instaclustr

Migrating Oracle to PostgreSQL®

March 2021

Kirk L. Roybal

Contents

Features and Benefits

There are a host of features that make PostgreSQL a compelling database platform and many of them are directly related to Oracle. We will focus on the ones that become architecturally interesting for Oracle to PostgreSQL migrations.

- PostgreSQL instances are not linked to financial requirements
- PostgreSQL foreign data wrappers are superior to federation
- PostgreSQL authentication hooks to nearly anything
- PostgreSQL is infinitely extendable

There are no external vendors participating in your growth strategy with PostgreSQL. Since there is no financial incentive to reduce the number of PostgreSQL instances, it is very common for PostgreSQL installations to have many more servers and services. These may be divided by business unit, traffic pattern, scalability, tuning, availability, recovery, logical services, or any other criteria. For our purposes, we'll focus on using PostgreSQL effectively as a service delivery platform with custom tuning parameters for your specific workload.

Foreign data wrappers are the rough equivalent of data federation. However, there are far more adapters to different data systems in PostgreSQL than with any other database. And the margin isn't small. Using various flavors of replication combined with foreign data, PostgreSQL provides a push-pull approach that allows for infinite configurability in your processing stack.

Not only can PostgreSQL use nearly any authentication type, it can use it locally or remotely as the protocol and bindings allow. This means that a 1-second authentication is no longer blocking a 20-millisecond query, or creating artificially high dwell times for service calls.

PostgreSQL provides an extension system with an unlimited API into the inner working of the database itself. You can add operators, data types, functions, and internals with access to any part of the system.

Requirements

You should know a few things before going down the road of Oracle to PostgreSQL migration. For starters, this is an **expert job**. There are coding assistance tools (e.g. Ora2Pg) that will create some basic structures for you. But these tools only create an outline around the actual code migration. There are no tools that are anywhere close to complete coding replacements. Even the tools that provide some level of schema migration are either incomplete or leave some room for interpretation along the way.

PostgreSQL has much more limited parallel query support. By that I mean that a single backend handles a single request for the lifetime of the request. There are a few parallel processes in PostgreSQL, such as sequential scanning or sorting. However, these do not compare to the parallel query structure of Oracle. You will not get the same per-query vertical performance curve.

- Memory contention works differently. There is a tuning art in getting approximately the same memory operation out of PostgreSQL as Oracle.

- PostgreSQL only has nested transactions when using the procedural language after version 13. Otherwise, there are no nested transactions at all.

- PostgreSQL does not require a dummy DB name or even a FROM clause. If you want to add 2 and 2 you can just SELECT 2+2.

- There are enough syntactical and grammar differences between plSQL and plpgSQL that every function/SQL conversion will have to be manually reviewed.

- There is an Oracle compatibility layer extension for PostgreSQL (orafce). This extension is about 90% complete. That is to say, not complete enough to perform an automated migration.

- PostgreSQL naming convention is generally recommended to use lowercase identifiers with under-scores for word boundaries. PostgreSQL provides a way to quote identifiers for extended characters and case retention, but it is not recommended to do so.

- Reverse key, bitmap, and join indexes are not supported in PostgreSQL.

- Tablespaces exist in PostgreSQL, but do not have anywhere near the purpose or flexibility as those in Oracle. They are not recommended use for simple data segregation or disk management purposes.

Steps of Migration

Assessment

- Data size
- Code complexity
- Code size
- Schema complexity
- Background processes
- Application integration
- Infrastructure

Schema Migration

- Data types
- Operators
- Referential integrity
- Triggers
- Functions

Data Migration

- Switchover strategy
- ETL vs. ELT
- Data enrichment
- Conversions
- Units and precision
- Representation at rest
- Representation in the line
- Representation to the user

Functional Testing

- Comparative testing harness vs. manual testing
- Regression management

Performance Testing

- Memory/Disk/Network/CPU scaling
- Bottleneck identification

Switchover

- High Availability
- Discovery Recovery

Assessment must be performed by a qualified database migration consultant with field experience. PostgreSQL has equivalent functions, data types, operators, and strategies for every Oracle workload. However, the strategies can vary greatly for how to specifically handle the situation. A direct translation is possible in trivial cases, but as the complexity increases, the underpinnings demand custom answers to each architectural challenge.

As a general rule, the latest available version of PostgreSQL should be the target. It is very likely that a couple of minor releases of PostgreSQL will occur during the development and implementation phase of the migration. The latest available minor version of PostgreSQL is recommended corresponding to the major version that was available at the beginning of the effort.

A sample dataset for functional testing is absolutely mandatory during the development phase, along with a basic regression testing framework. This will mitigate nasty surprises in the field. Ideally, the outputs of the PostgreSQL functions and processes should identically match the Oracle output at some point in the process.

Important Questions

Questions that will need to be answered for assessment include (but are not limited to):

- The total size of the database in bytes, number of tables
- The number of custom functions and approximate line counts
- Oracle version
- Oracle installed features (forms, etc.)
- Language of the applications
- Current backup strategy vs. intended backup strategy
- High availability architecture (including network infra.)
- Disaster recovery strategy

About Instaclustr

Instaclustr delivers reliability at scale through our integrated data platform of open source technologies such as Apache Cassandra®, Apache Kafka®, Apache Spark™ Elasticsearch™, Redis™, and PostgreSQL®.

Our expertise stems from delivering more than 100 million node hours under management, allowing us to run the world's most powerful data technologies effortlessly.

We provide a range of managed, consulting, and support services to help our customers develop and deploy solutions around open source technologies. Our integrated data platform, built on open source technologies, powers mission critical, highly available applications for our customers and help them achieve scalability, reliability, and performance for their applications.