Connecting Python and Node.js applications on Linux on Power Systems to Oracle databases

Using cx_Oracle and node-oracledb modules

Arun Sar
Krishna Harsha Voora
Mithun Hallikerehundi
Srirama Sharma

IBM Cognitive Systems
August 2018

Table of contents

Abstract

In the era of artificial intelligence (AI) and running microservices in containers, Python and Node.js are choices for building present generation applications. These applications may require connecting to various data sources for different types of the data. Because many customers use Oracle Database as their operational database, the cx_Oracle and node-oracledb modules provides an easy connectivity to an Oracle Database running on any platform. The validation work confirms that Python or Node.js application, using cx_Oracle or node-oracledb respectively, running on Linux® on IBM® Power Systems™ can connect to an Oracle Database running on IBM AIX®. Overall availability of these modules on Linux on Power® makes it easy for application developers to write the code and integrate. This enables the IBM PowerAI Enterprise platform to easily connect to an Oracle Database as one of the data sources.

This paper provides the commands and step-by-step guidance on how to establish connectivity to an Oracle Database using the node-oracledb and cx_Oracle modules.

Introduction

Linux on Power Systems provides an open and scalable infrastructure that is built to process a massive amount of data quickly, efficiently, and cost-effectively to formulate real-time actionable business insight. The Linux ecosystem alongside with legacy IBM AIX workloads provide a stronger value to the business that is looking for modernizing its operation using artificial intelligence, machine learning, and deep learning frameworks.

Python is widely used in the industry in numerous fields such as scientific packages, mathematical libraries, and web applications. JavaScript is one of the vital components of World Wide Web (WWW) along with HTML and CSS. Because it is lightweight and has attributes for ease of use, JavaScript has been widely considered in all the websites. Most of the web browsers have a dedicated engine to process JavaScript. Node.js is a server platform built on JavaScript and contains numerous JavaScript modules with features such as non-blocking I/O, which makes Node.js one of the most widely used platforms to develop scalable web and network applications. The cx_Oracle and node-oracledb modules come in handy and provide standard interface to connect applications developed using Python and Node.js with an Oracle Database.

IBM PowerAI is a software toolkit with deep learning frameworks and building block software designed to run on IBM's highest performing server. PowerAI makes deep learning, machine learning, and AI more accessible and more performant. By combining this software platform for deep learning with IBM Power Systems, enterprises can rapidly deploy a fully optimized and supported platform for AI with blazing performance. The PowerAI platform includes the most popular deep learning frameworks and their dependencies, and it is built for easy and rapid deployment.

Many Python and Node.js applications use deep learning or machine learning frameworks to create trained models. The accuracy of those models heavily depends on a range of vast data from different sources. One of the databases most widely used in the enterprise systems is the Oracle Database. This paper provides a brief overview and the steps to configure and use cx_Oracle and node-oracledb connectors.

Overview

cx_Oracle is a Python extension module that can be used to enable access to an Oracle Database. The recent version of cx_Oracle is 6 and has been tested to work with Python versions 2.7 and 3.4. cx_Oracle can be used with Oracle Database 11.2, 12.1, and 12.2 libraries.

cx_Oracle is an open source project licensed under BSD terms and is maintained by Oracle.

GitHub link for cx_Oracle is https://github.com/oracle/python-cx_Oracle.

The node-oracledb add-on for Node.js allows high performance Oracle Database applications which are developed using Node.js and Python. Node.js is an open source, cross-platform runtime environment for writing mid-tier and networking applications in JavaScript. The node-oracledb driver connects to an Oracle Database for fast and functional applications. It is an open source project with Apache 2.0 license. It is maintained by Oracle and is under active development.

GitHub link for node-oracledb https://github.com/oracle/node-oracledb.

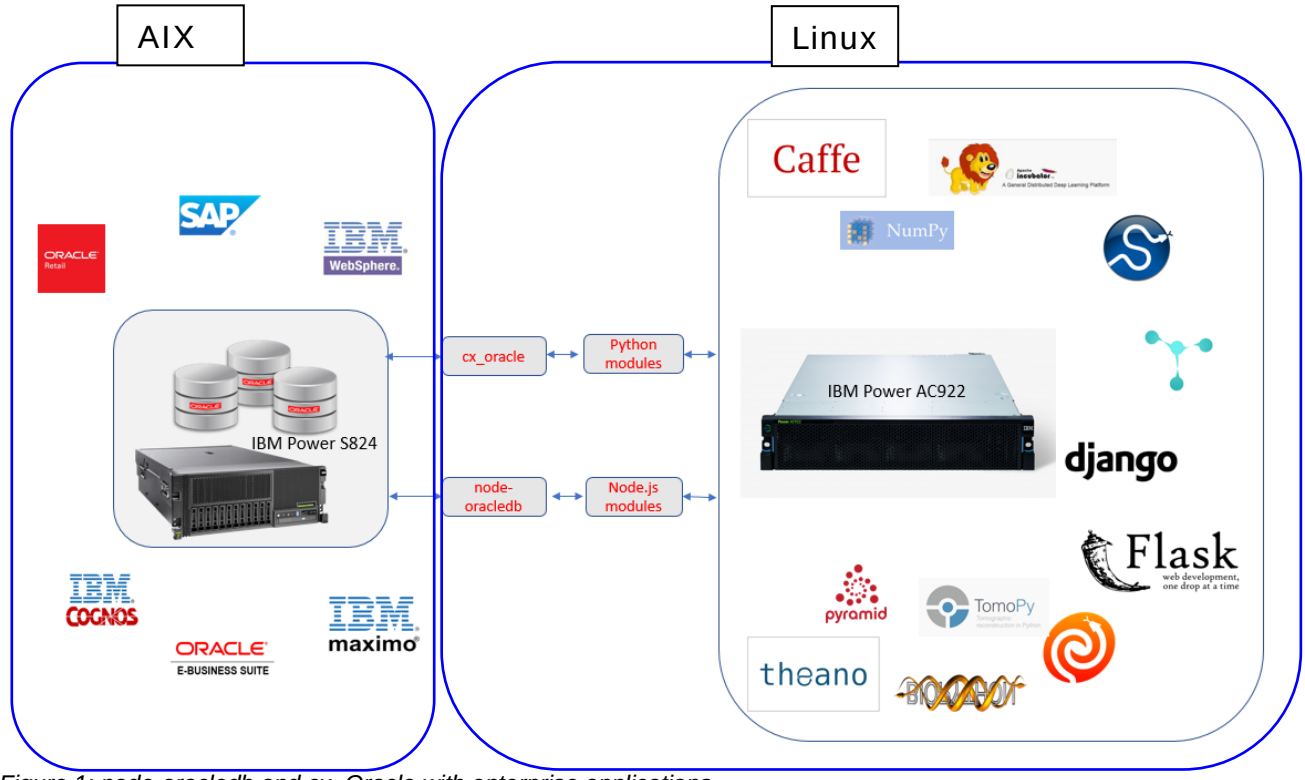

Figure 1: node-oracledb and cx_Oracle with enterprise applications

Figure 1 depicts IBM Power servers with an Oracle Database running on AIX operating systems. Many PowerAI frameworks, web applications, UI frameworks, mathematical, scientific and numerical libraries and applications can run on a powerful IBM POWER9™ processor-based servers, such as the IBM Power System model AC922.

cx_Oracle and node-oracledb, along with Oracle instant client provide built-in functions to connect the Oracle Database to applications built using Python and Node.js respectively, in preparing this document these were run on LoP.

Installation and configuration of cx_Oracle

This section provides the steps to install and configure cx_Oracle and node-oracledb plug-ins for applications running on Linux on Power Systems to access the Oracle Database.

Perform the following steps to download and configure python-pip and cx_oracle on UBUNTU 16_04 LTS ppc64le.

1. Install python pip. *pip* is a package management system used to install and manage software packages written in Python.

   ```
   $ apt-get install python-pip
   ```

2. Using the python pip install and upgrade cx_oracle package.

   ```
   $ python -m pip install cx_oracle –upgrade
   ```

3. Download the Oracle instant client available from the Oracle technology network at:

   http://www.oracle.com/technetwork/topics/linux-power-le-2835260.html

4. Create a directory and extract the instant client in the directory using the following commands.

   ```
   $ mkdir -p /opt/oracle
   $ unzip instantclient-basic-linux.leppc64.c64-12.2.0.1.0.zip
   ```

5. Create the necessary links and add the files to the library path using the following commands.

   ```
   $ sudo sh -c "echo /opt/oracle/instantclient_12_2 > /etc/ld.so.conf.d/oracle-instantclient.conf"
   $ sudo ldconfig
   $ export LD_LIBRARY_PATH=/opt/oracle/12.2/client64/lib:$LD_LIBRARY_PATH
   ```

6. Refer to the following URL to set up a schema for cx_oracle, examples demonstrate connectivity to the Oracle Database.

   https://oracle.github.io/python-cx_Oracle/

7. Navigate to the *sql* folder section under the sample section. Then, download and run the following SQL files:

   - DropSamples.sql
   - SampleEnv.sql
   - SetupSamples.sql

8. Download any sample copy (for example sampleEnv.py) to the folder where the SQL files are present, modify the parameters in the sampleEnv.py file as per the Oracle instance details and run the example file.

```
$ python Query.py
Get all rows via iterator
(1, 'Anthony')
(2, 'Barbie')
(3, 'Chris')
(4, 'Dazza')
(5, 'Erin')
Query one row at a time
(1, 'Anthony')
(2, 'Barbie')
Fetch many rows
[(1, 'Anthony'), (2, 'Barbie'), (3, 'Chris')]
```

Important connection parameters

Below is the connect function in the cx_Oracle package used to describe connection parameters.

```
connection = cx_Oracle.connect('user_name/password@IP_address: port/servicename')
```

where:

- `user_name` is the database user name
- `password` is the database user password
- `IP_address` is the host IP address
- `port` is the port at which the Oracle database is listening
- `servicename` depicts the name of the Oracle database service

Steps 1 to 8 under topic Installation and configuration of cx_oracle could be followed with the `yum install` command in place of apt-get install to download and configure the dependent packages on Red Hat Enterprise Linux ppcle64.

Linux kernel asynchronous I/O access library (*libaio1*) is a prerequisite package for cx_Oracle to work. If not available, you need to install it using the following command:

```
$ apt-get install libaio1
```

Installation and configuration of node-oracledb

You need to perform the following steps to configure and use node-oracledb on Ubuntu 16.04 LTS ppc64le:

1. Download the Oracle instant client available from the Oracle technology network at:

http://www.oracle.com/technetwork/topics/linux-power-le-2835260.html

2. Create a directory and extract the instant client in the directory using the following commands:

```
$ mkdir -p /opt/oracle
$ unzip instantclient-basic-linux.leppc64.c64-12.2.0.1.0.zip
```

3. Create the necessary links and add the files to the library path using the following commands:

```
$ sudo sh -c "echo /opt/oracle/instantclient_12_2 > /etc/ld.so.conf.d/oracle-instantclient.conf"
$ sudo ldconfig
$ export LD_LIBRARY_PATH=/opt/oracle/12.2/client64/lib:$LD_LIBRARY_PATH
```

4. Install the build essentials and SSL development libraries using the following commands:

```
$ sudo apt-get install build-essential
$ sudo apt-get install libssl-dev
```

5. You need to use the node version manager command (`nvm`) to download and install node.js. Run the following command to download nvm.

```
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

6. Close the current terminal and then open a new terminal.
7. Run the `-v nvm` command. The output displays the current nvm version, and nvm fetches the required node packages.

```
$ command -v nvm
```

8. Download and install Long term support version of node.

```
$ nvm install 8.9.4
```

9. The node version manager command will install node v8.9.4

```
$ npm install oracle/node-oracledb#v2.1.2
make: Leaving directory '/root/cx_examples/node_modules/oracledb/build'
npm WARN saveError ENOENT: no such file or directory, open '/root/cx_examples/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/root/cx_examples/package.json'
npm WARN cx_examples No description
npm WARN cx_examples No repository field.
npm WARN cx_examples No README data
npm WARN cx_examples No license field.
        + oracledb@2.1.2
added 2 packages in 62.078s
```

10. Download the examples from: https://github.com/oracle/node-oracledb/tree/master/examples

11. Download the demo.sql file.
12. Run demo.sql on Oracle Database to set up the basic schema.
13. Set up the connection string in the dbconfig.js file.
14. Run one of the example files (in this case, date.js) in the downloaded directory.

```
$ npm install async –save
$ node date.js
```

```
Inserting JavaScript date: Mon Jun 04 2018 17:48:07 GMT+0530 (IST)

Rows inserted: 1

Query Results:

[ [ 2018-06-04T12:18:07.077Z, 2018-06-04T12:18:07.000Z ] ]

Result Manipulation in JavaScript:

2018-06-09T12:18:07.077Z

2018-05-30T12:18:07.000Z

Altering session time zone

Query Results:

[ [ 2018-06-04T17:48:07.077Z, 2018-06-04T17:48:07.000Z ] ]

Result Manipulation in JavaScript:

2018-06-09T17:48:07.077Z

2018-05-30T17:48:07.000Z
```

Summary

The cx_Oracle module provides a wide range of in-built functionalities which are available out of the box and work with Oracle Database instant clients. Because most of the modern applications make use of Python, which is adopted due to its support for large third-party modules, readability and user-friendly data structures it is very important for these applications to interact with enterprise databases such as the Oracle Database.

Node.js is an event-driven, lightweight, extremely efficient, and non-blocking I/O package. With npm package manager (having a large open source ecosystem), Node.js along with Linux on Power Systems becomes the primary choice for Power AI, Internet of Things (IoT), machine learning, and applications in scientific and deep learning frameworks because of its enhanced performance, virtualization, and resilience.

Because cx_Oracle and node-oracledb are both widely used with open standards, the steps described in this paper can be used with all the applications using python and Node.js applications on Linux on Power Systems connecting to the Oracle Database.

Resources

The following websites provide useful references to supplement the information contained in this paper:

- Why Linux on Power?
  https://www.ibm.com/developerworks/library/l-linux-on-power/index.html

- node-oracledb 2.3 documentation
  http://oracle.github.io/node-oracledb/doc/api.html

- cx_Oracle documentation
  http://cx-oracle.readthedocs.io/en/latest/

- IBM Techdocs
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs

- IBM Power AI Enterprise
  https://www.ibm.com/in-en/marketplace/deep-learning-platform

- IBM Power Systems
  https://www.ibm.com/it-infrastructure/power

- IBM Redbooks®
  www.redbooks.ibm.com/

About the authors

Arun Sar

Arun Sar is a Senior Technical Staff Member in ISV organization of IBM Cognitive Systems business unit. As Open Source Database Technology Manager, he owns the responsibilities of building database ecosystem on IBM Power Systems. He can be reached at asar@us.ibm.com.

Krishna Harsha Voora

Krishna Harsha Voora has 6 years of experience with IBM as an administrator, a developer, and a consultant in UNIX® and Linux operating systems. He is actively contributing to Hyperledger Fabric on Power.

He contributes to Linux on Power ecosystem by porting the required software stack. He actively works on Linux on Power ISV Engagement.

Mithun Hallikerehundi

Mithun has 7 years of IT experience with Master of technology degree in computer science from Visveswaraya technological University and has been with IBM for the past 3 years as a Technical consultant. He has experience in retail, aerospace, and defense domains. He is currently working as an Oracle ISV consultant performing Oracle product certifications on IBM AIX. His expertise is in Java™ and is also involved in helping ISVs to port their solutions to Linux on Power. His is interested in open source databases and Power AI.

Srirama Sharma

Srirama Sharma has overall IT experience of more than 14 years and has been with IBM from last 7 years as Technical Consultant leading ISV and Open Source ecosystem enablement at IBM Systems Development Lab (ISDL). He is the technical lead for the *Blockchain on Power* mission globally working closely with Hyperledger Fabric community and evangelizes blockchain technology in various forums, conferences, and meet-ups. He is also involved in building and enabling open source-based solutions supporting ISV and customer opportunities across different technologies such as private cloud and cognitive. He is also a key player in the Financial Software and Systems industry.

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, ibm.com, AIX, Power, Power Systems and POWER9 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

The information provided in this document is distributed “AS IS” without any warranty, either express or implied.

The information in this document may include technical inaccuracies or typographical errors.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.