Public

Manual Valiant

Document version and date: V01 of 28-02-2012

Contents

1 Valiant

Valiant is a transformation tool. Its purpose is to transform XML documents to RDF files using an XSLT transformation. The basic idea is quite trivial, however currently a free tool supporting this kind of work was not present to our knowlegde. For that reason we have built this and contribute this work to the open source community.

The origin of Valiant is based in a use case experiment of the LOD2 project. For that reason the tool is tailored towards that situation. The base functionalities are:

- Input:
  - File system
  - webDav system
- Transformation:
  - XSLT using XML catalogs
  - Allowing the XSLT to be extended with JAVA functions
- Output
  - File system
  - WebDav system
  - Upload in Virtuoso RDF store

1.1 Base setup and use

The debian package sets up the following

- System wide configuration files
  - /etc/valiant/
- Main script
  - /usr/bin/valiant
- Libs
  - /usr/lib/valiant
- Documentation
  - /usr/share/doc
- Logs
  - /var/log/valiant.log

Valiant must be configured to be used. One can either update the system wide configuration either one can use a local configuration. The local configuration is controlled by environment variables

- VALIANT_CONFIG= absolute path to local valiant.properties
- VALIANT_XML_LIB = absolute path to XML transformation engine (e.g. saxon)
- VALIANT_XSTL_LIB = absolute path to XSLT transformation JAVA extensions

These environment variables enable you to control valiant's behaviour without updating source code.

The original source can be found at

http://code.google.com/p/lod2-stack/source/browse/#svn%2Ftrunk%2Fwp7%2Fvaliant

To start valiant use the commandline:

- Valiant [file|directory]

Valiant has one commandline argument: a file or directory. If the configuration file has been configured to take that parameter into account, valiant will read the file and export the result according to the valiant.properties configuration. If absent, it will try the webdav connection to find the files it has to transform.

1.2 Configuration

This is the main task before using the tool.

There are 2 different input configurations for the tool: The XML files can be local XML-file(s) on the disk, or remote XML-file(s) coming from a webdav. This configuration is modified in the "Inputfile" field in the properties file. Leaving this field empty will select the webdav as input configuration. When you select a path, than there will be, depending on the selected path, one or more local files transformed. Selecting the path of a single xml file will only transform this xml file. Directing the path to a directory, all xml files in this directory will be transformed.

The output configuration has 5 different possibilities. It's possible to store the results in files on the disk, upload them to Virtuoso or upload to a webdav. The last 2 possibilities' are to load the resulting RDF to Virtuoso and Webdav at the same time. Or write to files and upload to Virtuoso at the same time. The writing configuration (also the writing configuration mixed with Virtuoso) will write 2 different types of files on the disk. An RDF file with the transformed content. And a .graph file which contains the graphname where the file will be loaded to when uploading to Virtuoso.

The input & output configurations, and the different folders needs to be specified correctly in the valiant.properties file. Otherwise the tool won't run. An example of how the properties file could look like:

```
# webdav connection settings
host =
user =
password =
webdavUrl =

# virtuoso connection settings
virtuoso.host = localhost
virtuoso.port = 1111
virtuoso.username = dba
virtuoso.password = dba
# set to true|false to immediately load the generated rdf in virtuoso
virtuoso.load = true

# baseURI
baseURI = http://schema.mycompany.com/resource/

# filename regex
regex = [a-z][^\\.]*\\.xml

# max files to load or -1 for all files (eg. max = -1)
max = -1

# absolute path to xsl path to wkd.xsl (eg. /my/path/to/wkd.xsl
xslPath = /home/mysuser/...

# catalog path to catalog.xml
# default values should be ok
# configuration of the catalog via the CatalogManager.
```

```
catalogUrl = catalog.xml
catalogManager = CatalogManager.properties

# absolute output directory
rdfFolder = /home/myuser/...
# default should be ok
logFolder = /var/log/valiant/

# webdav output folder
# the location where the generated RDF files are uploaded.
webdavOutputFolder = /DAV/...

# inputfile folder (if null, transformation will go to the webdav)
inputfile =

# transformation mode (f = write to file, v = upload to Virtuoso, w =
upload to selected webdav, fv = write to file and upload to Virtuoso,
wv = upload to webdav and to virtuoso)
transformMode = f

# implementation mode (choose "xalan" or "saxon")
implementationMode = saxon

# exit processing on the first (IO) exception
haltOnFileError = false
```

2 Performance

2.1 Parallellism

In order to improve the throughput of the tool, valiant applies a simple rule. If the target file does not yet exists, then the transformation is executed. Otherwise the transformation is skipped. This rule allows for specific reruns on subsets of the original data. One just has to delete the corresponding results to define the subset.

This behavior also allows to improve the throughput by starting multiple instances. As each instance checks if the transformation has happened, the throughput is approximately doubled. Of-course the performance gain is limited by the hardware constraints (internal memory, processors, speed of disk).

2.2 High performance upload to virtuoso.

Valiant has the option to upload directly the resulting triples in Virtuoso. For a large amount of files this might not be the best choice. To improve the overall performance, one best uses the bulk upload of virtuoso. Valiant will generate the necessary files for that. <doc>.rdf is the file containing the content and <doc>.graph contains the graph in which the content has to go.

Before running the script, load.sql needs to be modified. load.sql should also be in the zip file. When opening load.sql, the content will contain next line at the bottom:

- ld_dir_all (in dir_path varchar);

- with:
  dir_path – path to the folder where the files will be loaded from.

Only thing that needs to be modified is '/home/wp7/rdf' in this command. This should be redirected to the chosen RDF folder specified in the properties file.

The virtuoso.ini file needs to be modified before this solution can work. This file can be found in the Virtouso directory (on an Ubuntu system this directory will be: /etc/virtuoso-opensource-6.1/). In this file, add the absolute path of the RDF folder to the "DirsAllowed" parameter. After modifying a restart of the Virtuoso service is required. Next command restarts Virtuoso service:

- sudo service virtuoso-opensource-6.1 restart

To run the script now, use next command.

- isql-vt 1111 <USERNAME> <PASSWORD> load.sql

Isql-vt is part of the Virtuoso distribution on Ubuntu.

3 License

4 Contact

TenForce BVBA
Haachtsesteenweg 378
B-1910 Kampenhout - Belgium
Phone +32 (0) 16 31 48 60
Fax +32 (0) 16 65 90 85
Website http://www.tenforce.com

Bert Van Nuffelen
E-mail Bert.Van.Nuffelen@tenforce.com
Website http://www.tenforce.com