

How Speedment™ Coexists With Oracle RDBMS

For Accelerated SQL Database Applications

March 31, 2017

Why Read This White Paper?

Relational databases like Oracle are superior when it comes to handling complicated querying and analysis of data. It is also a very robust and safe environment to store the data. However large databases especially the ones used for web applications often run into scalability problems with slow applications as a result.

The purpose of this paper is to describe how Speedment coexists with an Oracle database in order to avoid a risky migration and still achieve the extreme speed of an in-memory database.

1. Oracle DB App Acceleration

Speedment allows database applications to run orders of magnitude faster without upgrading the server hardware. All legacy tools will still work and there is no need for migration.

Speedment leverages an in-memory data store to accelerate data access by orders of magnitude. Since the memory store is read only and uses standard SQL to access the database, it coexists without any friction with existing Oracle applications both in terms of consistency and interface.

Since Speedment generates the code used to access the database from the database metadata, the application is guaranteed to build on bug free data access code that automatically adapts when the database data model changes. This means that when an evolving database model creates a mismatch with existing application code the problem will be found even before testing. Thus, the Speedment part of the application will automatically remain up to date with the data model and thus seamlessly coexists with a legacy application being in constant development.

For more details about Speedment, please see the Speedment - the Java Stream ORM White Paper.

2. Piecewise Implementation- No Migration

Upgrading a legacy database system can be a daunting task. With large numbers of tables in the database and many lines of code both in backend and frontend upgrading it all to a more modern data access framework may incur both high costs and considerable risk. Instead of starting a migration project, we advocate a piecewise implementation. Speedment is designed to facilitate such a process. The Speedment tool coexists nicely with the current backend. It will instantly deliver value to the selected piece of the solution whereas the rest remains unactuated.

A typical approach for a Speedment integration project is to single out a particular database query and its associated backend code. By replacing only that part of the backend data access, the system will leverage benefits of Speedment for the parts affected without making any change to the rest of the legacy system.

In general, adding a new data source to an existing system may introduce inconsistencies, but since the Speedment in-JVM-memory datastore is read-only, the acceleration does not add any new ways of writing data. Therefore, the underlying relational database remains the source of truth for all queries.

Speedment provides two different APIs. Speedment and Speedment Enterprise uses a Java Streams API, whereas the product named Ext Speeder leverages this API to auto generate a backend that has a full REST API that can be used by the frontend directly. When using the Java API, the Speedment code

will coexist with legacy backend code since it is standard Java code with a runtime that does not interfere with but rather builds on JDBC. For the Ext Speeder case, the auto generated backend provides a REST API to the parts of the data model that is integrated. This REST endpoint can coexist with the legacy endpoint, allowing the frontend client to choose which implementation to use for which problem.

3. Agnostic Deployment

While the section above describes how a partial Speedment migration will coexist with a legacy Oracle system with focus on the software handling the database, in some cases it may be required to be even more conservative in terms of legacy backend impact. Particular useful for the REST API case, the Speedment tool supports introducing a dedicated Speedment server while leaving the legacy backend solution untouched.

Since piecewise migration to Ext Speeder means introducing a separate REST endpoint for Speedment accelerated data access, this endpoint may be served from a dedicated server. The Speedment server then appears as a separate backend machine, accessing the relational database using standard JDBC connections.

While such a deployment means that the relational database now feeds two different back ends with data, the total number of requests is considerably smaller since the Ext Speeder backend only periodically updates its datastore contents while continuously providing the frontend with data from the datastore.

Speedment™ - the Java Stream ORM for Accelerated SQL Database Applications

+1 650 387 4069, info@speedment.com