/CopyRight 2016 SouthEast University, Yongfeng Huang cz_hyf@163.com)*/**

1. -----environment config----------

1.1 ---------eclipse config----------

elipse version：eclipse juno
jdk version: jdk1.6.0_45
note: due to the fact that CjegSim only generates one critical journey at a time point (different critical journeys exist at the same time point), for different jdk version (or different runtime environment), the critical journey cluster generated by CjegSim may be a little difference.

1.2 ---------database config----------

after cjegSim running, binary cjeg file will be output to disk (dataset directory).
As is efficient for calculating, however, is not straightforward to analyze.
We can output binary cjeg file to database for further analysis and thus mysql is needed.

We use xampp 1.7.3 which contains (apache 2.2.14, mysql 5.1.41, phpmyadmin 3.2.4-a web-based mysql manager)

2. ---------CjegSim config and run--------

2.1 --------config file: cjeg.txt----------

```
{
 "storename": "reality",   //dataset directory which contacts data can be read from and
                           //cjeg graph file can be written to
 "crawdadfile": "reality_all.txt",          //file name in dataset from which reads contacts.
 "idmap": "idmap.txt",                     //relationship between node name and node id,
                                          //when wr_flag is set to 0,
                           //node name corresponding to node id is displayed (for debug)
 "tau": 1,                 //edge traversal time
 "eta": 1,                 //reserve this settting (haven't use in our experiment)
 "tracename": "cjeg",      //after running, a new directory
                           //named $tracename$_$tau$ will be generated
                          //in directory $storename$ to store cjeg graph file
 "wr_flag": 1,             //0 just for debugging, thus cjeg graph will be outputted to screen.
 "debug": 0,               // if 1, more debug information will be output to screen.
 "loglevel": 2000       //debug loglevel only the information >= loglevel in code will be
                          //output to screen.
 "monitormemory": 0   //if you want to monitor memory used in the process of cjegSim
                          //running, setting this to 1, which will slow the running of cjegSim
}
```

2.2 ---------how to run cjegSim----------

steps:

(1) use eclipse to import project as an Existing java project into workspace
all source code is in directory src

(2) config cjeg.txt

(3) run CjegApp.java (located in package app, main function) as a java application

after running, "cjeg_1" (1 is the edge traversal time configured in cjeg.txt) directory will be generated in "storename" directory ("storename" is also configured in cjeg.txt such as "reality"). Contents in directory "cjeg_1" are the output of critical journey evolving graphs.

(4) **TestCjegRead.java** can be used to get all critical journey clusters between any given two nodes from "cjeg_1" generate in step (3) directly.

(5) create a database in mysql (database name must be in accordance with the "storename" in cjeg.txt, such as reality),
For example: you can execute a sql query "create database reality;" by phpmyadmin

(6) **CjegOutToMysql.java** (located in package tool) can be used to read the cluster from "cjeg_1" generated in step (3) and output clusters to mysql database created in step (5). This step may be slow, please wait for some time. Note: username and password in CjegOutToMysql.java must be set correctly before running. If you use xampp and haven't change default setting. username is "root" and password is null ("").

(7) now a new table "cjeg_crjourneyegmap_1" have been created in database named $storename$ such as "reality" and all critical journey cluster have been inserted into it.

When using seu dataset (about 1000 nodes), default memory may be not enough.
You could add memory as the following steps:

eclipse menu: window->preferences->java-installed JREs (jdk1.6.0_45)->Edit
input -Xms512m –Xmx2048m in "Default VM arguments"

3. ----------cjegSim architechture----------
see cjegSim.pdf in this directory

4. ----------directory description----------

config.txt config file for CjegSim
src: java source code

lib: java libarary used in cjegSim
configexample: some config file example, use it to replace cjeg.txt if we do experiment on different dataset.

reality: reality dataset and corresponding cjeg file will be outputted into this directory
rollernet: rollernet dataset and corresponding cjeg file will be outputted into this directory
tmg: tmg dataset and corresponding cjeg file will be outputted into this directory
seu: seu dataset and corresponding cjeg file will be outputted into this directory

5. ----------package description----------

package app:

main class, some class to parse config file

package core:

all core class，including journey, critical journey class, Node, vector, etc.
all view events handler is in Node Class.

package input

all class associated with input, including many events classes and eventlistener classes

package output

all class assocaiated with outputting CJEG
CjegTrace file refer to the DITL library in paper "Temporal Reachability Graph" which including three files: info, index(cluster offset in trace), trace(critical journey clusters)
info：text file, including basic information such as the number of clusters。
index：binary file, index for trace, every 1000 cluster will have an index in this file
trace file: binary file, including critical journey clusters.

package tool

CjegOutToMysql.java: read the trace file of CJEG and output it to mysql for analysis.
FindFormostJourney.java: find foremost journey from a start time, it is an implementation of paper "COMPUTING SHORTEST, FASTEST,AND FOREMOST JOURNEYS IN DYNAMIC NETWORKS" (for verify CJEG)

package test

TestCjegRead.java：get all critical journey clusters between any two nodes
TestCjegReadAtTime.java：the first critical journey cluster from a start time according to departure time, we can use it to find a foremost journey too.

Also we can use FindFormostJourney.java to find a foremost journey.
Foremost journey finding from TestCjegReadAtTime.java and FindFormostJourney must have the same arrival time.