

FIXEdge 6.5.0 response time test results

- [Overview](#)
- [Test Data](#)
 - [FIXEdge Machine](#)
 - [Client Machine](#)
 - [Single Session Echo Scenario](#)
 - [Two Sessions With Conversion Scenario](#)
- [Tests' Configuration](#)
- [Test Results](#)
 - [Single Session Echo Test](#)
 - [Two Sessions With Conversion Test](#)

Overview

All the presented tests are performed using some "reference" hardware and software. It's advised to perform the mentioned benchmarks using target infrastructure and product configuration to get the results that are realistic for the specific use case.

Test Data

FIXEdge Machine

- Intel(R) Xeon(R) CPU E5-2643 v3 @ 3.40GHz (2 CPU Hyper-Trading Enabled, 24 Cores)
- RAM 128 GB, 2133 MHz
- NIC Solarflare Communications SFC9120 (Firmware-version: 4.2.2.1003 rx1 tx1)
- Linux (CentOS 7.0.1406 kernel 3.10.0-123.el7.x86_64)
- SolarFlare driver version: 4.1.0.6734a

Client Machine

- Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz (2 CPU Hyper-Trading Enabled, 20 Cores)
- RAM 128 GB, 2133 MHz
- NIC Solarflare Communications SFC9120 (Firmware-version: 4.2.2.1003 rx1 tx1)
- Linux (CentOS 7.0.1406 kernel 3.10.0-123.el7.x86_64)
- SolarFlare driver version: 4.1.0.6734a

There are two physical servers interconnected via 10Gb NICs. There are 2 testing scenarios for this setup described below. The FIX message used for testing is:

```
8=FIX.4.2|9=1|35=D|49=BLP|56=SCHB|34=01|50=30737|52=20000809-20:20:50|11=90001008|1=10030003|21=2|55=TESTA|54=1|38=4000|40=2|59=0|44=30|47=I|60=20000809-18:20:32|10=000|
```

Where "|" is the FIX SOH separator.

Single Session Echo Scenario

- FIXEdge has one acceptor session configured.
- The client application has one initiator session configured.

The overall process looks like:

1. Client application connects to the FIXEdge instance and sends 200000 FIX 4.2 messages with a rate of 2000 messages per second.
2. FIXEdge receives the messages and matches it to the same session using business layer logic.
3. FIXEdge responds to the client application with the same message via the same TCP/IP connection (the same session).

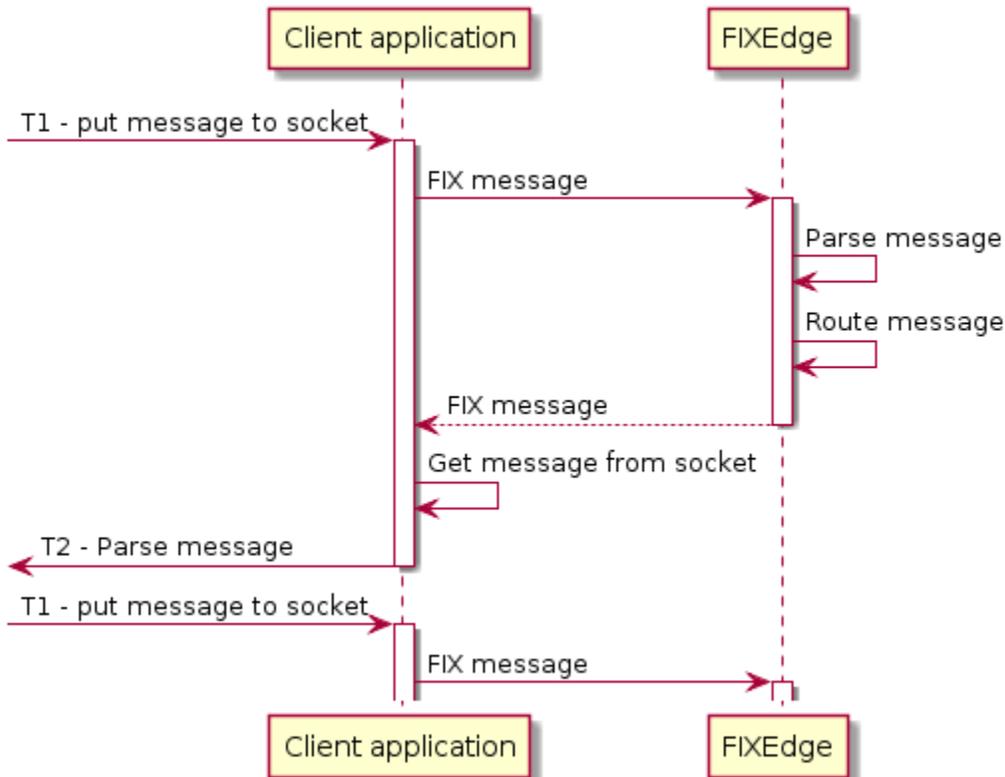
The response time measured by the client application is the difference between timestamps:

- t1 - timestamp is taken right before sending a message to the socket;
- t2 - timestamp is taken right after receiving the same message from the socket (from FIXEdge).

So the round-trip time formula is: $RTT=t2-t1$ and the measurement unit is microseconds.

The test scenario looks like:

Single Session Echo Test Scenario Diagram



Two Sessions With Conversion Scenario

- FIXEdge has two acceptor sessions configured.
- The client application has two initiator sessions configured.

The overall process looks like:

1. Client application connects to the FIXEdge instance (establishes session 1) and sends 200000 FIX 4.2 messages with a rate of 2000 messages per second.
2. Client application connects to the FIXEdge instance (establishes session 2) and starts to receive a message from another FIXEdge session.
3. FIXEdge receives the messages sent to it from client application (session 1).
4. FIXEdge uses business layer logic to route the message to another session and converts it from FIX 4.2 to FIX 4.4 protocol.
5. FIXEdge responds to the client application with the converted message via another TCP/IP connection (session 2).

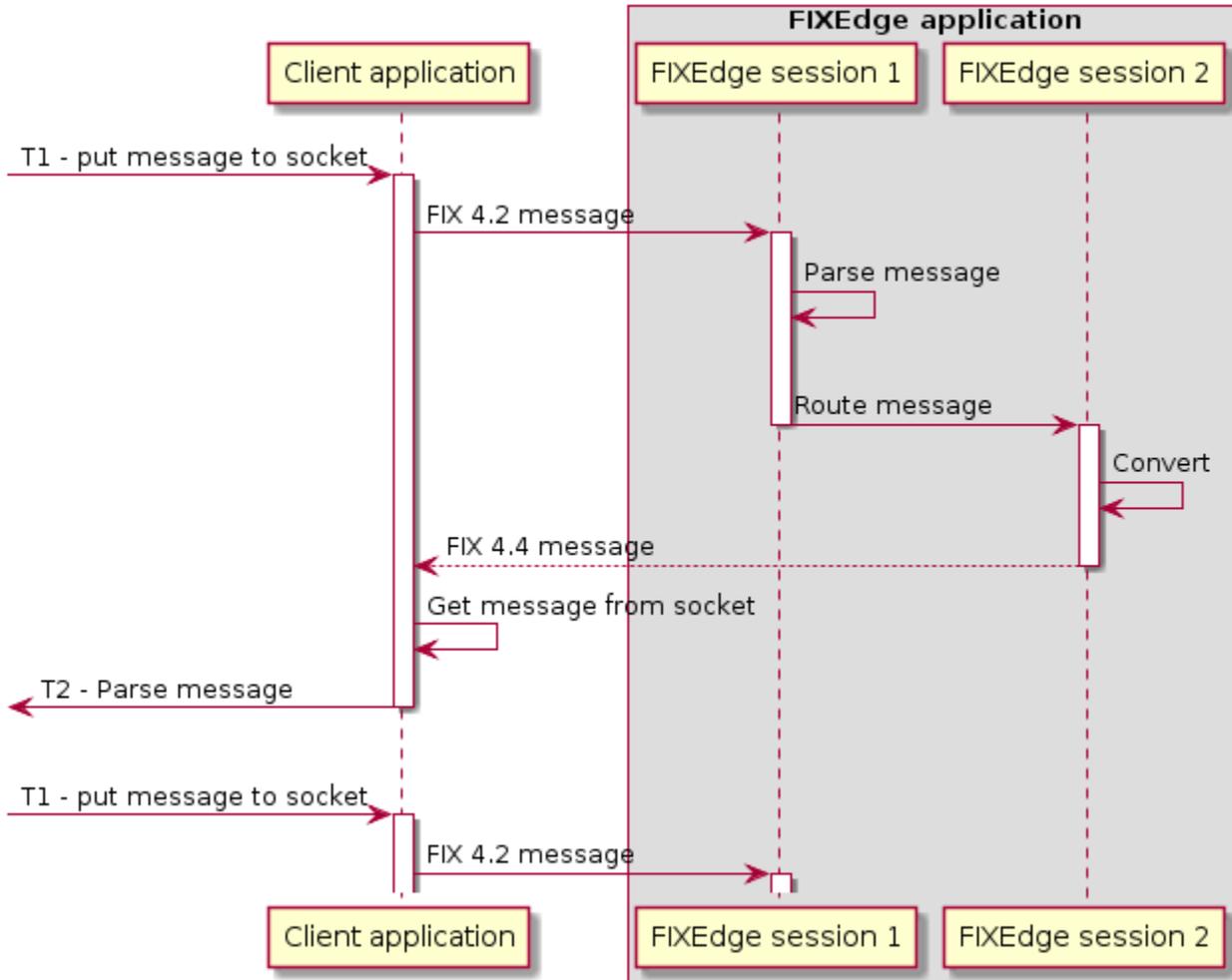
The round-trip time measured by benchmark is the difference between timestamps (just like in the previous test):

- t1 - timestamp is taken right before sending a message to the socket;
- t2 - timestamp is taken right after receiving the same message from the socket (from FIXEdge).

The round-trip time formula is: $RTT=t2-t1$ and the measurement unit is microseconds.

The scenario looks like:

Two Sessions With Conversion Scenario Diagram



Tests' Configuration

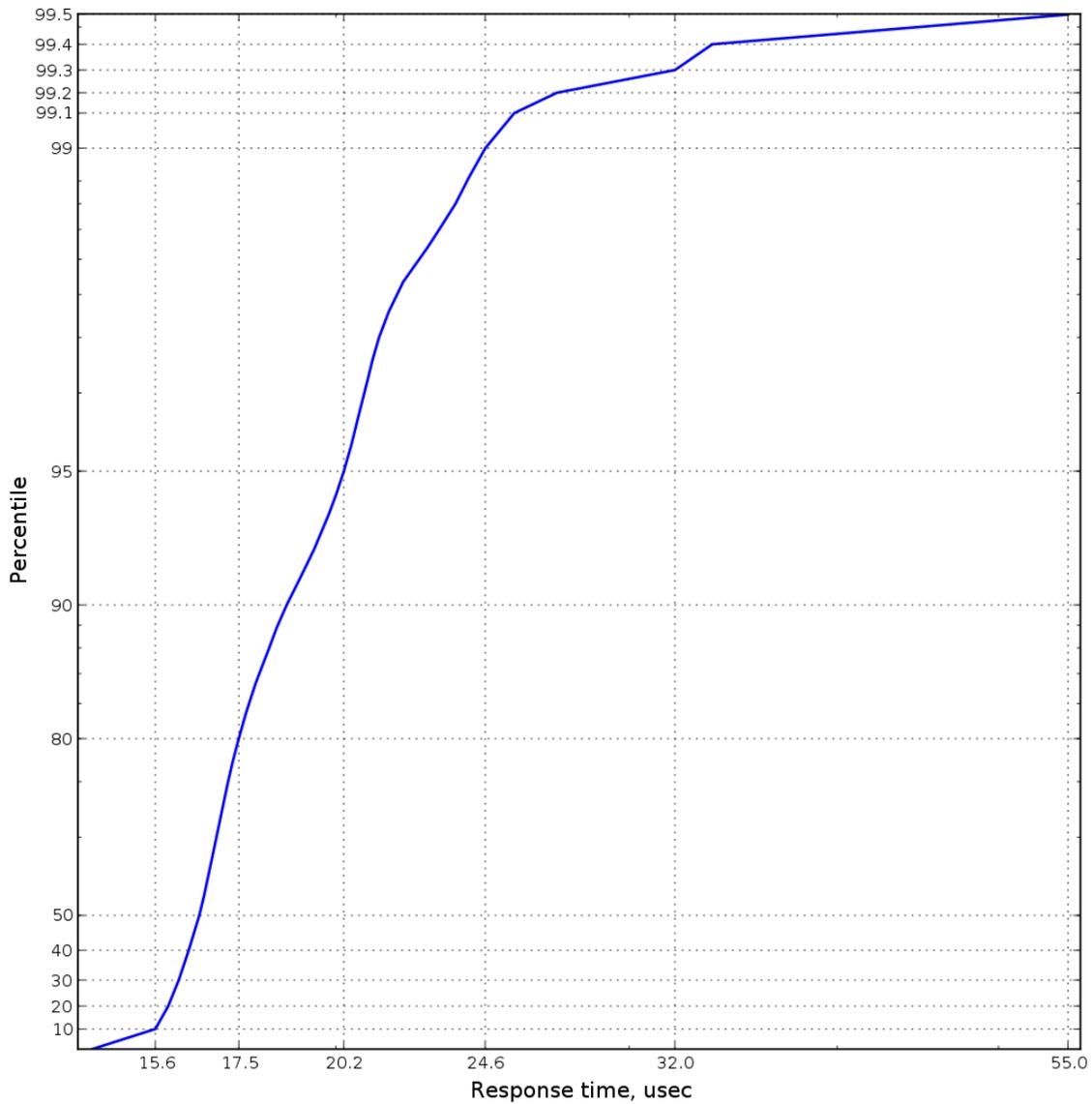
Properties	Value
Preallocated message buffers ¹	No
Thread Affinity ²	Yes
Thread pool or Aggressive Mode ³	Aggressive Mode
Message storage type ⁴	persistentMM
Solarflare OpenOnload ⁵	Yes
Message validation parameters	
MessageMustBeValidated ⁶	false
VerifyTagsValues	false
IgnoreUnknownFields	true

VerifyRepeatingGroupBounds	false
CheckVersionOfOutgoingMessages	false

1. Preallocated RAM buffers for FIX messages.
2. Pin session threads to CPU cores.
3. Aggressive Mode is when two dedicated threads created for each session - one thread to receive messages and one thread to send messages.
4. persistentMM is the message storage type represented by a memory-mapped local file.
5. OpenOnload is a Linux alternative networking stack that performs kernel bypass for the application data.
6. Message validation against FIX dictionaries takes a significant amount of all processing time so it's disabled for testing purposes.

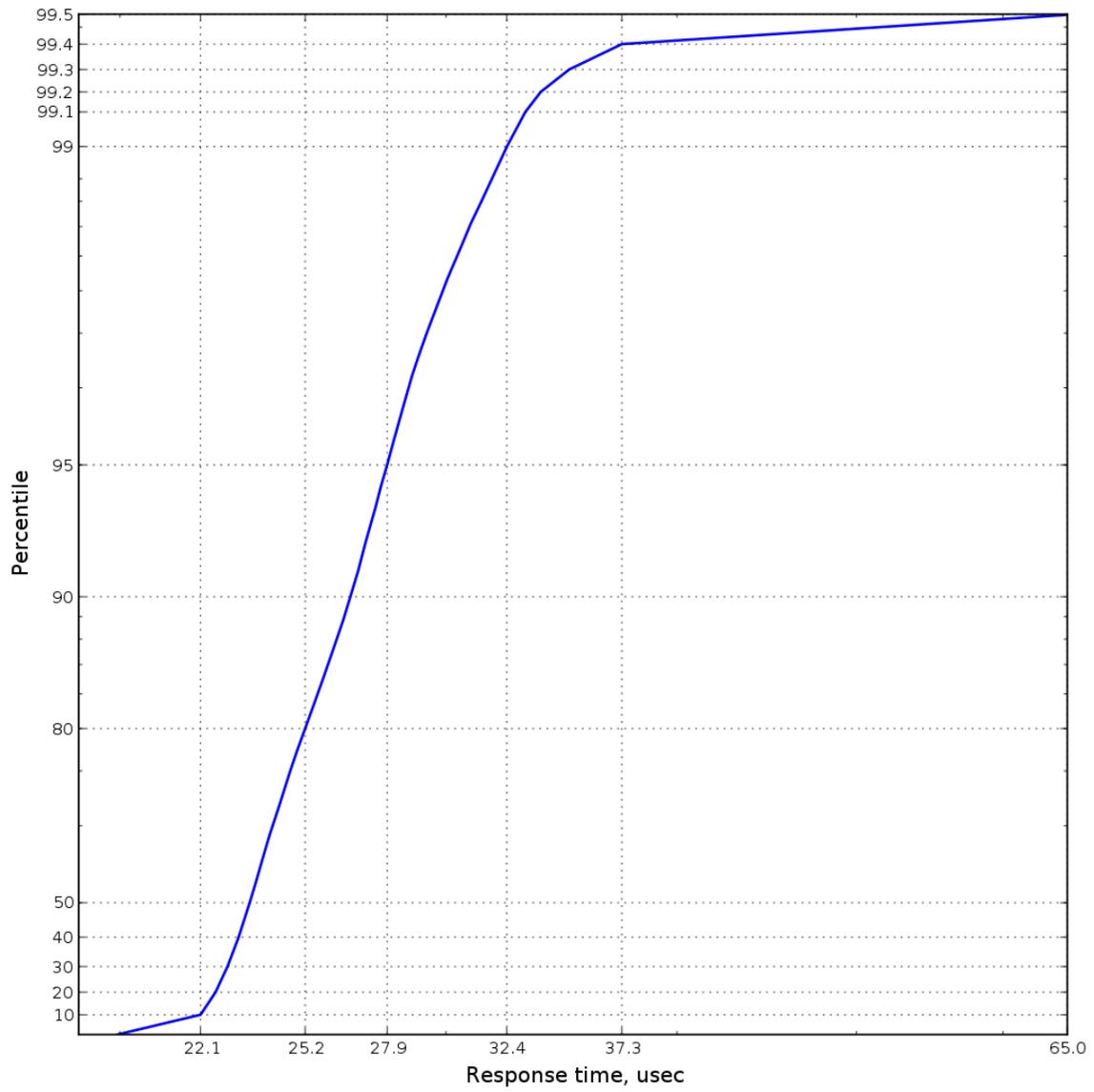
Test Results

Single Session Echo Test



Response time distribution chart (microseconds)	
Percentiles	Microseconds
50	16.6
90	18.7
99	24.6
99.5	55.0
General	Microseconds
Median Response Time	16.6
Average Response Time	17.2
Standard Deviation	6.6

Two Sessions With Conversion Test



Response time distribution chart (microseconds)	
Percentiles	Microseconds
50	23.5
90	26.6
99	32.4
99.5	65.0
General	Microseconds
Median Response Time	23.5
Average Response Time	24.3

Standard Deviation	6.8
--------------------	-----