PostgreSQL Foreign Data Wrapper development with Python.

Toufeeq Ockards
@tockards (Twitter handle, but I mostly tweet music videos)

PostgreSQL

- The world’s most advanced open source database.

Foreign Data Wrapper (FDW)

- SQL standard that deals with how a database management system can integrate data stored outside the database.
- **SQL/MED**

Python

- The world’s most advanced open source language.

Foreign/Federated Table

- **Storage engine** which allows a user to **create a table that is a local representation of a foreign (remote) table**

- a **transparent access method** for **external data**

Data

- Flat Files e.g CSV files, Log Files, /etc/passwd(kappa) 
- RSS feeds, Emails, Websites, APIs
- Directories, file shares
- Other Databases, Services

You would like

```
postgres=# select some_data from external_data_source;
```

Multicorn FDW

- PostgreSQL Extension.
- Allows you to Write FDW's in python.

MULTICORN

You have

```
Year,Make,Model,Length
1997,Ford,E350,2.34
2000,Mercury,Cougar,2.38
```

- A csv file sitting in /tmp (because mongo)

CSV Example (Python)

```python
from . import ForeignDataWrapper
from .utils import log_to_postgres
from logging import WARNING
import csv


class CsvFdw(ForeignDataWrapper):
    """A foreign data wrapper for accessing csv files.

    Valid options:
        - filename : full path to the csv file, which must be readable
          by the user running postgresql (usually postgres)
        - delimiter : the delimiter used between fields.
          Default: ","
    """

    def __init__(self, fdw_options, fdw_columns):
        super(CsvFdw, self).__init__(fdw_options, fdw_columns)
        self.filename = fdw_options["filename"]
        self.delimiter = fdw_options.get("delimiter", ",")
        self.quotechar = fdw_options.get("quotechar", '"')
        self.skip_header = int(fdw_options.get('skip_header', 0))
        self.columns = fdw_columns
```

CSV Example (Python)

```
    def execute(self, quals, columns):
        with open(self.filename) as stream:
            reader = csv.reader(stream, delimiter=self.delimiter)
            count = 0
            checked = False
            for line in reader:
                if count >= self.skip_header:
                    if not checked:
                        # On first iteration, check if the lines are of the
                        # appropriate length
                        checked = True
                        if len(line) > len(self.columns):
                            log_to_postgres("There are more columns than "
                                            "defined in the table", WARNING)
                        if len(line) < len(self.columns):
                            log_to_postgres("There are less columns than "
                                            "defined in the table", WARNING)
                    yield line[:len(self.columns)]
                count += 1
```

CSV Example (SQL)

```
CREATE SERVER csv_srv foreign data wrapper multicorn options (
   wrapper 'mycode.csvfdw.CsvFdw'
);
```

CSV Example (SQL)

```
create foreign table csvtest (
        year numeric,
        make character varying,
        model character varying,
        length numeric
) server csv_srv options (
        filename '/tmp/test.csv',
        skip_header '1',
        delimiter ',');
```

CSV Example (SQL)

```
select * from csvtest;
```

```
 year |  make   | model  | length
------+---------+--------+--------
 1997 | Ford    | E350   |   2.34
 2000 | Mercury | Cougar |   2.38
(2 lines)
```

SLOC Python 44
SQL 3 Statements

What else

- Use your ORM, BI Tool, Kitchen Sink above it.
- PostgreSQL 9.3 +
  - Insert, Update, Delete
- SQLAlchemy FDW
- LDAP FDW
- Docker FDW
- PGOSQUERY (like facebook osquery(see what I did with the like))
- Redis-FDW
- Mongo-FDW
- https://wiki.postgresql.org/wiki/Foreign_data_wrappers

SKA AFRICA
SQUARE KILOMETRE ARRAY

SKA SA Control and Monitoring Use Case

- We currently archive our sensor data to HDF5 Files and have legacy archived files in CCSV Format
- Query these Files via a server and use a PostgreSQL FDW to represent it in a table.

- **Time Series Sensor Data.**
  - sensor_name, ***sample_ts**, values_ts, status, value

SKA AFRICA
SQUARE KILOMETRE ARRAY

What we liked

- SQL interface remains the same.
- One interface for querying multiple data sources.
  - HDF5 File Format.
  - Inhouse CCSV Format
- The power of SQL.

SKA AFRICA
SQUARE KILOMETRE ARRAY

What ~~we~~ I did not like.

- Writing Unit Tests for the FDW.

SKA AFRICA
SQUARE KILOMETRE ARRAY

Future

- CEPH RADOS(distributed object store)
- SQL interface remains the same.

SKA AFRICA
SQUARE KILOMETRE ARRAY

Requirements

- PostgreSQL 9.1+
- PostgreSQL Development Packages
- Python Development Packages
- Python 2.7 >= or Python 3.3 or default python

SKA AFRICA
SQUARE KILOMETRE ARRAY

We are hiring.

- http://www.ska.ac.za/vacancies/