The Latency Difference between Depth of Book and BBO Feeds

NASDAQ, Arca, NYSE and NYSE MKT
August 8, 2016

Michael Lehr
Co-Founder, COO
MayStreet LLC
michael@maystreet.com
# Table of Contents

Abstract .......................................................................................................................... 4
Purpose ............................................................................................................................ 4
Exchanges and Feeds ...................................................................................................... 5
Terminology .................................................................................................................... 6
Methodology .................................................................................................................... 6
  NASDAQ: Matching QBBO with TotalView ................................................................. 7
  Arca: Matching XDP Book with XDP BBO ................................................................. 7
  NYSE & NYSE Mkt: Matching OpenBook Ultra with XDP BBO ............................... 8
  Files and File Formats ............................................................................................... 8
Data Availability from MayStreet ..................................................................................... 9
Results ............................................................................................................................ 10
  NASDAQ ....................................................................................................................... 10
  Arca ............................................................................................................................... 12
  NYSE ............................................................................................................................ 16
  NYSE Mkt ..................................................................................................................... 18
In Depth Observations ..................................................................................................... 20
  NYSE: Significantly different packaging of messages in the BBO and Depth Feeds ....... 20
  NYSE: Open Book Ultra vs XDP NYSE BBO: Session numbers ............................... 22
Solution .......................................................................................................................... 24
  Example using Bellport™ ........................................................................................... 24
  Totalview Raw Feed example ..................................................................................... 25
Discussion ....................................................................................................................... 27
  Key Points ..................................................................................................................... 27
  NASDAQ ....................................................................................................................... 27
  Arca ............................................................................................................................... 27
  NYSE and NYSE Mkt ................................................................................................. 28
Future Work ................................................................................................................... 28
About the Author .......................................................................................................... 29
**Figures**

Figure 1: Data Paths from Matching Engine to Users................................................................. 5  
Figure 2: TotalView vs. QBBO .................................................................................................. 11  
Figure 3: XDP Arca Book vs. XDP Arca BBO........................................................................... 13  
Figure 4: XDP Arca Book - Example Latency Spike by Session................................................ 14  
Figure 5: Arca - Multiple Latency Groupings within Single Session........................................... 15  
Figure 6: OpenBook Ultra vs. XDP NYSE BBO ......................................................................... 17  
Figure 7: OpenBook Ultra Mkt vs. XDP NYSE BBO ................................................................. 19  

**Tables**

Table 1: Depth and BBO feeds....................................................................................................... 5  
Table 2: Terminology ................................................................................................................... 6  
Table 3: BBO & Depth with Odd Lot Order Example................................................................... 8  
Table 4: Data Availability ........................................................................................................... 9  
Table 5: TotalView vs QBBO ..................................................................................................... 10  
Table 6: Arca percentiles pre- and post-6/20 performance update........................................... 12  
Table 7: XDP Arca Book vs. XDP Arca BBO ............................................................................ 13  
Table 8: OpenBook Ultra vs., XDP NYSE BBO ....................................................................... 17  
Table 9: OpenBook Ultra Mkt vs. XDP NYSE BBO .................................................................. 18  
Table 10: OpenBook Ultra: Sample latency per session .............................................................. 23
Abstract

NYSE, NASDAQ and Arca depth-of-book feeds are significantly more expensive than their best bid and offer (BBO) variants. Many firms are cost sensitive and commonly ask “what is the latency difference between the depth-of-book, BBO and SIP feeds?” in order to make a decision as to the cost/benefit of the different feeds. This technical paper will show the distinct difference between several depth-of-book feeds and their corresponding BBO feeds, and show that the BBO feeds are often faster. We propose that users can achieve a less latent view of the true top-of-the-book by arbitrating the BBO with the depth feed.

All derived data in this study is freely available from MayStreet upon request.

Purpose

Many financial firms need timely market data to make important trading decisions. For some firms, knowing the full order book is necessary. Some example of users that requires full depth order books is routing and managing ISO orders, understanding buy or sell pressure, or knowing if a product is lightly quoted.

The purpose of a market data feed handler is to recreate the state of the exchange in a client machine which will enable a user to incorporate the order book state into their business logic. The depth-of-book feeds provide an order-by-order view into the state of the matching engine, or as in OpenBook Ultra, a level-based view of the order book. The BBO feeds only provide the aggregate quantity at the highest bid price and lowest offer price, and in some cases may not include odd lot orders. The underlying matching engines that provide data are identical for both depth and BBO feeds. Since they represent the same information, then we can merge the data between the two feeds.

To incorporate timely information, users spend significant resources to reduce latency. Typically this may include colocation choices, low latency networks, fast NIC cards, fast servers, lean and fast software, or hardware-based solutions. Developers can spend weeks or months reducing processing and decision time as much as possible.

In today’s trading environment, 100s of nanoseconds are relevant for many firms. If a user is only considering internal processing time and assumes that the data entering the firm’s network is starting at time zero, they could overlook an important reality. Time zero is actually at the exchange in the matching engine. The data goes through multiple software systems to get to the users (see Figure 1: Data Paths from Matching Engine to Users). Since these are independent software systems managed by the exchange, by combining all flavors of market
data a user can achieve their goal of timely replicating the exchange state with reduced development, hardware and exchange costs.

![Diagram](image)

**Figure 1: Data Paths from Matching Engine to Users**

The visual above shows roughly the different data paths from the matching engine to the users. Once the data leaves the matching engine (or set of servers just after the matching engine), it enters independent software systems that will eventually enter the user’s network.

### Exchanges and Feeds

<table>
<thead>
<tr>
<th>Exchange</th>
<th>Depth</th>
<th>BBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nasdaq</td>
<td>TotalView</td>
<td>QBBO</td>
</tr>
<tr>
<td>Arca</td>
<td>XDP Book</td>
<td>XDP BBO</td>
</tr>
<tr>
<td>NYSE</td>
<td>OpenBook Ultra</td>
<td>XDP BBO</td>
</tr>
<tr>
<td>NYSE MKT</td>
<td>OpenBook Ultra</td>
<td>XDP BBO</td>
</tr>
</tbody>
</table>

Table 1: Depth and BBO feeds
**Terminology**

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed</td>
<td>A format of data coming out of the exchange. An exchange may have multiple feeds that convey the same exchange state. Ex: XDP NYSE Integrated, OpenBook Ultra, or XDP NYSE BBO, XDP NYSE Trades all represent NYSE</td>
</tr>
<tr>
<td>Session</td>
<td>A single stream of data within a feed. There can be multiple sessions within a feed which are commonly broken up by symbol range, or possibly by a mapping to exchange matching engines.</td>
</tr>
<tr>
<td>Channel</td>
<td>A duplicate copy of the same session. This will be a single multicast group if distributed over UDP multicast. The data must represent the same data, but need not always be packaged identically into the same packets.</td>
</tr>
<tr>
<td>Top of Book</td>
<td>The highest bid price &amp; quantity, and also the lowest offer price &amp; quantity.</td>
</tr>
<tr>
<td>Odd Lot Order</td>
<td>An order that is not of lot size. Typical lot size is 100 shares.</td>
</tr>
</tbody>
</table>

Table 2: Terminology

**Methodology**

MayStreet uses advanced lossless, high-precision, hardware-time stamped packet capture to collect raw exchange multicast data. This data is stored in nanosecond pcap files. Capture of both the BBO and the depth feed were on the same capture server. Since we were comparing two streams of data going over the same networks and captured over the same ports with hardware timestamps, this analysis would still be valid even if we did not have correct timestamps. We are NOT comparing the timestamp at the exchange to the timestamp on the machine. We are comparing only timestamps in the capture machine to itself.

The pcap data is fed through an application that utilizes the MayStreet Bellport™ feed handlers to build the book from the depth feed and parse the BBO feed. Ticks are stored in application memory and matched by comparing the timestamp at the matching engine, bid size, bid price ask size and ask price.

For our analysis, we subtracted the depth-of-book feed hardware packet timestamp from the matched BBO feed’s hardware packet timestamp.

\[
Diff = T_{S_{bbo}} - T_{S_{depth}}
\]

For the results below, a positive number means that the depth-feed packet came first. A negative number means that the BBO packet came first.
**NASDAQ: Matching QBBO with TotalView**

In the QBBO feed, we used the Quotation Message for all book updates. In this message, the interesting fields are: TimeStamp; NASDAQ Best Bid Price; NASDAQ Best Bid Size; NASDAQ Best Offer Price; and NASDAQ Best Offer Size. In TotalView, we used the Add Order Messages, Modify Order and Cancel Order Messages to build the book.

Using the built-order book, we compared the highest bid price, highest bid size, lowest offer price or lowest offer size changed with the QBBO feed. We then searched for a direct comparison with the NASDAQ Best Bid and NASDAQ Best Offer in the QBBO feed. Every top-of-book change in the depth feed matched with a corresponding change in the BBO feed.

**Arca: Matching XDP Book with XDP BBO**

XDP BBO does not output odd-lot orders. This means there are updates on BBO that are not on the depth feed as well as updates on the depths that are not on the BBO. Further, there are many updates that are on the depth feed that aren’t on BBO because changes are less than a full lot size and therefore do not cause an update to the reported BBO.

Both the BBO and Book messages have timestamps from the exchange that are used to correlate updates. There is a time-reference message that reports the second and then the Book messages and the Quote Message have a SourceTimeNS field.

In this report, we matched only the exact matches, i.e., only full lot sizes. This resulted in ~ 50% less matches but results were similar when rounded down.

Also, we decided to match on symbol, bid price, bid quantity, ask price, ask quantity and source time. This is last reference time + SourceTimeNS to ensure that we only include true matches. True negatives are acceptable, but false positives can sway our results, which is why we require full matches.
**Table 3: BBO & Depth with Odd Lot Order Example**

<table>
<thead>
<tr>
<th>Time</th>
<th>Depth Bid</th>
<th>Depth Offer</th>
<th>BBO Bid</th>
<th>BBO Offer</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150 @ 10.00</td>
<td>200 @ 10.01</td>
<td>100 @ 10.00</td>
<td>200 @ 10.01</td>
<td>BBO &amp; Depth different</td>
</tr>
<tr>
<td>1</td>
<td>50 @ 10.00</td>
<td>200 @ 10.01</td>
<td>2100 @ 9.99</td>
<td>200 @ 10.01</td>
<td>Tick on 2nd level. Only seen on top of BBO feed, but is available on depth feed.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2000 @ 9.99</td>
<td>200 @ 10.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>75 @ 10.00</td>
<td>200 @ 10.01</td>
<td></td>
<td></td>
<td>25 added to 10.00. Still under 100 so BBO not affected</td>
</tr>
<tr>
<td>4</td>
<td>2000 @ 9.99</td>
<td>200 @ 10.01</td>
<td></td>
<td>10.00 bid traded. BBO = Depth</td>
<td></td>
</tr>
</tbody>
</table>

**NYSE & NYSE Mkt: Matching OpenBook Ultra with XDP BBO**

Matching for OpenBook Ultra with XDP NYSE BBO is similar to Arca. Please refer above to “Table 3: BBO & Depth with Odd Lot Order Example” to see how the order books in the BBO and depth feed may not match. One difference between XDP Book and OpenBook Ultra is the timestamp resolution. OpenBook Ultra only has microsecond resolution as opposed to nanosecond resolution as with XDP Book & XDP BBO. However, the values in XDP BBO for NYSE has all 0’s for the nanoseconds, and so they match exactly i.e.: in OpenBook we’ll see 123456 microseconds, but in XDP BBO we’ll see 123456000 nanoseconds.

We decided to match on symbol, bid price, bid quantity, ask price, ask quantity and source time, which is last reference time + SourceTimeNS, again ensuring as stated above that we only include true matches. True negatives are acceptable, but false positives can sway our results, which is why we require full matches.

**Files and File Formats**

We match each tick individually into a large single raw match file and using python, we sum all of that data into a single summary stats file with the format below. In this summary file, we have the session of the BBO Feed (B0, B1, B2) and then the session of the Depth Feed (D0, D1, D2).
### Data Availability from MayStreet

<table>
<thead>
<tr>
<th>Data</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>High quality pcap</td>
<td>Fees apply. For certain feeds, data access rights, and exchange fees may also apply</td>
</tr>
<tr>
<td>Raw match csv file</td>
<td>Free. Available upon request.</td>
</tr>
<tr>
<td>Summary stats csv file</td>
<td>Free. Available upon request.</td>
</tr>
</tbody>
</table>

Table 4: Data Availability
Results

NASDAQ

Table 5: TotalView vs QBBO below lists the average difference for the delay between QBBO and TotalView. The percent is the percentile that the TotalView is faster than QBBO. A negative number means that QBBO is faster than TotalView. This is the packet receipt time of the BBO minus packet receipt of the depth which is captured on the same server.

TotalView is distributed as a single, multicast channel. QBBO is distributed as three multicast channels: NYSE Issues, Amex Issues and NASDAQ Issues.

- **TotalView is faster than QBBO around 92% of the time.**
- **QBBO is 600ns faster 5% of the time.**
- **QBBO is 4,300ns faster 1% of the time**
- **QBBO is 13,000ns faster 0.1% of the time.**

<table>
<thead>
<tr>
<th>Date</th>
<th>99.9%</th>
<th>99%</th>
<th>95%</th>
<th>90%</th>
<th>80%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160502</td>
<td>-13,196</td>
<td>-3,970</td>
<td>-410</td>
<td>316</td>
<td>936</td>
<td>2,450</td>
</tr>
<tr>
<td>20160503</td>
<td>-11,304</td>
<td>-3,580</td>
<td>-270</td>
<td>420</td>
<td>986</td>
<td>2,444</td>
</tr>
<tr>
<td>20160504</td>
<td>-12,934</td>
<td>-3,926</td>
<td>-400</td>
<td>330</td>
<td>964</td>
<td>2,510</td>
</tr>
<tr>
<td>20160505</td>
<td>-12,190</td>
<td>-3,764</td>
<td>-304</td>
<td>400</td>
<td>970</td>
<td>2,390</td>
</tr>
<tr>
<td>20160506</td>
<td>-12,704</td>
<td>-3,886</td>
<td>-370</td>
<td>350</td>
<td>976</td>
<td>2,560</td>
</tr>
<tr>
<td>20160509</td>
<td>-13,514</td>
<td>-4,547</td>
<td>-626</td>
<td>246</td>
<td>910</td>
<td>2,360</td>
</tr>
<tr>
<td>20160510</td>
<td>-13,770</td>
<td>-5,403</td>
<td>-994</td>
<td>113</td>
<td>720</td>
<td>1,976</td>
</tr>
<tr>
<td>20160511</td>
<td>-13,046</td>
<td>-4,650</td>
<td>-754</td>
<td>164</td>
<td>773</td>
<td>2,050</td>
</tr>
<tr>
<td>20160512</td>
<td>-12,394</td>
<td>-3,644</td>
<td>-396</td>
<td>340</td>
<td>974</td>
<td>2,494</td>
</tr>
<tr>
<td>20160513</td>
<td>-12,717</td>
<td>-3,750</td>
<td>-456</td>
<td>317</td>
<td>960</td>
<td>2,526</td>
</tr>
<tr>
<td>20160516</td>
<td>-12,756</td>
<td>-3,977</td>
<td>-504</td>
<td>290</td>
<td>930</td>
<td>2,400</td>
</tr>
<tr>
<td>20160517</td>
<td>-12,256</td>
<td>-3,526</td>
<td>-410</td>
<td>326</td>
<td>956</td>
<td>2,497</td>
</tr>
<tr>
<td>20160518</td>
<td>-10,720</td>
<td>-3,263</td>
<td>-360</td>
<td>334</td>
<td>916</td>
<td>2,336</td>
</tr>
<tr>
<td>20160519</td>
<td>-11,350</td>
<td>-3,724</td>
<td>-480</td>
<td>274</td>
<td>874</td>
<td>2,290</td>
</tr>
<tr>
<td>20160520</td>
<td>-13,370</td>
<td>-4,336</td>
<td>-554</td>
<td>270</td>
<td>910</td>
<td>2,386</td>
</tr>
<tr>
<td>20160523</td>
<td>-13,940</td>
<td>-5,410</td>
<td>-850</td>
<td>150</td>
<td>734</td>
<td>2,004</td>
</tr>
<tr>
<td>20160524</td>
<td>-14,107</td>
<td>-5,516</td>
<td>-934</td>
<td>130</td>
<td>710</td>
<td>2,000</td>
</tr>
<tr>
<td>20160525</td>
<td>-14,470</td>
<td>-5,346</td>
<td>-1,070</td>
<td>-124</td>
<td>670</td>
<td>2,054</td>
</tr>
<tr>
<td>20160526</td>
<td>-14,446</td>
<td>-5,340</td>
<td>-1,034</td>
<td>-116</td>
<td>673</td>
<td>2,040</td>
</tr>
<tr>
<td>20160527</td>
<td>-14,063</td>
<td>-5,014</td>
<td>-946</td>
<td>-90</td>
<td>690</td>
<td>2,074</td>
</tr>
<tr>
<td>20160531</td>
<td>-14,156</td>
<td>-5,184</td>
<td>-1,000</td>
<td>-86</td>
<td>704</td>
<td>2,130</td>
</tr>
</tbody>
</table>

Table 5: TotalView vs QBBO
Figure 2: TotalView vs. QBBO
Arca

Table 6: Arca percentiles pre- and post-6/20 performance update below lists the average difference for the delay between XDP Arca BBO and XDP Arca Book. The percent is the percentile that the XDP Arca Book is faster than XDP Arca BBO. A negative number means that XDP Arca BBO is faster than XDP Arca Book. This is the packet receipt time of the BBO minus packet receipt of the depth which is captured on the same server.

Table 3: BBO & Depth with Odd Lot Order Example below shows a significant improvement in speed in the XDP Arca Book on June 20, 2016. XDP Arca Book is faster than XDP Arca BBO 73% of the time before the 20th.

Before June 20th

- XDP Arca Book is faster than XDP Arca BBO 73% of the time.
- XDP Arca BBO is 5,000ns faster 20% of the time.
- XDP Arca BBO is 130,000ns faster 10% of the time.
- XDP Arca BBO is 2,000,000ns faster 5% of the time.
- XDP Arca BBO is 20,000,000ns faster 1% of the time
- XDP Arca BBO is 102,000,000ns faster 0.1% of the time.

After June 20th

- XDP Arca Book is faster than XDP Arca BBO 83% of the time.
- XDP Arca BBO is 17,000ns faster 10% of the time.
- XDP Arca BBO is 165,000ns faster 5% of the time.
- XDP Arca BBO is 4,178,000ns faster 1% of the time
- XDP Arca BBO is 24,349,000ns faster 0.1% of the time.

<table>
<thead>
<tr>
<th>Percent</th>
<th>Pre 6/17 (micro)</th>
<th>Post 6/20 (micro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>-5</td>
<td>1</td>
</tr>
<tr>
<td>10%</td>
<td>-103</td>
<td>-17</td>
</tr>
<tr>
<td>5%</td>
<td>-2,004</td>
<td>-165</td>
</tr>
<tr>
<td>1%</td>
<td>-20,533</td>
<td>-4,178</td>
</tr>
<tr>
<td>0.1%</td>
<td>-102,197</td>
<td>-24,349</td>
</tr>
</tbody>
</table>

Table 6: Arca percentiles pre- and post-6/20 performance update
Date | 99.9% | 99% | 95% | 90% | 80% | 50%
--- | --- | --- | --- | --- | --- | ---
20160606 | -100,528,590 | -13,399,385 | -569,478 | -22,203 | -1,975 | 4,322
20160607 | -73,468,179 | -21,536,849 | -1,972,965 | -49,450 | -3,040 | 4,977
20160608 | -42,372,323 | -6,910,623 | -72,702 | -12,078 | -954 | 5,290
20160609 | -103,079,282 | -22,983,983 | -2,045,430 | -52,348 | -3,487 | 4,165
20160610 | -80,079,600 | -16,049,288 | -935,048 | -30,570 | -3,822 | 3,792
20160611 | -91,414,392 | -23,173,452 | -2,645,165 | -104,897 | -5,260 | 4,537
20160612 | -146,789,111 | -30,128,978 | -3,590,733 | -6,951 | 4,225
20160613 | -105,224,356 | -22,042,953 | -72,702 | -12,078 | -954 | 5,290
20160614 | -121,677,063 | -21,491,446 | -2,269,216 | -7,428 | 3,666
20160616 | -17,646,417 | -2,778,396 | -31,948 | -10,848 | 2,704 | 7,994
20160617 | -16,292,251 | -2,195,664 | -32,179 | -10,632 | 2,832 | 8,382
20160618 | -17,107,479 | -2,147,705 | -29,973 | -9,559 | 3,256 | 8,546
20160619 | -20,540,029 | -3,123,783 | -67,294 | -14,677 | 1,758 | 7,838
20160621 | -34,127,138 | -8,484,274 | -744,015 | -38,236 | -19,773 | 7,361
20160622 | -35,882,920 | -6,754,272 | -233,195 | -18,813 | 1,274 | 7,994
20160623 | -25,550,190 | -3,055,394 | -40,300 | -10,960 | 2,779 | 8,128
20160624 | -30,338,622 | -4,788,228 | -116,423 | -16,234 | 1,534 | 7,727

Table 7: XDP Arca Book vs. XDP Arca BBO

Figure 3: XDP Arca Book vs. XDP Arca BBO
Figure 4: XDP Arca Book - Example Latency Spike by Session

Figure 4: XDP Arca Book - Example Latency Spike by Session is an example of a latency spike in XDP Arca Book vs. XDP Arca BBO. We’ve split the data by Arca session over the same time period. Normally we would expect the entire session to be delayed but in Figure 5: Arca - Multiple Latency Groupings within Single Session, you can see multiple different latency sets within a session, but more interesting is that the latency spike is very similar across all sessions.
This shows that there is some relationship between the TUs after the matching engine, but before the data is processed by the session in XDP Arca book. The latency delay is not related to the session, but related to the TU and different TUs are also related.
The table below lists the average difference for the delay between XDP NYSE BBO and OpenBook Ultra. The percent is the percentile that the OpenBook Ultra is faster than XDP NYSE BBO. A negative number means that XDP NYSE BBO is faster than OpenBook Ultra. This is the packet receipt time of the BBO minus packet receipt of the depth which is captured on the same server.

- **OpenBook Ultra** is faster than XDP NYSE BBO 7% of the time.
- **XDP NYSE BBO** is 48,000 faster 50% of the time.
- **XDP NYSE BBO** is 130,000ns faster 20% of the time.
- **XDP NYSE BBO** is 185,000ns faster 10% of the time.
- **XDP NYSE BBO** is 250,000ns faster 5% of the time.
- **XDP NYSE BBO** is 1,300,000ns faster 1% of the time.
- **XDP NYSE BBO** is 30,000,000ns faster 0.1% of the time.

OpenBook Ultra is broken into 26 different sessions based on letter. The volume of data on each session varies. Thus, these latencies do vary per session, but overall are all similar. XDP Nyse BBO has a single session.

<table>
<thead>
<tr>
<th>Date</th>
<th>99.9%</th>
<th>99%</th>
<th>95%</th>
<th>90%</th>
<th>80%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160603</td>
<td>-26,125,162</td>
<td>-1,502,335</td>
<td>-242,829</td>
<td>-180,684</td>
<td>-129,923</td>
<td>-46,864</td>
</tr>
<tr>
<td>20160606</td>
<td>-26,869,565</td>
<td>-1,279,339</td>
<td>-234,425</td>
<td>-179,060</td>
<td>-130,675</td>
<td>-47,378</td>
</tr>
<tr>
<td>20160607</td>
<td>-27,592,457</td>
<td>-975,251</td>
<td>-223,704</td>
<td>-174,850</td>
<td>-128,344</td>
<td>-45,635</td>
</tr>
<tr>
<td>20160612</td>
<td>-28,852,538</td>
<td>-1,599,967</td>
<td>-262,804</td>
<td>-188,142</td>
<td>-134,214</td>
<td>-48,094</td>
</tr>
<tr>
<td>20160613</td>
<td>-23,647,038</td>
<td>-1,485,742</td>
<td>-255,868</td>
<td>-187,360</td>
<td>-135,049</td>
<td>-49,769</td>
</tr>
<tr>
<td>20160614</td>
<td>-32,476,924</td>
<td>-1,918,196</td>
<td>-281,885</td>
<td>-193,559</td>
<td>-137,463</td>
<td>-50,455</td>
</tr>
<tr>
<td>20160615</td>
<td>-33,694,722</td>
<td>-1,719,527</td>
<td>-259,608</td>
<td>-186,734</td>
<td>-133,104</td>
<td>-47,900</td>
</tr>
<tr>
<td>20160616</td>
<td>-30,098,200</td>
<td>-1,261,861</td>
<td>-239,134</td>
<td>-181,333</td>
<td>-131,637</td>
<td>-47,885</td>
</tr>
<tr>
<td>20160617</td>
<td>-33,156,596</td>
<td>-1,425,616</td>
<td>-242,159</td>
<td>-181,817</td>
<td>-131,078</td>
<td>-47,497</td>
</tr>
<tr>
<td>20160618</td>
<td>-29,737,614</td>
<td>-1,597,010</td>
<td>-257,060</td>
<td>-186,979</td>
<td>-134,214</td>
<td>-48,593</td>
</tr>
<tr>
<td>20160619</td>
<td>-27,832,180</td>
<td>-1,305,460</td>
<td>-234,627</td>
<td>-177,287</td>
<td>-127,919</td>
<td>-47,140</td>
</tr>
<tr>
<td>20160620</td>
<td>-28,613,254</td>
<td>-1,899,540</td>
<td>-319,548</td>
<td>-198,670</td>
<td>-137,955</td>
<td>-51,453</td>
</tr>
</tbody>
</table>
The Latency Difference between Depth of Book and BBO Feeds

Table 8: OpenBook Ultra vs., XDP NYSE BBO

<table>
<thead>
<tr>
<th>Date</th>
<th>OpenBook Ultra</th>
<th>XDP NYSE BBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160628</td>
<td>-35,324,708</td>
<td>-1,673,393</td>
</tr>
<tr>
<td>20160629</td>
<td>-33,590,720</td>
<td>-1,337,088</td>
</tr>
<tr>
<td>20160630</td>
<td>-36,309,577</td>
<td>-1,571,603</td>
</tr>
</tbody>
</table>

Figure 6: OpenBook Ultra vs. XDP NYSE BBO
NYSE Mkt

Table 9: OpenBook Ultra Mkt vs. XDP NYSE BBO, below, lists the average difference for the delay between XDP NYSE Mkt BBO and OpenBook Ultra Mkt. The percent is the percentile that the OpenBook Ultra Mkt is faster than XDP NYSE Mkt BBO. A negative number means that XDP NYSE Mkt BBO is faster than OpenBook Ultra Mkt. This is the packet receipt time of the BBO minus packet receipt of the depth that is captured on the same server.

- **OpenBook Ultra Mkt is faster than XDP NYSE BBO <1% of the time.**
- **XDP NYSE BBO is 31,000ns faster 50% of the time.**
- **XDP NYSE BBO is 78,000ns faster 20% of the time.**
- **XDP NYSE BBO is 113,000ns faster 10% of the time.**
- **XDP NYSE BBO is 154,000ns faster 5% of the time.**
- **XDP NYSE BBO is 1,480,000ns faster 1% of the time**
- **XDP NYSE BBO is 26,100,000ns faster 0.1% of the time.**

OpenBook Ultra Mkt has a single session. XDP NYSE Mkt also has a single session.

<table>
<thead>
<tr>
<th>Date</th>
<th>99.9%</th>
<th>99%</th>
<th>95%</th>
<th>90%</th>
<th>80%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160711</td>
<td>-25,174,812</td>
<td>-874,228</td>
<td>-142,179</td>
<td>-109,166</td>
<td>-68,918</td>
<td>-30,264</td>
</tr>
<tr>
<td>20160712</td>
<td>-26,442,402</td>
<td>-1,002,156</td>
<td>-155,166</td>
<td>-113,793</td>
<td>-81,017</td>
<td>-30,324</td>
</tr>
<tr>
<td>20160713</td>
<td>-25,553,487</td>
<td>-1,042,493</td>
<td>-151,880</td>
<td>-112,421</td>
<td>-73,977</td>
<td>-32,045</td>
</tr>
<tr>
<td>20160714</td>
<td>-25,276,437</td>
<td>-765,398</td>
<td>-161,618</td>
<td>-117,979</td>
<td>-88,103</td>
<td>-33,476</td>
</tr>
<tr>
<td>20160715</td>
<td>-25,916,033</td>
<td>-742,368</td>
<td>-155,620</td>
<td>-113,867</td>
<td>-84,318</td>
<td>-30,771</td>
</tr>
<tr>
<td>20160718</td>
<td>-26,320,688</td>
<td>-1,027,569</td>
<td>-152,663</td>
<td>-112,377</td>
<td>-74,968</td>
<td>-29,780</td>
</tr>
<tr>
<td>20160719</td>
<td>-25,038,384</td>
<td>-923,388</td>
<td>-158,333</td>
<td>-115,157</td>
<td>-81,413</td>
<td>-36,009</td>
</tr>
<tr>
<td>20160721</td>
<td>-25,120,638</td>
<td>-1,093,581</td>
<td>-154,249</td>
<td>-113,182</td>
<td>-75,996</td>
<td>-33,103</td>
</tr>
<tr>
<td>20160722</td>
<td>-28,073,855</td>
<td>-2,994,270</td>
<td>-155,159</td>
<td>-111,521</td>
<td>-74,678</td>
<td>-30,905</td>
</tr>
<tr>
<td>20160723</td>
<td>-25,672,317</td>
<td>-1,245,037</td>
<td>-159,272</td>
<td>-115,901</td>
<td>-83,603</td>
<td>-33,588</td>
</tr>
<tr>
<td>20160724</td>
<td>-26,125,602</td>
<td>-2,250,851</td>
<td>-159,592</td>
<td>-114,731</td>
<td>-84,735</td>
<td>-32,357</td>
</tr>
<tr>
<td>20160725</td>
<td>-28,349,854</td>
<td>-3,894,031</td>
<td>-155,837</td>
<td>-112,399</td>
<td>-76,823</td>
<td>-29,481</td>
</tr>
<tr>
<td>20160726</td>
<td>-26,673,563</td>
<td>-2,131,119</td>
<td>-151,142</td>
<td>-110,381</td>
<td>-75,772</td>
<td>-28,468</td>
</tr>
</tbody>
</table>

Table 9: OpenBook Ultra Mkt vs. XDP NYSE BBO
Figure 7: OpenBook Ultra Mkt vs. XDP NYSE BBO
In Depth Observations

NYSE: Significantly different packaging of messages in the BBO and Depth Feeds

We show below that there is dramatic differences in how the feeds package their output data together. We see quote messages in the same packet in the BBO feeds, but split in the depth feed, and vice versa. Below we show a single example of two quote messages coming in different packets in the BBO feed, but arriving in the same packet later in the depth feed.

Packet 1 for XDP BBO. Packet with receipt microsecond of 092287

```
packet_begin timestamp(2016-06-03T15:02:24.092287+00:00:00)
dst_endpoint(233.125.89.0:11100) length(206) sequence_number '21141301'
packet_header( pkt_size_ '206' delivery_flag_ '11' number_msgs_ '5' seq_num_ '21141301' send_time_ '1464966144' send_time_ns_ '92048860'
)
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '91899000'
symbol_index_ '3822' symbol_seq_num_ '20975' ask_price_ '505700' ask_volume_ '100'
bid_price_ '505600' bid_volume_ '100' quote_condition_ 'R' (0x52) rpi_indicator_ ' ', (0x20) filler_ '117300' )
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '91947000'
symbol_index_ '10249' symbol_seq_num_ '49367' ask_price_ '547000' ask_volume_ '155100'
bid_price_ '545000' bid_volume_ '122700' quote_condition_ 'R' (0x52) rpi_indicator_ 'B' (0x42) filler_ '181326' )
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '91955000'
symbol_index_ '1749' symbol_seq_num_ '49367' ask_price_ '832700' ask_volume_ '200'
bid_price_ '831800' bid_volume_ '200' quote_condition_ 'R' (0x52) rpi_indicator_ 'A' (0x41) filler_ '69763' )
BBO,APTS,100,13.95,13500,14.0,2016-06-03T15:02:24.091953+00:00:00,1464966144091953000,1464966144092287000
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '91995000'
symbol_index_ '8917' symbol_seq_num_ '2135' ask_price_ '139500' ask_volume_ '100'
bid_price_ '139900' bid_volume_ '100' quote_condition_ 'R' (0x52) rpi_indicator_ ' ', (0x20) filler_ '12170' )
BBO,APTS,100,13.95,100,13.99,2016-06-03T15:02:24.091984+00:00:00,14649661440919984000,1464966144092287000
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '91995000'
symbol_index_ '8917' symbol_seq_num_ '2136' ask_price_ '139900' ask_volume_ '100'
bid_price_ '139500' bid_volume_ '100' quote_condition_ 'R' (0x52) rpi_indicator_ ' ', (0x20) filler_ '12171' )
packet_end timestamp(2016-06-03T15:02:24.092287+00:00:00)
dst_endpoint(233.125.89.0:11100) length(206) sequence_number '21141301'
last_sequence_number '21141305'
```
In OpenBook Ultra, we see two Delta Update Messages sent in the same packet that corresponds to two quote messages that were originally received in different packets. Notice below in the first packet the source time of 39744091 and source_time_micro_secs of 984, for a microsecond number of 091984, which you can match above in the BBO.

```
packet_begin timestamp(2016-06-03T15:02:24.092524+00:00:00)
dst_endpoint(233.75.215.96:60096) length(108) sequence_number '6283678'
header ( msg_size_ '106' msg_type_ '231' msg_seq_num_ '6283678' send_time_ '39744092'
product_id_ '115' retrans_flag_ '1' num_body_entries_ '2' link_flag_ '0' )
delta_update delta_update ( msg_size_ '46' security_index_ '8917' source_time_ '39744091' source_time_micro_secs_ '984' source_seq_num_ '5858' source_session_id_ '1' quote_condition_ '32' trading_status_ '79' price_scale_code_ '4' )
delta_update_price_points[1] ([price_numerator_ '139900' volume_ '100' chg_qty_ '100'
num_orders_ '1' side_ 'S' (0x53) reason_code_ '79' link_id_1_ '0' link_id_2_ '0'
link_id_3_ '0'] ) delta_update_price_points_count '1'
2016-06-03T15:02:24.092524+00:00:00 6283678 [APTS] bid: 100@13.9500000 ask: 100@13.9900000 New

delta_update delta_update ( msg_size_ '46' security_index_ '8917' source_time_ '39744092' source_time_micro_secs_ '53' source_seq_num_ '5859' source_session_id_ '1' quote_condition_ '32' trading_status_ '79' price_scale_code_ '4' )
delta_update_price_points[1] ([price_numerator_ '139500' volume_ '200' chg_qty_ '100'
num_orders_ '2' side_ 'B' (0x42) reason_code_ '79' link_id_1_ '0' link_id_2_ '0'
link_id_3_ '0'] ) delta_update_price_points_count '1'
2016-06-03T15:02:24.092524+00:00:00 6283678 [APTS] bid: 200@13.9500000 ask: 100@13.9900000 New
```

Packet 2 for XDP BBO. Packet with receipt microsecond 092425.

```
packet_begin timestamp(2016-06-03T15:02:24.092425+00:00:00)
dst_endpoint(233.125.89.0:11100) length(54) sequence_number '21141306'
packet_header ( pkt_size_ '54' delivery_flag_ '11' number_msgs_ '1' seq_num_ '21141306'
send_time_ '1464966144' send_time_ns_ '92188533' )
BBO,APTS,200,13.95,100,13.99,2016-06-03T15:02:24.092053+00:00:00,1464966144092053000,1464966144092425000
quote_message ( msg_size_ '38' msg_type_ '140' source_time_ns_ '92053000'
symbol_index_ '8917' symbol_seq_num_ '2137' ask_price_ '139900' ask_volume_ '100'
bid_price_ '139500' bid_volume_ '200' quote_condition_ 'R' (0x52) rpi_indicator_ ' ' (0x20) filler_ '12172' )
packet_end timestamp(2016-06-03T15:02:24.092425+00:00:00)
dst_endpoint(233.125.89.0:11100) length(54) sequence_number '21141306'
last_sequence_number '21141306'
```
The feed diff summary file:

| B0, D2, CSC, 100, 50.56, 100, 50.57, 2016-06-03T15:02:24.091899+00:00:00, 1464966144092287778, 1464966144092308647, -20869 |
| B0, D0, APTS, 100, 13.95, 13500, 14.0, 2016-06-03T15:02:24.091953+00:00:00, 1464966144091953000, 1464966144092287778, 1464966144092359162, -71384 |
| B0, D0, ABEV, 122700, 5.45, 155100, 5.47, 2016-06-03T15:02:24.091947+00:00:00, 1464966144091947000, 1464966144092287778, 1464966144092442616, -154838 |
| B0, D0, APTS, 100, 13.95, 100, 13.99, 2016-06-03T15:02:24.091984+00:00:00, 1464966144091984000, 1464966144092287778, 1464966144092524737, -236959 |
| B0, D0, APTS, 200, 13.95, 100, 13.99, 2016-06-03T15:02:24.092053+00:00:00, 1464966144092053000, 1464966144092425763, 1464966144092524737, -98974 |

In the bottom two matches, we see that the OpenBook Ultra has a single packet at time 1464966144092524737 (microsecond 092524), but is sent in BBO updates that are in two different packets 1464966144092287778, and 1464966144092425763 (microsecond 092287 and 092425 respectively). In addition to this, we see that the BBO timestamp of 1464966144092287778 actually had 4 updates for this product in the same packet that is in different packets in the Depth feed.

**NYSE: Open Book Ultra vs XDP NYSE BBO: Session numbers**

The Open Book Ultra has 20 sessions, based on the symbol letters. We observe that the latency between the Depth and BBO feeds, varies across sessions. For example in the below table (**Table 10: **OpenBook Ultra: Sample latency per session), we see at 12:25:28 EST, the OpenBook Ultra feed on Session 2 lags the XDP NYSE BBO feed, but on other sessions – D0, D18, D5 and D19, the updates on the BBO feed comes much later than the corresponding Depth Feed Updates.

This shows how the latency is most likely related to the production of the depth data on the OpenBook Ultra session.

<table>
<thead>
<tr>
<th>BBO Sess #</th>
<th>Depth Sess #</th>
<th>Product</th>
<th>Exchange Time</th>
<th>Exchange Time</th>
<th>BBO Receipt Time</th>
<th>Depth Feed Receipt Time</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>D19</td>
<td>XHR</td>
<td>2016-06-15 16:25:29.127687</td>
<td>1466007929 127687000</td>
<td>1466007929 131182186</td>
<td>1466007929 128562346</td>
<td>2,619,840</td>
</tr>
</tbody>
</table>
The Latency Difference between Depth of Book and BBO Feeds

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>D0</td>
<td>AXS</td>
<td>2016-06-15 16:25:29.126743</td>
<td>1466007929 126743000</td>
<td>1466007929 127241574</td>
<td>1466007929 131226018</td>
<td>-3,984,444</td>
</tr>
<tr>
<td>B0</td>
<td>D0</td>
<td>AXS</td>
<td>2016-06-15 16:25:29.126764</td>
<td>1466007929 126764000</td>
<td>1466007929 127241574</td>
<td>1466007929 131226018</td>
<td>-3,984,444</td>
</tr>
<tr>
<td>B0</td>
<td>D5</td>
<td>FTAI</td>
<td>2016-06-15 16:25:29.129030</td>
<td>1466007929 129030000</td>
<td>1466007929 131265223</td>
<td>1466007929 129446558</td>
<td>1,818,665</td>
</tr>
<tr>
<td>B0</td>
<td>D5</td>
<td>FTAI</td>
<td>2016-06-15 16:25:29.129289</td>
<td>1466007929 129289000</td>
<td>1466007929 131265223</td>
<td>1466007929 129778876</td>
<td>1,486,347</td>
</tr>
<tr>
<td>B0</td>
<td>D5</td>
<td>FTAI</td>
<td>2016-06-15 16:25:29.129305</td>
<td>1466007929 129305000</td>
<td>1466007929 131265223</td>
<td>1466007929 129852816</td>
<td>1,412,407</td>
</tr>
<tr>
<td>B0</td>
<td>D5</td>
<td>FSIC</td>
<td>2016-06-15 16:25:29.129475</td>
<td>1466007929 129475000</td>
<td>1466007929 131265223</td>
<td>1466007929 130070954</td>
<td>1,194,269</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CNK</td>
<td>2016-06-15 16:25:29.117284</td>
<td>1466007929 117284000</td>
<td>1466007929 119833171</td>
<td>1466007929 131271921</td>
<td>-11,438,750</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CHS</td>
<td>2016-06-15 16:25:29.117295</td>
<td>1466007929 117295000</td>
<td>1466007929 119833171</td>
<td>1466007929 131271921</td>
<td>-11,438,750</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CHS</td>
<td>2016-06-15 16:25:29.117316</td>
<td>1466007929 117316000</td>
<td>1466007929 119833171</td>
<td>1466007929 131328970</td>
<td>-11,495,799</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CHS</td>
<td>2016-06-15 16:25:29.117332</td>
<td>1466007929 117332000</td>
<td>1466007929 119833171</td>
<td>1466007929 131328970</td>
<td>-11,495,799</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CHS</td>
<td>2016-06-15 16:25:29.117368</td>
<td>1466007929 117368000</td>
<td>1466007929 119833171</td>
<td>1466007929 131387427</td>
<td>-11,554,256</td>
</tr>
<tr>
<td>B1</td>
<td>D2</td>
<td>CNK</td>
<td>2016-06-15 16:25:29.117386</td>
<td>1466007929 117386000</td>
<td>1466007929 119833171</td>
<td>1466007929 131387427</td>
<td>-11,554,256</td>
</tr>
<tr>
<td>B1</td>
<td>D0</td>
<td>AEO</td>
<td>2016-06-15 16:25:29.120555</td>
<td>1466007929 120555000</td>
<td>1466007929 131424516</td>
<td>1466007929 123329207</td>
<td>8,095,309</td>
</tr>
<tr>
<td>B0</td>
<td>D2</td>
<td>CDE</td>
<td>2016-06-15 16:25:29.117487</td>
<td>1466007929 117487000</td>
<td>1466007929 124832592</td>
<td>1466007929 131504349</td>
<td>-6,671,757</td>
</tr>
</tbody>
</table>

Table 10: OpenBook Ultra: Sample latency per session
Solution
A key question: How do you always get the benefit of the faster information?

We believe that is generally easy: in each tick that comes across, there is a matching-engine timestamp that we can use to compare the data, and see which feed is faster.

For example:

- In the depth, it is the time of the add, modify, cancel or trade message.
- In the BBO, it is the time that caused the new BBO to occur, which is the underlying add, modify cancel or trade message.
- This data is unique and in order per product.
- All that needs to be done is keep a ‘master’ BBO, which gets updated based on the top of the depth feed and the direct BBO feed.
- If the timestamp associated with the BBO update is newer than the newest depth update, then the BBO is faster.
- If the timestamp associated with the depth update is newer than the newest BBO update, then the depth is faster.

Example using Bellport™

```cpp
class ApuBboArbitrator : public bp::feed::AggregatedPriceFeedCallbackAdapter {
public:
    ApuBboArbitrator()
    : master_exchange_time_(bp::sys::UTCTime::Min())
    {}

virtual void OnAggregatedPriceUpdate(const bp::feed::AggregatedPriceUpdate& apu, void* /*closure*/) {
    if (apu.last_exchange_timestamp() > master_exchange_time_) {
        std::cout << "Depth: " << apu.top_of_book().ToString() << std::endl;
        master_exchange_time_ = apu.last_exchange_timestamp();
    }
}

virtual void OnBBOQuote(const bp::feed::BBOQuote& bbo_quote, void* /*closure*/) {
    if (bbo_quote.exchange_timestamp() > master_exchange_time_) {
        std::cout << "BBO: " << bbo_quote.ToString() << std::endl;
        master_exchange_time_ = bbo_quote.exchange_timestamp();
    }
}
private:
    bp::sys::UTCTime master_exchange_time_;}
```
Totalview Raw Feed example

If you are looking to match on the raw data, below are examples of the Quotation message from QBBO and the AddOrderNoMPIDAtribution from Total View.

The timestamp field in each message is identical in each feed’s binary data.

```c
// [size 34]
struct Quotation {
    char message_type;
    // NASDAQ OMX internal tracking number
    uint16_t tracking_number;
    // Nanoseconds since midnight
    char timestamp[6];
    // Stock symbol, right padded with spaces
    char stock[8];
    // Indicates the primary listing market for the issue.
    char security_class;
    uint32_t nasdaq_best_bid_price;
    uint32_t nasdaq_best_bid_size;
    uint32_t nasdaq_best_offer_price;
    uint32_t nasdaq_best_offer_size;
};

// [size 36]
struct AddOrderNoMPIDAtribution {
    char message_type;
    // Locate code identifying the security
    uint16_t stock_locate;
    // NASDAQ OMX internal tracking number
    uint16_t tracking_number;
    // Nanoseconds since midnight.
    char timestamp[6];
    // The unique reference number assigned to the new order at the time of
    // receipt.
    uint64_t order_reference_number;
    // The type of order being added.
    char buy_sell_indicator;

    // The total number of shares associated with the order being added to the
    // book.
    uint32_t shares;
    // The security symbol for which the order is being added.
    char stock_[8];
    // The display price of the new order. Refer to Data Types for field
    // processing notes.
    uint32_t price;
};
```


```c
};

// Convert
uint64_t nanos_past_midnight(0);
// Endian swap
char buff[6];
buff[0] = msg.timestamp[5];
buff[1] = msg.timestamp[4];
buff[2] = msg.timestamp[3];
buff[3] = msg.timestamp[2];
buff[4] = msg.timestamp[1];
buff[5] = msg.timestamp[0];
memcpy(&nanos_past_midnight, buff, 6);
```
Discussion

Many factors influence a purchasing decision: price, access, latency, depth and complexity to name a few. The ability to have knowledge about the state of the exchange is critical to market participants and the market data feeds are the only timely insight. If a firm wants to consistently receive up-to-date information of the market, then assuming that all feeds are equal can be a costly oversight.

The times and latencies that we discussed here is the ingress on the NIC card itself. It does not include the processing of the data into a useful, actionable form i.e., after book building on the depth feed. By the nature of the data, the depth feeds require more memory and cycles to create actionable data, which can result in delays and cache misses within the processing machine. What this means is there is probably at least 100s of nanoseconds to many microsecond difference in the internal processing time of the depth feed when compared to the BBO feeds in order to bring it to an actionable state. In order to reduce the variables and increase the applicability of this study, we decided to compare the ingress time of the packets, instead of pushing that data through the Bellport™ feed handlers. That said, for client decision making, users should at least have a constant estimated offset, which is applicable to their system.

Key Points

1. Nasdaq TotalView provides the price data faster than QBBO 90% of the time
2. XDP ArcaBook provides the price data faster than XDP Arca BBO 83% of the time
3. NYSE OpenBook Ultra provides price data faster than XDP NYSE BBO 7% of the time
4. NYSE Mkt OpenBook Ultra provides price data faster than XDP NYSE Mkt BBO <1% of the time

NASDAQ

The QBBO feed generally provides a one to one mapping of TotalView for top of book updates. There are odd lots orders present on QBBO. The percentage difference between the two feeds is not a lot compared to the other feeds at 0.1% at 12us, but it is present and significant. Users would be able to eliminate this by ingesting both feeds, but the additional cost and complexity should be considered. We have only compared the software TotalView feed, so the FPGA TotalView feed will likely have different characteristics.

Arca

Arca Book is generally faster than the XDP BBO. The BBO does not have odd lot orders, so the books may be different (see Table 3: BBO & Depth with Odd Lot Order Example). The BBO can be an indicator of better price information. If the BBO has a larger exchange timestamp, and
the price is better, then it may not be the full quantity that will subsequently be sent on the depth feed, but the price is real and can be acted upon. If the BBO has a larger exchange timestamp, and the price is worse there may be a small odd lot order with a better price. With possible millisecond delays though, this price information can still significantly improve decision making. The fact that users can know that their depth information is slow may still be an important signal.

**NYSE and NYSE Mkt**

The OpenBook Ultra feed is where we see the largest reason to arbitrate the feeds. OpenBook Ultra for NYSE is broken into 26 different sessions; for NYSE Mkt there is just one. For the XDP BBO there is one session. The BBO does not publish odd lot orders. OpenBook Ultra is almost always slower than the BBO. If users purchase OpenBook Ultra with the intention to get the faster feed, then they should switch to the BBO feed. Since it is almost always faster, arbitrating will constantly deliver timelier top of book price information. It does complicate processing logic when it doesn’t match, but at a minimum it can allow users to know when prices in the depth feed don’t exist anymore, or has been bettered.

**Future Work**

There are many questions still unanswered about what causes some of these latency spikes, as well as unanswered relationships within this data. The most obvious is going one level down and fully analyzing each session, TU, letter and symbol. There are also additional feeds from the exchanges, like the FPGA TotalView feed, or the integrated feeds, that we will analyze to see how they compare to each other. In addition, as time marches on the feeds and exchanges are constantly changing and so subsequent studies using the same methodology will provide interesting and insightful analysis.

We hope that this in-depth study and detailed analysis can help you and your firm in making more informed purchasing and development decisions.
About the Author

Co-Founder and COO Michael Lehr manages the operations and processes at MayStreet. His vision of structural organization and his technical mastery allow him to keep a firm footing on the necessary day-to-day happenings at MayStreet.

Michael has vast technical experience with a wide range of business and technology groups. Most recently, he was with Organic Motion where he held the positions of software architect, principal engineer and vice president, engineering. He oversaw the technical team, as well as coding in computer graphics and computer vision, to successfully usher products to the market.

Prior to Organic Motion, Michael was a computer vision AMTS at Sarnoff Research Labs in Princeton, NJ, where he was responsible for optimizing object tracking and image enhancement algorithms for aerial and ground video.

He joined Sarnoff after many years with Lockheed Martin IS&S where he worked on large distributed systems.

Originally from Tarrytown, NY, he received his BA in Computer Science and Brain and Cognitive Science from the University of Rochester and received an MS in Engineering (Computer and Information Science) from the University of Pennsylvania.

All data in this study is freely available upon request

For more information concerning this study, MayStreet and its expanding suite of market access and data analytic software for capital markets firms, including the best-in-breed Bellport™ flagship feed handler, please write to David Streltsoff, director of business development, at ds@maystreet.com.