Setting Up SNAP&CANAL

Ning Zhang #1, Yuanyuan Tian *2, Jignesh M. Patel #3

Computer Sciences Department, University of Wisconsin-Madison, USA

[1]nzhang@cs.wisc.edu [3]jignesh@cs.wisc.edu

* *IBM Almaden Research Center, USA*

[2]ytian@us.ibm.com

March 15, 2010

1 Installing Dependencies

In order to install the SNAP&CANAL, you have to make sure that PostgreSQL (`http://www.postgresql.org/`), GUESS (`http://graphexploration.cond.org/`), gcc (`http://gcc.gnu.org/`), and Java Version 6 (http://www.java.com/en/download/index.jsp) are correctly installed on your machine. PostgreSQL is used to store and index graph data; gcc is used for compiling the source codes; GUESS is used for visualizing the graph summarizations.

1.1 Installing PostgreSQL

For some Linux machines, PostgreSQL is installed by default. However, since SNAP&CANAL needs the libpq C library, but most default installation doesn't include this library, PostgreSQL needs to be installed from source code in your local directory.

1. **Download PostgreSQL** (version 8.1 or higher) from `http://www.postgresql.org/download/`.
2. **Install PostgreSQL.** If you are installing from source code, follow the instructions on `http://www.postgresql.org/docs/8.1/interactive/installation.html`. By default, the PostgreSQL executables are installed in directory */usr/local/pgsql/bin*, PostgreSQL libraries are in directory */usr/local/pgsql/lib*, and the data directory is */usr/local/pgsql/data*. However, we strongly suggest that a local directory is set as the data directory. As SNAP&CANAL uses PostgreSQL libraries to communicate to the database, please set LD_LIBRARY_PATH properly (Refer to `http://www.postgresql.org/docs/8.1/interactive/install-post.html` for detail). Besides, please set the PATH environment variable to be able to conveniently run any commands provided by PostgreSQL. Please refer to `http://www.linuxheadquarters.com/howto/basic/path.shtml` for how to set PATH variable.
3. **Change PostgreSQL configuration.**

   The default PostgreSQL buffer pool size is 8MB. In addition, the amount of memory to be used by internal sort operations and hash tables before switching to temporary disk files is set to 1MB by default. These values are too small for even a moderate database. So, the PostgreSQL configuration must be changed to improve the database efficiency.

One way of changing the configuration is to edit the configuration file *postgresql.conf*, which is normally kept in the data directory (*/usr/local/pgsql/data* by default). For details on how to change the configuration, please refer to `http://www.postgresql.org/docs/8.1/interactive/runtime-config.html`. Please make the following changes to the *postgresql.conf*: assuming that you have a 2GB memory, set the shared_buffers to at least 512MB (shared_buffers=65536), the work_mem to at least 128MB (work_mem=131072) and the maintenance_work_mem to at least 256MB (maintenance_work_mem=262144). (Please refer to `http://www.postgresql.org/docs/8.1/interactive/runtime-config-resource.html#RUNTIME-CONFIG-RESOURCE-MEMORY` for the description of each parameter.) Note that these parameters are commented by default in *postgresql.conf*, please first remove the comment mark #.

```
shared_buffers=65536

work_mem=131072

maintenance_work_mem=262144
```

Increasing the shared_buffers parameter may cause PostgreSQL to request more System V shared memory than your operating system's default configuration allows. One way to change the restriction on the System V shared memory is to edit the */etc/sysctl.conf* file by adding the following two lines (assume that we want to set the shared memory to 671088640 bytes):

```
kernel.shmall=671088640

kernel.shmmax=671088640
```

Then run the command "`sysctl -p`". Note that you need root access to change the System V shared memory size. Please refer to `http://www.postgresql.org/docs/8.1/interactive/kernel-resources.html#SYSVIPC` for more details.

1.2 Installing gcc

Most linux machines have installed gcc. To check, type "g++ -v" in the terminal. If the output contains some configuration and version information, then it shows that gcc has been installed in the machine. Otherwise, download the current version of gcc from `http://gcc.gnu.org/` and follow the instruction in the release package to install it.

1.3 Installing Java 6

Java 6 can be downloaded from `http://www.java.com/en/download/index.jsp`. Please follow the instructions in the release package to install Java 6. Please also set the PATH and CLASSPATH properly.

1.4 Installing GUESS

GUESS can be downloaded from `http://graphexploration.cond.org/download.html`. Please follow the instructions in README.TXT in the release package and tutorial/walk through in the manual (`http://guess.wikispot.org/manual`) to install GUESS. There is also a new tutorial for Mac users (`http://graphexploration.cond.org/MacGUESSinstall.pdf`).

Before visualizing graph summarizations, first modify the GUESS_HOME and GUESS_LIB variables in `guess.sh`. GUESS_HOME needs to point to where you have GUESS installed and GUESS_LIB is the path of GUESS_HOME/lib.

To visualize the graph summarization, first enter into the GUESS_HOME folder, and type "./guess YourGraph.gdf" in the terminal, where "YourGraph.gdf" is the input file of graph summarization. Or just type "./guess" and select "Load GDF/GraphML" button in the pop-up window and then select the location of the input file.

2 Installing SNAP&CANAL

2.1 Compile SNAP&CANAL

The SNAP&CANAL directory has 3 sub-directories. Directory `toolkit` contains the main source codes. Directory `data` contains the data used in SNAP&CANAL. And directory `guess_scripts` contains the scripts for visualizing graphs in GUESS.

2.1.1 Edit Makefile

First change directory to the `toolkit` subdirectory (`cd toolkit`). Before compiling the SNAP&CANAL code, the Makefile should be edited first.

Please change the values of the variables PG_HOMEDIR at the top of Makefile to the home directory of PostgreSQL. For example, if PostgreSQL is installed under /home/user/pgsql, then the variables PG_HOMEDIR is set as
`PG_HOMEDIR = /home/user/pgsql`.

Also change the CC3 variable to the full path of g++. For example, if g++ is installed under directory /usr/bin. Then set
`CC3 = /usr/bin/g++`.

2.1.2 Compile SNAP&CANAL

First run the command "`make clean`". And then run the command "`make`" to generate the binaries. After compilation, the following executables are generated:

bulkload A program to load a list of graphs into the database.

querysnap SNAP executable.

canal CANAL executable.

3 How to Use SNAP&CANAL

3.1 Starting PostgreSQL Server

Before running SNAP&CANAL, please make sure that PostgreSQL is already running Command pg_ctl can be used to start or stop the PostgreSQL database server (`http://www.postgresql.org/docs/7.3/static/app-pg-ctl.html`).

3.2 Creating the Database

Use `createdb` command to create a database. All the data will be stored in this database.
`createdb -E LATIN1 demo`

3.3 Loading Data into the Database

Please run the loaddata.sh script to load the data into the database.
`./loaddata.sh`

This script loads the following two datasets:

dblp_num The Database coauthorship graph with numerical attribute. This dataset is only used for CANAL.

dblp The Database coauthorship graph with 2 cutoffs. This dataset is only used for SNAP.

Also, editing "dblp.list" file and changing the first line to: `../data/dblp_canal_3.gdf` will load the database coauthorship graph with 3 cutoffs, and so on so forth.

3.4 Configuring SNAP&CANAL

The file `periscope.config` contains the runtime configuration of SNAP&CANAL. Before executing any programs in SNAP&CANAL, first modify the file `periscope.config`. Change the DB_USER variable to your username in PostgreSQL (the default username is postgres). And change the TEMP_DIR variable to the absolute directory of a temporary directory. The descriptions of these two variables as well as other variables are as follows:

DB_NAME The postgreSQL database name where the data are stored. The default name is demo.

DB_USER The user who has the access to the database.

DB_PWD The password for the database user.

TEMP_DIR The **absolute** path of a temporary directory. This temporary directory is used to store some temporary files generated during the loading process. The SNAP&CANAL loading program does not load data into the database by inserting one tuple at a time. Instead, SNAP&CANAL first store all the tuples in temporary files, and bulkload all the data into the database. Bulkloading dramatically decreases the loading time. The temporary directory is used for these intermediate files. Make sure that the temporary directory has enough space (has least several GB and 50 times of input graph data size) to hold all the temporary files. These files are deleted after the loading process. The users can use */tmp* for the temporary directory.

3.5 Description of SNAP&CANAL Commands

3.5.1 *bulkload*: Load a List of Graphs into the Database

The graphs will be loaded into the PostgreSQL database named by the DB_NAME parameter in the configuration file. Make sure that the database DB_NAME is already created by the user DB_USER and the database DB_NAME is empty (you can use the dropdb command first and then createdb command in PostgreSQL to create an empty database). The executable for loading graphs is called *bulkload*. The synopsis of the usage is:

bulkload `[Configuration File] [List of Graphs] [Orthology File Name] [Dataset Name] [Index Choice] [Directed Graph]`

The descriptions of the parameters are as follows:

Configuration File This parameter is the name of the SNAP&CANAL configuration file.

List of Graphs This parameter is the name of a file listing all the graphs to be loaded. This file contains the paths of all the graph files, each line for one graph file. The graph files must conform to the GUESS .gdf format.

Orthology File Name The file name that contains the list of orthologous groups, the loading or querying algorithms will assign an integer id to each group.

Dataset Name The name of the dataset. Make sure this dataset is NOT already in the database.

Index Choice Default value is 0, do not change.

Directed Graph 0 for undirected graph; 1 for directed graph.

3.5.2 *querysnap*: K-SNAP Queries

The following is how to use the `querysnap` script:

querysnap `[Configuration File] [Dataset Name] [Graph ID] [Attribute Name] [Edge Type] [Resolution] [Output Directory]`

For example: `./querysnap periscope.config dblp 1 prolific coauthor 4 /home/nzhang/`

The descriptions of the parameters are as follows:

Configuration File This parameter is the name of the SNAP&CANAL configuration file.

Dataset Name The name of the dataset. Make sure this dataset is in the database, and is consistent with DB_NAME in `periscope.config`.

Graph ID The ID of the graph in the dataset that you want to generate a summary from.

Attribute Name The name of user selected attribute, based on which the summary will be generated.

Edge Type The type of the edges, based on which the summary will be generated.

Resolution The resolution of the summary (the number of groups in the summary).

Output Directory The directory where the summary file will be written to.

3.5.3 *canal*: CANAL algorithm

The following is how to use the `canal` script:

canal `[Configuration File] [Dataset Name] [Graph ID] [Attribute Name] [Edge Type] [Number of Cutoffs]`

For example, `./canal periscope.config dblp_num 1 prolific coauthor 3`

The descriptions of the parameters are as follows:

Configuration File This parameter is the name of the SNAP&CANAL configuration file.

Dataset Name The name of the dataset. Make sure this dataset is in the database, and is consistent with DB_NAME in `periscope.config`.

Graph ID The ID of the graph in the dataset.

Attribute Name The name of **numerical** attribute, based on which the cutoffs will be generated.

Edge Type The type of the edges.

Number of Cutoffs The number of cutoffs on the numerical attribute domain (e.g. 2 cutoffs).

4 GUESS scripts

Some scripts were written for visualizing graph summarizations. To use the scripts, first load the file of graph summarization into GUESS, click "File" and then click "Run Script". Select the corresponding script for the specific dataset. The descriptions of the scripts are as follows:

dblp_canal_3.py visualize all graph summarizations of DBLP dataset based on 2 cutoffs.

dblp_canal_3.py visualize all graph summarizations of DBLP dataset based on 3 cutoffs.

wiki.py visualize all graph summarizations of wikipedia dataset.

5 Dealing with Large Graphs

If the graph to be summarized is very large, we suggest keeping some auxiliary information in memory to speed up the process of summarizing graph. To use this feature, compile the program with LARGE_DATASET (in `summarizer.h`) switched **on**, and the value of MEM_PREFETCH in (in `summarizer.h`) should be equal to the number of nodes in the graph. If there is a segmentation fault, it means the machine memory is not big enough to hold the auxiliary information. (e.g. in our testing, 8GB memory can easily deal with a graph with 200,000 nodes.) In this release package, we simply provide the most straightforward strategy to store the auxiliary information (e.g. use a two-dimensional array), and there must be some more efficient methods that use less memory to do the same jobs.

6 Frequently Asked Questions

Those questions are frequently seen when first using SNAP&CANAL:

Question 1: error occurs when execute select nodeID, attrValue from dblp_num_Node where graphID=1 and attrName='prolific' order by attrValue; command failed: ERROR: relation "dblp_num_node" does not exist

Solution: check DB_NAME in `periscope.config` to see if it is consistent with [Dataset Name] in your input.

Question 2: Error: wrong edgetype or graph id

Solution: check [Edge Type] in your input to see if it is consistent with the "type" attribute in your *.gdf data file. Also, check [Graph ID] in your input. By default, [Graph ID] is 1.

Question 3: Error: wrong attribute name or graph id

Solution: check [Attribute Name] in your input to see if it is consistent with the node attribute (the first line in your *.gdf data file). Also, check [Graph ID] in your input. By default, [Graph ID] is 1.

References

[1] Y. Tian, J. M. Patel, V. Nair, S. Martini, and M. Kretzler. Periscope/GQ: A graph querying toolkit. In *VLDB*, 2008.

[2] Y. Tian, R. Hankins, J. M. Patel Efficient Aggregation for Graph Summarization. *SIGMOD*, 567-580, 2008.

[3] N. Zhang, Y. Tian, J. M. Patel Discovery-Driven Graph Summarization. *ICDE*, 2010.