Migrating Applications to PostgreSQL - Experiences

Open Technology Centre (OTC) had undertaken migration of few legacy applications to Open Source Based Application. The migration has to be applied at multiple layers depending up on the design principle used by the application.

The article shares the migration process utilized, the issues that are to be carefully addressed, techniques adopted and concludes with a set of recommendations.

R.Ramesh
Technical Director, Tamil Nadu
rramesh@nic.in

T.Edward Sam
Principal Systems Analyst, Tamil Nadu
edwardsam@nic.in

In this article the experience of the team in migration of two applications with different design principles are shared. In one of the applications most business functions were done at the application server level while the others has been implemented and concentrated most functions in the back end database. In both cases the migration was carried out to use Open Source database PostgreSQL. The first application is Office Procedure Automation (OPA) whose back-end was migrated from MS SQL Server. The second application is Employees Provident Fund Organization (EPFO) Application which was migrated from Oracle back end to PostgreSQL. In both the applications, the middle tier was JAVA. The migration activity involves migrating the database schema, porting data and migrating SQL Statements in the application program.

The migration process is described below:

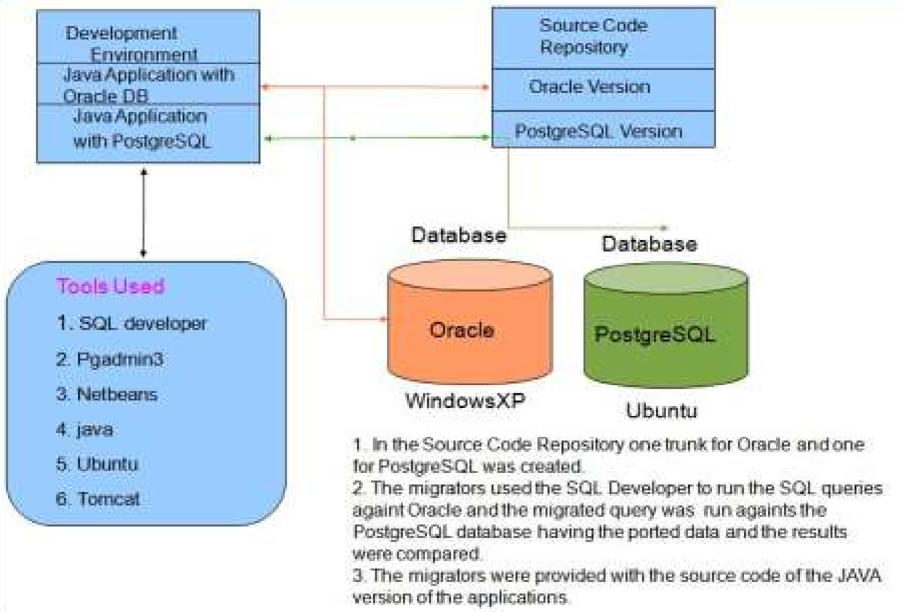
Database Level

Schema Conversion

i) Identify the equivalent data type and replace with it. The preferred mode of conversion is manual. Export schema as a script and alter the data types. Tools such as "Sqlways" and Open Source Tool "Kettle" can be used; manual inspection is necessary.

ii) Care must be taken in the selection of data types so that loss of data does not happen.

Porting Data: The data porting can either be achieved by using native database facility or by third party tool. In case of MS-SQL Server 2000, Data Transmission Services (DTS) along with PostgreSQL ODBC. For Oracle to PostgreSQL, Open Source Tools such as Kettle can be used; or commercial tool such as "Sqlways" can be used.



1. In the Source Code Repository one trunk for Oracle and one for PostgreSQL was created.
2. The migrators used the SQL Developer to run the SQL queries againt Oracle and the migrated query was run againts the PostgreSQL database having the ported data and the results were compared.
3. The migrators were provided with the source code of the JAVA version of the applications.

Migration Environment

Trigger/Procedure: Tools can help reduce the time in the conversion of Triggers and Stored Procedures. Open Source Tool Ora2pg (a perl script) and the "Sqlways" can be used. But manual verification of functionality is a must.

Identifying equivalent functions/facility in the target database which may not be available as a direct facility. A combination or a work around can provide similar functionality. Eg. Nested Table in Oracle functionality can be achieved through Arrays in PostgreSQL.

Application Level

At the application level depending upon the quality of the application and the developer's awareness about standards conformance various changes are to effected in the DML statements.

SQL Statement conversion: SQL syntax can be converted by using tools which will reduce time. The converted query can be run against the ported database; the out put has to be checked with the equivalent native SQL statement to confirm the correct functionality.

Syntax difference: Some of the Syntax differences are listed below.

i) Concatenation function differs in most of the RDBMS. Oracle uses ||, PostgreSQL uses || and MS SQL Server uses +.

ii) Decode function.

iii) Execute Query/Execute Update

iv) Date format

Procedures Invocation: The style of invocation of Procedures depends on the JDBC driver of the target database. This requires changes in the JAVA program.

Approaching Migration

- **Steps Involved**

  i) Create a replica of the production environment that needs to be migrated.

  ii) Migrate Schema and Port Data to the PostgreSQL. Check the data randomly and ensure that porting is correct.

  iii) Create a separate application instance which is connected to the Ported Data.

  iv) Covert the Procedures/Triggers. This can be done in parallel with the JAVA program migration.

  v) Prepare list of Statements that require to be changed. The List will get augmented as the migration progresses. Share the list with the migration team.

  vi) Run the Application Against the ported data, capture the error and correct the SQL errors in JAVA programs (Server routines).

- Team Composition: Depending up on the complexity & code size, the following team composition is recommended.

  i) Team for Converting JAVA Programs

  ii) Team for Converting Procedures and Triggers.

Recommendations for Migrators

Based on the experience gained the following recommendations are made for project teams proposing to migrate applications to PostgreSQL.

- While migrating, use standards based SQL statements instead of using same statements along with a supplementing function package such as "Orafce". Example: Use "COALESCE" instead of "nvl".
- Avoid dependency on Supplementing or Compatibility packages. They are to be used only as a last resort. Even when such packages are used, use it only for those tasks for which equivalents are not available.
- Create indexes depending on the query used in the application. This will improvise the performance.
- It is imperative to understand that tools assist a developer in migration and rarely can be automatic. Human inspection, validation and verification are imminent in any migration project.

Recommendations for Developers

- Use Object-relational mapping (ORM) tool where ever possible. This will ensure that the project will be neutral to all databases.
- Avoid dependency if application is implemented using a specific database, avoid using database specific features. Use the more universal ANSI standard based calls rather than vendor specific calls. ex. Use CASE .. SWITCH instead of Decode. i

Acknowledgement- *Special thanks to Sh. S. Ramachandran, PSA for his contribution in preparing the article.*