Hybrid OLAP, An Introduction

Richard Doherty
SAS Institute
European HQ
Agenda

- Hybrid OLAP overview
- Building your data model
- Architectural decisions
- Metadata creation
- Report definition
Hybrid OLAP overview

• What is Hybrid OLAP (HOLAP)?

• Why is it so good?
What is HOLAP?

- HOLAP takes the best features from Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP)
- MOLAP applications typically exploit single “cubes”
  - SAS/EIS Multidimensional report
- ROLAP applications exploit relational data stores
  - SAS/EIS Motore extension
What is HOLAP?

• HOLAP applications exploit multiple cubes and relational data stores on multiple servers transparently.

• OK, but what’s good about that?
• Users of C/S packaged MOLAP were faced with two issues:
  – optimising performance
  – enhancing scalability
MOLAP issues

- A typical C/S packaged OLAP implementation looked something like this:
The answer

- In response to these issues, the Institute has made the HOLAP model available
- Resolves both scalability and performance issues
- How...?
HOLAP architecture
Optimum performance

- HOLAP utilises the servers’ compute resources
- Only the results are downloaded
Enhanced scalability

- Data source can be a combination of datasets, views & MDDBs (star schema support)
- Larger sub-cubes can be stored in datasets
Implementation steps

- Building your data model
- Architectural decisions
- Metadata creation
- Report definition
Building your data model

Product+ → Supplier Group

Sector Group → Region

Family Article → Day

Year Month

Supplier Supplier Group

Shop Geo+
Building your data model

• Data model
  – Hierarchies
  – Classes
  – Cardinality

• Data stores
  – Crossings
    • *NWAY* crossing
    • Aggregations
Hierarchies

- How should our HOLAP application think?
- What questions does it have to answer?
- What hierarchical information do we have available?

Product+  
Sector  
Group  
Family  
Article
Classes

• Levels of the hierarchy
• Parent/child business relationship
Cardinality

- The number of unique values in a class
- Sector has low cardinality
- Article has high cardinality
- Cardinality will influence
  - which “crossings” we store
  - how we store them
  - where we store them
Data stores - crossings

- Sector-Year-Region
Data stores - crossings

- Crossing - point of navigation
- Crossing cardinality will influence our data storage architecture
- HOLAP allows “splitting” by crossing
Data storage architecture

• “Splitting” by crossing, an example
  – Sector-Year-Region & Sector-Year-Region-Supplier group -> MDDB1
  – Sector-Year-Region-Supplier group-month -> MDDB2
Data storage architecture

• “Splitting” by crossing, an example
  – Sector-Year-Region-Supplier group-month-group
    -> dataset1
  – Sector-Year-Region-Supplier group-month-group-
    - - - - -article -> View of RDBMS star schema
Data storage architecture

- MDDB or dataset or view?
- Depends on:
  - cardinality
  - performance expectations
  - server availability
Data storage architecture

• Typically detail or “NWAY” crossing will be stored in a star schema
  – SAS datasets
  – SAS View of RDBMS star schema
  – Large volume

• Aggregations stored in a combination of MDDBs and SAS datasets
Server architecture

- Central Warehouse
  - DMDB
  - Multi-Dimensional
  - Archive

- Departmental Warehouses
  - Data Mart
  - Info Mart
  - Multi-Dimensional
  - Detail Store

- Personal Data and Viewers
  - PC Fat Client
  - Net PC Thin Client

Mainframe or UNIX Server

- LAN or NT Server
  - Win 95, NT or Net PC
Server architecture

• Exploit existing hardware infrastructure

• Star schema on high end server with plenty of disk space

• Aggregations on intermediate server(s)
Evolving architecture

- Iterative process
- Performance tuning
  - log dataset
    - frequently accessed crossings
    - frequently requested summarisations
    - access time
Evolving architecture

- Can create new MDDBs/datasets/views
- Remove unused MDDBs/datasets/views
- Enhance server usage
- Monitor use of system
Metadata creation

- Data model
- Architecture
  - data stored in appropriate media
  - data stored on appropriate server(s)

But how can we utilise all of this????

Metadata!!!!!
Metadata creation

• Create a “proxy” MDDB
  – 1 cell
  – only used for metadata purposes
  – all hierarchical information
  – all analysis variables

• Register this “proxy” MDDB with the EIS metabase
Metadata creation

• “Proxy” MDDB registration
  – 2 new metabase attributes
    • _dummddb - tells the metabase that this is a “proxy” MDDB
    • _assdata - allows you to define the different MDDBs/datasets/views and the associated servers defined to this HOLAP model.
Metadata creation

Demonstration
Report definition

• Define OLAP reports as normal
• Specify the “proxy” MDDB as the data source
• In the “Advanced” window change the default model to:
  – SASTOOL._DMDBB.HOLAP_M.CLASS
• Run the report!
Report definition

Demonstration
Conclusion

- HOLAP is simple to implement
- HOLAP offers scalability & performance to client-server OLAP applications
- HOLAP enables the exploitation of existing hardware & software infrastructures
- HOLAP is the answer
Questions ?