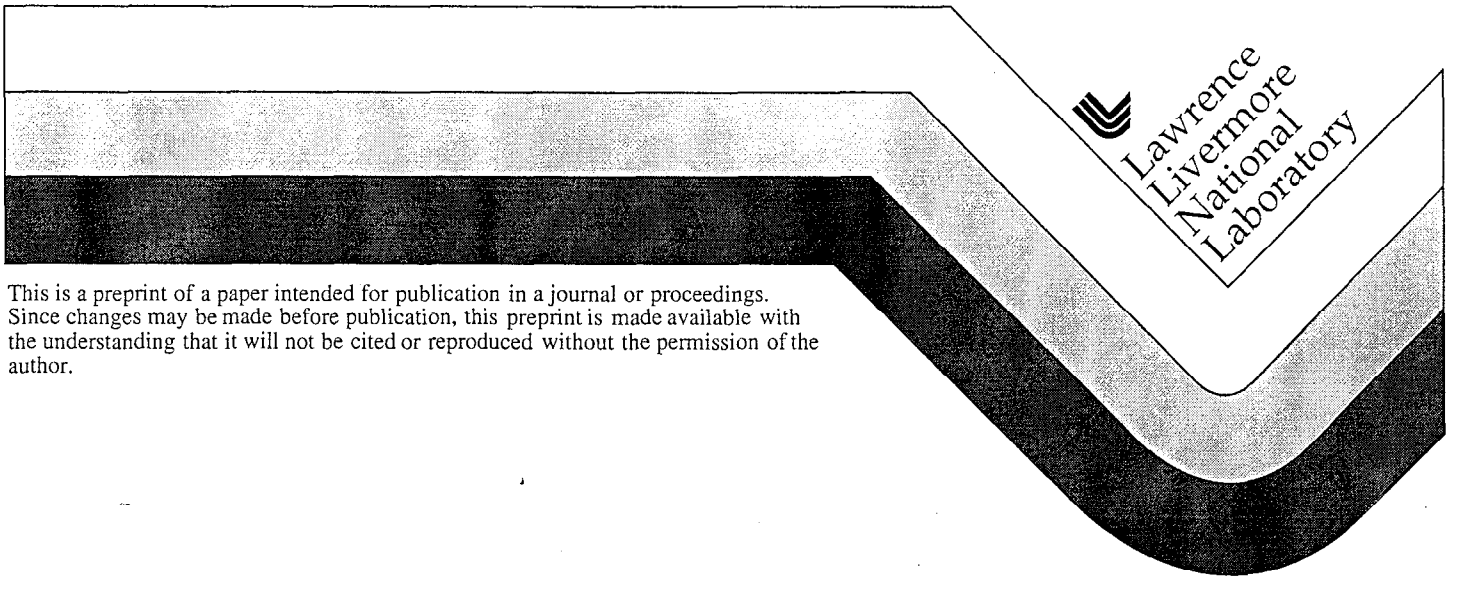
UCRL-JC-133071
PREPRINT

Sizing Oracle Applications Projects, A Technical Perspective

Andrew Rivenes

This paper was prepared for submittal to the
Oracle Applications Users Group
San Diego, CA
April 16 - 22, 1999

February 1999



Sizing Oracle Applications Projects, A Technical Perspective

Andrew Rivenes
Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Sizing Oracle Applications Projects, A Technical Perspective

Andrew Rivenes
Lawrence Livermore National Laboratory

Abstract

This paper will review the issues involved in sizing an Oracle Applications project. The topics will include initial machine sizing, the different environments required, refresh procedures, and final system sizing. Installation and patching requirements will be addressed along with a methodology for handling long term capacity planning.

Overview

The purpose of this paper is to address the issues of sizing an Oracle Applications project and highlight the issues involved from a technical perspective. As an Oracle Applications DBA I have assisted with numerous Oracle Applications project implementations and invariably technical architecture issues surface that cause frustrations or delays the project. If more technical architecture planning was done at the beginning of a project many of these problems could be avoided or dealt with in a more efficient manner.

For this paper, sizing an Oracle Applications project is defined as providing the appropriate Oracle Applications systems (e.g. a development system, a test system, a mapping system, etc.) to support the project and identifying the resources required to create a production system. In practice this analysis can be a very complex undertaking. Projects are frequently started with specific funding and even specific hardware environments pre-defined. However, this is typically done before a thorough understanding of the final system has been made or even before the project has been formally created and staffed. One of the principal goals of this paper is to show that these decisions must be made as part of the project. In fact, the project plan must be geared to identifying the needed resources throughout each of the project phases and incorporating these "unknowns" into the project timeline. The final production system hardware specifications should be determined as a result of the project's technical architecture findings.

After all, how can an organization identify and purchase a system's hardware before knowing the system's requirements? Will the system have enough memory and disk space? Will the disk space support the I/O call rates required by the application? What are the I/O call rates? Will hardware redundancy be required? What about CPU requirements? One of the goals of the project must be to define these requirements so that the final production system can support the intended service level of the application. This extremely complex task is most often underestimated by the initial project planners. Many projects allocate insufficient resources to define these specifications, and yet this is one of the most common reasons that clients are disappointed with the results of their Oracle Applications projects.

Why is sizing so hard? Invariably someone asks the question, "With all of the Oracle Applications systems that have been implemented why is it so hard to size my system?" The basic answer boils down to the fact that Oracle Applications is a complex system that provides great flexibility. No two installations are ever quite the same. Business practices such as how the charts of accounts is defined or allowing the reuse of account codes can have a dramatic impact on the performance of the system.

The ultimate technical goal of an Oracle Applications system implementation is to create a production system that meets performance, growth, and system availability requirements. In order to accomplish this goal, information about the system implementation must be gathered, analyzed, and tested. During this process the technical staff must also support the needs of the rest of the project team by creating and maintaining environments to perform analysis, design, build, test, and training.

The challenge to the project manager is to build all of these requirements into the project plan in order to avoid surprises and the need to "go back to the well" to ask for more hardware, time, or resources.

Project Planning

When sizing an Oracle Applications project, organizations must consider more than the hardware sizing. They must also consider the number of technical staff, the timing of the technical support for each project phase, and sufficient lead time for hardware procurements, installations, benchmarking, stress testing, and technical architecture design (e.g. backups, recovery, operational support, printing, etc.). There also must be enough time allocated between the time the production sizing requirements are identified and the purchase, installation, and software setups are completed.

In general I have found that not enough resources are dedicated to the technical staff and that it is assumed that the DBA tasks are generic to any install. While some of the DBA tasks are generic to any Oracle Applications project there is a lost opportunity for insuring project success if resources and tasks are not defined for a Technical Architecture study. This activity should also include a production system sizing and stress testing study. It is imperative to understand the need to differentiate the environment setups required to support the project versus the effort required to size the production system. Very often, unless the project is very large or the project managers very experienced, the Technical Architecture tasks are not resourced adequately to perform such an analysis. The actual production system sizing essentially requires a project within a project.

Often, project managers treat production system sizing as merely another step in the Technical Architecture tasks requiring the calculation of disk space, CPU cycles, and memory usage based on generic sizing spreadsheets. Issues related to workload characterization[1] and the system's ability to meet peak workload demands are not considered, and no modeling or predictions are ever made. There is a tendency to expect that the DBA will install the system and then go away until needed again, and that the functional team will require all of the project's resources.

Once the analysis phase is complete and the design and build phases begin there should be enough information available to start the design of the production system. Business volumes, client volumes, and expected usage should be known, and service level requirements should be defined. Based on this information an analysis of the production system's specifications can begin. A key decision will need to be made as to the confidence level[1] of the final result. Since the primary goal of this analysis is to determine the production system architecture (e.g. define the production system hardware and capacity) how much effort and expense is actually invested must be determined based on the preciseness[1] required of the result. This issue must be addressed at project inception rather than after the fact. In order to perform an adequate Technical Architecture study, sufficient time and resources must be made available. It also does little good to perform the study if the production hardware decisions have already been made. There have been many good papers and books written on sizing and capacity planning. The best advice is to plan on adding a specialist to help setup or advise on the creation of the study and definition of the tasks.

The risk for not properly sizing a system is that the system will fail to perform or there won't be enough capacity to support the workload. The old adage that "we can just throw hardware at the problem" can work fine for small to medium sized systems. For larger systems there may not be hardware big or affordable enough to buy more capacity. And it may not just be a question of adding more hardware that "masks" a performance problem. There may be fundamental architecture problems with the design, and unless a sizing study is performed these defects may not get detected.

Technical Resource Requirements

When building the project plan it is important to consider the resources required to install and setup each of the Oracle Applications environments required for the project. Factors to consider are the time required to perform machine setup, the installation of the Oracle Applications software, the installation and configuration of the client software (including application server(s), NCA or SmartClient), patch application, and operational configuration (e.g. setup of backups, support for start/stop/monitoring, printer setups, help desk functions, etc.) The time required should not be underestimated, especially if the site is new to Oracle and/or Oracle Applications. The technical expertise and availability of the project staff must also be considered.

---

[1] These terms are taken directly from "Pedicting Computing System Capacity and Throughput", see Sources.

Environment Setup Issues

If the site is new to Oracle and/or Oracle Applications then the hardware and support infrastructure may not exist to support the project or the final system. The project plan must factor these issues into the timeline even if the actual responsibility is out of the project scope.

Technical Expertise

Organizations and project managers must consider the technical expertise of the project staff. Oracle Applications is a complex system, both from the functional as well as the technical perspective. With the addition of SmartClient and now NCA, Oracle Applications has grown increasingly complex with the need for expertise in networking, printing, web servers/browsers, desktop workstations, DBA skills and Oracle Applications system administration. Release 11 does not alleviate this problem as it now requires the use of the NCA technology stack.

A typical new Oracle Applications Release 10.7 NCA installation may take an experienced installer one to two weeks, but a new installer can take much longer. Depending on the product mix and environment additional time may be required to apply application patches, configure system administration options, and support general operational issues.

Technical expertise will also have a great impact on the quality of the overall technical architecture and the required supporting documentation. Understanding the architecture required to support the service level, system availability, recoverability and then being able to communicate this information is a valuable skill set that takes experienced individuals.

Project Technical Architecture

The components of a project's technical architecture should be designed to provide the documentation and specifications required to deliver the final production system. It is this process that must address all of the technical issues required for a successful implementation. The following technical architecture tasks are typical of many implementations:

- Define scope and objectives
- Prepare an architecture strategy
- Establish requirements
- Develop a conceptual model
- Conduct an architectural baseline
- Define system availability requirements
- Define any future application requirements
- Develop a reporting strategy
- Refine the conceptual architecture
- Define architecture subsystems
- Propose architecture subprojects
- Design application security architecture
- Design application functional architecture
- Design logical application and database architecture
- Design physical database architecture
- Design hardware and network architecture
- Develop system capacity plan
- Assess performance risks
- Design system management procedures

Depending on the size and complexity of the project these tasks can be major undertakings. The technical architecture steps should be tailored to meet the requirements of the project and should be considered at the time of

project planning. For instance, if the system will require high availability then this will be a critical factor in the design of the system (e.g. RAID, hardware redundancy, backup and recoverability, etc.). It is also important to consider the timing involved when planning for the production sizing study so that the acquisition of the production hardware, based on the results of the study, won't delay the project.

Oracle Applications Architecture

Architecture

In addition to hardware sizing there are Oracle Applications architectural issues that must also be considered as part of the overall sizing of an Oracle Applications system. The following lists the key architecture options available:

Oracle Server vs. Oracle Parallel Server

For very large installations where one database server is not big enough to handle the expected workload the Oracle Parallel Server option may be appropriate. There are performance considerations that should be evaluated before choosing to implement the Parallel Server option, but Oracle Applications is supported with the Oracle Parallel Server option.

Monolithic vs. Client/Server Architecture

The Oracle Applications database and the Oracle Applications product files/concurrent manager processes can run on the same machine or can be separated onto two or more nodes. Character mode users can also run in a single machine environment, or can be separated to one or more additional nodes.

Parallel Concurrent Processing

Parallel concurrent processing is a further expansion on the concept of the above client/server architecture. Parallel concurrent processing spreads concurrent managers across multiple nodes in a network system. This architecture distributes the processing load and provides fault tolerance in case one or more nodes fail.

Server Partitioned Mode

Server partitioned mode is a special configuration wherein different Oracle Server versions are used for linking the Oracle Applications executables and for running the database server. This is supported in both monolithic and client/server environments (see previous description). For example, this mode is how Oracle8 support was implemented for Release 10.7.

Client Access Options

Character Mode Forms

A SQL*Forms 2.3/2.4 character based interface. This has been the traditional interface for Oracle Applications, but is only supported through Release 10.7 and not on Windows NT at all. This is a fairly lightweight front end that does not require much network or computing resources.

SmartClient

SmartClient was Oracle Applications' first attempt at a GUI interface. The SmartClient architecture places user interface-oriented application processing on the client, and data-intensive application processing on the server. SmartClient was deployed as a Microsoft Windows application (e.g. 16-bit) and is supported through Release 10.7.

NCA

The application partitioning that is fundamental to the SmartClient architecture is extended to enable Oracle Applications with the Network Computing Architecture. In NCA, the database server partition remains essentially unchanged, and the user interface logic and the presentation layer are separated. User interface logic moves off the desktop and onto a middle tier of application servers, which can scale to accommodate increases in user load. Presentation management is handled by software downloaded to Java-enabled desktops. NCA is first supported in Release 10.7 and is the only option supported for Release 11.

Oracle Self-Service Web Applications

Oracle Self-Service Web Applications provide a lightweight (e.g. web browser) interface into selected Oracle Applications. Oracle Self-Service Web Applications use the Oracle Web Application Server and the PL/SQL cartridge to generate dynamic html pages, with data retrieved from database. It leverages JavaScript for some client-side processing to supplement the PL/SQL code running in the database. Some standard batch interface programs from Oracle Applications are used for processing transactions originated from these web applications. In addition, these applications access an active data dictionary, the Web Application Dictionary, containing information on the display elements and display formatting for the dynamically generated html pages.

Setup Options

Commerical vs. Government (Public Sector for Release 11)

Commercial or Government versions of the Oracle Applications products are available.

Multiple Organizations

Multiple Organization Architecture - Multiple Organization Architecture is a simpler way of implementing multiple sets of books. There is only one installation of each product, and the data contained in the product schemas is partitioned by organization, and hence by set of books.

Multiple Reporting Currencies

New for Release 11, Multiple Reporting Currencies allows the use of more than one functional currency. Implemented through separate sets of books, this feature allows transactional level accounting records to be reported and maintained in multiple functional currencies.

Globalizations

A Globalization is a functional enhancement to the base product, developed by an Oracle subsidiary or the International Applications Group, to meet a mandatory requirement for a country.

Localizations

Oracle Applications includes localizations, which are extensions consisting of additional forms, reports, or database objects. The localizations provide the functionality needed to meet the unique business needs of a region, whether one country or a group of countries. In Release 10.7, many localizations are incorporated into the core Oracle Applications products. You have the option to install the remaining localizations when you run AutoInstall.

Multilingual Support

Multilingual Support (MLS) is an Oracle Consulting Services solution for providing multi-lingual capabilities within Oracle Applications. It is available for Oracle Applications Release 10.6.1 character mode and above, and a standard solution is available for Release 11.

Release 11 Specifics

- Oracle8 is the only supported Oracle Server database version.

- NCA is the only option for client access and there is no support for character mode forms or SmartClient.
- Oracle Reports moves to the Oracle Reports Web Server, although the current character based reports remain.

Architecture Diagram

The following Oracle Applications architecture diagram attempts to show the technical architecture options available and the complexity involved.

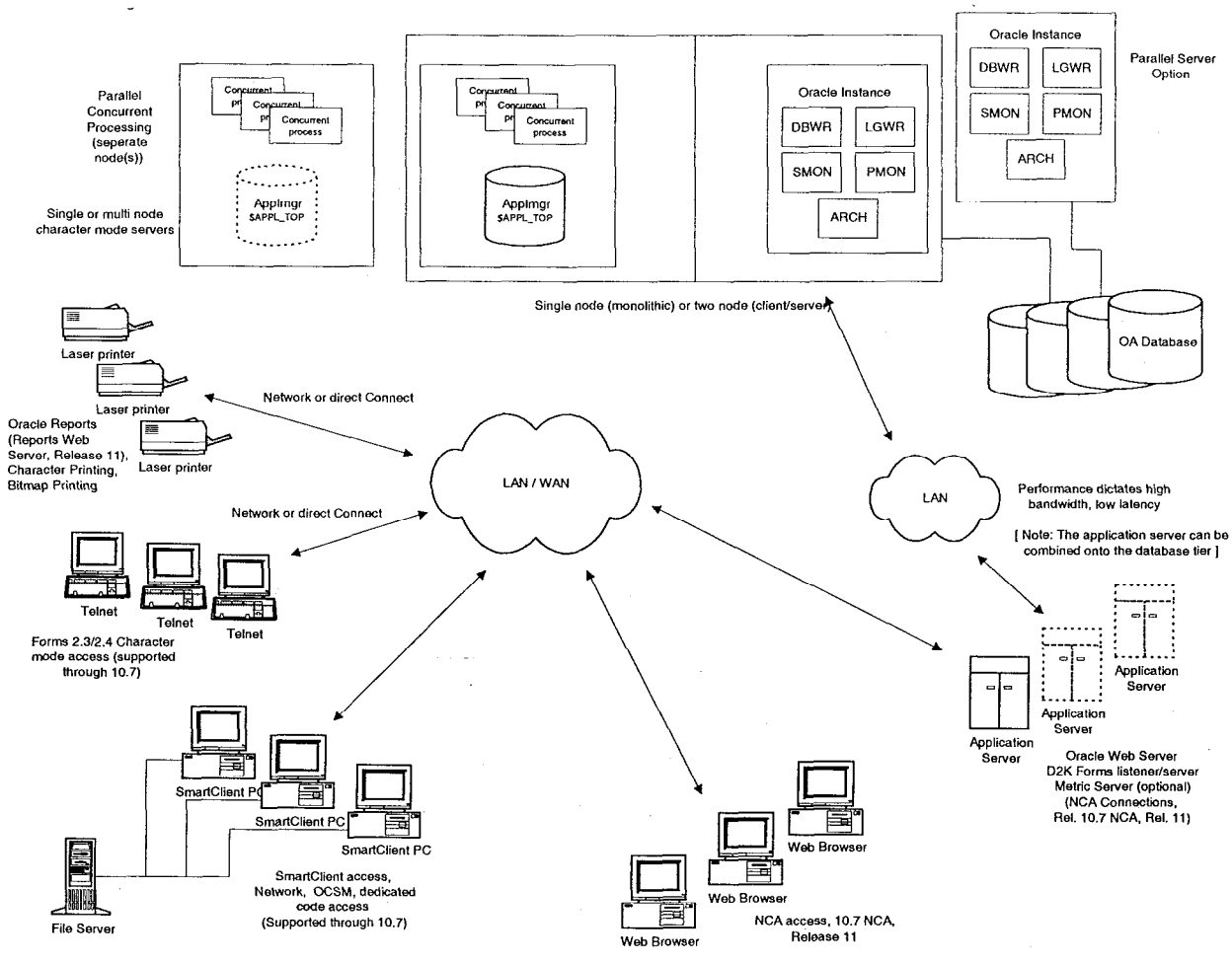


Figure 1: Oracle Applications Architecture

Hardware Sizing

Hardware sizing is really a two phased effort. Initially, the project environments must be created and supportable (e.g. managing growth, refreshes, patching, backups, and availability). Once this set-up is accomplished the technical staff must then focus on studying the production system requirements. This study must not only take into account the size of the production system, but also all of the support systems needed to keep the production system running. There are many factors that play into hardware sizing. However the sizing of the production system is a much more involved process than the sizing of the support and project only systems. The risk factor in a non-performing support system is much lower than in the final production system. In fact, the sizing of the support and project systems is essentially an exercise in memory and disk space calculations. These systems though, can provide additional insight into the sizing of the production system hardware.

Many companies have a vendor preference when deciding on the hardware for an Oracle Applications system (e.g. vendor, OS, machine types). In general, as long as the vendor is supported by Oracle, the actual brand is not a great

concern. The architecture however can come into play, especially when considering UNIX vs. NT. It is important to consider the limitations imposed by an architecture when trying to devise a hardware layout. NT for instance does not currently support multiple APPL_TOP directories, making it impossible to use one machine for two systems (e.g. test and development environments on the same machine). NT also doesn't support character mode forms which could be a consideration for Release 10.7 customers.

Also, it is generally unwise to try and manage multiple vendor hardware (e.g. the support systems on HP while the production system is targeted for Sun). Although Oracle may be supported on multiple platforms there are issues such as port specific bugs, system behavior, being able to physically copy databases between platforms for environment refreshes (Oracle doesn't support the physical copy of a database across different hardware platforms), licensing, and Oracle Support issues. This is mainly an issue for the database tier and it is not as critical that the application tier be the same platform as the database tier unless they will co-exist on the same physical hardware.

For the production system, most of the technical issues like how many CPUs, how much memory, how many disk controllers, and RAID vs. individual disks should be left to be determined by the production system sizing study and the project's requirements.

Initial Sizing (Project Environments)

Once a hardware vendor is selected then the Oracle supplied sizing spreadsheets and installation manuals can be used to determine disk and memory requirements for the initial support systems. There are rough CPU requirements available, usually as TPC specifications, that can be used for initial CPU sizing. Depending on the size and flexibility of the project, the experience of the project personnel, the budget available, and the hardware vendor's support, a suitable demonstration/development environment(s) can be obtained with a minimum of effort. Even at this initial level, an understanding of the project scope and client environment is important. A pre-project sizing analysis for the project environments, taking into consideration initial volumes, service level requirements, recoverability, and client training is the safest way to handle the initial hardware acquisition. Since this effort may have a very low degree of precision leasing of hardware and/or obtaining expandable systems should be strongly considered.

Final System Sizing (Production Environment)

The sizing of the production system should be done based on defined metrics in order to insure that the actual system will meet the service level requirements defined by the Technical Architecture tasks. Therefore, the initial conceptual architecture, defined business volumes, and workload characteristics should be used as input to a system sizing/capacity planning study. The detail of this study should be determined based on the precision of the required result and the risk that can be tolerated if the study is wrong. There should be well defined goals created with specific deliverables identified. The study should incorporate the project's goals and should address each component defined in the conceptual architecture. Again, this is a task that is complex and important enough to warrant the advise and guidance of an expert. Even if it is only for a one day session to validate the study and its methodology, an expert should brought in to review the proposed study and perhaps the results.

For an Oracle Applications production system the following architectural areas should be targeted:

- Database server
  - Size for peak load (CPU, memory, I/O call rates)
  - Do architectural decisions penalize performance? (e.g. RAID 5 for redundancy may impose an excessive write penalty consider RAID 0+1)
  - Workload - do long running concurrent manager processes on the database server saturate the system? (e.g. May need to consider moving the concurrent managers to one or more separate nodes)
  - Is there sufficient disk capacity to support the storage requirements
  - Can the backup system meet the backup and recoverability requirements
- Oracle Application Server - Concurrent processing
  - Peak load (CPU, memory, I/O call rates, network utilization - if separate from the database server)
  - If character mode users, their peak workload
- Printing, especially the increased network capacity required for bitmap printing

- NCA Application server
  - Web Server load, fail over
  - DNS service rate
  - Peak users per server capacity
  - Code management for multiple servers, network impact on dedicated vs. shared file systems
  - Metric server load balancing for multiple forms servers
- Client setup & management
  - Code distribution (SmartClient - dedicated, OCSM, file server, for shared files must consider network bandwidth)
  - Sizing (CPU and memory)
- Network sizing (bandwidth and latency)
  - Application server(s) to database server (NCA)
  - Clients to application server (NCA)
  - SmartClient - effect of latency on round trips, bandwidth of concurrent users

Project Environments

One of the key areas of sizing an Oracle Applications project is to understand the environments needed to implement a production system. A single Oracle Applications production system will generally require the installation and support of multiple Oracle Applications environments. These multiple environments are then used to support setups, development, test, conversions, training, system sizing, and pre-production/quality assurance functions.

An individual Oracle Applications environment is made up of three basic components. The first is the database server, which supports the database objects that comprise the data and server side logic portion of Oracle Applications. The second component is the Oracle Applications product files (formerly identified by Oracle as the Applications Server). These files support the installation and maintenance of the applications, the use of the character mode forms, the execution of the concurrent manager processes, and the Oracle Reports, SQL scripts, and other programs and utilities that are run by the concurrent manager processes. In Release 11 Oracle has combined both of these components into a "database tier" and refers to this tier as the database server. The third component consists of the user interface products of which there are currently three. There is the character mode access that is supported through a SQL*Forms 2.3/2.4 interface (Note that the character mode forms can reside with, or be separate from, the Oracle Applications product files, but are not supported on NT). There are two GUI access methods, the SmartClient product which runs in Microsoft Windows and the NCA product which runs through an internet browser connecting to an application server. In NCA, the application server now refers to the "application tier" which consists of the Web Application Server, Forms Server and optional Metrics Server. The desktop browser and Java client applet form the "client tier".

Multiple environments are defined as separate database and product file installations (e.g. database server) for each Oracle Applications system. These environments can be created on one or more machines except on NT as mentioned above. Separate Oracle Applications environments will usually be set up for development, testing, training, and production. In addition, a standalone database may be required for a Designer/2000 repository to support custom development. These environments may also be required beyond the initial project for ongoing patch testing and customization support. These environments will require coordination and management. Some of them will require periodic "refreshes" from backups or another Oracle Applications system, and all will require patch application management.

Depending on how patches are managed, one or more client "environments" will be required for access to the multiple Oracle Applications environments. Because SmartClient and NCA patches are applied to the database server (e.g. the Oracle Applications product files and Oracle database) and the client software (e.g. SmartClient code or NCA application server), these environments must be synchronized. For this reason, one or more client environments will have to exist simultaneously.

Generally a project will start with the creation of a "Demo" environment. Oracle has created several versions of demonstration environments called Demonstration Product Groups. These demo environments contain all installed

products set up with simulated company information to allow customers to see how the Oracle Applications work without requiring actual setup. Once the project is defined and the functional teams begin analysis, a "development" environment is required so that a chart of accounts can be defined and setup information can be started.

Since some setup steps can only be performed once, often times the creation of the development environment and the setup information is an iterative process. This is where, the "virtual" database, comes into play. The creation of an Oracle Applications environment is a time consuming process. A database must be created and prepped, the Oracle Applications software must be loaded and the installation process run, and server patches and client software installed. By taking full database and product file backups at designated times in the project, the installation, setups, and initial data loading can be preserved. This backup allows for an iterative process that is more efficient than a re-installation. This process can be carried forward throughout the project. The "virtual" database may move forward after each phase of the project to encompass initial installation, setup information, initial data loading, customizations, etc. until a point is reached where the "virtual" database will be frozen and used for the final production system setup. This will be further addressed in a later section.

In addition to the demonstration and development environments, an environment for testing customizations and data loading may also be needed. Other environments needed during the project might include a training and sizing study environment, or an environment just to map setup information. The following provides some typical environments and descriptions:

Demonstration Environment

Purpose: One of Oracle's demonstration product group environments. Applications training, initial analysis.

Features: Periodic backups not required.

Development Environment

Purpose: Customization, conversion, interface development

Features: Backups needed, unstable, patched when necessary

Test/Conversion Environment

Purpose: Customization, conversion, interface, user, and new patch testing

Features: Backups needed, initial patch testing, unstable

Mapping Environment

Purpose: Hold setup information, used for refreshes, virtual database snapshots.

Features: Multiple backup versions.

System Sizing Environment

Purpose: Used for the system sizing study.

Features: Backups not required.

Training Environment

Purpose: On-going internal system application training

Features: Refreshed periodically from a frozen backup, highly stable, patched only when bugs prevent training (requires a new frozen backup), regular backups not required.

Production Environment

Purpose: Production environment

Features: Highly stable, backup/archiving required, patched only after patch acceptance on test/development systems

Designer Repository Environment

Purpose:	Custom development Designer/2000 repository.
Features:	Highly stable, individual user database accounts, backups required

The following diagram illustrates the environments that may be required and the overlapping nature of their co-existence.

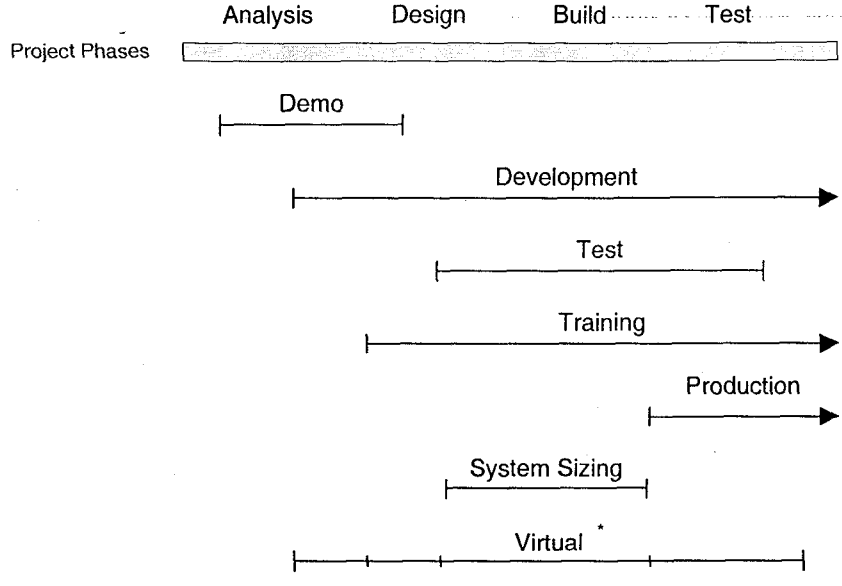


Figure 2: Project Environment Timeline

Patching Issues

Oracle Applications requires many patches to be installed in order to function properly. During the course of any project this patching will consume a considerable amount of time and, if not managed properly, can increase the risk of project delays due to lost time in testing environments that haven't been patched as expected.

There are two types of patches, bug patches and release updates. Both usually have a client component (e.g. SmartClient or NCA patch) and a server component (e.g. Oracle Applications product files and database). Patches are supplied with the initial Oracle Applications product shipment and are obtained from Oracle's World Wide Customer Support (WWS) organization through email, ftp, or mail (e.g. tape or CD) based on a TAR request or proactive product level patches. Server patches are installed using the Oracle Applications AutoPatch Utility (e.g. adpatch), and client patches are installed either with a special Oracle Applications SmartClient utility called "otto.exe", or for NCA with the adpatch utility. This section will outline the administrative issues of creating and maintaining an Oracle Applications patching methodology.

Patch Architecture

Bug level patches typically affect a single Oracle Applications product. Major product level patches generally affect an entire Oracle Applications product group. An example of a bug level patch might be one that fixes an error in an Oracle Payables form, whereas a major product level patch would include the SmartClient server updates patch, the NCA server update, or the Required Technology Update.

Patches are typically distributed in a compressed format which varies by platform. In a UNIX environment patches are usually distributed as compressed tar files and in the Windows environment they are usually distributed as self-extracting executables. Most NCA patch files are being distributed as zip files.

Patch Testing

Once a patch has been identified and obtained from Oracle WWS it will need to be installed and tested in a suitable "test" environment. After the patch has been installed and tested, it can then be installed into the production environment. Testing should identify that the patch fixes the problem at hand and that it does not introduce any obvious "side effects". Also, most Oracle patches are not "reversible". If a patch is applied that causes more harm than good, then either a restore from backup will be required, or a new patch to fix the "bad" patch will have to be obtained from WWS. For this reason, make sure that good backups exist before patches are applied.

Record Keeping

A record should be kept of the patch's status as it is applied to each environment. Ideally this record will identify the TAR associated with the patch, the dates and environments it has been applied to, and the testing status. Avoid relying on the Oracle Applications product file directory and the install logs, as this is a difficult and error prone method of record keeping.

Installation Coordination

The installation of patches must be coordinated with all users of the system. This coordination is needed to avoid unexpected outages for the project team and to allow the technical staff the time to apply the patches. In order to install an Oracle Applications bug level patch, the concurrent manager processes must be shut down. This allows for possible relinking of Oracle Applications product files executables. To install an Oracle Applications major product level patch all users must be logged off the system and the concurrent manager processes must be shut down. Since any Oracle Applications project will require the installation of many patches, patch time must be built into the project plan.

For projects that implement the SmartClient or NCA architectures some of these patches will contain both a client component (e.g. SmartClient or NCA patch) and a server component (e.g. Oracle Applications product files and database). These two components are dependent upon each other and their application must be coordinated in order to avoid client side errors. In addition, multiple client installations (e.g. \apps10 or \appsnca directories) may need to be maintained for each Oracle Applications system in order to allow testing and application of new patches that affect only one, or a few, applications. This can allow unaffected project teams to access the system without interruptions.

Refresh Procedures

Refresh procedures will vary depending on environment and requirements. For full copies from production or synchronization between support systems, a database copy can be taken. In order to support this the source and target machines, they may be the same or different, must have the same basic architecture (e.g. operating system, Oracle version, disk space, etc.). This is another reason for requiring that, at the very least, all database server hardware be from the same vendor and of similar architecture.

If partial copies of data are required, an extract procedure should be created to allow creation of an environment with a subset of the source system's data. Typically a backup copy of the system with setup data and no transactional data is best used as the initial target. This is another argument for periodic backups at different phases of system setup (e.g. the virtual database). Then export/import or data loads through standard API's can be used to populate test data.

For the training environment there may be a need for a "frozen" backup copy that can then be used to refresh the database after each training session. This task will place additional burdens on the backup system and the support staff to perform the actual refreshes and should be accounted for in the project plan.

Before any refreshes or restores occur, the source and target database environments should be in sync in regards to patch application. Alternatively, once the refresh is complete, the database portion of the patches that have not been applied to the source database should be reapplied.

Virtual Database

During the course of the project, there may be a need to allow for the migration of setups and data established during mapping to subsequent environments. This process is sometimes referred to as *instance synchronization*. Oracle provides no methods for transferring setup information, but there are third party products available to manage setup information across multiple environments. Whether these products are worth the cost or not should be evaluated based on the project's requirements. Most often the initial setup information is maintained through the virtual database and further setup data is re-keyed. This method requires that detailed documentation be kept.

To support instance synchronization, a virtual database (VIRTUAL) should be established to hold each progressive phase of setup information. This database will have multiple versions and will be a full backup copy of the mapping database at various stages of the implementation cycle. When configuration decisions are finalized, the project team will perform the required configuration in the mapping database and it will then be backed up to create a new VIRTUAL database version. At the end of the Solution Design stage, this environment will contain all the setup information required for the Business Systems Test, Performance Assurance Testing, End-User Training, and Production environments. The VIRTUAL database backups will be used to re-create the test, and possibly the development database, without requiring further installs of the Oracle Applications software, and will allow the project the flexibility to try different configurations without having to incur the downtime of multiple installs. Refinements to the VIRTUAL database may be made as a result of each of these activities.

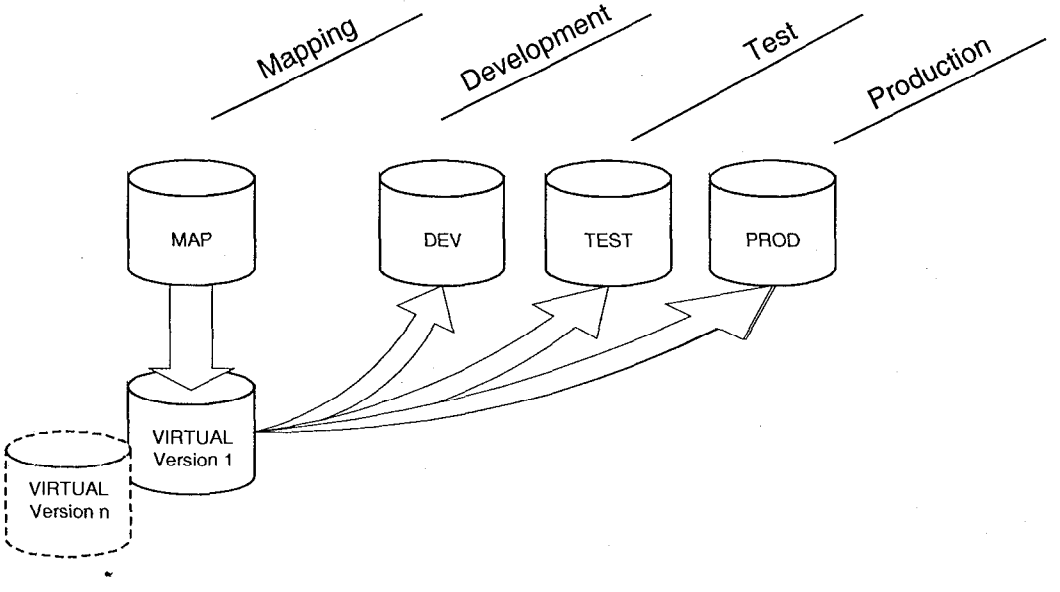


Figure 3: Virtual Database Synchronization

Long Term Capacity Planning

The best time to start capacity planning is at system deployment, and as part of an Oracle Applications implementation project, this is a great opportunity to begin gathering the data necessary for capacity planning projections. Whether a new implementation or an upgrade, the system will have changed enough to make a good starting point for tracking resource utilization. There are many sources of information available on capacity planning and most every DBA book has examples of how to implement some level of resource tracking. However, the key for long term capacity planning, and validation of the production sizing study, is the ability to characterize the system workload and identify trends over time. A common tendency is to only monitor resource usage in response to performance problems, which misses the opportunity to prevent performance issues by reacting proactively to problem areas.

In a paper that I presented with a colleague at the 1995 Fall OAUG conference titled "ORACLE7 I/O Monitoring and Tuning" we proposed a simple interval based method of monitoring database I/O and some key contention areas in the database to provide a long term perspective on database activity.

With this type of data, information is available to deal with both sudden spikes in usage (e.g. users calling and complaining about the system being slow) and longer term usage growth (e.g. total database I/O has grown 20% during the last 6 months). In the case of users complaining about the system being slow it may be possible to identify an area of contention (e.g. an Accounts Payable datafile has three times the I/O it "normally" has) because metrics are available that will show what the system "normally" does. This then provides focus for identifying the causes of this spike in usage. As for longer term trends, it may be that growth in I/O is because of added usage, which could be validated by an increase in the number of logged users. Or it may that a table that has grown to the point of needing one or more additional indexes.

Database monitoring can be broken down into three basic categories. The first category, and the most common, is point-in-time monitoring. Point-in-time monitoring is used to identify an immediate performance problem. An example of this might be the use of the Oracle Enterprise Manager Top Sessions product to identify a database session consuming large amounts of CPU time.

The second category of monitoring is of short duration and high impact, and is used to identify a known performance problem. The most common tool is probably the "utlbstat/utlestat" scripts supplied with the Oracle Server and are typically used to identify I/O bottlenecks and session wait problems. The Oracle Trace and Expert products, a combination of Oracle Server utilities and Oracle Enterprise Manager products, can also be used to provide very detailed monitoring information.

The third category of monitoring is typified by long duration and very low impact "snap shots" of the database in order to perform capacity planning and trend analysis. This type of monitoring usually captures a summary of database statistics and attempts to correlate them over some period of time, usually days, weeks, or months.

The common theme of all of these strategies is an attempt to identify and manage database performance to meet service level requirements. In addition to database monitoring, there are other areas of an Oracle Applications system that must also be considered. The database is only one component, hardware/OS monitoring for the database and application server(s), client performance, network bandwidth, and application logic must also be included in a comprehensive monitoring strategy.

Conclusions

Project planning must consider technical architecture requirements. This requires an initial understanding of the project requirements and the Oracle Applications technical architecture in order to allocate enough resources to the Technical Architecture tasks.

Understanding the project environments required, and the time and resources required to support them, will help the technical staff meet the requirements of the project.

Incorporating patching time into the project plan (e.g. padding extra time) will provide the technical staff with enough time so that other task timelines will not be impacted.

Production system hardware should be obtained based on specifications identified by a system sizing study that incorporates expected workload, business volume requirements, and technical architecture. This may also require the addition of a capacity planning expert for advice and validation.

Implementing a long term capacity planning strategy can help validate the production sizing study, provide the metrics, and help set expectations, about the performance of the system. Hopefully, the information obtained will allow the technical staff to proactively manage system performance and keep an active involvement in the sizing of the system to avoid future problems which could taint the perception of an otherwise successful implementation project.

About The Author

Andy Rivenes has been an Oracle DBA working with Oracle Applications since 1992. He has worked for Oracle Corporation as a Principal Consultant specializing in Oracle Applications technical architecture and currently works for Lawrence Livermore National Laboratory. Mr. Rivenes also maintains an Oracle Applications DBA web site at www.appsdba.com and can be reached at arivenes@llnl.gov.

A special thanks to Michael Ogden and Gary Wachowicz for their help in editing this paper.

Sources

Predicting Computing System Capacity And Throughput, Craig A. Shallahamer, Oracle Services
Oracle Applications, Release 10.7 for UNIX, Installation Manual, Part No. A47542-1
Oracle Applications Installation, Release 11 for UNIX, Part No. A57983-02