Bitcoin RDF Model in AllegroGraph

For more examples visit – https://github.com/franzinc/agraph-examples

Introduction
This example demonstrates an RDF model for Bitcoin chain data as well as a Python tool to pull the data from a Bitcoin node into an instance of an AllegroGraph graph database. The model description itself can be found in the Turtle file model.ttl.

The following Turtle example demonstrates how this RDF model can be used to represent complete chain entities (given example is a genesis block – the first block in the mainnet Bitcoin chain; script strings omitted for brevity):

```
@prefix : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
@prefix btc: <bitcoin://>

btc:blk0
:height 0;
:hash "000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f";
:time 1231006505;
:version 1;
:transaction btc:4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b.

btc:4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
:lockTime 0;
:input [:unlockScript "…".];
```

```
:output [:amount 5000000000; :lockScript "…".].
```

Setup

The following examples assume AllegroGraph triple store and assume it is already installed and running on the target machine. The following AG instance settings settings are assumed as well:

```
host: localhost (default);
port: 10035 (default);
username: aguser;
password: agpassword.
```

We also assume the following bitcoind settings:

```
host: localhost (default);
port: 8332 (default);
username: btcuser;
password: btcpassword.
```

First, install the tool by cloning this repository, setting up virtual environment and installing the dependencies:

```
git clone http://github.com/franzinc/agraph-examples
cd agraph-examples/data/bitcoin
python3 -m venv .
source ./bin/activate
pip3 install -r requirements.txt
```

The following command starts the process of loading bitcoin chain data into an AG repository named bitcoin using 4 loader processes:

```
./convert.py \
–source=http://btcuser:btcpassword@localhost:8332 \
–destination=http://aguser:agpassword@localhost:10035 \
–name=bitcoin \
–workers=4 \
–clear
```

Example queries

Following are the examples of using SPARQL to extract different information about block data:

- number of known blocks:

```
PREFIX : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
SELECT (COUNT(*) AS ?count) WHERE { ?b a btcm:Block. }
```

- total number of transactions:

```
PREFIX : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
SELECT (COUNT(*) AS ?count) WHERE { ?tx a btcm:Transaction. }
```

- transaction in block 400:

```
PREFIX : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
SELECT ?txid
WHERE {
?b a :Block.
?b :height “570001”^^xsd:int.
?b :transaction ?tx.
?tx :txid ?txid.
}
```

- transactions sending more than 1000 BTC:

```
PREFIX : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
SELECT ?tx
WHERE {
?b a :Block.
```

```
?b :transaction ?tx.
?tx :output ?out.
?out :amount ?amt.
}
GROUP BY ?tx
HAVING (SUM(?amt) > 100000000000)
```

- transactions sending BTC to Pirate Bay's address:

```
PREFIX : <https://raw.githubusercontent.com/franzinc/agraph-examples/master/data/bitcoin/model.ttl#>
SELECT ?tx
WHERE {
?tx :output ?out.
?out :lockScript ?s.
FILTER REGEX (?s, "<tpb address>").
}
```

Going Beyond Blockchain with Directed Acyclic Graphs (DAG)

Crypto Slate – January 2019

by Dr. Jans Aasman, CEO, Franz Inc.

If organizations could only augment blockchain's strengths—its immutability, security, and decentralization—while addressing its latency and scalability issues, it could become the vaunted enterprise tool it was initially intended. That day will soon come courtesy of Directed Acyclic Graphs (DAGs).

Blockchain’s premise is straightforward, utilitarian, and more lucrative than that of any other new technology to recently emerge. This distributed ledger system promises near real-time updates of transactions between remote parties for trustworthy, impenetrable peer-to-peer networks, eliminating the need (and expense) of middlemen.

Read the full article at Crypto Slate.