

A Scotas white paper
September 2013

Scotas Push Connector

Introduction

When you have to perform searches over big data, you need specialized solutions that can deal with the velocity, variety and volume of this valuable information which analysis allows you to implement better solutions and to delineate appropriate business strategies. An important challenge in this area is how to bridge the two worlds: the company's big data and normalized enterprise data in an efficient, real-time and integrated way.

Scotas products

Our unique push technology can capture automatically your relational data changes and automatically keep in sync an external searching instance.

We currently have support for Solr and ElasticSearch products.

Scotas Solr Push Connector

Solr Push Connector provides a near real-time free-text search solution which can be integrated really easy on any website or application. You can start with one machine and scale to hundreds.

This is a two-way connector which integrates Oracle database with Solr/Lucene search engine.

The key features of Solr Push Connector are:

- Sends data to the Solr out-of-the-box installation via JSON over HTTP.
- Consumes data during query time with new SQL operators and functions.

This connector does not require the Oracle internal JVM since it is developed completely on PL/SQL. It also works on Oracle 10g express, 10g standard and Enterprise Edition and obviously with any 11g installation.

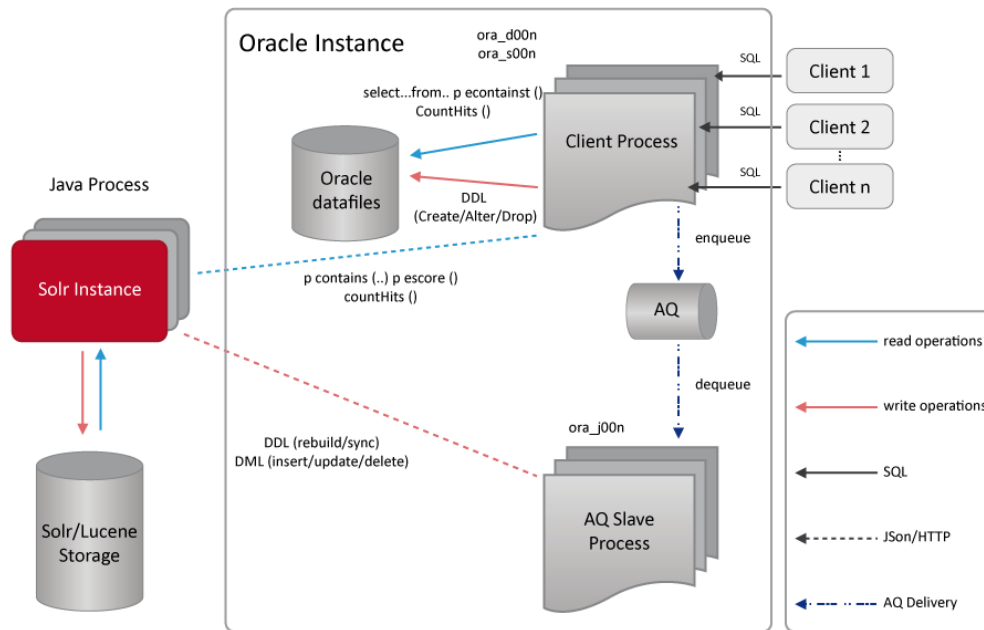


Figure 1: Diagram of Scotas Solr Push Connector architecture.

Scotas Elasticsearch Push Connector

Like Scotas Solr Push connector if you need Solr searching facilities but with elastic deployment the Elasticsearch Push Connector is the solution for your application.

ElasticSearch is an Open Source (Apache 2), distributed, RESTful and designed for the cloud search engine built on top of Apache Lucene.

Scotas Elasticsearch is a connector which keeps your SQL data in-sync with your Elasticsearch out-of-the-box installation using JSON over HTTP, it also provides query facilities in SQL by adding new relational operators and functions.

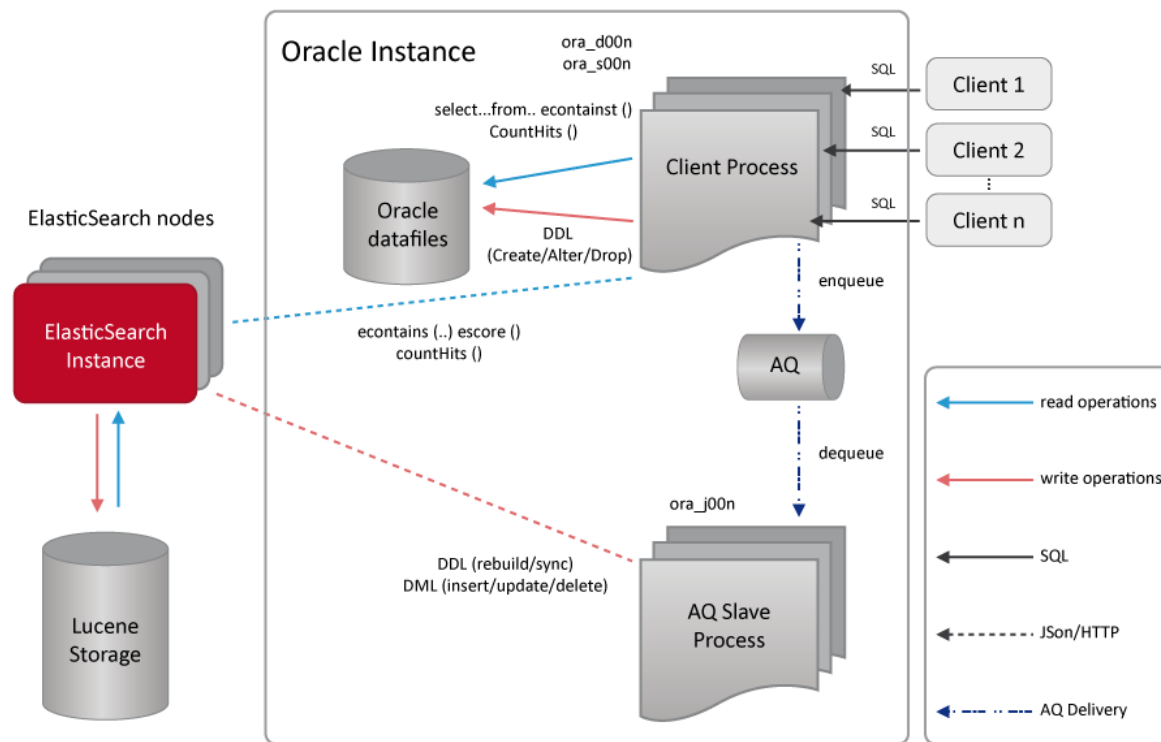


Figure 2: Diagram of Scotas ElasticSearch Push Connector architecture.

Examples of Use

From the DBA or application developer's perspective the integration of Scotas Push Connector with your current system is very simple.

We give you an example of using our Solr push connector's product.

Creating an Index

Suppose you have a table called PRODUCTS full of records. The columns of this table are id, cat (for category), name, features, price, etc. Also, you have a Solr stand-alone working on your environment. Once Scotas Push Connector is installed on your Oracle® RDBMS, you need to create an index into the table and fields you want to synchronize with your external Solr.

The next SQL sentence is very similar to the way to normally create indexes except for a couple of parameters that Solr Push Connector requires.

```
CREATE INDEX PRODUCTS_PIDX ON PRODUCTS (ID) INDEXTYPE IS PC.SOLR
PARAMETERS (' {Updater:"localhost@8983", Searcher:"localhost@8983", CommitOnSync:true, SyncMode:OnLine, LockMasterTable:false, IncludeMasterColumn:false, HighlightColumn:"name, features", DefaultColumn:text, ExtraCols:"id \"id\", cat \"cat\", name \"name\", features \"features\", manu \"manu\", includes \"includes\", price \"price\", popularity
```

Code 1: Example of Solr index creation with Scotas Push Connector.

The first difference you can find is the `INDEXTYPE` that Solr Push Connector uses (`PC.SOLR`). The `PARAMETERS` list can be adjusted to your current environment. In the example of Code 1, the `Updater:"localhost@8983"` and `Searcher:"localhost@8983"` point to the Solr stand-alone updater and searcher process. `DefaultColumn:text` means that Push Connector will use the column `text` as the default one in case that the query doesn't specify any. `ExtraCols:"id \"id\", cat \"cat\", name \"name\", features \"features\"...` are the other columns that will be indexed too.

The index creation is all that you need to start indexing documents with Scotas Push Connector.

Running Queries

Depending on the type of results you want to run there are different classes of SQL queries.

It is very important to mention that all SQL queries interact with the Solr stand-alone installation, process and return the results. It is totally transparent from your point of view. That is the power of Scotas Push Connector.

For example, to search for all the documents which contain the text `video`, run this query:

```
SELECT id FROM PRODUCTSP where SCONTAINS(id, 'video')>0
```

Code 2: SQL query to get documents that match with `video` string.

If you want to match with only one field, add the field name in the following way. For example, here it matches all the documents which has `video` in their `name`.

```
SELECT id FROM PRODUCTS P WHERE SCONTAINS (ID, 'name:video')>0
```

Code 3: SQL query to get documents that match with `video` string in the `name` field.

For range search like looking for prices between up to 400 dollars, run this query:

```
SELECT id,price FROM PRODUCTS WHERE SCONTAINS(ID,'video AND price:[* TO 400]')>0
```

Code 4: SQL query to get documents that match with “video” string and price less than 400.

In some applications, it’s important to compute the score of the matching documents. Just add the `SSCORE(1)` in the column list.

```
SELECT NAME,ID,SSCORE(1) FROM PRODUCTS WHERE SCONTAINS(ID,'video',1)>0
```

Code 5: SQL query to get documents that match with “video” string plus the score.

If you want to sort your documents by a specific column like “price”, the SQL query is the next one:

```
SELECT ID,NAME,PRICE FROM PRODUCTS WHERE SCONTAINS(ID,'video','pricedesc')>0
```

Code 6: SQL query to get documents that match with “video” string and are sorted by “price”.

The get faceted results from Solr, there is a new function called `SOLRPUSHCONNECTOR.facet()` where you need to set the index name and the field you want to facet.

```
SELECT SOLRPUSHCONNECTOR.facet('TUTORIAL_PIDX',null,'facet.field=cat').fields.to_char() F FROM DUAL
```

Code 7: SQL query to get facets.



440 N. Wolfe Road | Sunnyvale, CA 94085 | United States

www.scotas.com

© 2013 Scotas - All Rights Reserved - Patent pending.