Proposal: Reception, Processing, and Display of Stock Market Data Using the Solarflare SFA6900 FPGA Board

Team Name: We Love Money
Naman Parashar (np2437), Miles Sherman (ms4543), Pranav Sood (ps2729), Kevin Wong (kw2500),
and Art Iakovlev (ai2283)
Fu Foundation School of Engineering and Applied Science
Columbia University, New York, NY 10027

I. Overview

As was discussed in class, a major obstacle for stock markets around the world is the management of market data that is continuously growing in quantity and complexity. Classically, the transmission, receipt, and processing of this data has been handled by software. However, because of the exponential growth in data as well as latency requirements, hardware is quickly making its way into the industry. The most notable example of this would be the way data is transmitted from the NASDAQ stock exchange. The NASDAQ currently transmits market data both from classical software platforms and more recently from a high speed FPGA platform for extremely low latency.

For our project, we plan to utilize the Solarflare AoE FPGA card to design a system to receive stock market data, log it in real time, and broadcast the data (to an x86 software we will develop). Thanks to this new card, which can receive and transmit data at up to 10Gbits/s, we will be able to maintain the L1 market data book for one or more stock symbols in real time (assuming access to real time NASDAQ market data as provided by David Lariviere, otherwise the market data stream will be simulated based on previous market snapshots).

An L1 market book maintains up to date information on the status of every stock in a certain exchange. For each stock, this information includes ticker symbol, bid price, bid size, ask price, ask size, last sale price, last sale size, last sale timestamp, last quote timestamp, and current day's volume.

II. Functionality

While our idea is still in its preliminary stage, we have a basic concept of how our design will work. Assuming we have access to us an incoming stream of market data from a major stock exchange, the card will first input the data via Ethernet. Data will be decoded, its type (ticker symbol, ask price, bid price, etc) will be identified, and the data will be stored in a data structure. This data structure will be held either in the FPGA's on-board memory or on the included DDR memory on the Solarflare card depending on space requirements. With the data structure in place and constantly updating in real time, we will then broadcast parts of our data structure as output.

To provide a user-accessible interface for the information we gather, we will implement a simple user interface to view the stored L1 book.

III. Inputs

Our only input for this project will be market data from a major stock exchange. We are hoping that we can either acquire this real time data through David Lariviere or else leverage access to it by proposing our project to the NASDAQ. Otherwise, we will use sample market data as previously recorded. The data will be received via Ethernet using the Solarflare board's SFA6900 ApplicationOnload Engine.

IV. Outputs

For this project we will output a broadcast of the market data we have stored at any given moment. The data will output via the PCI of the AoE board to a Linux machine which will be loaded with an x86 software to handle the updates. The software will display the current state of the book we are keeping on the board and will update in real time. We are also considering the option to search for multiple stocks' data depending on the progress we make in hardware.

V. Algorithm

From a very high level, our algorithm will include:

1. Input data from stream.
2. Decode ethernet packet.
3. Decode received data.
4. Store in hash map.

5. Output data to Linux client.
6. Display data using x86 software console.

VI. Hardware/Software Responsibilities

Because this project aims to innovate upon a classical software application in hardware, our project will be very hardware-focused. We will use software strictly to monitor the data structure and process the user interface. The remaining workload will be handled by the FPGA (Stratix V) and the rest of the Solarflare card.