Consuming Time in Financial Servers – Instrumenting your Applications

Version V1.0

May 2011

Confidential

Contents

1 Introduction

So you've got the best, most accurate time to the NIC card in your server because your system is using the Korusync PTP PCIe card.

Now what ?

Precise time is the fundamental foundation upon which Latency metrics and critical real-time application decisions are built.

How that time is delivered into the server's applications and how it is used and consumed by the applications is an issue that needs to be solved on a bespoke basis.

If you are looking to utilise time for Pre-trade qualification, in real time, then you have a completely different requirement to the use of time for back testing purposes. Different server architectures or application requirements will determine the optimal basis upon which time is delivered and consumed.

Korusys provides a "Timing Toolkit" of API calls and Linux daemons that allow the user to select the optimum solution for their application. With our Timing Toolkit and our consultancy expertise we can ensure you have the most efficient and accurate timing solution for your needs.

2 Instrumenting the Internal s/w

Instrumentation in HFT architectures have traditionally been based upon the network model of transaction logging. Multiple servers communicate via standard network connections and network based systems 'sniff' the transaction packets off the wire, timestamp them and store for replay.

Whilst this architecture suffices for packet latency measurements it suffers from several failings when applied to instrumenting HFT algorithms.

- The timestamp is made as the packet is in the network and not at the point where the trading decisions are being made.
- Network instrumentation systems typically suffer from the fact that they do not have a coherent, accurate, timebase at the servers or at the network tap points
- Multi server architectures are scaling down to single server multi-core architectures
  - The previously observable network traffic is now hidden in traffic between cores

In order to address these issues a new paradigm in time based instrumentation is required.

Instrumentation Inside the Algo

Traditional methods of measuring packet latency at a network level only provide half the picture that an algorithm designer needs.

As multi server systems collapse onto multi core systems, and hence server to server communications are replaced with core to core communication the observability of the system reduces accordingly.

For true observability, instrumentation needs to be embedded inside the machine.

The panacea for instrumenting HFT algorithms would be to be provide a zero overhead way of measuring and recording, to the clock cycle level, events within the application code.

Measuring at this level allows latency and true time to become a variable in the trading decision.

Recording at this level allows for true repeatability in back testing for algorithm development.

The Korusys Application Track and Application Clock tools uniquely deliver this ability to the system.

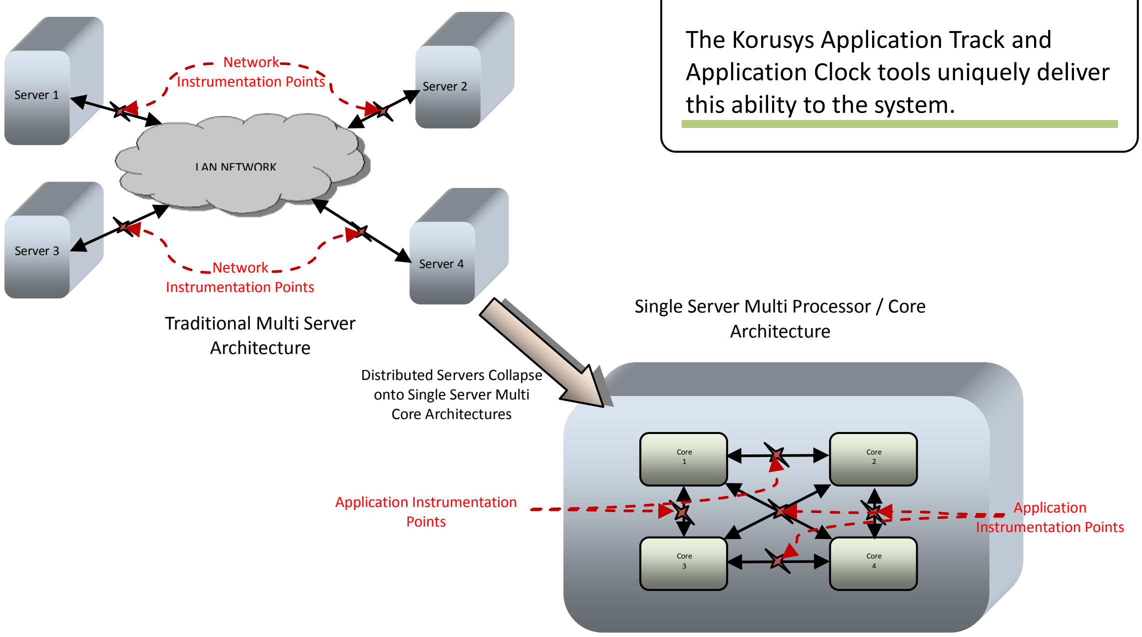

Figure 1 : The Multi Core Observability Problem

3 The Timing Toolkit

The Korusys Timing Toolkit is a set of plug and play components that allow the user applications to make best use of the highly accurate time provided by the PCIe card.

The Timing Toolkit consists of four functional pieces that can be selected by the user as appropriate to the application requirements.

The Timing Toolkit

Korusync API
This is the standard API function set that is utilised by the various components of the Timing Toolkit or can be accessed directly by user applications.

Korusyncd
This is the standard Korusync daemon supplied as part of the cards ecosystem which initialises the PTP card and disciplines the Linux system time to within tens of nanoseconds of the PTP card time.

ApplicationClock
This is a super lightweight software clock that resolves all of the common issues prevalent in the system clock provided by Linux. Low latency access, scales across multiple cores, truly monotonic and nanosecond accurate.

ApplicationTrack
This is an extremely lightweight instrumentation daemon that provides a method of timestamping and logging points within the applications themselves with minimal overhead. The highly precise log can then be used for back testing or regulatory purposes

Figure 2 overleaf depicts four application scenarios where various combinations of the Timing Toolkit have been selected to address some typical use case scenarios.

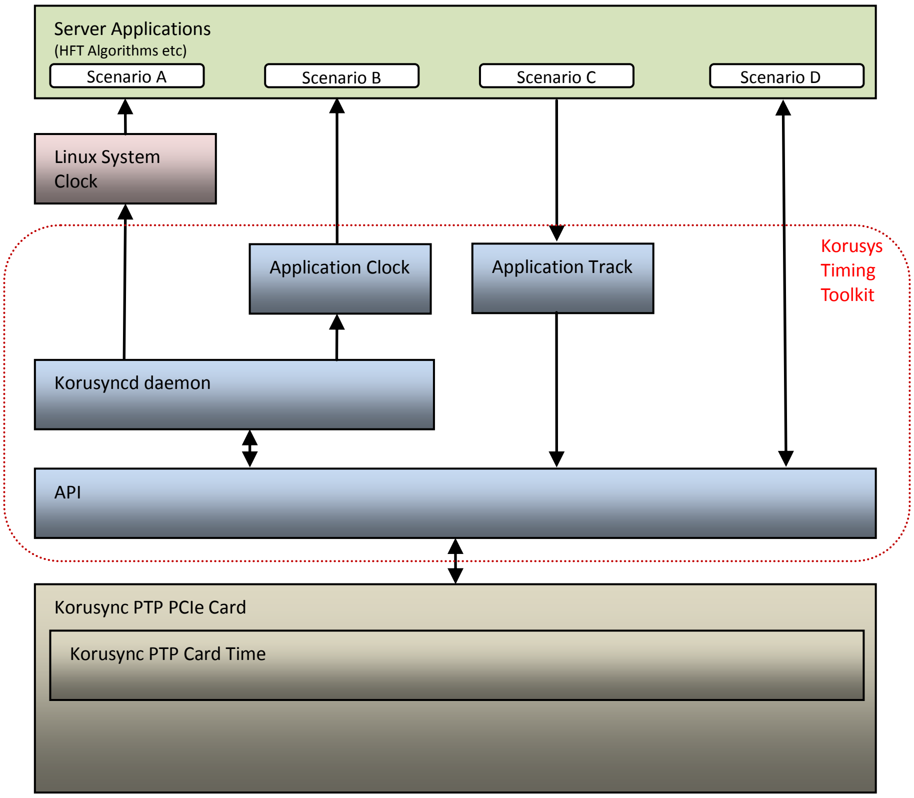

Figure 2 : Timing Toolkit Use Case Scenarios

These use case scenarios defined in the following table are implemented by utilising the Timing Toolkit in the manner depicted in Figure 2 above.

Scenario A	Applications that utilise the system time in Linux can benefit from a much more accurate representation of system time by utilising the Korusync Daemon. This scenario would typically be used where it is not desirable for the current application software to be re-tasked to use different API calls. Instead korusyncd disciplines the Linux server clock to high accuracy and thus allows standard Linux access to this more accurate time.
Scenario B	Applications that utilise the system time in Linux can benefit from a much more accurate and, crucially, extremely low latency access to time by utilising the Korusync AppClock. This scenario would typically be used where the overhead of accessing Linux system time is adversely affecting the operation of the application software itself. Applications use AppClock API calls to get the time in place of the standard Linux system get time calls. AppClock API calls scale across multiple cores, provide extremely low latency access and nanosecond precision time that is truly monotonic.
Scenario C	Applications that are timing and performance critical can benefit from the instrumentation functions that are provided by the Korusync AppTrack daemon.

	AppTrack is an extremely lightweight logging daemon that can be accessed by any application or process in the server. Simple, user defined, text strings are passed to the daemon to denote when events or instrumentation points within the software have been reached. These text strings are precisely timestamped, wrapped in a syslog message, packetized and sent through to a user defined external server where system logs are kept. All functions are provided by the PCIe card and have no load on the system software.
Scenario D	Applications can utilise the standard API calls directly into the PTP card. This scenario would typically be used by customers who wish to use the card as the primary source of time and utilise their own software systems to provide time internally on top of this timebase.

3.1 Precise time into the Application – Application Clock

Time and latency within the machine can be an important input to real time HFT algorithm decisions. The more precise and accurate the time available, then the more granular and statistically accurate basis algorithm decisions can be made.

This *requires* time to be available within the machine directly to the applications in a lightweight, monotonic fashion.

Time can be made available from the Korusync PCIe card in any way the user requires to match the requirements of the system design.

- Korusyncd daemon can discipline Linux system time to within less than 100 ns of PTP card time. But access to Linux system time can have its own limitations on latency and accuracy when software actually tries to consume the time.

- An API call can be made directly to the card to retrieve the current time. This will be an extremely accurate time however, as with all PCIe reads (on any card) it will have a significant latency. The time returned will be extremely accurate as to when the request hit the PCIe card but will not include the latency of the actual read time through the North bridge and onto the PCIe bus and back again. Beware of claims that an API call is the most efficient way of getting time to your system, these claims typically ignore this latency through the bridge and PCIe bus.

Korusys have solved these issues with the release of the Application Clock daemon which is provided as a drop in replacement for the Linux system clock.

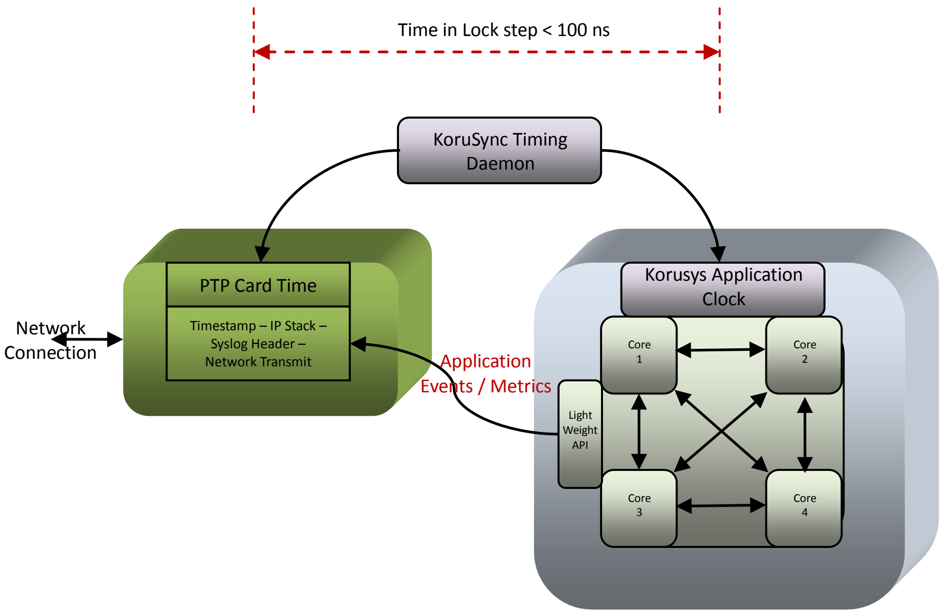

Figure 3 : Application Clock

Key features of the Application Clock :

- Access to it is ultra lightweight compared to the Linux System clock. Everything is in user mode, no need for context switching into kernel mode.
- High resolution and accuracy. Nanosecond resolution and accurate to within 100 nS of PTP card time
- Scalable and monotonic across multi-core systems.

3.2 Logging Events for Back Testing – Application Track

One of the key issues for HFT algorithm development is the ability to understand the latencies and processing time *inside* the system. Providing this observability inside a live system was traditionally considered too inaccurate and too high an overhead in applications, such as HFT, which rely on ultra low latency, high speed processing.

Other vendors propose solutions to these problems by offering logging with timestamping at an external card. Unfortunately because the external cards time is not consistent with Linux system time upon which the algorithms rely, the two times drift away from each other and thus nullify the usefulness of the recorded statistics. The real time decisions made by the applications used a different time value to the timestamped log making your back testing unrealistic and irrelevant

Application Track solves all these issues and provides the ability to record events inside the algorithms with highly accurate timestamps and in a manner that has no perceivable effect on the operation of the algorithms themselves.

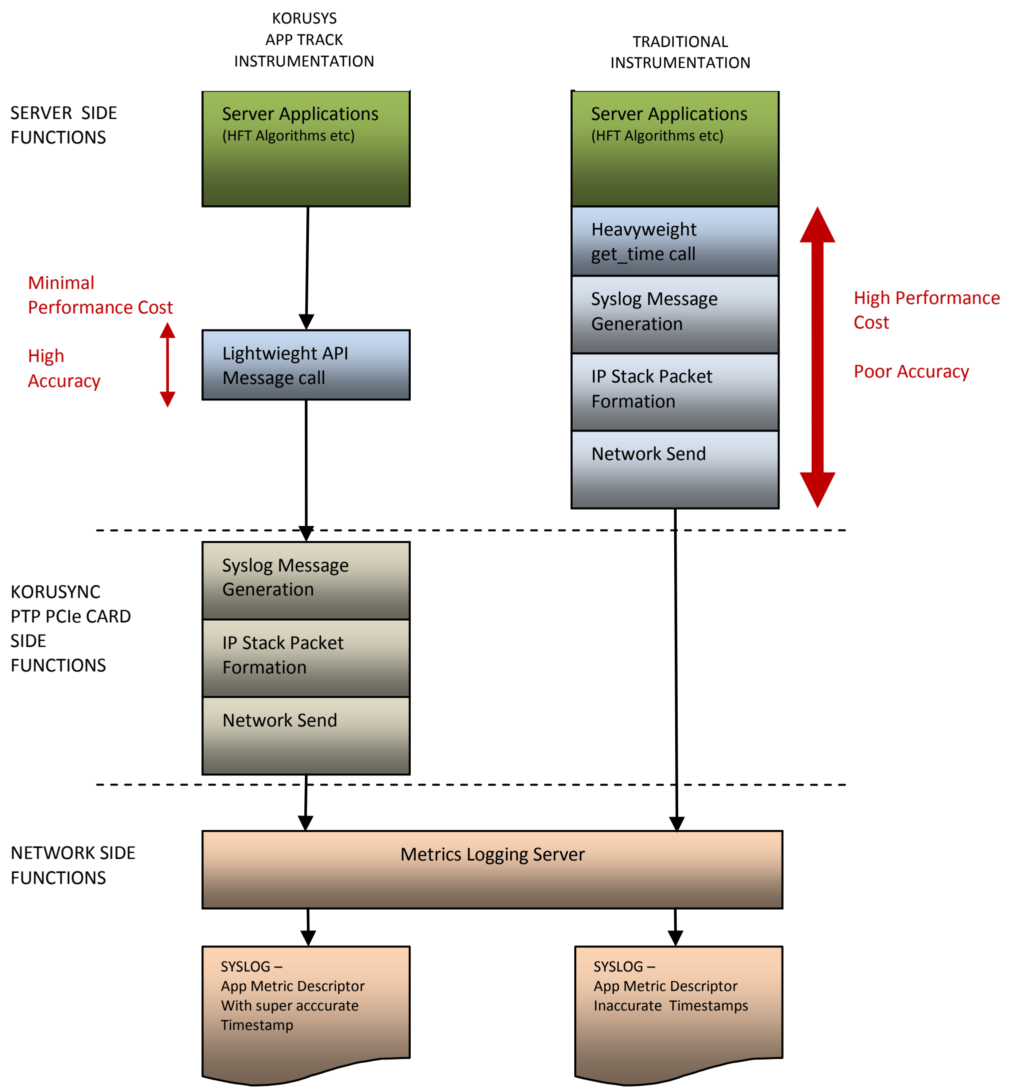

Figure 4 : Application Track Benefits of Instrumenting Internal Processes and Events

Linux system time (or the Korusync Application Clock) is locked to the PTP card time so the timebase on both is coherent, consistent and highly accurate.

The overhead of calling the Logging function is negligible because all the heavy lifting (IP stack functions, network send, timestamping, syslog messaging) is done on the PTP card itself.

4 Summary

The Korusys PTP PCIe card and associated Tming Toolkit suite of software provides an unprecedented level of timing accuracy and associated benefits to the HFT Server infrastructure.

Best in class synchronisation expertise, from a company who have been designing and implementing PTP solutions since the protocols inception, is coupled with a super lightweight software framework to allow the full potential of the timing to be utilised.

The Timing Toolkit provides a one stop shop to solve timing, recording and utilisation issues prevalent in current HFT architectures.

5 About Korusys Ltd

Korusys Ltd are leading experts in packet based synchronisation techniques providing both consultancy services and synchronisation products to various market segments.

Korusys Ltd is also a trusted provider of Electronics Design Services. Focused primarily on FPGA, ASIC, and Embedded Software design and development, Korusys Ltd has earned a reputation for high quality, right first time developments for a wide variety of clients.

Please visit us at http://www.korusys.com for contact and product information or visit our reseller partners at http://www.chronos.co.uk