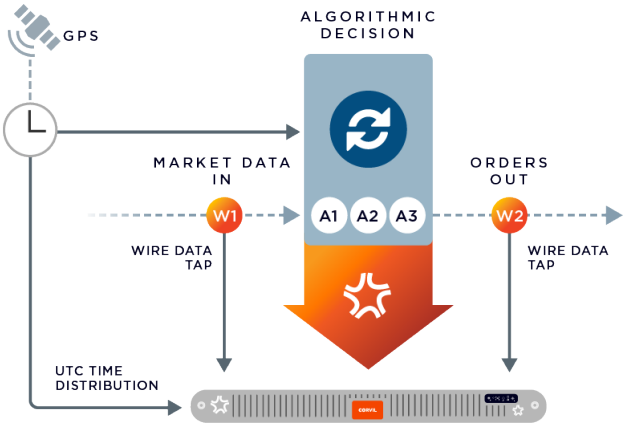
PICO Corvil Analytics | APP AGENT

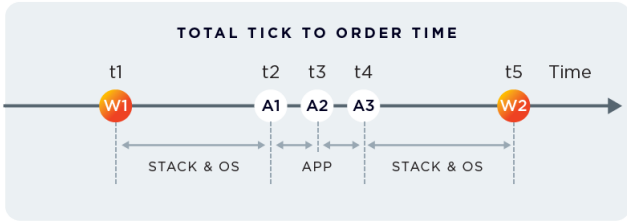
Low Overhead Precision Visibility of Application Events

The App Agent delivers precision visibility of application events with ultra-low overhead. The agent solves the problem of efficiently offloading the work of serializing, batching and publishing time-stamped application events to enable application and transaction transparency, event reporting, and performance optimization. An easy-to-integrate software library keeps your code clean, lean and fast. Combined with Corvil Analytics network visibility, we eliminate traditional agent blind spots such as network stack latency, garbage collection and process scheduling.

HIGHLIGHTS

- Ultra Low Application Overhead
  Typically < 10ns
- Simple to Use API supporting C++, C and Java
- Precision Accuracy UTC clock synchronized
- High Performance Up to 200,000 events per second
- DevOps Ready: See code perform in production





CORVIL APP AGENT

Designed for high-throughput, the agent can sustain over 200,000 events per second per core. If you need more throughput you can use multiple agent publisher threads to handle it. The App Agent is provided as a software library with a straightforward API, example code and documentation. It is easy to integrate, with minimal dependencies and includes support for C++, C, and Java.

DEVOPS READY

With ultra low application overhead, you can get the same granular performance visibility in production as you do in development and testing. Teams can now make use of Corvil's real-time analytics for both network and application performance to unify their toolsets across development and operations environments. The Corvil APIs allow full and easy integration with DevTest and production monitoring tools – for example, to retrieve a latency distribution for every hop from wire through application after each automated test run.

ACCURATE INSTRUMENTATION

With Corvil's unrivaled experience in accurate timestamping and clock synchronization, you can have complete confidence in the App Agent data. The App Agent monitors the stability of the host's UTC clock against the free-running CPU counters to flag anomalies that could indicate unreliable clock synchronization. Where external UTC synchronization is not provided, you can still measure accurate intra-application latencies on a single host.

EXAMPLE USE CASES

MiFID II Reportable Events
Provide UTC synchronized reportable events from decision events within your trading application with minimum disruption and overhead, even when you have to log software events.

Application Performance Monitoring
Add reliable microsecond performance visibility to your application development and test process.

DevOps Production Monitoring
Low overheads mean you can afford to get the same visibility in production, so DevOps can monitor performance in both dev and production with a single toolset.

Wire-App-Wire Latency Measurement
Measure accurate wire-app-wire latency for full end-to-end visibility of trigger event on wire to decision event in app to response event on wire e.g., tick-to-order.

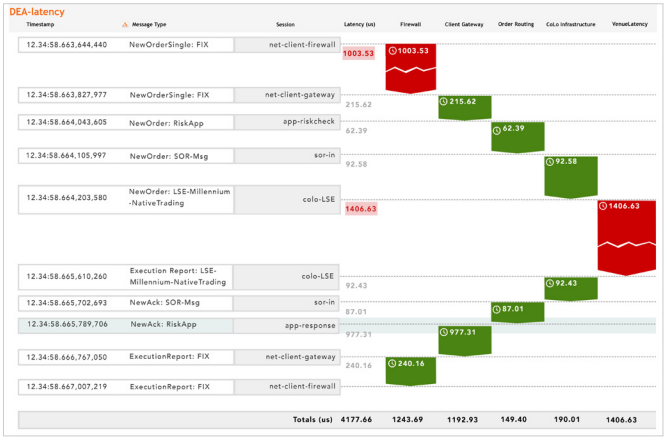
CLIENT USE CASE

CHALLENGE - Limited Quantitative Understanding of Algo Response Time Influence on Trade Outcome

The success of many trading strategies relies on the ability to react to new information and get orders to market in the lowest possible time. The firm needed to optimize tick to order time in a low-latency environment.

SOLUTION – Eliminate Blind Spots with Ultra Low Overhead App Instrumentation to Measure Algo Response Time

The firm selected Corvil App Agent to instrument their application. App Agent enables low-latency instrumentation of key events in their code with accurate timestamping. Events were then streamed to Corvil appliances, allowing application events to be correlated with wire data (tick and order) events, giving a real-time accurate picture of trade latency.



ULTRA LOW OVERHEAD

The App Agent is especially designed for use in latency-sensitive and high-throughput environments. In critical code sections, the agent code simply reads a low-level CPU counter, keeping the impact to less than 10ns. At a later point, the API is used to forward these raw timestamps to the publisher component. The publisher component takes care of the alignment with real (UTC) time, as well as the batching and forwarding of events, supporting over 200,000 events per second. For minimum overhead, developers can use the simple "getRawTime" call in the critical code section, and publish the event thereafter. The App Agent takes care of the publishing of data for logging and/or latency calculation.

ELIMINATE BLIND SPOTS

Traditional application instrumentation suffers from several difficult-to-eliminate blind spots. For example, if a software process is waiting for a hardware event, such as the arrival of network data, the application has no way of knowing how long it takes before it is notified of the event. System events such as garbage collection and process scheduling can occur in ways that make it difficult to detect with pure software instrumentation.

In contrast, network instrumentation sees the complete end-to-end time of data processing, from network-packet-in to network-packet-out. This gives clarity and accuracy to questions such as: How long does it take from receiving a market-data-tick to generating an outgoing order? But network instrumentation cannot provide a breakdown of hops inside the application, nor can it see events such as software callbacks over RDMA.

With the App Agent, Corvil makes it possible to combine the best of these worlds, to track data end to end – from the network, through application hops, and back to the network. With multi-hop visibility, you can immediately see the latency hot spots, allowing you to optimize applications and monitor operational performance.