PTSOSSessbase Workshop
Leon T. Liem
Senior Product Manager
PTS Server Technologies

Agenda

• Welcome and Introductions
• OLAP & Essbase Basics
• Installation
• Cube Building & Administration
• Partitioning
• Tuning and Optimization
• Security
• Related Tools
I. Welcome and Introductions

Instructor and Class Participants

• Who are you?
  • Name
  • Company
  • Role

• What is your prior experience
  • OLAP
  • Business Intelligence
  • Data Warehouse Design

• What are your expectations for this class?
Course Audience

- This course is designed for:
  - Technical architects
  - Technical business analysts
  - Configurators or developers
  - Application administrators
  - Database administrators

Workshop Goal

Enable participants to install, design, customize and configure Oracle’s Hyperion Essbase by employing recommended methodology and leading practices for successful implementations.
Course Methodology

- Subject matter is delivered by:
  - Lecture and slide presentations
  - Software demonstrations
  - Class discussions
  - Hands-on labs

II. OLAP & Essbase Basics
Basic OLAP “Principles”

- OLAP is commonly used to:
  - Analyze summarized data
  - Analyze numbers
  - Provide an intuitive data model
  - Satisfy the needs of a variety of users
    - Executives, managers, analysts & other business users
    - From power users to novices
- OLAP should not be used to:
  - Store non-numerical data
  - Handle transaction detail
    - On-Line Transaction Processing

Essbase - Beyond the Basics

- Ability to Generate and Synthesize Information
  - Creating, using, storing and providing information to other users and other systems.
- Calculation Capabilities
  - Provides a calculation engine for deriving results and creating aggregations, consolidations and complex calculations.
- Modeling Capabilities
  - Forecasting, trend analysis, and other complex models, such as optimization, statistical analysis or other unique functions.
What is a Cube?

- Intersecting Dimensions
- Form Data Cells
- OLAP Storage Paradigm
- Multidimensional databases are array (or matrix) structures, not related tables
- We talk about cells not fields

Hierarchies and Calculations

Dimensions are organized into hierarchies

- Gross Margin (+)
  - Sales
  - COGS
- Ratios (~)
  - Gross Margin% (~) [Formula: ;"Gross Margin"/"Gross Sales";]
  - Customer Share (~) [Formula: ;"Net Sales"% Customer->"Net Sales")/100;]
**Relationships in a Hierarchy**

- Dimensions and members can be identified by a number of different Familial relationship terms:
  - Parents and Children
  - Siblings
  - Descendants & Ancestors
  - Generations & Levels

- **Parent**
  - Any member that has a child below it

- **Child**
  - Any member that has a parent above it

---

**Relational vs. Multidimensional Metadata**

<table>
<thead>
<tr>
<th>Market key</th>
<th>Description</th>
<th>Region</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>California</td>
<td>W</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>Oregon</td>
<td>W</td>
<td>46</td>
</tr>
<tr>
<td>3</td>
<td>Washington</td>
<td>S</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>Utah</td>
<td>S</td>
<td>56</td>
</tr>
</tbody>
</table>

**Market (Alias: Total for All US Regions)**
- East (+)
  - California (+)
  - Oregon (+)
  - Washington (+)
- West (+)
  - Utah (+)
  - Nevada (+)

*Members and their attributes are referred to as metadata*
Why Essbase?

- **Robust Analytic Engine**
  - Sophisticated calculations – statistical, financial, forecasting
  - Supports write-back – planning, forecasting, "what-if" modeling
  - Financial Intelligence
  - Time understanding
  - Incremental and procedural calculations
- **Consistency**
  - One version of the truth
  - Centralized metadata, data, and business logic
- **Ease of use**
  - Designed for business user – business terminology, no scripting, database structures
  - Relationship aware – navigation, drilling
  - Alternate roll-ups
- **Performance**
  - “Speed of Thought” analysis
  - Highly Scalable
    - User scalability – supports 100’s concurrent users, clustering
    - Data scalability – Aggregate storage mode
- **Integration**
  - Open and standards based
  - Database agnostic

---

Feature Set

- Write back
- Attribute Analysis
- Ragged Hierarchy
- Time Balance
- Expense Tag
- Alternate Hierarchies
- Dynamic Time Series
- Currency Support
- Member Formulas
- Procedural Calculations
- Linked Reporting Objects
- Two Pass Calculations

- Partitioning
- Data Mining
- Drill-through Support
- Hybrid Integration
- SQL Interface
- Java, C & VB APIs
- Scripting Languages
- Localization
- Unicode Support
- Incremental Data load
- Incremental Calculation
- Trigger support
**Essbase Technical Architecture**

Client

- Smart View
- Reporting and Analysis
- Analytic Admin Services

Server

TCP/IP – HTTP – SOAP

- XMLA
- ADM
- C-API
- VB-API

- Analytic Provider Services
- Analytic Services
- Calculator
- Kernel
- Block Storage
- Aggregate Storage
- Shared Services
- Licensing Services

Analytic Administration Services

Relational (Hybrid Analysis & Advanced Relational Access)

MOLAP

**Essbase Functional Architecture**

Client

- Administration
- Analysis
- Reporting
- Spreadsheet

Server

- Language Based Access: MAX L, MDX
- API Based Access: Java, C, VB
- Optimized Storage: ASO / Hybrid / ARA
- BSO / Hybrid
- Rules
- Calculator
- Linked Reporting Objects
- Dimensionality:
  - Shared – via BPMA
  - Shared/Unshared

- Attributes
  - Dimensions
  - UDAs
- Security
  - Data
  - Meta Data
- Partitioning
- Drill-Through
- Linked Reporting Objects
- Data Import
- Meta Data Repository
- Security
- Interoperability
- Logs

Fail-Over and Load Balancing
III. Installation

Essbase Installation

- Pre-requisites
  - License Server (went away in 9.3.1)
  - Shared Services
- Essbase Install Order
  - Essbase Client
  - Essbase Server
  - Essbase Administration Services (server & client)
  - Essbase Integration Services (optional)
  - Analytic Provider Services (optional)
- Post Installation Steps
  - Register products with Shared Services (not required for IS).
  - Configure the database repository (required for AS and IS).
  - Automatically (recommended) or manually deploy to a Java application server (required for AS, IS & APS).
Platform Support for Essbase

- Operating Systems
  - Windows 2000/2003
  - Windows 2003 64-bit – Itanium2 & x64
  - Windows XP/Vista – Clients only
  - Solaris 9/10 – 32/64-bit SPARC V9
  - AIX 5.2/5.3 - 32/64-bit PowerPC
  - HP-UX 11.11 (RISC)
  - HP-UX 11.23/11.31 (Itanium)
  - Oracle Enterprise Linux 4.0
  - Redhat Linux AS/EL 4.0

Platform Support for Essbase – cont’d

- Application Servers (For EAS & APS)
  - Oracle 10g (Release 2 & 3)
  - IBM WebSphere 6.0.2.11/6.1.0.5
  - BEA WebLogic 8.1.6/9.1
  - Apache Tomcat 5.0.28
- Database Repositories (For EAS & EIS only)
  - Oracle 11g
  - Oracle 10g 10.1.0.5/10.2.0.2
  - Oracle 9i-9.2.0.5
  - IBM DB2 8.2/9.1
  - Microsoft SQL Server 2000 SP3a /2005 SP1
- Third-Party Components
  - IE 6/7 & Firefox 1.5.03/2.0.0.3 (EAS)
What’s Included in the Essbase Installation

- Essbase Server
- Sample applications
- Runtime Client
  - Allows communication to remote Essbase Server when accessing with API or scripting language.
- Essbase Spreadsheet Add-in for Excel
- Application Programming Interface (API)
  - Java, C & VB

Exercise: VM Set-up & Install Client Tools
Main Steps

- Unzip and enable VMware Image
- Install Client Tools
  - EIS Console
  - Excel Add-in
  - EAS Console

IV. Building and Administrating Essbase Cubes
Demo: EAS Walk-through

Main Steps

- Launch EAS Console
- Add Essbase Server
- Open Outline
- Familiarize with Features & Functionality
Introduction to Block Storage Option (BSO)

Building an Essbase Cube

- Essbase Load Rules
  - Created in Admin Services
  - Data and metadata sourced from flat files, spreadsheets or relational databases (using SQL Interface)
  - Typically, one Load Rule is created for each dimension build and data load, but these can be combined to a single Rule
  - Extremely flexible, some data transformations capability
- Integration Services
  - GUI, drag and drop environment
  - Data and metadata sourced from relational database (usually a star or snowflake schema)
  - Enables drill-through to detail and hybrid storage functionality
  - Some data transformations are possible.
Building an Essbase Cube – cont’d

- Data Integration Management
  - Informatica PowerCenter
  - Essbase adapter
- API
  - Full set of APIs
  - Java, C & VB
  - Leveraged by partners

Essbase Storage Options

- Block Storage Option (BSO)
  - Original storage mechanism for Essbase
  - For applications that perform interactive planning, allocations, and sophisticated analytics, such as sales forecast applications or profitability analysis applications
  - Where extensive write-backs and procedural calculations are required.
- Aggregate Storage Option (ASO)
  - Developed in 2005
  - For large-scale, sparsely distributed data that is categorized into many, potentially large dimensions
  - Greater flexibility for reporting and analysis
  - Much larger data volumes can be accessed
Multidimensional Databases are Inherently Sparse Structures

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>172</td>
</tr>
<tr>
<td>Time</td>
<td>21</td>
</tr>
<tr>
<td>SalesRep</td>
<td>27</td>
</tr>
<tr>
<td>Channel</td>
<td>32</td>
</tr>
<tr>
<td>Product</td>
<td>209</td>
</tr>
<tr>
<td>Customer</td>
<td>32765</td>
</tr>
</tbody>
</table>

Matrix Size = (172 * 21 * 27 * 32 * 209 * 32765) intersections

21,370,660,375,680 intersection points!

How do we handle that?

BSO Solution: Dense/Sparse Settings

- User tags dimensions sparse or dense.
- Creates unique Essbase storage structures (**BLOCK** and **INDEX**).

  Reflect Density within the block
  Reflect Sparseness within the index
Setting Dense and Sparse Dimensions

Tagging some dimensions dense creates the block

\[ \text{DIM (172, 21, 27, 32, 209, 32765)} \]

Unique combinations of the sparse dimensions form the index

Question, what happens when a dense dimension in tagged sparse vs. when a sparse dimension is tagged dense?

BSO Solution: Dense/Sparse

<table>
<thead>
<tr>
<th>Dim</th>
<th>Members</th>
<th>Type</th>
<th>Structure</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>172</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SalesRep</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>209</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>32765</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### BSO Solution: Dense/Sparse

<table>
<thead>
<tr>
<th>Dim</th>
<th>Members</th>
<th>Type</th>
<th>Structure</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>172</td>
<td>DENSE</td>
<td>Block</td>
<td>ess*.pag</td>
</tr>
<tr>
<td>Time</td>
<td>21</td>
<td>DENSE</td>
<td>Block</td>
<td>ess*.pag</td>
</tr>
<tr>
<td>SalesRep</td>
<td>27</td>
<td>SPARSE</td>
<td>Index</td>
<td>ess*.ind</td>
</tr>
<tr>
<td>Channel</td>
<td>32</td>
<td>SPARSE</td>
<td>Index</td>
<td>ess*.ind</td>
</tr>
<tr>
<td>Product</td>
<td>209</td>
<td>SPARSE</td>
<td>Index</td>
<td>ess*.ind</td>
</tr>
<tr>
<td>Customer</td>
<td>32765</td>
<td>SPARSE</td>
<td>Index</td>
<td>ess*.ind</td>
</tr>
</tbody>
</table>

Setting dense dimensions will cause those dimensions to be stored on disk in the .pag file(s).

Setting sparse dimensions will cause those dimensions to be stored on disk in the .ind file(s).
## Index and Block Modularize the Matrix!

<table>
<thead>
<tr>
<th>Dim</th>
<th>Members</th>
<th>Type</th>
<th>Structure</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>172</td>
<td>DENSE</td>
<td>Block</td>
<td>ess*.pag</td>
</tr>
<tr>
<td>Time</td>
<td>21</td>
<td>DENSE</td>
<td>Block</td>
<td></td>
</tr>
<tr>
<td>SalesRep</td>
<td>27</td>
<td>SPARSE</td>
<td>Index</td>
<td>ess*.ind</td>
</tr>
<tr>
<td>Channel</td>
<td>32</td>
<td>SPARSE</td>
<td>Index</td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>209</td>
<td>SPARSE</td>
<td>Index</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>32765</td>
<td>SPARSE</td>
<td>Index</td>
<td></td>
</tr>
</tbody>
</table>

- Dense Dimensions make up the block
- Sparse Dimensions make up the index
- Block creation is driven specifically by sparse member combinations that contain data.
- Block size: \((21 \times 172) \times 3612 \text{ cells} \times 8 \text{ bytes/cell} = 28,896 \text{ bytes} \approx 28.2 \text{ k}\)
- Computer now able to move sections (block and index pages) of the matrix in and out of memory with efficiency

## BSO Design Objective #1

*Reflect data density in the block and data sparseness in the index.*

*Create the fewest number of blocks with the highest density.*

There is a relationship between block density and block count!
Exercise: Build BSO Cube using EIS

Main Steps

• Connect to the Data Warehouse
• Build an OLAP Model
  • Facts
  • Dimensions
  • Hierarchies
• Build a Metaoutline
  • Apply multidimensional attributes
• Create the Essbase Cube
• Query the Cube using EAS or Add-in
Introduction to Aggregate Storage Option (ASO)

Why was ASO Developed?

- Historically... Essbase Geared Toward Financial Applications
  - Fairly Dense Data
  - Support Hyperion's Applications Nicely
- For Some Applications
  - Essbase Requires Ability to Handle Extreme Sparsity
  - Current Engine Not Optimal
  - Outline Size Limited
  - Small Batch Windows
  - Increased Data
Multi-dimensional Applications…
Challenging Applications for “Block Storage”

- Customer Analysis
  - Many dimensions for analysis, and potentially millions of customers

- Sales Forecasting
  - Write back intensive, what-if analysis, each user creating and managing their own forecast

- Procurement Analysis
  - Many products being tracked across many vendors

- Profitability Analysis
  - Cost allocations across products and customers

- Logistics Analysis
  - Near real time updates of product shipments

- Financial Consolidations
  - Currency conversions, inter-company eliminations

We Needed a New “Mouse Trap”

- 100x Data Scalability over Essbase 6X
  - More Dimensions
  - More Members per Dimension

- 10x Faster Load & Calc
  - Allow for a “balance” between query performance and disk space

- Optimization of Sparse Data Sets
- 20x Smaller Footprint
Best-in-Class Multi-dimensional Analytics

- What-if?
- Business rules
- Forecasting
- Allocations
- Currency

Enterprise Analytics

- Massive data volumes
  - Real time
  - Data Mining
  - Navigation
  - Sparse data

How Does Aggregate Storage Work?

- Load data only at level 0
- Read-only technology
- Create aggregate views

- Algorithm selects and stores “most expensive” queries
- Queries are dynamic at runtime, leverage the nearest stored view
Breakthrough Performance at HBOS

Sparse Dims
57 x 2254 x 6544 x 12826 x 13930

Aggregate Storage was 395X faster on load and calc, and required 580X lower disk storage compared to Block Storage

New Metadata Scalability Delivers Detailed Insight

- Delivered at major bank in partnership with IBM
- Instantaneous query performance
- The power to maximize insight for everyone

Account Level Analysis

14,000,000 customer accounts

3/4 billion rows of data
**Outlines in ASO**

<table>
<thead>
<tr>
<th></th>
<th>BSO</th>
<th>ASO</th>
<th>Determined By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max # members in outline</td>
<td>Up to 3 million</td>
<td>Up to 15 million</td>
<td>Memory available and size of member/alias names; ASO outline paging</td>
</tr>
<tr>
<td>Size of outline on client</td>
<td>Larger</td>
<td>Smaller</td>
<td>ASO max 5000 leaf node members in memory at one time</td>
</tr>
<tr>
<td>Size of outline on EAS or Essbase server</td>
<td>Larger</td>
<td>5 x Smaller</td>
<td>ASO outline paging</td>
</tr>
<tr>
<td>Size of outline on disk</td>
<td>Smaller</td>
<td>2-4 x Larger</td>
<td>ASO outline pages contain more free space than BSO outline pages</td>
</tr>
</tbody>
</table>

**What’s Out of Scope**

- Write back *
- Expense Tag *
- Dynamic Time Series *
- Currency support
- Member formulas (Stored)
- Procedural Calc
- Incremental member Restructure
- LRO *
- Mix & Match Storage types in the same application
- Two Pass Calc

* This functionality is available when using partitioning
Exercise: Build ASO Cube

Main Steps

- Create ASO App/DB
- Build outline
  - Using EAS Load Rules
  - SQL Interface
  - Flat Files
- Load Data
- Query the Cube using EAS or Add-in
V. Partitioning

Introduction - Partitioning

- Sharing/Distribution of Data Across Applications
- Different DB - different Apps
- Different DB - different Apps/Servers

Shared Partitions

Data Source  Data Target
Partitioning Types

- Replicated
- Transparent
- Linked

Replicated Partitioning

Slice of Source DB copied to Target DB
Replicated Partitioning

• Advantages
  • Offset database failures
  • Reduction in network activity (proximity of data)
  • Maintenance flexibility for owners of source data

• Disadvantages
  • Increased disk space
  • Stale data - refresh must be batched

Transparent Partitioning

Data retrieved via Target DB located in Source DB
### Transparent Partitioning

- **Advantages**
  - Reduce use of disk space
  - Target data up to date with source data
  - Parallel loads/calcs - increased performance
  - Location of data unknown to user
  - Smaller DBs - avoid irrelevant dimensions/members distributed control and maintenance, master view of all data

- **Disadvantages**
  - Increased network activity
  - Source DB failure - crashes target DB
  - Dual maintenance/calcs on both source and target

### Linked Partitioning

Drill across to data cells in another DB

![Linked Partitioning Diagram]

- Spreadsheet User
  - Spreadsheet 1 (Corporate Sales)
  - Spreadsheet 2 (East Database)

[Data Target] [Data Source]

Linked Cells
**Linked Partitioning**

- **Advantages**
  - Navigate data in wrapped in different context/dimensions
  - Outline synchronization less of a factor
  - Good retrieval performance
  - Avoid DBs with irrelevant dimensions/members
  - Parallel load/calcs
- **Disadvantages**
  - Drill across not invisible to user
  - Security management complexities
  - Reporting tools may not support functionality

---

**BSO and ASO Combined/Blended Solution**

To maximize functionality, **combine**

- **Historical and Forecast**
  - Federated View
  - Aggregate or Block Storage
- **SKU Product Profitability**
  - Aggregate Storage Option
- **Product Profitability Forecast**
  - Dense/Sparse Storage Option
- **Highly additive and dimensional**
- **High write back and analytical**
- **Transparent Partition**
- **Multi user write back**
When to Partition

- Should the data be closer to the people who are using it? Is the network being stressed because users are accessing data that is far away?
- Would a single failure be catastrophic? If everyone is using a single database for mission-critical purposes, what happens if the database goes down?
- Does it take too long to perform calculations after new data is loaded? Can you improve performance by spreading the calculations across multiple processors or computers?
- Do users want to see the data in different application contexts? Would you like to control how they navigate between databases?
- Do you have separate, disconnected databases storing related information? Does the related information come from different sources? Are you having trouble synchronizing it?
- Will you add many new organizational units? Would they benefit from having their own databases? Partitioned databases help you grow incrementally.
- Are users having to wait as other users access the database?
- Do you want to save disk space by giving users access to data stored in a remote location?
- Should you reduce network traffic by replicating data in several locations?
- Do you need to control database outlines from a central location?

When NOT to Partition

- Do you have resource concerns? For example, are you unable to purchase more disk space or allow more network traffic?
- Do you perform complex allocations where unit level values are derived from total values?
- Are you required to keep all databases online at all times? Keeping databases online can be a problem if you have databases in several time zones, because peak user load may differ between time zones. Using linked and transparent partitions exacerbate this problem, but using replicated partitions might help.
- Are the databases in different languages? Essbase can only partition databases if both databases use the same language, such as German.
Design Considerations for a Partition

- Which database should be the data source and which the data target? The database that "owns" the data should be the data source. Owning the data means that this is the database where the data is updated and where most of the detail data is stored.
- Are some parts of the database accessed more frequently than others?
- What data can you share among multiple sites?
- How granular does the data need to be at each location?
- How frequently is the data accessed, updated, or calculated?
- What are the available resources? How much disk space is available? CPUs? Network resources?
- How much data needs to be transferred over the network? How long does that take?
- Where is the data stored? Is it in one location or in more than one location?
- Where is the data accessed? Is it in one location or in more than one location?
- Is there information in separate databases that should be accessed from a central location? How closely are groups of data related?

Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Replicated</th>
<th>Transparent</th>
<th>Linked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up-to-the-minute data</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reduced network traffic</td>
<td>x</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>Reduced disk space</td>
<td>.</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Increased calculation speed</td>
<td>x</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Smaller databases</td>
<td>.</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Improved query speed</td>
<td>x</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>Invisible to end users</td>
<td>x</td>
<td>x</td>
<td>.</td>
</tr>
<tr>
<td>Access to databases with different dimensionality</td>
<td>.</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>Easier to recover</td>
<td>x</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Less synchronization required</td>
<td>.</td>
<td>.</td>
<td>x</td>
</tr>
<tr>
<td>Ability to query data based on its attributes</td>
<td>.</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Exercise: Partition ASO -> BSO Cube

Main Steps

- Run Partition Wizard
- Query the Cube using EAS or Add-in
VI. Advanced Calculations

Sophisticated Analytics

- Forecasting & trending
  - Multi user write-back
- Cross Dimensional Allocations
- Attribute cross-tab analysis
  - Enhances customer, product, financial reporting
  - sum, count, max, min, average
- Cross Cube Calculations & analysis
  - Share cube information
  - e.g. Inflation, Tax, Currency
- Expanded statistical functions
  - medians, modes, correlations, standard deviations and more
- Over 200 “Out-of-the-box” functions
  - Extensible calculation engine
- Data Mining
Calculations in Essbase (BSO)

- **Outline Formulas**
  - Used for less complex calculations
  - Follow outline relationships or formulas
  - Calculate entire database

- **Calc Scripts**
  - Calculate all or a portion of the database
  - Control the order in which dimensions are calculated
  - Perform complex calculations
  - Define calculations other than the calculations that are defined by the database outline

Calc Script Sections

- **Information**
  - Comment syntax (/*...*/)

- **Housekeeping**
  - SET commands control how the script processes
    - SET MSG, SET NOTICE, SET UPDATECALC, SET CALCPARALLEL #
  - Data manipulation commands prepare input data for calculation
    - CLEARBLOCK, CLEARDATA, DATACOPY

- **Baseline Fix**
  - Defines focus of the calculation
    - FIX()...ENDFIX
  - Nesting permitted
Calc Script Sections – cont’d

• Normalization
  • Prepare data for consolidation
    • Allocate upper-level inputs to level 0 blocks
    • Make adjusting entries
    • Facilitate intercompany eliminations
    • Correct or otherwise fix anomalous data

• Main Rollup
  • Consolidates dimensions
  • Uses CALC DIM or CALC ALL commands

• Back/Post Calculations
  • Correcting consolidated values (Two-pass calculations)
  • Apply advanced calculations on consolidated data
    • @TREND, @SPLINE, @IRR, etc.

Calculation Commands

• Data Declarations
  • ARRAY, VAR

• Control Flow
  • EXCLUDE...ENDEXCLUDE, FIX...ENDFIX, LOOP...ENDLOOP

• Functional
  • AGG, CALC DIM, CLEARDATA, DATACOPY, SET UPDATECALC

• Conditionals
  • IF, ENDIF, ELSE, ELSEIF

• Member Formulas
  • Expenses = Payroll + Marketing;
  • Profit%;
Calculation Functions

- Boolean
- Relationship
- Calculation Operators
- Mathematical
- Member Set
- Range and Financial
- Allocation
- Forecasting
- Statistical
- Date & Time
- Miscellaneous
- Custom-defined

Exercise: Create Calc Scripts
Main Steps

• Create a calc script using the @MDALLOCATE function to create a baseline budget.

• Create a calc script using the @TREND function to create a forecast for 2008.

VII. Tuning and Optimization
What is BSO Essbase Optimization?

Batch Process vs. Planning Process vs. Query Response

• Three Primary Areas
  • Optimizing the outline
  • Optimizing DB caches and settings
  • Optimizing the calculation script

Art form approached systematically if not scientifically.

Database Design: The Outline

• It is said that 80% of tuning Essbase should be done during the design
• Dimensionality Considerations
  • How many, how large, how deep
• Outline Consolidation and Business Rules (Formulae)
  • Formulae in the outline vs. calc script
  • Member tags
    • Dynamic Calc non-store in block
    • Label Only on navigational members
• Dense/Sparse settings
Database Design: Refining the Outline

• Dimensionality Considerations
  • Minimize the number of dimensions
  • Eliminate inter-dimensional irrelevance
  • Reduce “flatness” in dimensions
  • Order of the dimensions (hourglass shape):
    • Any dense dimensions tagged as “Accounts” first
    • Any dense dimensions tagged as “Time” second (if it is dense)
    • All remaining dense dimensions from largest to smallest
    • Sparse dimensions form smallest to largest

Database Design: Refining the Outline – cont’d

• Outline Consolidation and Business Rules (Formulae)
  • Store data only when necessary
    • Use Label Only and Dynamic Calc tags to reduce physical block size (logical block size is not affected)
  • Use Dynamic Calc when:
    • Upper level dense members
    • Dense members with formula
    • Dynamic Calc (and Store) & Sparse Members
      • Sparse member with small fan-out (<7)
    • Dynamic Calc tags on Sparse Dimensions needs to be implemented with extreme caution:
Database Design: Refining the Outline – cont’d

• Dense/Sparse settings
  • Theoretical vs. Practical
    • Some database configurations will be optimal not according the best sparse / dense configuration but according to requirements specific to the model at hand
  • Monitor block size (<100K on Windows, <200K on UNIX), block density, total number of blocks

Calc Script Optimization

• Isolate specific sections of database to increase calculation efficiency:
  • FIX…ENDFIX focus on block subsets
    • FIX whenever possible!
  • IF…ELSE…ELSEIF…ENDIF fine tunes logic
    • Use IF to resolve logic where FIX cannot be used
  • CROSS DIM OPERATOR -> granularity to cell level
    • Can be expensive if across sparse dimensions
    • Try to keep within block
  • Use combinations of FIX and IF as required
Is there a difference?

Each of the following pieces of code will achieve the same results:

A: 
```plaintext
FIX (California, Nevada)
Sales
if ("Jan-2007" and Budget)
  Sales = Sales*1.15;
endif
ENDFIX
```

B: 
```plaintext
FIX (Budget, California, Nevada, "Jan-2007")
Sales=Sales*1.15;
ENDFIX
```

The B code runs in ~25% less time!

Essbase Caches Overview

- **Index Cache**
  - Repository of index pages in Server RAM
  - Add the sizes of all ESS*.IND files and use the sum for the index cache
  - If this is too large, make the index cache 50-75% or as large as possible

- **Data Cache**
  - Repository of blocks in Server RAM
  - 0.125 x sum of the ESS*.PAG or as large as possible

- **BSO Essbase Storage Manager:**
  - Manages the index and data caches
  - Provides addressability to data blocks for server functional layers such as the calculator and the reporting module.

- **Calculator Cache**
  - Roadmap for Block Creation
  - Tracks Children and Parents
ASO Optimization

- Optimizing the outline
- Optimizing query performance
  - Aggregate Views
  - Query Tracking
- Optimizing buffers and caches
  - Data loads
  - Outline compaction and paging

ASO Outline

- Dimension Types
  - Stored
    - Best used for speed of Aggregation
    - Only (+) aggregation allowed
    - No Formulas
    - Accounts can’t be Stored
  - Dynamic
    - Calculated not aggregated
    - All calcs done at retrieval time
    - Multiple Consolidation Symbols (+, -, /, etc)
    - Formulas (MDX)
    - Not part of Aggregate View Selection
Aggregate Views

• Materialization
  • Default
    • For the default selection, Essbase analyzes stored hierarchies and assumes an equal chance that any aggregate cell will be retrieved
    • Doesn't Guarantee best performance.
  • Query Tracking
    • AS “watches” what is retrieved for duration of Query Tracking allowing optimization based on what is pulled in.
    • Query Tracking is done in memory and stops if any of the following occur
      • Loading or clearing data
      • Materializing or clearing an aggregation
      • Turning off query tracking
    • During batch process a report script can be run to simulate user requests to help aid in materialization

Data Loading

• Data Load
  • Uses a data load buffer that organizes the data
    • Multiple files can be added in to the buffer
  • Buffer commits to the database
  • If values have been calculated and stored through an aggregation, Essbase automatically updates higher-level stored values when data values are changed. No additional calculation step is necessary. The existence and size of an aggregation can affect the time it takes to perform a data load.
    • It is recommended that you delete aggregations first, load data, then re-aggregate in a separate process
  • Incremental Load
    • Coming soon
Outline Optimization

• Outline Compaction
  • ASO Outlines get very big!
  • Deleted members remain as part of the outline until Compaction happens

• Outline Paging
  • Paging an outline into memory enables Essbase to handle very large outlines, but potentially increases data retrieval time.
  • Outline paging cache default set to 8 MB
  • If hit ratio on cache is low (<90%), increase cache in multiples of 8 MB

Exercise: Tuning cubes
Main Steps

- Review outline properties
- Reset caches
- Restructure cube

VIII. Security
Essbase Security Options

• Native Security Mode
  • Default option
  • Security managed using EAS

• Shared Services
  • Must first migrate any Essbase Server applications and any existing Essbase users and groups to Shared Services.
  • User provisioning across multiple products
  • External authentication
  • Single sign-on

Essbase Security Layers

• Server-wide settings
  • Manage the activities of users connected to the Essbase Server such as session and request management, lock management, connection management, and password and user name management

• Application and database settings
  • Set minimum permissions for applications and databases

• Users and groups
  • These permissions take precedence over minimum permissions defined for applications and databases

• Database filters
  • Define database permissions that users and groups can have for particular members, down to the individual data value (cell)
Permission Settings

• None
• Metaread
• Read
• Write
• Calculate
• Designer
• Administrator
• Additional permission at the User/Group level
  • Filter Access

Essbase Filters

• Control security access to data values or cells
  • For one or more dimensions, specify member or list of members and apply access level associated with them
• Filter access levels
  • None
  • Read
  • Write
  • Metaread
    • Overrides all other access levels
Shared Services Components

- Databases (relational and OpenLDAP)
- Web application server
- Configuration Utility™
- External Authentication Configuration Console
- Shared Services User Management Console™

Shared Services – User Provisioning

- Role based
- Centrally managed security
- Distributed security model
- Integrates with external systems
**External Authentication**

To use external authentication, you must have an authentication directory containing corporate user information. The following types of authentication repositories are supported:

- Oracle Identity Management (OIM)
- Lightweight Directory Access Protocol (LDAP)
  - Sun ONE, Novell eDirectory, IBM Directory Server, Domino LDAP
- Microsoft NTLM and Active Directory (MSAD)
- IBM Tivoli Directory Server
- SAP Enterprise Portal, SAP R/3, Netweaver BI (SAP BW)
- OpenLDAP 2.3.37

---

**Hyperion System 9 Foundation Shared Services Technical Architecture**

Client diagram showing:

- Browser (JavaScript)
- Client (WebDav)
- Custom Application

Server diagram showing:

- Struts Framework
- WebDav Servlet
- Security Abstraction Layer
- Metadata Abstraction Layer
- Security Client
- Content Management Service

Repository diagram showing:

- Native
  - OpenLDAP
- External
  - LDAP
  - MSAD
  - NTLM
- RDBMS
Exercise: Security

Main Steps

• Create filter
• Create user/group
• Query cube to verify new security settings
• Migrate to Shared Services
IX. Related Tools – Analytic Provider Services (APS)

Analytic Provider Services

- Smart View Client
- BI+ Tools
- Partner/Custom XMLA Tools
- Partner/Custom Java Tools

Java API
Smart View Communication
XMLA API

High Availability / Failover

Analytic Provider Services

Stand Alone or Clustered Data Source
Leveraging Database Clusters

- **Database clustering**
  - Read only
  - Transparent to end user
  - Query aware
    - State maintained in APS
- **Failover**
  - During failure, connection rerouted to active instance
  - Disable and enable data sources from UI or utility

High Availability Scenario

1. Procedural calcs are run, where necessary to get data to level zero and execute member formulae.
2. Data is exported to disabled cluster agent for upper level calcs and/or aggregations. During export/copy write-back is disabled.
3. Once data is aggregated, completed agent is toggled to active and second cluster agent is set to disabled.
4. Process repeats from write-back instance on scheduled intervals.
Failover Scenario

When one node fails, APS reroutes any current query and all subsequent queries away from the bad node until it returns to a good state.

Connection Pooling

Analytic Provider Services

Connection Pool 1

Connection Pool 2

The fully operative database deployment strategy prevalence get a single connection pool for each database. Conceptually, it's like having a single connection pool for each database. Instead, there are multiple connection pools, each managing a specific subset of database connections. This approach optimizes resource utilization and ensures efficient failover and load balancing.
X. Related Tools – System 9 BI+

Oracle BI Suite Enterprise Edition Plus

Common Enterprise Information Model

Oracle BI Server
- Intelligent Caching Services
- Optimized Data Access Services
- Intelligent Request Generation

OLTP & ODS Systems
- Data Warehouse
- Data Mart
- SAP, Oracle, PeopleSoft, Siebel, Custom Apps

Files
- Excel
- XML

Business Process
- Financial Performance Management Applications
- Hyperion Essbase
Hyperion System 9
The Management System for the Global Enterprise

- Integrates Financial Management Applications with a BI Platform
- Single Workspace = Simplification
- Broadest BI Functionality
- Foundation Services and Data Integration for Data Consistency and Synchronization

Focus on Hyperion System 9 BI+

- One Client
  - A single web based environment to deploy, train and use
- Full Interaction
  - Users can work with applications, explore data and create reports by the Web
- MS Office Integration
  - Seamless connection between information and end user productivity tools
A Single Web-Based Interface for All Modes of BI

Dashboards ▪ Ad Hoc Query ▪ Production Reporting ▪ Financial Reporting ▪ Scorecards
▪ Advanced Analytics ▪ Data Mining ▪ Visualization ▪ Metrics Management

“Windows on the Web” User Experience

• Consistent
  • Any mode of BI
  • Any user profile
  • Online & disconnecte

• Interactive
  • Drag & Drop
  • Drill down & across
  • Pivot
  • Filter

• Efficient
  • Pure thin client
  • Minimal round-tripping
Transparent User Access to Any Source

- Relational
- Multidimensional
- Financial Management Applications

Self-Service Report & Dashboard Creation

Business Analyst

- Click
- Point
- Generate

- Components
- Existing Queries and Reports
- Templates
Lifecycle Management

- Lifecycle Management is technology and methodology needed to create a consistent way to move changes from development to production
- Three components:
  - Change Management
    - Version and check-in of artifacts
    - Audit Trails
    - Change Reports
  - Process Management
    - Review and Approval Process
    - Notifications
  - Configuration Management
    - Configuration of the process
    - Automation of builds and artifact loads
    - Packaging of artifacts

Single Point of User Management & Security

- Create user or utilize ‘corporate’ user definition
- Assign user to group or utilize ‘corporate’ group assignment
- Assign user ‘roles’; one set of common roles across modules
- Provision users to access application domains
Scalable Service Oriented Architecture

Standards-based
- SOAP, XML & WSDL
- J2EE & .NET Development

Scalable, High Performance
- Distribute and replicate components
- Services optimized for task-at-hand
- 64 Bit Support

Flexible User Models
- High interactivity thin client for consumption
- Rich studio products for authors

Hyperion BI+ Modules

- Direct to Source Query & Reporting
- High Volume Production Reporting

- Ad-hoc Multidimensional Analysis
- Integrated with FPM Applications
- Financial Dashboards

- XBRL Support
- Asymmetrical Reports
- Management Report & Books

- Excel, PowerPoint, Word and Outlook
- Integrated with FPM Applications
- Offline Planning
Hyperion Interactive Reporting

- Ad-hoc query and analysis for relational data sources
- Broad range of data visualization
  - Graphs and charts
  - Pivots
  - Traffic lighting
  - Dashboards with template based wizard
- Supports mobile users through offline capability
- Spotlight exceptions to identify and monitor performance

Hyperion SQR Production Reporting

- **High Performance**
  - Large data volumes
  - Data processing in memory
  - Single data pass for all outputs
- **Multiple Data Sources / Output Types**
  - RDBMS, Cubes, Files, SAP
  - HTML, PDF, RDBMS, Printers, Interactive Reporting
- **Advanced Reporting Capabilities**
  - Report Surfing via Web
  - Pixel-Perfect Output (bills, invoices, bar codes, checks/receipts)
  - Conditional formatting & printing
  - Secure Report Bursting
  - Unicode support (single/double byte)
- **Report Lifecycle Management**
  - Generate, publish, categorize, version, index, archive, delete
- **Personalized Distribution**
  - Subscribe, Notify, Alert
  - UI highlights relevant content to user
Hyperion Web Analysis

- Intuitive, powerful and highly graphical online analysis for all types of users
- Advanced analysis features such as sorting, ranking, filtering, and user-defined calculations
- Variety of visualization options: spreadsheet, charting, animated pinboards, and traffic lights
- Supports Hyperion Applications, Hyperion Essbase Analytics, Hyperion Enterprise Analytics and SAP BW

Hyperion Financial Reporting

- Designed for end users to quickly produce book quality financial and management reports
- Out of the box financial functions can be easily embedded in reports
- Financial formatting for planning and forecast reporting
- Easily navigate and search corporate structures such as product or business units to create individual reports
- Wide range of output options, including HTML, PDF and online viewing
- Supports Hyperion Applications, Hyperion Essbase Analytics, Hyperion Enterprise Analytics and SAP BW
Hyperion System 9 SmartView Makes BI Pervasive in Microsoft Office

Web Based Portal Support
Best-in-Class Visualization = HVE

On the Horizon...Hyperion Smart Space
The Industry’s First Composite Application Framework for BI

- Uses the SOA of Hyperion System 9
- Libraries of BI Gadgets to be built by Hyperion and Partners
- Defined by role: Sales, Marketing, Finance, Operations
- Real-time update, bi-directional
- Process aware: Planning, Forecasting, Closing
- Built-in Collaboration