llfix

Low latency FIX engine

Coreware

[1]

OPEN-SOURCE EDITION

Single-digit microsecond encoding & decoding (incl. message serialisations and validations)

Easy integration: header only with no mandatory dependencies

TCP Administration interface

FIX version agnostic, all versions supported

Linux & Windows

MIT licence (allows commercial & closed-source use)

OPEN-SOURCE BENCHMARKS: llfix vs fix8 vs QuickFix

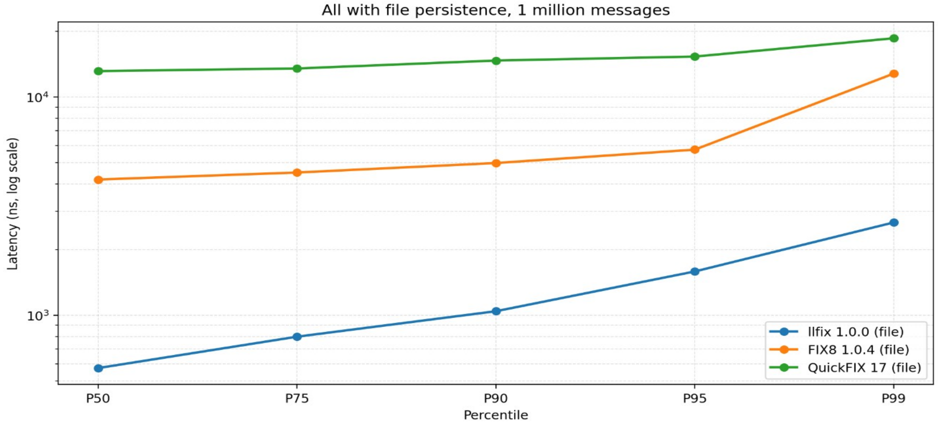
Benchmarks were conducted using the open-source editions on identical hardware and operating system environments. All engines were compiled with -O3 and executed under Solarflare Onload and threads were pinned to isoalated cores.

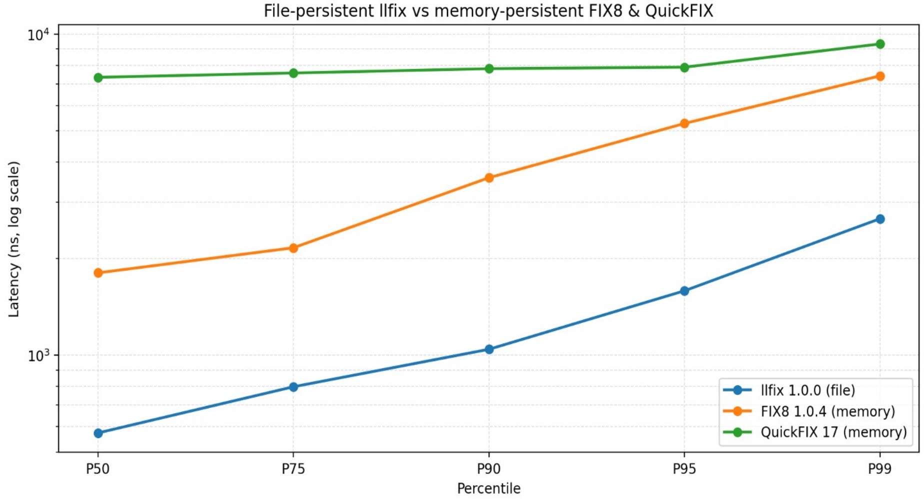
Measured latency includes encoding,and enqueueing to NIC (not wire-to-wire) for 1 million messages.

Message serialisation in llfix is always enabled and included in the measurements. llfix was compared against the memory-persistent and file-persistent configurations of FIX8(1.0.4) and QuickFIX(17).

System : Intel Xeon Gold 6134, Solarflare 8000 Series SFN8522 PLUS, RHEL9.4 & GCC 11.4.1

```
8=FIXT.1.1|9=218|35=D|34=2|49=CLIENT1|52=20251231-
18:21:36.457245600|56=EXECUTOR|50=SNDR_SUB|
57=SRVR_SUB|11=1|55=NOKIA.HE|54=1|38=10|44=10000|40=2|
59=0|453=2|448=PARTY1|447=D|452=1|448=PARTY2|447=D|
452=3|60=20251231-18:21:36.457245600|10=221|
```


All with file persistence, 1 million messages


File-persistent llfix vs memory-persistent FIX8 & QuickFIX

CORE

COMMERCIAL EDITION

Additional Features

Multithreaded FIX Server

Solarflare TCPDirect for FIX clients

SSL/TLS

High availability

Admin GUI

Code generator & dictionary validations

Full source code with no checkers, access to bug tracker & support

You can choose between header-only and static library builds

COREWARE

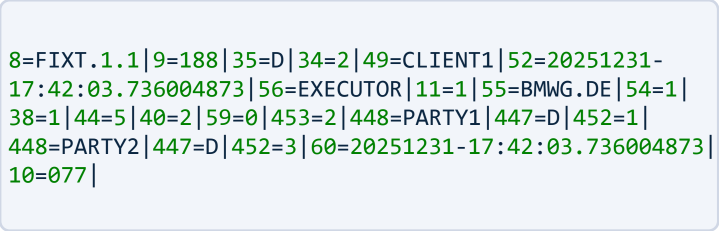
MULTITHREADED SCALABLE FIX SERVER

- Multithreaded FIX server allows to optimise message throughput by taking advantage of CPU cores

- A session is associated with one of worker threads after a successful logon. It is ensured that each session is processed by only its associated worker thread.

- Benchmark :

4.7 million messages in total from 32 clients, on loopback device
System : 2 x Intel Xeon Gold 6134 , 16 physical cores in total, RHEL9.4 & GCC 11.4.1
Tunings : Pinning to isolated CPU cores, disabled hyperthreading, maximised CPU frequency
Includes message serialisation and dictionary & other validations



```
8=FIXT.1.1|9=188|35=D|34=2|49=CLIENT1|52=20251231-
17:42:03.736004873|56=EXECUTOR|11=1|55=BMWG.DE|54=1|
38=1|44=5|40=2|59=0|453=2|448=PARTY1|447=D|452=1|
448=PARTY2|447=D|452=3|60=20251231-17:42:03.736004873|
10=077|
```

	Througput	Avg. latency
Singlethreaded	222423 messages per second	4.49 µs
Multithreaded	564177 messages per second	1.77 µs

SOLARFLARE TCPDIRECT FOR FIX CLIENTS

- Solarflare Onload accelerates networking by implementing the BSD sockets API in user space, allowing applications to run without recompilation. Solarflare TCPDirect bypasses the BSD sockets interface and its semantics to achieve lower latency.

- Measured latency includes encoding, message serialisation, and enqueueing to NIC (not wire-to-wire) for 1 million messages

System : Intel Xeon Gold 6134, Solarflare 8000 Series SFN8522 PLUS, RHEL9.4 & GCC 11.4.1

Tunings : Pinning to isolated CPU cores, disabled hyperthreading, maximised CPU frequency

```
8=FIXT.1.1|9=218|35=D|34=2|49=CLIENT1|52=20251231-
18:21:36.457245600|56=EXECUTOR|50=SNDR_SUB|
57=SRVR_SUB|11=1|55=NOKIA.HE|54=1|38=10|44=10000|40=2|
59=0|453=2|448=PARTY1|447=D|452=1|448=PARTY2|447=D|
452=3|60=20251231-18:21:36.457245600|10=221|
```

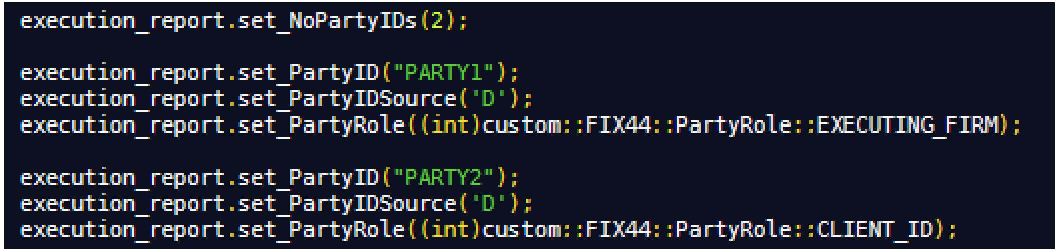
	P50	P75	P90	P95	P99
Onload 8.1.3.4	573 ns	798 ns	1044 ns	1588 ns	2663 ns
TCPDirect	493 ns	536 ns	685 ns	1437 ns	2629 ns

CODE GENERATOR & DICTIONARY VALIDATIONS

Generates C++ classes per message type with encoders/decoders

Creates enums from your FIX dictionary, reducing integration time

Built-in dictionary validation (QuickFIX format) ensures protocol correctness



```
execution_report.set_NoPartyIDs(2);

execution_report.set_PartyID("PARTY1");
execution_report.set_PartyIDSource('D');
execution_report.set_PartyRole((int)custom::FIX44::PartyRole::EXECUTING_FIRM);

execution_report.set_PartyID("PARTY2");
execution_report.set_PartyIDSource('D');
execution_report.set_PartyRole((int)custom::FIX44::PartyRole::CLIENT_ID);
```

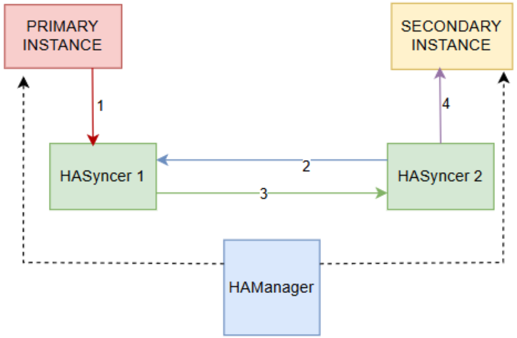
HIGH AVAILABILITY

Sequence numbers and messages synchronised reliably via TCP-based HA Syncer

Receiver-driven to minimise the load

HA Manager performs leader election for continuous availability

Manual failovers possible via the Admin GUI



Normal operation :

- 🟥 HASycner1 receives data
- 🟦 HASycner2 requests from HASyncer1
- 🟩 HASycner1 responds
- 🟪 HASycner2 updates secondary instance

Failover : In case primary instance fails, HAManager promotes the secondary instance and syncers switch roles

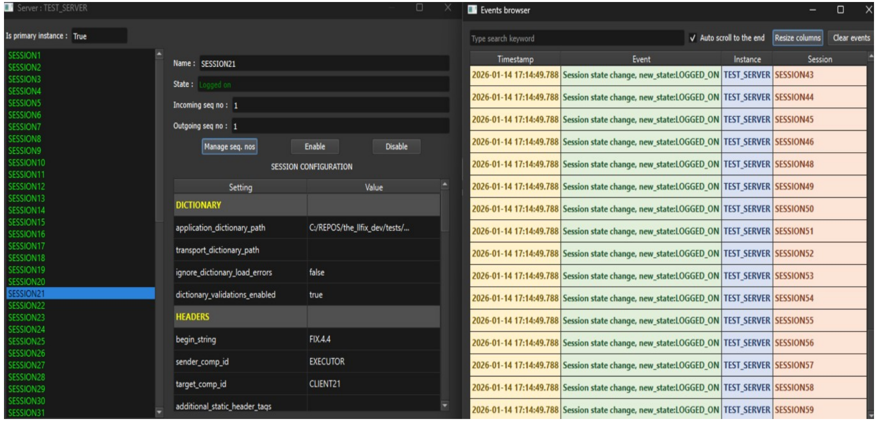
ADMIN GUI

Manage sessions of your instances via TCP

Promote/demote instances and enable/disable sessions

Manage sequence numbers, inspect configs, monitor events

Python based CLI tool also available



Is primary instance : True

SESSION1
SESSION2
SESSION3
SESSION4
SESSION5
SESSION6
SESSION7
SESSION8
SESSION9
SESSION10
SESSION11
SESSION12
SESSION13
SESSION14
SESSION15
SESSION16
SESSION17
SESSION18
SESSION19
SESSION20
SESSION21
SESSION22
SESSION23
SESSION24
SESSION25
SESSION26
SESSION27
SESSION28
SESSION29
SESSION30
SESSION31

Name : SESSION21
State : Logged on
Incoming seq no : 1
Outgoing seq no : 1

SESSION CONFIGURATION

Setting	Value
DICTIONARY	
application_dictionary_path	C:/REPOS/the_llfix_dev/tests/...
transport_dictionary_path	
ignore_dictionary_load_errors	false
dictionary_validations_enabled	true
HEADERS	
begin_string	FIX.4.4
sender_comp_id	EXECUTOR
target_comp_id	CLIENT21
additional_static_header_tags	

Timestamp	Event	Instance	Session
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION43
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION44
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION45
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION46
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION48
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION49
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION50
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION51
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION52
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION53
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION54
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION55
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION56
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION57
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION58
2026-01-14 17:14:49.788	Session state change, new_state:LOGGED_ON	TEST_SERVER	SESSION59

LICENSING

Per-Server Licence

£250 monthly or £2500 annual

Enterprise Licence

Unlimited servers

- Priority support
- Bespoke terms

Unlimited usage included for Test/QA/UAT environments for all licence types

COREWARE