Logging

Elaine Wah
ewah@umich.edu

Updated: October 14, 2013

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 sections in the filename, each separated by underscores (examples refer to the filename `simulations_1_14-Oct-2013_15.56.06.txt`):

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

Observation number (e.g., here it is observation #2)

Date (dd-MMM-YYYY format)

Time (hh.mm.ss format, 24-hour clock, e.g., time here is 1:45:00 PM)

2 Reading Lines

A line in a log file will look like

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

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

1) Error
2) Info
3) Debug

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

model num is the model number of the simulation. The default value is 0, but it can be set to other numbers for merging log files.

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

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

Agent ID s are positive, and 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 behaving as expected.

3 Configuration File

The log file excerpted in the following sections is located at `simulations_1_14-Oct-2013_15.56.06.txt`, and was generated using the following `simulation_spec.json` file:

```
{
    "assignment": {
    },
    "configuration": {
        "modelName" : "mySimulation",
        "modelNum" : "1",
        "simLength": "15000",
        "presets" : "TWOMARKETLA",
        "CDA" : "num_0",
        "CALL" : "num_0",
        "BASICMM": "
           num_0_sleepTime_200_numRungs_10_rungSize_1000",
        "ZI": "num_250_bidRangeMin_0_bidRangeMax_5000",
        "ZIR": "num_0_maxqty_5",
        "ZIP": "num_0",
        "AA": "num_0",
        "tickSize": "1",
        "nbboLatency": "100",
        "arrivalRate": "0.075",
        "reentryRate": "0.005",
        "meanValue": "100000",
        "kappa": "0.05",
        "shockVar": "150000000",
        "privateValueVar": "100000000"
    }
}
```

4 Simulation Initialization

When a simulation is initialized, the following items are specified: 1) the name of the simulation, 2) the random seed used to run the simulation, 3) the full configuration, 4) number and type of each market, and 5) market ID. For example:

```
2|    0|1| mySimulation
2|    0|1| Random Seed: -1198902552
2|    0|1| Configuration: {"assignment":{},"configuration":{"
   modelName":"mySimulation","modelNum":"1","simLength
   ":"15000","presets":"TWOMARKETLA","CDA":"num_2","CALL":"num_0
   ","BASICMM":"num_0_sleepTime_200_numRungs_10_rungSize_1000","
   ZI":"num_250_bidRangeMin_0_bidRangeMax_5000","ZIR":"
   num_0_maxqty_5","ZIP":"num_0","AA":"num_0","tickSize":"1","
   nbboLatency":"100","arrivalRate":"0.075","reentryRate
   ":"0.005","meanValue":"100000","kappa":"0.05","shockVar
   ":"150000000","privateValueVar":"100000000","LA":"num_1"}}
2|    0|1| Created Market: CDA [1]
2|    0|1| Created Market: CDA [2]
```

In the previous file each lines means:

`mySimulation`**:** Is the name of the simulation. This is used in the observation file and give a unique name to a specific simulation.

`Random Seed : -1282951707`**:** This prints out the random seed used to generate this simulation.

`Configuration :...`**:** Is the entire `simulation_spec.json` file for this simulation.

`Created Market : CDA [1]`**:** Specifies that market 1 is a CDA market.

5 Market simulation

After markets are created the simulation begins running.

Agent Strategy: An example of an agent strategy is present in line 8 of the log file:

```
2|    0|1| ZI (1) ZIAgent: executing ZI strategy position
   =0, for q=1, value=$98.905 + $0.091=$98.996
```

Agent strategies can take many forms depending on the agent, but they'll mostly look something like the preceding line.

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

```
<agent> (<price>, <quantity>) -> <market>
```

where a negative quantity is used to indicate sell orders. An example can be found on line 9 of the log file:

```
2|    0|1| ZI (1) ($100.152, 1) -> CDA [1]
```

Market Quote Update: When a market updates its internal quote you'll see a line like the following:

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

If a bid or ask doesn't exist than instead of seeing a price you'll see a hyphen (-). An example can be found in line 10 of the log file:

```
2|     0|1| CDA [1] (Bid: $100.152, Ask: - )
```

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

Transaction: When two agents transact you'll see two lines like the following:

```
<agent1> transacted to position <position1>
<agent2> transacted to position <position2>
```

An example can be found on lines 49 and 50 of the log file:

```
2|    49|1| ZI (6) transacted to position 1
2|    49|1| ZI (8) 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: <nb bid>, Ask: <nb offer>)
```

An example can be found on line 77 of the log file:

```
2|  130|1| CDA [1] -> SIP quote (Bid: $101.64, Ask: - )
```

SIP 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 like line 97 in the log file:

```
2|  205|1| Routing ZI (14) ($0.00, -1) -> CDA [2] (Bid:
   $127.735, Ask: - ) to NBBO (Bid: $129.358, Ask: null)
```

which indicates the price disparities that caused the routing.

Latency Arbitrage: When latency arbitrage exists you'll see a series of log messages like the following output from like 13 to line 22 in the log file:

```
2|     4|1| CDA [2] (Bid: - , Ask: $99.979)
2|     4|1| LA (251) detected arbitrage between CDA [1] (Bid
   : $100.152, Ask: - ) and CDA [2] (Bid: - , Ask: $99.979)
2|     4|1| LA (251) ($100.065, -1) -> CDA [1]
2|     4|1| ZI (1) transacted to position 1
2|     4|1| LA (251) transacted to position -1
2|     4|1| CDA [1] (Bid: - , Ask: - )
2|     4|1| LA (251) ($100.065, 1) -> CDA [2]
2|     4|1| LA (251) transacted to position 0
2|     4|1| ZI (2) transacted to position -1
2|     4|1| CDA [2] (Bid: - , Ask: - )
```

It starts with a market updating its quote and exposing an arbitrage opportunity. The Latency Arbitrageur (LA) will then announce that it detected the arbitrage and what follows is subsequent bids being placed and transactions that will leave the LA at net position 0.