Oracle Rdb Migrations – Assets and Solutions Overview

Introduction

This document attempts to summarize the various solution options that VSI has available to assist VSI OpenVMS customers who might be interested in migrating away from Oracle Rdb. It should be noted that the text does not aim to discuss the migration process or go into any specific detail regarding the relative advantages of one solution over another but is simply aimed at identifying some of the tools and solution options that are available today and how they might be relevant. It is envisaged that that this document will be updated over time as additional tools and solution options are identified.

The text includes references to specific projects where one or more of the tools discussed have been used to successfully deliver the required solution.

VSI Services Assets

The following sections summarize at a high level the various tools VSI and the VSI Application Services team have that could be utilized or adapted as part of any Oracle Rdb migration project. It is important to note that in some cases these tools would likely require modification, and this would need to be factored into the overall effort for the project in question. It is also likely that there are few other items we have neglected to mention below, but in general terms, and working in conjunction with partners we are well-placed in terms of tools and technical expertise to advise Oracle Rdb users interested in migrating to another database on their options and to provide services to deliver a successful migration solution.

SQL Relay

We have to date largely focussed on using SQL Relay to provide a replacement solution for the Oracle RDBMS client on OpenVMS and have invested considerable effort into developing an embedded SQL pre-processor (and associated API) that can be used in conjunction with the SQL Relay client on VSI OpenVMS to replace the Oracle Pro*C, Pro*COBOL, and Pro*Fortran embedded SQL pre-compilers, however SQL Relay could also prove to be very useful for some Rdb migrations, particularly for situations where customers want to move to databases off-platform for which there is no native OpenVMS (or where it would perhaps be cost-prohibitive to port or implement some such client). SQL Relay provides considerable flexibility and could potentially be appropriate for quite some number of situations, however it is likely that some additional tooling would be required for a fair number of

projects, but the cost of any such tooling would be included in what we would charge to the customer, and we could then leverage any such tools on subsequent projects, where appropriate.

Given that there is some level of complexity with the configuration of SQL Relay (it is more than just running an embedded SQL pre-compiler and linking in a client library), it would also be reasonable to charge customers for support of the SQL Relay environment (in addition to any tools or library code we might develop). It should also be noted that while our embedded SQL pre-compilers have been targeted at addressing the Oracle client situation, these tools and APIs could be readily adapted to work in conjunction with other database targets and/or with other programming languages.

UnixODBC

VSI and members of the OpenVMS community have ported versions of open-source UnixODBC library to VSI OpenVMS, and VSI have used it for several projects in the telecommunications and financial services sectors. As the name suggests, the library provides an open-source ODBC implementation, and in addition to the library itself, VSI and others have used this library to port ODBC drivers for SQL Server and Sybase, MariaDB, PostgreSQL, and Oracle RDBMS. The library is also compatible with the Mimer ODBC driver. The SQL Server and Sybase ODBC driver uses the FreeTDS API, and the MariaDB and PostgreSQL drivers leverage their respective OpenVMS client APIs. UnixODBC can therefore be used to implement connectivity with the databases mentioned in a standard and efficient manner and can be used in conjunction with tools such as the VSI SQLMOD pre-compiler. Work done to date by VSI for various customers using UnixODBC in conjunction with the FreeTDS client API shows generally very good performance and stability, and installation and configuration of the software is straightforward.

In general terms, UnixODBC incurs somewhat less overhead than SQL Relay and would likely be the preferred approach for any situations where there may be a choice between the two, although that being said, there may be situations where some of the advanced features of the SQL Relay server could be used to good effect to achieve very high levels of scalability that might not be directly possible via more direct ODBC interface.

References:

- Modernization prior to x86-64 migration

VSI SQLMOD

This is a tool VSI are actively developing that can be used to pre-compile Rdb SQLMOD files into C code, which can then be compiled and linked with COBOL applications. The resultant linked applications then interact with SQL Server via UnixODBC using the FreeTDS ODBC driver. To date, the tool has been demonstrated to successfully parse more than 99% of the 1300+ SQLMOD files to which it has been applied, and while there are some minor issues still to be resolved we are confident that the tool is working well. While the SQLMOD pre-compiler it its current form tool has been implemented to generate code that can be called from COBOL and to interact with Microsoft SQL Server, it would be straightforward to enhance the solution to support other languages and database targets, providing a largely generic solution for any customer that uses Rdb SQLMOD to interact with

their Oracle Rdb databases, with most modifications likely to be related to dealing with variances in SQL dialects and date/time handling.

It should also be noted that the solution fully integrates with VDD (the VSI Data Dictionary), allowing it to be used in situations where SQLMOD code references CDD records and types.

The VSI SQLMOD solution also includes a useful unit-test tool called CBO that can be used to create scripts to test individual SQLMOD procedure or sequences of such procedure calls and to compare the results obtained for Oracle Rdb with those obtained for the target database, and the VSI SQLMOD pre-compiler includes functionality to generate sample test cases for all procedures in a given SQLMOD file that can be adapted as necessary for specific test cases. The testing tool integrates with VDD and includes a powerful scripting capability.

RTO

This is a tool that was originally developed for a specific customer project to partially automate the conversion of an Oracle Rdb database to Oracle RDBMS; however, the tool has now been modified to support other targets (leveraging UnixODBC), including SQL Server, PostgreSQL, and Mimer. The tool is able to interrogate the Oracle Rdb metadata and (with some limitations) generate from this an equivalent schema for the target database, including entities such as table definitions, constraints, and indexes. It should be noted that the RTO tool is not perfect or necessarily complete in this regard for all target database options, but it provides an excellent start and easily be enhanced to address specific project requirements. The tool also provides facilities to read data from source Oracle Rdb tables, perform any transformations that might be necessary, and insert the resultant data into corresponding tables in the target database. This direct data transfer mechanism may not be sufficiently performant for the conversion of any Oracle Rdb database of appreciable size (despite the tool providing parallel processing capabilities), however the RTO tool also includes options to publish the contents of the Oracle Rdb database to message queues or streams, which can be done very quickly and efficiently, whereupon the data can be consumed and inserted into the target database in a decoupled manner, independent of any interaction with the source database. There are also facilities to export table contents to text files that can then be imported using import tools provided by the target database.

References:

- Moving from Oracle Rdb to Oracle RDBMS

PostgreSQL client for VSI OpenVMS

VSI provides a well-proven and current port of the PostgreSQL client for VSI OpenVMS, which can be used as part of any Oracle Rdb migration to PostgreSQL on Linux (or Microsoft Windows). The client kit includes an embedded SQL pre-processor and API for C, however it would be necessary to develop new or adapt existing pre-compilers to support other programming languages, which could be readily done, if required.

References:

- Multi-site database migration

MariaDB client

The situation with regard to MariaDB is similar to that for the PostgreSQL client, although in the case of MariaDB, it should be noted that VSI currently does not have any embedded SQL pre-processors. As noted previously it would be possible to develop new or adapt existing pre-compilers from other products to generate code that could be used in conjunction with the MariaDB client API.

FreeTDS

FreeTDS provides an API and various command line tools that can be used on VSI OpenVMS to interact with Sybase (SAP ASE) and Microsoft SQL Server databases (both databases use the TDS wire protocol, albeit different versions of the protocol). The FreeTDS API and associated tools are well-proven, with several large VSI OpenVMS customers in the telecommunications, financial services, and manufacturing sectors using it on VSI OpenVMS Integrity and VSI OpenVMS x86-64 as a replacement for the unsupported OpenVMS Sybase client. From work done for these and other customers, VSI additionally has developed embedded SQL pre-compilers for both C and COBOL, however it should be noted that these pre-compilers are currently Sybase centric, and some work would be required to enhance them for use with Oracle Rdb embedded SQL code.

References:

- OpenVMS x86-64 migration project
- Modernization prior to x86-64 migration
- Complex Alpha to Integrity migration
- Change of master database to SQL Server

VDD (VSI Data Dictionary)

The goal of VDD is to provide a near-drop-in replacement for Oracle CDD (which uses Oracle Rdb), and it could have a very important role to play, as many VSI OpenVMS users use CDD extensively to ensure consistency of data structures across all application tiers, from the user interface down to the database or RMS file level, with it not being uncommon for users to have many thousands of field and record definitions stored in their CDD repositories that are used by their applications to ensure such consistency. Having a near-drop-in replacement solution will save Oracle CDD users a great deal of time and cost from an application and database migration perspective and will considerably reduce the overall risk profile of any such migration work. VDD can currently be used by all VSI language compilers, DECforms, Datatrieve, ACMS, and TDMS. There is still some work to be done to complete the VDD product and to resolve several outstanding issues, however Beta versions of the product have been successfully used by several field-test users.

Punk

This VSI solution may or may not be useful in the context of Oracle Rdb migrations, but it is worth at least mentioning, as there may be some potential use-cases. The Punk gateway essentially provides a means of interacting with Oracle Rdb databases via the PostgreSQL wire protocol, meaning that (with some constraints) any PostgreSQL client implementation can theoretically be used to interact with an

Rdb database. This may be useful in certain contexts where it might be desirable to move incrementally (using some form of phased approach) from Oracle Rdb to PostgreSQL, or to create links between a PostgreSQL database and one or more Oracle Rdb database tables. It be noted that Punk does not currently support encrypted connections or the ability to present database metadata in a manner expected by certain clients; however, both of these matters can be easily addressed and in general terms the solution works very well.

ESQL and ESQLC

These two tools refer to the embedded SQL pre-processors for COBOL and C that were mentioned above in the discussion about FreeTDS. As noted, these tools are Sybase-centric and some work would be required to adapt them for use in an Oracle Rdb context, but such adaptation would certainly be feasible. For example, changes would be required to accommodate differences in SQL grammar between Sybase and Oracle Rdb, and Oracle Rdb supports (or more precisely permits the use of) a somewhat richer set of datatypes in application code that interacts with the database. These tools could also be readily enhanced to include support for VDD.

References:

- OpenVMS x86-64 migration project
- Complex Alpha to Integrity migration

Vole and Rdb CDC

In addition to providing facilities to process sequential log files, the VSI Vole solution also includes functionality to intercept and process changes to Oracle Rdb database records, leveraging the Oracle Rdb LogMiner interface. Such a facility is potentially useful for some Oracle Rdb database migration scenarios where the Oracle Rdb database is still being actively used and any changes need to be replicated in the new (target) database, or possibly RTO might be used to export the contents of an Oracle Rdb database to a set of queues or streams, and Vole could then be activated to intercept any ongoing changes to the Rdb database and append those changes to the corresponding queues or streams, effectively eliminating the need for a big-bang database cutover, however the practicalities of any such approach would need to be carefully considered on a case-by-case basis.

RCOCVT

In addition to supporting the use of SQL in application code via embedded SQL and SQLMOD, Oracle Rdb also supports the older RDML (Relational Data Manipulation Language) as a host-language embedded query interface for accessing the relational database from programming languages such as COBOL, Fortran, BASIC, Pascal, and C, with COBOL arguably being the most common. While RDML was largely supplanted by SQL once full SQL integration in Rdb became available, it is still found in many older applications. RCOCVT is a prototype tool for converting COBOL code with embedded RDML to COBOL and embedded SQL, whereupon one of the other solutions mentioned in this document may be used to migrate the resultant embedded SQL code to another database platform.

The above tools essentially fall into four high-level and somewhat inter-related categories, namely development tools, conversion tools, integration tools, and client APIs for various target database options. This categorisation is summarized in the following table, noting that some of the tools fall into several categories:

Tool	Category(s)
SQL Relay	Development tool, client API(s)
UnixODBC	Client API(s)
VSI SQLMOD	Development tool
RTO	Database and data conversion/migration
PostgreSQL client for VSI OpenVMS	Client API
MariaDB client	Client API
FreeTDS	Client API
VDD (VSI Data Dictionary)	Development, conversion (CDD)
Punk	Integration
ESQL and ESQLC	Development tool
Vole and Rdb CDC	Integration, database and data conversion/migration
RCOCVT	Development tool

Looking at it very simply, development tools would be used to facilitate the continued maintenance and development of existing applications on VSI OpenVMS following migration to the new database environment and would work in conjunction with client APIs and/or integration tools to provide the necessary connectivity to the new database environment. Client APIs implement the wire protocol utilised by the database in question and provide a set of API functions that can be used by application code to interact with the target database environment. Development tools such as embedded SQL pre-compilers generate code that leverages these APIs, either directly, or via an abstraction layer to simplify code generation and to facilitate cleaner integration with the application language in question. Integration tools in this context are essentially middleware used to help facilitate data migration in some way, and conversion tools are used to automate or semi-automate aspects of the database migration process, including data conversion and migration, database schema replication, and modification of application code.

It should be noted that none of these tools alone equate to providing a complete migration solution but rather they can be used to facilitate (possibly with modification) one or more aspects of an overall

solution, and in general any such solution needs to be designed and planned on a case-by-cases basis, taking into careful consideration a myriad of technical and non-technical factors. Superficially there is also overlap between some of these tools. For example, the pre-compilers all perform essentially the same function; however, what they generate is dependent on the target API and/or database.

Key third-party and partner solutions

The following sections briefly describe several partner and/or third-party solutions that are likely to be relevant to specific Oracle Rdb migration scenarios.

Mimer

Mimer is a well-proven high-quality database solution for VSI OpenVMS and was the first commercial database to be ported and supported on VSI OpenVMS x86-64. In recent years, Mimer have done a lot of work to include various Oracle Rdb compatibility features into the Mimer database product including support for Rdb-specific datatypes and syntactical constructs, and support for modular SQL compilation. Most recently they have invested considerably in developing a toolset and processes to help customers to migrate from Oracle Rdb to Mimer. This work has yielded positive results, with several migrations having been successfully completed or currently being tested. For those customers wanting to replace Oracle Rdb and keep the database on VSI OpenVMS, there is no question that Mimer is an excellent option.

Sector7

Sector7 provide a powerful solution to move Oracle Rdb customers using SQLMOD or embedded SQL from Oracle Rdb to PostgreSQL (and potentially to other targets as well). This is a well-proven proven and credible solution that includes some excellent features, including performance optimizations for cursor operations, which are often poorly implemented in many Oracle Rdb applications.

SharkSQL

SharkSQL is a new and innovative, distributed relational database management solution for VSI OpenVMS and Microsoft Windows Server (with planned support for Linux). It features full SQL-99 compliance with partial SQL-2013 support, ACID-compliant transactions ensuring data consistency and lost update protection, advanced security measures such as per-table encryption and packet-level security, and a built-in two-phase commit implementation for seamless distributed operations across networks. The database includes an intuitive CLI for management tasks such as backups, restores, and configurations, automatic parameter optimization for immediate usability, and support for MySQL/MariaDB wire protocols, enabling easy integration with existing applications via ODBC, JDBC, .NET, Python, and C interfaces. SharkSQL includes various Oracle Rdb compatibility features similar to those provided by Mimer, but arguably more comprehensive.