OLAP Council
APB-1 OLAP Benchmark
Release II

November 1998

Introduction

The OLAP Council has sponsored the development of an analytical processing benchmark, the APB-1. The benchmark simulates a realistic On-Line Analytical Processing (OLAP) business situation that exercises server-based software. The goal of the APB-1 is to measure a server's overall OLAP performance rather than the performance of specific tasks. To ensure the relevance of the APB-1 to actual business environments, the operations performed on the database have been carefully chosen to reflect common business operations. These operations include the following:

Bulk loading of data from internal or external data sources

Incremental loading of data from operational systems

Aggregation of input level data along hierarchies

Calculation of new data based on business models

Time series analysis

Queries with a high degree of complexity

Drill-down through hierarchies

Ad hoc queries

Multiple on-line sessions

Successful OLAP applications must provide "just-in-time information". The key to effective decision-making is having the right information at the right time. Judging a server's ability to accomplish this goal is more than simply measuring an OLAP server's processing performance. Its abilities to represent complex business relationships and to respond to changing business requirements are equally important.

While the APB-1 does not try to measure a system's ability to respond to change, the amount and clarity of programming code are used as a qualitative measure of responsiveness. The audience of the APB-1 must be able to evaluate a given solution both in terms of its quantitative and qualitative appropriateness to the task. For this reason, publication of APB-1 benchmark results must include both the database schema and all code required for executing the benchmark.

For the purposes of comparing the performance of different combinations of hardware and software, a standard benchmark metric called AQM (Analytical Queries per Minute) has been defined. Broadly stated, AQM represents the number of analytical queries processed per minute including data loading and computation time. Thus the AQM incorporates data loading performance, calculation performance and query performance into a singe metric.

The AQM metric is calculated as follows:

Total time in seconds to perform the incremental data load +

Total time in seconds to perform batch computations, if required +

Total time in seconds to execute all queries =

Total time in seconds for AQM measurements.

AQM = Total number of queries executed * 60 /

Total time in seconds for AQM measurements

The AQM rises as overall system performance improves.

It is generally agreed that OLAP applications require a multidimensional view of data. For this reason, the benchmark is specified in multidimensional terms. The terms *dimension*, *dimension member*, and

hypercube used in this document do not, however, preclude the use of any commercially available database management system.

The APB-1 is a general OLAP application and does not reflect the entire range of OLAP requirements. The benchmark should not be used to test whether a given database management product contains a full OLAP feature set. Benchmark results will vary based on the execution platform and workloads. Comparisons not based on identical runs on identical equipment are not advised. Customers should not substitute the APB-1 for application benchmarking where specific product features or operational characteristics are required.

Application Environment

The APB-1 contains a set of business operations that exercise basic functionality essential to OLAP applications. The benchmark database balances the need to approximate a real world business application with the need to provide a performance benchmark that can be executed without extreme effort. To this end, the benchmark reduces the diversity of operations while retaining fundamental functionality.

The OLAP application used in the benchmark is a sales and marketing analysis system. The benchmark is a synthesis of general business practices, not a model of a specific industry or market. The database contains the information required by a supplier to analyze product sales to customers through distribution channels over time. Units sold, dollar sales, costs, and margins are tracked by actual, budget, and forecast scenarios.

The database design has no structural requirements. The varied nature of database technologies (multidimensional and relational) and the lack of generally accepted design criteria (denormalization is the rule) would make any structural requirements prejudicial in nature. Specifically, storing calculated values is neither strictly forbidden nor encouraged. Whether the calculated values are preprocessed and stored in the database or are calculated at query time, the time to calculate values is included in the computation of the AQM.

OLAP applications tend to be updated incrementally at a specified time interval (i.e., weekly or monthly). Therefore, although database setup and loading of historical actuals is required in order for the benchmark to perform queries, these operations are not included in the calculation of AQM. Loading current month actuals and loading current year budget are included in the calculation of AQM. Calculated values stored in the database are treated the same way, i.e., if a calculated value depends only on historical actuals, it is not included in the AQM. If, however, a calculated value depends on current month actuals, current year budget, or a mix of historical and current actuals/budget, then the calculation is included in the AQM.

Database Structure

The logical database structure is made up of six dimensions: time, scenario, measure, and three aggregation dimensions that define the database size (product, customer, and channel). The APB1GEN program (described later) uses an input parameter to determine the number of members in each of these dimensions. The minimum number of members required in each dimension and the relationships between the dimensions are described below.

Product

Of the three aggregation dimensions, the product dimension has the most members. The number of members in the product dimension is ten times the number of members in the customer dimension. The minimum number of members in the product dimension is 10,000. The product hierarchy is a steep hierarchy containing seven levels. Each member of the hierarchy contains at most one parent. Every member of the hierarchy, except the member at the top level, has a parent. The bottom level of the product hierarchy contains 90% of the members. The top level of the product hierarchy contains a single member (the grand total). The other members of the hierarchy are distributed among the five remaining levels.

The names of the levels in the product dimension hierarchy are:

Top

Division

Line

Family

Group

Class

Code

Customer

The number of members in the customer dimension is 100 times the number of members in the channel dimension. The minimum number of members in the customer dimension is 1,000. The customer

hierarchy is a flat hierarchy containing three levels. The bottom level of the customer hierarchy contains 90% of the members. The top level of the customer hierarchy contains a single member (the grand total). The remaining members belong to the middle level of the hierarchy. Each member of the hierarchy contains at most one parent. Every member of the hierarchy, except the top-level member, has a parent.

The names of the levels in the customer dimension hierarchy are:

Top

Retailer

Store

Channel

Of the three aggregation dimensions, the channel dimension has the fewest members. The number of members in the channel dimension is an input parameter to the APB1GEN program. The minimum number of members in the channel dimension is ten. The channel hierarchy contains two levels. The bottom level of the channel hierarchy contains all but one of the members. The top level of the channel hierarchy contains a single member (the grand total). Each member of the hierarchy contains at most one (1) parent. Every member of the hierarchy, except the top-level member, has a parent.

The names of the levels in the channel dimension hierarchy are:

Top

Base

Time

The time dimension is made up of two years (1995 and 1996) of monthly members. A Julian calendar (January to December) is used. The time hierarchy includes quarterly, yearly, and year-to-date aggregations. The current month used in the benchmark processing is June 1996. Periods before this are considered historical periods. Periods after this are considered future periods.

The most frequently used aggregation along the time dimension is summation. The exceptions to this are:

Inventory - Uses an ending balance calculation

Cost factor measures - Use an average balance calculation

Percentage measures - Cannot be aggregated

The ending balance calculation computes the parent value as the last period of its children. For example, the ending balance of Quarter 1 is March. An average balance calculation computes the parent value as the sum of its children's values divided by the number of children. For example, the average balance of Quarter 1 is the sum of January, February, and March divided by 3. The percentage measures cannot be aggregated along the time dimension, but rather must be computed from the aggregations of their components. For example, percent variance for Quarter 1 cannot be calculated from the percent variances of January, February, and March. The values must be calculated from the Quarter 1 values of actuals and budget.

The queries reference a set of time dimension calculations. The current periods (June 1996, Quarter 2 1996, Year 1996, and YTD 1996) are compared with the previous period and the same period last year. This comparison is stated as both a value difference and a percent change.

Scenario

There are three base values in the scenario dimension: two are input from data files and one is modeled from the other two. The input scenarios contain actuals and budget and the modeled scenario contains forecast. The scenario members are valid for differing time frames. The actual scenario contains data from January 1995 to June 1996. The budget and forecast scenarios contain data from January 1996 to December 1996.

The forecast scenario is modeled as a function of both the actual and budget scenarios. The computation of the forecast scenario is:

> Calculate total 1996 forecast values by the retailer level of the customer hierarchy as the annualized values of the second half actuals of 1995 increased by 15%. Allocate to the months in 1996 based on the 1996 budget. Allocate to the stores from the retailer based on the year-to-date actuals. Allocate to products based on year-to-date budget.

The queries also reference a set of scenario dimension calculations. Variance includes both value and percent differences.

Budget vs. Actuals

Forecast vs. Actuals

Measures

There are ten measures, five input and five calculated. They are:

Input

Units Sold varies by product, customer, channel, time, and scenario

Dollar Sales varies by product, customer, channel, time and scenario

Inventory varies by product, customer and time

Product Cost varies by product, time and scenario

Shipping Cost varies by customer, time and scenario

Calculated

Average Price = Dollar Sales / Units Sold

Cost = Units Sold * (Product Cost + Shipping Cost)

Margin = Dollar Sales - Cost

Margin Percent = Margin / Dollar Sales

Smoothed Sales = 6 month moving average of dollar sales

Data Files

There are two sets of data files. The first set of data files is required to initialize the database. The second set of data files is required for the incremental processing. The APB1GEN program generates all files and presents them as fixed field length ASCII.

To allow for the greatest diversity in database designs, the files containing the product, customer, channel members, and hierarchy are presented in both tree and level formats. The tree format presents the dimension hierarchy as a record for each member of the hierarchy that has a parent. The record contains fields for the child member, the parent member, and the name of the level to which the child member belongs. The level format presents the dimension hierarchy as a record for each dimension member at the bottom level of the hierarchy. The record contains a field for each level in the hierarchy.

Hierarchy data files are provided for the product, customer, and channel dimensions. The scenario and measure dimensions do not have hierarchies and therefore do not have hierarchy data files.

The data files reference scenario members by the names ACTUAL, BUDGET, and FORECAST.

The data files do not explicitly reference members of the measure dimension. The naming of the measure dimension members must be exactly as described in the measures section above.

The data files reference members of the time dimension at the month level. The names of the members in the data files are dictated by the benchmark and are in the format of YYMMDD (950301 will be the first day of March in the year 1995). Benchmark implementors can choose to use the member names or they can translate the member names to another format during the data load. It is important to remember that any processing that occurs during the incremental load is included in the computation of the AQM performance metric.

The data files will not be in any sorted order. To do so would prejudice the benchmark and favor certain database designs. Sorting of the data before loading can be performed, but the sorting time will be included in the calculation of the AQM.

Initialization Data Files

Product Hierarchy - Level Format (PRODHIER.APB)

Column Width Field

1 12 Code

13 12 Class

25 12 Group

37 12 Family

49 12 Line

61 12 Division

Product Hierarchy - Tree Format (PRODTREE. APB)

Column Width Field

1 12 Member

13 12 Parent

25 12 Level

Customer Hierarchy - Level Format (CUSTHIER.APB)

Column Width Field

1 12 Store

13 12 Retailer

Customer Hierarchy - Tree Format (CUSTTREE.APB)

Column Width Field

1 12 Member

13 12 Parent

25 12 Level

Channel Hierarchy - Level Format (CHANHIER.APB)

Column Width Field

1 12 Base

Channel Hierarchy - Tree Format (CHANTREE.APB)

Column Width Field

1 12 Member

25 12 Level

Time Hierarchy - Level Format (TIMEHIER.APB)

Column Width Field

1 6 Mon

7 6 Qtr

13 4 Yr

Time Hierarchy - Tree Format (TIMETREE.APB)

Column Width Field

1 6 Member

7 6 Parent

13 7 Level

Product Cost (PRODCOS.APB)

Column Width Field

1 12 Product - Code Level

13 12 Scenario - Actual, Budget

25 6 Time - January 1995 through December 1996

31 10 Value - 999999.999

Shipping Cost (CUSTSHIP.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Scenario - Actual, Budget

25 6 Time - January 1995 through December 1996

31 10 Value - 999999.999

Historical Sales (HISTSALE.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Product - Code Level

25 12 Channel - Base Level

37 6 Time - January 1995 through May 1996

43 10 Unit Sales - 9999999999

53 10 Dollar Sales - 9999999.99

Historical Inventory (HISTINV.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Product - Code Level

25 10 Inventory 199501

35 10 Inventory 199502

45 10 Inventory 199503

55 10 Inventory 199504

65 10 Inventory 199505

75 10 Inventory 199506

85 10 Inventory 199507

95 10 Inventory 199508

105 10 Inventory 199509

115 10 Inventory 199510

125 10 Inventory 199511

135 10 Inventory 199512

145 10 Inventory 199601

155 10 Inventory 199602

165 10 Inventory 199603

175 10 Inventory 199604

185 10 Inventory 199605

Incremental Load

Current Month Sales (CURRSALE.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Product - Code Level

25 12 Channel - Base Level

37 10 Unit Sales - 9999999999

47 10 Dollar Sales - 9999999.99

Current Month Inventory (CURRINV.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Product - Code Level

25 10 Inventory - 9999999999

Budget (BUDGET.APB)

Column Width Field

1 12 Customer - Store Level

13 12 Product - Code Level

25 6 Time - January 1996 through December 1996

31 10 Unit Sales - 9999999999

41 10 Dollar Sales - 9999999.99

Query Execution (QRYxxx.APB)

Column Width Field

1 24 Query number

25 24 Parameter 1

49 24 Parameter 2

73 24 Parameter 3

97 24 Parameter 4

121 24 Parameter 5

145 24 Parameter 6

Queries

OLAP queries are ad hoc and very dynamic. The types of information requested span the full scope of the available data. Queries must be able to take advantage of the business relationships represented in the database. The time periods, products, customers, and channels must be dynamically generated from their respective hierarchies. For example, a query requesting the sales for a predetermined list of stores is the exception rather than the rule in analytical processing. The more common type of query asks for the sales of all the stores of a given retailer.

Time is a unique dimension in any application and as such, has special characteristics. Hierarchical relationships (the months in a given quarter) are not the only way to request time periods. Relative positioning is just as important. The most often used of the relative positioning functions is the time period immediately preceding the requested one and the same time period a year ago.

The queries vary in how they treat null values. The query definition states for each dimension whether to suppress members where all values are null. This means that the server should not generate a requested dimension member if every value it returns is null.

To best simulate a realistic operating environment, each of the queries will have substitution parameters. These parameters will not only state the dimension members, but also the function to apply to them. Note that the computation of the members lists for the returned data set must be performed on the server and not on the client.

The benchmark queries are detailed below. The definition of each query will contain the following information:

Query description

Parameter definition

Functional query definition

Query 1 - Channel Sales Analysis

This query shows actual units sold, dollar sales, and Average Price for a given channel. The product, customer, channel, and time period vary with each execution of the query. Only the instances of product and customer that contain data should be returned.

parameter 1 = ?product

parameter 2 = ?customer

parameter 3 = ?channel

parameter 4 = ?time

get UNITS SOLD, DOLLAR SALES, AVERAGE PRICE

by SCENARIO = "ACTUAL"

by PRODUCT = <children(?product)> option suppress null

by CUSTOMER = <children(?customer)> option suppress null

by CHANNEL = <?channel>

by TIME =<children(?time)>

Query 2 - Customer Margin Analysis

This query shows actual sales, cost and margin for a given customer for the sum of all channels for a requested period. The product, customer and time period vary with each execution of the query. Only the instances of product that contain data should be returned.

parameter 1 = ?product

parameter 2 = ?customer

parameter 3 = ?time

get UNITS SOLD, DOLLAR SALES, COST, MARGIN, MARGIN PERCENT,

PRODUCT COST, SHIPPING COST

by SCENARIO = "ACTUAL"

by PRODUCT = <children(?product)> option suppress null

by CUSTOMER = <?customer>

by CHANNEL = attribute("level", "TOP")

by TIME = <?time>

Query 3 - Product Inventory Analysis

This query shows actual sales, cost and inventory for a given product regardless of channel. The product and customer vary with each execution of the query. All months from January 1995 through June 1996 should be returned.

parameter 1 = ?product

parameter 2 = ?customer

get UNITS SOLD, DOLLAR SALES, COST, INVENTORY

by SCENARIO = "ACTUAL"

by PRODUCT = <?product>

by CUSTOMER = <children(?customer)>

by CHANNEL = attribute("level", "TOP")

by TIME = "9501" through "9606"

Query 4 - Time Series Analysis

This query shows actual sales and a 6-month moving average of smoothed sales for a given customer and a group of time periods over all channels. The product, customer, and time period vary with each execution of the query.

parameter 1 = ?product

parameter 2 = ?customer

parameter 3 = ?time

get DOLLAR SALES, SMOOTHED SALES

by SCENARIO = "ACTUAL"

by PRODUCT = <children(?product)>

by CUSTOMER = <?customer>

by CHANNEL <all members including total channel>

by TIME = <children(?time)>

Query 5 - Customer Budget

This query shows a customer's budget for all twelve months in 1996. The customer varies with each execution of the query.

parameter 1 = ?customer

get UNITS SOLD, DOLLAR SALES, AVERAGE PRICE, COST, MARGIN

by SCENARIO = "BUDGET"

by PRODUCT = attribute("level", "TOP")

by CUSTOMER = <?customer>

by TIME = "9601" through "9612"

by CHANNEL = attribute("level", "TOP")

Query 6 - Product Budget

This query shows a product's budget for all quarters in 1996. The product varies with each execution of the query.

parameter 1 = ?product

        get UNITS SOLD, DOLLAR SALES, AVERAGE PRICE, COST, MARGIN

by SCENARIO = "BUDGET"

by PRODUCT = <?product>

by CUSTOMER = attribute("level", "TOP")

by TIME = children("1996")

Query 7 - Forecast Analysis

This query shows a channel's forecast for a given time period in 1996. The product, customer, and time period vary with each execution of the query.

parameter 1 = ?product

parameter 2 = ?customer

parameter 3 = ?time

        get UNITS SOLD, DOLLAR SALES, AVERAGE PRICE, COST, MARGIN

by SCENARIO = "FORECAST"

by PRODUCT = <children(?product)> option suppress null

by CUSTOMER = <children(?customer)> option suppress null

by TIME = <?time>

Query 8 - Budget Performance

This query shows budget vs. actual and this year vs. last year's performance for the current month and current year-to-date. The product and customer vary with each execution of the query.

parameter 1 = ?product

parameter 2 = ?customer

get DOLLAR SALES[SCENARIO:"ACTUAL"]

DOLLAR SALES[SCENARIO:"BUDGET"]

val_diff(DOLLAR SALES, SCENARIO, "ACTUAL", "BUDGET")

pct_diff(DOLLAR SALES, SCENARIO , "ACTUAL", "BUDGET")

last_year_val_dif(DOLLAR SALES[SCENARIO:"ACTUAL"])

last_year_pct_dif(DOLLAR SALES[SCENARIO:"ACTUAL"])

by PRODUCT = <?product>

by CUSTOMER = <?customer>

by TIME = {"9606", YTD("9606")}

Query 9 - Forecast Performance

This query shows forecast vs. actual and this month's vs. last month's performance for the current month and current year-to-date. The product and customer vary with each execution of the query.

parameter 1 = ?product

parameter 2 = ?customer

get DOLLAR SALES[SCENARIO:"ACTUAL"]

DOLLAR SALES[SCENARIO:"FORECAST"]

val_diff(DOLLAR SALES, SCENARIO, "ACTUAL", "FORECAST")

pct_diff(DOLLAR SALES, SCENARIO , "ACTUAL", "FORECAST")

last_period_val_dif(DOLLAR SALES[SCENARIO:"ACTUAL"])

last_period_pct_dif(DOLLAR SALES[SCENARIO:"ACTUAL"])

by PRODUCT = <?product>

by CUSTOMER = <?customer>

by TIME = {"9606", YTD ("9606")}

Query 10 - Ad Hoc

This query answers ad hoc questions of the database. The measure, scenario, product, customer, channel and time period vary with each execution of the query.

parameter 1 = ?product

parameter 2 = ?customer

parameter 3 = ?channel

parameter 4 = ?time

parameter 5 = ?scenario

parameter 6 = ?measure

get <?measure>

by SCENARIO = <?scenario>

by PRODUCT = <?product>

by CUSTOMER = <?customer>

by CHANNEL = <?channel>

by TIME =<?time>

Query Distribution

The following chart shows the distribution of queries over the course of an APB-1 benchmark execution.

10% Query 1 - Channel Sales Analysis

10% Query 2 - Customer Margin Analysis

15% Query 3 - Product Inventory Analysis

3% Query 4 - Time Series Analysis

5% Query 5 - Customer Budget

5% Query 6 - Product Budget

15% Query 7 - Forecast Analysis

20% Query 8 - Budget Performance

15% Query 9 - Forecast Performance

2% Query 10 - Ad Hoc

Implementation Rules

The software used to implement the APB-1 benchmark must be generally available. The use of software that is engineered specifically for the benchmark is not acceptable. The use of alpha, beta, limited release or any other non full-production or general availability software is not acceptable. The software must be orderable, receivable and installable by customers, in full general availability / production status.

The APB-1 benchmark has the following two database design restrictions: All data must be stored on the server and all calculations must be performed on the server. Each benchmark implementor may decide how much and which derived data to pre-calculate.

The APB-1 is executed in six steps:

1 - Execution of the APB.EXE program to produce hierarchy files and historical data

2 - Database initialization and historical data load and optional pre-calculation

3 - Execution of the APB.EXE program to produce incremental data files

4 - Incremental data load and optional pre-calculation

5 - Execution of the APB.EXE program to produce query data files

6 - Query execution

The APB.EXE program is executed between each stage of the benchmark process to prevent an implementor from "wiring" the benchmark. This processing sequence hides specific knowledge about the next benchmark step until it is needed. This is done to ensure that the database design is application specific not data specific. It is important that the calculation of the AQM metric apply to general application processing and not to a particular benchmark execution.

The purpose of the APB-1 is to measure the performance of database servers for OLAP applications. All processing must be done on the server. The client in this benchmark exists only to provide mechanisms for supplying parameter substitutions, to pass the query to the server, and to write the returned data to a disk file for audit. The computation of dimension member sets and derived data are specifically prohibited from being performed on the client.

The AQM is computed as the total number of queries executed times 60 divided by the execution time in seconds to perform steps 4 and 6 above. The execution time for step 4 begins when the incremental data

files start being processed. If the benchmark implementor prefers the data files in a sorted order and performs that sort, the sort time is included in the execution time. The execution time for step 6 begins when the first user initiates the first query and ends when the last user completes the last query. The results of each query must be written to disk on the client platform to be available for auditing.

The APB.EXE Program

The APB.EXE program is used to generate data and query information for the APB-1 benchmark. The program should be run with each benchmark execution. This ensures that the implementation of the APB-1 is not designed for a specific data and/or query set.

The APB.EXE program controls the size of the domain of each of the three aggregation dimensions (product, customer, and channel). When the APB.EXE program is used to initialize the benchmark database, the number of members for the channel dimension is given as a parameter. The channel dimension must have at least 10 members. The customer dimension is generated with 100 times the number of members in the channel dimension. The product dimension is generated with 1,000 times the number of members in the channel dimension.

The number of records in the historical sales data file is controlled by the density parameter of the APB. EXE program. This parameter, given as percentage, determines how many of the total possible of channel, customer, and product combinations at the lowest level in the hierarchy will have data for each of the 17 historical months. Remember that 10% of the members of each of these dimensions are aggregates.

records = (length(CHANNEL) * .9) *

(length(CUSTOMER) * .9) *

(length(PRODUCT) * .9) * 17 * density%

The number of records in the historical inventory data file will be approximately three times greater than the number of records in the sales file. This is because although not every product is sold in every month, to sell a product there must be inventory. Once all of a product is sold, there is no more inventory. The APB.EXE program manages these operational characteristics.

The current month sales and inventory data files will be 1/17th the size of the actual sales and inventory files because the two files contain data only for one month.

The final function of the APB.EXE program is to generate a file for each query stream in the benchmark

execution. A single query stream is equivalent to a user but with no think time, so each query stream represents the workload of a large number of real-world concurrent users. The number of queries generated for each query stream is 250 times the number of members in the channel dimension. The total number of queries across all query streams in the benchmark execution is 250 times the number of members in the channel dimension times the number of query streams. The minimum number of query streams allowed is 10, the maximum number of query streams allowed is 10,000

Full Disclosure and Auditing

Full disclosure is required by anyone publishing APB-1 benchmark results so that others can duplicate the results using the same documentation, settings, products and environment. The disclosure includes the information an evaluator needs to judge a server's processing performance and its suitability to the task of analytical processing.

Disclosure items:

- Audit report
- Database schema
- Any code, scripts, programs and / or interactive user processes used to create the database and structures
- Any code, scripts, programs and / or interactive user processes used for the data loading
- Any code, scripts, programs and / or interactive user processes used for any pre-calculations
- Any code, scripts, programs and / or interactive user processes used for the execution of each query
- Any code, scripts, programs and / or interactive user processes used on the client workstations
- Any code, scripts, programs and / or interactive user processes of any type used to implement the benchmark
- Interactive user processes may be documented via screen shots
- Total number of users simulated
- Number of users simulated on each client workstation
- Size of each of the benchmark input data files generated by APB.EXE
- All settings specified in APB.EXE
- Final database size, including all data and indices
- Any other storage required. Examples: temp space, sort space, rollback space, etc.
- Time to perform historical data load and calculation (from step 2 of benchmark)
- Time to perform incremental load and calculation (from step 4 of benchmark)
- Total number of queries processed by server
- Total time to process all queries
- AQM metric
- OLAP server software including specific version and any point release
- OLAP server tuning parameters
- Any additional server software used and tuning parameters

- Server hardware platform, software and configuration (processor, memory, disk)
- Client hardware platform(s), software and configuration (processor, memory, disk)
- Network hardware and software and configuration
- Operating system name, version, and configuration parameters

An auditor certified by the OLAP Council must perform an independent audit of the APB-1 benchmark results. The auditor cannot have any financial gain from the benchmark other than fees directly related to the audit. The auditor cannot be involved, in any manner, with the execution of the benchmark, other than to ensure its proper execution.

The audit of the APB-1 benchmark will consist of the following six tasks:

1 - Verifying that the benchmark was run in the prescribed order as detailed in the implementation rules section.

2 - Examining the programming code on both the client and the server to verify that:

- The client's sole role is to pass the queries to and receive the results from the server
- The server performs all calculations

3 - Verifying that the results returned are correctly calculated. Thirty queries (three of each query type) must be randomly chosen and must be audited to:

- Ensure data accuracy
- Determine that the dimensions members returned are correct

4 - Ascertaining that all disclosure items are correctly stated in the disclosure document.

5 – Verifying that the benchmark implementation conforms completely to the OLAP council APB-1 benchmark specification, addenda and errata. Auditor must also specify which revision of the APB-1 benchmark specification and which version of the APB.EXE program was used.

6 - Producing a report detailing the audit steps and results.

The OLAP Council sponsored development of the APB-1 benchmark, which was first released in April 1996 for public comment. The APB-1 benchmark Release II was released in November 1998.

Addenda and Errata to the APB-1 Benchmark:

0. **Auditors and auditor certification requirements**

   The OLAP council must certify all auditors. Organizations with individuals wishing to become certified APB-1 benchmark auditors may contact the OLAP council via electronic mail to admin@olapcouncil.org

   0. OLAP Council certification process

      The OLAP council certifies individuals, not organizations, to be auditors of the APB-1 benchmark. To become certified, the individual must submit documentation of their experience in implementing OLAP systems to the OLAP council. Auditors may not be affiliated with any vendor of OLAP software. After reviewing the individual's credentials, the OLAP council will vote as to whether to certify the individual as an official auditor of the APB-1 benchmark. The OLAP council may vote to decertify any individual at any time.

      If an organization wishes to have multiple individuals certified as auditors of the APB-1 benchmark, each individual must be certified separately.

   1. Current certified auditors (last updated 11/98)

   George Spofford voice: (617) 864-6516

   Dimensional Systems fax: (617) 492-3508

   22 Lowell St. email: Spofford@dimsys.com

   Cambridge, MA 02138 web: http://www.dimsys.com

0. **Clarifications to forecast calculations**

   The forecast logic in the "Scenario" section of the benchmark states: " Calculate total 1996 forecast values by the retailer level of the customer hierarchy as the annualized values of the second half actuals of 1995 increased by 15%. Allocate to the months in 1996 based on the 1996 budget. Allocate to the stores from the retailer based on the year-to-date actuals. Allocate to products based on year-to-date budget."

1. Measures to be calculated using the forecasting calculation.

    Unit Sales, Dollar Sales and Cost are to be calculated using the forecasting calculation.

2. Measures to be derived from the forecast measures

    Smoothed Sales, Margin, Margin Percent, and Average Price are to be derived from the forecast measures specified in section 2.1.0

3. Diagram and Narrative of Forecast Calculations

Diagram courtesy of George Spofford, Dimensional Systems

The forecast scenario is modeled as a function of both the actual and budget scenarios. The computation of the forecast scenario is performed in 4 logical steps. :

Step 1: Calculate total 1996 forecast values for the retailer level of the customer hierarchy and the top level of the product hierarchy as the annualized values of the second half actuals of 1995 increased by 15%.

Step 2: Allocate total 1996 forecast values for the retailer level of the customer hierarchy and the top level of the product hierarchy to the months in 1996 based on the 1996 budget.

Step 3: Allocate to the store level from the retailer based on the year-to-date actuals.

Step 4: Allocate to the code level products from the top level based on year-to-date budget.

0. **Year to date calculations**

    The "current period" as specified by the benchmark is June 1996. The benchmark refers to "year to date actual" and "year to date forecast." Both of these refer to "Year to June 1996."

0. **Clarification of calculations described in the queries**
1. Ordering of value differences

    Queries 8 and 9 specify that the query returns "val_diff" of Dollar Sales for two scenarios.

The queries are intended to show actual Dollar Sales performance vs. Budget or Forecast. If actual sales are less than budgeted or forecasted sales, this is generally considered a bad thing and should thus result in a negative value for the difference.

Therefore, val_diff is calculated as follows:

For query 8, val_diff = Actual - Budget

For query 9, val_diff = Actual - Forecast

2. Ordering of percent value differences

    Queries 8 and 9 specify that the query returns "pct_diff" of Dollar Sales for two scenarios. The queries are intended to show actual Dollar Sales performance as a percentage of Budget or Forecast.

    Pct_diff is calculated as follows:

    For query 8, pct_diff = (Actual - Budget) / Budget

    For query 8, pct_diff = (Actual - Forecast) / Forecast

3. Order of computation of difference calculations

    In queries 8 and 9, the "val_diff", "pct_diff", "last_year_val_dif", and "last_year_pct_dif" calculations are to be performed last. In other words, they are to be computed from other calculated / aggregated members.

0. **Clarifications to aggregation rules**

    The "Time" section of the benchmark states:

    "The most frequently used aggregation along the time dimension is summation. The exceptions to this are:

    Inventory - Uses an ending balance calculation

    Cost factor measures - Use an average balance calculation

    Percentage measures - Cannot be aggregated

The ending balance calculation computes the parent value as the last period of its children. For example, the ending balance of Quarter 1 is March. An average balance calculation computes the parent value as the sum of its children's values divided by the number of children. For example, the average balance of Quarter 1 is the sum of January, February, and March divided by 3. The percentage measures cannot be aggregated along the time dimension, but rather must be computed from the aggregations of their components. For example, percent variance for Quarter 1 cannot be calculated from the percent variances of January, February, and March. The values must be calculated from the Quarter 1 values of actuals and budget."

1. Definition of "Cost factor measures"

   "Cost factor measures" are Product Cost and Shipping Cost

2. Definition of "Percentage measures"

   "Percentage measures" refers to Margin Percent.

3. Calculation order for Margin Percent

   Margin percent is to be calculated after aggregation of its constituent measures. Margin Percent defined as Margin / Dollar Sales. To calculate Margin Percent, first calculate Margin and Dollar Sales for the requested time period. Then compute Margin Percent last.

   Example: Margin Percent for Quarter 1 = Margin for Quarter 1 / Dollar Sales for Quarter 1.

   The same calculation ordering rules that apply to Margin Percent also apply to the percentage measures described by the queries, which are "pct_diff" and "last_year_pct_dif."

   In this example, it would be incorrect to calculation Margin Percent for each month in Quarter 1, and then add these three Margin Percents together. This is what is meant by "Percentage measures - Cannot be aggregated" in section 5.0.0.

4. Calculation order for Cost

   Cost is to be calculated at the leaf cells, and then aggregated.

   5.4.0 Calculation order for Average price

Average Price is to be calculated after aggregation of its constituent measures.

5. Calculation order for Smoothed Sales
   Smoothed Sales is to be calculated after aggregation of its constituent measures

6. Calculation for Smoothed Sales at non-leaf products, customers and channels
   Calculated Smooth sales values should be available for all levels of products, customers and channels. To calculate Smoothed Sales, first perform the aggregation of the constituent measures up the product, customer and channel dimensions as applicable, and then resolve the Smoothed Sales equation using the aggregated data.

7. Calculation for Smooth Sales at non-leaf time periods

   Value for Smoothed Sales only exist for leaf level time periods (Month level periods). For all other members of the Time dimension, the value for Smoothed Sales is Null.

8. Calculation order for Time averaging

   Time based calculations that require averaging are to be performed after aggregation of constituent components. Averaging is to be performed after aggregation.

0. **Clarification of the data to be returned by the queries**
1. Queries that specify suppression of null values

   Queries 1, 2, and 7 specify that null values are to be suppressed in particular dimensions. Due to the sparse nature of the APB-1 data set, some instances of these queries will return no data. For example, in query 2 the benchmark generator may specify a combination of Time, Customer, and Product for which no data exists. Since query 2 also specifies that missing members in the product dimension are to be suppressed, the entire result set is therefore suppressed.

   The acceptable behavior in this case is to return to the client an empty result set, or to return an error message indicating that the query returns no data. Other behaviors are not acceptable.

2. Layout of data to be returned by the queries

   The APB-1 benchmark does not impose a particular layout of the result set to be returned by the queries. The implementor may choose the layout of the result set for each query.

3. Queries that return missing/sparse data

   In queries 3, 4, 5, 6, 8, 9, and 10, the page, row, and column dimensions of the result set must be fully populated will all members specified by the query regardless of the sparsity

or density of the data cells returned.

Some queries in the benchmark will return entirely missing/sparse data values. The required behavior in this case is to return the results set with the page, row, and column dimensions fully populated will all members as specified by the query, and to return null value indicators as described in section 6.4.0 within the body of the result set.

6.4.0 Null value indicators

The APB-1 data set is highly sparse. Many data cells specified in particular queries may contain missing/sparse data. All data cells in the result set should contain either a numeric value, or an indicator that the data cell is missing/sparse. The implementor must document the treatment of null values for query, reporting, and calculation purposes as part of the full disclosure.

5. Labels for calculated members in Query 8
   The labels for the calculated members val_diff, pct_diff, last_year_val_dif, and last_year_pct_dif specified in Query 8 must be returned exactly as follows:
   "Act vs Bud", "Act pct Bud", "Act vs Last Yr", and "Act pct Last Yr"
6. Labels for calculated members in Query 9

   The labels for the calculated members val_diff, pct_diff, last_period_val_dif, and last_period_pct_dif specified in Query 9 must be returned exactly as follows:

   "Act vs Fcst", "Act pct Fcst", "Act vs Last Pd", and "Act pct Last Pd"

7.0.0 Function of the client query programs

7.1.0 The client query programs can perform only the following functions:

- Sequence queries for execution by the query streams;
- Read the files specifying which queries to run and the substitution parameters for these queries generated for them by the APB.EXE program;
- Generate the executable query text for each query;
- Complete the executable query text by replacing the substitution parameters by the values generated for them by the APB.EXE program;
- Direct the executable query to the appropriate data structures on the system under test;
- Submit each complete executable query text to the system under test for execution;
- Receive the output data resulting from each query execution from the system under test;
- Perform simple column and row formatting for ouput readability such as aligning fields in a matrix report (no label insertion/translation or data manipulation or any other formatting is allowed);

❍ Measure the execution times of the queries and compute measurement statistics;
❍ Maintain an audit log of query text and query execution output;

The driver shall not perform any function other than those described above.

7.2.0 Specifically, the client query programs shall not perform any of the following functions:

❍ Any sorting or translation of the query output data;
❍ Any calculation of the query output data;
❍ Any null suppression of the query output data;
❍ Any other filtering or other client-side processing of the query output data;
❍ Execute the queries in any other order than that generated by the APB.EXE program;
❍ Submitting to the system under test the values generated for the substitution parameters of a query other than as part of the executable query text submitted;
❍ Submitting to the system under test any data other than the compliant executable queries;
❍ Artificially extending the execution time of any query

0. **Clarifications to server-based processing requirements**

The APB-1 benchmark definition states the following regarding server based processing requirements:

"The benchmark simulates a realistic On-Line Analytical Processing (OLAP) business situation that exercises server-based software."

"The purpose of the APB-1 is to measure the performance of database servers for OLAP applications. All processing must be done on the server. The client in this benchmark exists only to provide mechanisms for supplying parameter substitutions, to pass the query to the server, and to write the returned data to a disk file for audit. The computation of dimension member sets and derived data are specifically prohibited from being performed on the client."

1. Prohibition of client-based processing

The OLAP benchmark is clear in specifying a requirement for server-based processing. The only processing allowed on the client is specified in section 6.1.0.

2. Clarification "Computation of dimension member sets"

"Computation of dimension member sets" refers to specification of the members to query and to calculate. Many of the queries, for example, specify that the query is to be executed

for the children of a particular member. The server, not the client, must generate the specific list of child members.

For example, query 3 specifies that it is to be executed with the children of a member of the customer dimension. In this example, the client may not perform the processing to specify the individual children of the member for the executable query text. Instead the server must perform the processing required to identify the children of the member while executing the query. The executable query text should contain only the parent member and whatever other text or command is required to tell the server that the query has specified that the children of that particular parent are to be used.

A two-step process is also prohibited – the client may not submit a command to the server that returns the specific children of the member to the client, which the client then uses to generate an executable query text specifying individual children.

3. Clarification "Computation of derived data"

Computation of derived data refers to any calculation of any type. All calculation and all query processing must be performed by the server and is explicitly prohibited from being performed on the client. The only processing allowed on the client is specified in section 6.1.0.

0. **Fair Metric Comparisons**

Comparisons of APB-1 OLAP Benchmark results measured against databases of different sizes, sparsity settings or query streams are believed to be misleading because the AQM metric does not scale up proportionally with an increases in database size, sparsity or numbers of query streams.

If results measured against different database sizes, sparsity settings, or numbers of query streams appear in a printed or electronic communication, then each reference to a result or metric must clearly indicate the database size against which it was obtained, the sparsity settings, and the number of query streams. This applies to metrics quoted in text or tables as well as those used to annotate charts or graphs. If metrics are presented in graphical form, then the test database size on which metric is based must be immediately discernible either by appropriate axis labeling or data point labeling.

In addition, the results must be accompanied by a disclaimer stating:

"The OLAP Council believes that comparisons of APB-1 OLAP Benchmark results

measured against different database sizes, sparsity settings or query streams are misleading and discourages such comparisons."

9.1.0 Required Reporting Components

To be compliant with the APB-1 OLAP Benchmark disclosure standards and the OLAP Council's fair use policies, all public references and printed or electronic communication to any APB-1 Release II result must include the following components:

○ The number of members in the product dimension
○ The benchmark sparsity setting
○ The AQM rating
○ The name and version number of the OLAP Server software
○ The date of the audited results
○ The name, model, number and speed of CPU's, and amount of RAM of the server hardware.
○ The name, model, number and speed of CPU's, amount of RAM, and number of clients for the client hardware.

Following is an example of compliant reporting of an APB-1 result:

NumberBuddy version 1.0 achieved an AQM of 1,013 on January 10, 1998 with 10,000 products, 5% data density and 100 query streams, running on a HAL/9000 with 4-100GHz CPUs and 100 Gigs of RAM coupled with 4 TRS-80 clients each with a 100 MHz CPU and 64 MB of RAM.

Vendors may "feature" their results in printed or electronic communication as they see fit as long as the required reporting components are disclosed and factual. Any comparison of different APB-1 Release II results must be accompanied by all required reporting components for each result.

0. **Transition to APB-1 Benchmark Release II**

The APB-1 Benchmark Release II rules and specifications will go into effect as soon as the APB-1 Benchmark specification Release II (this document) and the APB.EXE program Release II are posted on the OLAP Council web site. Until both of these items are posted on the OLAP Council web site the rules and specifications of the original APB-1 Benchmark specification are in effect. Once the APB-1 Benchmark Release II goes into effect, no new results for the original APB-1 Benchmark may be audited, generated or released.

Comparisons of results of the original APB-1 OLAP Benchmark and the APB-1 OLAP

Benchmark Release II are believed by the OLAP Council to be misleading and are not allowed in any public, printed, or electronic communication.

0. **Questions about and additional clarifications to the APB-1 benchmark**

Parties with questions about the APB-1 benchmark or requesting clarifications to the APB-1 benchmark may send electronic mail to admin@olapcouncil.org.

The OLAP council administrator, a vendor-neutral third party, monitors this electronic mail address. The administrator will answer questions directly if possible.

For those questions that cannot be answered by the OLAP council administrator and therefore require submission to the OLAP council APB-1 benchmark subcommittee, the administrator will ensure the anonymity of the submitting party to avoid any breach of confidentiality. The reason for this procedure is to prevent the members of the OLAP council benchmark subcommittee from knowing what specific vendor is working on the benchmark.

The APB-1 benchmark subcommittee has the final right of interpretation of the ABP-1 benchmark. The benchmark subcommittee will respond to the electronic mail submission directly, and, if warranted, may publish the question and the response in the APB-1 benchmark addenda and errata.

11.1.0 Constituency of the APB-1 benchmark subcommittee (last updated 11/98)

All certified auditors and all members of the OLAP council are eligible to serve on the APB-1 benchmark subcommittee. Voting members of the OLAP council who are also members of the subcommittee are eligible to vote on revisions or additions to the APB-1 benchmark, associated addenda, and errata.

Current members of the OLAP benchmark subcommittee include:

Applix	Martin Richmond-Coggan
Dimensional Systems	George Spofford
IBM	Bill Malloy

NCR   [Steve Hummel](#)

Hyperion Solutions   [Daniel Druker](#)

OLAP Council   [David Lourie](#)

OLAP Council   [Larry Barbetta](#)

Oracle   [Doug Liles](#)

All members of the OLAP benchmark subcommittee were heavily involved in the process to revise the original OLAP benchmark and to create the OLAP Council APB-1 Benchmark Release II, a process which began in January 1998 and was completed in November, 1998.