July 10, 2017

Brent J. Fields, Secretary
Securities and Exchange Commission
100 F Street, NE
Washington DC

Re: Release No. 34-80041, File No. SR-CHX-2017-04, Chicago Stock Exchange, Inc., Notice of Filing of Proposed Rule Change to Adopt the CHX Liquidity Enhancing Access Delay ("Filing"); Release No. 34-80740, File No. SR-CHX-2017-04, Chicago Stock Exchange, Inc., Order Instituting Proceedings to Determine Whether to Approve or Disapprove a Proposed Rule Change to Adopt the CHX Liquidity Enhancing Access Delay ("Order")

Dear Mr. Fields:

Let's take another look at the Chicago Stock Exchange's ("CHX") response to my criticisms of its speed bump implementation details.[1] As I've already said,[2] in the examples in its response,[3] CHX apparently assumed any overheads solely attributable to its speed bump are zero and will always be zero, even under load.[4] I believe that assumption is incorrect, and that this incorrect assumption leads to an incorrect equivalence between its proposed speed bump and other exchanges. Instead, I believe its speed bump software overheads might degrade nonlinearly under load, further disadvantaging non-market maker participants.

But in this letter I'll stipulate to CHX's assumptions for its examples, take a closer look, and see where we go. CHX's assumptions are:[5]

- There are three exchanges each of which receives identical order flow at identical times:
    - CHX with LEAD's 350 microsecond intentional [*sic*] in effect.
    - IEX with its 350 microsecond intentional hardware delay in effect.
    - ARCA with no intentional delay in effect.
- All three exchanges have matching engines that take 100 microseconds to process the average message.

As in CHX's example (following my example[6]) we'll assume a stock has market moving news break at 10:00:00.000.000. I'll also assume that stock has, say, three LEAD market makers assigned to it, each of which has orders in the book at just two price levels on each side, for a total of four resting orders per market maker.

---

[1]Letter to Eduardo A. Aleman, Assistant Secretary, SEC, from James G. Ongena, Executive Vice President and General Counsel, Chicago Stock Exchange, June 30, 2017 ("CHX Response")
[2]Letter to Brent J. Fields, Secretary, SEC, from R. T. Leuchtkafer, July 7, 2017.
[3]CHX Response, pages 19-25.
[4]CHX Response, page 18. Under the heading "Assumptions" CHX does not show any LEAD overhead time, presumably because for the purposes of its following examples it assumes that overhead time is and will always be zero.
[5]*Id.*
[6]Letter to Brent J. Fields, Secretary, SEC, from R. T. Leuchtkafer, June 15, 2017, pages 2-5.

After the news breaks at 10:00:00.000.000, the three LEAD market makers respond by cancelling their resting orders. Other participants respond by cancelling their orders too. Let's see what might happen.

Seq#	Time received	LEAD MM	Delayable message	
1	10:00:00.000.011	N	Y	
2	10:00:00.000.100	Y	N	LEAD MM #1 cancels resting orders on bid and offer sides.
3	10:00:00.000.102	Y	N	
4	10:00:00.000.114	Y	N	
5	10:00:00.000.116	Y	N	
6	10:00:00.000.130	N	Y	
7	10:00:00.000.140	Y	N	LEAD MM #2 cancels resting orders on bid and offer sides.
8	10:00:00.000.142	Y	N	
9	10:00:00.000.144	Y	N	
10	10:00:00.000.146	Y	N	
11	10:00:00.000.150	N	Y	
12	10:00:00.000.160	Y	N	LEAD MM #3 cancels orders on bid and offer sides.
13	10:00:00.000.162	Y	N	
14	10:00:00.000.164	Y	N	
15	10:00:00.000.166	Y	N	

At this point, the LEAD queue of delayable messages has three messages in it and looks like this:

Seq#	Time received	LEAD MM	Delayable message
1	10:00:00.000.011	N	Y
6	10:00:00.000.130	N	Y
11	10:00:00.000.150	N	Y

On the other hand, all messages from LEAD MMs have bypassed this queue and are on their way to the matching engine. Under CHX's assumptions, each message takes 100 microseconds to process and, as I understand CHX's various filings, because they all arrived within the 350 microsecond "Fixed LEAD Period" from the first delayable message received at 10:00:00.000.011, all messages from LEAD MMs in this example will be processed *before* any message on the LEAD queue is processed.

In other words, messages on their way to the matching engine, and their respective processing times, are:

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds)
2	10:00:00.000.100	Y	N	10:00:00.000.200
3	10:00:00.000.102	Y	N	10:00:00.000.300
4	10:00:00.000.114	Y	N	10:00:00.000.400
5	10:00:00.000.116	Y	N	10:00:00.000.500
7	10:00:00.000.140	Y	N	10:00:00.000.600
8	10:00:00.000.142	Y	N	10:00:00.000.700
9	10:00:00.000.144	Y	N	10:00:00.000.800
10	10:00:00.000.146	Y	N	10:00:00.000.900
12	10:00:00.000.160	Y	N	10:00:00.001.000
13	10:00:00.000.162	Y	N	10:00:00.001.100
14	10:00:00.000.164	Y	N	10:00:00.001.200
15	10:00:00.000.166	Y	N	10:00:00.001.300

It's only at this point that messages in the LEAD queue will be processed:

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds)
1	10:00:00.000.011	N	Y	10:00:00.001.400
6	10:00:00.000.130	N	Y	10:00:00.001.500
11	10:00:00.000.150	N	Y	10:00:00.001.600

We'll focus on delayable messages in the LEAD queue. Message number 1, received at 10:00:000.011, won't be processed until 1,389 microseconds later, despite being received first; message number 6, received at 10:00:000.130, won't be processed until 1,370 microseconds later; message number 11, received at 10:00:000.150, won't be processed until 1,450 microseconds later.

Let's see how Arca would perform:

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds)
1	10:00:00.000.011	n/a	n/a	10:00:00.000.111
2	10:00:00.000.100	n/a	n/a	10:00:00.000.211
3	10:00:00.000.102	n/a	n/a	10:00:00.000.311
4	10:00:00.000.114	n/a	n/a	10:00:00.000.411
5	10:00:00.000.116	n/a	n/a	10:00:00.000.511
6	10:00:00.000.130	n/a	n/a	10:00:00.000.611
7	10:00:00.000.140	n/a	n/a	10:00:00.000.711
8	10:00:00.000.142	n/a	n/a	10:00:00.000.811
9	10:00:00.000.144	n/a	n/a	10:00:00.000.911
10	10:00:00.000.146	n/a	n/a	10:00:00.001.011
11	10:00:00.000.150	n/a	n/a	10:00:00.001.111
12	10:00:00.000.160	n/a	n/a	10:00:00.001.211
13	10:00:00.000.162	n/a	n/a	10:00:00.001.311
14	10:00:00.000.164	n/a	n/a	10:00:00.001.411
15	10:00:00.000.166	n/a	n/a	10:00:00.001.511

Again, we'll focus on messages that in CHX's model would be subject to the LEAD delay. (These are messages 1, 6, and 11.) At Arca, message number 1, received at 10:00:000.011, is processed 100 microseconds later; message number 6 is processed 481 microseconds later; message number 11 is processed 961 microseconds later.

How about IEX?

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds to process after traversing IEX's 350 microsecond coil)
1	10:00:00.000.011	n/a	n/a	10:00:00.000.461
2	10:00:00.000.100	n/a	n/a	10:00:00.000.561
3	10:00:00.000.102	n/a	n/a	10:00:00.000.661
4	10:00:00.000.114	n/a	n/a	10:00:00.000.761
5	10:00:00.000.116	n/a	n/a	10:00:00.000.861
6	10:00:00.000.130	n/a	n/a	10:00:00.000.961
7	10:00:00.000.140	n/a	n/a	10:00:00.001.061
8	10:00:00.000.142	n/a	n/a	10:00:00.001.161
9	10:00:00.000.144	n/a	n/a	10:00:00.001.261
10	10:00:00.000.146	n/a	n/a	10:00:00.001.361
11	10:00:00.000.150	n/a	n/a	10:00:00.001.461
12	10:00:00.000.160	n/a	n/a	10:00:00.001.561
13	10:00:00.000.162	n/a	n/a	10:00:00.001.661
14	10:00:00.000.164	n/a	n/a	10:00:00.001.761
15	10:00:00.000.166	n/a	n/a	10:00:00.001.861

Focusing on messages 1, 6, and 11, they are processed 450, 831, and 1,311 microseconds later, respectively.

The following table is a summary:

Seq#	Time received	Total time to finish (microseconds), including speed bumps and processing		
		Arca	IEX	CHX
1	10:00:00.000.011	100	450	1,389
6	10:00:00.000.130	481	831	1,370
11	10:00:00.000.150	961	1,311	1,450

Even in this simple construct, with three LEAD market makers posting two orders per side, CHX performs considerably worse than Arca and IEX.

And the more popular CHX's speed bump becomes, the more its non-LEAD market makers suffer. Let's expand the example to include five LEAD market makers:

Seq#	Time received	LEAD MM	Delayable message	
1	10:00:00.000.011	N	Y	
2	10:00:00.000.100	Y	N	LEAD MM #1 cancels resting orders on bid and offer sides.
3	10:00:00.000.102	Y	N	
4	10:00:00.000.114	Y	N	
5	10:00:00.000.116	Y	N	
6	10:00:00.000.130	N	Y	
7	10:00:00.000.140	Y	N	LEAD MM #2 cancels resting orders on bid and offer sides.
8	10:00:00.000.142	Y	N	
9	10:00:00.000.144	Y	N	
10	10:00:00.000.146	Y	N	
11	10:00:00.000.150	N	Y	
12	10:00:00.000.160	Y	N	LEAD MM #3 cancels orders on bid and offer sides.
13	10:00:00.000.162	Y	N	
14	10:00:00.000.164	Y	N	
15	10:00:00.000.166	Y	N	
16	10:00:00.000.180	Y	N	LEAD MM #4 cancels orders on bid and offer sides.
17	10:00:00.000.182	Y	N	
18	10:00:00.000.184	Y	N	
19	10:00:00.000.186	Y	N	
20	10:00:00.000.190	Y	N	LEAD MM #5 cancels orders on bid and offer sides.
21	10:00:00.000.192	Y	N	
22	10:00:00.000.194	Y	N	
23	10:00:00.000.196	Y	N	

In this example with five LEAD market makers, the LEAD queue of delayable messages still has three messages in it and looks like this:

Seq#	Time received	LEAD MM	Delayable message
1	10:00:00.000.011	N	Y
6	10:00:00.000.130	N	Y
11	10:00:00.000.150	N	Y

Non-delayed messages sent to the matching engine, and their respective processing times, are:

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds)
2	10:00:00.000.100	Y	N	10:00:00.000.200
3	10:00:00.000.102	Y	N	10:00:00.000.300
4	10:00:00.000.114	Y	N	10:00:00.000.400
5	10:00:00.000.116	Y	N	10:00:00.000.500
7	10:00:00.000.140	Y	N	10:00:00.000.600
8	10:00:00.000.142	Y	N	10:00:00.000.700
9	10:00:00.000.144	Y	N	10:00:00.000.800
10	10:00:00.000.146	Y	N	10:00:00.000.900
12	10:00:00.000.160	Y	N	10:00:00.001.000
13	10:00:00.000.162	Y	N	10:00:00.001.100
14	10:00:00.000.164	Y	N	10:00:00.001.200
15	10:00:00.000.166	Y	N	10:00:00.001.300
16	10:00:00.000.180	Y	N	10:00:00.001.400
17	10:00:00.000.182	Y	N	10:00:00.001.500
18	10:00:00.000.184	Y	N	10:00:00.001.600
19	10:00:00.000.186	Y	N	10:00:00.001.700
20	10:00:00.000.190	Y	N	10:00:00.001.800
21	10:00:00.000.192	Y	N	10:00:00.001.900
22	10:00:00.000.194	Y	N	10:00:00.002.000
23	10:00:00.000.196	Y	N	10:00:00.002.100

At this point messages in the LEAD queue will be processed:

Seq#	Time received	LEAD MM	Delayable message	Time matching engine will finish processing (each message takes 100 microseconds)
1	10:00:00.000.011	N	Y	10:00:00.002.200
6	10:00:00.000.130	N	Y	10:00:00.002.300
11	10:00:00.000.150	N	Y	10:00:00.002.400

Messages 1, 6, and 11 are processed 2,189, 2,170, and 2,250 microseconds after receipt, respectively. Importantly, all we've done is increase the number of LEAD MMs from three to five. In this example, the more LEAD MMs CHX allows in a stock, the more other participants can suffer. Similarly, if we suppose each LEAD MM was quoting four price levels on both sides instead of just two price levels on both sides, other participants would see even greater delays. And so as I understand it, using CHX's assumptions for its examples, the more LEAD MMs there are and the more orders they manage[7] the more other participants can suffer.

At Arca, on the other hand, messages 1, 6, and 11 are still processed 100, 481, and 961 microseconds later, respectively. At IEX, they're still processed 450, 831, and 1,311 microseconds later, respectively.

Seq#	Time received	Total time to finish (microseconds), including speed bumps and processing		
		Arca	IEX	CHX
1	10:00:00.000.011	100	450	2,189
6	10:00:00.000.130	481	831	2,170
11	10:00:00.000.150	961	1,311	2,250

I don't see any equivalence at all among the exchanges here. Using CHX's assumptions for these examples, I see a material impact against non-LEAD MM participants that can quickly get worse the more LEAD MMs CHX allows in a stock and the more orders they post.

Sincerely,

R. T. Leuchtkafer

[7]Within the Fixed LEAD Period.