ORACLE®

PRIMAVERA

P6 Reporting Database Ver 2.0/2.1 Planning and Sizing

An Oracle White Paper
April 2011

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Contents

Introduction

Detailed planning is essential to successfully deploying the P6 Reporting Database and P6 Analytics. Both products have multiple components and dependencies. With a deep understanding of all the moving pieces, and a clear strategy to gather the critical requirements, it is possible to make these products an integral part of your project management infrastructure.

These products are fundamentally a generic data warehousing implementation. The implementation does not differ greatly from any custom data warehouse implementation in that it involves several databases, servers and a controlling ETL process. Oracle Primavera provides the code to perform the ETL process.

When planning each implementation, you need to consider:

- The requirements for physical storage and for CPU processing are very large.
- The implementation includes both an operational database and star schema.
- The optional P6 Analytics that contains a pre-built application including dashboards, reports and underlying metadata to more easily expose information to end users.
- Differences in data, amount of change, and business requirements.

This is by no means the end of the planning and sizing process.

This document assumes a firm understanding of the P6 Reporting Database and P6 Analytics architecture.

This paper provides the following information to start you on the road to true project management business intelligence:

1) Review the critical performance factors for the product.
2) Outline a methodology for planning an installation.
3) Look into the physical storage requirements of the data warehouse.
4) Address the server performance requirements of the ETL process.

Critical Performance Factors

Four Key Areas of the ETL

While the ETL process has many individual aspects, there are four general areas that will affect performance:

1) Pulling data between servers.
2) Managing the updates of the component databases.
3) Performing PL/SQL / SQL transformation operation on the database server.
4) Calculating key project management data using the P6 API.

Nearly half of the critical fields for reporting are not physically stored in the P6 EPPM Database schema. Because of this, the P6 Integration API will be executed during the ETL process, and will have a significant impact on performance and pose a key challenge to P6 Data warehouse solution that other applications do not need to address.

Pulling Data between Servers

As with any ETL process, there are elements of data movement revolving around the E (Extract) and L (Load) in ETL.   In a typical implementation architecture, the P6 EPPM Database, Staging, and data warehouse (ODS/Star) are deployed on separate physical servers.   As a result, the bandwidth must be maximized and latency minimized between servers.   Ideally, the servers reside in the same data center with gigabit ethernet connection between servers. (Oracle Primavera performance tests are performed with servers in a central data center with gigabit connections.)   Throughput of server communication should be verified. This can be done by performing basic file copy, or FTP operations, between the servers.

There are two types of data movement processes.   The first is standard SQL either with INSERT-SELECT or CREATE TABLE AS (CTAS) syntax using an Oracle database link. While the Oracle RDBMS is efficient at moving data through the link, the overall performance is dependent on the physical network connection.   The second type of data movement uses the SQL Loader (sqlldr) Direct Path to push flat file data directly into the Oracle database.   SQL Loader inserts data at a much higher rate than normal SQL INSERT by bypassing much of the overhead, and directly creating the data blocks.

Merging Updates into Target Database

P6 Reporting Database R2 significantly changes the method of updating tables incrementally in the target schema.   This is the process by which the changes are merged into the base tables.

P6 Reporting Database R2 leverages PL/SQL Bulk operations to process smaller batches while performing interim commits.   This results in linear scaling of update operations (for example, if it takes five minutes to update one million rows, it will take ten minutes to update two million rows). Rows per second performance remains constant, regardless of amount of change or data size.

Many of the processes, including the PL/SQL Bulk operation processes, are run in parallel threads.   The use of parallel PL/SQL processes not only increases the potential scalability of this process, but also the demand for CPU on the server.

Space requirements are also increased by the new data movement process.   Changed rows are stored in temporary tables (starting with TMP_) before being processed by the PL/SQL bulk inserts.   These tables remain after the process, and are not cleared until the next run of the incremental process. The space requirements for temporary tables are dependent on the amount of change being processed.   This will be addressed in more detail in the Physical Sizing section of this document.

PL/SQL-based Transformations

Some of the transformation process is done with PL/SQL.   The largest portion of this is referred to as *direct SQL*.   These are SQL update statements run directly against tables to perform simple transformation that does not require business logic.   These processes are generally parallel, and very CPU intensive, on the database server.   Some transformations on larger tables have been moved from the staging database to the Oracle ODS Database/Oracle Star Database schemas to reduce unnecessary data movement between servers.

P6 API Calculations: ETLCalc

The heart of P6 Reporting Database ETL is the execution of the P6 Integration API in a process referred to as *ETLCalc*.   ETLCalc is a specialized execution of the P6 Integration API against the staging database. The staging database is made to *look* like the P6 EPPM Database by having all the required schema elements, even if they are not needed for reporting.   While the other parts of the ETL process involve primarily database operations, ETLCalc has elements of both database and application processing.
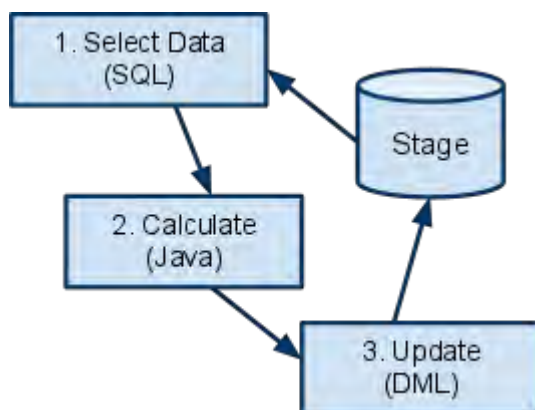
1) **Data Queries** - The API business objects select data with queries similar to the P6 Application.   This process requires some of the same indexes used by the application.
2) **Calculation** - Code in the Java business objects is executed to calculate fields (for example, Earned Value, Start and Finish dates).   Daily activity and resource assignment spreads are also calculated. Spread data is saved to flat files on disk.

3) **Update** - Update (DML) statements store calculated values back to the Oracle Stage database.

Planning Process

Why Planning is Key

As previously mentioned, the product is fundamentally a data warehouse. Without proper planning, a successful implementation will be difficult to achieve. Achieving a successful implementation requires a structured approach that will give you the necessary insights to make correct decisions about the physical and logical aspects of the implementation. This section outlines the planning methodology to guide you through the process step-by-step, including:

1) Requirements Phase
2) Planning Phase
3) Testing Phase
4) Initial ETL Phase
5) Operational Phase

Requirements Phase

The first phase in any data warehouse implementation is to understand what the users of the system want to get from the solution.   This includes:

▶ The types of reports
▶ Level of detail
▶ Time frame
▶ Data freshness

This information must be gathered before making any hardware or architecture decisions.   Time spent at this phase will greatly reduce the risk during the rest of the implementation and the subsequent operation of the data warehouse.

A wide variety of reporting and analytics results can be achieved with the product. However, not all of these may be required in a given installation.   There are two broad categories of reporting solutions:

▶ Operational
▶ Analytics/Business Intelligence.

Operational reporting covers the day-to-day, actionable reports used by project managers, resource managers and other tactical personnel in the organization.   This type of reporting is typically the traditional, tabular reporting that is repeated on a daily basis.   A key consideration of operational reporting is the scheduling and delivery of the reports.   The combination of the Operational Data Store (ODS) and Oracle BI Publisher addresses the scheduling, execution and the delivery of the reports.

Key Questions to Ask about Operational Reporting

▸ **When will reports be run?**   Perhaps the individual users will need to execute on demand.   Often reports are needed prior to the start of work on a given day.   These considerations will affect the timeliness of the data.   The current ETL process is designed to be at most a daily process. If there are specific reporting needs that require updates during a given day they may need to be addressed by other means.

▸ **How will reports be delivered?** Getting the right reports to users at the right time is key.   BI Publisher offers multiple ways to deliver reports from the Oracle ODS Database.   These include email, HTTP, WEBDAV, direct printing, et al.   The logistics of setting up these delivery methods must be considered during the planning process.

▸ **What will the reporting load be on Oracle ODS Database?** One of the major considerations affecting subsequent decisions will be the load on the reporting server.   This includes:

▸ **Oracle ODS Database:**   Queries will be executed against the Oracle ODS database to fullfil reporting requests.   This usage will likely peak during specific times of the day. This peak load must be considered as the requirement.   Since the exact types of queries are unknown at this point, it is important to gain a broad understanding of what the load will be:

  ▸ **How many users are accessing the database at the same time?**   This will determine the maximum load on the database server.

  ▸ **Is the reporting on individual projects or across the entire database?**   Aggregate queries will put more load on the server than project-specific queries.

  ▸ **Is the reporting done in batch or interactively?**   More interactive reporting will increase the demands on both the server CPU and I/O subsystem.

▸ Many of the same considerations previously mentioned for the database should be applied to the Oracle BI Publisher reporting server.

Operational reporting has the distinct advantage of being very well defined and constant.   On a day-to-day basis the reporting load will be fairly consistent.   This is not the case for P6 Analytics.   P6 Analytics reporting is, by nature, dynamic.   The Oracle star Database schema and OBI integration was designed to allow a very rich environment. This means that the daily load on the data warehouse server and OBI will vary greatly.

The BI Server component is capable of robust caching of query results, which can greatly mitigate performance concerns. The effectiveness of caching depends on how much users share security. If every user's project access is distinct (including the level of access to cost fields) then the cache will only be effective for each user individually.

Key Questions to Ask about P6 Analytics/Business Intelligence

▸ **Who will access P6 Analytics?** A diverse set of users from the CEO to resource/project managers may require access to P6 Analytics. Each may have different requirements and use cases to consider.

▸ **What are the default ways of filtering?** By default, user requests for analytic information will include all the data accessible by that user. That may be more time consuming, and may include more information than necessary. Consider ways of filtering data, such as Project Codes and Portfolios.

▸ **What codes are used for reporting?** While the Oracle ODS Database includes all the data from the P6 EPPM database, the Oracle Star Database schema includes only a subset codes among activity, resource and project codes. Before moving forward, you must determine which codes are critical for analysis.

Planning Phase

Once the requirements of the resulting data warehouse are well understood, the planning for the installation can really begin. As with any data warehouse, physical storage demands are high. Because the calculation process places a unique demand on the ETL, this data warehouse implementation may require higher CPU/memory requirements. More detailed information will be provided in subsequent sections of this document. First we will examine the high-level aspects of planning the implementation.

Two ETL Processes

There are two related ETL processes which have unique aspects. The FULL ETL process is usually the longest, and it will have the most impact on the system. Most impact will be on ETL Server because it will involve business rule calculations on every activity and resource across the entire P6 EPPM database. Since all the data must be moved, rows are moved in basic operations between the individual databases. The full process will need to allocate space for all the rows plus temporary space to calculate and store the spread information. Depending on the size and complexity of your P6 EPPM database, running a Full ETL can take a long time to complete.

The Incremental ETL will be the major operational concern of the data warehouse. The timeliness and availability of the entire system depends on the regular and consistent completion of this process. The Incremental ETL process is dependent on throughput, which is dependent on the amount of changes to the P6 EPPM database between runs.

Monitoring Usage

Understanding the amount of changes to key tables is critical to the performance of the usage monitoring process.   This can be easily gathered on a daily basis from the audit columns on each table (for example, UPDATE_DATE) and the REFRDEL table.   The amount of change should be monitored for several weeks, and periodic peak activity should be noted.   The peak usage times are important to keep track of, since they will be used as the basis of hardware decisions. The following tables should be monitored:

- ▸ PROJECT
- ▸ TASK
- ▸ TASKACTV
- ▸ TASKRSRC
- ▸ TASKMEMO
- ▸ UDFVALUE
- ▸ RSRCHOUR
- ▸ PROJWBS

Basic Monitoring

Changes to the table (insert and update) are counted with the following query.   (This assumes running the query at the end of the day to get all the changes from that day.)   This query is repeated for all the critical tables.

```
SELECT count(*)
FROM <table>
WHERE update_date > trunc(sysdate)
```

Delete rows are queried from REFRDEL in a single step

```
SELECT TABLE_NAME, count(*)
FROM REFRDEL
WHERE delete_date > (sysdate)

GROUP BY TABLE_NAME
ORDER BY TABLE_NAME
```

Entire projects are recalculated, not just single activities. The Activity and Resource Assignment DAO (Data Access Object) is calculated at a project level, even if only a single activity is updated. The following queries will give a rough estimate of the effect of the cascading nature of the changes on Activity and Resource Assignment calculations.

```
select count(*)
from task t
where proj_id in (select proj_id from task
                  where update_date > trunc(sysdate))
```

```
select count(*)
from taskrsrc t
where proj_id in (select proj_id from task          where
update_date > trunc(sysdate))
```

A more precise picture of usage can be gained using P6 Auditing. While this can be used, it is not necessary at this point since only a general understanding of amount of change is required now.   Now is the time to look for large scale patterns in the updating of P6 EPPM Database that may affect incremental ETL performance.

Physical Sizing

Overview of Physical Space Usage

The physical space requirements of the data warehouse consist of more than just copies of the project management data.   Space requirements will vary with the amount of data processed from the P6 EPPM Database.   In this release, there is a more pervasive use of temporary data in order to increase the overall performance of the ETL process.   This results in quite a few temporary tables (prefix TMP_).   In total, the system uses space for the following types of data:
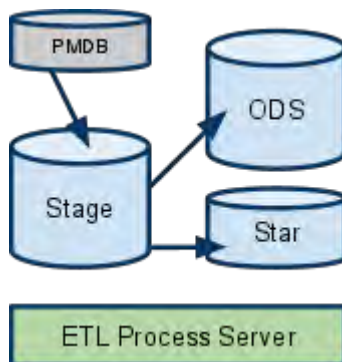
▸ **Core Project Management Data** - This includes all the physical fields that exist in the P6 EPPM Database.   This is approximately all the data in the P6 EPPM Database.   This data exists in two places in the data warehouse: Oracle Stage Database and Oracle ODS Database.

▸ **Logical Fields** - The fields that are not physically stored as part of the P6 EPPM Database are calculated and stored in the data warehouse.   While this is less than the size of the P6 EPPM Database, it may be as much as 50% of the total.

▸ **Temporary Flat Files** - A large portion of the calculated data is generated to flat files and loaded into target database with SQL*Loader Direct Load method.   This is the only part of the process that uses a large amount of file system space that is not already allocated to the database.   The amount of data will vary greatly, depending on the number of activities and resource assignments. The most space will be used during the initial ETL process, when all values are calculated at once.

- **Temporary Incremental Files** - During the incremental ETL process source data is initially loaded into temporary tables in the target database. This is done so that the network-dependent process of moving data is separated from the CPU and I/O intensive processes of updating the target database. Since this is completely incremental, the amount of space usage depends on the number of rows changed in the source P6 EPPM Database.
- **Fact/Spread Data** - Spread and Fact data total size depend on the number of activities and resource assignments, the average length of activities and the total window (date range) of the data warehouse.   Because of this, it will be treated as a distinct group. It is part of the ODS and is fundamental to the dimensional schema (Star).
- **Other ETL Tables** - There is some database space usage specific to the ETL process.   This space is trivial relative to the core P6 EPPM data.
- **ETL Process Installation** - This includes shell scripts, sql files and JAR files.

Physical Components

There are four physical components to consider in sizing the data warehouse.   Three of the components are schemas in the Oracle RDBMS.   When discussing physical components further they will be treated as separate instances of the Oracle database, or a physical server, although this is not necessarily required (see Processor Sizing for more details).   There is no direct, size impact on the P6 EPPM Database above normal space usage and we will not consider it as one of the physical components to be sized. Some additional space may be required in P6 to store REFRDEL information for the incremental process. The components are:

- ETL Process Server
- Oracle Staging Database
- Oracle Operational Data Store Database
- Oracle Star Database

ETL Process Server

While this server is the central controller of the ETL process, it represents only a small portion of physical space used.   This may or may not be the same server as the staging database.   The only files, other than the ETL process files, are log files from each run and the Temporary Flat files.   The flat files could be several gigabytes to many tens of gigabytes in size.   (See the *Calculating Spread Sizes* section for details.)   Flat file sizes will be roughly equivalent to the size of the Activity Spread and Resource Assignment Spread tables in Oracle ODS Database.

Oracle Staging Database

The staging database is the heart of the ETL processes.   This schema is a mirror of the P6 EPPM Database to a degree that simulates the P6 EPPM Database so P6 Integration API can leverage it.   The tables also have additional columns to hold values calculated by the P6 Integration API.   For any given application table, the initial set of fields are populated with data from the P6 EPPM Database.   The remaining fields are added to the table and updated by the ETLCalc process.

A rough estimate for sizing the staging database is that it will be twice the size of the P6 EPPM Database.   Since the P6 EPPM database includes other table data that is not extracted into the Oracle Stage database, this may not be exactly accurate. However, it will serve as an adequate initial estimate.

Index usage is also nearly identical to the P6 EPPM Database since the P6 Integration API executes similar business rule queries.   Index building is deferred until after the initial table load.

Incremental Considerations

During the incremental process, changed rows are moved into temporary versions of the staging tables. For each P6 EPPM table extracted, there is a temporary mirror table (for example, the TASK table has TMP_TASK).   Space usage for these tables can be calculated as a percentage of data changed.   If it is expected that on the peak day 10% of the data will be changed in the database during the daily ETL process, then this can be used to estimate the size of the temporary table. While this document uses the term **temporary table,** these are not Oracle Temporary Table types.

These are created on the fly by the ETL process stored procedures, and the data is cleared before each process. The data does reside in the tables between incremental process runs and can be used to diagnose process issues. If the P6 EPPM Database TASK table is 500 MB, then temporary may be as much as 50 MB. Do not include indexing because the temporary table is an Index Organized table. It does not require the same level of indexing as the base table.

The other temporary table created is used to capture deleted rows from the REFRDEL table in the P6 EPPM Database. This table is very small, and contains only the key fields necessary to perform the delete (primary key of the table and the SKEY).

For a complete list of tables extracted from P6 EPPM look at the contents of the directory <ETL HOME>/scripts/stage_load_incr). Each file name represents an individual P6 EPPM table extraction.

Other Oracle Stage Database Tables

There are several tables with the prefix ETL_ that are used during the ETL process. Sizing may vary over time, and with the amount of processing. This should not represent any significant part of the Oracle Stage Database size.

Estimating the Size of the Oracle Stage Database

When estimating the size of the Oracle Stage Database, first determine the general size of the P6 EPPM Database. This will overestimate the size because it will include some tables and indexes that are not extracted from the P6 EPPM Database. However, the majority of Project Management data is extracted.

```
select round(sum(bytes)/1024/1024,1) "MB"
from user_segments
MB
----------
   33686.6
```
(Execute this query as P6 EPPM Administrator user)

For planning purposes Oracle recommends allocating at least two times the size of the P6 EPPM Database for the stage database.

Calculating Spread Sizes

The number of daily spread rows is difficult to calculate with any certainty. The actual number of buckets depends on factors such as calendar work days, large differences in dates relative to the data date, ETL time duration and the general exclusion of zero value rows.   A quick calculation uses a "best guess" on the average number of days for activities and resource assignments.

```
    Total Activities: 1,000,000 X Average Activity Length: 5 = Total
Spreads: 5,000,000
```

Queries for Spread Sizing

Activity Spread Estimate Based on Median Activity Length

```
select
median(
greatest(nvl(target_end_date,to_date('12122000','mmddyyyy'))
 ,nvl(act_end_date,to_date('12122000','mmddyyyy'))
             ,nvl(reend_date,to_date('12122000','mmddyyyy'))

,nvl(rem_late_end_date,to_date('12122000','mmddyyyy')) )
-
least(nvl(target_start_date,to_date('12122199','mmddyyyy'))
             ,nvl(act_start_date,to_date('12122199','mmddyyyy'))
             ,nvl(restart_date,to_date('12122199','mmddyyyy'))

,nvl(rem_late_start_date,to_date('12122199','mmddyyyy')) )
) * count(*) Spread_Rows
from task t inner join project p on p.proj_id = t.proj_id and
orig_proj_id is null
where task_type in ('TT_Task','TT_Rsrc')
```

Resource Assignment Spread Estimate Based on Median Activity Length

```
select
median(
greatest(nvl(target_end_date,to_date('12122000','mmddyyyy'))
```

```
            ,nvl(act_end_date,to_date('12122000','mmddyyyy'))
            ,nvl(reend_date,to_date('12122000','mmddyyyy'))
            ,nvl(rem_late_end_date,to_date('12122000','mmddyyyy
')) )
-
least(nvl(target_start_date,to_date('12122199','mmddyyyy'))
            ,nvl(act_start_date,to_date('12122199','mmddyyyy'))
            ,nvl(restart_date,to_date('12122199','mmddyyyy'))
            ,nvl(rem_late_start_date,to_date('12122199','mmddyy
yy')) )
) * count(*) Spread_Rows
from taskrsrc tr inner join project p on p.proj_id = tr.proj_id and
orig_proj_id is null
inner join task t on t.task_id = tr.task_id
where task_type in ('TT_Task','TT_Rsrc')
```

Oracle ODS Database

The Oracle ODS database is the target database for operational level reporting. It contains a standard relational schema that mirrors the physical P6 EPPM tables, but it uses the object and field names from the P6 Integration API. These tables contain both the original, physical columns from the P6 EPPM Database and the calculated fields.   The space usage for Oracle ODS Database can be derived from a combination of the size of the Stage database, and the size of the spread data.   The Oracle ODS Database has the following types of table data:

▸ **Derived Application Tables** - These are tables that have a one-to-one mapping with a table in Stage (and, ultimately, with a table in P6 EPPM).
▸ **Spread Tables** - This is a combination of the detailed, daily spread data and aggregate tables.
▸ **Hierarchy Tables** - These tables map the underlying hierarchical relationships (such as EPS, WBS, etc.).

Indexing in the Oracle ODS Database defaults to the same indexing as the P6 EPPM Database.   This should be augmented and adjusted based on site specific reporting needs.

Incremental Considerations

As with the stage database, movement of data into the Oracle ODS Database is done by using temporary tables.   The application level tables will have corresponding TMP_ tables.   The Oracle ODS Database versions will be larger than Oracle Stage Database because they will contain both physical and calculated columns.   Otherwise, the process is identical to that used for the Stage database.

Estimating the Size of the Oracle ODS Database

The estimated size of the Oracle ODS Database is derived from two primary components: the estimated size of Stage and estimated size of spread data.   The majority of Oracle ODS Database data is simply a copy of the Oracle Stage Database tables.   Thus, the previously calculated size of the Oracle Stage Database can be used.   The remaining space usage comes mostly from the ActivitySpread and ResourceAssignmentSpread tables.   Any remaining data will be estimated as a percentage of the spread data (including aggregate spread tables and hierarchies).

Data Component	Calculation	Rows	Size Example
Stage Data	Total of Stage	n/a	40 GB
ActivitySpread	300 bytes/row	5,000,000 x 2*	3.0 GB
ResourceSpread	175 bytes/row	5,000,000 x 2*	1.6 GB
Other	30% of Spreads	n/a	0.4 GB
Total			45 GB

* Temporary data stored during loading process results in copies of spread data during initial ETL

Oracle Star Database

The Oracle Star Database contains a dimensional data model that includes four fact tables and the supporting dimensions.   For the most part, the dimension tables map directly to common staging tables (that is, W_PROJECT_D will map to the PROJECT table).   However, the Oracle Star Database will contain fewer rows because baseline projects are not directly accessible.   In general, the Oracle Star database is still much smaller than Oracle ODS Database or Oracle Stage Database.

The fact data represents the largest portion of data in the Oracle Star Database. As with any Star schema, this data is the most granular data and, by default, there are no aggregate tables built to support rollup queries. The primary two fact tables contain activity and resource assignment spread data respectively. The size of these tables will be the same as the corresponding Oracle ODS Database tables (ActivitySpread and ResourceAssignmentSpread).

The next largest fact table contains resource utilization data (W_RESOURCE_LIMIT_F). This differs from other fact tables in that the data size is not a function of the number or size of projects. Instead, it is function of the number of resources in the database, and the size of the data warehouse reporting window. There is a daily value for everyday of the reporting period, and for each resource. For example, if the reporting window spans five (5) years (1,825 days), and there are 1,000 resources in the database, the total records in the fact table will be 1,825,000.

The final fact table is the smallest, and it has only project-level data. The difference is that this table is a trending table, with snapshots of the data over time. The amount of snapshots depends on the interval chosen during installation (weekly, monthly, financial period). The granularity of this fact table is only down to the project-level; it contains no spread information. Calculate the number of rows using the total non-baseline projects times the number of snapshots. This will grow over time, so the yearly total for a 10,000 project database with weekly snapshots will be 520,000 rows.

Estimating the Size of Oracle Star Database

Only the fact tables will be considered for Oracle Star Database sizing purposes because they are responsible for most of the data. Of the four fact tables in Star, two of the tables (W_ACTIVITY_SPREAD_F and W_RESOURCE_ASSIGNMENT_SPREAD_F) are identical to the equivalent spread tables in ODS. See Estimate the Size of ODS for details. The rows for the remaining fact tables were calculated in the previous sections. Spread and resource limit data is initially loaded into holding tables (_FS suffix), so sizes are doubled for these tables.

Data Component	Calculation	Rows	Size Example
W_ACTIVITYSPREAD_F	300 bytes/row	5,000,000 x 2	3.0 GB
W_RESOURCE_ASSIGNMENT_SPREAD_F	175 bytes/row	5,000,000 x 2	1.6 GB
W_RESOURCE_LIMIT_F	70 bytes/row	1,825,000 x 2	0.125 GB
Dimensional and Temporary	20% of Spread	n/a	0.9 GB

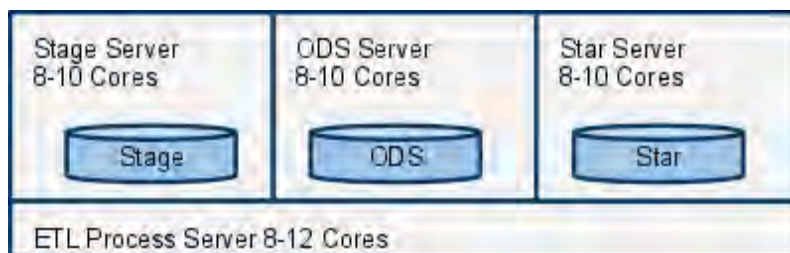| W_Project_History_F	265 bytes/row	6000	0.04 GB
Total			5.6 GB

Physical Hardware

When evaluating the physical hardware requirements, there are two distinct areas to consider.

▸ The performance of the ETL process, both full and incremental.
▸ The performance and concurrency of the online reporting solution.

While the ETL process is fixed regarding concurrency, the reporting needs will vary greatly. The demands on P6 Reporting Database and P6 Analytics may change from day-to-day. For performance sizing of OBI, please refer to the technical documents for the specific component (BI Publisher, BI Server, and Answers/Dashboards). This document will focus on the performance of the ETL process and queries generated against the warehouse databases (ODS and Star).

Consideration for ETL Scalability and Performance

The ETL process for P6 Reporting Database R2 was designed with multi-core processor systems in mind.   Instead of a serialized process Java is used to create a multi-threaded process to run concurrent threads of SQL*Plus, SQL*Loader and the P6 Integration API.   This can result in multiple runnable threads on the various servers.   This also means that process can be adversely affected by having to compete with other applications sharing the same resources.   Therefore, an ideal configuration would have dedicated cores available for each components in the process.

This is an ideal configuration that is meant to minimize contention.   By dedicating resources to each of the physical components, concurrent performance will be maximized during peak usage.   Different steps in the ETL process put a variety of loads on each component.   For example, the ETLCalc process puts the maximum thread load across both the the the staging database (SQL select and update) and the ETL server (P6 Integration API).   In this release, there is no concurrent processing occurring simultaneously on both ODS and Star servers. Therefore, from the ETL perspective, they could share the same physical hardware. Concurrent reporting usage should be considered in determining the correct CPU requirements for ODS and Star.

Network

While there is a distinct advantage to separating the components, there is an underlying assumption that the network connections between servers have very high bandwidth and very low latency.   These servers should be on the same network, with gigabit connections.   Any increase in latency will have a significant effect on the ETL performance.

Memory

With a large number of parallel processes running on large sets of data, the demands on memory will be very high.   The components of the data warehouse system should only be run on 64-bit operating systems to allow for large memory allocations.   Constraining memory quickly reduces performance.

The database servers need both block buffer and individual process memory.   These servers should always be setup using Dedicated Server (not Shared Server).   For an Oracle 11g database, the recommend minimum MEMORY_TARGET is 2 GB to 10GB depending of the P6 EPPM Database (for 10g, set SGA_TARGET to the same minimum value).

The Java process on the ETL Process Server is running multiple threads in the same process. Only run with a 64-bit version of the JRE to allow for larger memory allocation.   The maximum memory allocation for the Java process is configurable during setup (Max Heap Size).   The default is 1 GB. This may be inadequate for many datasets, however, and may cause failures in the ETLCalc process. Start with a minimum of 4 GB of memory for the Java process.

Incremental ETL Considerations

The performance of the incremental ETL process is of paramount importance.   The expectation is that the incremental process completes in a narrow window of time (typically nightly).   The process must duplicate all the individual user updates from that day, but in a much shorter amount of time.   While the resources used for incremental ETL are not in constant use, they are intensely exercised during the brief ETL process.

Extract and Load

Incremental ETL differs most from the full ETL in the way data is extracted and loaded. These processes are completely PL/SQL based, and use only resources on the database server. While extract and load may not put extreme pressure on the database server during the full ETL, the CPU load during the incremental ETL process will be much higher.   At any given time, there may be as many as ten database threads running PL/SQL. These are bulk PL/SQL inserts, updates and deletes making this very runnable.   This is the reason for recommending 10 or more cores on the database server, so each thread has an available CPU in which to run.   The performance of each individual core will ultimately determine the speed (rows/second) of the extract and load processes.

Summary of Physical Hardware Sizing

When planning for the physical hardware for the P6 Reporting Database, consider the following basic guidelines.

Size of P6 EPPM Database

Overall, the size of the database is going to play a large role in the performance.   There is a direct relationship between the database size and the performance of the full ETL process, since all records must be processed. There is some relationship between database size and the incremental process, since more project data will likely translate into more usage and more amount of change.

Amount of Change

Day-to-day, it is the performance of the incremental process that is the primary concern.   While database size plays a role, it is ultimately volume of change that is going to affect performance.   Even a small user community can generate tremendous amount of change with functions like copy/paste, delete project and create baseline as well as simple changes to global resources like calendars.   Careful monitoring of amount of change prior to installation will enable you to better plan.

Complexity of Project Data

Consider the actual data in the P6 EPPM Database.   A very complex EPS/WBS structure will greatly increase the processing time because these hierarchical elements require global processing even when the changes are small. The average size of activities can increase the memory requirements of the calculation process.   Larger projects will cause more volume of change because the entire project is recalculated based on any changes within the project. This does not include changes to non-scheduling data, such as activity codes and UDFs.

Expectations and Operations

There may be an expectation that for operational purposes the incremental ETL needs to complete in a smaller time frame.   These considerations may increase the hardware requirements.

Following is the hardware recommendations based on P6 EPPM Database size. For Database Deployment size refer to Timing Estimation Section

Small Database Deployment

ETL Server, Oracle STAGE Database, Oracle ODS Database (and Oracle STAR Database) on same box

CPU	Quad Core 3.46 GHz or equivalent
RAM	4 GB without STAR; 6 GB with STAR

Note:
1. This assumes light Business Analytics (Oracle STAR Database) usage. Otherwise a separate server should be allocated for Oracle STAR Database
2. 2 GB RAM should be added if OBIEE installed on the same box

Medium Database Deployment

Oracle ODS Database

CPU	Quad Core 3.46 GHz or equivalent
RAM	4 GB

ETL Server, Oracle STAGE Database, Oracle STAR Database

CPU	Quad Core 3.46 GHz or equivalent
RAM	8 GB

Large Database Deployment

ETL Server, Oracle STAGE Database

CPU	8 Core 3.46 GHz or equivalent
RAM	16 GB

Note: For shorter ETL window, Oracle ETL Server and Oracle STAGE Database should be in two separate servers.

Oracle ODS Database

CPU	8 Core 3.46 GHz or equivalent
RAM	16 GB

Oracle STAR Database

CPU	8 Core 3.46 GHz or equivalent
RAM	16 GB

Recommendations Summary

Following is the overall summary of recommendations for ETL Server and Database Servers

Stage\ETL server

8-12 Cores for ETL Process Server

8 -10 Cores for each Database Server
Gigabit Ethernet connection between servers and in same datacenter
64 bit OS
4 GB Minimum of memory for the Java process.

Database Servers

	Oracle Stage Database	**Oracle ODS Database**	**Oracle Star Database**
TableSpace Autoextent	Yes	Yes	Yes
Recommended Temp Tablespace	Minimum 2 files with file size set at OS max	Minimum 2 files with file size set at OS max	Minimum 2 files with file size set at OS max
Recommended UNDO Tablespace	Minimum 2 files with file size set at OS max	Minimum 2 files with file size set at OS max	Minimum 2 files with file size set at OS max
Minimum Space	2X P6 EPPM Database	2.5 X P6 EPPM Database	Same as P6 EPPM Database
SGA	8GB	8GB	8GB

Planning Revisited

Testing Phase

At this point, there should be enough information to make an initial decision on hardware for the implementation.   The next phase is to validate the hardware using a copy of the real P6 EPPM Database and the proposed hardware.

The full ETL process testing will quickly give visibility to the performance and memory usage of the ETLCalc process.   However, it is not a true indication of extract and load performance since the full ETL is optimized for loading all rows.   Still, a good test run of the full ETLCalc will give confidence in the performance. A key indication of performance is to look at the number of rows processed by the ACTIVITY, ACTIVITYNOTE, ACTIVITYSPREAD and RESOURCEASSIGNMENT DAOs.   The following row is from etlprocess.log file for a full run of the ETL.

```
Rows updated = 4819534 in DAO for ACTIVITY in processor Full, API, Processing
time (ms) 6238008, (seconds): 6238.008
```

Check the rate at which rows were processed for this DAO.   In this case, it is 772 rows/second.   Compare this rate to rates in the following table, and verify that the volume of change is above the threshold rate.   Anything below that rate may indicate an issue.

Data Access Object (DAO)	Threshold Rate (rows/second)
Activity	500
ActivityNote	100
ActivitySpread	2,000
ResourceAssignment	1,000

Performance of this aspect of the ETL is a good indication of the overall performance of the system because it is a combination of P6 API (Java), database reads, database writes, and file system I/O.   The ETLCalc process will also represent the majority of the time spent during the full ETL process.   Consult the *Summary* section of *etlprocess.html*; no line item should have a longer elapsed time than the ETLCalc.

Testing Incremental

Testing volume of change from the production P6 EPPM Database is a challenge.   One method is to take a copy of the production database, run scripts to apply changes to the P6 EPPM database, and then run the incremental process.   However, since the ETL process is loosely coupled to Project Management (spread data is saved to flat files on disk), you can do the following:

1) Take a copy of the production database at the end of the day.   Note the date and time of this backup. You will need it later.
2) Take another copy of the production database following a representative day of user activity. Try to choose a day that is on the high end of usage.
3) Restore the first copy of the P6 EPPM Database to the testing environment.
4) Run the full ETL process.
5) Connect to the Oracle Stage database, and update the ETL_INCR_HISTORY table with the date and time of when the first copy of the production database was made: UPDATE ETL_INCR_HISTORY SET LAST_RUN_DATE = '<initial copy date from step one>'
6) Replace the copy of P6 EPPM Database with the second version from step 2.
7) Run the incremental process.

By updating the ETL_INCR_HISTORY table, you can force the copy of the data warehouse to pull any changes made to the P6 EPPM Database after that date.

Timing Estimation

The following timing numbers are only an approximation. The actual numbers for your deployment depend on the data structure and the hardware used.

		Deployment Categories		
		Small	**Medium**	**Large**
Number of Objects	Projects	200	1,000	50,000
	Activities	100,000	1,000,000	5,000,000
	Activities per project	5,000	10,000	20,000
	Resources	500	1,000	4,000
	Resource Assignments	100,000	1,000,000	5,000,000
	Resource Assignments per project	5,000	10,000	20,000
	Full ETL Approximate completion time	~ 2-5 hours	~ 10 hours	~ 20 hours

Conclusion

Following a systematic approach to evaluating, planning and testing the architecture for your P6 data warehouse is the only way to assure a successful implementation.   With careful examination of the requirements, data sizing and user activity the appropriate hardware choices can be made early in the process.

ORACLE®

Oracle Corporation provides the software
that powers the internet.