VERSANT

BENCHMARK REPORT

V/OD IN THE TELECOMMUNICATIONS INDUSTRY

A side by side comparison of the Versant Object Database with an RDBMS

Written by: Versant China Support Center, April 2010

Translated by: Versant U.S. Support Center, September 2010

Version Date: 9/20/2010

Website: www.versant.com

INTRODUCTION

This benchmark report is based on a basic application performance test simulating a specific business model to process Call Data Records (CDRs) that is typical for the telecommunications industry. The purpose of the report is to highlight the performance advantages of the Versant Object Database (V/OD) compared to a commercial relational database system (RDBMS).

Telecommunications operators are facing great performance challenges in their current operation environment using an RDBMS. When processing huge amounts of Call Data Record (CDR) objects, the existing system operated by one of the leading Chinese telecom operators was quickly running out of server resources. To avoid additional hardware investments the operator turned to Versant to evaluate a more efficient database solution.

Many telecom related applications are facing similar challenges in their production environment. This benchmark report evaluates two fundamentally different database systems, V/OD, a modern high performance object database system, and a commercial RDBMS and investigates how these systems perform in such a real world application environment.

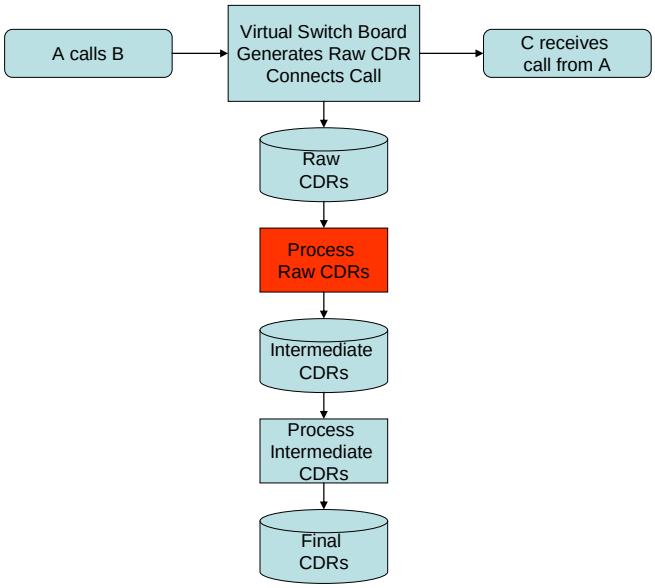
Benchmark Requirements & Business Logic Description

The benchmark uses a subset of a Call Data Record (CDR) data model which is causing the performance problems in a virtual switch board application (VSB). The principle functionality of the virtual switch board application is as follows:

- An incoming phone call needs to be routed to an extension or a mobile phone. The incoming phone call generates a “raw CDR”, e.g., A connects to VSB.
- The VSB looks up the extension or mobile phone forward, and connects the call. This generates a second “raw CDR”, e.g., VSB connects to B.
- Periodically, the VSB runs a process to match up these “raw CDRs” and generates an “intermediate CDR”, e.g., A has called B. This process requires a SELF JOIN in a relational database, the most critical bottleneck in the virtual switch board application.

- The intermediate CDR is subsequently processed for billing and other services. This process generates a “final CDR A called B” in the system

The following picture illustrates the data flow:



The benchmark measures the processing time of the application both for V/OD 8 and the RDBMS. The data used in the benchmark is actual raw CDR data generated by a virtual switch board system, representing approx. 3m raw CDRs, the data volume of 1 – 3 operating days of a VSB.

1) The production system is multi-threaded. Each thread is activated each second to store 30~50 objects in the database. More than 1,000,000 transactions/calls are stored in the database per day. The average number of transactions could exceed 10,000,000 per month. The production system is supposed to manage the data for about 6 months, a volume that can grow easily to 100m CDRs.

2) About 98% of the raw CDR is representing a pair like call (A,VSB) and call(VSB,B). A is the original caller, VSB is the virtual switch board, B is the receiving party, either an extension in the VSB or a forward to a mobile phone. The benchmark application searches for a matching raw CDR pair from A to B.

3) Periodically, the benchmark application processes the raw CDR pairs and generates an intermediate CDR. The system must generate the intermediate CDR by finding all A->B call logs resulting from A->VSB and VSB->B CDRs. This processing of the raw CDRs includes 5 sets of *select* and *insert* operations.

4) The process described in 3), finding call (A, B), is causing the main performance issues in the RDBMS, because it requires a SELF referencing JOIN operation.

5) The application logs the system performance statistics data and generates the benchmarks for the database operations *object insertion* and *object query*.

The Benchmark Environment

The benchmark results in this report are based on the following hardware / system software environment:

Benchmark Environment	
CPU Frequeny	2672.52MHz (Real Data)
CPU Count	1/ 4 Core/ 8 Thread
Memory	6GB
Disk	1TB / 10000RPM
OS	Windows 2003 Advanced Server

The benchmark is performed on a „clean“ environment, e.g., there are no side effects from potential production issues such as 3rd party software, database or memory fragmentation, etc.:

1) The benchmark is performed on a server platform with only the minimum 3rd party software stack installed, including Microsoft Windows 2003 Advanced Server and applicable patches, SUN JDK, the benchmark application itself and certain server CPU/Memory monitor software.

2) Every second, the benchmark records the total CPU and memory usage, great indicators of the overall utilization of the system to perform the tasks. The CPU/Memory data is collected by a 3rd party system monitoring tool. The performance impact of the tool is minimal and can be ignored.

The Benchmark Application

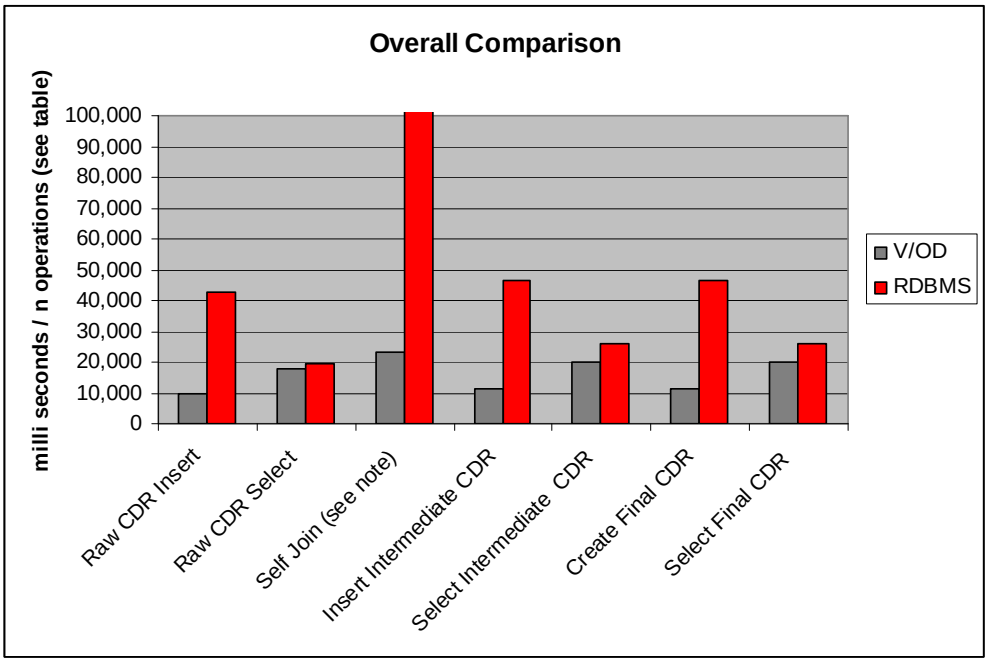
The benchmark application is written in Java.

This benchmark is not following any standard benchmark methodology, such as TPC or 007, but rather uses a specific business model and use case, providing more meaningful and realistic results when comparing two rather different database technologies.

BENCHMARK SUMMARY

We performed the benchmark with a target data set of 3,000,000 raw CDR objects.

Following is a graph and a summary table with the results (green and orange rows represent significant deviations between the tested database systems, white rows represent comparable results):



Note: The Self Join results for the RDBMS are “off the charts”, see table below.

Table: Benchmark Results

	VOD 8.0			RDBMS			Difference
	Time in ms	Avg Mem %	Avg CPU %	Time in ms	Avg Mem %	Avg CPU %	Factor RDBMS vs. V/OD
Raw CDR Insert Time for 10,000 transactions	9,903	43.46%	0.58%	42,865	43.46%	0.23%	4.3
Raw CDR Select Time for 1,000 objects	18,107	52.00 %	16.58%	19,460	51.25%	25.23%	1.1
Self Join (retrieve related Raw CDR) Time for 1,000 objects	23,423	52.55%	25.01%	1,803,923	68.89%	93.07%	77.0
Insert Intermediate CDR Time for 10,000 transactions	11,203	50.55%	1.23%	46,465	51.55%	1.56%	4.1
Select Intermediate CDR Time for 1,000 objects	20,111	52.85%	18.23%	25,768	54.62%	40.23%	1.3
Create Final CDR	11,203	54.85%	3.23%	46,465	54.80%	3.10%	4.1
Select Final CDR Time for 1,000 objects	20,111	56.00 %	20.23%	25,768	54.62%	39.18%	1.3

BENCHMARK ANALYSIS

The benchmark data shows that the Versant Object Database 8.0 performs many areas significantly faster. **Most interesting is, that the true bottleneck that required a SELF JOIN operation in the relational database, was a “non issue” with V/OD 8. Versant could resolve the requirements due to its data modeling capabilities almost 80 times faster, helping the telecom operator to scale his operation without increasingly expensive hardware and software license costs.**

Following is a more detailed analysis of these findings:

- **V/OD shows massive performance advantages over the RDBMS on INNER JOIN and complex queries.**
  The RDBMS must create a large dynamic intermediate query, with a compute cost several dozen times greater than V/OD 8. In addition, V/OD's performance object insertion and for complex queries has only a linear decrease with increasing numbers of CDRs.
  The performance advantage of V/OD 8.0 is achieved by not only saving the objects, but also saving object relationships. Those relationships must be recreated by the RDBMS every time the intermediate CDR is required.. V/OD 8 basically doesn't have to create intermediate queries, which results in massive performance gains and overall lower requirements on the server CPU and memory.
- V/OD 8 and the RDBMS have similar CPU usage when inserting objects / records in a single transaction. However, V/OD 8 is approx. 4 times faster, and according to the benchmark data (especially for a large amount of the data), CPU utilization and processing time shows a more linear increase as the database grows.
- V/OD 8.0 is significantly more CPU efficient while being much faster. This CPU utilization obviously varies with different use cases, e.g., queries.

EXHIBIT A: V/OD BENCHMARK DATA

V/OD 8 Raw CDR Benchmark

Benchmark Task List	
Task Definition	Batch process to generate raw CDRs in V/OD 8
Target Data Set	Generate business data simulation for about 3 days in V/OD 8, approx. 3,000,000 records
Benchmark Target	To test object creation performance of V/OD 8.
Start Time	2010-03-23 15:24:45
End Time	2010-03-23 16:14:59
Benchmark Span	50 Min. 33 Sec.
Demo Case Run Status	Normal
Available CPU Usage Snapshot Count	3,015
Max CPU Usage	100%
Avg. CPU Usage	0.58%
Available Memory Usage SnapShot Count	3,015
Max Memory Usage	53%
Avg. Memory Usage	43.46%
Insertion Time Cost Sample Count	300
Min Insertion Time Cost on each 10,000 objects	0 ms
Max Insertion Time Cost on each 10,000 objects	28,609 ms
Avg Insertion Time Cost on each 10,000 objects	9,903.33 ms
Avg Insertion Time Cost per object	0.99 ms

RDBMS Raw CDR Benchmark

Benchmark Task List	
Task Definition	Generate raw CDRs in RDBMS
Target Data Set	Generate business data simulation for about 3 days in RDBMS, approx. 3,000,000 records.
Benchmark Target	To test record insertion performance of RDBMS.
Start Time	2010-03-24 8:16:29
End Time	2010-03-24 11:54:13
Benchmark Span	3 Hours 38 Min. 44 Sec.
Demo Case Run Status	Normal
Available CPU Usage Snapshot Count	13,065
Max CPU Usage	98.44%
Avg. CPU Usage	0.23%
Available Memory Usage SnapShot Count	13,065
Max Memory Usage	53%
Avg. Memory Usage	43.46%
Insertion Time Cost Sample Count	300
Min Insertion Time Cost on each 10,000 objects	0 ms
Max Insertion Time Cost on each 10,000 objects	98,453 ms
Avg Insertion Time Cost on each 10,000 objects	42,865 ms
Avg Insertion Time Cost per object	4.29 ms