GPU PROJECT

Design and Functional Specification

Project Requirements and Architecture

<May 2, 2013>

Owners and List of Contacts

Name	Email	Phone	Role
Antony Drew			Developer

Signoffs

Phase	Name	Date	Signature
Completed		5/3/2013	

Revision History

Date	Reason for change(s)	Author(s)

Table of Contents

About this Document

Purpose and Contents

What this specification does

This document describes functionality of the trading program using CUDA C GPU computing.

What this specification does not do

This is not a project plan. It is a user guide.

Assumptions

The user has some general knowledge about computing overall.

Questions and Comments

If you have questions or comments regarding this document, contact: antonydrew@gmail.com

Software Overview

Product description

This CUDA C application is used as a research tool in terms of global futures trading strategies. A user can easily change the rules and metrics for trading inside the C code. This tool helps to explore what happens to securities at extreme **overbought/oversold** levels aiming to potentially provide the user with a means to make a profit. More importantly, it is a tool to show speed benefits of using hybrid GPU computing rather than CPU alone. Note that there are varying levels of complexity of the program and versions in both Windows (Visual Studio 2008 w/.NET 3.5 or higher) and LINUX (command line or free Eclipse IDE). Past testing performance does not guarantee future trading success, however.

Product functional capabilities

This trade product provides historical back-testing and portfolio optimization capabilities. It also provides reporting metrics allowing the user to further explore various markets facilitating decision-making.

It does **NOT** provide DATA DOWNLOAD capabilities via a tool such as .ASP NET. This is because futures data is **NOT** publically available like YAHOO STOCKS. Futures data is expensive and is purchased on a subscription basis through vendors like BLOOMBERG, CSI and TELERATE who distribute it via flat files (csv). This data is on bitbucket can be downloaded here:
https://bitbucket.org/adrew/gpu/commits/c96e33bed2dbf7adb97995cf6caa9fd7731e7af6

For the sake of testing this program, a static set of csv/dat files (owned by developer) was used here. Note that futures data is preferable since it truly provides global asset diversification via stocks, commodities and bonds (as opposed to YAHOO stock data).

User characteristics

Users will typically be traders or people interested in finance and computers, overall.

User operations and practices

This is a straightforward application with only a few user inputs though code is somewhat complex: 1000 lines (or 3000 lines with proper carriage returns) – to save space and redundancy, similar syntax are grouped on the SAME line as well as simple (1-line) for-next loops. There are also several error-catching (try/catch) routines in case of user error. Note that there is **no** GUI/UI interface for the user – variable user inputs are found at the beginning of the program under **#define** in the preprocessor area of the CUDA C file called "**ban.cu**" (line 22).

To **execute** program in Windows, user should simply double-click **DemoCu.exe** in the **DEBUG** folder of the project (wherever user has saved) or can open **.sln** file of project in Visual Studio 2008 IDE and then hit "run" button.

For **LINUX**, user must navigate to the **DEBUG** folder of the project (wherever user has saved) via the terminal command line and then type: **./DemoCu**

General constraints

The user must have an NVIDIA GPU device along with the CUDA SDK toolkit (5.0) and video drivers installed: https://developer.nvidia.com/cudadownloads . The user must have a Windows or Linux (Ubuntu) OS either 32-bit or 64-bit. For server version, the user must have access to GPU KEPLER TESLA cluster and internet connection – for local machine version, no internet is needed.

The user must download the entire GPU repository from bitbucket above: https://bitbucket.org/adrew/gpu - simply go to this link and click-on the DOWNLOAD button. Note that DATA folder (mentioned above) that has all the market data flat files and save this to a specific location. The user must than point to this location by changing the PATH variable in the preprocessor area of the CUDA C file called "**ban.cu**". The program will not work without being able to access these data files (which are very large).

Here are some sample PATHS (line 22 of "ban.cu"):

```
#define PATH "/home/tony/DemoCuMultiGPU/"       (LINUX user)
#define PATH "C:\\"                             (Windows user)
```

Other software

Yes – NVIDIA CUDA SDK toolkit and video drivers as mentioned above already.

For program compilation (assuming user has changed something), user can build in Visual Studio 2008 IDE or ECLIPSE in LINUX. If user does not have ECLIPSE, then this must be done in terminal at command line:

- Go to /DEBUG folder of project (wherever is saved) and type:
    - **make clean**
    - **make all** or **make**
    - **./DemoCu**

System Architecture

Description:

The GPU demo is only made up of 1 PROJECT inside 1 SOLUTION. For **Windows** users, I was able to split these up into 3 source files:

- "ban.cu" (main program file – entry point that calls other functions & #include files below)
- "utility.c" (contains definitions of ALL functions)
- "utility.h" (contains forward declaration of ALL functions contained in "utility.c")

For **LINUX** users, I had to put everything in 1 file since ECLIPSE IDE has trouble including other references:

- "ban.cu" (main program file – entry point that calls other functions and contains ALL functions (on bottom) and forward declarations of functions (on top))

i. **CUDA C (.cu files)**
   a. Contains logic and architecture of main program & is entry point
   b. Structure is 5 nested "for-next" loops: LOOP BY GPU DEVICE NUMBER >>> LOOP BY MARKET >>> LOOP BY TIME PERIOD >>> LOOP BY FIRST MODEL PARAMETER >>> LOOP BY SECOND MODEL PARAMETER:

ii. **Forms**
   a. N/A

iii. **WebClient**
   a. N/A

iv. **General Data Structures Utilized:**
   a. Arrays (constant time/linear time)
   b. Pointers to arrays (constant time/linear time)
   c. Constants in preprocessor set by user under **#define**

Class Diagrams

N/A - There are **no** class diagrams for this project since C is not an OOP language.

Database Schema

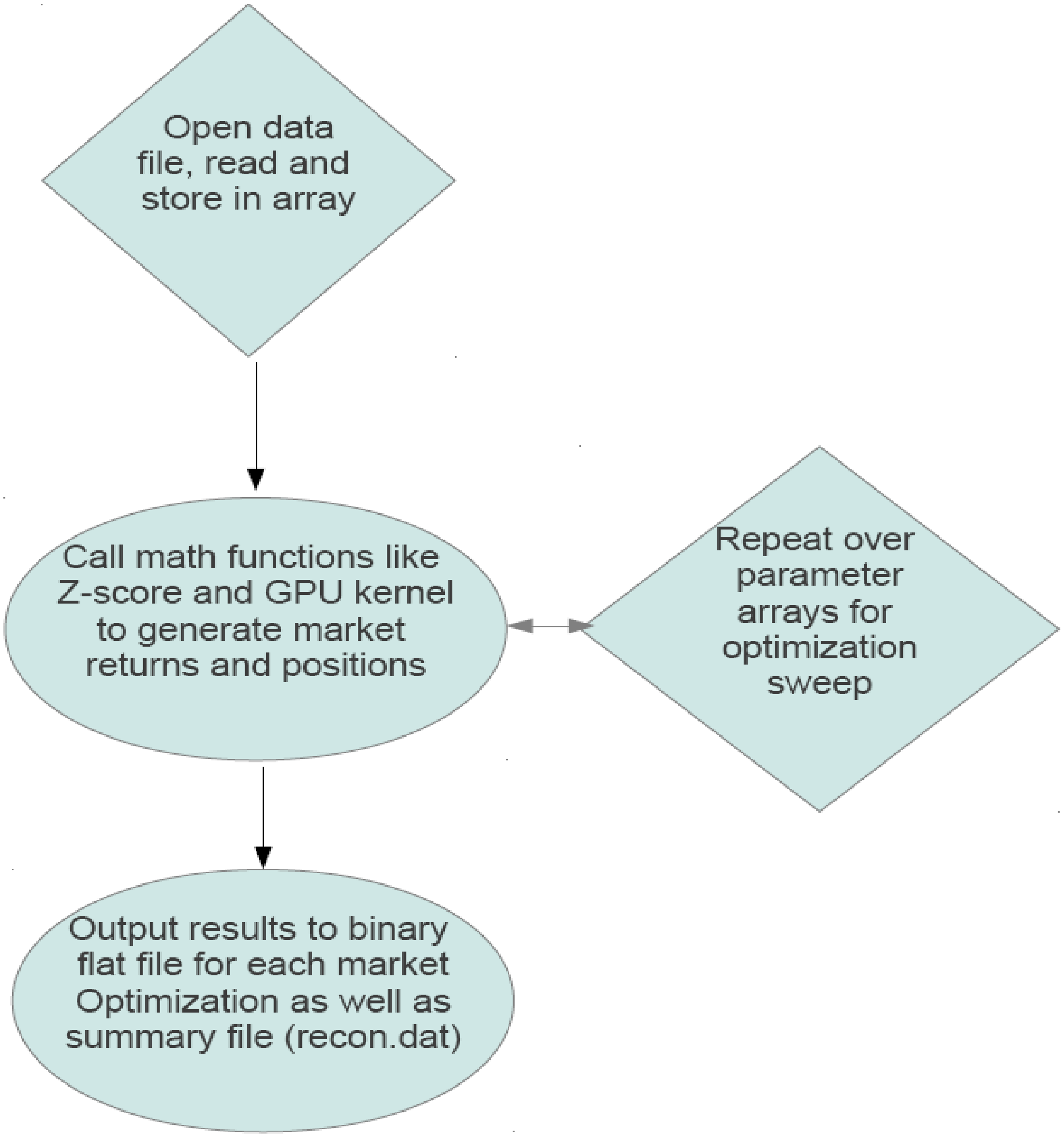
N/A – There are **no** databases used in this program. As mentioned already, data flat files (.dat) are uploaded and processed from vendor/subscription service.

Specific Function Description – FLOW CHART

1. GPU Project

Description

The general flow of the project can be described by the following flow chart:


Open data file, read and store in array
Call math functions like Z-score and GPU kernel to generate market returns and positions
Repeat over parameter arrays for optimization sweep
Output results to binary flat file for each market Optimization as well as summary file (recon.dat)

User Interface

N/A – there are NO GUI/UI's as mentioned already.

Inputs

All user-adjustable inputs are in the preprocessor section of the "ban.cu" file at line 22. They include:

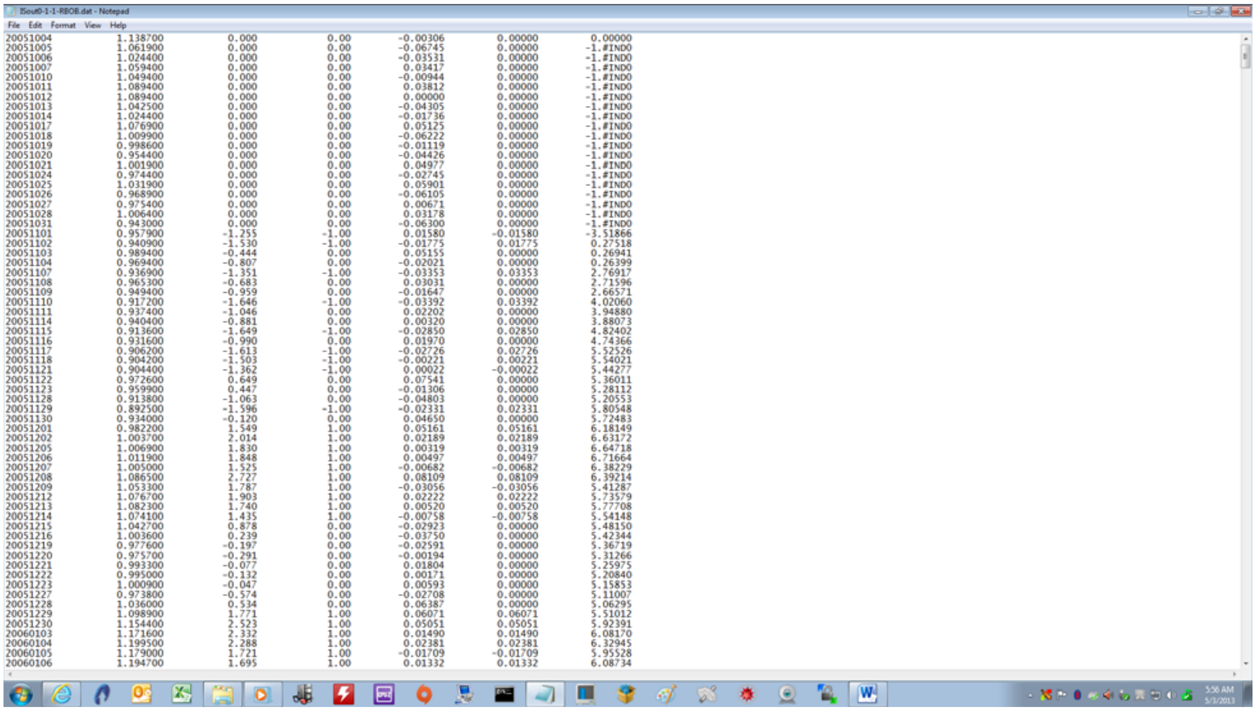
- PATH – this is location where user has downloaded data files
- LOOKBACK – length of days in history to generate returns for rolling optimization
- STEP – frequency in days that optimization is re-run
- NUMI – number of markets – 27 max total, but user may want to run only subset
- STR (only in Multi-GPU/Multi-Stream version)

Processing

During processing, a terminal will open and scroll through optimization output for each time period of each market – the terminal will close on completion or wait for user to hit key depending on OS. This is an intensive program so the user should not run other applications.

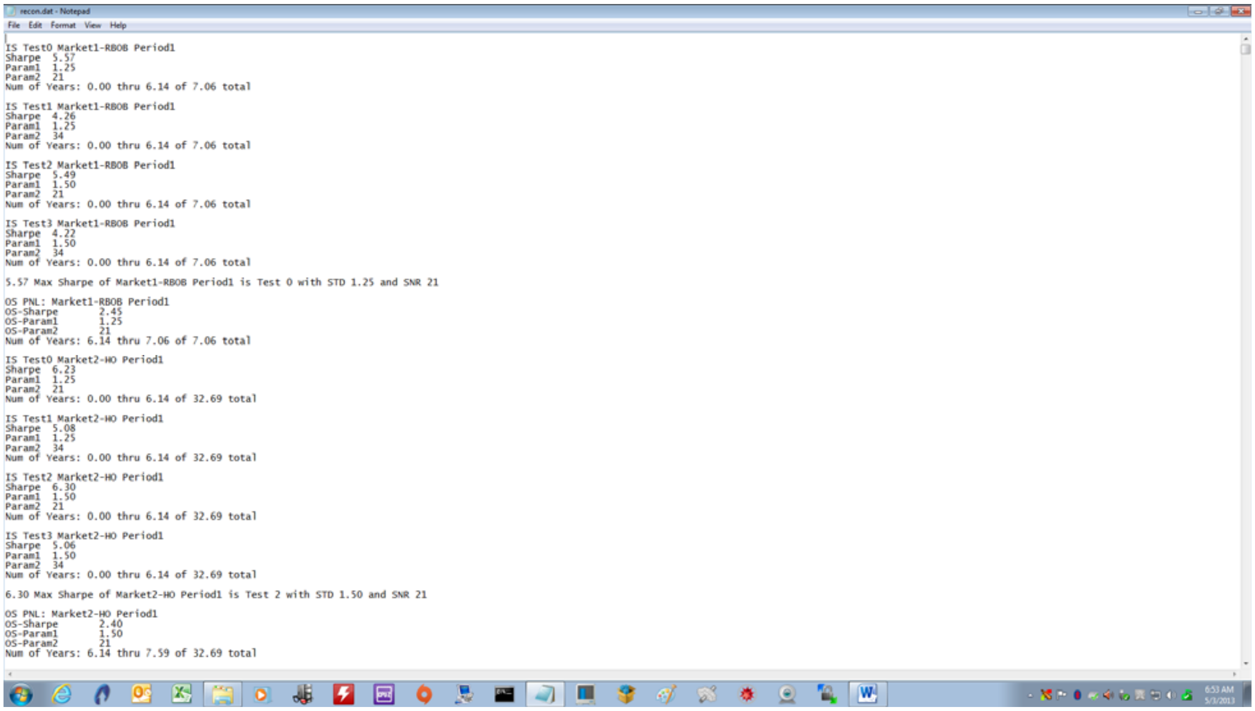
Reports

Reports are **NOT** customizable – the results of every period IN-SAMPLE/OUT-SAMPLE optimization are OUTPUT to binary flat files (.dat) wherever the user has set the PATH variable (#define PATH at line 22 of preprocessor):



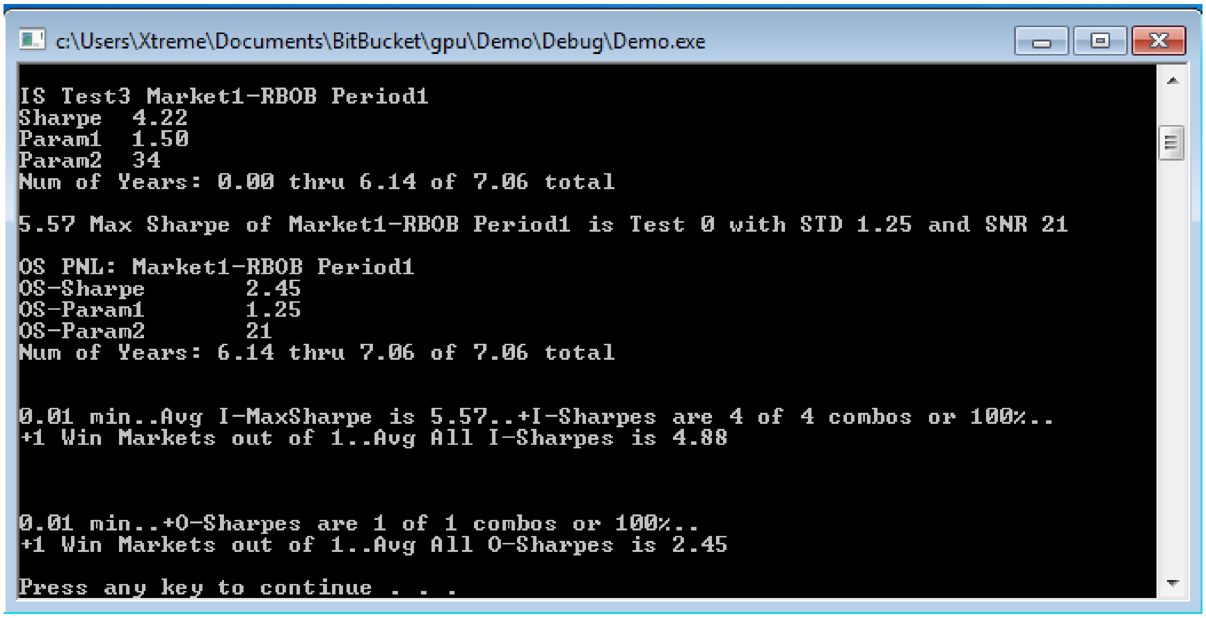
```
20051004      1.138700     0.000      0.00    -0.00306    0.00000    0.00000
20051005      1.061900     0.000      0.00    -0.06745    0.00000   -1.#IND0
20051006      1.024400     0.000      0.00    -0.03531    0.00000   -1.#IND0
20051007      1.059400     0.000      0.00     0.03417    0.00000   -1.#IND0
20051010      1.049400     0.000      0.00    -0.00944    0.00000   -1.#IND0
20051011      1.089400     0.000      0.00     0.03812    0.00000   -1.#IND0
20051012      1.089400     0.000      0.00     0.00000    0.00000   -1.#IND0
20051013      1.042500     0.000      0.00    -0.04305    0.00000   -1.#IND0
20051014      1.024400     0.000      0.00    -0.01736    0.00000   -1.#IND0
20051017      1.076900     0.000      0.00     0.05125    0.00000   -1.#IND0
20051018      1.009900     0.000      0.00    -0.06222    0.00000   -1.#IND0
20051019      0.998600     0.000      0.00    -0.01119    0.00000   -1.#IND0
20051020      0.954400     0.000      0.00    -0.04426    0.00000   -1.#IND0
20051021      1.001900     0.000      0.00     0.04977    0.00000   -1.#IND0
20051024      0.974400     0.000      0.00    -0.02745    0.00000   -1.#IND0
20051025      1.031900     0.000      0.00     0.05901    0.00000   -1.#IND0
20051026      0.968900     0.000      0.00    -0.06105    0.00000   -1.#IND0
20051027      0.975400     0.000      0.00     0.00671    0.00000   -1.#IND0
20051028      1.006400     0.000      0.00     0.03178    0.00000   -1.#IND0
20051031      0.943000     0.000      0.00    -0.06300    0.00000   -1.#IND0
20051101      0.957900    -1.255     -1.00     0.01580   -0.01580   -3.51866
20051102      0.940900    -1.530     -1.00    -0.01775    0.01775    0.27518
20051103      0.989400    -0.444      0.00     0.05155    0.00000    0.26941
20051104      0.969400    -0.807      0.00    -0.02021    0.00000    0.26399
20051107      0.936900    -1.351     -1.00    -0.03353    0.03353    2.76917
20051108      0.965300    -0.683      0.00     0.03031    0.00000    2.71596
20051109      0.949400    -0.959      0.00    -0.01647    0.00000    2.66571
20051110      0.917200    -1.646     -1.00    -0.03392    0.03392    4.02060
20051111      0.937400    -1.046      0.00     0.02202    0.00000    3.94880
20051114      0.940400    -0.881      0.00     0.00320    0.00000    3.88073
20051115      0.913600    -1.649     -1.00    -0.02850    0.02850    4.82402
20051116      0.931600    -0.990      0.00     0.01970    0.00000    4.74366
20051117      0.906200    -1.613     -1.00    -0.02726    0.02726    5.52526
20051118      0.904200    -1.503     -1.00    -0.00221    0.00221    5.54021
20051121      0.904400    -1.362     -1.00     0.00022   -0.00022    5.44277
20051122      0.972600     0.649      0.00     0.07541    0.00000    5.36011
20051123      0.959900     0.447      0.00    -0.01306    0.00000    5.28112
20051128      0.913800    -1.063      0.00    -0.04803    0.00000    5.20553
20051129      0.892500    -1.596     -1.00    -0.02331    0.02331    5.80548
20051130      0.934000    -0.120      0.00     0.04650    0.00000    5.72483
20051201      0.982200     1.549      1.00     0.05161    0.05161    6.18149
20051202      1.003700     2.014      1.00     0.02189    0.02189    6.63172
20051205      1.006900     1.830      1.00     0.00319    0.00319    6.64718
20051206      1.011900     1.848      1.00     0.00497    0.00497    6.71664
20051207      1.005000     1.525      1.00    -0.00682   -0.00682    6.38229
20051208      1.086500     2.727      1.00     0.08109    0.08109    6.39214
20051209      1.053300     1.787      1.00    -0.03056   -0.03056    5.41287
20051212      1.076700     1.903      1.00     0.02222    0.02222    5.73579
20051213      1.082300     1.740      1.00     0.00520    0.00520    5.77708
20051214      1.074100     1.435      1.00    -0.00758   -0.00758    5.54148
20051215      1.042700     0.878      0.00    -0.02923    0.00000    5.48150
20051216      1.003600     0.239      0.00    -0.03750    0.00000    5.42344
20051219      0.977600    -0.197      0.00    -0.02591    0.00000    5.36719
20051220      0.975700    -0.291      0.00    -0.00194    0.00000    5.31266
20051221      0.993300    -0.077      0.00     0.01804    0.00000    5.25975
20051222      0.995000    -0.132      0.00     0.00171    0.00000    5.20840
20051223      1.000900    -0.047      0.00     0.00593    0.00000    5.15853
20051227      0.973800    -0.574      0.00    -0.02708    0.00000    5.11007
20051228      1.036000     0.534      0.00     0.06387    0.00000    5.06295
20051229      1.098900     1.771      1.00     0.06071    0.06071    5.51012
20051230      1.154400     2.523      1.00     0.05051    0.05051    5.92391
20060103      1.171600     2.332      1.00     0.01490    0.01490    6.08170
20060104      1.199500     2.288      1.00     0.02381    0.02381    6.32945
20060105      1.179000     1.721      1.00    -0.01709   -0.01709    5.95528
20060106      1.194700     1.695      1.00     0.01332    0.01332    6.08734
```

Summary report “recon.dat” of all market optimizations looks like:



```
IS Test0 Market1-RBOB Period1
Sharpe  5.57
Param1  1.25
Param2  21
Num of Years: 0.00 thru 6.14 of 7.06 total

IS Test1 Market1-RBOB Period1
Sharpe  4.26
Param1  1.25
Param2  34
Num of Years: 0.00 thru 6.14 of 7.06 total

IS Test2 Market1-RBOB Period1
Sharpe  5.49
Param1  1.50
Param2  21
Num of Years: 0.00 thru 6.14 of 7.06 total

IS Test3 Market1-RBOB Period1
Sharpe  4.22
Param1  1.50
Param2  34
Num of Years: 0.00 thru 6.14 of 7.06 total

5.57 Max Sharpe of Market1-RBOB Period1 is Test 0 with STD 1.25 and SNR 21

OS PNL: Market1-RBOB Period1
OS-Sharpe       2.45
OS-Param1       1.25
OS-Param2       21
Num of Years: 6.14 thru 7.06 of 7.06 total

IS Test0 Market2-HO Period1
Sharpe  6.23
Param1  1.25
Param2  21
Num of Years: 0.00 thru 6.14 of 32.69 total

IS Test1 Market2-HO Period1
Sharpe  5.08
Param1  1.25
Param2  34
Num of Years: 0.00 thru 6.14 of 32.69 total

IS Test2 Market2-HO Period1
Sharpe  6.30
Param1  1.50
Param2  21
Num of Years: 0.00 thru 6.14 of 32.69 total

IS Test3 Market2-HO Period1
Sharpe  5.06
Param1  1.50
Param2  34
Num of Years: 0.00 thru 6.14 of 32.69 total

6.30 Max Sharpe of Market2-HO Period1 is Test 2 with STD 1.50 and SNR 21

OS PNL: Market2-HO Period1
OS-Sharpe       2.40
OS-Param1       1.50
OS-Param2       21
Num of Years: 6.14 thru 7.59 of 32.69 total
```

Output Screen:



```
IS Test3 Market1-RBOB Period1
Sharpe  4.22
Param1  1.50
Param2  34
Num of Years: 0.00 thru 6.14 of 7.06 total

5.57 Max Sharpe of Market1-RBOB Period1 is Test 0 with STD 1.25 and SNR 21

OS PNL: Market1-RBOB Period1
OS-Sharpe       2.45
OS-Param1       1.25
OS-Param2       21
Num of Years: 6.14 thru 7.06 of 7.06 total


0.01 min..Avg I-MaxSharpe is 5.57..+I-Sharpes are 4 of 4 combos or 100%..
+1 Win Markets out of 1..Avg All I-Sharpes is 4.88



0.01 min..+O-Sharpes are 1 of 1 combos or 100%..
+1 Win Markets out of 1..Avg All O-Sharpes is 2.45

Press any key to continue . . .
```

Security

N/A – This program is operated at user discretion on local machine or server