Document for the Stock Trading Client

Yoshihiro Nagano

July 30, 2013

1 Summary

This is the client program for the virtual stock trading game.

2 Protocol

The server side sends company list in uint32_t * 12 data array structure at the beginning of each turn. And that server responses to clients request by the same data structure. The table showed below(Table1) is the strucuture comming from the server.

KEY	CODE
COMPANY_ID	VALUE
COMPANY_ID	VALUE
...	
COMPANY_ID	VALUE

Table 1: data format

KEY is the random number for detecting whether a request is in the same turn or not. It is issued turn by turn. CODE specificate the type of the message. The table showed below(Table2) is the list actual CODE list.

When the beginning of the turn, the CODE in the message from the server will be 0x000, and all VALUEs are filld by the stock price of each company. Client side decide own strategy by this information, and send the requests to the server. Clients can send two types of requests(BUY and SELL), and the CODEs are 0x100 and 0x101 respectively. The upper limit of the number of sending requests in each turn is 5. If the server receive more than 5 requests from one client, she will send a response the CODE type of which is 0x404 or 0x405.

Beginning of the turn(company list)	0x000
Request accepted	0x001
Game end	0x002
Request type: Buy	0x100
Request type: Sell	0x101
Error: Unkown CODE	0x400
Error: Invalid KEY	0x401
Error: Too many requests	0x402
Error: ID not exsists	0x403
Error: Too many BUY requests	0x404
Error: Too many SELL requests	0x405

Table 2: data format

3 Usage

This client has several strategies and the examples of each strategies are compiled in default. So you can use these strategies in easily. In addition, you can simply adding a new strategy or put them together because this program is structured. You can develop the strategy part without taking care of the low layers(especially network communication part or data handling part).

If you want to connect to the server by using the prepared binary, You have to type the command in the terminal like

```
./(binary name) (service) (port num)
```

or if you are alone and want to easily connect to the server by using all 4 clients in parallel, put the command like

```
python send.py (service) (port num) ./(binary name)
```

send.py is like the script which can run 4 binaris in parallel. In addition, this script can log one of the binary's standard output. I made 8 types of binaries, so you can select proper program in each game. I will explain some binaries simply at the next section.

4 Strategy Type

4.1 buysell

This strategy is the most simplest and the most reliable. This strategy will send 5 BUY requests in even turn, and 5 SELL requests in odd turn.

4.2 do_nothing

This program will do nothing after connect to the server. This program is exsits for the debugging.

4.3 test

The difference between this program and others is the output. This program will simply outputs the stock prices of every companies. Any other comments are not throwed to standard output, so it makes easy to log the fluctuation of the company's stock price. This program is exsits for the analyzing the system.

4.4 attack_v1

This strategy was effective before 2013/7/18. This program will send a huge amount of requests to the server and lead the server to segmentation fault. This program targeted at the bug in the server program, but that bug is fixed now(2013/7/30).

4.5 aimbug_v1

This strategy was also effective before 2013/7/18. This program sends 5 BUY requests in every turn and will get a huge amount of money which exceeds UINT32_MAX. The bug in the server program which targeted by this strategy is fixed now(2013/7/30).

5 Proof that It Works

I put the log files that was saved by send.py. These files is in the directory named 'logs'. The name of the log file is the date when the program was executed. So, please look at that log files(especially the latest one!!).