Oracle Machine Learning Feature Highlight
OML4Py Universal Client: Getting Started
OML AskTOM Office Hours
Move the Algorithms; Not the Data!

Sherry LaMonica
supported by Mark Hornick
Product Management, Oracle Machine Learning
Agenda

• Overview
• Prerequisites and system requirements
• Installation and configuration
• Demo
Overview
OML4Py Universal Client
Why should I use the OML4Py client?

• Connect to Oracle Autonomous and on-premises Oracle Database instances
  – single standalone client
  – version 19c or 21c

• Enables use of external notebook environments
  – Jupyter, JupyterLab, and Zeppelin notebook environments
  – Python IDEs like PyCharm, Spyder
OML4Py Architecture
Connect to the database from a single client package

Linux 7 or 8

OML4Py universal client package

Oracle Database 19c

Oracle Database 21c

Oracle Autonomous Database 19c

Oracle Autonomous Database 21c
Python Client IDEs
Choose from a variety of Python notebook and client IDE's
Prerequisites and System Requirements
OML4Py Client Components and Prerequisites

- Oracle Linux 7 or 8
- Oracle Database Client or Instant Client
  - version 19c or 21c
- OML4Py 1.0 client installer
  - installs oml package
- Python 3.9.5
  - built from source
- OML4Py supporting packages
  - cx_Oracle, scikit-learn, scipy, matplotlib, numpy, pandas
Linux OS library prerequisites

perl-Env is required by OML4Py, the others are required by Python.

- perl-Env
- zlib-devel
- libffi-devel
- bzip2-devel
- openssl-devel
- tk-devel
- xz-devel
- readline-devel
- libncurses-devel
- libuuid-devel
Python Library Prerequisites
Open-source Python library prerequisites

cx_Oracle 8.1.0     matplotlib 3.3.3
Pandas 1.3.4        scipy 1.7.3
scikit-learn 1.0.1  numpy 1.21.5

OML4Py 1.0 is certified with the listed dependency versions
Installation and Configuration
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
Operating System Prerequisites
Check for missing OS dependencies

$ rpm -qa perl-Env
$ rpm -qa zlib-devel
$ rpm -qa libffi-devel
$ rpm -qa bzip2-devel
$ rpm -qa openssl-devel
$ rpm -qa tk-devel
$ rpm -qa xz-devel
$ rpm -qa readline-devel
$ rpm -qa ncurses-devel
$ rpm -qa libuuid-devel

Identify missing dependencies using rpm commands. If installed, the return value contains the library name and version.

perl-Env is installed

If the library is not installed, no value is returned from the rpm command.

bash-4.2$ rpm -qa ncurses-devel
bash-4.2$ ncurses-devel is not installed
Install missing OS dependencies

- Install missing libraries using `yum` or `rpm` commands
- Running `yum` commands requires root or `sudo` access
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
Install Python from Source
https://www.python.org/downloads/release/python-395

**Files**

<table>
<thead>
<tr>
<th>Version</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gzipped source tarball</td>
<td>Source release</td>
</tr>
<tr>
<td>XZ compressed source tarball</td>
<td>Source release</td>
</tr>
</tbody>
</table>

Choose either Source release

**Download Python and untar the archive**

$ wget https://www.python.org/ftp/python/3.9.5/Python-3.9.5.tar.xz

$ tar xvf Python-3.9.5.tar.xz

**Configure and install**

$ export PREFIX=`pwd`/Python-3.9.5
$ cd $PREFIX
$ ./configure --prefix=$PREFIX --enable-shared

$ make clean; make
$ make altinstall

**Python environment variables**

$ export PYTHONHOME=$PREFIX
$ export PATH=$PYTHONHOME/bin:$PATH
$ export LD_LIBRARY_PATH=$PYTHONHOME/lib
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
Install Open-Source Python Package Dependencies

Install using pip

```bash
$ pip3.9 install pandas==1.3.4
$ pip3.9 install scipy==1.7.3
$ pip3.9 install matplotlib==3.3.3
$ pip3.9 install cx_Oracle==8.1.0
$ pip3.9 install threadpoolctl==2.1.0
$ pip3.9 install joblib==0.14.0
$ pip3.9 install scikit-learn==1.0.1 --no-deps
$ pip3.9 uninstall numpy
$ pip3.9 install numpy==1.21.5
```

*install in this order to ensure correct dependency versions*
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
Download Oracle Instant Client
Use the RPM or ZIP file to install the Basic Package and connect to the database

Oracle Instant Client Downloads for Linux x86-64 (64-bit)

See the Instant Client Home Page for more information about Instant Client.
The installation instructions are at the foot of the page.

Version 19.14.0.0.0 (Requires glibc 2.14)
Base - one of these packages is required

<table>
<thead>
<tr>
<th>Name</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Package (ZIP)</td>
<td><a href="#">instantclient-basic-linux.x64-19.14.0.0.0dbru.zip</a></td>
</tr>
<tr>
<td>Basic Package (RPM)</td>
<td><a href="#">oracle-instantclient19.14-basic-19.14.0.0.0-1.x86_64.rpm</a></td>
</tr>
</tbody>
</table>

choose either installer, the RPM package requires root or sudo access
Download Oracle Instant Client
Use the RPM or ZIP file to install the SQL*Plus package for SQL and PL/SQL

Oracle Instant Client Downloads for Linux x86-64 (64-bit)

See the Instant Client Home Page for more information about Instant Client.
The installation instructions are at the foot of the page.

<table>
<thead>
<tr>
<th>Tools - optional packages</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL*Plus Package (ZIP)</td>
<td>instantclient-sqlplus-linux.x64-19.14.0.0.0dbru.zip</td>
</tr>
<tr>
<td>SQL*Plus Package (RPM)</td>
<td>oracle-instantclient19.14-sqlplus-19.14.0.0.0-1.x86_64.rpm</td>
</tr>
</tbody>
</table>

choose either installer, the RPM package requires root or sudo access
Install Oracle Instant Client
Install the Basic package using either the RPM or ZIP file

RPM installation

$ wget https://download.oracle.com/otn_software/linux/instantclient/1914000/oracle-instantclient19.14-basic-19.14.0.0.0-1.x86_64.rpm

$ sudo rpm -ivh oracle-instantclient19.14-basic-19.14.0.0.0-1.x86_64.rpm

$ export LD_LIBRARY_PATH=/usr/lib/oracle/19.14/client64/lib

Zip file installation

$ wget https://download.oracle.com/otn_software/linux/instantclient/1914000/instantclient-basic-linux.x64-19.14.0.0.0dbru.zip

$ unzip instantclient-basic-linux.x64-19.14.0.0.0dbru.zip

$ export LD_LIBRARY_PATH=/path/to/instantclient_19_4
Install Oracle Instant Client
Install the SQL*Plus package using either the RPM or ZIP file

**RPM installation**

```bash
$ wget https://download.oracle.com/otn_software/linux/instantclient/1914000/oracle-instantclient19.14-sqlplus-19.14.0.0.0-1.x86_64.rpm

$ sudo rpm -ivh oracle-instantclient19.14-sqlplus-19.14.0.0.0-1.x86_64.rpm

$ export LD_LIBRARY_PATH=/usr/lib/oracle/19.14/client64/lib
```

**Zip file installation**

```bash
$ wget https://download.oracle.com/otn_software/linux/instantclient/1914000/instantclient-sqlplus-linux.x64-19.14.0.0.0dbru.zip

$ unzip instantclient-sqlplus-linux.x64-19.14.0.0.0dbru.zip

$ export LD_LIBRARY_PATH=/path/to/instantclient_19_4
```
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
Install OML4Py Client
Download, unzip, and view the optional arguments for the client.pl script

Download OML4Py client installation zip file from the Oracle Machine Learning for Python Downloads page on the Oracle Technology Network

Unzip the file

$ unzip oml4py-client-linux-x86_64-1.0.zip

$ ls client
  client.pl
  OML4PInstallShared.pm
  oml-1.0-cp39-cp39-linux_x86_64.whl
  oml4py.ver

$ perl -Iclient client/client.pl --help
Oracle Machine Learning for Python 1.0 Client.

Usage: client.pl [OPTION]...
Install, upgrade, or uninstall OML4P Client.

-i, --install       install or upgrade (default)
-u, --uninstall     uninstall
-y                   never prompt
--ask                interactive mode (default)
--no-embed          do not install embedded python functionality
--no-automl         do not install automl module
--no-deps           turn off dependencies checking
--target <dir>      install client into <dir>
Install OML4Py Client
Install the client using the client.pl script

$ perl -Iclient client/client.pl
Oracle Machine Learning for Python 1.0 Client.

Checking platform .................. Pass
Checking Python .................... Pass
Checking dependencies .............. Pass
Checking OML4P version ............. Pass

Current configuration
Python Version ................... 3.9.5
PYTHONHOME ....................... /opt/Python-3.9.5
Existing OML4P module version .... None
Operation ........................ Install/Upgrade
Proceed? [yes]
Processing ./client/oml-1.0-cp39-cp39-linux_x86_64.whl
Installing collected packages: oml
Successfully installed oml-1.0
Done

Verifying compatibility between the Linux, Python version, supporting packages with the OML4Py client version

Python configuration
Install/Upgrade or Uninstall

Installation completed successfully!
Installation Steps
Steps to install and configure the OML4Py client

1. Verify prerequisites
2. Install Python
3. Install open-source Python supporting packages
4. Install Oracle Client or Instant Client
5. Install the OML4Py Client components
6. Configure Oracle Wallet
OML4Py client Wallet architecture

Wallet Architecture

TNS_ADMIN=/path/to/wallet

OML4Py client

python

sqlplus /@TNS_ALIAS

oml.connect("", "", dsn="TNS_ALIAS")

tnsnames.ora

TNS_ALIAS ADB=[..]
TNS_ALIAS ADB_POOL=[..]
TNS_ALIAS ODB=[..]
TNS_ALIAS ODB_POOL=[..]

sqlnet.ora

WALLET_LOCATION=[..]
SQLNET.WALLET_OVERRIDE=TRUE

SERVER=POOLED
required for AutoML, MLX and parallel-enabled embedded Python execution

Oracle Autonomous Database or Oracle Database

database connection

client Wallet directory

AutoML, MLX and parallel-enabled embedded Python execution
To download the Oracle Wallet:

• Go to cloud.oracle.com

• Log in with the credentials provided when creating the Oracle Cloud account

• Select the Autonomous Database, followed by the DB Connection
Autonomous Database Wallet Credentials

mkstore -wrl wallet directory -createCredential tns alias username password

Assign Wallet Credentials for ADB medium service level

mkstore -wrl /path/to/wallet -createCredential myadb_medium OMLUSER Welcome12345

mkstore -wrl /path/to/wallet -createCredential myadb_medium_pool OMLUSER Welcome12345

repeat steps for service levels: low, low_pool, high, high_pool
Oracle Database Wallet Credentials

mkstore -wrl wallet directory -createCredential ODB tns alias username password

Assign Wallet Credentials

mkstore -wrl /path/to/wallet -createCredential

Client Wallet directory

myodb pyquser pyquser

ODB tns schema name

password

export TNS_ADMIN=/path/to/wallet

Client Wallet directory

myodb_pool pyquser pyquser

ODB tns alias

schema name

password
Wallet Configuration
Add server pool connection to tnsnames.ora for Autonomous Database

```

```

Repeat for service levels low, low_pool, high, high_pool
Optionally add proxy server address and port number
**Wallet Configuration**

Add tns alias and server pool connection to tnsnames.ora for Oracle Database

```plaintext
myodb=(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
  (https_proxy=www-proxy-address.com) (https_proxy_port=80)
  (HOST=myhost) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=myservicename)))

myodb_pool=(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
  (https_proxy=www-proxy-address.com) (https_proxy_port=80)
  (HOST=myhost) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=myservicename)
  (SERVER=POOLED)))
```

Use the same tnsnames.ora for Autonomous Database and Oracle Database
Optionally add proxy server address and port number
Wallet Configuration

Add Wallet location and override parameter to sqlnet.ora

```
WALLET_LOCATION=(SOURCE=(METHOD=file) (METHOD_DATA=(DIRECTORY="/path/to/wallet")))
SSL_SERVER_DN_MATCH=yes
SQLNET.WALLET_OVERRIDE=TRUE
SQLNET.USE_HTTPS_PROXY=on
```

$ export TNS_ADMIN=/path/to/wallet

Optionally enable proxy tunneling

To use the password credential stored in the Wallet when connecting to the database
Logging into the Database using Wallet

Connect to your databases from the universal client using wallet credential

Log into Autonomous Database

```python
$ sqlplus /@myadb_medium_pool
SQL> show user;
USER is "OMLUSER"

oml.connect(user="",
password="",
dsn="myadb_medium",
automl="myadb_medium_pool")

oml.isconnected()
>>>True
```

Log into Oracle Database

```python
$ sqlplus /@myodb_pool
SQL> show user;
USER is "PYQUSER"

oml.connect(user="",
password="",
dsn="myodb_medium",
automl="myodb_medium_pool")

oml.isconnected()
>>>True
```
Demo
Thank you

sherry.lamonica@oracle.com