applications on the existing hardware/software platforms was still at an unacceptable level. It was also determined that fifteen editors using SQL*Forms applications and 5 programmers doing normal compilations and file editing could significantly slow down the system.

In conjunction with Oracle consultants, who had previously worked on SOI programs, all possible solutions were examined and the final decision was made to streamline the existing application programs as much as possible for the current processing year and to purchase Oracle version 6 with the Transaction Processing Option (TPO). It was understood that the table-level locking problems would be resolved, since no two editors ever worked on the same return simultaneously, therefore also decreasing the possibility of ever having row level locking.

Phases In Migration Processing

In order to accomplish the above migration a methodical approach was developed. This methodological approach can be adapted as needed to any migration process. The number and content of these steps were developed with respect to the logical division and sequencing of the tasks at hand. Also, included with each step outlined are brief notes on the problems and solutions encountered during the SOI migration process.

Step 1 Compatibility Issues

The initial step in the migration process is to identify and resolve any compatibility issues among the hardware platform, the operating system version, and all software package versions. Particular attention should be paid to making sure that versions of the software packages agree and that if at all possible different versions may be run simultaneously.

With regard to the hardware platform, no problems were encountered because Oracle version 6 is supported on Vax 11/780's.

The operating system version, on the other hand, posed a few dilemmas. Oracle version 6 required that the operating system be Ultrix 3.0 or higher. Oracle version 5.1.22 was not fully supported on Ultrix 3.0. The problem at hand was one of timing on the part of everyone concerned. Since the delivery date of Oracle version 6 (production) was unknown, the only solution that presented itself was running the two versions of Oracle at the same time. It was soon discovered that in order to fully operate version 5.1.22 under Ultrix 3.0, a patch tape from Oracle was required. Once the patch tape was received, the operating system was upgraded to Ultrix version 3.0 and version 5.1.22 was reloaded along with the patch tape. The system was then ready for the installation of version 6.0.

A new version of SPSSX was also purchased which also required Ultrix 3.0. After the operating system was upgraded, the new version of SPSSX was loaded. All other minor software packages were reloaded and experienced no operational problems during the operating system upgrade.

Documentation of the above problems and solutions was initiated to aid in future migrations.

Step 2 Degree of Transparency

In the second phase of the process it is necessary to determine the degree of transparency of the intended migration to the user community. Depending on the production environment, user transparency needs vary from nonexistent to a maximum need for full transparency. This degree of transparency decided on should be based on the system involved and should coincide with any pre-scheduled system downtimes. This must then be used as a base reference point for all succeeding steps and the decisions related to them.

It was decided that the above migration would be performed on the system which had the least

amount of impact on production processing. User transparency is of the utmost importance in the SOI production environment, and any computer downtime must be kept to an absolute minimum. For example, attended system backups, which are done daily, are always scheduled for between shift changes or after the evening shift ends. All hardware and software upgrades are scheduled for weekends, usually resulting in complete user transparency. The nature of SOI processing is such that tax year changes are staggered for each application throughout the year leaving approximately one month when there is no active computer processing. Uncertain delivery dates of version 6 software, coupled with the need for maximum user transparency, precipitated the decision that all existing applications would continue to operate under version 5.1.22 for the duration of their systemic life cycle. Studies which were currently in the design phase and all subsequent studies would be developed using version 6, again being dependent on delivery dates. The studies that experienced only tax year changes would be the applications migrated and wouldn't be brought on-line in the production environment until after complete testing.

It is believed that the above approach allowed for the maximum amount of time for the retraining of the developers and DBA's, as needed, while the effect on production was kept to a minimum.

Step 3 Modification Compatibility

The third phase of a migration process is to define the degree of modification needed for current user applications for complete compatibility with the new version. Research should be conducted into all changes between the two versions of software. The results of this research must then be conditioned to the application needs and processing environmental needs. The best approach is, first to accumulate a list of all the version changes, then to eliminate those which would have no impact on the application or environment. With only the pertinent changes at hand, this step will become more manageable.

Due to the nature of differences between version 5 and version 6, most of the application changes that needed to be made—such as the effect of the new reserved word list—were minimal and it was determined that the changes could be made quickly.

The DBA's function, however, will experience the greatest amount of changes with the new version 6 software. For example, their focus has evolved from trying to keep the database tables sufficiently large, with a minimal number of database files linked, to the dilemma now faced as one of moving database files across the disk drives for efficient operations. Also, indexes which basically stayed in the same partitions as the tables they were related to now needs to be moved to promote system performance and to reduce head contention on the disk drives.

It was found that a detailed description of all applications for each computer system should be drawn up. The number of tables, columns, and rows of data need to be listed along with the indexes associated with each application. Frequency of use, for each table and index, should also be listed as in the case of SOI applications. (For example, a tax return may have multiple Schedule C's attached to one and only one Form 1040 tax form.) These lists helped the DBA plan for the use of the database tables and to evenly distribute the work load across the disk drives.

Step 4 Migration Scheduling

The next step in the process involves estimating migration time frames and scheduling the conversion process with respect to degree of migration transparency. For this task, PERT charts should be developed with respect to the system and the nature of the resident applications. The charts are immensely helpful when planning for programming manpower needs and keeping management and the user community informed and involved.

In the case of SOI, the charts spanned several months because of the special needs of each of the studies as mentioned earlier. In general, each of the charts contained a calendar with the migration initiation date being a base point. Each current study end date was listed, along with those studies currently in the design phase and their expected production dates. Any anticipated new studies were listed to complete the frame of reference for the migration process. It was found that by using these charts when the delivery dates changed, time frames could be moved with ease, thus facilitating the migration process even more.

Step 5 Migration Process

During the fifth phase, the actual physical migration process must be initiated with the successful uploading of the new Oracle software version, according to Oracle's documented procedures. The

new database structure should then be checked and validated to ensure proper working condition.

The majority of errors occurring at this point were due to misunderstanding the documentation and not precisely following the outlined procedures, as is the case in any migration process. It was found, however, that due to the great degree of advanced planning and design of this process, even the unexpected problems were not that difficult to deal with. For the effected applications, two developer/programmers always worked in tandem to verify the work. Although this may seem to be a redundant work effort, it proved to be invaluable in reducing syntax errors.

Step 6 Verifying System Operability

Verifying the operability of the migrated system by identifying and correcting any major problems is the next logical step. All modified application programs should then be brought on line. The users individual environments should then be adjusted to access the newly migrated Oracle kernel. Again, minor system malfunctions need to be identified and corrected.

Since two versions of Oracle were operating simultaneously, the editors/users that worked on different applications found very little difference in the "look" (appearance) of the newly migrated applications. The majority of the editors/users noted that the commit rates had vastly improved over the version 5.1.22 applications. The only (problem?) situations encountered at this stage was that the editors/users only wanted to work on the migrated applications because they felt that they were making more efficient use of their time.

Step 7 Documentation

Documentation of the entire migration process should be completed as the final phase of this method. Development of a revised implementation plan may need to be accomplished if there are other installations which will be undergoing the same migration of software.

The complete documentation package was delivered to all sites. Notes to the system administrator and database administrator were inserted notating the changes that should be made specific to their installation. Also noted were the areas where flexibility existed in the migration process dependent upon their particular application environment.

Conclusions

In reviewing this migration management methodology it becomes very apparent that an exceptional amount of emphasis is placed on the planning and design phases. At the same time, what is also very apparent is that the actual migration produced very few unmanageable and unplanned for problems. It is highly recommended that some type of similar methodology, whether it is based on this particular plan or not, must be devised for any migration process to insure success.

*This paper was originally presented at the Oracle 1989 International User Week Conference in Dallas, Texas on October 5, 1989.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1991 — 287-084/21322