**ULL (Ultra Low Latency) Architectures for Electronic Trading**  
*NYU SPS Summer 2017 Session 9 sessions (approx 2 ½ hours each) –– Ted Hruzd*   
Tuesdays May 30, Jun 6-13-20-27, July 11-18-25, 6:15-8:45 pm (Thurs July 6 will be online class to complete by July 11)

On-Line Registration - <https://www.sps.nyu.edu/professional-pathways/topics/finance/asset-management-and-investment-strategies/FINA1-CE9515-ull-ultra-low-latency-architectures-for-electronic-trading.html#more-details>

**Course Objectives**

Develop advanced skills in architecting electronic trading (ET) and market data applications for ultra low latency (ULL), for competitive advantage, and for positive ROI. At end of course one will have developed expertise in end-end architecture of ET applications and infrastructure, including:

* Tick-2-Trade applications with single digit micro seconds, even with sub 1 micro seconds
* How to architect for deterministic latencies even in times of volume spikes
* Why ‘Meta-Speed’ (info how to used speed) is more important than pure speed
* Proper use of multi-layer 48 port $18K ULL switches, FPGA’s, GPU’s, MicroWave technologies
* Market Data Feed Handlers in FPGA; Order Books in Intel Cores
* Integration of FPGA’s and Intel cores via high speed caches, eventually FPGA’s and cores on same die (Intel-Altera current and upcoming enhancements)
* When, how to architect market data order books and FIX engines in FPGA based NIC’s
* Multi core, high speed cache Intel based servers
* Linix 7.2 kernel and NIC tuning
* Kernel bypass technologies including RDMA and LDMA
* Leading FPGA based NIC(s) – from SolarFlare, ExaBlaze, Enyx
* Single tier (or simplified spine-leaf); Ex: from Plexxi networks
* Layer 1 and multi layer network switches (Metamako, ExaBlaze, xCelor, FixNetics)
* Innovative, deterministic latency Market data routing – from start-up LightFleet
* SDN (Software Defined Networks) – when applicable for ULL trading applications
* New binary FIX protocol for ULL order routing
* ULL messaging middleware (29 West LBM/UME) and 60 East Tech AMPS
* ULL software design (deep vectors ex Intel’s AVX-512 & multi threading – OpenMP, TBB)
* Storage, including NVME Flash
* Tools (some free) to attain performance optimization insights
* Network appliances - detailed timings/analytics - network, market data, and order routing – Corvil, Instrumentix, SolarCapture
* Big Data and Event Stream processing, real time analytics for seeking alpha (trade opportunities)
* Fundamentals of FPGA design and programming
* Network performance analysis via WireShark and potentially also via Corvil
* Programming trading algo’s via basic Python
* Machine learning / neural networks for seeking alpha via basic R programming
* ROI analysis

**PreReq – (for most, expecting basic to intermediate expertise, unless noted)**

* Most important: at least 2 years working with electronic trading applications/infrastructures as Developer, SA, network admin/engineer, Architect, QA analyst, tech project mgr, operations engineer, manager, CTO, CIO, CEO, vendor or consultant providing technology to Wall Street IT,
* TCP/IP, UDP, multicast (basic knowledge),
* Linux OS and shell or scripting (ex bash, perl); at minimum basic familiarity of output and usefulness of core Linux commands such as sysctl –a, ethtool, ifconfig, top, ls, grep, awk, sed, and others listed later in this syllabus
* Intel servers, cores, sockets, GHz clock speed, NUMA
* Network routers, switches
* 1 or more network protocols from BGP, OSPF, EIGRP, MPLS, IB
* FIX protocol
* Market Data, at minimum contents of equities consolidated feeds
* Visio (will use for homework assignments)
* Python (very basic will be fine – a 2 hour reading assignment will be arranged for beginners). We will use a text written for traders with zero programming experience that quickly trains them how to use small set of Python for creating trading algo’s
* R programming (nice to have. Will use basics that one can learn in 1-2 hours),

**Course Logistics**

* 9 sessions @2 ½ hours each 6:15-8:45 pm, NYU MidTown – 11 W 42nd Street, Room 1036; start May 30 (Tue) once a week, end July 25; no class July 4; substituted with online July 6 class
* Tech book(s) to download to kindle
  + Architects of Electronic Trading, Spephanie Hammer, Wiley 2013
  + Ultimate Algorithmic Trading Systems ToolBox, George Pruitt, Wiley, 2016
  + ***OPTIONAL -- NEW: Machine Learning with R – 2nd Edition by Brett Lantz, PACKT Publishers***
* Multiple web site links to technical white papers and tech analyses (ex [www.nextplatform.com](http://www.nextplatform.com), <http://intelligenttradingtechnology.com/>, <http://datamanagementreview.com>, [www.tabbforum.com](http://www.tabbforum.com), [www.tradersmagazine.com](http://www.tradersmagazine.com)
* Visio OR simply draw on whiteboards, send pictures (some homework assignments)
* Extensive use of white board by instructor and students. Sessions will present students with few infrastructures to architect per specific business success criteria
* Grading:
  + 1/3 class participation in in-class architecture designs -white board sessions)
  + 1/3 quizzes / tests
  + 1/3 - Homework – visio, wireshark analysis, basic python algo programming

**Session 1 – Tue May 30**

**ULL components: CoLo, switches, FPFA, servers, OS, networks, software & middleware, market data**

* Will present **3 diagrams** with **co-lo ULL** architectures that generate orders destined for trading venues, utilizing Layer 1 switching + FPGA’s for market data & order flow with a target of sub 1 micro second Tick-2-Trade (T2T) latencies. Latencies will be deterministic, even at peak loads, as long as switch and FPGA’s process at line speed.
  + Metamako MetaMux switches feeding to Intel Cores or to another FPGA solution such as from Algo-Logic or NovaSparks
  + Algo Logic for 100% FPGA based market data – order flow solution just for CME
  + Solutions utilizing xCelor FPGA based appliances for market data normalization but Intel cores for order book builds

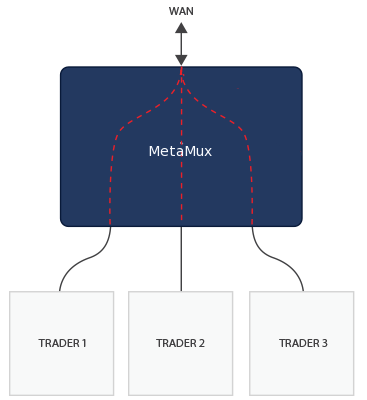
**Metamako MetaMux**



**Multiplexing Connections (Order entry)**

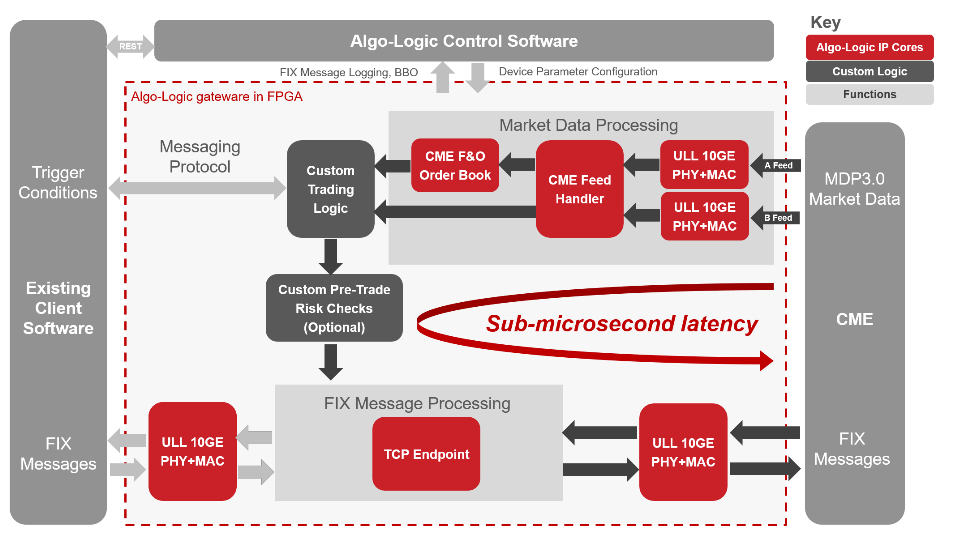
Multiple connections can be funneled into a single connection in around 80 ns using [MetaMux](https://www.metamako.com/products/metamux-48.html), assuming there is no congestion. otherwise packets will be queued. Packet ordering is strictly FIFO (first-in first-out) with no starvation if a link is saturated.

In this trading example, multiple trading clients are multiplexed into a single link with around 100 ns latency. The downstream direction can be configured to use [MetaMux's](https://www.metamako.com/products/metamux-48.html) layer 1 matrix switching in order to deliver 5 ns in that direction. Therefore, the combined latency through the switch is a little over 85 ns compared to a few hundred nanoseconds in each direction for a conventional switch.



MetaMux can both tap and aggregate up to 16 bi-directional connections in a single device, with an overhead of only 4 ns, while simultaneously significantly improving the quality of the signals, providing a re-patching capability, providing network insights, and all at a [lower cost than traditional passive optical taps](https://www.metamako.com/news/updates/metaconnect-a-better-way-to-tap.html).

**Algo-Logic**



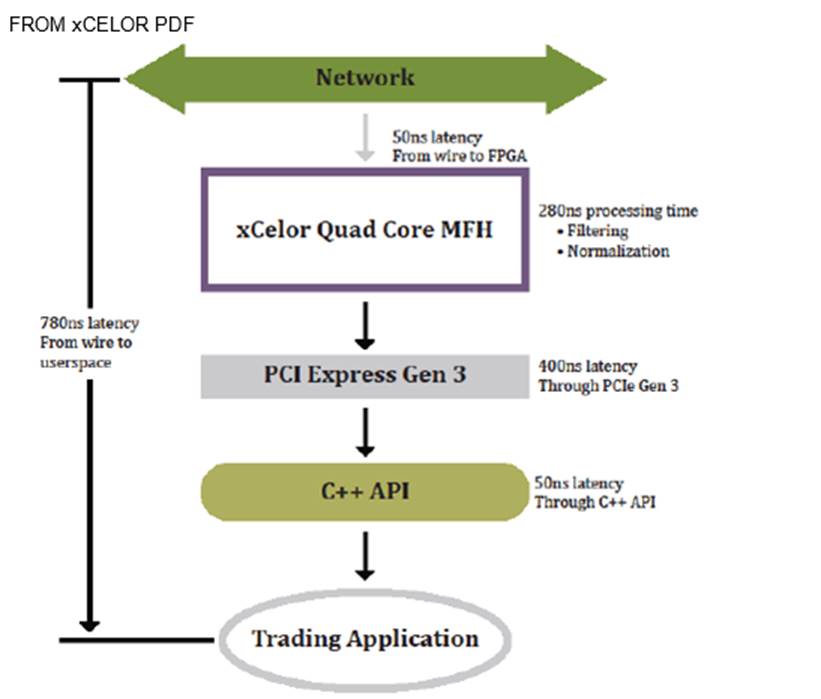
Sub-microsecond wire-to-wire latencies are achieved by receiving CME MDP 3.0 tick data directly into the FPGA on a 10G link, detecting opportunities, and placing trades in form of FIX messages encapsulated in TCP packets using the ultra low latency (ULL) 10G TCP Endpoint.

### Pre-Built Modules:

* [**ULL 10GE PHY+MAC**](http://algo-logic.com/phymac):
  + Lowest round trip latency of 89.6ns
  + Cut through packet processing to avoid buffering delays
* **CME Feed Handler**:
  + A/B faster feed arbitration resulting in earliest possible market data event detection
  + Filtering on subscribed multicast channels
  + MDP3.0 message processing and parsing
* [**CME Futures & Options Order Book**](http://algo-logic.com/futures-options-orderbook):
  + Book building for instruments that have real and implied orders
  + Reporting L2 snapshots with the best bid offer (BBO) information
  + Recovery based on CME Natural Refresh mechanism in the event of packet loss
* [**10G TCP Endpoint**](http://algo-logic.com/tcp):
  + 100% FPGA accelerated full TCP termination
  + Lowest packet processing latency
  + Full TCP protocol support including fast retransmission
* **FIX Message Processing**:
  + Session tracking
  + FIX message processing and parsing

**xCelor**

**FPGA switch**



\*\* all in FPGA – market data feed -> feed handler with filtering (Exchange – Network 🡪 MFH )

switch reads the packet to prepare it for (b) feed handler normalization in FPGA.

**xCelor** - 48 port Layer 1 software programmable switch with deterministic 2-4 nSec latency from exchange to any port or any port to another port with zero jitter

The **XPM40G-MUX** switch is most impressive. Out of the box it provides:

* Approx. 2-4 ns market data fan-out via its L1 switch component (faster than Metamako’s 5 ns)
* Instantaneous decodes of the feeds, filters over 12,000 symbols, normalizes the data --- for all major US feeds. In essence it is a FPFA feed handle that sends to order flow app (ex C++) over PCIe in 780 ns. Thus its Tick-to-Trade (T2T) is 780 ns – time of ingress from Exchanges to point were order flow app receives it. BTW the longest latency is the PCIe transmission of 400 ns.
* 5 SSD;s with each 1 up to 2 TB to store pcap data –with time stamps within 10 ns accuracy. The PCAP’s can be forwarded to Corvil for advanced network/transaction analytics, or it can be forward to a SolarFlare appliance that is completing with Corvil – not here yet all the way but with plans to get there and cheaper.

In sum, the key advantage xCelor has over competitors are (1) FPGA feed handlers – all major US feeds + (2) SSD storage. The 780 ns T2T is faster than Metamako – AlgoLogic CME feed combo of approx. 900 ns, but not as fast as The ZL appliance Marcus builds (patent pending Intel Core – FPGA cache coherency) that results in 400 ns T2T -- but then ZL does not build feed handlers.

* xCelor may have industry lead in immediate and fastest feed handler normalization with their 330 ns timings.  Other solutions (ex: Metamako) ingest the feed in 5 ns for market data fan-out or to connect with another device / app for feed normalization
* xCelor **new Layer 5 FPGA switches** now ingest 1 feed on 1 port at Layer 5, parsing symbols, then distribute to “custom” (by symbol) trading sessions off the other 31 ports.

**FIXNETICS**

* Zero Latency (ZL) FPGA based appliance + Intel cores
* Intel focus on parallelizing code more than doubling processing speeds now
* Broker (integrated with client); broker in control
* The future – shared infrastructure
* Protocol agnostic
* Multiple asset classes
* ZL Risk
* Q3

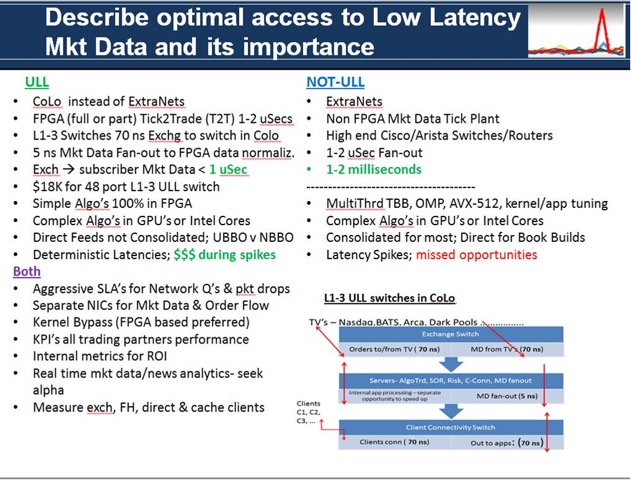
The combo of MetaMako MetaMux 48 port $18K multi layer switches complement ZL very well, servicing it with market data and FIX order flow in 5ns and 70ns respectively, and also in sending orders to venues in 70ns.

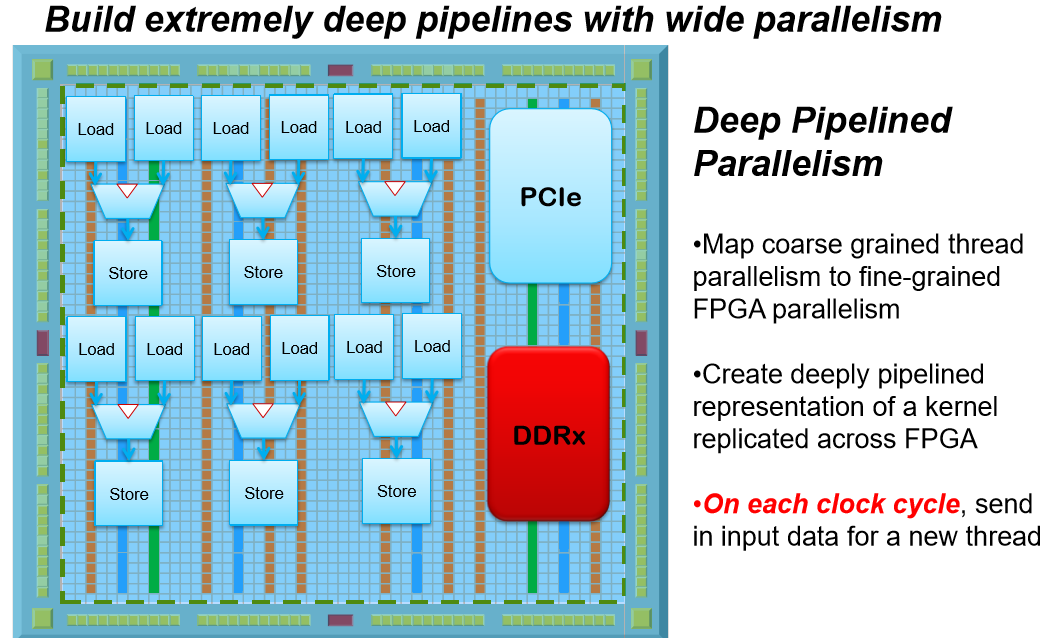
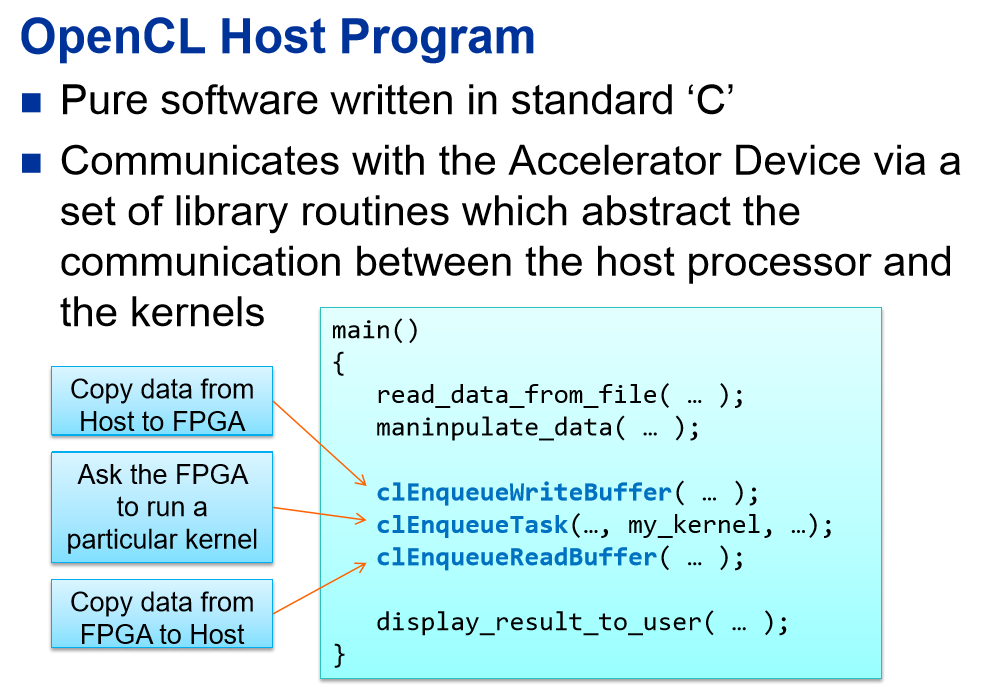
**Intel Cores + FPGA:**

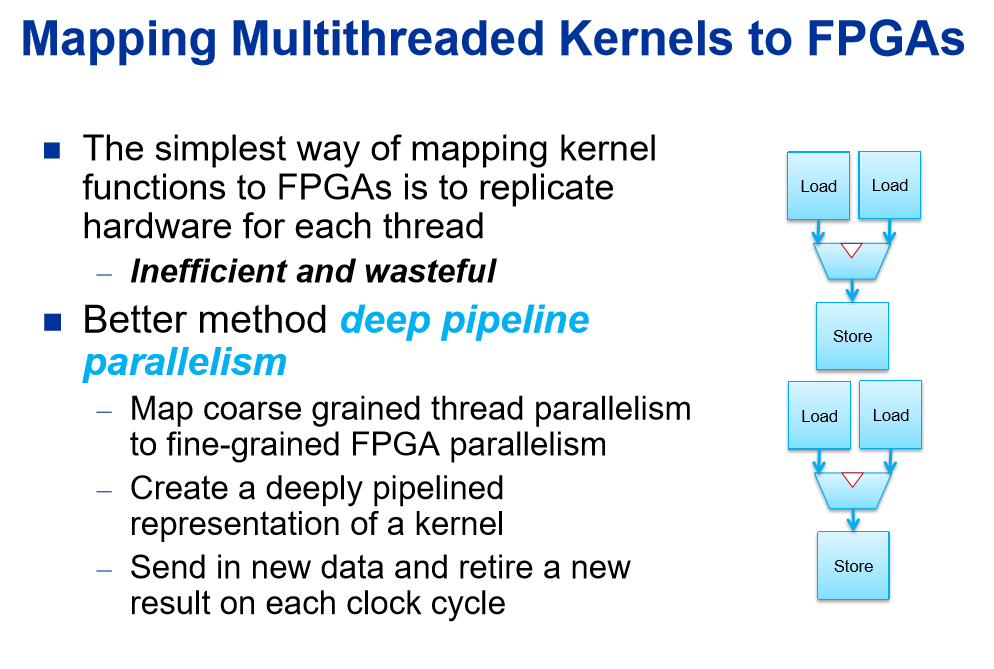
* joint FPGA/Intel-core appliance (which one?) for deterministic latencies, Tick 2 Trades (T2T) consistently under 1 uSec;  actually expect 400 ns?
* quick software to hardware transport for
* cache coherent CPU core -FPGA
* significant factor in providing deterministic latencies, even under spikes in peak message processing.  This overcomes other ULL solutions which encounter jitter during spikes.  
    
  •       The market for deterministic latencies has been growing. Very often peak trading revenue is maximized by trading applications that process with little or no jitter during most volatile market data spikes  
    
  •       All serious traders track their partners fill rates, successful trades, real time TCA, and very importantly - latencies.  Those trading partners that Q and choke during market volume spikes or trading volume can be taken off trading lists for peak volume periods or taken off indefinitely for any periods.  This should motivate more hedge funds, prop traders, market makers, agency brokers, sell side, dark pools, and exchanges to invest in partial or full FPGA based ET pending meticulous, accurate ROI projections.

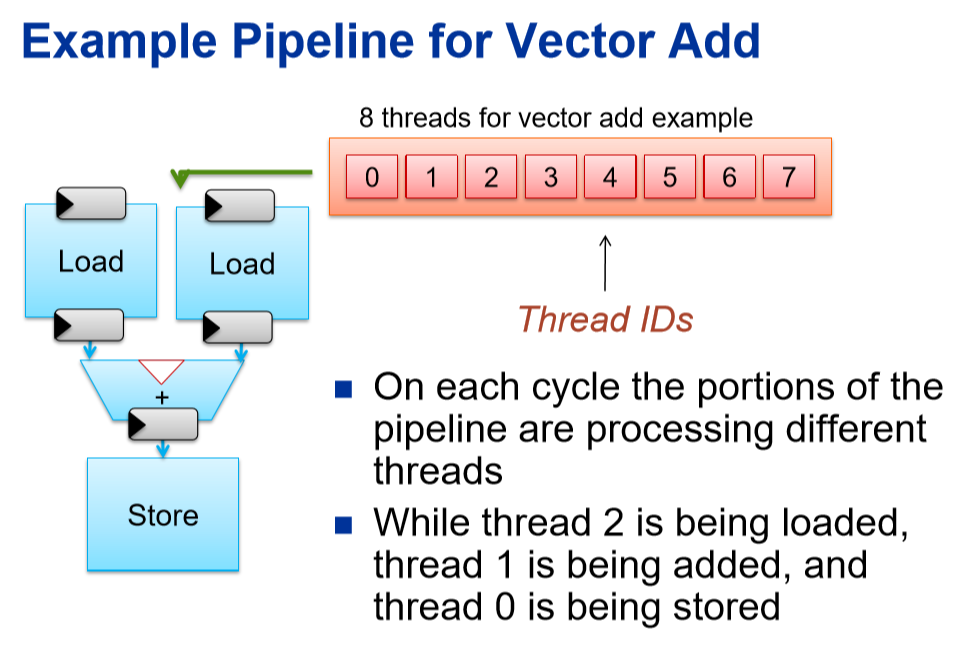
**Intel Core – Altera FPGA - Cache Coherency**

Upon receiving process data in a core on 1 socket, a mirror of that data is directly transported to cache on another socket for an FPGA process in \_\_\_\_\_. No driver, bus, or PCIE are in play.



* Will briefly present an alternative architecture utilizing a Single Tier network (Plexxi)
* We will periodically revisit the **co-lo ULL** architectures throughout this course when we cover specific architecture components in depth (Algo Trading and/or SOR that feeds this architecture, use of FPGA, and Layer 1 switching)
* Partial FPGA and non FPGA alternative architectures
* **Key advantages of FPGA** (reference -**Ted’s A-Team Doc**)
  + wire speed processing
  + deep pipelining
  + Superior parallelization (massive)
  + can act immediately on specific events (per Verilog @always clause)
  + Data transformation (ideal for feed handlers) -- contrast with GPU’s
  + deterministic latencies
  + accelerates algos of many ops on small data
  + re-program
  + create ‘cores’ in FPGA – Sweden Royal Inst of Tech
* **FPGAs have many registers + many tiny RAM blocks** which can be useful for some pipelines.
  + If you can break down your algorithm into many tiny steps with a register at the end of each step, you can feed in a new data value on each clock tick. During each clock cycle, the whole algorithm is performed, but each part is done on a different "piece" of the input data. After the clock has ticked enough times to cover the number of register stages in your pipeline, the answer for the data which went in that many ticks ago comes out.
* From Altera:
* 
* 





* Why speed of processing (& ULL market data) still matter & will for next several years at least
* Why Meta-Speed is more important than pure speed (reference Corvil-Tabb Doc)
  + Meta-Speed Deep Dive (10 minutes)
* Why Layer 1 switches
* Layer 1 switch with integrated cores and FPGA for risk checks (intro to FixNetics ZL – 40 risk checks)
* **Visio Diagram** -High speed Buy-Side real time analytics for seeking alpha (trade opportunities) & infrastructure analytics to route to proper exec broker.
  + Data – historical & RT market data, news analytics (intro to RavenPack or RP), twitter, and aggregated Coril session latency analytics via OLAP dimensions, machine learning – neural networks for routing to best exec broker for specific alpha opportunity
* Exchange (Trading Venue) connectivity
* Layer 2/3 aggregation in new switch appliances
* Some leading ULL vendors:
  + Metamako
  + Algo-Logic
  + Nova-Sparks, with Nova-Link product (to an order routing partner ex: Algo-Logic)
  + Corvil
  + Intel / Lenovo
  + SolarFlare
  + ExaBlaze
  + Enyx
  + xCelor
  + LightFleet
* Role of Linux kernel tuning for ULL – use network-latency profile & common Linux best practices
* Present some Linux configurations to critique (ex: no K bypass, same NIC for mkt data & order flow)
* What electronic trading organizations will prosper in space of ULL ET now & in future? Which may very well fail, even disappear? Why is role of ROI critical? Difficulties of proper ROI analysis
* FPGA’s – Verilog vs OpenCL programming. Present **FPGA Module**.
  + Verilog ‘module’ design subassembly with defined connections to other modules
  + Registers store state, function as variables
  + Vectors
  + Multiple counters with multiple clocks all process in parallel
  + Use ‘<=’ when adding to output Q. use with ‘sequential logic’ – will be done in parallel, then next sequence (<= is ‘non-blocking, order of assignments is irrelevant)
* FPGA –
  1. Module,
  2. UCF,
  3. Binary to FPGA
* ‘always’ clause executes whenever any entities change
  1. Ex: always @(b or e)
* Present some **FPGA Verilog code;**
* **then OpenCL code**
* More Re: FPGA’s
  1. Cost effective / quickly address competitiveness
  2. Intelligent circuitry design – advantage over GPU’s
  3. Turnkey compilers may fall short
     + However – FiberBlaze hooks into multiple paths into NIC IP’s
  4. Ideal for automated risk mgt
  5. FPGA filtered data sets for speed advantages (xCelor)
     + L5 switch
     + FH
     + xCelor’s Market Feed Handler (MFH) is a low-profile, PCIe card that delivers ultra-low latency to process market data from a myriad of exchanges.
  6. 10 Gb message processing 300 ns (2013) Stratix V

**Some Core points regarding FPGA’s**

GPUs are well suited for computationally intensive tasks. A GPU is a better choice when one wants to perform floating-point arithmetic, for example. Conversely, FPGAs excel at simple data transformation. xCelor FPGAs speed up trade-related processes. This architecture allows parallelization. An FPGA is able to operate on a physical electronic signal as it comes through a cable on a server. Theoretically, one can consume data as fast as it is delivered—at what is called link speed.

* FPGAs are actually slower than CPUs, but they are capable of much more throughput. An FPGA can be programmed to perform many tasks in parallel. Many, many instructions can be carried out in a single clock cycle, which accounts for the speed advantage.
* Previously, FPGAs were available on a network card inside a server. Now, FPGAs are being integrated into the latest network switches, which opens up new categories of applications from centralized feed handling (replacing traditional “ticker plants”) to risk management to preparing a real-time feed for transmission over microwave.
* Market participants need to meet the challenge of how to achieve SEC compliance with the lowest latency penalty. The best way is the one that entails the least transformation of data. In this area, FPGAs represent a real step forward. The latest switches that feature FPGAs embedded within them can permit users to check data as it crosses a switch—with net-zero latency penalty.
* All modern servers contain multiple CPUs. They spin off a thread to handle concurrent execution. The lack of determinism inherent in this model can lead to the problem of race conditions. With FPGAs, one can program exact operating instructions and avoid such issues. If there is a bug, you (the owner), can fix it. You are not reliant on a provider. The caveat is that FPGAs must be programmed intelligently.
* programmers can write code in C or Java and a compiler like Impulse C will turn it into FPGA code. But programs like those lack the “brains” to orchestrate complex machinery.
* “Intelligent code” means putting thought into making the most of the capabilities of the FPGA chip. This requires a human being to carefully study what’s needed and what the chip can do, and marry the two in the most efficient way. The code generated from a higher-level Java/C compiler won’t do this at all; it will do what you tell it as simply as it can, using a small part of the chip but leaving much of the parallel FPGA capabilities to waste.
* procedural languages, are incapable of expressing the parallel possibilities of FPGA.
* switches with FPGA chips embedded within them are becoming available, FPGAs can add value in areas like microwave networks. Sending market data over microwave is limited by bandwidth and reliability
* FPGA chip inside a switch can solve these problems, filtering down a full feed to only those symbols that are interesting at the remote market center, and adding in redundant updates to tackle the reliability problem.
* Hardware accelerated appliances for ULL and deterministic performance
* **Ted’s FPGA Hand-out:** -- FPGA design & programming (I/O blocks + Logic blocks, OpenCL for creating “kernels” + synchronization for parallelism )
* Why performance tends to be very deterministic with FPGA’s & why deterministic performance (latencies) are critical for HFT and algo traders
* Pitfalls of FPGA's
* FPGA's vs GPU’s, Intel Phi (**Intel Doc**), and multi cores
* Feeds in FPGA –architecture, performance, design, support
* Switch crossbars or caches for fan out with TCP distribution
* ***HOMEWORK #1 – Ted will describe 1 or 2 ET applications, then ask for an architecture that can be Visio (or picture of white board drawing – to save time). Students will be required to justify their choice(s) in 1-2 pages max.***
* ***HOMEWORK #2 –reading assignments (2-3 weeks to complete) -*** 
  + ***Architects of Electronic Trading, Stephanie Hammer, Wiley 2013***
  + [***https://developers.redhat.com/blog/2015/02/11/low-latency-performance-tuning-rhel-7/***](https://developers.redhat.com/blog/2015/02/11/low-latency-performance-tuning-rhel-7/)
  + [***https://access.redhat.com/sites/default/files/attachments/201501-perf-brief-low-latency-tuning-rhel7-v1.1.pdf***](https://access.redhat.com/sites/default/files/attachments/201501-perf-brief-low-latency-tuning-rhel7-v1.1.pdf)
* ***OPTIONAL – listen to 1 hour Webinar -- was Dec 8, 2016 Panelist for an A-Team Webinar re: perspectives on strategic ULL market data architectures & how trading firms can realize ROI, seek alpha, expand market share, address risks and compliance.***
  + ***access webinar recording here*:**[**http://bit.ly/2fXujEo**](http://resource.datamanagementreview.com/e1t/c/*VBwH5W7BsPcQW1Pqsyt7Lf_sn0/*M9wMHHyTSXFW6H5Vry3ZGj9N0/5/f18dQhb0Sq5x8Y9-RWW9jkTmC8LCvkbN56Bs7cs1bY0W6Dk5Sd1Nvw7TVM7Yn31rLBRdW8nvsK76255SxW6PQ8B55wL96LW1sL3M03MybvyW4XHP4_8q55k7Mh4zfDTGpMvW62VmxB3W5gBdN3P1zVPRJ3w1W8n7jxp66kNrjVRbkL83RFKksW1kT5_t1s6krSVXwWyZ4KDypnW1L4xyj6FKg-TW2WcyHp7B1cJ7W7v5DdL8m7Y9zW5wM1V88q5FTlW4HCfKw8W1JLSN2KSCYpTlGs9W55T8N87w3wPSW1NC3Kc1F6KSLW7zNv8v1D7ZgkVqB3kG1RBdQ4W8WQPjH1zNqGqW8SVfXK7BjF3YW8Zw2297rpD1cW91fV8q50DwN2W529GyK47n77QVsC8d045TKpbVy-wfZ2S1XpkW2PTq2v8PFsdjN491wcHmjGr1W1FJD0M7ZjRCNVpQ2sJ7YhhDdMfTlLH1KsWcW1K0ns-5r5pfsW6cBq-d7fkLv-N5mZ4YkWDM00W2vD4qF7ZT2vvW4lmyFX56fHKhW5vJkwG4jwH_YW51YX1J7xhklzW3y19vC3wqspBf4MnC5411)**.**
* **(may add more reading assignments here …):**
  + **Get a head start and read parts Ted will specify in** Ultimate Algorithmic Trading Systems ToolBox, George Pruitt, Wiley, 2016 ( 4 weeks to complete )
* **ADD link to Market Data White Paper (OPTIONAL)**

**Session 2 – Tue June 6**

**Part1: 45 Minutes: FPGA Deep Dive – Guest Lecturer John Lockwood, CEO, PhD Algo-Logic**

**Part2: start to learn about Machine Learning / AI for Electronic Trading**

Entertain review of last week + questions & discussion of assigned readings.

Next:

**Algo-Logic details (Quick Sum):**

* Pre-built IP cores, reduce Time-2-Market
* Receive CME MDP 3.0 tick data directly into FPGA on 10G, detect opportunities, place trades as FIX messages encapsulated in TCP packets using ULL 10G **TCP EndPoint** (full TCP functionality in FPGA – 76 ns latency – 20 Gbps – 140 Gbps, multiple ports/FPGA, portable between Altera & Xilinx, SF AOE, integrate with LL Library)
* Compatible with most servers, managed via SW interfaces
* GUI with REST API’s
* LL messaging protocol specifies triggers
* ROI – can replace switch ports
* “GateWare” – PCIe/DMA FPGA – fully programmable, compiles to full parallel logic, runs in FPGA
  + Rapidly transfers data between FPGA logic, processors, memory – even to OMS

***JOHN LOCKWOOD – CEO ALGO-LOGIC will provide us with a deeper dive in FPGA technologies & their futures (45-minutes)***

Next …

**role of Big Data for alpha seeking and input into ULL trading algo’s**

**Middleware, Analytics, Machine Learning, leading to end-end ULL Architectures**

Analytics & Machine Learning: to seek alpha and for infrastructure analytics

* Intro to Big Data Analytics & Machine Learning (focus on neural networks)
* Role of Java 8 new streams API
  + Speeds up extracting insight from large collections via methods such as:
    - Filter, sort, max, map, flatmap, reduce, collect
  + Use with ArrayLists, HashMaps (does not replace them)
  + Stream is 1-time use object
* Intro to Complex Event Processing (CEP) and Event Stream Processing (ESP)
* Databases - Never in the path of ULL
* Column based (contiguous memory) vs relational
* KDB and OneTick - leading players in high speed market data tick databases
* Event Stream Processing (ESP) - use ESP to seek alpha
* Combine market data with News sentiment analytics to seek alpha,
* Intro to Ravenpack news sentiment analytics
* Intro to Spark
* Role of new storage technology (ex NVMe Flash drives)
* In-mem analytics ex HANA, Spark
* Corvil - intro to how to configure Corvils and how to analyze FIX order flow with it
* **Corvil touting nano granular ns timestamps (of all packets per transaction stages)** …
  + Which leads to Machine Learning (ML) of prior customer experiences + similar customer experiences as additional input to real time alpha seeking strategies …
  + Intuitive GUI of transactions at all stages (ex: Buy Side 🡪 Sell side 🡪 exchange 🡪 order ACK then order Exec 🡪 sell side 🡪 buy side
* Machine learning, neural networks in R or Python - create equations to project latencies
* Machine learning for Latency analysis, tuning insight, seeking alpha -trade opportunities
* Programming for multi threaded trading risk analytics
* **Class Ex** – output from Corvil streams will be provided. Students will analyze and determine how latencies can be projected using neural networks (design only – no programming) -- or we may do this together in class – Ted to attain sample data as input to neural networks for latency predictions + sample data for alpha seeking

***HOMEWORK –(1) prepare for 30 minute quiz start regarding FPGA’s. (2) -complete another FPGA based architecture per app requirements Ted will specify***

**Session 3 – Tue June 13**

**Quizz then complete Big Data – ML/AI**

**QUIZZ (30 minutes)**

\*\*\* we will immediately review quiz over next 30 minutes

Remaining 2 hours of class:

Complete ML/AI with a PPT with hands on Neural Networks Demo. Ted will provide this PPT 1st class May 30.

***HOMEWORK –complete visio (or white board drawing / picture -save time) integrating alpha seeking opportunities to a ULL trading architecture – full details in class***

**Session 4 – Tue June 20**

**Linux Performance Tuning**

**Deep Dive into Red Hat Linux 7.2 low latency configuration & tuning, kernel bypass, PTP & NTP, then more details regarding ULL architectures from Class 1**

*I will present/explain following best practices regarding Linux tuning in a way that will lead to some white boarding designs.*

* Deep dive into Linux 7.2 network-latency configuration
  + Base config includes (perf over power saving):
    - Tcp\_fastopen=3 (2 way handshake – encryption of cookie of client @ init, so reconnect is 2 way, using the cookie)
    - Enable Intel\_pstat & min\_perf\_pct =100 (Ghz steady; disable fluctuations)
    - Disable THP (Transparent Huge Pages of 2 MB under K control)
    - Cpu\_dma\_latency
      * @ c\_states, keep cores from sleeping; part of QoS
    - Busy\_read 50 uSec (100 uSec for large# pkts) & busy\_poll 50 uSec (skt poll recvQ of NIC, disable net interrupt); cores “active”
      * BUT --- K bypass much better **(discuss 3 methods of K bypass)**
      * **Add pictures of K bypass**
    - Numa\_balance 0 (no auto NUMA mgt)
  + Disable unnecessary daemons and services (ex firewalld & iptables)
  + **Disable auto NUMA balancing (decrease interrupts**)
  + Max # ring buffer size
    - Dev driver drains buf via soft IRQ (other tasks not interr vs hard interr)
  + Set RFS (Recv Flow Steering)- increase CPU cache hits,forwards pkts to consuming app
  + TCP SACK- retrains only missed bytes)- tcp\_sack+1
  + TCP Window scaling – up to 1 GB
  + Sysctl –w net.ipv4.tcp\_low\_latency=1
  + Timing and scheduling:
    - Sched\_latency\_ns (20 ms default; increase!!)
    - Sched\_min\_granularity (4 ms default; increase!)
      * Increase # procs, threads – formula may lower this 4 ms
    - Some applications may benefit from tickles kernel
      * (ex: small # procs, threads at no more than # cores)
    - Sched\_migration\_ns (default500 uS; increase!)
      * This pertains to period of “hot” cache, prevents pre task migration
* Basic Linux and Server measures and utilities for performance analytics:
  + BIOS updates and tuning
  + Turbostat - reports on processor topology, frequency, idle power-state statistics, temperature, and power usage on Intel® 64 processors. Also - identify the rate of system management interrupts (SMIs)
  + Lstopo – system topology (cores and mem)
  + Lscpu - from sysfs and /proc/cpuinfo; CPUs, threads, cores, sockets, NUMA nodes; CPU caches and cache sharing, family, model, bogoMIPS - calibrate an internal busy-loop- "the number of million times per second a processor can do absolutely nothing".; byte order, stepping.
  + Numactl- Control NUMA policy for processes or shared memory; EX: numactl --physcpubind=+0-4,8-12 myapplic arguments Run myapplic on cpus 0-4 and 8-12 of the current cpuset. To check:

**show, -s** Show NUMA policy settings of the current process.

* + Numastat - Optimal performance is indicated by high **numa\_hit** values and low **numa\_miss** values
  + Tuned- daemon that uses **udev** to monitor connected devices and statically and dynamically tunes system settings according to a selected profile.
  + **Tuned-admin network-latency configuration (set profile**)
  + Isolcpus – isolate CPU cores from kernel scheduler
  + Interrupt affinity or isolation
  + Irqbalance- distribute hardware interrupts across processors on a multiprocessor system
  + **Check gamut of process (pid) info, much pertaining to performance in /proc/<pid>; for ex: files numa\_maps, stat, syscall**
  + Tuna – control processor and scheduler affinity
    - **Options: Isolate sockets from user space, push to socket 0**
  + VTune Amplifier 2016
    - CPU, GPU, threads, BW, cach, locks, spinTm, FxCalls, serial+Par Tm,
    - ID code section ID for parallelization; ex: TBB – more control over OpenMP
    - MPI analysis ex locks, MCDRAM analysis
  + Intel’s PCM (Performance Counter Monitor) – major enhancements
    - Ex: times specific threads hit/miss L1-2-3 caches and measures cache times and impacts of misses; helps ID priority procs, threads for cache
  + Tx Profilers – Wily, VisualJVM, BeaWLS, valgrind, custom FREE– T/S, ESP correl, ML
  + Perf: perf top –g (functions)
    - **Perf counters in hardware (cpu) , with kernel trace points (ex: cache miss, cpu-migration, softirq’s)**
  + **Strace** -runs the specified *command* until it exits. It intercepts and records the system calls which are called by a process and the signals which are received by a process.
  + Ftrace- uses the frysk engine to trace systemcalls
    - sycalls of procs and threads
    - Dynamic kernel fx trace, including latencies (ex: how long proc wakes/starts
    - /debug/tracing
    - Trace\_clock
  + Dtrace for Linux:
    - Dynamic – cpu, fs, net resources by active procs, can be quite specific
    - Log of args /fx
    - Procs accessing specific files
    - # New processes with arguments
    - dtrace -n 'proc:::exec-success { trace(curpsinfo->pr\_psargs); }'
    - # Pages paged in by process
    - dtrace -n 'vminfo:::pgpgin { @pg[execname] = sum(arg0); }'
    - # Syscall count by process
    - dtrace -n 'syscall:::entry { @num[pid,execname] = count(); }' ….. specific syscall ct per process or thread
    - **also ‘canned’ scripts for processes with top tcp and udp traffic, ranking of processes by bandwidth**
  + SystemTap – ex:probe tcp.setsockopt.return
    - **Uses strace points for kernel and user probes**
    - **Script thief.stp – interrupts by procs histogram**
  + dynamically [instrumenting](https://en.wikipedia.org/wiki/Instrumentation_%28computer_programming%29) running production [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel)-based [operating systems](https://en.wikipedia.org/wiki/Operating_system). System administrators can use SystemTap to extract, filter and summarize data in order to enable diagnosis of complex performance or functional problems.
  + **SysDig Tool – only syscalls, dump for post processing scripting**

* Oprofile uses hw counters, tracks mem access and L2 cache, hw interrupts
  + Mpstat, vmstat, iostat, nicstat, free, top, netstat, ss **[filter/script for analytics]**
* VM (Virtual Memory) and page flushes, optimize market data caches
* KERNEL- SW in mem, tells CPUs what to do; interface between HW & procs, MMU & VM
  + Slab allocation= mem mgt for k objects, eliminates frag
* Slow network connections and packet drops
* Intro to NetPerf tool
* NIC tuning
* Kernel bypass, LDMA, RDMA
* Kernel bypass with NIC vendors (SolarFlare, Mellanox, ExaBlaze,) – description how each work
  + SolarFlare OpenOnLoad sets up all socket calls in user space instead of kernel space, with dedicated socket connection & data handled in NIC memory
    - Without DirectTCP or EF\_VI tune config files:
    - EF\_STACK\_PER\_THREAD 1
    - EF\_STACK\_LOCK\_BUZZ 1 – threads spin this long
    - Set NUMA\_BALANCE 0 (no NUMA auto Mgt, no interrupts)
    - Disable irqbalance
    - Run sysjitter to test your infrastructure and user space
  + Mellanox VMA linked library to user space, also sets up user space calls to NIC; Connect-IB NIC allows non-contiguous memory transfers for app-app; RV offload – speeds up MC; MLNX OFED open fabric verbs for IB and Ethernet; PCIe switch & NVMe over Fabric; MPI offloads; 2 ports at 100 Gbps; IB & Ethernet connections < 600 ns latency
  + Mellanox – switches & NIC’s
  + **SN2000 series Ethernet switches (with Cumulus Linux)**
  + 32 100 GigE ports + 4 adapters
  + 300 ns latencies, zero frame loss, very deterministic (touting their own Asics, not Broadcom)
  + 16 MB buffer – microburst absorption
  + Congestion Monitoring Event – awareness of location, depth, duration of congestion, works with …. (next)
  + Cumulus Linux – enables apps to register fro & monitor depth (and slow down)
  + For MultiCast - Drops at slow receiver egress port, allowing fast servers part of MC group to keep running
  + **ConnectX-5 EN NIC (RDMA enabled NIC)**
  + 2 – 100 GigE ports
  + With kernel bypass – sub-600 ns latencies – very deterministic per burst buffer offload and other acceleration features, (may be on par with SolarFlare, or faster now)
  + Burst Buffer offload (no CPU interference)
  + Embedded PCIe switch – allows one to build “Machine Learning” appliance
  + NVMe over Faafric (NVMf) offloads – speed up storage IO
  + MultiHost technology – multiple hosts can connect into single adapter, separating PCIe interface into multiple interfaces (so no need ToR switch)
  + High speed SDN with Open V-Switch (OVS) data plane in NIC (not CPY hogging hyperVisor)
  + Enyx NICs: differs from SF and MX network stack in user space (can be CPU intensive)
    - Enyx places full TCP stack in hardware (FPGA); reduce jitter
* Network appliances:
  + ExaBlaze Fusion
  + Metamako MetaApp
  + FixNetics ZeroLatency
* Precision Timing – PTP and NTP
* PTP Symmetricom Sync Server S300s – NTP & PTP, owned by Microsemi GM
* GPS Satellite satisfies UTC Req.
* MIFIF II and PTP (software (sw) + hardware(hw) critical for accuracy; Req: 100 uSec + UTC
  + Symmetricom PTP GM 6 ports +/- 4 ns, <25ns to UTC
  + GPS -> GM-Spectracom->B-Clock(Arista7150s-FPGA timing+NAT)->servers-PTP-sw with FPGA based NIC’s ex: Exablaze ExaNIC models) – or SolarFlare NIC’s with HW timestamps
    - Linuxptp - ptp4I & phc2sys (can act as B-Clk) sync PTP hw clock on client, including VLAN tagged interfaces and bonded interfaces to master (GM) but with kernel; Dmons can’t consume MC; K delivers pkt to bonded interface SF’s sfptpd does all in HW; can sync every SF adapter; ptpd – mult platforms but just sw.
      * Timemaster – on start, reads NTP & PTP time servers, starts daemons, can sync sys clock to all time servers in multiple PTP domains
  + Master-slave time sync (ex:
  + PTP Timing within 6 ns –
  + consider disable tickless kernel : nohz=off (for accuracy) BUT test this and app impact
  + PTP in hardware best but costs; do ROI
  + If multiple interfaces in diff networks, set reverse FWD mode to loose mode
  + Cmd: ethtool –T <int> -verify timestamp ---- (for hw)
  + “timemaster” reads config of PTP time source
  + Cmd: systemctl start timemaster
  + ExaNIC FPGA can be programmed for extra analytics; some base programs available
  + MC if sync msg from Master but UDP unicast delay msg from slave to Master
  + PTP assumptions:
    - Network path symmetry (hence switch, router, FW, OS impact this)
    - Master and slave accurately measure when at pt of send/receive
    - Every hop can reduce PTP accuracy
  + PTP options:
    - Each slave clock direct cables to master .. but complexity. Cost …
    - Dedicate PTP switch infrastructure; switch PTP aware & eliminate switch delay or act as PTP M B-Clk; do not mix traffic
    - In dedicated LAN, PTP thru switch L2 Bcast to PTP bridge (server as B-Clk & bonded interface mgr), sends MC to FW (--if no SF; FW has list MC groups, IGMPv3 config ), MC to PTP clients for Time Sync, best if clients have with SF <add PICTURE>
      * FW configured for IGMP3, has necessary config allowing PTP-Bridge & clients to join std PTP MC group 224.0.1.129
      * Sfptpd can work on bonded interfaces so PTP clients need specify mgt interface to get PTP TS (from PTP bridge)
    - Hardware time stamps at every point
  + More PTP details:
    - Slaves periodically send messages back to Master (sync)
    - Sfptpd 🡪 file or syslog; ptp4l 🡪 stdout
    - Offset: amt Slave Clk off from Master
    - Freq Adjustment: how much clock oscillator adjusts to run at same rate as Mstr
    - Path Delay: how long to Slv & VV
    - Metrics – collectd, applies RegEx
  + NTP – selects accurate Time servers from multiple (ex 3); polls 3 or more servers
    - Keep stratum levels to no more than 2
    - Keep 3 clock sources near for sync
    - Use switches with light of no queuing
    - Use “timekeeper” – transforms any server into a timing appliance
* **Class Exercise (we will do together in class)–Explain different approaches to kernel bypass of following: ExaBlaze, SolarFlare, Mellanox, Enyx). Explain strengths and advantages of each; advise what specific electronic trading applications would best benefit from each.**
  + **Explain how following Linux Tuning options will will impact latencies**
    - **Swappiness =0;**
    - **Dirty-ratio 10;**
    - **Background- ratio 10**
    - **NIC interrupt coalescing (pre kernel-bypass)**
    - **Ring buffer increase**
    - **UCP receive buffer at 32 MB**
    - **Netdev-backup 1000000 (traf stored before TCP/IP proc; 1/core)**
  + **Explain what following commands produce for latency analysis:**
    - **ifconfig command**
    - **Netstat –s (send/recv Q’s)**
    - **Ss utility**
  + **Detail major benefits of VTune and DTrace and when you would use either**

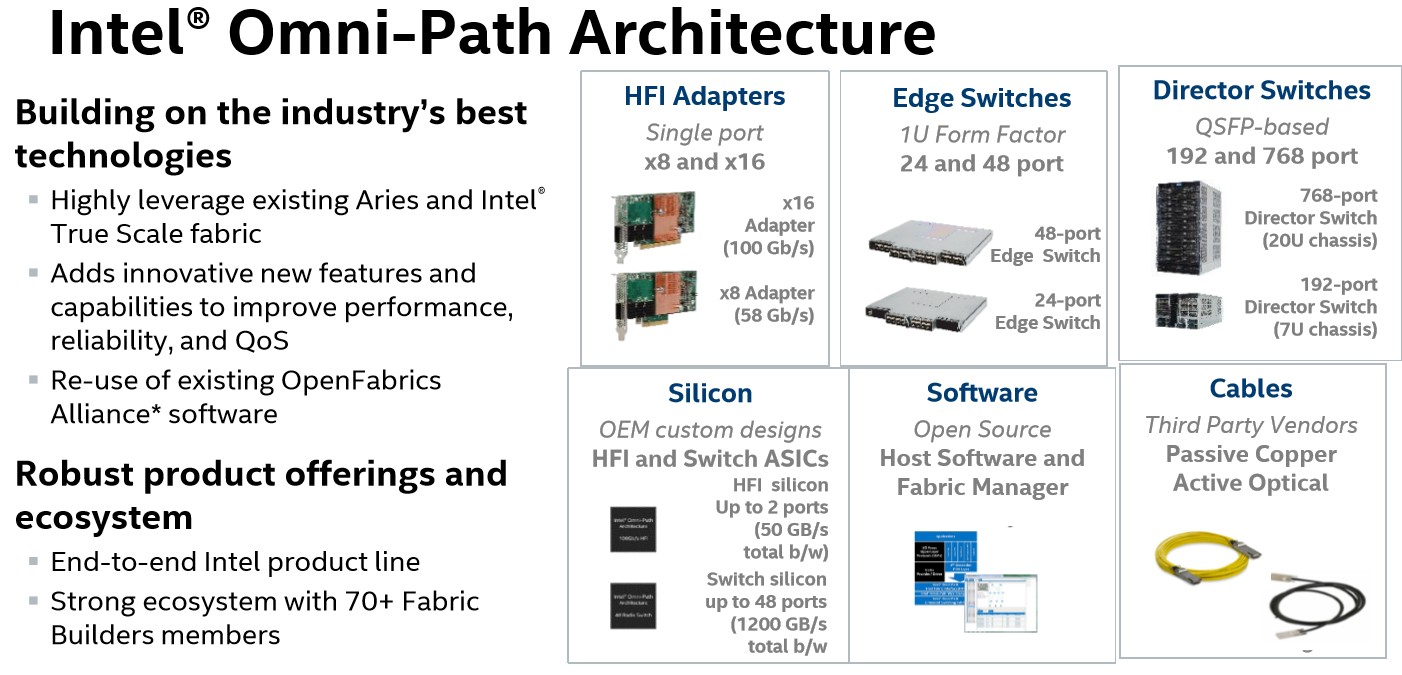
***HOMEWORK – complete prior week reading assignments, prepare for 30 minute quiz next class regarding:***

* ***Optimal Linux kernel tuning per application requirements (Red Hat Doc + class notes)***
* ***Benefits of FPGA’s and GPU’s (Text book + Algo-Logic doc)***
* ***How multi layer switches work (Metamako Doc)***
* ***Differences in tuning to Speed 1 (raw) vs Meta-Speed (Corvil & Tabb Doc)***
* ***\*\*\*\* IN CLASS -- I will spend 15-20 minutes detailing what is most important from the above.***
* ***ALSO – will present another sub optimal Linux configuration and ask you to correct / justify (should take approx. 30 minutes) – submit online before next week’s class***

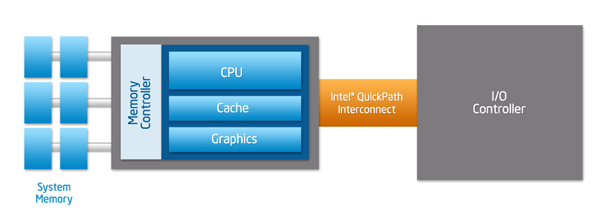
**Session 5 – Tue June 27**

**Latest ULL Intel Technologies, other server, memory, Flash devices, ULL messaging architectures, network protocols, SDN, Multi-Cast architectures + Market Data**

* Programming with Multiple core multi thread, parallelism
* Vectorize application code
* Design – Internal loops with deep vector instructions, outer loops with parallelization (threads)
* Servers, sockets, cores, caches. MCDRAM (Intel Phi)
* Core speeds GHz vs more cores, larger and faster caches
* Over clocked servers - features and what applications can benefit
* Linux, Solaris, Windows, other ex SmartOS, Mesosphere DC OS
* How to benchmark performance, analyze, tune
* NUMA aware processes and threads
* Optimize cache assignments per high priority threads
* Intel technologies including ...
* AVX-512 deep vector instructions (speeds up FP ops)
  + 6-8 registers; more ops/instruction; less power
* TBB thread Building blocks (limit oversubscription of threads)
  + OpenMP- explosion of threads
* Omni-Path high speed / bandwidth interconnect (no HBA, fabric QoS, MTU to 10K, OFA verbs,105 ns thru switch ports, 50 GB/s bi )
  + <http://www.intel.com/content/www/us/en/lustre/omni-path-fabric-sc16-presentation.html?wapkw=omni+path>
  + OFED RDMA verbs



* QuickPath: mult pairs serial links 25.6 GB/s (prior to Omni-Path)
  + Maintain communications even with link failure
  + Mem controllers integrated with microprocessors
  + Replaced legacy bus technology
  + Cache coherent
  + Can connect 1 Socket core with another Socket’s FPGA
  + <http://www.intel.com/content/www/us/en/io/quickpath-technology/quickpath-technology-general.html>



* **Shared Memory is faster than Mem Maps; allows multiple procs read/write into shared mem among the procs – without OS read/write commands. Procs just access the part of shared mem of interest.**
  + **Discuss ex of server proc sending HTML file to client; file is passed to mem then net function copies mem to OS mem; client calls OS function which copies to its own mem; contrast with Shared Mem**
* PCIE
* C ++ vs Java for ULL
* Lists vs vectors
* Iterate lists
* Role of FPGA, GPU, MicroWave networks for ULL
* C/C++, Java, Python, CUDA, FPGA - OpenCL: programming design considerations
* Java 8 new streams API and lambda expressions - for analytics
* **Class Ex** **– Explain how Quick Path & Omni Path both improve latencies and advise which is preferred for ULL and why**
* New age networks - Spine leaf to single tier
* RTE central to routers – lists routes to particular network destinations + metrics + immediate neighbor topology info, per specific routing protocols
  + Network ID (destination subnt)
  + Cost/metric
  + Next hop
* IPsec – authenticates & encrypts each IP packet
  + Negotiates cryptographic keys to be used per session
  + SSL is for transactions
* SDN (Software Defined Networks)
  + Cisco ACI + Tetration
  + Cloudistics
  + Plexxi
  + NSX
* Pico - ULL SDN vendor
* Options-IT – colo managed ULL infrastructure
* Cisco and Arista switches for ULL
* Cisco ACI and Cisco Tetration – Deep machine Learning to automatically optimize large networks
* Switches with deep buffers, great for Big Data Analytics
* Configure Routers for ULL - LLDP, MLAG, VRRP, VARP (active-active L3 gateway)
* Network protocols - BGP, OSPF, HSRP
* Arista 7124FX with  EOS
* Arista 7280E – 100 GigE ToR, VXLAN, LANZ, 9 GB deep buffers
  + Virtual output Q eliminates HOL blocking
* Plexxi switches - a disruptive technology – single tier
* Plexxi optimal bandwidth via its SDN
* Optimal VLANs configuration for analytics
  + Use trunks from 1 switch to another switch after defining a VLAN, or use router
* VPLS (Virtual Private LAN Service) also for analytics
  + Enet based multiPt-multiPt over IP or MPLS
* Decrease network hops for speed
  + (ex: Slim-Fly: Low diameter network architecture if not ready for Single Tier)
* Network protocols:
  + EBGP: external, distance vector, via paths, network policies, rule sets, finite state machine; BGP peering of AS-AS
  + BGP-MP: multi protocol + IPv6, unicast & MC; use for MPLS-VPN
  + OSPF: interior within AS, link state routing with metrics of RTT, amt data thru specific links, and link reliability
  + MOSPF: uses group membership info from IGMP + OSPF DB, builds MC trees
  + EIGRP: OSPF + more criteria: latencies, effective BW, delays, MTU
  + MPLS: between network nodes, short path lables – avoid complex lookups in RTE table; multi protocol- includes ATM, Frame Relay, DSL
  + IB: hw, light wt, no pkt reordering; link level flow control, loss less, QoS virtual lanes 0-14, RDMA verbs (adds latency), UFM tool, 4000 byte MTU (adds latency as this MTU must fill before transmission)
  + OPA: Intel’s Omni Path Architecture: 100 Gbps, new 48 ports switch silicon, silicon photonics, no HBA, 50% decrease infra vs IB, 100-110 ns / port, congestion control – reroutes traffic, MTU up to 10K
  + FC: FiberChannel – optical pkts, units of 4 10-bit codes, 4 codes=TransWord; meta-data sets up link and Seq; tools-Agilent, CATC, Finisar,Xyratex; FC: similar to Enetpkt ex: mult frames assembled with src, dest
  + IGMP: used by [hosts](https://en.wikipedia.org/wiki/Host_%28network%29) and adjacent [routers](https://en.wikipedia.org/wiki/Router_%28computing%29) on [IPv4 networks](https://en.wikipedia.org/wiki/IPv4_network) to establish multicast group memberships

**Market Data**

* **Ted’s MC hand-out**
* Multicast (MC) performance considerations
  + Turn on IGMP Snooping **on Switch**
    - Switch listens to IGMP conversations between hosts/routers; maps links that require MC streams; Routers periodically query; 1 member per MC group per subnet reports.
  + Clients issue IGMP join requests to MC groups
  + Routers solicit group member requests from direct connect hosts
  + PIM-SM (Sparse Mode …low % MC) requires a Rendezvous Point (RP) router
  + Routers in PIM domain provide mappings to RP (exchange info for other routers)
    - PIM domain: enable PIM on each router
    - Enable PIM sparse mode on each interface
  + After RP, forward to receivers down shared distribution tree
  + When receiver’s 1st hop router learns source, it sends join message directly to source
  + [Protocol Independent Multicast](https://en.wikipedia.org/wiki/Protocol_Independent_Multicast) (PIM) is used between the local and remote MC routers, to direct MC traffic from the MC server to many MC clients.
* Message based appliances, including FPGA based
* Direct feed normalization
* Conflation to conserve bandwidth
* NBBO
* Levels 1 and 2 market data
* Depth of book builds (in FPGA’s or new multi core servers)
* Smart order routers
* **Doc regarding leading vendor solutions**
* Exablaze NICs and switches v Metamako switches for market data
* ENYX FPGA NICs and Appliances for market data and order flow
  + FPGA’s now come equipped with own co-processor (ARM core) which can be programmed
* Nova Sparks FPGA based market data ticker
* Fixnetics Zero latency - multi thread risk checks in FPGA and order processing in parallel on a core(a)
  + 40 different risk checks
* Other products -- Exegy, Algo logic, Redline, SR labs
* Consolidated feed vendors Bloomberg and Thomson Reuters
* Use of new Intel Technologies (hardware & software) for alpha seeking strategies
* **Class Ex – (1) white board sessions where students will design ULL market data and multi cast architectures, per specific business/application criteria. (2) Given a Visio of a large network but with only a few MC groups and subscribers, identify the likely path(s) to sources few. Include choice of router as RP.**

HW – Mkt Data white paper.

* **Quick intro Python** - reference -- Ultimate Algorithmic Trading Systems ToolBox, George Pruitt, Wiley, 2016
* Python algo trading examples
* Intro to Wireshark
* Intro to FIX Protocol
* Intro to Wireshark with FIX protocol “Plug-in”
* TCP, UDP, multicast (MC), then analysis via WireShark, Corvil
* **OPTIONAL**
* Next Gen Firewalls (ex: Fortinet)
  + 1 platform end-2-end, with multiple security related aspects including anti-virus, malware, intrusion detection, database and OS access controls, web filtering, web app security, user ID awareness, standard rules access, internal segmentation (into functional security zones, limits spread of malware & mischief, identifies mischief & quarantines infected devices; shares all info via its fabric to whole network; Zero Trust Policy – places the FW in network center, in front of data
  + Empow – Security Orchestration product
* Kerberos
  + Client/server network authentication protocol via secret key cryptography, stronger than traditional firewalls as they focus on external threats whereas Kerberos focuses on internal
  + Each KDC has copy of Kerberos DB; Master KDC has copy of realm DB, which is replicated to slave KDC’s @ regular intervals; DB password changes are in the Master; slaves grant Kerberos ticket servers / services, time series critical & create ACLs too; Kerberos daemons are started on the Master, assigns hostnames to Kerberos realms, ports, slaves
  + Opportunity for Docker container security:
    - Kerberos for access to multiple levels of container types (ex: checking account KYC vs withdrawal .. acct mgr vs authenticated client)
* IPTables – may opt to disable for ULL, rely on external FW
  + Set up, inspect tables of IP packet filter rules; each table: built-in “chains” & user defined chains; chains list rules to match set of packets
  + Required for server “routers” NAT aware
    - Rte intercepts, determines NAT@

**DNS** - DNS server, or name server, is used to resolve an IP address to a hostname or vice versa:  
master DNS server for your domain(s), which stores authoritative records for your domain.

* A slave DNS server, which relies on a master DNS server for data.
* A caching-only DNS server, which stores recent requests like a proxy server. It otherwise refers to other DNS servers.
* A forwarding-only DNS server, which refers all requests to other DNS servers.

HP OpenView:

HP OpenView software allows you to manage your IT infrastructure components in an ordered, standardised manner, defining rules, actions and alerting characteristics on faults or potential issues in your IT environment.  It is primarily used for monitoring servers, devices, networks, databases & applications to ensure faults are detected and alerted upon in a timely manner.  Proactive monitoring is also possible to ensure alerts are received before a fault occurs, thereby providing time to fix a potential problem before it escalates and affects the business

ITRS Geneous

SolarWinds

detect, diagnose, and resolve network performance issues

F5 Big-IP LB

* blend of software and hardware that’s a load balancer and a full proxy. It gives you the ability to control the traffic that passes through your network.
* smart evolution of Application Delivery Controller (ADC) technology. Solutions built on this platform are load balancers. And they’re full proxies that give visibility into, and the power to control—inspect and encrypt or decrypt—all the traffic that passes through your network
* smart evolution of Application Delivery Controller (ADC) technology. Solutions built on this platform are load balancers. And they’re full proxies that give visibility into, and the power to control—inspect and encrypt or decrypt—all the traffic that passes through your network.
* iRules – app acceleration

MPI for parallel processing

portable message-passing programs in [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B),

* **END- OPTIONAL**
* **Class Exercise – Determine whether Single Tier Networks improve ULL versus Spine Leaf. If so explain why. Several scenarios will be presented and students will architect networks on white boards.**

***HOMEWORK –Read Python algo trading program, examine code***

**Session 6 – Thu July 6 (1 ½ hour class, ON-LINE)**

* Special lab session: Python for algo trading + extra wireshark training,

**Session 7 – Tue July 11**

**Complete / Review topics from last week; start with**

Deep Dive into the Python Code – Bollinger Bands

***HOMEWORK –2 weeks to complete; Ted will provide extra data to feed the Python Bollinger Band algo program, ask for modifications (enhancements), ask to run then analyze new output.***

**Session 8 – Tue July 18**

**Address outstanding Python / Algo questions/issues; then cover high speed messaging & middleware, infrastructure ROI, and cloud technologies for ULL, + application specific details regarding ULL apps**

Middleware, High Speed Messaging

* 60 West AMPS
  + Pub/Sub engine, exploits parallelism, multi socket/core/10GigE
  + Speeds up message processing through entire message life cycle
  + Can combine multiple formats in single payload – FIX, JSON,XML, Google Proto-Buf, and unparsed binary message types
  + Built in CEP
* 29 East LBM (UME)
  + Daemon less
  + Topics, sources, destinations with timer tuning of NAKs and retransmissions
  + % BW allocated to retransmissions
  + Config subscribers
  + most modern networking hardware is capable of passing wire-speed multicast with appropriate configuration parameters. The key is discovering configuration problems early and diagnosing them. The "msend", "mdump", and "mpong" tools can help. You can use them to evaluate your network's capacity to carry multicast traffic.
  + RAI
  + Mtools
* New FIX Binary protocol in beta promises to lower latencies
* Importance of High Speed messaging for Algo Trading
* Intro to basic algo’s for trading equities (ex VWAP, Volume Participation, use of AVX and RSI)
* How to back-test algo’s for trading
* How to conduct ROI for new ULL architectures
* Why traditional cloud architectures fall short for ULL
* Cloud for analytics - pitfalls vs best practices
* Micro services potential
* **Class Ex – output from application logs will be provided. Students will analyze and determine how AMPS can be configured for both high speed middleware and event stream processing for analytics**

End-End ULL Architectures

* Co-Lo with 500 ns order ack times (revisited with our new knowledge)
* Dark pools
* Algo Trading (servers, appliances, or FPGA’s) in the architecture
* Smart Order routers
* Prop Trading
* Exchanges
* **Class Ex – VISIO or white boarding of a new trading system TBD, applying all learned in course**

***HOMEWORK –1 more weeks to complete; Last week, Ted provided extra data to feed the Python Bollinger Band algo program, asked for modifications (enhancements), asked to run then analyze new output.***

**Session 9 – Tue July 25**

**Review Python assignment then cover future ULL architectures ; then cover ULL ET FUTURES**

Futures, including Cloud Architectures for ULL

* Ted’s A-Team strategic projections for next 2 years
* Projections on new technologies’ impacts on ULL – may include:
  + new Intel cores & software,
  + adoption of Single Tier networks,
  + impact of in memory machine learning for alpha generation of trading signals,
  + integration of deep machine learning from cloud to live trading networks via high speed interconnects, to an asynchronous Q, with NO latency impact,
  + applicability of block chains,
  + system reliability engineering (SRE),

< open slot to catch up on past material + student’s questions pertaining to future of ULL for ET>

**REFERENCES for more info**

<https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Performance_Tuning_Guide/sect-Red_Hat_Enterprise_Linux-Performance_Tuning_Guide-Performance_Monitoring_Tools-turbostat.html>

<https://linux.die.net/man/1/hwloc-bind> re lstopo command

<http://www.linuxtopia.org/online_books/linux_kernel/kernel_configuration/re46.html>

<http://www.computernetworkingnotes.com/network-administrations/dns-server.html>

<http://xcelorgroup.com/>

APPENDIX (more will be added over next few weeks)

**Market Data**

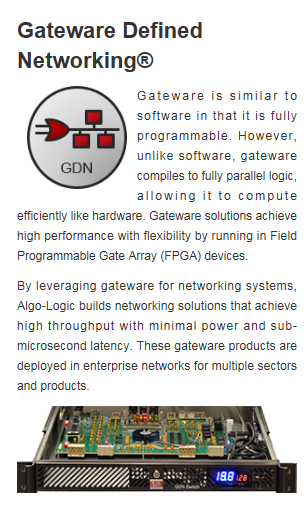
* **Ted’s MC hand-out**
* Multicast (MC) performance considerations
  + Turn on IGMP Snooping **on Switch**
    - Switch listens to IGMP conversations between hosts/routers; maps links that require MC streams; Routers periodically query; 1 member per MC group per subnet reports.
  + Clients issue IGMP join requests to MC groups
  + Routers solicit group member requests from direct connect hosts
  + PIM-SM (Sparse Mode …low % MC) requires a Rendezvous Point (RP) router
  + Routers in PIM domain provide mappings to RP (exchange info for other routers)
    - PIM domain: enable PIM on each router
    - Enable PIM sparse mode on each interface
  + After RP, forward to receivers down shared distribution tree
  + When receiver’s 1st hop router learns source, it sends join message directly to source
  + [Protocol Independent Multicast](https://en.wikipedia.org/wiki/Protocol_Independent_Multicast) (PIM) is used between the local and remote MC routers, to direct MC traffic from the MC server to many MC clients.
* Message based appliances, including FPGA based
* Direct feed normalization
* Conflation to conserve bandwidth
* NBBO
* Levels 1 and 2 market data
* Depth of book builds (in FPGA’s or new multi core servers)
* Smart order routers
* **Doc regarding leading vendor solutions**
* Exablaze NICs and switches v Metamako switches for market data
* ENYX FPGA NICs and Appliances for market data and order flow
  + FPGA’s now come equipped with own co-processor (ARM core) which can be programmed
* Nova Sparks FPGA based market data ticker
* Fixnetics Zero latency - multi thread risk checks in FPGA and order processing in parallel on a core(a)
  + 40 different risk checks
* Other products -- Exegy, Algo logic, Redline, SR labs
* Consolidated feed vendors Bloomberg and Thomson Reuters
* Use of new Intel Technologies (hardware & software) for alpha seeking strategies
* **Class Ex – (1) white board sessions where students will design ULL market data and multi cast architectures, per specific business/application criteria. (2) Given a Visio of a large network but with only a few MC groups and subscribers, identify the likely path(s) to sources few. Include choice of router as RP.**

HW – Mkt Data white paper.

Week 4 will have a visio assignments:

***HOMEWORK – 2 Visio designs – 1 for a 1 uSec T2T, 2nd for more modest (includes internal alpha seeking) 10 uSec T2T***

ALGO-LOGIC



FPGAs are perfectly suited to performing highly repetitive tasks on vast amounts of data.

#### AI/Machine learning

Just as the human brain processes vast amounts of data simultaneously, the same must be possible for neural networks and machine learning algorithms. Unavoidable overhead associated with modern CPU's mean hardware acceleration is essential in these fields. With Reconfigure.io you can use the popular Go language to program and reprogram cloud-based FPGAs to meet this need.

state-of-the-art compiler renders Go into FPGA-compatible hardware descriptions. Everyone, from single-person startups to established companies, can now put FPGA's to work.

FPGAs provide significant speed enhancements (10–100x) over the same code running on traditional CPUs. While FPGAs run at a slow speed relative to modern CPUs, they complete many, many tasks at the same time resulting in a dramatic overall speed increase.

* Goroutines are functions that can run concurrently with other functions.
* Channels are pipelines through which Goroutines can communicate and synchronize their operation.
* Select statements help organize the process of switching between channels, and give programmers the ability to control when parallel operations can run.
* Reconfigure.io then seamlessly transforms your Go code to run on the parallel architecture of the FPGA.

MiFID II Tx Reporting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | **MiFID II: Time to Tackle Transaction Reporting in Detail**  By Steve Barnes, VP of Technology, AQMetrics  Following the recent changes to mandatory transaction reporting (TR) being introduced under MiFID II, there are now a number of vital new reporting obligations which the markets must rapidly become familiar with. Yet as was revealed by a recent webinar I took part in on TR data and data management challenges, it was clear that the industry is hungry for additional clarity and detail around the specific changes to these reporting requirements.   **Scale of the challenge**   A particularly concerning point for firms, which I covered in the webinar, involves the new set of reporting fields and the potential difficulties involved in extracting this data from their existing systems. For example, general fields will require extra steps in an ETL process to continue to ensure the quality of the data. There is also an emphasis on the mandatory use of Legal Entity Identifier (LEI) codes. In the case of the buyer and seller fields, all legal entities will need to be referenced against a global LEI database before submitting to the regulator - or the report will fail. This requires extra due diligence to protect the data quality.   In the case of the buyer consisting of an actual person, details such as their first name, last name and date of birth may need to be obtained from the human resources department. A firm’s internal systems may never have had to interface with HR before and, depending on the size of the organisation, the data may also be in siloed systems. This might involve developing a data programme to get these systems working together. Accessing these systems may take a longer time than firms initially expect, which is especially worrying for firms that have not started the process yet.   **Additional considerations**   Furthermore, the addition of OTC derivatives has also further complicated matters in relation to data fields due to questions around how to classify them under the ISO data standards. The algorithm needs to differentiate between decision maker and executer, alongside additional difficulties on determining where to apply the pre-trade waiver to commodity derivatives. Sourcing the data can also be tricky if the systems have not interfaced before, not been updated for some time or if they are older. This will ultimately prove to be a key data reporting challenge for most firms.   Webinar participants were also interested in the potential impact of the incoming General Data Protection Regulation (GDPR), due to be enacted in 2018, and the wider data protection issues raised by the mandatory collection of data about individuals. In terms of details such as names and date of birth being required then they ought to be protected by existing data protection rules. In terms of third-party data through an Approved Reporting Mechanism (ARM), then firms will need to be in-line with GDPR policy as they are taking someone else’s data.   **Effort that pays**   Overall, data quality is key to most of these requirements. For example, AQMetrics uses programmatic regulatory control checks to ensure the data quality meets the required standards before submitting to the regulator. These checks are relatively simple for certain numerical fields ie dates, country codes etc but problems can arise when the data is unstructured ie names or passport details. The source system data may originally be of poor quality for historical reasons. As a result, performing a data quality exercise on the existing systems for data quality before it gets channelled through for transaction reporting is essential. Data control is especially key under MIFIR. Moving from manual systems, such as records held on excel spreadsheets, to more automated systems may involve a third-party who can implement better auditing and governance processes.   AQMetrics offers a full range of risk and compliance solutions for MiFID II transaction reporting and are now accelerating our offering by also registering as an ARM. Firms tend to find that once their TR is being correctly implemented, then the benefits for getting MiFID II right go further than just regulatory compliance. It is very useful for firms to have a single view of the data they are collecting, where is came from and knowing the limits of their systems. This data can then be used elsewhere in the organisation. But the main takeaway I hope practitioners gained from the webinar was simply to not leave updating their systems for the new requirements to the last minute. Many of the necessary changes may be much bigger and more expensive than initially expected, so it really does pays to start early.   The full webinar recording: “MiFID II: Data for transaction reporting” is available [here](http://aqmetrics.us12.list-manage2.com/track/click?u=c97175dfdcbdb102d11a5e5fb&id=dddb9db818&e=738a4dcbca).    Webinar link: [http://resource.datamanagementreview.com/webinar-recording-mifid-ii-mar2017](http://aqmetrics.us12.list-manage.com/track/click?u=c97175dfdcbdb102d11a5e5fb&id=005b705fac&e=738a4dcbca)    More information is available at [http://www.aqmetrics.com](http://aqmetrics.us12.list-manage.com/track/click?u=c97175dfdcbdb102d11a5e5fb&id=357c955fb3&e=738a4dcbca), or follow us on Twitter @AQMetrics. | | | | |

Exchange operator Nasdaq has announced the opening of a new equities platform targeting fintech firms and technologists.

The new platform, dubbed “Nasdaq Ventures,” is a global venture investing program focused on cultivating talent and technology advancement within financial services — spurring innovation that ensures Nasdaq’s technology and services are at the forefront of the industry.

Investments provided by Nasdaq Ventures will range from $1M to $10M and seed- to Series C funding. Each investment will work with and support our core business strategies and mission.

Initial focus will include: digital transfer/payment/transaction processing; machine and artificial intelligence; emerging and frontier marketplace; and data & analytics.

“As a financial technology leader, we are committed to the further development of global markets through innovative technology that advances our clients' ambitions,” Nasdaq said in a statement. “Because of this commitment, we launched Nasdaq Ventures in order to partner with revolutionary technologists who are shaping the fintech sector.”

The goal of Ventures is to partner with unique fintech companies worldwide – and the main objective is to identify and collaborate on new technologies and groundbreaking services and solutions. To do so, Nasdaq has partnered with three firms to launch the platform.

First is Chain, a technology company on a mission to enable a smarter and more connected financial system by digitizing the world’s assets. Based in San Francisco, they partner with leading organizations around the world to build, deploy, and operate blockchain networks that enable breakthrough financial products and services.

Chain was founded in 2014 and has raised over $40 million in funding from a variety of companies, including Nasdaq, since their inception. Chain is featured on Forbes Top 50 Fintech picks.

Chain's enterprise-grade Chain Core platform underpins the Nasdaq Linq offering, which is a digital ledger technology that leverages a blockchain to facilitate the issuance, cataloguing and recording of transfers of shares of privately-held companies on The NASDAQ Private Market.

The second company is Digital Reasoning, a leader in cognitive computing that enables firms across financial services, enterprise, health and government understand underlying context behind human communication and become more efficient in their activities. In January 2017, **Digital Reasoning** was recognized by CB insights amongst the most promising **AI** companies around the globe. Digital Reasoning is featured on Forbes Top 50 Fintech picks.

Digital Reasoning’s eComms monitoring for proactive compliance and Nasdaq’s industry leading and award-winning SMARTS Surveillance solutions will deliver a unified approach to trade surveillance by integrating trading alerts with natural language processing and machine intelligence-based technology all within a single interface.

And the third company, Hanweck, is a provider of real-time risk analytics on global derivatives markets focusing on the large-scale risk problems of banks, broker/dealers, hedge funds, central counterparties and exchanges — where the number of instruments and positions number in the millions.

Hanweck delivers its risk analytics as a real-time service — usually in the form of a data feed — thereby dramatically simplifying integration with its customers' risk architecture.

**London, New York, 26 April, 2017** – Vela Trading Technologies (Vela), a global leader in high performance trading and market data technology, has launched a fully-managed European Best Bid Offer (EBBO) solution in response to increased client demand to comply with the changing pre-trade transparency and best execution rules which will come into force with MiFID II in January 2018.

The EBBO solution aggregates market data from multiple primary exchanges and multilateral trading facilities (MTFs) across Europe to provide a consolidated view of liquidity for a given instrument. It calculates both top of book and market depth in real time and updates with each underlying tick. Clients have the flexibility to customise these venues based on their best execution policy.

Built on Vela’s SMDS ticker plant technology, the EBBO uses a single process for data retrieval, data normalisation, consolidated book construction, and distribution via either multicast or transmission control protocol (TCP). SMDS offers an industry-leading latency profile and a low hardware footprint whilst offering high performance and throughput. SMDS already powers an equivalent Vela product covering US venues to provide a faster alternative to the Securities Information Processor (SIP).

MICRO SERVICES

Amazon EC2 Container Service (ECS) is a highly scalable, high performance [container](https://aws.amazon.com/containers/) management service that supports [Docker](https://aws.amazon.com/docker/) containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances. Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure. With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes, and IAM roles. You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements. You can also integrate your own scheduler or third-party schedulers to meet business or application specific requirements.

Oracle Bare metal cloud

Robin – AWS or Azure?

That’s in contrast to approaches such as [VMware NSX](https://www.sdxcentral.com/products/nsx/), where virtualization is run by the hypervisor.

Ellison described the off-box element as “software that runs on our special network interface adapter card,” but Oracle isn’t providing any other detail.

During the Amazon Web Services Summit last week, AWS announced the general availability of the [FPGA-accelerated Amazon EC2 F1 instances](https://info.ryft.com/acton/ct/17117/s-003c-1704/Bct/l-0299/l-0299:58c6/ct1_0/1?sid=TV2%3AKT31fBbPA). Combining the powerful F1 with the easy-to-use Ryft Cloud technology gives organizations the unprecedented ability to quickly and easily extract critical insights from all their data.

Ryft Cloud enables users running Elasticsearch to get fast, actionable insight from their cloud-based data **72X faster** than is possible with traditional analytics infrastructures.

Capabilities provided by Ryft running on AWS include simple access to data search and analysis with Elasticsearch. With Ryft and the F1 instance, Elasticsearch users can eliminate data indexing, transformation and curation and extend Elasticsearch to:

* **Accelerate workflows** with the ability to deploy pre-index and post-index searches
* **Speed search and analysis** across unstructured data and JSON, XML, LOGs, CSV, TSV and other files with no transformation
* **Increase the power of edit distance** with user selectable changes to high distance requests for fuzzy Hamming or Levenshtein searches
* **Enhance wildcard searches** to include leading wildcard characters
* **Supercharge regular expression** and machine learning, coming later in 2017

EXABLAZE

**Sydney, April 27, 2017:** Ultra Low Latency networking specialist [Exablaze](https://exablaze.com/) has announced [FastMux](https://exablaze.com/exalink-fusion-mux), a major new addition to the ExaLINK Fusion product family. [FastMux](https://exablaze.com/exalink-fusion-mux) reduces aggregation latency to a minimum of 49ns (nanoseconds), making it about half the latency (or twice the speed) of the previous generation of multiplexing devices.

'This makes the [ExaLINK Fusion](https://exablaze.com/exalink-fusion) the fastest crosspoint-enabled network multiplexer on the market,’says [Exablaze](https://exablaze.com/) chairman Dr Greg Robinson. '[FastMux](https://exablaze.com/exalink-fusion-mux) allows 15 connections to be aggregated into a single upstream port, such as a long-haul link or an exchange order entry point, with unrivalled speed and flexibility. What’s more, the latency through the device is almost identical for all 15 ports.



The [ExaLINK Fusion FastMux](https://exablaze.com/exalink-fusion-mux) firmware upgrade ships with a [performance verification report](https://exablaze.com/downloads/pdf/FastMux_Verification.pdf) that shows the latency distribution of all ports as well as the configuration used for testing. The report was generated using Exablaze’s High Precision Timing network capture technology ([ExaNIC HPT](https://exablaze.com/exanic-x10-hpt)) which assures 0.25ns (250 picosecond) measurement resolution.

The [FastMux](https://exablaze.com/exalink-fusion-mux) [performance report](https://exablaze.com/downloads/pdf/FastMux_Verification.pdf) provides a best case (minimum of minimums) latency of 48.79ns, an average case (median of medians) latency of 53.79ns, and a maximum case (maximum of maximums) of 58.79ns. These latencies vary by less than 1.00ns between ports, regardless of the number of ports in use, so users can be confident of achieving this performance in real world deployments.

All 48 ports on the [ExaLINK Fusion](https://exablaze.com/exalink-fusion) (including [FastMux](https://exablaze.com/exalink-fusion-mux) ports) are **crosspoint-enabled**. This provides essential support for layer 1 mode features such as tapping for logging, patching for failover, and packet counters and signal quality statistics for monitoring. This feature makes the [ExaLINK Fusion FastMux](https://exablaze.com/exalink-fusion-mux) the most flexible aggregation device on the market.

<https://exablaze.com/media/blog-fastmux>

[**https://www.sps.nyu.edu/professional-pathways/topics/finance/asset-management-and-investment-strategies/FINA1-CE9515-ull-ultra-low-latency-architectures-for-electronic-trading.html#more-details**](https://www.sps.nyu.edu/professional-pathways/topics/finance/asset-management-and-investment-strategies/FINA1-CE9515-ull-ultra-low-latency-architectures-for-electronic-trading.html#more-details)