

Тестирование графического интерфейса трейдинговых терминалов в условиях высокочастотной торговли

Иван Бобров¹, Алексей Зверев²

¹ООО «Инновационные Трейдинговые Системы», Россия, 156013, г. Кострома, ул. Ленина, 20

Ivan.Bobrov@exactpro.com

²Exactpro Systems LLC, 4040 Civic Center Drive, Suite 200, San Rafael, CA 94903, USA

Alexey.Zverev@exactpro.com

Аннотация. При работе систем высокочастотного трейдинга в состоянии максимальной нагрузки количество изменений котировок может достигать нескольких тысяч в секунду. Для слежения за рынком на экранах терминалов применяются специальные графические средства, реализованные в виде программного обеспечения. Тестирование такого программного обеспечения представляет собой специфическую задачу. В статье рассматривается метод проверки функциональности графического интерфейса трейдинговых систем. Для тестирования предлагается сохранять графическое представление областей интерфейса в момент их перерисовки операционной системой. Полученные таким образом данные в оцифрованном виде сравниваются с данными, полученными из независимого источника котировок. В статье описывается реализация предложенного метода и анализируются результаты с точки зрения применимости в условиях реальной торговли.

1 Введение

В последние годы высокочастотная торговля занимает большую часть торгов на соответствующих электронных площадках. По данным РТС в 2010 году на долю торговых роботов в обороте на срочном рынке РТС FORBS приходилось примерно 50%, а их доля в общем количестве заявок в определенные моменты достигала 90% [1]. Системы, ориентированные на высокочастотную торговлю, многокомпонентны и строятся из расчёта максимальной отдачи и надёжности. Основными компонентами, напрямую используемыми пользователями, являются торговые терминалы и шлюзы, находящиеся под управлением специализированных протоколов, таких как FIX, NATIVE [2]. За информацию, на базе которой эти компоненты формируют представление о рынке, отвечает компонент, являющийся важной составляющей любой торговой платформы и

отвечающий за поставку котировок [3]. Поставщик котировок предоставляет котировки и другую финансовую информацию в режиме реального времени, что позволяет получателям обрабатывать и представлять её конечному пользователю в том или ином виде.

2 Торговый терминал и шлюз

Торговый терминал – это программное обеспечение, позволяющее трейдеру в режиме реального времени получать всю необходимую информацию о ходе торгов на мировых биржевых и внебиржевых рынках, а также анализировать полученные данные посредством встроенных в торговый терминал инструментов и делать выводы, на основе которых совершать торговые сделки. [4].

Основным отличием торгового терминала является графическая оболочка, посредством которой происходит отображение данных на экран. В случае с подключением через шлюз, все данные, как правило, передаются в формате протокола, по которому совершено подключение (см. рисунок 1). В итоге сам торговый терминал может быть подключен непосредственно через подобный протокол к рынку.

И шлюз и терминал, получают данные, основанные на потоке, распространяемом компонентом поставщика котировок. В случае со шлюзом пользователь самостоятельно производит расшифровку полученного пакета данных, а в терминале вся информация представляется в удобочитаемом виде.

На сегодняшний день количество заявок, обрабатываемых биржей, в течение торговой сессии исчисляется миллионами. Особенностью же HFT логики является высокоскоростной обмен сообщениями с рынком в зависимости от модели торгового робота. В ходе эволюции средств связи и коммуникации скорость обработки запросов значительно возросла, а время ответа системы на запрос стало исчисляться в микросекундах [5].

Таким образом, в момент максимальной нагрузки рынка торговый терминал получает информацию со скоростью, с которой вывод на экран делает невозможным попытки визуально отследить все изменения, происходящие в окне клиента, вследствие чего пользователь не сможет дать адекватную оценку корректности отображаемой информации.

При приёме данных через шлюз весь поток информации будет сохранен в виде сообщений, полученных в рамках установленной сессии и доступных для анализа, что позволяет выполнить их проверку на соответствие спецификации без каких-либо серьёзных потерь и в любой удобный для этого момент времени.

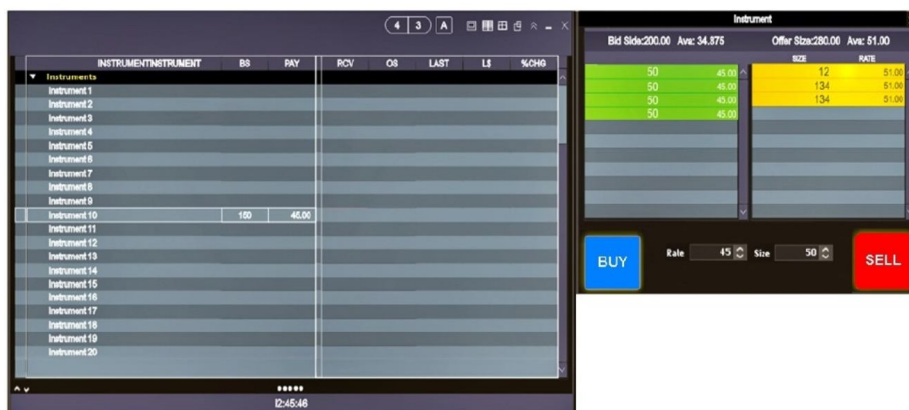


Рис. 1. Пример графического интерфейса трейдингового терминала.

3 Симуляция нагрузки

Для тестирования системы в состоянии высокочастотного трейдинга в первую очередь необходимо создать условия, совпадающие с представлениями о высокой нагрузке. Для создания подобных условий может быть использован инструмент «Генератор нагрузки» (англ. *LoadInjector*), разработанный компанией Exactpro Systems LLC. Инструмент позволяет согласно сконфигурированному сценарию создавать и поддерживать продолжительную отправку и приём большого количества сообщений, доводя таким образом торговую систему до необходимого уровня нагрузки.

Объектом тестирования выступает графическая оболочка торгового клиента, из которой необходимо получить поток данных для анализа. В качестве эталона для сравнения в ходе анализа может быть использован шлюз, подписка через который на мониторинг отправки котировок позволит получить достаточно надёжную картину происходящего на протяжении всего периода теста.

Для этого организуется подключение по протоколу шлюза с последующей подпиской на необходимые обновления, которые будут сохранены в виде набора сообщений в формате используемого протокола обмена данными.

Тесты проводились при помощи симуляции исторической маркет даты, доступной для общего доступа на сайте histdata.com, воспроизводимой с различной скоростью.

4 Запись данных из графического интерфейса

Требуется сохранить аналогичный поток изменений в окне торгового терминала. Для осуществления этого есть несколько возможных способов.

Первый способ - запись области экрана в видеофайл посредством специальных утилит, таких как CamStudio [6], с последующей обработкой и

конвертацией в набор изображений, отражающих блок, относящийся к содержимому книги заявок. В этом случае потребуется дополнительно декодировать данные из полученных изображений, представив их в виде буквенно-числового представления содержимого каждой стороны книги заявок для возможности сравнения. Также ввиду ограничений скорости работы подобных программ и записи видео с конечной картинке экрана есть риск потери достаточно большого количества изменений. Программа снимает изображение непосредственно с экранной картинке, в то время как драйвер видеоприбора может послать гораздо большее количество кадров, которые в ходе наложения могут перекрыть изменения, произошедшие в предыдущем кадре. Таким образом, при записи видео со скоростью 200 кадров в секунду весь выход за пределы такой скорости со стороны видеодрайвера ведёт к потере потенциально пригодных для анализа данных.

Второй способ – это прямой доступ к объектам формы терминала. Для этого средствами операционной системы потребуется установить, какие объекты расположены в окне, как они упорядочены, какой имеют тип и какие параметры этих объектов позволят прочесть их текущее состояние и содержимое. Так с установленным интервалом можно будет считывать состояние нужных компонентов, а также сохранять состояния книги. Существенным минусом данного способа будет высокая нагрузка на процессор компьютера, на котором будет расположена такая программа и терминал, в результате чего снизится производительность самого терминала, что делает тест отдаленным от реальных условий.

Третьим способом является перехват всех попыток изменить внешнее представление окна терминала самой операционной системой. Таким образом мы сможем гарантировать, что все попытки перерисовки будут сохранены, а на выходе будет массив изображений, доступный для анализа и конвертирования в читаемый набор данных.

Суммируя все вышесказанное, мы видим, что первый способ наименее интересен ввиду присутствия риска потери данных, что не позволит сделать объективными результаты анализа. Второй способ весьма специфичен и может быть использован лишь в ограниченном наборе графических интерфейсов, и ввиду того, что каждый из них может быть реализован по-разному и на разных платформах, возникает необходимость индивидуального подхода и индивидуального алгоритма. Также риск сокращения производительности самого приложения торгового терминала ввиду параллельной работы достаточно сложной реализации по считыванию изменяемых данных, делает недопустимым использование такого метода в текущем исследовании.

Наиболее эффективным остается метод с перехватом действий перерисовки, так как он практически исключает потери и будет работать по менее воздействию на терминал алгоритму. Однако это, в свою очередь, потребует максимально быстродействующего накопителя, так как в ходе теста все изменения должны быть записаны в графические файлы типа BITMAP. Сжатие исключается с целью снизить затраты центрального процессора и минимизировать потери ресурсов.

На данном этапе алгоритм применим для большинства торговых клиентов, содержимое книжки которых отражается на экране терминала. Однако стоит учитывать, что реализация отображения не одинакова в разных клиентах.

Объект тестирования и реализованный алгоритм узко специализированны относительно объекта тестирования и для применения алгоритма против другого клиента он потребует переработки в зависимости от того как будет представлена информация на экране терминала. Таким образом основных изменений потребует часть преобразования снятой графической информации.

5 Microsoft Detours

В качестве инструмента для реализации выбранного метода, используется библиотека Microsoft Detours [7]. Данная библиотека позволяет устанавливать перехватчики стандартных событий системы и использовать собственную процедуру, предваряющую дальнейшее выполнение (Рисунок 2.). В случае с торговым терминалом, перехватчик устанавливается на событие перерисовки любого графического элемента экранной формы торгового терминала.

Процедура сохраняет графическое представление переданной области, которая будет изменена, в виде файла, в имени которого сохраняется идентификатор окна, штамп даты и времени, а также координаты, по которым происходит перерисовка. На основании полученного набора файлов делается выборка по зонам, в которых отображается информация. Особенностью различных графических интерфейсов трейдинговых терминалов является их способность одновременно показывать ситуацию сразу по нескольким торговым инструментам в отдельно открытых окнах или наоборот - показывать лишь один инструмент строго в одном окне интерфейса. Для записи в таком случае может быть использовано как одно окно одного инструмента, так и весь набор окон с привязанными к ним инструментами.

Следующим шагом является преобразование графического представления в читаемый массив значений. На основании полученных значений, а также значений из подписки к шлюзу, строятся два набора книжек [8] с определенным шагом времени (t). Во внимание принимается максимальное количество заявок, отображаемых в торговом терминале.

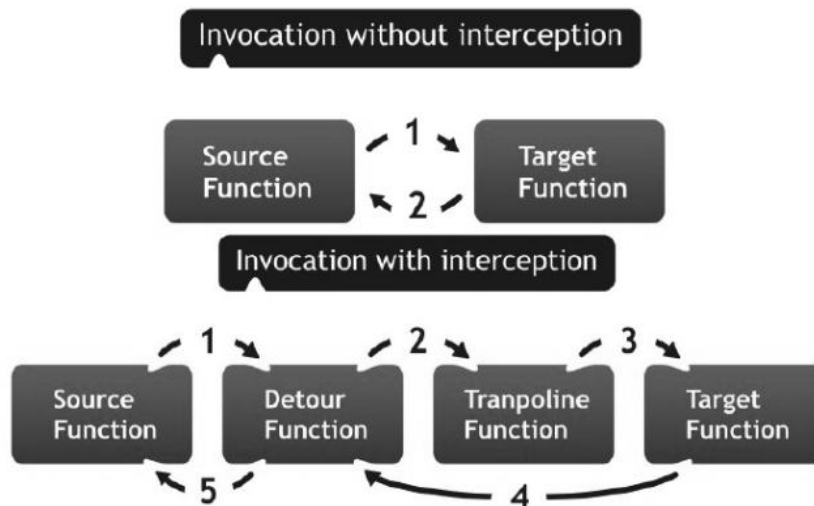


Рис. 2. Схема работы перехватчика из библиотеки Microsoft Detours.

6 Сравнение данных

Полученные наборы состояний сравниваются между собой и оценивается разница между ними. Набор данных от шлюза:

$$GW() = GW(t_1), GW(t_2), GW(t_3), \dots, GW(t_N) \quad (1)$$

Набор данных от графического интерфейса:

$$SC() = SC(t_1), SC(t_2), SC(t_3), \dots, SC(t_N) \quad (2)$$

Необходимо учесть, что количество данных на единицу времени со стороны шлюза допустимо больше, чем со стороны графического интерфейса, ввиду того, что графический адаптер не способен отразить все изменения книжки, если они превышают его характеристики по записи в видеопамять в момент времени t .

Сравнение происходит по двум признакам: полное попадание и полное непопадание. Для каждого случая производится подсчёт разницы во времени между найденными соответствиями.

Нарушение порядка даже заявок на книжке принимается как полное несоответствие книжек. Таким образом, на выходе получаем график, подобный представленному на рисунке 3.

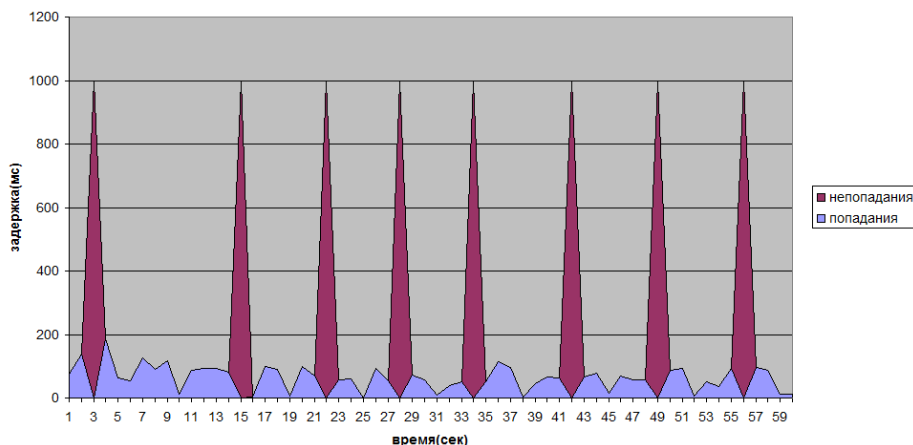


Рис. 3. Пример графика совпадений книжек на одной минуте теста.

График позволяет оценить динамику временной разницы между соответствиями книжек от эталонного источника и графического интерфейса, а также оценить потери и корректность отображаемой информации. На основании такой оценки при детальном рассмотрении конкретных приложений можно модифицировать алгоритмы нагрузочного сценария, что даст возможность усилить обнаруженные негативные эффекты, будь то несовпадение книжек или серьезная разница во времени между соответствиями.

Принимается во внимание, что основная фаза тестирования происходит на основании предзаписанного сценария нагрузки, который может быть заменен сконфигурированным тестом, в котором будут учтены типы сообщений и их частота на проблемных участках анализа.

На графике отражена зависимость отставания книжки, отраженной в трейдинговом терминале, от информации, полученной через предоставляющий информацию о котировках (market data) шлюз. По оси абсцисс откладывается время наблюдения в секунду, а по оси ординат - задержка в миллисекундах между временем котировки и временем получения соответствующей книжки в трейдинговом терминале. При помощи искусственных точек с задержкой 1000 мс, мы отметили на этом графике точки полного непопадания.

По кривой с попаданием мы можем анализировать задержки между получением событий через электронные интерфейсы и отрисовкой их в пользовательском интерфейсе. Эта важная информация позволяет конечным пользователям оценивать актуальность и своевременность информации, отражаемой на экране. График с непопаданием показывают, насколько часто графический интерфейс отражает книжку, полностью не соответствующую реальности. Анализ этих графиков лучше проводить, рассматривая плотность точек на графике "непопадания". Например, если "непопадания" встречаются редко и длительность непопадания ограничена, это означает, что, если сразу после "непопадания" имеется "попадание", то, с точки зрения конечного пользователя, искаженная книжка будет незаметна, поскольку в течение

короткого интервала она будет перерисована другой, правильной, книжкой. Однако, с другой стороны, если плотность "непопаданий" велика, то возрастает вероятность того, что пользователь увидит некорректную книжку, и такое поведение следует считать существенным дефектом пользовательского интерфейса.

На основе предложенного подхода (Рисунок 4) был разработан инструмент, позволяющий перехватывать все изменения графического интерфейса, а также дополнительные инструменты для распознавания графических данных, построения книг заявок и сравнения двух наборов книг между собой [9].

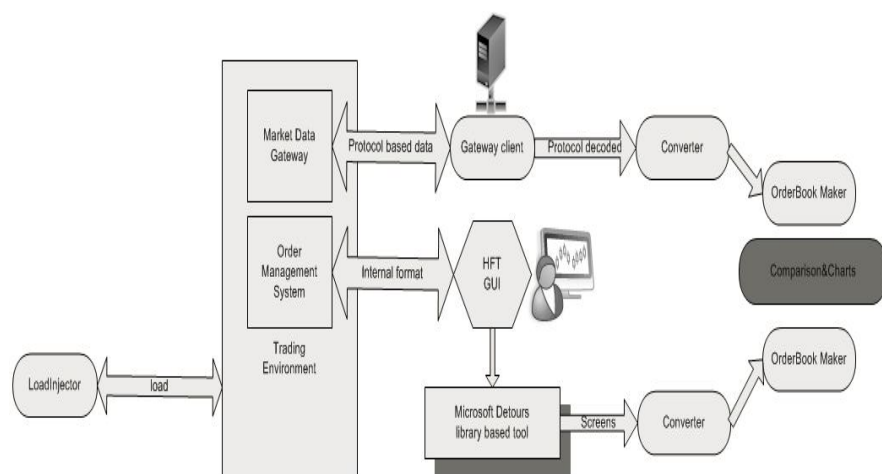


Рис. 4. Графическая схема метода, разработанного на базе библиотеки Microsoft Detours

7 Заключение

В данной статье мы показали, что имеется техническая возможность валидации активного потока котировок, демонстрируемого пользователю через элементы графического интерфейса. Мы разработали инструменты, реализующие представленную методику снятия данных. В результате построенный набор инструментов позволяет эффективно анализировать графическую выдачу и сравнивать её с независимым цифровым потоком, а также вычислять задержки и потенциальные несовпадения двух потоков.

Источники

1. Slon.ru - Деловые новости и блоги // [Электронный ресурс]. – Режим доступа http://slon.ru/economics/lyudi_luchshe_robotov-970172.shtml
2. Официальный сайт FIXprotocol // [Электронный ресурс] - Режим доступа <http://www.fixprotocol.org/>
3. Wikipedia // [Электронный ресурс]. - Режим доступа http://en.wikipedia.org/wiki/Market_data
4. Форекс журнал для трейдеров // [Электронный ресурс]. - <http://fortrader.ru/forex-terminals/>
5. Leo King.: London Stock Exchange smashes world record trade speed with Linux. / Computerworld UK // [Электронный ресурс]. <http://www.computerworlduk.com/news/networking/3244936/london-stock-exchange-smashes-world-record-trade-speed-with-linux/>
6. Официальный сайт CamStudio // [Электронный ресурс]. - Режим доступа <http://camstudio.org>
7. Dejan Lukan.: API Hooking with Microsoft Detours / INFOSEC Institute Resources // [Электронный ресурс]. - Режим доступа <http://resources.infosecinstitute.com/api-hooking-detours/>
8. Wikipedia // [Электронный ресурс]. - Режим доступа http://en.wikipedia.org/wiki/Order_book_%28trading%29
9. Zverev,A., Bobrov,I, Pryadkina,N.: Testing of HFT GUI. / ExTENT conference // [Электронный ресурс]. – Режим доступа <http://www.slideshare.net/extentconf/extent3-exactpro-testingofhftgui-12944204>