Logging

Elaine Wah
ewah@umich.edu

Updated: September 29, 2014

1 File Location

All log files are saved in the folder `logs` that is created in the directory that contains the `simulation_spec.json` file. Logs (.txt files) are named according the following convention: There are four main sections in the filename, each separated by underscores (examples refer to the filename `simulations_test_0_0_2014-05-12-13-36-04.txt`):

Directory or path The location of spec file, with forward slashes replaced by underscores (e.g., the base directory here is `simulations/test`)

Observation number (e.g., here it is observation #0)

Simulation number (e.g. here it is simulation #0, simulation numbers start at 0)

Date and time Format YYYY-MM-DD for date, followed by hyphen then time in HH-mm-ss format, 24-hour clock

2 Reading Lines

A line in a log file will look like

```
<simulator time>| <message type>| <message>
```

simulator time is the current time in ticks of the simulator when the message was produced.

message type is the level of the logging message. It can be one of three numbers:

1) Error
2) Info
3) Debug

message This is the actual log message which can be a number of different things about the progression of the simulation. Two useful things to be aware of are:

Market IDs are within square brackets, e.g., `[1]`.

Agent IDs are within parentheses, e.g., `(1)`; each agent has a *unique* agent ID.

Note that within the `config` directory there is a file called `env.properties`, which is used to specify the logging level. Log level 2 will be adequate for most debugging purposes. While printing to `System.out` will be useful for initial debugging, looking at the logs and observation files will be necessary in order to verify that agents are interacting and behaving as expected.

3 Configuration File

The log file excerpted in the following sections is located within the `docs` folder (`simulations_test_0_0_2014-05-12-13-36-04.txt`), and was generated using the following `simulation_spec.json` file:

```
{
    "assignment": {
    },
    "configuration": {
        "randomSeed": "271828",
        "numSims": "1",
        "modelName": "example",
        "presets": "NONE",
        "simLength": "60000",
        "tickSize": "1",
        "nbboLatency": "100",
        "mktLatency": "-1",
        "arrivalRate": "0.075",
        "reentryRate": "0.0005",
        "meanValue": "100000",
        "kappa": "0.05",
        "shockVar": "1E8",
        "privateValueVar": "1E8",
        "CDA": "num_2",
        "CALL": "num_0",
        "LA": "num_1",
        "BASICMM": "num_0_numRungs_10_rungSize_1000",
        "MAMM": "num_0",
        "WMAMM": "num_0",
        "ZI": "num_0_bidRangeMin_0_bidRangeMax_5000",
        "ZIR": "num_61_maxqty_10",
        "ZIP": "num_0",
        "AA": "num_0"
    }
}
```

4 Simulation Initialization

When a simulation is initialized, the following items are specified: 1) the random seed used to run the simulation, and 2) the full configuration. For example:

```
0| 2| Random Seed: 271828
0| 2| Configuration: {"assignment":{},"configuration":{"
   randomSeed":"271828","numSims":"1","modelName":"example
   ","presets":"NONE","simLength":"60000","tickSize":"1","
   nbboLatency":"100","mktLatency":"-1","arrivalRate
   ":"0.075","reentryRate":"0.0005","meanValue":"100000","
   kappa":"0.05","shockVar":"1E8","privateValueVar":"1E8","
   CDA":"num_2","CALL":"num_0","LA":"num_1","BASICMM":"
   num_0_numRungs_10_rungSize_1000","MAMM":"num_0","WMAMM":"
   num_0","ZI":"num_0_bidRangeMin_0_bidRangeMax_5000","ZIR
   ":"num_61_maxqty_10","ZIP":"num_0","AA":"num_0"}}
```

In the previous file each line means:

`Random Seed : 271828`: The random seed used to generate this simulation run.

`Configuration :...`: The full `simulation_spec.json` file for this simulation.

5 Market simulation

After markets are created the simulation begins running.

Agent Strategy: An example of an agent strategy is present in line 5 of the log file:

```
3| 2| ZIR (1) executing ZI strategy position=0, for q
   =1, value=$59193 + $1872=$61065
```

Agent strategies can take many forms depending on the agent, but they will generally look similar to the preceding line.

Agent Submit Order: When an agent submits a order it will take the form of:

```
<agent> <order type> <quantity> @ <price> in <market>
```

where order type is either `BUY` or `SELL`. An example can be found on line 6 of the log file:

```
3| 2| ZIR (1) SELL 1 @ $65241 in CDA [1]
```

Market Quote Update: When a market updates its internal quote it will be reflected in the log as:

```
<market> (Bid: <bid price>, Ask: <ask price>)
```

If a bid or ask doesn't exist than instead of a price there will be a hyphen (-). An example can be found in line 7 of the log file:

```
    3| 2| CDA [1] (Bid: - , Ask: 1 @ $65241)
```

Note this is the market's internal quote, and may not available to every agent immediately.

Transaction: When two agents transact the quantity and price are indicated as well as the new positions of the traders involved:

```
<buyer> bought <quantity> from <seller> @ <price> in <market>
<agent1> transacted to position <position1>
<agent2> transacted to position <position2>
```

An example can be found on lines 32 and 34 of the log file:

```
   46| 2| ZIR (2) bought 1 from ZIR (6) @ $48251 in CDA [2]
   46| 2| ZIR (2) transacted to position 1
   46| 2| ZIR (6) transacted to position -1
```

NBBO Update: After enough time has a passed a market's quote update will propagate to the NBBO. This will look like:

```
<market> -> SIP quote (Bid: <bid>, Ask: <offer>)
```

Where bid and offer are the updated values that the SIP is getting. An example can be found on line 67 of the log file:

```
  103| 2| CDA [1] -> SIP quote (Bid: - , Ask: 1 @ $65241)
```

Reg NMS Order Routing: When an order comes in that can be routed for better execution at another market according to the NBBO you'll see a line such as the following, from line 89 in the log file:

```
  129| 2| Routing ZIR (14) SELL 1 @ $76512 from CDA [2] (
     Bid: 1 @ $39220, Ask: 1 @ $119163) to NBBO CDA [1] (
     BestBid: 1 @ $52041 from CDA [1], BestAsk: 1 @ $65241
     from CDA [1])
```

which indicates the price disparities in market quotes that led to the routing.

Latency Arbitrage: When a latency arbitrage opportunity is found by a latency arbitrageur (LA) there will be a series of log messages similar to the following output from line 260 to line 270 in the log file:

```
386| 2| CDA [2] (Bid: 1 @ $59387, Ask: 1 @ $69484)
386| 2| LA (62) detected arbitrage between CDA [1] (Bid:
   1 @ $71477, Ask: 1 @ $88972) and CDA [2] (Bid: 1 @
   $59387, Ask: 1 @ $69484)
386| 2| LA (62) SELL 1 @ $70480 in CDA [1]
386| 2| ZIR (18) bought 1 from LA (62) @ $71477 in CDA
   [1]
386| 2| ZIR (18) transacted to position 1
386| 2| LA (62) transacted to position -1
386| 2| CDA [1] (Bid: 1 @ $63464, Ask: 1 @ $88972)
386| 2| LA (62) BUY 1 @ $70480 in CDA [2]
386| 2| LA (62) bought 1 from ZIR (34) @ $69484 in CDA
   [2]
386| 2| ZIR (34) transacted to position -1
386| 2| LA (62) transacted to position 0
```

It starts with a market updating its quote and exposing an arbitrage opportunity. The LA will then announce that it detected the arbitrage opportunity. It then submits orders to each market, resulting in two transactions that leave LA at net position 0.