Overview of the STAC-M1™ Test Harness (v2.0)

STAC-M1 provides a framework for consistent, repeatable testing of low-latency feed-handling solutions. The test specifications make no assumption about the architecture of the "stack under test" (SUT)—whether FPGA or CPU, in-process library or distributed system, Ethernet or InfiniBand, etc. The STAC-M1 specifications draw from the input of leading trading firms and vendors on the STAC Benchmark™ Council (www.STACresearch.com/council). STAC-M1 v2 is currently in a beta stage, in preparation for submission to the Council for approval.

In a STAC-M1 test, the inputs are recorded raw exchange messages played back via hardware at multiples of their original rate. The outputs are market data updates normalized to a specified structure in the main memory of a server. The tests yield numerous metrics related to latency, throughput, and other key performance indicators. Each test sequence yields statistics on the one-way latency from the time that an exchange message is available to the SUT to the time that normalized data is available to a Consumer application.

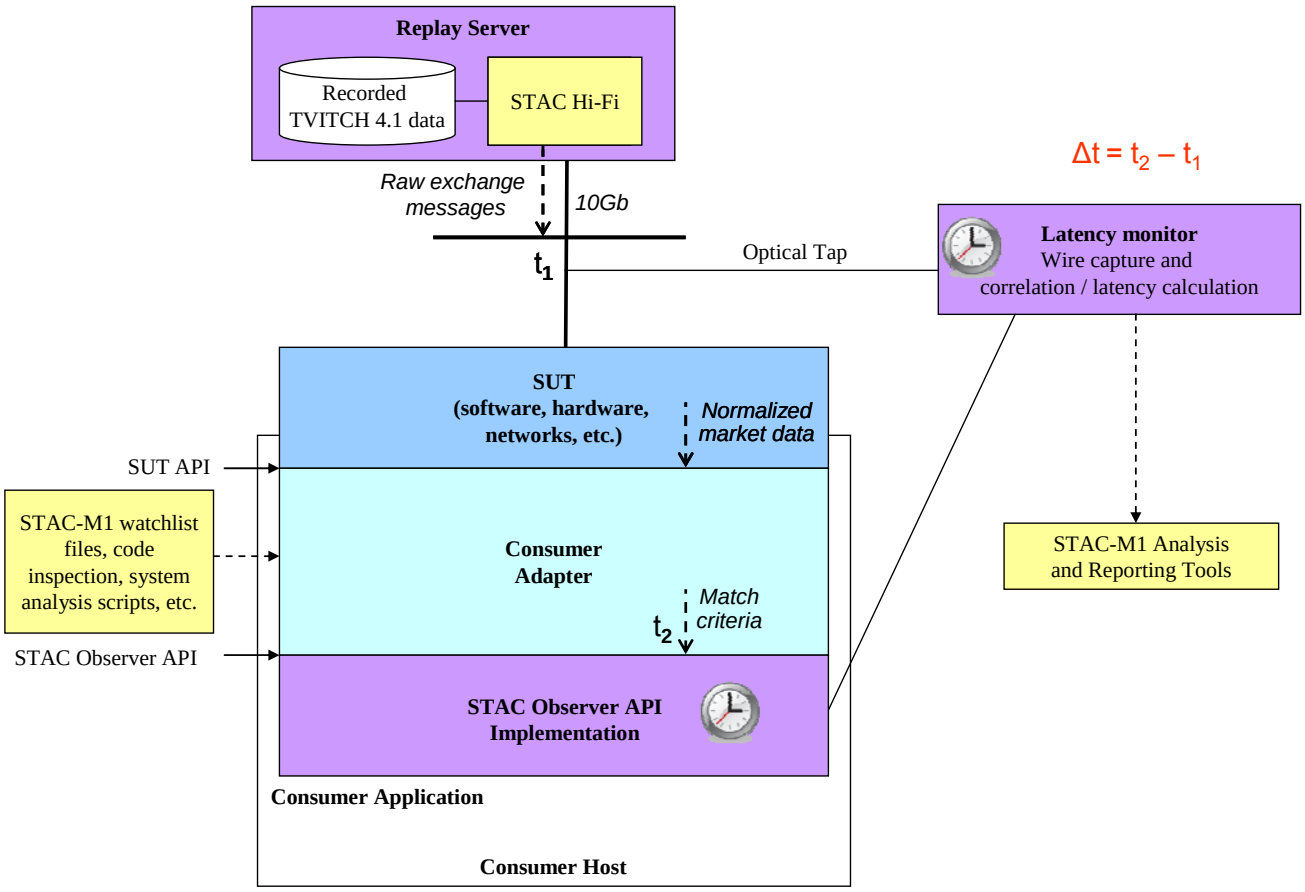

Figure 1: The STAC-M1 test setup using NASDAQ data

STAC-M1 v2 requires addenda that define variants of the tests specific to particular datafeeds. The center of focus so far for STAC-M1 v2 has been the 10 Gbps UDP-based NASDAQ TotalView ITCH 4.1 feed (the TVITCH4.1-10G-Agg addendum). Unlike all previous STAC-M1 Benchmark specifications, this addendum requires more than just data normalization and distribution. The SUT must also manage the order book for each equity, supplying the Consumer with the top 5 bids and offers in each book. Only exchange messages that change the top 5 orders are supplied to the Consumer. In addition to measuring performance during the market-open period, this harness also scrutinizes the SUT during the period leading up to market close, since that is often the most difficult challenge for order book handlers. These functional requirements and playback scenarios impose a significant processing burden on the SUT.

Figure 1 illustrates the test harness, using NASDAQ TVITCH data as an example. Recorded data plays at various rates, and the SUT is responsible for delivering updates to the test client. Update latency is shown as Δt, which is from the availability of raw exchange data on the wire to the availability of consumable, normalized data in the test client.

At the center of the test setup is the SUT, shown in dark blue. This includes all hardware and software required to deliver data to the Consumer application. These could be an embeddable library, separate processes, or even multiple machines (that last possibility is why the SUT box exceeds the Consumer Host boundaries in the diagram). The SUT also includes the SUT API library.

In the lighter shade of blue is the Consumer adapter. This is code typically written by the SUT provider, which binds the SUT API library to the STAC Observer API, which is responsible for timestamping and persisting observations. The Consumer Adapter takes in data via the SUT API, populates a 'C' structure with field values from market data updates, and posts observations to the STAC Observer API.

Components in yellow are supplied by STAC. These components are responsible for providing a repeatable workload to the SUT, validating required SUT functionality, and analyzing and reporting results that conform to the benchmark specs. They control the watchlists, field types, and other application particulars in accordance with usage patterns that trading firms have indicated are common for automated equities trading in the US. While these particulars vary from application to application in the real world, STAC-M1 tends toward the patterns that yield the most conservative performance measurements. Input data comes from STAC Hi-Fi, which leverages proprietary hardware to replay pre-recorded data with precise fidelity to the timing of the original datafeed stream, including microbursts.

Components in purple are supplied by one or more firms that specialize in packet capture and replay, network probes, latency monitoring, and time synchronization. Such components are required in order to make the STAC-M1 Test Harness software work. Use of a given latency monitor requires software development. STAC-M1 is open for integration with a variety of products, but the first latency monitor provider to complete integration with STAC-M1 is TS-Associates. The latency monitor observes messages both on the wire and in memory, ensuring that the timestamps for those observations are very tightly synchronized. In-memory observation requires the latency monitor provider to supply an implementation of the STAC-M1 Observer API. This involves software but may also involve hardware, as in the case of the Application Tap from TS-Associates. The latency monitor also correlates the inbound and outbound messages to calculate pair-wise latencies, which it then supplies to the STAC analysis tools.

For more information, please contact council@STACresearch.com.