

Porting DBMS Linter to OpenVMS/Itanium



Table of contents

Introduction	2
RELEX, the vendor of Linter	2
Linter, the first Russian DBMS	2
The features of Linter	2
The porting nuances	2
Alignment	3
The endianness (the byte order)	4
The platform-dependent code	4
Testing	4
The portable testing system	4
Continuous test runs	5
Compatibility	5
SQL language and API	5
To migrate from another DBMS	5
Future plans	6
Resources	6
Learn more	6

In the area of data security, Linter, the RDBMS produced by RELEX, is one of the world leaders among database management systems. DBMS Linter has very simple and portable architecture that is easy to install and configure, so it was requested to be ported to OpenVMS/Itanium®.

Introduction

RELEX, the vendor of Linter

The RELEX group, one of the leading Russian software vendors, has been in the industry since 1990. The group consists of the companies RELative EXpert Systems, since 1990 (that gives the RELEX abbreviation to the name of the group), and RMCSOft since 2004.

The RELEX group develops, delivers, and supports both custom and packaged products in the areas of information security, BI, Web-based systems, embedded and real-time complexes, and mobile solutions. The flagship product of RELEX is the professional multi-platform DBMS named Linter.

Linter, the first Russian DBMS

The history of Linter began in 1980, the time of the development of relative database theory.

The RDBMS project called BARS was successfully completed in 1983. That was designated for PDP-11 clones but in 1985 its developers, situated in the city of Voronezh, Russia, decided to implement the concept of mobile DBMS that could be compatible with many platforms and use open standards. By 1990 this brought to the set of DBMSes that were compatible with most of the platforms available at that time including Intel® 8086-based families as well as VAX-based ones. The resulting product introduced under the name of INTEREAL DBMS happened to be the Linter's prototype. Linter currently is ported to most of the modern hardware and software platforms including real-time systems and mobile platforms. It was brought to OpenVMS/Alpha around the millennium but there was no Itanium support; hence in 2012 it was decided to fill this gap.

The features of Linter

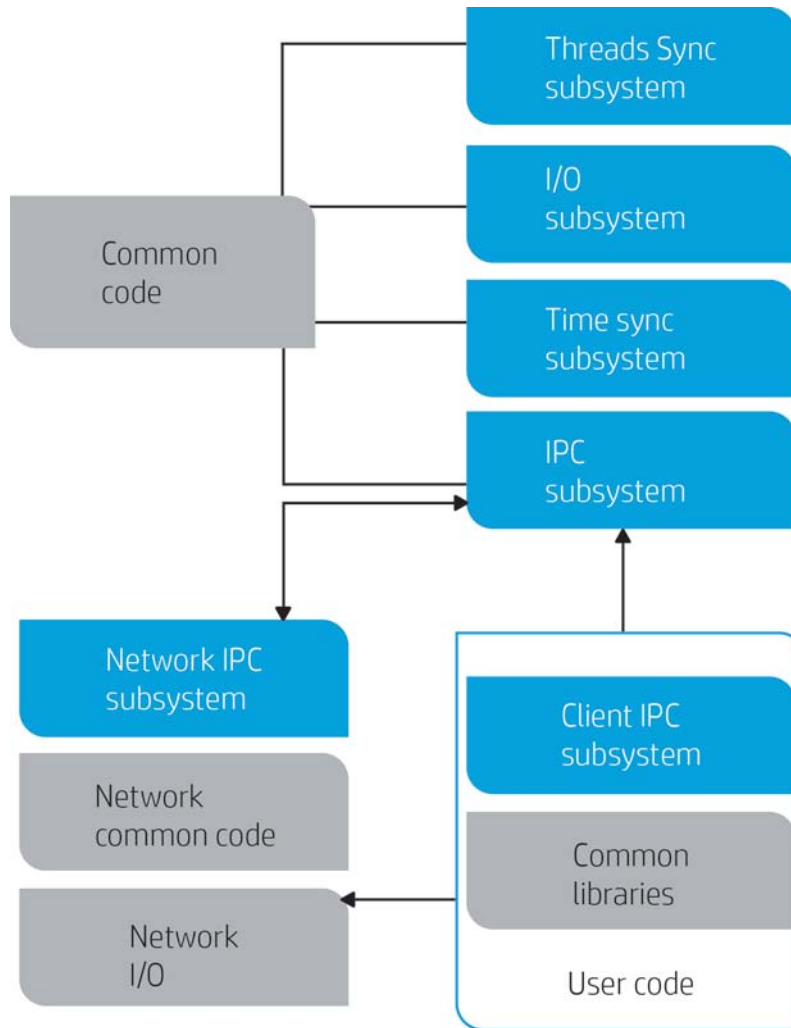
In order to be portable and to support real-time operating systems DBMS Linter:

- has very simple architecture
- can be easy installed and configured
- has low footprint and makes limited usage of computing resources
- offers very reliable code
- provides high performance for OLTP transactions

Linter is thus very useful for the embedded solutions, e.g., for the control systems like SCADA.

The porting nuances

Linter has the modular-based DBMS kernel written in ANSI C. Thus, while porting, for the most part of the source code this allows not to pay attention to what the goal platform is. But it is not possible to make all parts of the code completely abstract from the hardware or operating system used. The parts that need attention are highlighted in blue in figure 1. The common code is normally platform-independent.

Figure 1. The code structure for the portable system

The code designers should take care of the following nuances:

Alignment

This may be the one of the most significant problems while porting the code. The fields, structures, and variables should be aligned according to the particular platform's word boundaries. Attention should be paid to the compiler and the code optimizer behavior as they may align the structures automatically according to the platform rules or, vice versa, may make it unaligned while packing the structures to minimize the memory and stack usage. In both cases the optimizer may change the order of the variables in the structures and the structures may obtain some "holes" not used by data but needed for successful alignment. This needs to be taken into account for any operations with data so for every particular platform you need to understand what a compiler and code optimizer can do for you to align the data and what rearrangements you need to do manually.

You need to remember that it's not worth making any "hardcoded" assumptions about the order of the data in the structures and its addresses as you need to call only the aligned data and despite that keep the code portability. E.g., if the platform has 32-bit word you need to load the 32-bit integer starting from the address that is a multiple of 4 while if you have 64-bit word you should load the same integer starting from the address that is a multiple of 8.

The endianness (the byte order)

Different platforms may use different byte orders. A big-endian means that a machine stores the most significant byte first while a little-endian one, vice versa, stores the least significant byte first. Some architectures are bi-endian so the endianness can be switched using some hardware or software features. Itanium architecture is bi-endian so OpenVMS on Itanium has little-endian architecture as well as on Alpha and VAX while HP-UX on the same Itanium hardware has big-endian architecture like on PA-RISC. This should be considered for the data exchange over a network. The client or server should make data conversion if the endianness doesn't match.

The platform-dependent code

The particular platform may use the specially written code or special code development rules to speed up the input-output, interprocess communications, as well as process and threads synchronization (e.g., locking). The code to be ported may require to be adopted for usage of these functions to get the optimal performance.

Despite most operating systems including more or less standard POSIX functions in run-time libraries, it is still not guaranteed that these libraries will lead to performance effectiveness. While porting Linter to OpenVMS/Itanium it was found that OpenVMS V8.3 (used by the customer who requested the port) has much faster native functions for input-output than the POSIX ones. It was also found that the thread-to-thread communications in the pthreads happened to be enormously slow to meet the requirements. So the developers decided to return to AST-based native mechanisms for the interprocess communications that were used for the old Linter versions running on VAX/VMS and use thread-less architecture for the database engine kernel. As a result the number of requests per second processed by the redesigned kernel grew about 10 times.¹ The trade-off was that it required the change of about 2 percent of the total code. Despite the fact that the AST-based code already existed for the old VAX/VMS-oriented versions of Linter it still needed to be adopted for OpenVMS V8.3 on Itanium.

The above considerations are also worth taking into account while porting the Open Source application written for Linux to OpenVMS.

Testing

Tests are always the most complex part of any porting project. It is almost impossible to develop any complex software without any automatic or automated testing system as the potential problems should be found early and quickly. The testing system for Linter DBMS is specially developed by RELEX for cross-platform tests.

The portable testing system

The next significant potential problem for ported code would be how it will be tested. It is very expensive test that manually. The special cross-platform DBMS testing system was developed to check all subsystems of the DBMS automatically or in automated mode on a lot of platforms.

The testing system contains more than a hundred test sets for the DBMS to check SQL language, SQL extensions, procedure-oriented programming language, full text search, security, geometric data types, replication subsystem, backup subsystem, more than 10 different APIs for the DBMS as well as the automated tests of GUI utilities for different platforms.

The test sets contain the ones for regression tests, performance tests, and stress tests. Regression tests are used to check the DBMS health and quality so they are regularly updated to reflect the new scenarios and situations found in production. The performance tests are designed to check the performance dependence on the code changes. The stress tests allow to run the DBMS using huge amount of data for a long time to check the reliability. It is possible also to run the tests in a random manner and to create the multi-user, multi-threaded complex test profiles that can combine several tens of thousands of simple tests.

The check takes into account the different operating system architectures and hardware features including the CPU type and word size. The test system is written in C and UNIX® Shell using some GNU utilities. As a result, the test code and scripts to run are portable.

¹ Test configuration on rx3600 shows about 100 requests per second with POSIX-threaded database engine but performance grew to 1000 requests per second on the same hardware and the database while using the database engine kernel with AST-based interprocess communications.

Continuous test runs

The continuous integration system is used to find the problems early and thus deliver a reliable product. The system compiles, links, and tests the sources continuously using several different platforms in parallel. The tests also consider the difference in features for each platform. The results of each platform run are emailed to the developers so they may quickly test the changes they made, check the influence, and react accordingly.

The advanced automated testing system makes the port much easier and faster. This particular Linter port to OpenVMS/Itanium took just about 3 man-months.

Compatibility

The implemented set of standards is the critical factor for the success of the DBMS. It is difficult to find the development platform that would be impossible to use Linter with.

SQL language and API

Linter DBMS allows the use of most of the possibilities listed in the SQL: 2008 specification. It supports the data types introduced in this standard, the subroutines callable from SQL (stored procedures), `AS` for the `CREATE TABLE` statement, `SEQUENCE` generator to identify the columns, `MERGE` statement, `TRUNCATE TABLE`, Window functions, triggers `INSTEAD OF` and much more.

The existing programming libraries for Linter allow to use the data stored in the DBMS while using C, C++ (including QT library), C#, Java, Perl, PHP, Ruby, TCL, Delphi, and 4GL Intcom.

Linter DBMS also supports ODBC, JDBC, OleDb, .Net (including MonoDevelop plugin) interfaces.

The main migration toolkit was developed to migrate from Oracle to Linter. It contains the automatic database converter that allows to load both data and the database structure to Linter from Oracle dump.

The tool converts the sequence generators, tables, indexes, views, and the data it contains at the time of migration. Oracle RDBMS uses only `NUMBER` type for numeric data and represents all other variants of numeric data required by SQL standard as `NUMBER` variants so Linter converter may optionally convert numeric data into other standard types like `INT`, and `BIGINT` using the precision and scale of Oracle's `NUMBER`.

The stored procedures may be converted from Oracle PL/SQL to Linter's stored procedures language using an automated tool. The converter can correctly translate most of the widely-used PL/SQL patterns like cycles, transitions, exception blocks, and procedure calls. This instrument also performs automated substitutions for Oracle and Linter functions with different names and syntaxes if required. The tool also converts Oracle error codes into Linter error codes as well as translates the global variables, cursors, types and subtypes, and data structures. The highly different patterns are commented to simplify the manual code change and optimization. Thus the triggers and the stored procedures are saved as if developed for Linter. This simplifies further support for the translated code.

It is important to note that there is no need to install or use Oracle RDBMS for the conversion. It is enough to have just an Oracle's dump file with the needed database. The converter can run in any operating system supported by Linter so the initially Oracle-based database can be migrated to an OS that was never supported by Oracle.

OCI (Oracle Call Interface) emulator for Linux DBMS simplifies the migration for the application using Oracle COM Interface. These applications may also use the emulation of Oracle internal tables in addition to more than 400 Oracle SQL functions and stored procedures.

To migrate from another DBMS

It is possible to export the database structure and data as the set of SQL commands and load it to Linter DBMS. As Linter supports most SQL standard features it is not too difficult to correct the exported data if such correction is needed. More automated tools are planned for in the future.

Future plans

It is planned to add SQL/PSM language support to significantly extend the Linter stored procedure abilities. SQL/PSM collects many convenient features and functions of PL/SQL, PL/pgSQL, TransactSQL and also will allow to use the elements of object-oriented programming. It is planned to meet major ISO/IEC 9075-4:2011 functional requirements using SQL/PSM.

The structure of DBMS will also gain object-oriented elements like table inheritance and object-oriented methods that will allow to consider the tables of the database as the actual object-oriented models.

The current development plans also include tools for the full conversion automation for triggers and stored procedures of the main DBMSes like Oracle, PostgreSQL, Interbase/Firebird, MS SQL, and MySQL into Linter's stored procedure formats.

Conclusion

The experience of Linter port to OpenVMS on Itanium shows that:

- Even the code designed for portability may require significant rework to get the performance on the target platform.
- The automated test system reduces the required labor and allows to find the problems at the early stages.
- It is possible to bring the applications from different versions of Oracle DBMS, including unsupported old ones, to current Linter version on the same OpenVMS on Itanium. This allows to keep the applications running in the supported environment.
- Further development will allow to enhance the Linter on OpenVMS on Intel® Itanium® abilities and performance.

Resources

RELEX company and products overview
relex.ru/en/main

Linter DBMS
linter.ru/en/main

HP OpenVMS
hp.com/go/openvms

Learn more

For more information on Linter and how you can achieve Itanium support for Linter, visit linter.ru/en/main.

Share your feedback or queries on the VMS Technical Journal [here](#).

Sign up for updates
hp.com/go/getupdated



Rate this document

© Copyright 2013 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel, Itanium, and Intel Itanium are trademarks of Intel Corporation in the U.S. and other countries. Oracle and Java are registered trademarks of Oracle and/or its affiliates. UNIX is a registered trademark of The Open Group.

