OLAP Council
APB-1 OLAP Benchmark

Comparative Analysis of Audited
Arbor Essbase 5 and Oracle Express 6.1
APB-1 Benchmark Results

March 1998

Key Findings

- ❑ Arbor® Essbase® OLAP Server completed the entire OLAP benchmark in 49 minutes, six times faster than Oracle Express.

- ❑ Arbor Essbase completed the entire benchmark nearly two hours before Oracle Express finished the bulk load and calculation and more than three hours before Oracle Express finished the incremental load and calculation.

- ❑ Arbor Essbase required nine lines of code to implement the benchmark; Oracle Express required 1,176 lines— 130 times more lines of code.

- ❑ Arbor Essbase was nearly twenty times faster than Oracle Express for loading and calculating data.

- ❑ Arbor Essbase used eight times less disk space than Oracle Express to store the same data. Oracle Express exploded the raw data provided by the OLAP council by a factor of 86.

- ❑ Arbor Essbase used a single 4.2 GB disk drive to implement the benchmark, whereas Oracle Express used more than 23 GB of storage including a 16.8 GB RAID-0 disk array.

- ❑ Arbor implemented the benchmark application exactly as a user would; Oracle's benchmark results do not reflect real-world applications:

  - Oracle ran 250,000 static queries programmed in advance into stored procedures. Arbor used only ad-hoc dynamic queries.

  - Oracle turned off data integrity to artificially enhance performance. Arbor did not sacrifice data integrity or recoverability.

- ❑ Both the Arbor Essbase and the Oracle Express benchmarks were run with identical servers, clients and networking, and were audited by the same auditor.

Introduction

This document compares the results of Arbor Essbase 5 and Oracle Express 6.1 for the industry-standard OLAP Council APB-1 OLAP benchmark. The reader may find it helpful to review the OLAP benchmark specification on the OLAP Council Web site at www.olapcouncil.org.

This document also refers to the OLAP Report. The OLAP Report is an independent review of OLAP technologies, applications, case studies and products that is available on the Web at www.olapreport.com. The OLAP Report includes an extensive independent analysis of both the Arbor Essbase and Oracle Express OLAP benchmark results.

The full disclosure of the Oracle Express OLAP benchmark may be found on Oracle Corporation's Web site at www.oracle.com. The full disclosure of the Arbor Essbase OLAP benchmark may be found on Arbor Software's Web site at www.arborsoft.com.

For a detailed, side-by-side comparison of how the OLAP benchmark application is implemented in Arbor Essbase and in Oracle Express, see Arbor Software's Web site at www.arborsoft.com.

OLAP benchmark background

The OLAP Council designed the APB-1 OLAP benchmark to provide a means of comparing the overall performance and difficulty of implementing a realistic sales and marketing analysis application with different OLAP servers. The OLAP benchmark measures the time required to load and calculate a standard OLAP application and to execute a large number of sophisticated multi-user queries. The OLAP council mandates a full disclosure of all code required to implement the benchmark.

The OLAP benchmark implements a six dimensional sales and marketing analysis application. The application analyzes two years of information for 10,000 products, 1,000 customers and 10 channels of distribution. It tracks a wide range of business measures across historical, actual, budget and forecast scenarios.

The benchmark is implemented in three phases. In the first phase, the OLAP server performs a bulk data load of historical information. In the second phase, the OLAP server performs an incremental data load of current and budget information. Both of the data loads are followed by optional precalculation of derived data. In the third and final phase, client workstations issue 250,000 sophisticated multi-user queries to the OLAP server.

Analytical Query Time (AQT) is the total time for the incremental load and calculation and for executing 250,000 queries, divided by 250,000. AQT represents the average time per query, including incremental load and calculation time. Analytical Queries per Minute (AQM) is the inverse of the AQT, and represents the average number of queries per minute including incremental load and calculation time.

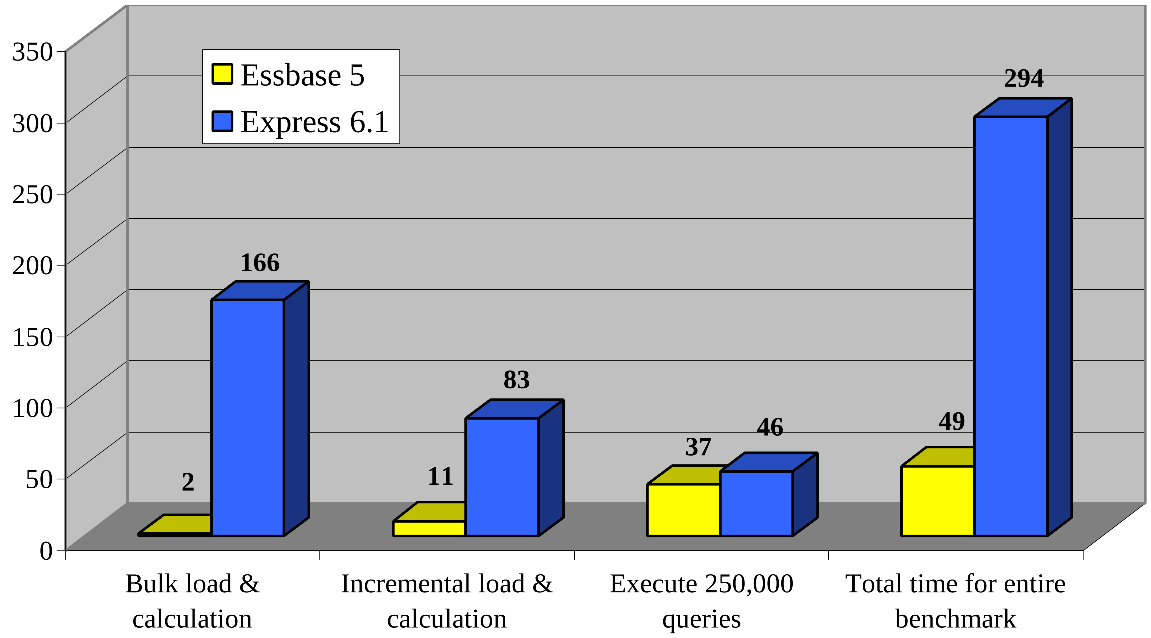
The total benchmark time is the total time required to complete the bulk load and calculation, the incremental load and calculation and to execute 250,000 queries. The total benchmark time differs from the AQT/AQM in that the latter do not include the bulk load and calculation.

Arbor Essbase dramatically outperformed Oracle Express

Arbor Essbase dramatically outperformed Oracle Express in the OLAP Council APB-1 OLAP benchmark. The audited results are summarized below:

	Arbor Essbase 5	Oracle Express 6.1	Comparative Results
Bulk load and calculation	0:02:00	2:45:48	**Arbor Essbase was 83 times faster**
Incremental load and calculation	0:10:44	1:22:50	**Arbor Essbase was 7.7 times faster**
Executing 250,000 queries	0:36:30	0:45:38	**Arbor Essbase was 1.25 times faster**
Total time for entire benchmark	0:49:14	4:54:16	**Arbor Essbase was 6 times faster**
AQT (Analytical Query Time)	0.01134 seconds per query	0.03083 seconds per query	**Arbor Essbase was 2.7 times faster**
AQM (Analytical Queries Per Minute)	5,293 queries per minute	1,946 queries per minute	**Arbor Essbase was 2.7 times faster**

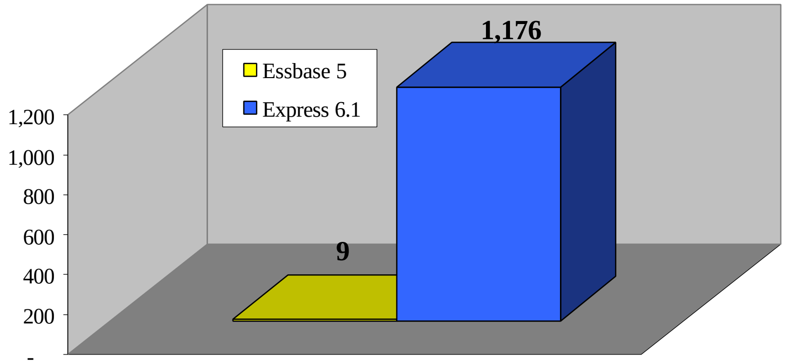
OLAP benchmark results (times in minutes, shorter is better)



Arbor Essbase required 130 times fewer lines of code than Oracle Express

The volume and complexity of the code required to implement the benchmark is a direct reflection of the time and expense to deploy and maintain real world analytical applications. The OLAP Council mandates the full disclosure of all code required to implement the benchmark application, which makes it easy to compare the complexity of implementing an identical application in different OLAP servers.

Arbor Essbase required just 9 lines of code to implement the APB-1 OLAP application. Arbor used Arbor Essbase Application Manager, a graphical OLAP application development and management tool, to build the OLAP benchmark application. Applications created using Essbase Application Manager are data driven and automatically adapt to changing business requirements, minimizing the cost of ongoing maintenance.



Arbor Essbase required 130 times fewer lines of procedural code than Oracle Express to implement the OLAP benchmark.

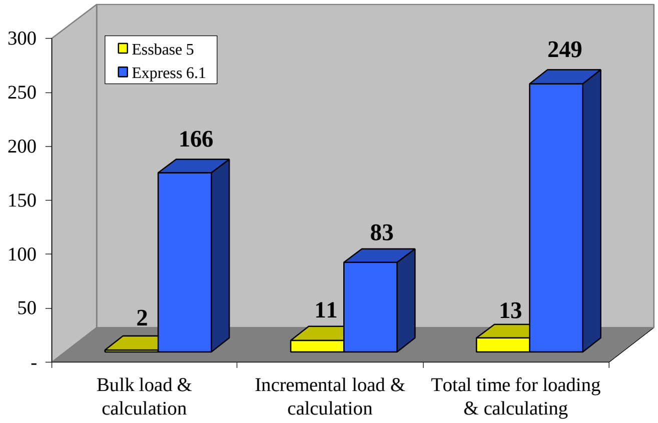
Oracle Express required 1,176 lines of complex procedural code to implement the same application. The document on Oracle's Web site disclosing the Oracle Express code required to build the application is 23 pages long. In Oracle Express, when business requirements change, programmers must manually rewrite this lengthy code, creating an expensive ongoing maintenance burden.

The OLAP Report at www.olapreport.com concludes: "**the Express code is extremely lengthy and forbiddingly complex compared to that required for Arbor Essbase**."

Arbor Essbase loaded and calculated data 19.5 times faster than Oracle Express

Because OLAP applications may be updated with fresh information weekly, daily, hourly or even in real time, the speed of loading and calculating data drives both the capacity and responsiveness of the OLAP server.

Arbor Essbase completed the bulk load and calculation phase of the OLAP benchmark in two minutes, and completed the incremental load and calculation phase in 10 minutes and 45 seconds for a total loading and calculation time of 12 minutes and 45 seconds.



Arbor Essbase loads and calculates the data for the OLAP benchmark application nearly 20 times faster than Oracle Express.

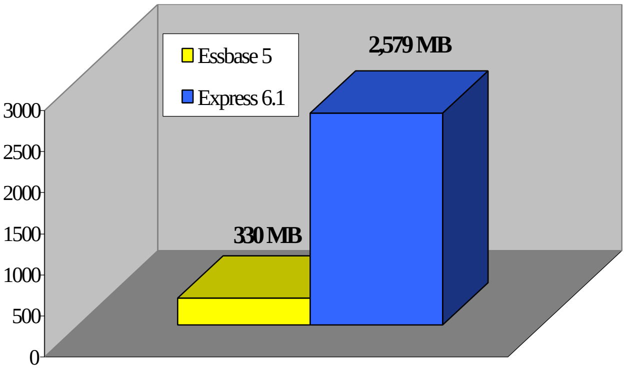
Oracle Express required 2 hours and 45 minutes to complete the bulk load and calculation phase, more than 80 times slower than Arbor Essbase. Oracle Express required an additional hour and 23 minutes to complete the incremental load and calculation, more than 7 times slower than Arbor Essbase. For loading and calculating all data, Oracle Express required a staggering 4 hours and 9 minutes, 19.5 times slower than Arbor Essbase.

Arbor Essbase stored data eight times more efficiently than Oracle Express

Arbor Essbase used 7.8 times less space than Oracle Express to store the data for the OLAP benchmark application. Storage efficiency is an important factor in the scalability and performance of OLAP servers.

Arbor Essbase required only 330 MB to store the data for the benchmark application. Arbor Essbase includes patented sparsity management facilities that maximize the efficiency of storing and managing sparse OLAP data. Arbor Essbase also provides several compression algorithms to physically compress data before it is stored on disk.

Oracle Express exploded the same data by a factor of 86, requiring 2,579 MB to store it in Oracle Express. Poor sparsity management and lack of data compression dramatically limit the scalability of Oracle Express.



Arbor Essbase required eight times less space than Oracle Express to implement the OLAP benchmark application

Arbor used a single 4.2 GB disk drive; Oracle used 23 GB with a disk array

Arbor used a single 4.2 GB Ultra-SCSI disk drive to implement the entire OLAP benchmark. The operating system, the OLAP benchmark data files, the Arbor Essbase server software, and the Arbor Essbase database files were all stored on this single disk drive. Oracle used 23.1 GB of disk space to implement the benchmark. Oracle used a 2.1 GB Ultra-SCSI disk drive for the operating system, a 4.2 GB Ultra-SCSI disk drive for temporary space, and a 16.8 GB disk array made up of four 4.2 GB Ultra-SCSI disk drives in a RAID-0 configuration for the Oracle Express database files.

Oracle used multiple disks to enhance the Oracle Express results. The Oracle Express database was 2.6 GB in size yet was stored on the 16.8 GB disk array, wasting more than 14 GB of disk space. The operating system and benchmark data files were placed on their own separate disk drives, wasting even more disk space. Using extra disk drives speeds performance by reducing the amount of work each disk drive must perform. This does not come without a price: the disk drives used by Oracle cost more than five times more than the disk drives used by Arbor.

Both benchmarks used the same server, clients, networking, and auditor

Arbor and Oracle used identical server computers, networking and client computers. Both Arbor and Oracle used identical 4-processor Hewlett Packard NetServers with 1 GB of RAM running Windows NT 4.0. Both Arbor and Oracle used 10 base-T Ethernet networks running TCP/IP connected to four identical client computers. The sole hardware difference was that Arbor used a single 4.2 GB Ultra-SCSI disk drive to implement the entire OLAP benchmark, whereas Oracle used 23.1 GB of Ultra-SCSI disk storage including a 16.8 GB RAID-0 disk array.

George Spofford from Dimensional Systems audited both the Arbor Essbase 5 and the Oracle Express 6.1 OLAP benchmarks. George Spofford is an OLAP council certified APB-1 benchmark auditor, and can be reached via e-mail at george@dimsys.com.

Arbor used only dynamic queries; Oracle used static queries banned by industry standard benchmarks

Arbor used only dynamic queries in the OLAP benchmark. Arbor’s results reflect exactly the performance that users can expect to see in real applications using any of the more than 50 best-in-class Essbase-Ready™ tools and applications that are integrated with Arbor Essbase.

Oracle hard coded and compiled static queries into 10 stored procedures inside Oracle Express. Oracle used these stored procedures to run all 250,000 of the OLAP benchmark queries. Pages 19 to 21 of the Oracle full disclosure document show the code used to create the static query stored procedures. Real world OLAP users issue an enormous variety of interactive ad-hoc queries, not a small number of pre-defined, hard-coded static queries. Oracle’s results, therefore, do not reflect the performance that real users can expect to see in real applications.

Each of the 10 stored procedure programs was compiled in advance and executed only a single predefined, parameterized static query. The following, from page 4 line 85 of the Oracle disclosure, is the Oracle Express code that compiles the stored procedures.

```
" Finally, with all objects now defined, compile the QRYn programs:
_Counter = 10
while _Counter ge 1
do
trap on QRYCMP noprint
cmp &joinchars('QRY' convert(_Counter text 0 no no))
QRYCMP:
```

As an extreme example, query program 10 is hard-coded, so that if the user asks for “Actual” data, the program immediately returns “No Inventory Data for Scenario Actual” without querying the Oracle Express database to see if data really exists. Here is a sample of the code from page 21, line 15 of the Oracle disclosure that implements query program 10:

```
if arg5 ne 'Actual'
then signal 'QRY10.01' -
joinchars('No Inventory data for scenario ' arg5 '.')
fetch INVENTORY labeled
return
doend
```

Oracle’s use of static query programs clearly violates the spirit of the APB-1 benchmark specification, which states “OLAP queries are ad hoc and very dynamic.” The industry standard TPC-D benchmark specifically bans stored procedures and static queries on the grounds that they allow vendors to generate artificial results that are not representative of real-world implementations.

The OLAP Report at www.olapreport.com has the following to say about Oracle’s use of precompiled static queries: **“Clearly, this must improve the query performance, but it is unlikely that real customer applications could use such extreme performance tweaks without losing all flexibility.”**

Oracle turned off data integrity to artificially enhance performance

Arbor ran the OLAP benchmark exactly as real-world customers would, without sacrificing data integrity or recoverability.

Oracle explicitly turned off data integrity and recoverability. Turning off data integrity speeds loading and calculation performance at the expense of data integrity, rollback and recoverability. Few real-world customers are willing to risk turning off data integrity and recoverability for business critical OLAP applications.

The Oracle Express programming language includes a keyword called *inplace* that can be used when creating Oracle Express databases. When a data structure is defined as *inplace*, updates will be directly written to the database without any data integrity or rollback capability. In the Oracle Express benchmark, Oracle stored the vast majority of data in *inplace* data structures. The Oracle Express code below showing how the *inplace* structures are defined is from the Oracle disclosure page 4, line 30.

```
define INVENTORY integer <TIME PRODCUST_CP<>> inplace
define BUDGET decimal <MEASURE TIME PRODCUST_CP<>> inplace
define FCST decimal <MEASURE TIME PRODCUST_CP<>> inplace
define ACTUAL decimal <MEASURE TIME ACTUAL_CP<>> inplace
```

The OLAP Report at www.olapreport.com analyzes Oracle’s use of *inplace* updates to turn off data integrity and recoverability and states that Oracle’s results are **“not representative of the typical performance achievable in most real-world applications which would rarely risk using such techniques.”**

Summary of findings

Arbor Essbase dramatically outperformed Oracle Express in the industry standard OLAP benchmark.

Arbor Essbase completed the entire OLAP benchmark in just 49 minutes, finishing the benchmark more than three hours before Oracle Express finished loading and calculating the database. Arbor Essbase completed the entire OLAP benchmark six times faster than Oracle Express, outperformed Oracle Express by a factor of 19.5 for loading and calculating data, and required eight times less disk space to store the same data.

The Arbor Essbase benchmark was performed under real-world operating conditions, similar to those used by more than 1,400 corporations using Arbor Essbase today. The APB-1 OLAP benchmark results prove conclusively that Arbor Essbase is the fastest performing, easiest to deploy OLAP server for enterprise OLAP applications.