Project Lab Proposal Form

The University of Chicago's Master of Science Program in Financial Mathematics (MSFM) partners with companies through the Project Lab course (FINM 36000), to provide students with hands-on research experiences, putting knowledge gained from the classroom into practice.

You can find additional details, along with FAQs within the Project Lab Company Guide: https://uchicago.app.box.com/s/p4d37a28doxsh8xlm6dap2lwt2gd8f33

If you would like to propose multiple projects, please complete a new proposal form for each project.

If you have any questions, please contact Sue Clark at **sueclark@uchicago.edu**.

Email *

agerasev@dvtrading.co

Company name (and division/department/desk, if applicable): *

DV Trading

Physical address of organization: *

425 S Financial Place Ste 2800, Chicago, IL 60605

URL and/or description of organization: *

www.dvtrading.co

How many teams is your organization sponsoring for this project? *

1

Up to six students may be assigned to your team. If you would prefer a different number, please indicate so here.

Company Representative name: *

Alex Gerasev

Company Representative title: *

Trader

Company Representative email address *

agerasev@dvtrading.co

If applicable, additional Company Representative name(s), title(s), and email(s):

Are any representatives also alumni of the University of Chicago? If yes, please indicate their name and applicable degree program:

Yes - Alex Gerasev MSFM 2010

Aside from the initial briefing meeting and the final presentation, how often will the team meet with the representative(s) to deliver progress reports and receive advice? *

Weekly

Will such discussions be face-to-face, or by some form of teleconferencing? If face-to-face, will the meetings be at your company site, on-campus, or some other arrangement? *

Prefer face-to-face on campus; occasionally teleconferencing

Will the team members be required to sign non-disclosure agreements? *

No

Project topic/title: *

High-performance analysis of market-by-order market data

Background motivation of the project:

Market-by-order (MBO) market data is commonly used by modern exchanges where information (such as "add", "delete", "modify", etc.) about every individual order is published and recipient is responsible for building limit order books. Exchanges typically implement custom binary protocols on top of UDP multicast for efficiency purposes. Processing MBO data feeds is often a challenging task and requires strong programming as well as analytical skills. This is true for both real-time consumption into trading strategies, where the goal is to minimize latency associated with each individual event, as well as historical research and analysis where performance directly affects execution speed as number of messages to be processed is often in billions.

When it comes to programming, there is often no "one size its all" solution and potential areas for research and optimizations are numerous. What data structures should be used to store orders? Should standard constructs such as those provided by STL be used or should we make our own? Should all orders be placed in a single storage or split by symbol? Are there efficient hashing techniques that can be applied to speed up lookup of orders by order ID? What compiler optimization flags should be used?

Furthermore, MBO feed processing may require programming techniques very different from those used in other areas. While other applications may be concerned with executable size, memory consumption, compilation time, or code readability, in trading there is one thing and one thing only that matters: execution speed. This often results in code which looks very different from what is suggested by Stack Overflow or ChatGPT.

Skills required (e.g. programming languages/software packages/statistical models or tools, etc.): *

At least intermediate level of C++ required. Modern C++ (at least C++11) preferred. Some experience with programming high-performance or low-latency systems and associated optimization techniques would be very helpful but is not required. Familiarity with Unix-like system (which includes Mac OSX) is required. Participants are strongly encouraged to use UNIX, Linux, or Mac OSX computer for research and development.

Working knowledge of Git (either standalone or together with e.g. Github) is required.

Participants with beginner level of C++ will NOT be able to successfully complete this Project Lab.

Project objectives or questions to be investigated: *

1. Set up "performance lab" virtual environment where performance metrics may accurately captured. These may include: number of messages processed per unit of time; time to process a certain historical market data file (e.g. one day or one week); end-to-end latency of processing an individual message or event.
2. Develop specific techniques for improving performance of MBO feed processing.
3. Evaluate performance improvements associated with each suggested technique and present justification based on empirical data.
4. Using the MBO feed processor, establish framework for finding trading signals in historical MBO market data.

Implementation details:

During initial presentation participants will be provided with overview of the MBO feed protocol including live examples and views of live MBO order books; overview of the historical PCAP data file; overview of the "starter kit" C++ framework; and other relevant artifacts.

An objective will be set for each week following the initial presentation. Participants will be expected to do relevant research, implement various approaches (typically by extending the "starter kit" framework), perform necessary measurements, arrive to conclusions, and present their findings at the following weekly meeting.

Project lab participants are expected to work as a team. Single set of materials, rather than different ideas coming from individual team members, should be presented each week.

Example of weekly objective:

Feed processor needs to maintain many counters for various types of events, e.g. number of messages processed of each type, protocol sequence numbers, etc. What programming constructs may be used to achieve the best possible performance while (hopefully) still offering some code readability?

Suggest different approaches, implement them, measure and analyze results, provide observations, and make final recommendation.

Reference materials, if any:

This project lab participants will be provided with the following:

1. Full day (or more) of historical MBO market data captured at one of the leading U.S. futures exchanges. Data will be provided in PCAP ("packet capture") format and has been accurately captured by DV Trading using its low-latency infrastructure co-located at the exchange, with nanosecond resolution.
2. Full specifications for the binary MBO protocol published by the Exchange.
3. C++ "starter kit" with basic framework for reading the PCAP, parsing individual messages per the Exchange protocol, and displaying some basic information about each message. Project Lab participants will be expected to build on top of this "starter kit" both in terms of features and performance improvements.
4. Additional useful artifacts such as Wireshark plugin so that data can be viewed in Wireshark software; useful configuration files; etc.

Students may add this project lab to their resume, including the company name and a general description of their research (abiding by any applicable NDA). We advise they use the following format:

Company Name

Quantitative Researcher - University of Chicago Project Lab

General description of research

Please note below if you prefer something different.

Any additional comments: