StackOverflow data to PostgreSQL

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 14 June 2009

https://www.bortzmeyer.org/stackoverflow-to-postgresql.html

The great social site Stack Overflow <https://www.bortzmeyer.org/stack-overflow.html> just announced <http://blog.stackoverflow.com/2009/06/stack-overflow-creative-commons-data-du> the publication of its entire database under a Creative Commons free licence. I believe it is the first time such an important social networking site publishes its data, so it is a great day for data miners. In this small article, I will explain how I entered these data into a PostgreSQL database for easier mining. (The work was done on a Debian machine but it should work on any Unix.)

The original file is huge (200 megabytes today, and growing). To avoid killing the Stack Overflow servers, it is distributed in a peer-to-peer fashion with BitTorrent. I just downloaded the torrent file http://blog.stackoverflow.com/wp-content/uploads/so-export-2009-06.7z.torrent to my download directory and BitTorrent does the rest <https://www.bortzmeyer.org/screen-bittorrent.html>. I then extract the XML files with p7zip. Each XML file store a class of Stack Overflow objects :

— Users, today 88,558 (but many of them registered once but never came back afterwards),
— Posts, today 182,742 questions and 698,923 answers,
— Comments, today 705,085,
— Votes,
— Badges.

I then created one SQL table for each class. The complete DDL instructions are in file (en ligne sur https://www.bortzmeyer.org/files/so-create.sql). I can then create the database and its schema :

```
% createdb --encoding=UTF-8 so
% psql -f so-create.sql so
```

I am normally a fan of integrity constraints in databases. But, in the case of Stack Overflow, there are many obvious integrity constraints that I did **not** include because they are violated at least one, especially in old data (presumably during beta testing of the site). For instance, 18,239 users (20 %) has no name (see for instance <http://stackoverflow.com/users/57428>, the one with the highest reputation) and therefore I cannot write `name TEXT NOT NULL`.

Same problem with the accepted answer, some posts reference an answer which is not available (for instance post #202271 <http://stackoverflow.com/questions/202271> has an accepted answer in the XML file, #202526, which does not exist).

Once the database is set up, we just have to parse the XML files and to load them in the database. I choose the Python programming language and the ElementTree <http://effbot.org/zone/element-index.htm> XML library. I produce SQL files which uses the `COPY` instruction.

The Python program is available as (en ligne sur `https://www.bortzmeyer.org/files/so-so2postgresql.py`). To execute it, you just indicate the directory where the Stack Overflow XML files have been extracted. Then, you run PostgreSQL with the name of the produced SQL COPY file, and with the name of the database :

```
% python so-so2postgresql.py /where/the/files/are > so.sql
% psql -f so.sql so
```

This `so-so2postgresql.py` program requires a **lot** of memory, because it keeps the entire XML tree in memory (that is a weakness of the XML library used). Do not attempt to run it on a machine with less than eight gigabytes of physical RAM, swapping will make it awfully slow. You may also have to increase the available memory with `ulimit`.

Once the program is over, you can start studying the data :

```
so=>  SELECT avg(reputation)::INTEGER FROM Users;
 avg
-----
 183
(1 row)

so=> SELECT avg(reputation)::INTEGER FROM Users WHERE reputation > 1;
 avg
-----
 348
(1 row)
```

Users start with a reputation of 1. Ignoring these users, in the second SQL request, allows to compare only active users.

The first analysis produced with this database was an exploration of the "fastest gun in West" syndrome (available at <https://www.bortzmeyer.org/stack-overflow-short-tail.html>, in French).

Many people already posted on the subject of the Stack Overflow database. For instance :

`https://www.bortzmeyer.org/stackoverflow-to-postgresql.html`

— Data Mining the StackOverflow Database <http://sqlserverpedia.com/blog/sql-server-tutorial/data-mining-the-stackoverflow-database/> (SQL server, only as a video in this article but other articles <http://sqlserverpedia.com/wiki/StackOverflow_Question_Demographics_Query> are more text-friendly),
— What interesting stats can I obtain from the Stack Overflow data-dump ? <http://stackoverflow.com/questions/951056/what-interesting-stats-can-i-obtain-from-the-stack-overflow-dat with many interesting statistics,
— Stack Overflow : Up and Down Voting Pattern Analysis <http://lanai.dietpizza.ch/geekomatic/2009/06/09/1244565360000.html>,
— Statistics about tags, users and questions at StackOverflow, ServerFault and SuperUser <http://spwho2.com/Sites/StackOverflow/>,
— How to Import the StackOverflow XML into SQL Server <http://www.brentozar.com/archive/2009/06/how-to-import-the-stackoverflow-xml-into-sql-server/>, which focuses on the import into a database, like this article,
— A Python script to load StackOverflow data <http://www.rdbhost.com/downloads/import_so_data_sep09.py.txt> in the rdbhost <http://www.rdbhost.com/> system. Unlike my script, it uses SAX and not DOM and therefore can run with much less memory,
— StackOverflow question statistics <http://www.johndcook.com/blog/2009/04/15/stackoverflow-ques> (nice graphs).
— Franck Piochaud made a much improved version <https://github.com/badmonster-nc/stackoverflow_in_pg> (specially performance-wise) of my code.

---

https://www.bortzmeyer.org/stackoverflow-to-postgresql.html