Premier IT Professional Service and Solution Provider of Oracle Applications and Technologies

- E-Business Suite Consulting
- Enterprise Apps Managed Services
- OBIEE, Pre-Built BI Analytics
- Hyperion EPM
- Middleware & Integration
- Infrastructure Services
- Microsoft .Net Dev & Support
- Apps University

*Selected by Oracle as BI Pillar Partner*

- High Value ROI
- Best Practice Methodology
- Local / Global Service Delivery
- Subject Matter Experts
The following is intended to outline Speaker's ideas on the topic. 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.
Agenda

- OBIEE System Architecture Overview
- Performance Expectations
- Approach to Performance Issues
- Tuning Tips
- Question and Answers
Typical BI Farm

[Diagram showing a typical BI farm with components like EBS, SAP, Data warehouse, ETL, OBI-EE, Weblogic, Exbase, Hyperion Application, and Disaster Recovery Location with Oracle Data Guard]

Source Systems
- Database Tier
- Middle Tier

Disaster Recovery Location
- Data warehouse
- Middle Tier
- Firewall
Agenda

• OBIEE System Architecture Overview
• Performance Expectations
• Approach to Performance Issues
• Tuning Tips
• Question and Answers
Performance Expectations

- Response Time
- Optimized Design
- BI Server Performance
- Database Performance
- Iterative Approach
Common Issues

- Report Running for Long time
- OBIEE Not Responding
- Logging in...
- BI Publisher bursting delayed
- Specific iBot failed
Agenda

• OBIEE System Architecture Overview
• Performance Expectations
• Approach to Performance Issues
• Tuning Tips
• Question and Answers
Sources for Debugging

- Logs
  - NQServer.log
  - NQQuery.log
  - Saw0.log
  - Java Host Log
  - Application Server Log
  - Scheduler log
  - Trace Files
- Usage Tracking
- Performance Counters
- Network trace
- Resource Utilization in EM
Agenda

- OBIEE System Architecture Overview
- Performance Expectations
- Approach to Performance Issues
- Tuning Tips
- Question and Answers
Variable Management

• Scope of the Variable
• Consolidate Initialization Blocks where possible
• Disable unused Initialization Blocks
Report Modeling

- Column Filters
- Union Reports
- Non Cacheable SQL Element
- Large Result Sets
- Prompts on Fact Tables
Data Modeling

- Snowflake to Star
- Detail vs. Aggregate tables
- Opaque Views and Materialized Views
- Managing Joins
  - Driving Table
  - Outer Joins
  - Foreign Key vs. Complex Join in Logical Layer
Infrastructure

- **CPU**
  - Chips, Cores, Cores/Chip, Multithreading, Clock Speed
- **Memory**
  - Emulation, Size VM
- **Disk**
  - RAID, I/O Controller
- **Network**
  - Latency, Compression
Operating System Tuning

• Release TCP/IP closed connections faster
• Tune TCP Wait times
• Increase File descriptors
• Increase Backlog connections queue
• Adjust the MaxUserPort
• Power Options to High Performance
• Enable /3 GB Switch for 32-BIT Operating systems
J2EE Server Tuning

- Configure each application on separate OC4J container.
- Configure Multiple JVMs
- Tune Heap Memory Size
• Tune HTTP Server Compression / Caching
  ▪ Why use Web Server Compression / Caching for Oracle Business Intelligence?
    ▪ Bandwidth Savings
    ▪ Improves request/response latency
# Benchmarks with Compression

<table>
<thead>
<tr>
<th>Pages</th>
<th>HTTP Response Size (Kbytes)</th>
<th>HTTP Response Size with Compression (KB)</th>
<th>Compression ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard with 3 Tables and 3 Charts (each table has 5<del>10 rows, 3</del>5 cols)</td>
<td>297.5</td>
<td>39</td>
<td>86</td>
</tr>
<tr>
<td>Dashboard with 1 Table (25 rows, 10 columns)</td>
<td>210</td>
<td>28.5</td>
<td>86</td>
</tr>
<tr>
<td>Dashboard with 1 Large Table (300 rows, 10 columns)</td>
<td>938</td>
<td>79</td>
<td>91</td>
</tr>
</tbody>
</table>
Database Tuning

• Initialization Parameters
• Index Strategy
• Optimal Redo log sizing
• Statistics
• Layout
Database Tuning

• Initialization Parameters
  ▪ PARALLEL_DEGREE_POLICY
  ▪ OPTIMIZER_MODE
  ▪ OPTIMIZER_INDEX_COST_ADJ
  ▪ OPEN_CURSORS
  ▪ MEMORY_TARGET
```sql
SQL> select * from v$memory_target_advice order by memory_size;
```

<table>
<thead>
<tr>
<th>MEMORY_SIZE</th>
<th>MEMORY_SIZE_FACTOR</th>
<th>ESTD_DB_TIME</th>
<th>ESTD_DB_TIME_FACTOR</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>0.5</td>
<td>458</td>
<td>1.344</td>
<td>0</td>
</tr>
<tr>
<td>270</td>
<td>0.75</td>
<td>367</td>
<td>1.0761</td>
<td>0</td>
</tr>
<tr>
<td>360</td>
<td>1</td>
<td>341</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>450</td>
<td>1.25</td>
<td>335</td>
<td>0.9817</td>
<td>0</td>
</tr>
<tr>
<td>540</td>
<td>1.5</td>
<td>335</td>
<td>0.9817</td>
<td>0</td>
</tr>
<tr>
<td>630</td>
<td>1.75</td>
<td>335</td>
<td>0.9817</td>
<td>0</td>
</tr>
<tr>
<td>720</td>
<td>2</td>
<td>335</td>
<td>0.9817</td>
<td>0</td>
</tr>
</tbody>
</table>
• Initialization Parameters
  ▪ PARALLEL_DEGREE_POLICY
  ▪ OPTIMIZER_MODE
  ▪ OPTIMIZER_INDEX_COST_ADJ
  ▪ OPEN_CURSORS
  ▪ MEMORY_TARGET
  ▪ Processes
  ▪ USE_LARGE_PAGES
Database Tuning

- **Index Strategy**
  - Goal to Optimize Query performance
  - Understand Star Schema Queries
    - So how do we go about optimizing these queries?
    - Tuning a star query has two important criteria, they are:
      - Create a bitmap index on each of the foreign key columns in the fact table or tables
      - Set the initialization parameter STAR_TRANSFORMATION_ENABLED to TRUE.
Database Tuning

• Optimal Redo log sizing
  ▪ The size of the redo log files can influence performance.
  ▪ Larger redo log files provide better performance.
  ▪ Rule of thumb increase Redo log size so that not more than 3 log switches per hour.
Database Tuning

• Statistics
  ▪ Gathering Table and Index statistics is key for optimizer to build optimal execution plan.
  ▪ Gather Workload statistics on Source and Target databases.
    ▪ Desired system statistics are computed when database is under significant workload.
  ▪ BIAPPS Customers can automate gathering index statistics during ETL
    ▪ Edit the customsql.xml under DAC_HOME
      ▪ D:\app\oracle\product\10.1.3.4PRODDAC\bifoundation\dac\CustomSQLs\customsql.xml
    ▪ Set cascade=>True
• Tablespace Layout
  ▪ Isolate tablespace for FACT and DIMENSION tables
  ▪ Isolate Index tablespace for FACT and Dimension tables
  ▪ TEMP and UNDO data files should be on RAID 1 for faster I/O
The most important performance and scalability features are:

- CONNECTION POOLING
- QUERY REUSE AND CACHING
- Configuration Parameters
- Statement Cache
• Connection Pooling
  ▪ BI Server can be configured with one or more connection pools for each database.
    ▪ Specific users or groups can be assigned to specific connection pools.
    ▪ An administrator can give certain groups higher priority.
  ▪ Limit maximum number of connections to keep open.
    ▪ This setting will prevent database servers from being overloaded.
BI Server Tuning

- Query / Cache Tuning
- Oracle BI Server to intelligently re-use previous query results, a capability called "query caching".

![Diagram showing ORACLE BI Server, Web Server, BI Server, Database, and Database Server with caching arrows between them.]
BI Server Tuning...

• BI Server Configurations
  ▪ Increase session limit based on Client connections.
  ▪ Define proper limits for server and database threads.
  ▪ Use faster disks to data cache storage
  ▪ Cache aggregate rollup hits
  ▪ Define proper cache entry size
  ▪ Define Temporary path to faster disk
    ▪ Ex. /dev/shm on Linux
BI Server Tuning...

- Statement Cache
  - TUNE THE STATEMENT CACHE SIZE
    - If Oracle database is used set Statement Cache Size to 0 or a lower value.
    - Oracle JDBC Driver consumes lot of memory when calling statements from the Weblogic Server Cache.
    - Setting the statement cache size to 0 will disable Weblogic Server Cache.
      - For all other databases ex. Sql server maintain the default value for Statement Cache.
Partitioning and MVs

• Partition Advantages
  ▪ Partition Pruning
  ▪ Partition - Wise Joins
  ▪ Reduce Maintenance times for Indexes
  ▪ Improve Web query performance

• Materialized views speed up query performance
  ▪ Build Summary views for expensive joins and aggregation operations

• Partition materialized views to take advantage as partition tables
Partitioning and MVs...

• Compression for MVs
  ▪ Compression can be employed
    ▪ Reduces the storage space
    ▪ Faster access of data
Speaker Information

• Narasimha Rao Madhuvarsu L V
  ▪ Email: narasimha.mlv@appsassociates.com

• Bharath Terala
  ▪ Email: bharath.terala@appsassociates.com
Contact:
Bill Saltys, VP Business Development and Operations

Email: Bill.Saltys@AppsAssociates.com
Phone: (978) 399-0230 (O)

Apps Associates LLC
40 Nagog Park,
Acton, MA 01720
USA