UNQS

UPLB Network Queue Simulator

User Manual

Contents

About UNQS 1
Setting the configuration file 2
- The configuration file 2
- Database connection 2
  - Username 2
  - Password flag 2
  - IP address 2
  - Port number 3
  - Database name 3
  - Table name 3
- Bandwidth 3
- Schedule type 3
- Timeout 3
- Start time and End time 3
- Debug flag 3

Running UNQS 4
- Without configuration file 4
- With configuration file 4

Output file 5
- Timestamp 5
- Bandwidth 5
- Dropped flows count and size 5
- Switched flows count and size 5
- Total flow wait time 5
- Duration 5
- Throughput 6

About UNQS

UPLB Network Queue Simulator (UNQS) is a command-line open-source Java program developed for observing network performance. Data collected with the open-source *ntopng* that is stored in a MySQL database is simulated in the program. The program will output the computed the results in a text file.

Setting the configuration file

The configuration file

UNQS may read a configuration file to set values that the program needs. Its format was inspired by ntopng's configuration file. It can include comments (indicated by a # symbol) and whitespaces, both of which are ignored by the file reader. All values that can be changed begin with two hyphens. An example configuration file is shown below.

```
# database connection
--username=root
--password
--ip-address=127.0.0.1
--port=3306
--database=db
--table=flowsv4

# in bits per second
--bandwidth=1000000

# 0=FIFO, 1=PQ, 2=WFQ
--schedule=0

# in seconds
--timeout=60
--starttime=0
--endtime=86400

#--debug
```

Figure 1. Sample configuration file

Database connection

The following items are used to connect to the database.

Username

The username is a required field that is set using `–username=` followed by the name of the user. In Fig. 1, *root* is the default and administrator user per MySQL naming convention.

Password flag

The password flag (`--password`) is added if the database user has a password. Otherwise, the password flag may be omitted. In Fig. 1, a password flag is included which means username *root* has a password.

IP address

The IP address (`--ip-address=`) is a required field that identifies where the database is hosted. Dotted-quad notation is used when writing the IP address. In Fig. 1, the IP address is the local IP address `127.0.0.1`.

Port number

The port number (`--port=`) is a required field to know which port number is used by the database. It is an unsigned 16-bit integer that a computer associates to a certain process or service. The default port number that is reserved for MySQL database is `3306`.

Database name

The database name (`--database=`) is a required field to know which database will be accessed. It follows MySQL's naming convention. In Fig. 1, the database to be used is named `db`.

Table name

The table name (`--table=`) is a required field to know which table contains the flow data. Flow is a connection between two hosts, which is the row data that *ntopng* outputs. The table name follows MySQL's naming convention. In Fig. 1, the table to be used is named `flowsv4`, which is the default table name given by *ntopng* when dumping flow data of hosts using IPv4.

Bandwidth

The bandwidth (`--bandwidth=`) is a required field to know the rate at which flow data will be processed. Bandwidth should be written in unit *bits per second* (bps), so if using bandwidth written in bytes (Bps), it must first be converted. In Fig. 1, the bandwidth to be used is 1000000bps a.k.a. 1Mbps.

Schedule type

The schedule type (`--schedule=`) is a required field to know what scheduling algorithm will be used to process flow data. `0` is for First-In First-Out (FIFO), `1` is for Priority Queueing (PQ), and `2` is for Weighted Fair Queueing (WFQ). The default algorithm is FIFO.

Timeout

The timeout (`--timeout=`) is a required field to know how long to wait in the queue before dropping the flow. Timeout is also known as Time-To-Live (TTL) and it is in unit *seconds*. The default timeout shown in Fig. 1 is 60 seconds.

Start time and End time

The start time (`--start-time=`) and the end time (`--end-time=`) are required fields to know the which data to start processing and which data will be last processed. The values are in the form of Unix timestamp which is counted in the seconds that have passed since January 1, 1970 UTC (Coordinated Universal Time). In Fig. 1, the start and end times means an entire day (the entire January 1, 1970, to be exact) will be simulated in the program.

Debug flag

The debug flag (`--debug`) is added if the program will be set in debug mode. This is useful for the programmers who wish to extend the existing version of UNQS.

Running UNQS

UNQS is a compiled Java program run via command line, therefore the user must have Java installed. The command line interface must be opened in the location path `program`. Note that if there is a password flag, it will ask for a password (Fig. 2). For security purposes, the password is not visible in the command line while typing.

```
MySQL password:
```

Figure 2. MySQL password prompt

Running without configuration file

To run the program using the default configuration embedded in the program,

```
java -cp .:mysql-connector-java.jar UNQS
```

Figure 3. Command to run UNQS without configuration file

It will prompt you the following

```
Continue with the default configuration? (Y/N)
```

Figure 4. Default configuration prompt

If yes, the default configuration prompt defined in *Configuration.java* is used. Otherwise, it will ask for the name of the configuration file.

```
Enter <filename>.conf (must be in same directory of UNQS):
```

Figure 5. Configuration file name prompt

If all the values in the configuration file are correct, the simulation will proceed. If they are not, appropriate error messages will be shown, such as error in database connection, error in the configuration file, etc.

Running with configuration file

To run the program using a user-defined configuration file named *filename.conf*,

```
java -cp .:mysql-connector-java.jar UNQS filename.conf
```

Figure 6. Command to run UNQS with configuration file

If the password flag is set, Fig. 2 will also be prompted. Like when running without configuration file, appropriate error messages will be shown when necessary.

Output file

The output file follows the format *YYYY-MM-DD_x.txt* where
YYYY is the year,
MM is the month, and
DD is the day which is the scope of the data being processed, and
x is the bandwidth value in bps.
A sample output file is shown below.

```
------[ FIFO SIMULATION SUMMARY ]------
TIMESTAMP = 2017-11-13
BANDWIDTH = 40000000 bps

     flows_dropped_size = 0.0 b
     flows_dropped_cnt = 0 flows

     flows_switched_size = 9.39051817096E11 b
     flows_switched_cnt = 554392 flows

     total_wait_time = 63964958344 seconds
     duration = 710346 seconds
     throughput = 1321963.9683984988 bps
```

Figure 7. Sample output file *2017-11-13_40000000.txt*

Timestamp

Timestamp is the start time of the processed flow data converted into readable date format.

Bandwidth

Bandwidth is the rate of processing flow data in bps.

Dropped flows count and size

Dropped flows count is the number of flows that reached timeout and dropped flows size accumulated the total bits that were lost.

Switched flows count and size

Switched flows count is the number of flows that were switched before timeout and switched flows size accumulated the total bits that were serviced successfully.

Total flow wait time

Total flow wait time (in seconds) sums each flow's wait time in the queue, whether they are dropped or switched.

Duration

Duration counts the number of seconds it took to process the flow data given the set timeout, schedule type, and bandwidth.

Throughput

Throughput (bps) is computed as switched flows size divided by duration.