Random Notes

In the News Topics:

relationship between interest rates, fed reserve not changing interest rates bc of election, feds, banks, property, debt, japanese debt, depreciation of the yen, cpi, inflation, yield curves, Greece EU bailout, Taiwan semiconductors, new semiconductor factories in the US (not best ones), consolidated Audit trail, Social Security, Medicare, Influencing elections using crude oil (2022 midterms), linux backdoor, Democrats,

Japenese Yen devaluation:

Answer: The Japanese economy has been stagnant since the 1990s, following the 1985 Plaza Accord agreement where Japan and the US agreed to devalue the US dollar against other western currencies, including the Yen. This caused consumers in Japan to be able to import cheap American goods, increasing consumption and making the Japanese economy grow temporarily but suffer in the long-term, as domestic producers became less competitive to foreign importers and Japan's aging population decreased the size of the workforce.

The Japanese economy doesn't grow and the American one does, so the dollar is more in demand over time versus the yen. This shift is increased by the fact that the US is a net importer of foreign goods, meaning it exports a lot of dollars. The same doesn't necessarily apply to other net exporters, since they might export American dollars instead of their own currencies.

The central bank of Japan tries to lower interest rates and go into debt to stimulate consumer spending and therefore end the 3-decade recession, but it hasn't worked so far. The Japanese yen isn't demanded abroad and foreigners buy Japanese debt in USD, so printing money (which otherwise would help) is impossible, since nobody will use the new yen. Now interest rates can't go any lower and Japan is essentially out of monetary tools to deal with their fiscal problems.

Edit: Just to clarify, the Plaza Accords caused a shock that kickstarted Japan's 3-decade recession. It didn't cause it. Long-term economic factors like an aging population, no immigration, and a low birth rate did.

If theres inflation -> increase interest rates

NVIDIA, AI, CUDA

Lecture 02: Orders, Order Books, Trades, Fundamentals

The Life Blood of HFT:

- Orders
- Orderbooks
- Trades(fills, matches, etc)

Understanding the life cycle of orders(Definitions):

- Buy:
  - To exchange (usually) fiat currencies such as dollars, yuan, yen, etc) for a financial instrument(stock, bond, futures, crypto)
  - Orders to “buy” are also referred to as “bids”(like in auctions)
  - If a trader is “buying” an asset, they may also say they are “long” the asset
  - The reason traders buy long is to buy the asset at a lower price and then sell at a higher price
  - SalePrice - BuyPrice -trading fees = Profit(or loss)
- Sell:
  - The inverse of a buy orde;r typically exchanging some asset for currency
  - Orders to “sell” are also referred to as “offers’ (as in, “i am offering this for sale”)
  - If you are previously “long” an asset(meaning you own it), and wisht o “dispose” of it(sell), this is often referred to as “covering”
  - Selling(especially in large amounts) can often be referred to as “liquidating”
- Note:
  - RanIn some markets there is a distinction in the “types” of sell orders(sell vs short)

“Selling” vs “Shorting”

- If you are selling something you already own, this is plainly referred to as “selling” or a sale”

- “Shorting”: you acn also sell something you do not already own, and you do so buy borrowing someone else’s position(and paying them for a loan) and then selling it in the market
- Traders “short” assets as a means to profit from the price of something going down
  - BuyPrice - SellPrice - LoanCost - tradingFees = Shorting Profit(or loss)

Covering:

- Covering refers to either selling what you bought or buying back what you shorted
- It can also be called closing out on your positions
- “Covering” because you are eliminating future risk to the changes in the price of the asset
- If you no longer own or are short the asset, you are no longer at risk to changes in its price

Order:

- An order is a request by a trader expressing a desire to buy or sell some asset
- An order doesn’t mean that a trade is necessarily equal nor does it cause a trade to occur
- An order can also be referred to as a “quote”; as in calling up a company and askign them to “quote you the price’ for some good or service

Fundamental Order Operations:

- New Order: submit a new order to an exchange or other trading party
- Modify order: change some parameter(s) of a previously submitter order(modify size up or down, for example)
- Cancel Order: cancel a previously placed order

“Market” Order Definition(buy_or_sell, asset, size):

- Buy_or_sell: whether you are placing a “buy” or a “sell” order
- Asset: the specific asset(stock, bond future, option that the trader wishes to buy or sell
- Size: the size(number of “units” that the trader wants to trade of that asset
- This type of order is known as a market order
- It is a “market” order, because the trader is willing to pay whatever the “best” price available

Example Market order:

- Buy 100 shares of MSFT
  - MSFT: the asset that the trader wishes to purchase
  - 100: the number of units

- Possible responses form teh exchange:
    - FILLED! For 100 shares @ 398.90(random price)
    - Partial FILL: 20 shares @ $398.90; This implies your order for the other 80 shares is still waiting to be filled
    - Pending: order was accepted but has not been filled because there are no sell orders to match against
    - Rejected; for some reason you are not allowed to rtrade(market is closed, you do not have enough money, etc)

What is the Best Price?

- If you are buying, the “best” price (for you) is the lowest sell price or “offer”
    - You want to pay the least
- If you are selling, the “best” price for you is the highest buy price(or “bid”)
    - You want to be paid the most money in exchange for whatever you are selling
- These two motivations are opposite and contradictory:
    - Buyers want to pay the least while sellers want to be paid th emost

The entire point of HFT, automated trading, exchanges is:

- To match willing buyers and sellers
- To provide incoming buyers the lowest price from the “list” of available sell orders
- To provide incoming sellers the highest price from the “list” of available buy orders
- Traders especially HFT, constantly monitor the list of all buy and sell orders, evaluating whether to trade in response
    - This is the fundamental role of algorithmic and high-frequency (low latency) trading

“Order Book”

- An aggregated list (or sum) of all buy and sell orders at each price
- The order book in constantly changing

Limit Orders:

- Most orders are not market orders, they are limit orders
- Limit Order Definition:
    - Buy_or_sell: whether you are placing a “buy” or a “sell” order
    - Asset: what you want to buy or sell
    - Size: the maximum amount you want to buy or sell
    - Price: the Highest price you are willing to buy or the lowest price you are willing to sell for
- Example limit order:

- “Sell 100 shares of MSFT for at least $398.45”
  - Buy_or_sell: “SELL”
  - Asset: MSFT
  - Size: 1000
  - Limit price: 398.45
- Limit orders are what create(or make) the order book
  - Only orders that do not trade(or not trade completely) are “posted”(added) to the order book
  - For a buy order to be added to the order book, it’s limit price must be less than the lowest outstanding sell price
  - For a sell order to be added to the order book, it’s limit price must be more than the highest outstanding buy price

Definition: “Market-Maker”

- Trading firms that simultaneously post both bids and offers on the order book
- By posting both bids and orders, the order book becomes “two-sided” and the market now “exists”
- Other traders can now both buy and sell because of the resting bids and offers submitted by market makers
- When a new incoming order matches against a previous resting order, that new order is said to be “taking” or “aggressing”
- When an incoming buy or sell order does not “match”(trade), then it is said to “post” or “rest” on the order book
- Once resting, it will wait there to potentially be filled against future incoming orders on the opposite side:
  - Resting buy orders wait inside the order book for an incoming sell order willing to sell at or below the limit buy price
  - Resting sell orders wait inside the order book for an incoming buy order willing to buy at or above the limit sell price
-

Order Book(actual definition)

- The order book is a collection of orders where the buying and sellers DISAGREE on the “fair price”
  - Buyers are not willing to pay at least the lowest resting sell price
  - Sellers are not willing to be paid at least the highest resting
- This is essentially digital/automated “haggling” (bartering, negotiating, etc) on what is a “fair “price
- Order Book More Stuff:
  - It is constantly changing in response to all other orders, trades, and information on earth

    - to understand financial markets today is to understand this automated haggling process displayed on the ticker tape and order book
- Order Books as a data structure:
    - Can be viewed as a pair of priority queues
    - Buy order queue: orders are inserted with highest prices given priority(front of the queue)
    - Buy orders with lower prices are further back in the queue(because they will match only after higher prices)
    - Sell order queue: orders are inserted with the lowest prices given priority
    - Sell orders with higher prices are further back in the queue(because they will match only after the lower prices)
- What about when multiple limit orders are submitted with the same price?
    - Most exchanges operate(price, time) priority
        - "Better" prices (higher bids or lower asks) always get priority over "worse" prices (lower bids or higher asks)
        - At the same price, which every order was submitted "earlier" gets priority over later orders
            - Speed = priority
            - Priority = profit
            - Speed = priority = profit
        - "(price, time)" priority queues is why SPEED matters
- "Queue Priority"
    - Where your order is in the queue of other orders resting on the order book
    - To be faster than other traders it to be first in line in the priority queue
    - This allowes the fastest traders to capture profitable trading opportunities(prices) before slower traders
- Order book data structures continued:
    - Can also be viewed as a hash map or binary tree of queues
    - OrderBook[bid_or_ask][price].append(order) (add a new order on either bid or ask side, at the specific price) the the priority queue
    - Priority queues can also be implemented as linked lists, binary trees, etc
    - Caveat!
        - There are many different types of orders and participants which may create many more queues per price level
        - Example: "displayed" visible on order book vs "hidden"(in the queue but not displayed to other traders)
        - At the same price all "displayed" orders will take priority over any "hidden" orders resting at the back of the order book
        - Other examples of this may include enhanced priority for certain classes of teasers(retail, obligated market makers, etc)

Trades:

- A "trade" occurs when a buyer and seller agree on both instrument, price, and size

- One incoming buy order can match against or or multiple resting sell orders
- One incoming sell order can match against one or multiple resting buy orders
- We often refer to trades as “ticks” (dating back to physcial ticker plants” which used ot be an electronic type writer to print out prices)
    - “Ticker tape”: list of trades
    - “Ticker plants”: software application that sends out market data(trades, and/or quotes); we will cover this extensively later
    - “Tick-to-trade”: a measure of reaction of time of traderS(software or hardware) form time a message is received to reaction order is submitted

Other terminology:

- BBO: best bid and offer
    - Bbo example: 24.45 x 24.47
        - This summarizes that the highest bid is 24.45 and the lowest ask is 24.47
    - BBO summarizes quickly a rought estimate of the cost to both buy and sell some asset
    - “Bid-Ask Spread or simply “spread
        - Spread = best ask - best bid
- The spread determines how expensive it can be to trade in and then back out of an asset
- Bid-ask spread determines to cost for traders aggressing
- It also determines the profit of market-makers(because they are the ones selling it to you are the higher prices)
    - Whereas “taking” traders want a “tight” bid-ask price, market-makers prefer a “wide” bid-ask spread
- Definition: “tick size”
    - The minimum price increment between quotes
    - For example, in cash U.S. equities(stocks) the “tick size” on exchanges is $0.01(on penny)
    - On other trading venues and assets, it can vary wildly(.25, .000001, 1/16th, etc)
- Definition: Volume
    - Volume typically refers to the numbe of units(shares, contracts, etc) that a particular asset has traded
- Definition: Liquidity
    - Liquidity typically refers to the amount of units resting on the order book(how “liquid” the markets are)
    - Traders and exchanges prefer more liquid markets because it lowers the cost trading, especially when dealing in large size

Lecture 03: Market Data Feeds and Exchange Architecture

Public Data Feeds

- A data feed which is replicated to many different customers, either for free, or for cost. It provides information about the sate of a market that insen't limited to individual traders involved in the trade. Usually, all traders will receive(for a specific subscription) the exact same data via the public data feeds
- Many trading firms are gainst the exorbitant costs of sbuscribed to the various data feeds
- Exchanging profits from feeds, many exchanges derive a substantial amount of revenue form their offering of "proprietary" public

Private Data Feeds

- Feeds that are offered only to the customer actually trading and contain proprietary details about their own trading activity
- Private Data Feeds Analogy:
    - Think of the private feeds as the API computer's intended equivalent of what YOU see when you login to your own trading account: list of prior trades, fill prices, execution times, list of outstanding orders in the market, details on the order types, etc
- HFT traders rely on a combination of both public market data feeds and private order entry feeds to develop their understanding of the the past, current, and future state of the order book and financial markets

Value of Trading Live

- Because private data feeds(from trading activity) offer additional insights into the market structure, firms that trade a lot of have a systemic advantage vs new entrants or smaller firms who will know less about the market

Basics Review of Exchange Side Software Architecture

- Exchange Systems are composed of the following
    - Gateways(GWs)
    - Ticker Plants(TP)/ Market Data Generators(MDG)
    - Order Matching Engines(OME)

    - Drop Copy(DC)
- Gateways:
    - A “gateway” is an exchange-side application for which customers have a direct(typically TCP) connection for trading
    - Used by traders to both send and receive messages related to their own order traffic
- Messages to the Gateway:
    - Submit order execution messages:
        - Initiate a new order
        - Modify existing order(change size, price, etc)
        - Cancel existing order(whether unfilled or already partially filled)
    - Session-related messages(depends on exchange/protocol)
        - heartbeats (checking health of connection)
    - disable/enable order entry per account
    - Receive status updates on their own orders
        - Order accepted
        - Order rejected
        - Order filled(fully or partially)
        - Order cancelled (either buy cancel request or expiration)
- Session Allocation Options:
    - Typically customers will be randomly or otherwise allocated a “sesssion”/”port” on a particular gateway which is used by other customers
    - Sharing gateways can lead to unfairness. If your order is on a gateway being heavily used by another customer, it may reach the market slower than via other gateways
    - Some exchanges put all customers on a single gateway(per market)
    - Some exchanges allow customers to buy their own individual private gateway
    - Some exchanges charge for sessions, others give them for free but based on trading volume
- Ticker Plant:
    - A “ticket plant” is an exchange-side application that typically generates anonymous publicly available market data. Exchanges can have one or often many ticker plants offering different granularities of data feeds
- Order Matching Engine(OME):
    - The order matching engine is an exchange-side application responsible for actually processing customer orders, matching them if possible, and maintaining the order books of all outstanding orders that could be matched in the future. The OME is the brain of the exchange, doing the majority of the important operations
- Drop-Copy
    - Drop-Copy is an exchange-side application that streams the outbound order traffic(similar or identical to what is on the gateways) often combining the traffic from multiple trading gateway sessions, trading teams, or even multiple trading firms

    - Often used as a second source of all activity on the exchange (based on order acknowledgments, fills, etc)
    - Will typically be reconciled with what the individual trading machines are reporting on their activity
    - Can also be used by clearing firms to track the activity of sessions across multiple firms and fed into various risk systems

Data Feeds”Types”

- By types we mean a specific commonly offered stream of data continuing some subset of trades and / or the order book
- Common Types:
    - Last Trade/Last sale
    - Level 1: the top of book / “BBO”
    - Level 1/2: SIP: NBBO - “consolidated’ feeds
    - Level 2: Depth of book
        - Partial depth(best N levels)
        - Full depth (all price levels)
    - Level 3: Market by order

- Last Sale:
    - What is most commonly used on trading charts showing the “price”
    - Dramatically less data than order books(trades are much rarer than order submissions/updates)
    - Does not actually tell what the current price to buy or sell is
- Level 1 - Top of Book
    - Provides “BBO” - Best (highest) bid and Best(lowest) offer
    - May or may not include the actual size posted
    - Useful for knowing at least the “best” price one could get at a particular time (ignoring hidden, midpoint, etc)
    - Not as useful for knowing the full cost to trade (especially in larger sizes), as one can not see subsequent prices
- Level 2 - Depth of book
    - Provides depth of book information
    - At a minimum provides total(displayed) size at each price level
    - May provide the # of orders as well
    - May includes all price levels or only the nearest N best price levels(highest bids / lowest asks)
    - Often limited to 10 levels for historical reasons
- Level 3 - Market By Order
    - Provides details on every single (displayed) order added to the order book

  - Allows customers to:
    - Computer the queue priority of each order
    - Compute the level 2 depth of book (by adding up the total size per order)
    - Computer the level 1 (from level 2, just take the top)
  - Extremely useful for backtesting(otherwise you cannot estimate your queue priority accurately)
- Consolidated Feeds(SIP)
  - "SIP": Securities Information Processors
  - In cash us equities, every exchange must stream their current prices to the "SIP" aggregator for that particular stock
  - Every company choses the "primary listing" exchange to list their stock on (almost always NYSE or NASDAQ)
  - NASDAQ and NYSE are responsible for aggregating the trades and quotes form all US exchanges and providing the "SIP"
- "Direct" vs "Consolidated"
  - Consolidated feeds: feed that aggregate trades and quotes across all stock exchanges in the US
    - Cheaper
    - slower(all data form different data centers has to first be routed to the SIP location)
    - Missed some data (odd lots, order imbalances)
  - Direct Feeds: feeds directly from each exchange(NASDAQ, NYSE, CBOE, IEX, etc)
    - Much more expensive
    - faster(sometimes much faster)
- Odd lots vs Round Lots
  - Round lots are orders for shares in multiple of 100
  - Odd lots are orders for shares that are less than 100
- Consolidated Feeds(SIP):
  - The SIP contains some data (like trades and book updates) but most orders are missing(depth, odd lots, etc)
  - This has been controversial as it encourages (forces) traders to pay for more expensive direct feeds
  - Elite firms will typically build their own SIP internally from the direct feeds
- SIP - NYSE / CTA
  - "CTA": Consolidated Tape Association:
  - Name of NYSE's consolidated feed entity for doing consolidated feeds for all NYSE listed symbols
  - Also referred to as Tape A and Tape B
  - Tape A:
    - All NYSE listed stocks

    - Tape B:
        - Various other exchanges including NYSE America,ARCA, etc. Contains many ETFS
- “UTP”: unlisted trading privilege: here
    - Also referred to as “tape c” of the SIP
    - Type of UTP SIP:
        - UQDF - UTP Quotes
        - UTDF - UTP Trades
- Competing Consolidators
    - SEC has proposed “competing consolidators” to break the monopoly on SIP
    - Change voting structure
- Technical Aspects of Market Data:
    - “A” vs “B” feeds
        - Very common for exchanges to maintain two separate networks for redundancy. Each network contains either the “A” or the “B” feed of the market data
    - A/B Arbitration
        - For each multicast channel of market data, subscribe to both feeds
        - Choose which ever one arrives earliest(based on sequence
        - Discard the second message contiaining redundant information
    - Real World Examples:
        - CME GLINK
        - EUREK
    - A/B market data recording:
        - Due to storage costs, many first will only record the A feed
        - This is based on the false assumption that the A feed always comes first(False)
        - Smarter traders will record both feeds(or at least the timestamps of each message on both feeds)
        - If most of the market is only reacting to A, then you may hae advantage specifically when the B feed is first
        - When backtesting your strategies, you want to know when the earliest source of data arrived, not when the A feed alone arrived

Public Market Data networks:

- Most exchanges route both order entry(private messages) and market data(public messages) over the same networks
- This can introduce queuing and delays, both
    - To the individual customer receiving private messages and public messages
    - And to the other customers who have fiber connections to the same customer-facing switch

- At least one exchange (eurex) has separate networks for public market data vs private entry
- Many others do not, forcing customers to purchase more fiber optic cables to reduce collisions
    - This doesn't account for the problem of other trader's activities slowing down your own connection
- Multiple feeds per fiber
    - For extremely latency competitive environments, traders also face the challenge of how many feeds to subscribe to on a single fiber optic cable
- ## Feed/Fiber options
    - cheapest/easiest solution: subscribe to all data feeds on a single fiber (or pair if A/B)
        - Risks queueing where other traders find out about trades faster than you because you weren't listening on other channels
    - Feed/fiber decisions
        - Expensive: use dedicated fiber(s) for each individual multicast channel
        - Reduces risk of collision and queueing leading to delayed important information
        - Can also be combined with separating out private from public data individually(even if not done by the exchange)
        - Can lead to an explosion in the number of cables needed to connect to an exchange
        - Exchanges may not mind this as they make insane $$$ from fiber connections
- ## Technical Aspects: Batched or Individual Data Feeds
    - ### Fill Distribution:
        - Suppose a single very large order for 1000 lots matches with 1000 passive resting orders within the matching engine itself in the markets this results in one aggressor incoming trade matching with 10000 resting passive orders. This is technically one thousand separate trades, all part of a single "matching event"
    - ### Fill Distribution Decisions:
        - Should the exchange send:
            - A single message detailing that 1000 lots were filled?
            - A stream of individual one-lot fill messages?
            - Batches of every N orders being filled
        - ### Fill Distribution
            - Some exchanges will send individual fill messages
            - Other exchanges will send a single aggregated message(especially if only last sale, BBO, etc)
            - Some exchanges even mix and match depending on the specific market and matching algorithm(FIFO vs non-FIFO)

- Fill Distribution Innovation - “trade summary”
  - CME introduced a feature where before broadcasting individual fill messages per order it first sends out a “trade summary message” which dictates the total fill size that is about to be described in more detail in subsequent messages(and possible packets)
  - Technical Aspect -Multicast Feeds
    - Exchanges have to choose how many different multicast feeds to offer. Offering multiple multicast feeds allows the exchange to distribute and parallelize matching and ticker plants across different machines potentially improving performance(not all order books have to be on the same machine)
    - Other exchanges of single multicast feed containing all trading symbols)
  - Other Data Feed Topics
    - Latency to receive data from other exchanges
      - Virtu SUED NYSE over advantageous microwave tower_
    - Do public market data feeds reveal “too much”
  - Speed bumps to prevent public(and private) market data form revealing trades ‘too early”

Lecture 7: Computer Architecture for ULL

Overview of a computer:

- Motherboard
- CPU
- RAM DIMMs
- PCIe
  - Storage
  - Networking
  - Chipset
  - Power

Overview of a modern CPU

- ALU: arithmetic logic unit
- CISC(Intel) vs. (RISC(MIPS, ARM, etc.) instruction set debate
  - CISC: complex assembly instructions that do several things per instruction
- RISC:
  - Simple instruction set that requires more operations, but each operation faster

Common ISAs(Instruction Set Architecture)

- x86_64(Intel and AMD)
- ARM(32 and 64bit)
- RISC-V(open-source-gaining attention as alternative to ARM)
- FPGA related CPUs

“P” vs “E” Cores

- Recent architectures will sometimes have two different types of cores
    - “P” core: (p)erformance core. Larger and higher
    - “E” core: (e)fficient core, smaller with lower power consumption but also lower performance

Caches explained:

- The “further” away data is form the hardware doing the computation, the slower the transfer
- Registers and L1 cache the fastest
- L2 higher latency than L1
- L3 higher latency than L2
- RAM higher latency than L3
- NUMA RAM on another socket is higher latency than locally attached RAM
- Hard drives/storage is higher latency than RAM
- Remote hard drives/storage(can be) higher latency than local storage

Cache Hierarchy

- Caching algorithms: temporal and spatial locality
- L1(Level 1) Separate Instruction and Data Caches
- Raw assembly instructions vs compiled micro-ops
- L2 shared cache per core
- L3 shared cache per die
- L4 - eDRAM

SIMD(Single instruction multiple data)

- MMX, SSE, SSE2, SSE3, AVX, AVX512
- Optimized insturctioons that can operate on multiple variables at once
- Think linear algebra, matrix multiplication, DSP routines(fourier transforms), etc
- Also usedul for parsing/ tokenizing (locate all bytes where thee is a space, etc)
- Very useful for audio and video processing(adain DSP/FFT)

Parallelization Techinques:

- Simultaneous Multithreading(SMT) - Intel’s “Hyper-Threading”
- Useful if two threads are using different parts of the same ALU or the same core
- Some HFT technologists insist on always disabling it, others (like me) will sometimes use it carefully fine tune everything

Parallelization Conitnued:

- Multiple cores per socket
- Multiple sockets per server
- Multiple servers per rack
- Multiple racks per data center

Intel XEO CPU E5-2643 v3

- Haswell (thrid gen) of core architecture
- 6 cores/ 12thread(HT)
- QPI Lanes

E5-2643 v3-Cache

- L1 Cache per core

    - L1(I): 32KB per core for microops (instructions)
    - L1(D): 32KB per core for data
- L2 cache; per core, shared between data and instructions
- L3 cache: 20MB- shared across all cores on the CPU

E5-2653 v3 - Clock Speeds

- Max Turbo Frequency: 3.7GHz
- Intel Turbo Boost Technology 2.0 Frequency: 3.7
- Processor Base Frequency: 3.7 GHz
- TDP(Thermal Design Power): 135W

E5-2653 v3 - Instruction Set Features:

- Intel AVX2 (SIMD vector instructions): 256-bit instructions
    - Fused multiply-add (FMDA3): multiple/accumualte(DSP, etc)
- Gather support: single instruction to fetch memory from non-contiguous regions(DMA)
- Integer ops; shifts, blends, arithmetic operations
- Bit manipulation:
    - Bitwise AND/OR/XOR across 256-bits per instruction
- INTEL AES instruction

Vector Warning!

- Vector instructions like AVXX2, AVX512, etc) can do a lot more work per clock
- But this comes at the cost of greater heat
- Therefore common that if using these specific instructions, max clock speed is lowered
- Therefore some people thinking they're speeding everything up in fact they are actually lower fMAX
- If enough fo their code is NOT vector reliant, theb the lower fmax negates
- Need to check fo every single application and evne potentially thread/core

E5-2653 v3-Virtualization Techomlogy VT-x

- Intel VT-x hardware acceleration for virtual machines
- Very common for many years now, used by virtualBox, VMware, Xen, you name it

ET-2653 v3 VT-d

- Intel VT-d: hardware acceleration for virtualization of I/O
    - Helps to enable “bare metal” performance when running virtualized machines
    - Can be used to give a VM dedicated access to PIC-E cards(like GPUs or NICs)
    - Some NIC drivers/vendors also have accelerated support for sharing VMs built into the NICs themselves

Why did CPUs stop getting faster:

- CPU DB: Recording Microprocessor HISTORY
- Frequency Clock Scaling - INTEL's TurboBoost
- Server-grade XEON(locked) vs. Consumer Gaming (unlcoked) multipliers

Expanding Roles of the CPU:

- Integrated Memory Controller(Nehalem)
- Integrated PCI Express controller (sandy bridge)
- Integrated GPUs(foundin Intel and M1 laptops)
- Other accelerators(neural networks, Gaussian accelerators for voice recognition, etc.)

- CPU to CPU interconnects from INTEL
    - North-Bridge
    - QPI
    - UPI
    - CXL(potentially transformative

Transition to heterogenous computing:

- GPUsNVdia, AMD, even Intel)
- FPGAs (Altera - now intel, and Xilinx - now AMD)
- AI ASIC accelerators (Google TPU, Intel GMM NNA, Apple Neural)

Low Latency Tuning Techniques:

- “Performance”
    - Can be viewed either from “highest throughput” or “lowest latency”
- Highest throughput: the most amount of data can be pushed through the system
    - Don’t care if the “first” result is delayed, so long as the time to process ALL data is lowest
- Lowest latency: the fastest reaction time to an individual packet
    - Care less about throughput(within limits) but fastest reaction time to market data/order messages

Throughput vs Latency:

- Often throughput and latency have opposite setting and preferences
- Maximizng throughput can dramatically hurt latency
- Minimizing latency can dramatically hurt throughput
- Think “webservers handling million of GET request for cat photos” vs “getting market data and placing a trade to avoid losing $1 billion”

Kernel-Bypass:

- Allow user mode code to directly read from and write to NIC
- Can substantially reduce latency
- Solarflare’s onload was the defacto for this:
    - Works by intercepting C-calls to the network and passing them directly to NIC instead of visa the kernel
    - Eliminateds the need for a (latency expensive) context switch from user to OS to user mode again
    - Extremely common in trading optimization
- Mellanox also had VMA, but it was never as popular
    - Much better latency but mistakes wre made

NUMA:

- Non-uniform memory access
- Different CPU cores will have different latencies to memory and PCI-e
- PCI-E == network card used for market data and trading
- Careful tune which cores access which memory and which network cards
- Rely on single producer single consumer data structures and lcokelss queues

NUMA- Single vs Multisocket

- The “great” HFT debate

- For dual socket: use one socket with a lot of OS tasks, keeping the 2nd “cleaner” with regards to cache
- Managing multiple sockets can make scheduling threads and IRQs more complicated
- Comes with the advantage of double (or more more) the number of cores, cache, PCIe lanes, etc

Interrupt Handling:

- Turn of IRQ coalescing(batches up multiple events before notifying kernel with IRQ)
- IRQ steering: carefully control which cores process which interrupts
    - Interrupts you don’t care about for latency, steer to the “dump core”
    - Interrupts that ar relevant to trading, perhaps assign to a completely dedicated core
    - Can tuen via driver configs but also directly on the command line using proc
- Other kernel tuning parameters
    - Adjust buffer sizes
    - Enable/disable certain nic features(checksum offloading, etc)
    - Mdofiy TCP stack parameters to reduce latency

TCP Features:

- TCP_NODELAY
- Selective ACKs
- Possible DSCP flags (used for QoS)
- TCP windows sizes

CPU core affiuintiy / isol_cpi

- Control which programs and threads will run on which core
- Steer all non critical threads to the “dump core(s)”
- Keep pristine cores for the ultra low latency and sensitive tasks
- Attempt to ensure that code for a particular thread can fit in L1 Cache
    - Avoid recompilation
- Ensure cores don’t keep hopping across NUMa nodes with data
- Fewer cores -> higher clock frrequnecy
- Not always obvious the exact proper solution, highly code and market dependent

NMI:

- NMI is a non maskable interrupt
    - interrup s that can’t be turend off by the operation system
- Used for tasks like notifying about memory errors, checking temperature, adjust fan speeds
- Can disable in the BIOS(sometimes)

Tickless Kernel

- By default, kernel used to be configured to interrupt every core “a lot” (100-1000Hz)
- This can corrupt cache and interrupt critical trading threads running on dedicated CPUs
- Used to require recompilation of kern; but can now be set by a kernel flag at runtime
- Modern kernels now use tickless systems by default, but you may end up working on older systems

CPU sleep states

- By default, unsed CPUs will go into lower modes, so called “C-states”

- This saves power, but at the cost of higher latency to bring the C-state back to fully functioning
- Often common to disable these energy saving C-states entirely
- Also common to disable all core that aren't critically needed to reduce power
- More advanced coding may sometimes attempt to spin up cores to handle backglogs but otherwise allow powering down

Use Huge Pages

- By default linux kernel allocates and traks memory on a relatively small scale -4kb
- If you application is using many GBs (or even TB) of ram, this ia a lot of pages to track
- Huge pages change the page size to 2MB or 1 GB depending on CPU support

Hardware Checks

- PCIe speeds/ version(PCIe v2, v3, v4, etc)
- Number of PICe lanes(a 16x slot may only have 4 or 8x lanes)
- Power management and other advanced features enabled in BIOS
- Transceiver speed configs (1gbps, 10gbps, 25gbps)
- Power requirements(possible external GPU plugs)
- Additional cooling requirements(default fan controller of system may not read network card temps)

Advanced NIC features

- Receive side scaling(of interrupts)
- Receive flow steering
    - Interrupts to the core reading the data
- Transmite Packet Steering (XPS)
    - Selects which transmit queue to use

Lower Level Packets

- Regular socket calls
- Kernel bypass accelerated(onload, vma)
- TCPDirect
- Raw usermode
    - Efvi
- CTPIO
- FPGA-assisted
    - LDA example
    - Delegated send
    -

Lecture 11 - Networking

Ethernet:

- Constitutes both layer 1 (physical layer) and layer 2 (data link) layer in the OSI stack
- Ethernet is the most common wired networking protocol
- Describes methods of locally transmitting data between devices
- Relines on higher level protocols(IP, UDP, TCP, etc)

- Was not intended to scale globally, used for local networks

Ethernet(in finance)

- All colo exchanges that offer direct connections to customers rely on ethernet
- Most exchanges typically rely o fiber optic connections(either MM SR or SM LR)
- In the US, most commonly 10 GBPS and 40GBPS depending on venue

Ehternet Frame layout:

- Destination MAC address(6 bytes)
- Source MAC address (6bytes)
- “Ether Type” 2 bytes) - what type of packet is in side the ethernet frame
- Payload - actual data inside; typical IP
- FCS: Field Checksum(4 Bytes)

Internet Protocol:

- Intended for globally linking multiple networking
- Initially ufnded by DARPA especially for military networks during the cold war
- Each computer and router maintains “routing tables” that describe how to pass the packet on the next hop
- Eventually packet sqwill hopefully reach their destination

UDP:

- Extremely simple stateless protocol
- Low overhead
- Allows for both unicast, multicast, broadcast
- Not designed to be reliable
    - UDP packets can arrive out of order and the client software will get packets out of order
    - UDP packets can be lost and the protocol itself does not support reliability

UDP in Finances:

- For colo, UDP multicast is the defacto standard for distributing market data
- By using multicast, an exchange computer need only send a single message
- That one message is automatically replicated by the routers to all clients listening
- All clients will hopefully receive that packet between 1ns - 1000 ns

UDP Header

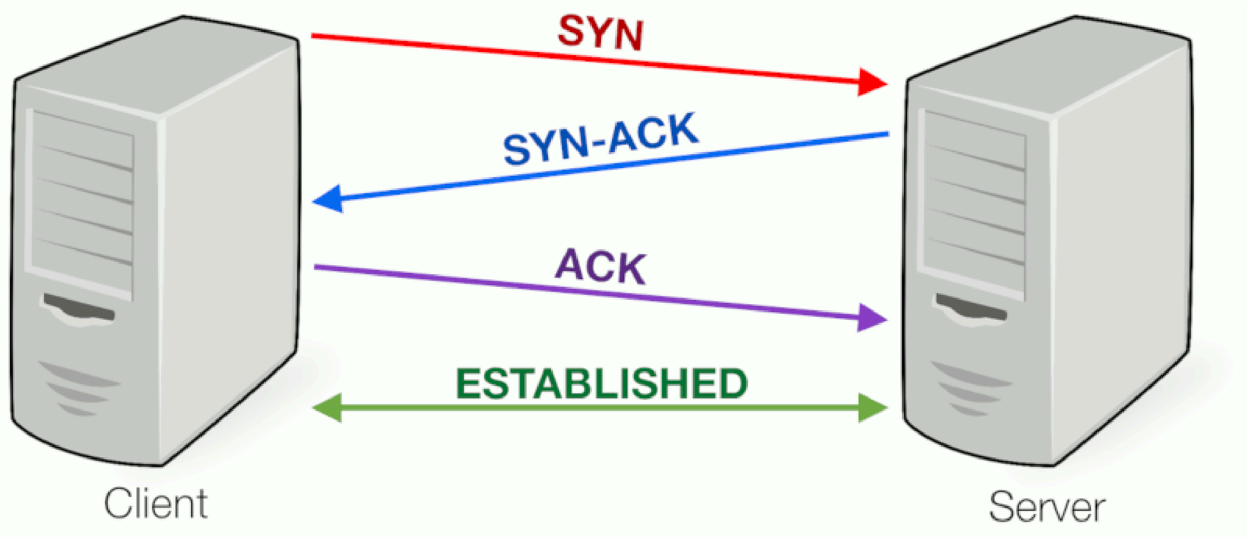
- Source port(2 bytes)
- Dest port(2 bytes)
- Length (of both header and payload) (2 bytes)
- Checksum (2 bytes)

TCP

- Designed for reliability
    - Ensures that data is received by the user in order
    - Ensured that missing/propped packets are retransmitted
    - Accomplishes this by establishing sequence numbers for both sides
- Whenever a pretty send N bytes they increment their sequence number sid ebay N Sender saves the data they send until it is acknowledged

TCP Session Creation

- “Three way handshake”

- Server begins listening on a particular port(and specific IP maybe)
- Client attempts to connect to that IP and port of the server
- Thus the server begins the “three way handshake”
- 

TCP vs UDP:

- UDP can be considered like sending a text message
    - Sometimes you get text messages out of order
    - Sometimes one of N text messages fail to send but the others jumbled
    - Recipients of text messages see that messages are out of order/missing
- TCP is like a phone call

TCP reliability:

- How can the receiving side detect that data is missing?
    - Sender starts with a sequence of N (randomly selected)
    - Receiver rcieves 100 bytes with seq N
    - Receiver receives another 100 bytes with seq N + 100
    - Receiver receives another 100 bytes with seq N + 200
- Receiver suddenly receives another 100 bytes
    - But with seq N + 400
    - what s wrong?
        - Sequence number should have increased by 100, not 200
        - Receiver knows that 100 bytes of data is missing
        - Those 100 bytes are from the previous packets

TCP window sizes:

- For each connection, each TCP receiver must allocate memory to store the incoming data
- Senders must monitor the window size of teh other side(receiver)
- Must ensure not to overfill the buffer resulting in guaranteed drops

TCP enhanced features:

- SACK
    - Selective acknowledgement

- By default, say 100 packets are sent, and the receiver reports that data corresponding to packet 3 is missing…
  - TCP engine will retransmit the datta in packets 3-1000
  - This is extremely wasteful
- Sack enhances TCP so that the receiver reports each packet it receives
- This way the transmitter only needs to retransmit the missing data
  - For exmaple only resend data from packet 3
  - Rather than having to resend data for packets 3-100

Nagle's Algorithm

- Imagine you are writing an app like telnet or ssh, the usertypes one keystroke at a time:
- QUestion:
  - Do you trtransmit an entire packet for every character
  - Minimum sized ethernet frame for IP/TCP:
    - ethernet(14) + IP(20) + TCP(20) + payload(1 byte) - rounded up to 64 bytes
    - Efficiency - 1/64 = ~1.5
- The TCP stack waits for more data to be request to be sent before sending the actual packet
- Useful for increasing bandwidth(maybe) when user code is providing a few bytes at a time
- Can be atrocious for HFT code
- Your code calls send(orderPayload) and then moves on to other lines
- Did you order actually get sent out on the wire
- When did it actually get sent out on the write

Window Size Scaling:

- Original TCP protocol dates aback to 1970s when memory and link speeds were much slower/smaller
- Original TCP protocol: window size field is oly 2 bytes
- 2^16th = `64kb
- 64kb is tiny especially on a 10gbps ethernet connection
- Window scaling: allows for much larger window sizes

TCP Keepalive:

- By default if no data is sent by either party, no packets are sent
- Makes it more difficult to determine if the other sied is finished
- TCP keep alive periodically send essentially probe messages to make sure the other side is still there
- Could have trading latency advantages because of caching
- However, if those probe messages block actual order messages from going out, not good

TCP in finance:

- The definitive protocol(typically) for order entry/private order status
- Much more complex protocol
- Many vendors and firms implement their own TCp stacks to save on latency

- Can lead to vendor incompatibility
- Can lead to extremely weird bugs and edge cases that can crash or lock markets

Prolems with TCP

- You are a trader. A very important market event happens. On a tcp connection you submit thousands of orders in order to mitigate risk. An order is dropped, what happens to your orders?!
- You are an exchange
    - You are listening on multiple ports, one port per customer session
    - You receive 10 messages from 10 traders, the following order
    - What order doe s teh TCP stack actually return these messages to the software
        - Based on the time?
        - Based on the port number?
- You are an exchange
    - For fairness, you allow all customers to connect to a single gateway, all messages fromall clients come down a single NIC
        - One customer has many fills sent ot them
        - The customer's TCP stack request that gateway to retransmit those packets

TCP optimizations

- TCP NIC
    - Many acceleration features are now built into the NIC itself
    - checksums(ethernet, IP, TCP)
    - Batching up TCP frames into large transmits

Lower Level TCP

- Regular socket calls
- Kernell bypass accelerate (onload, vma)
- TCP direct
- Raw usermode
    - Efvi
- CTPIO
- FPGA-assisted
    - LDA example
    - Delegated send

TCP in Hardware:

- TOE-TCP offload engine
    - An IP block commonly sold by vendors for implementing TCP in a low latency fashion
- Typically coupled as a part of a broader set of protocols
    - TCP + IP + ethernet MAC + ethernet PCS
    - By putting all state in a single IP core reduces dual clock fifos
    - Lower latency than separating functionality
    -

Lecture 12 - Timing and Capture

Local Clocks:

- In electrical engineering, local clocks are typically driven by some sort of oscillator
- The goal of the oscillator is to toggle between values at a pre-defined and deterministic frequency or period
- Oscillator can be intending for either analog or digital means

Oscillator Problems:

- This period/frequency is not completely stable in many instances
  - Temperature: typically temperature coefficient therefore heat alters the local oscillators
  - Age: components “wear” overtime and their performance can change
  - Componeont variation in manufacturing process

Oscillator Types:

- Crystal Oscillators(XO, VXCO, TXCO, OXCO)
- Atomic oscillators(cesium, rubidium)
- GNSS oscillators (GPSDO)

Crystal Oscillators:

- Quartz Crystals:
  - Exhibits piezoelectric effect: electric field generation mechanical oscillations and vice versa
  - Similar to a microphone/speaker
- XO
  - Refers to a “regular” / simple crystal oscillator
  - Found on countless electronics
- XO issues
  - The frequency of oscillations of a quartz crystal is not perfect
  - Therefore numerous more advanced configurations have been created to compensate for this
    - VXCO
    - TCXO
    - OXCO
- VXCO
  - Voltage controlled crystal oscillator
  - Allows for adjusting the frequency slightly by applying an external voltage control signal
  - Variable voltage applied to a varactor diode or variable capacitor
  - By adjusting the capacitance, adjusts teh resonant frequency of the circuit allowing for fine tuning
- TCXO
  - Temperature-controlled crystal oscillator
  - One of the primary variables that can alter the resonance of XOs is the ambient temperature around the circuit

  - TCXOs attempt to adjust for and correct basedon the sensed temperate(for example, thermistors)
- OXCO
  - Oven controlled Crystal oscillator
  - TXCO attempts to mitigate the impact of temperate by compensating for the direct impact of temperature changes
  - OXCO attempts to eliminate the impact of temperature by eliminating temperature changes
  - House the XO inside of a mini oven which is kept at a specific temperate of often 75-90 degrees celsius
  - OXCOs have better performance at the cost of larger size and higher power consumption

Definition of a second

- The current international standard definition of one second is define by
  - The duration of 9,192,631,770 periods of the radiation corresponding to the transition between two hyperfine levels fo the ground state of the cesium -133 atom
  - In other words, the definition of “time” is literally defined by the physics of specific atoms
  - Some of the most prices clocks are atomic clocks

Atomic clocks types:

- Cesium
- Rubidium
  - Compared to cesium atomic clocks, rubidium clocks are:
    - Lower cost
    - Lower power
    - Smaller size
- Rubidium Disadvantages
  - Disadvantages over cesium:
    - Cesium is the official standard
    - Rubidium clocks are more susceptible to local environmental
    - Cesium clocks are more reliable ove the logn term
- CSAC
  - Chip scale atomic clock
  - Relatively new
  - Much smaller than tradition clocks and can fit on mother board
  - Still based on cesium but on a much smaller device
- Hydrogen Masers
  - Super fancy
  - No one in finance(publicly) has one of these

GPSDO

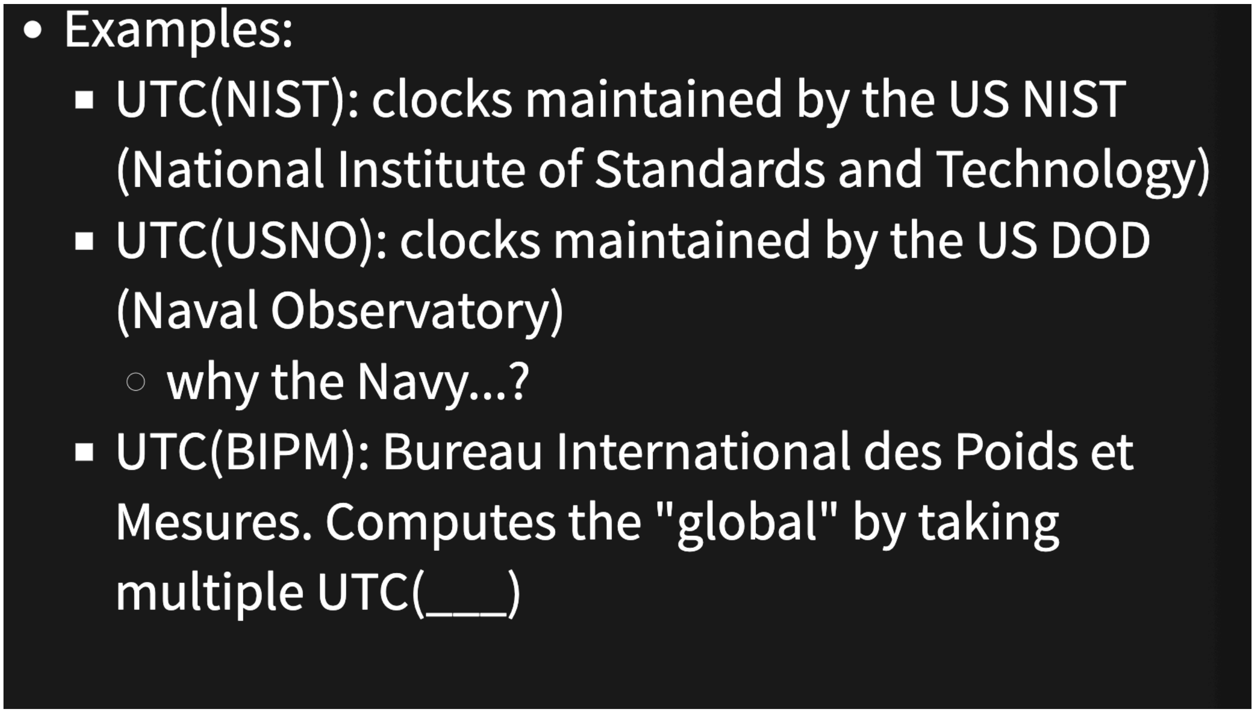
- Global positioning system disciplined oscillator

- Locates a single GPS satellite and uses its waveform to generate a prices reference signal
- Somewhat “cheating” as its really relyingo n the atomic lcocks inside the GPS sattelite
- Relatively low cost, useful when one has line of sight of satellites(desont’ work well indoors)

Time Synchronization

- We need to synchronize multiple clocks not on same circuit board or computer
    - Synchronize our own geographically distributed clocks
    - Synchronize our own local time against one or more definitive sources of time

Timing Standards

- Naming Scheme “UTC(___” where ___ is some official source
- 

- Examples:
    - UTC(NIST): clocks maintained by the US NIST (National Institute of Standards and Technology)
    - UTC(USNO): clocks maintained by the US DOD (Naval Observatory)
        - why the Navy...?
    - UTC(BIPM): Bureau International des Poids et Mesures. Computes the "global" by taking multiple UTC(___)

Difference across standards

- The great thing about standards is that everyone has their own
- Even two different government run labs from teh same government both with fancy atomic clocks can’t agree to the nanosecond or even tens of nanoseconds

Mistakes were made:

- 2011 researchers working on OPERA experiment at CERN detected Neutrinos traveling faster then speed of light
- Was caused by a loose cabel between a gps receiver and the detector
- Once the connection fixed, the speed of light held

Clock Synchronization:

- NTP
- PTP
- PTM
- GPS/PPS

NTP

- Network time protocol
- Been around for decades
- Essentially your computer sends a message “what time is it?”
- And an NTP server replies with what time it is
- What are the problems with this?

PTP

- Precision time protocol
- Attempts to correct for some of the flaws of NTP

PTM

- Prevision time measurement
- Just as non-constant transmit times across networks inhibits NTP, so does transmit across PCIe between NICs and CPUs
- PTM attempts to correct for this variability across the PCI-express bus

GPS

- Global positioning service (refers to teh US DOD’s specific system)
- More broadly ow referred to as a GNSS (global navigation satellite systems)
- How does GPS work?
  - Each satellite maintains multiple precise clocks
  - Each satellite broadcasts out both where it is (almanac) and what time it is when sending out
- Receivers (without nay send of actual time)
  - Receive almanac
  - Receive multiple signals
  - Solve system of equations
- GPS math
  - Solves for 4 variables
    - Latitude
    - Longitude
    - Altitude
    - Time
- GNSS Networks
  - USA:GPS
  - EU: GALILEO
  - RUSSIA: GLONASS(don’t use)
  - China: Beidou
  - INDIA ; NavIC (not global)
  - Japan: QZSS(not global)
- GPS problems
  - Extremely lower power
  - Subject to jamming and spoofing
  - Speed of light is not constant through the air

- Correcting for Ionosphere
    - There is a frequency component to the distortion / speed change of EM waves
    - If a GNSS receiver is a multi-frequency(and those frequencies are sufficiently different), possible to correct
    - L1 / L5 receivers (now in the lab)
- Other corrections
    - Use of external data sources(GPS assistance services)
    - Relies on fixed receivers around the world that monitor GNSS signals and then computes required corrections
    - Requires internet connection but this correction data can be downloaded
- Somewhat local syncing
    - GNSS receivers have “special mode” that allow one to lock two GNSS receivers within a few kilometers
    - If they’re located in approximately the region, then linkely to suffer same ionospheric distortions
    - This supposedly allows for synchronization with +/- 1-2 nanoseconds
- Time in the Data Center
    - “Grandmasters”: typically dedicated computers that contain and act as:
        - GNSS receiver
        - NTP and PTP grandmaster
        - PPS(pulse per second) source
        - 10Mhz or other reference signal source

Rahul’s Notes

Lecture 2

Data Center:

- Barbarians at the Gateways from Barbarian at the Gate which is about private equity firms taking companies assets and selling them
- Data center: massive cooling, power, network, computers
    - A datacenter is a building built to house a lot of computers
    - Co-lovation- places where trading companies can put their computers in the same place as the exchange
        - Make markets fairer, every trader can put computers the same distance away
        - Don’t want traders to buy buildings next to it
        - Generally guarantee co-location, price list per month
- Milimiter wave length, 10 GBPS
    - In air .99 speed of light
    - Fiber optic is .6 speed of light

  - If it rains, it goes down
  - Mahway(NYSE), Nasdaq
  - CME/Cermak in Chicago
- Inside a data Center
  - Racks - 19in wide, muktieple server
  - Don’t build walls around server racks because of cooling
  - Height is standardized
  - Don’t design your own chips and racks
- Exchange:
  - Gateway:
  - Private message by people specifically licensed to connect to the exchange
    - Tickerplant: produce all the public data that trailers get.

Questions from Rahul’s Lectures

What is Broker-Dealer vs Broker?

- A broker is the business of buying and selling securities on behalf of its clients
- And a dealer buys and sells securities for its own accounts
- Broker-Dealer may appeal to investors who want to be more proactive in managing their own portfolio
- Market Maker: A market maker is typically a brokerage firm or a specialist firm that stands ready to buy and sell securities at publicly quoted prices. They facilitate trading by providing liquidity to the market. Market makers continuously quote bid and ask prices for securities ensuring there’s always a buyer or seller for particular security. They make make money though the spread which is the difference in bid and ask prices
- Market Taker: is an individual or institution that accepts the prices offered by market makers and executes those trades at prices. They typically place market orders or take liquidy from the order book consuming the liquidity provided by market makers.

Packet Capture in High-Frequency Trading (HFT)

1. Introduction to Packet Capture

- Definition: Packet capture involves recording network traffic so that the data can be analyzed later. In the context of HFT, it refers to capturing all network traffic related to market data and trade execution.
- Purpose: Used to analyze network efficiency, audit trading strategies, and ensure compliance with regulatory requirements.

2. Importance in HFT

- Latency Sensitivity: HFT relies on ultra-low latency. Packet capture helps in identifying network delays and potential bottlenecks.
- Strategy Validation: Traders use packet capture to verify that their algorithms are performing as expected in real trading environments.
- Compliance and Audit: Regulatory bodies often require records of all trades and quotes. Packet capture ensures that HFT firms maintain complete logs of their trading activities.

3. Technologies Used

- Network Taps and Span Ports:
  - Network Taps: Devices used to access data flowing through a network without interrupting the flow.
  - Span Ports: Mirrored ports on a switch that send a copy of network packets to another port on the same switch where the capture device is connected.
- Timestamping: High-precision time-stamps are crucial in HFT for latency measurement and sequence tracking.
- Packet Capture Software: Tools like Wireshark for general use, or more specialized software designed for high-speed trading environments.

4. Data Captured

- Market Data: Real-time price updates, volume, and order book status.
- Trade Execution Data: Details about order submissions, modifications, cancellations, and trade confirmations.

5. Analysis of Captured Data

- Latency Analysis: Measure the time taken for orders to reach the exchange and for responses to come back.
- Pattern Recognition: Identify patterns in trading data that could indicate successful strategies or potential market abuse.
- Algorithm Testing: Compare theoretical trading results with actual performance as seen in the captured packets.

6. Considerations and Best Practices

- Storage and Bandwidth: Packet capture in HFT generates massive amounts of data, necessitating ample storage capacity and bandwidth.
- Security: Captured packets may contain sensitive information. Ensuring data security and compliance with data protection regulations is critical.
- Performance Impact: Minimizing the impact on trading system performance while capturing packets is essential. Using hardware-accelerated or dedicated capture solutions can help mitigate performance degradation.

7. Regulatory Aspects

- MiFID II in Europe: Requires accurate timestamping and record-keeping for all trades.
- SEC Rule 613 in the U.S.: Mandates consolidated audit trails for all trading activities.

8. Future Trends

- Increased Resolution: As exchanges and trading entities strive for greater precision, the resolution of packet capture may increase.
- Machine Learning: Use of AI and machine learning to analyze captured data for complex patterns and predictive analytics.

Conclusion

Packet capture is a critical component of the technological infrastructure in high-frequency trading. It not only aids in operational efficiency and strategy refinement but also ensures compliance with increasingly stringent regulatory requirements.

This note sheet should provide a comprehensive overview and serve as a reference point for deeper exploration or implementation in HFT environments.