# Parallel Applications User Guide

## 1 App Configuration

The parallel application requires a JSON configuration file to be passed to the parallel application using a command line parameter. The JSON file location is passed using the -f option.

In the JSON config file a bool called is\_server is required so the parallel app knows to configure the app as a client or server.

This would look like,

“is\_server” : true

The client app also must have an app name,

Client App Naming Convention

Predictor\_[MARKET\_TYPE]

For example: Predictor\_Technology

### Server app configuration

#### 1.1.1 UDP Socket configuration

server\_socket with integer fields ip and port.

Ip corresponds to the IP of the local device.

port corresponds to the port the receive socket is started on.

##### Example JSON configuration

"server\_socket": {

"ip" : "127.0.0.1",

"port" : "10000"

},

#### 1.1.2 Client List

Client list is an array of clients that contains info about clients the server intends to connect too. It contains string fields for the app name, ip, port number, and stock list.

app\_name is the string name of the client that identifies the app which sent a given message.

ip is the ip address of the client to send messages too.

port is the port number of the client to send messages too.

stock\_list is a string of stock tickers separated by comas.

"client\_list": [

{

"app\_name": "Predictor\_Technology",

"ip": "127.0.0.1",

"port\_number": "10001",

"stock\_list" : "SYK,AAL"

}

]

### 1.2 Client app configuration

#### 1.2.1 UDP socket configuration

server\_socket with integer fields ip and port.

client\_socket with integer fields ip and port.

Ip corresponds to the IP of the server app.

port corresponds to the port the servers receive socket is on.

##### Example JSON configuration

"server\_socket": {

"ip" : "127.0.0.1",

"port" : "10000"

},

    "client\_socket": {

        "ip" : "127.0.0.1",

        "port" : "10001"

    },

#### 1.2.2 Python scripting configuration

Some parameters for running a python script must also be included. Such as the location of the python predictor scripts, the python interpreter to use, and info regarding the predictor scripts.

    "python\_model\_script\_dir" : "C:\\Users\\acichy\\TestCode\\Stock\_Market\_Predictor\\src",

    "python\_venv" : "C:\\Users\\acichy\\TestCode\\Stock\_Market\_Predictor\\myenv\\Scripts\\python",

    "download\_data": {

        "script\_name": "get\_data",

        "function\_name": "download\_stock\_data"

    },

    "train\_model": {

        "script\_name": "main",

        "function\_name": "main"

    },

    "predict\_model": {

        "script\_name": "use\_model",

        "function\_name": "main",

        "trained\_model\_path": "TrainedModels"

    }

## Usage

After starting the parallel app using a command such as,

parallel\_app.exe -f parallel\_config.json

A menu with 4 options will be shown. The options are,

1. Download Stock Data
2. Train Model
3. Make predictions

q. Quit

After selecting 1-3 the server will send the corresponding command to each client on the list via UDP messages.

### Message types

#### 2.1.1 Parallel App Message types

##### 2.1.1.1 Setup Message:

Message ID: 0

Purpose:

Sent from a client to a server to identify the client as online and ready to receive commands

##### 2.1.1.2 Stock List Message:

Message ID: 1

Purpose:

Sent from a server to a client to inform the client of the list of stocks it is expected to use.

When receiving this message the client must download the latest stock data for each of the stocks on the list.

Message Content:

List of comma deliniated stock tokens. For Example: "SYK,AAL,etc..."

##### 2.1.1.3 Stock Data Ready Message:

Message ID: 2

Purpose:

Sent from a client to a server to inform the server all stock data has been downloaded.

##### 2.1.1.4 Begin Training Message:

Message ID: 3

Purpose:

Sent from a server to a client so the client begins training on the stock data.

Client will sequentially train a model on the stock list.

Message Content:

Training hyper parameter list

[EPOCHS].[LEARNING\_RATE]

##### 2.1.1.5 Training complete message:

Message ID: 4

Purpose:

Sent from a client to a server to indicate training has been completed.

##### 2.1.1.6 Make Predictions message:

Message ID: 5

Purpose:

Sent from a server to a client to use the trained model to make predictions on the stock list.

Message Content:

TBD, probably nothing.

##### 2.1.1.7 Predictions complete message:

Message ID: 6

Purpose:

Sent from a client to server returns the prediction results

Message Content:

[STOCK]:[0 | 1], ...

Contains the stock ticker and the clients recommendation whether or not to invest in the stock.

Parallel App Message structure

[APP\_NAME].[MESSAGE\_ID].[MESSAGE\_CONTENTS]