Scaling Real-World Large Financial Systems

From iPhone to Cloud

John Davies

Your host

John Davies

- Serious geek
  - Hardware, assembler, C, Objective-C, C++, Java, Occam, this that and the other etc.
  - Author of several Java related books, mostly Wrox, J2EE, JMS etc. and recently O'Reilly's 97 Things...
- Ex-global chief architect at two major investment banks
  - BNP Paribas and JP Morgan Chase
- Co-founder and CTO of C24, sold to IONA in 2007
  - IONA was then sold to Progress Software in 2008
- Co-founder and CTO of Incept5 and director of OnixS
  - Board advisor and Chief Architect at Verifi
- Father to three boys lovely boys and married to a French wife

Agenda

- What's it all about?
  - Trading 101, Exchanges, Futures, Options, Positions etc.
- What about the technology?
  - Faster is better?
- Algo-trading on the CME, ICE, NYSE, LSE
  - Algorithmic trading engines, sounds cool, what are they?
- Using Amazon's EC2 (trading in the cloud)
  - Yes, this talk is hype and buzz-word compliant
- Where do the phones come in?
  - Sliced bread for the traders

Your Keynote speaker

- At 2:30 this morning!

It's going to hit the fans!

Computer trading fears

An explosion in the amount of trading propelled by computers is ...rs that exchanges could be ...ut by rogue trades triggered ... running out of control.

Computer-driven trading boom raises meltdown fears

Trading in equities and derivatives is being driven increasingly by mathematical algorithms used in computer programs. They allow trading to take place automatically in response to market data and news, deciding when and how much to trade similar to the autopilot function in aircraft.

Analysts estimate that up to 60 per cent of the trading in equities markets is driven in this way.

- From the yesterday's FT (26 Feb 2010)

Trading 101 (the basics)

- The trading floor is simply a large market where people agree the price before taking delivery
  - Agriculture (Corn, Soybeans, Soy Products, Wheat, Pork, Cattle, Butter, Milk etc.)
  - Metals (Gold, Silver, Copper, Lead, Zinc etc.)
  - Indices (NASDAQ, S&P, DJ etc)

- The price can be for immediate delivery or delivery at some time in the future

- Some products are physically delivered, some are virtual
  - Gold is traded but rarely delivered, the owner's name simply changes
  - Stocks and shares, warrants and bonds are "paper", the titles change but there is no delivery

Exchanges everywhere

- There are literally dozens of exchanges, it's a confusing scene as they're all buying and merging with each other
  - CME (Chicago Mercantile Exchange)
  - ICE (InterContinentalExchange)
  - NYSE (New York Stock Exchange)
  - LSE (London Stock Exchange)
  - OMX (Aktiebolaget Optionsmäklarna/Helsinki Stock Exchange

- Each exchange specialises in local markets and stock
  - Some stocks are traded on several markets

- Many exchanges trade the same commodities so prices can very from exchange to exchange
  - Foreign Exchange, Gold, Copper, Oil, Rice etc. etc.

Futures

- Imagine a miller who buys wheat from a farmer
  - The price of wheat changes every day based on the available supply and usual market jitters
- The miller wants to be sure he can still buy wheat in 6 months time so he locks in a price
  - This is a 6 month future
- If the price goes up the miller wins because he can buy at the agreed price, if the price goes down he's lost out and the farmer wins
  - Either way, both parties have reduced the risk of price variance by agreeing a future price
- In theory the miller could have also hedged against falling wheat prices
  - A clever miller might sell the contract and (in theory) hedge the loss on the cash price against the gain in the futures price

Options

- An option, as the name suggests is an option to buy, not an obligation
- In the previous example the miller could have taken out an option to buy at an agreed price (the strike price)
- Option prices are never quite as good as the future price because someone else is taking on the risk
  - The price is not as good but using an option the miller can avoid loosing too much if the price falls, it's like an insurance
- The right to buy and sell are called put and call options

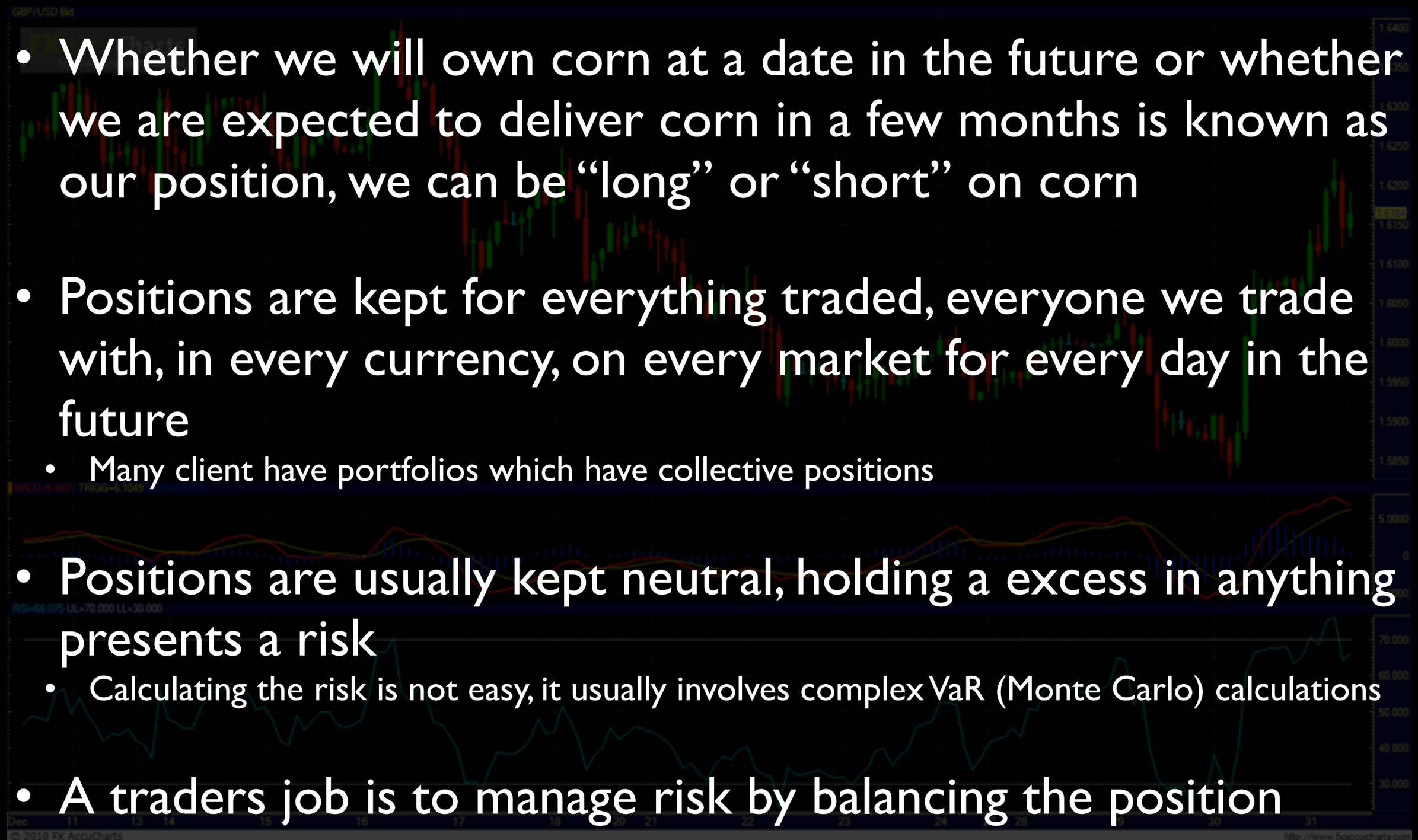
And how do we make money?

- Imagine a commodity, corn this time
  - We can buy it today or get a future price, one month, two months, one year etc. or we can make a call (option)
  - We can buy one month and sell another
  - We can combine buying, selling, puts and calls in the same future
- Every price changes by the second, each market it slightly different from the next

- Buy corn at €137/tonne and sell at €137⅜ and you make money (37½c per tonne)
  - Buy 5,000 tonnes and that's €1,875

- Buy 1 month and sell 2 months and cover that with another trade to sell 1 month and buy 2 months and you've no need to take delivery or part with cash

Positions



- Whether we will own corn at a date in the future or whether we are expected to deliver corn in a few months is known as our position, we can be "long" or "short" on corn
- Positions are kept for everything traded, everyone we trade with, in every currency, on every market for every day in the future
  - Many client have portfolios which have collective positions
- Positions are usually kept neutral, holding a excess in anything presents a risk
  - Calculating the risk is not easy, it usually involves complex VaR (Monte Carlo) calculations
- A traders job is to manage risk by balancing the position

Money money money

- Someone makes money, someone else loses, why do we bother?
- Well we still have to buy and sell corn, wheat, oil, rubber, live-stock etc.
- Weather, disease and demand can effect prices
- Some people prefer to reduce risk, that's an opportunity for others to sell the risk on
- Risk itself is traded
- When the risk is mis-interpreted or someone calls in on a debt then things start go pear-shaped

On the floor at the CME...

Computer-based trading

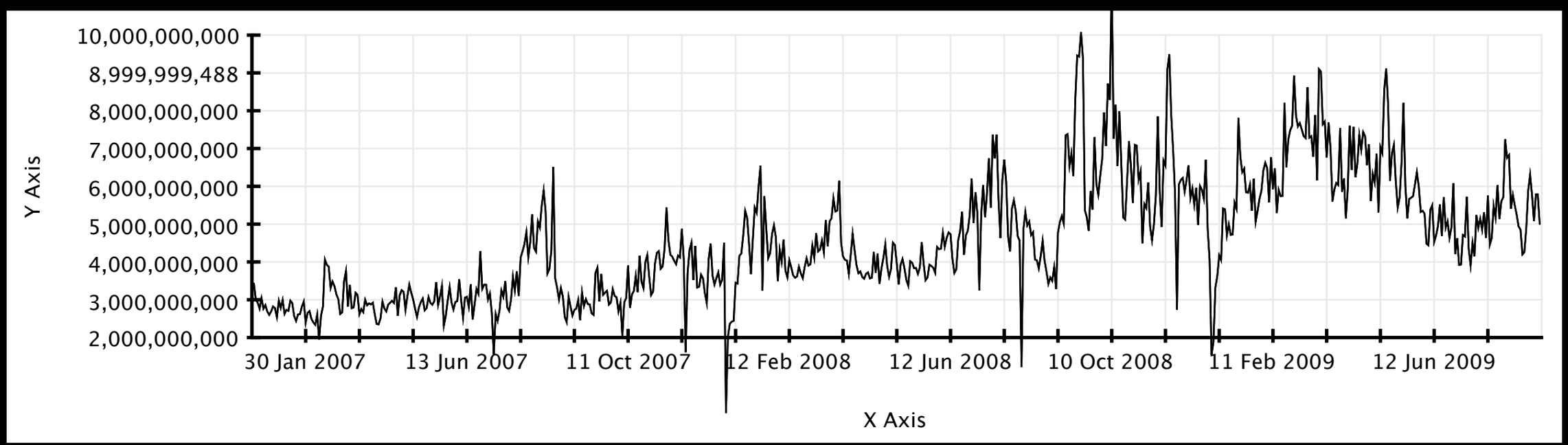
- Since the mid-80s the exchanges have become increasingly electronic
- Initially computers calculated positions and suggested hedges to cover risk
- Today computers make the same calculations and actually execute the trades automatically
- Once the trades are made, the positions are re-assessed and more corrections are made
- All this happens in a few milli-seconds

Volumes are going crazy

- While banking, the car industry, manufacturing, travel and many other industries have had a hard time over the last 2 years, technology has continued it's Moore-law progress
- Our laptops and servers have double the memory, double the disk space and typically doubled the number of cores
- And we need it, the data we're processing has more than doubled...

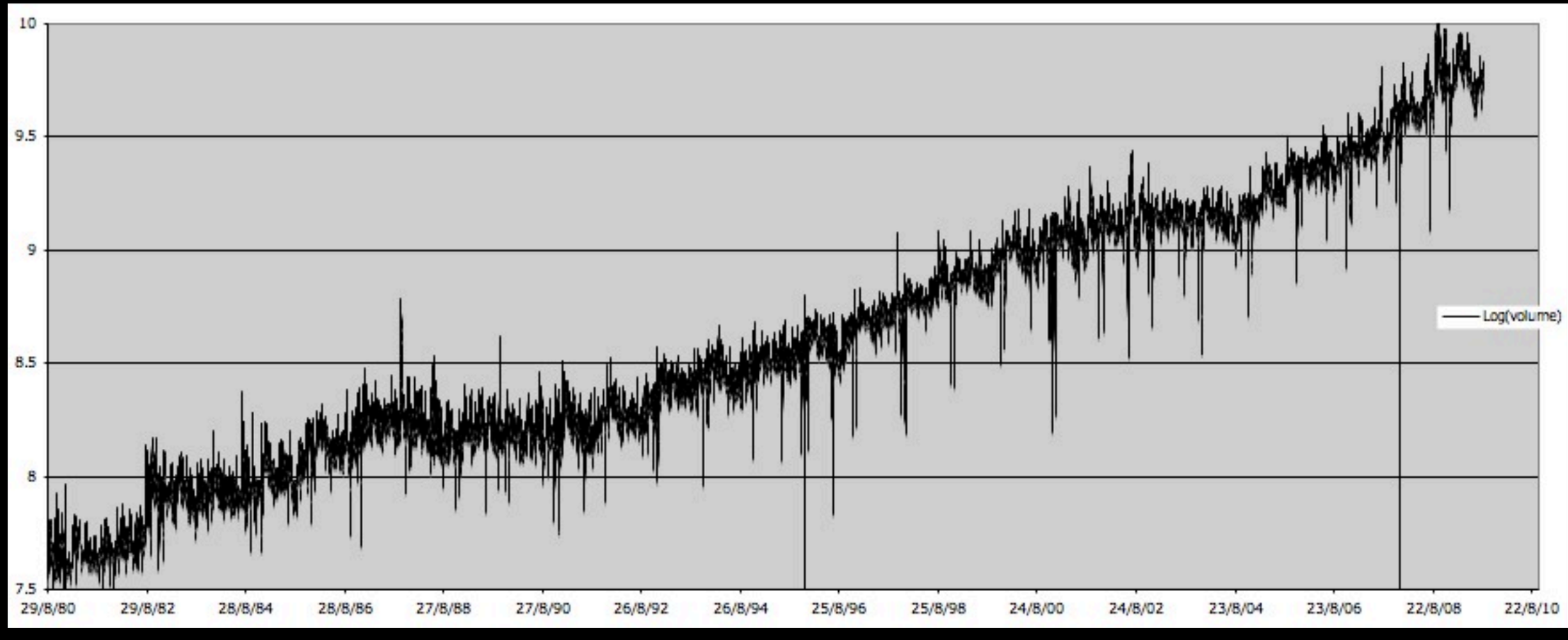
Volumes are going crazy

- While banking, the car industry, manufacturing, travel and many other industries have had a hard time over the last 2 years, technology has continued it's Moore-law progress
- Our laptops and servers have double the memory, double the disk space and typically doubled the number of cores
- And we need it, the data we're processing has more than doubled...



We need more CPU power

- Whether the the stock market is going up or down we need more and more "horse power" over time
- E.g. The Dow Jones daily trading volumes since 1980
  - Log(vol) vs time i.e. 8 = 100m, 9 = 1 billion, 10 = 10 billion (per day)



Moore's law to the rescue?

- Do the calculations and the volumes are not necessarily doubling every 18 months
- But add new financial products, increasing performance expectations, new markets and we exceed with Mr. Gordon Moore's transistor expectations by some margin
- Just like the increasingly large volumes of data in the music and film industry we're constantly on the leading edge
- My point is that the workload and challenges are not getting easier, we need better architecture and innovative solutions

Algorithmic Trading

- It's not that complicated... (in theory anyway)
  - You need to have access to data from the exchange
  - You come up with a trading strategy (algorithm)
  - You write the code (usually in C, Python, R or something similar)
  - You deploy it to a machine as close as you can physically get to the exchange (co-hosting)
  - ... and collect your money :-)

- The algorithm is basically a program that says something like
  - If MSFT < opening price and APPL > 30min moving avg and APPL > opening price then buy APPL, sell MSFT
- But they can get pretty complex
  - Predictive models based on market data feed harmonics trying anticipate where the market will be in 200 µs
  - Correlation trading of stocks that appear unrelated, but mathematically correlate

- The quicker you execute the trade the better your chances

Connecting to the Exchange

- Of the top 20 or so exchanges there are about 10 different protocols to connect
  - FIX, ITCH, OUCH, PITCH, SCRATCH, many are optimised for performance
  - Proprietary APIs can reduce latency to the order of 20µs
- FIX / FAST is a good standard approach, it comes in several versions and the latest can use 3 different encodings...
  - Standard FIX (tag/value pairs, all tags are integers)
  - FixML (a very verbose XMLized version of the above)
  - FAST (FIX Adapted for STreaming, like the standard version but compressed)
- When you've worked that out there's the venue specific dialect that need tuning for each exchange
- Finally certification (anything from weeks to months)

Java, C++ or .NET?

- Since the FIX engine is connecting to your back-end servers you usually have to make a choice between...

- C / C++
  - Still regarded as the fastest FIX engines and usually the choice for the arbitrage traders
  - Latency is reliable (i.e. no garbage collection)
  - Supporting all different versions of Linux, UNIX, MS, 32bit, 64bit etc. is a real pain

- Java
  - Surprisingly only 3rd place, Java FIX engines are very fast but unless they are carefully designed garbage collection can be a major issue
  - Easiest to integrate into other architecture, most flexible

- .NET
  - The most popular simply due to the reason that most small businesses start off on
  - Microsoft platforms (Excel etc.) and this fits in best
  - Less used in the larger businesses (banks, major firms etc.)

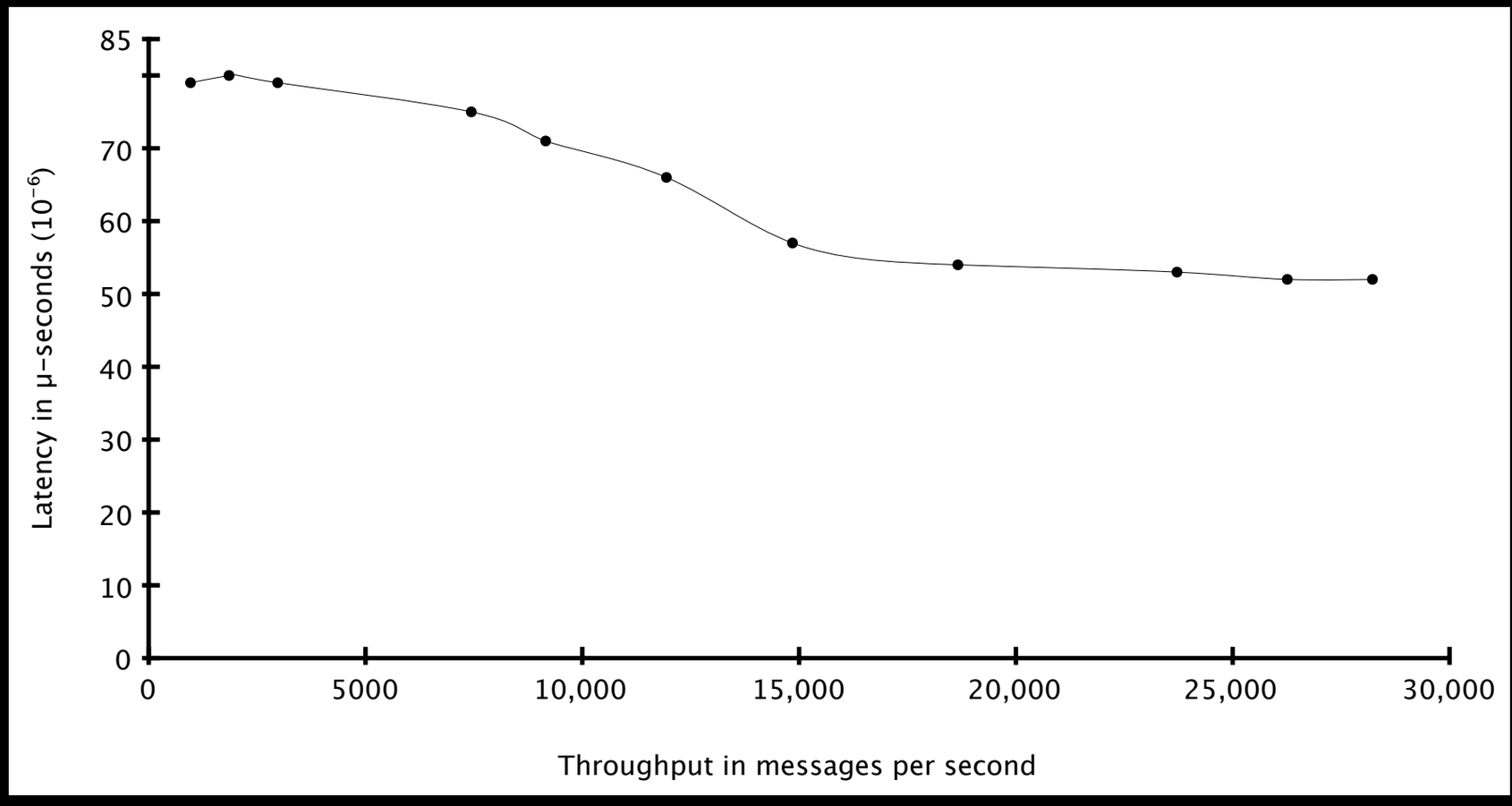
It's not JMS but it's pretty simple

- A small snippet of sample code to connect to a FIX engine (in an exchange) and send an order

```
final PropertyBasedSettings settings = new PropertyBasedSettings("simple/SimpleBuySide.properties");
final Engine engine = Engine.init(settings);

final Version fixVersion = Version.getByNumber(settings.getString("FIXVersion"));
Session sn = new Session(settings.getString("SenderCompID"),
    settings.getString("TargetCompID"), fixVersion);

sn.setSenderSubID("SenderSubID (50) field");

sn.setInboundApplicationMessageListener(this);
sn.setStateChangeListener(this);

sn.logonAsInitiator(settings.getString("CounterpartyHost"), settings.getInteger("CounterpartyPort"));

Message order = new Message("D", fixVersion);
order.set(Tag.HandlInst, "1");
order.set(Tag.ClOrdID, "Unique identifier for Order");
order.set(Tag.Symbol, "IBM");
order.set(Tag.Side, "1");
order.set(Tag.OrderQty, 1000);
order.set(Tag.OrdType, "1");

sn.send(order);
```

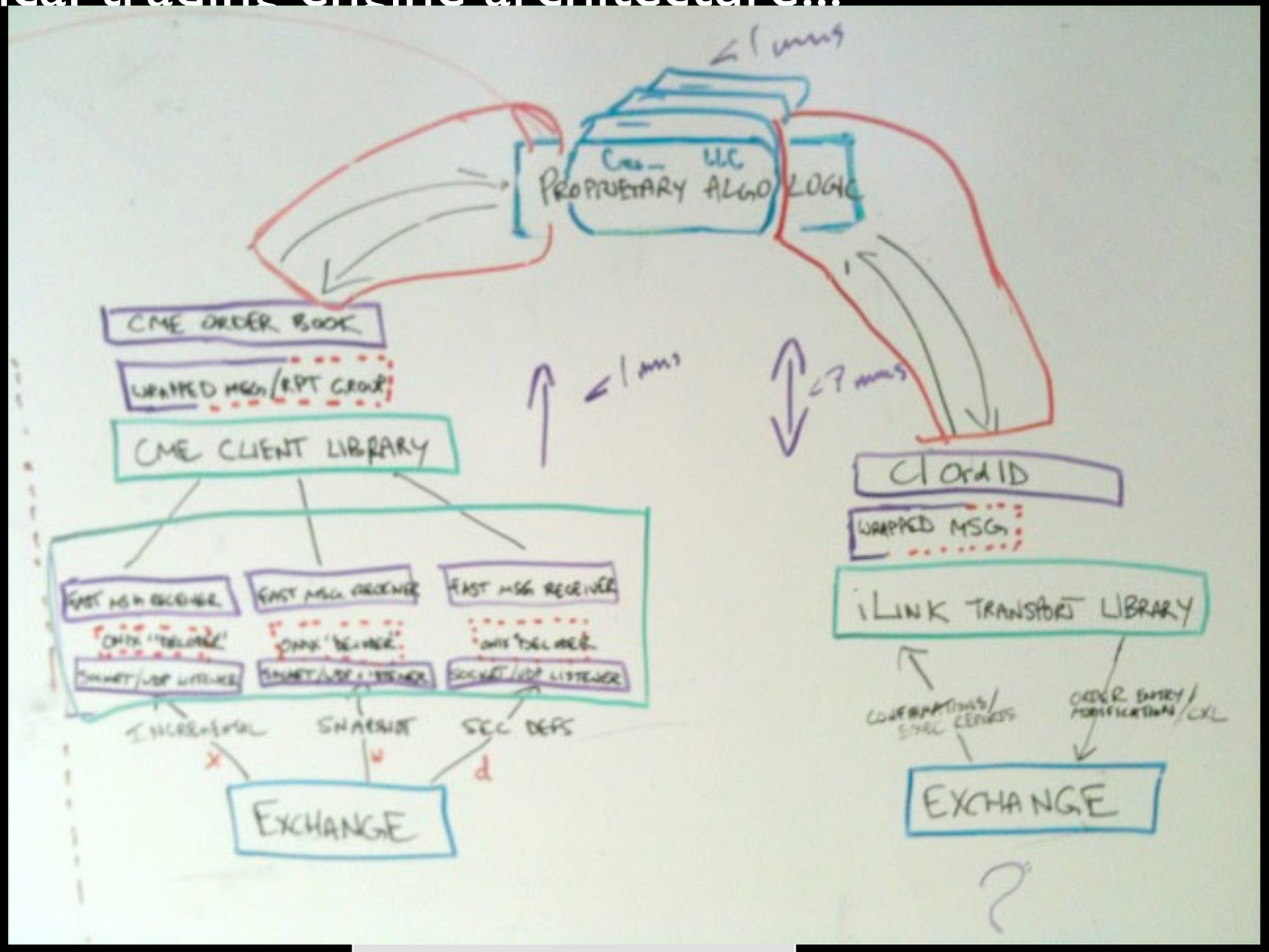
Latency is critical

- Typical latency for "processing" a FIX message is under 60µS
- The processed information has to be entered into the virtual order book and new positions calculated



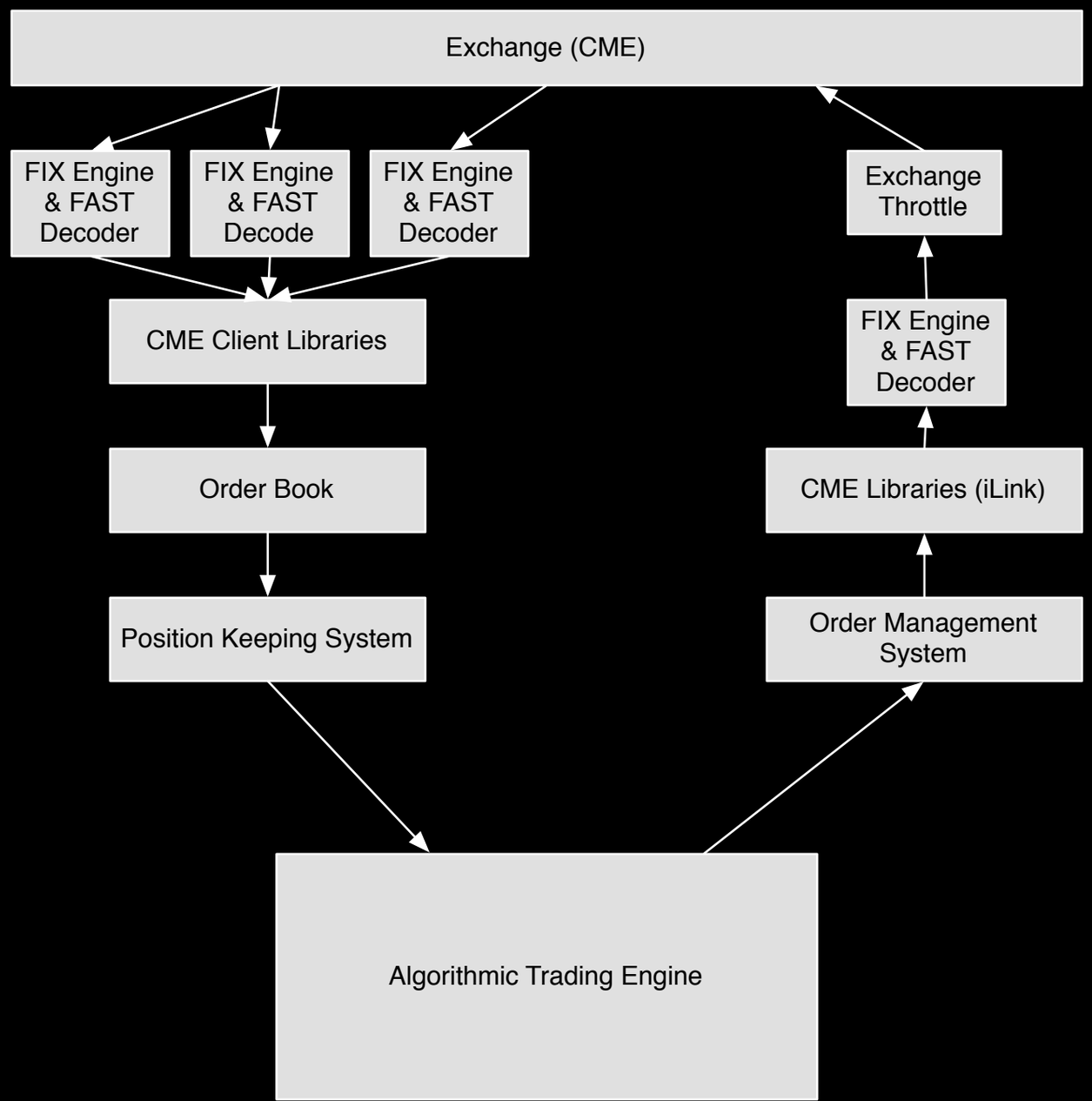
Algo-trading architecture

- A typical trading engine architecture...



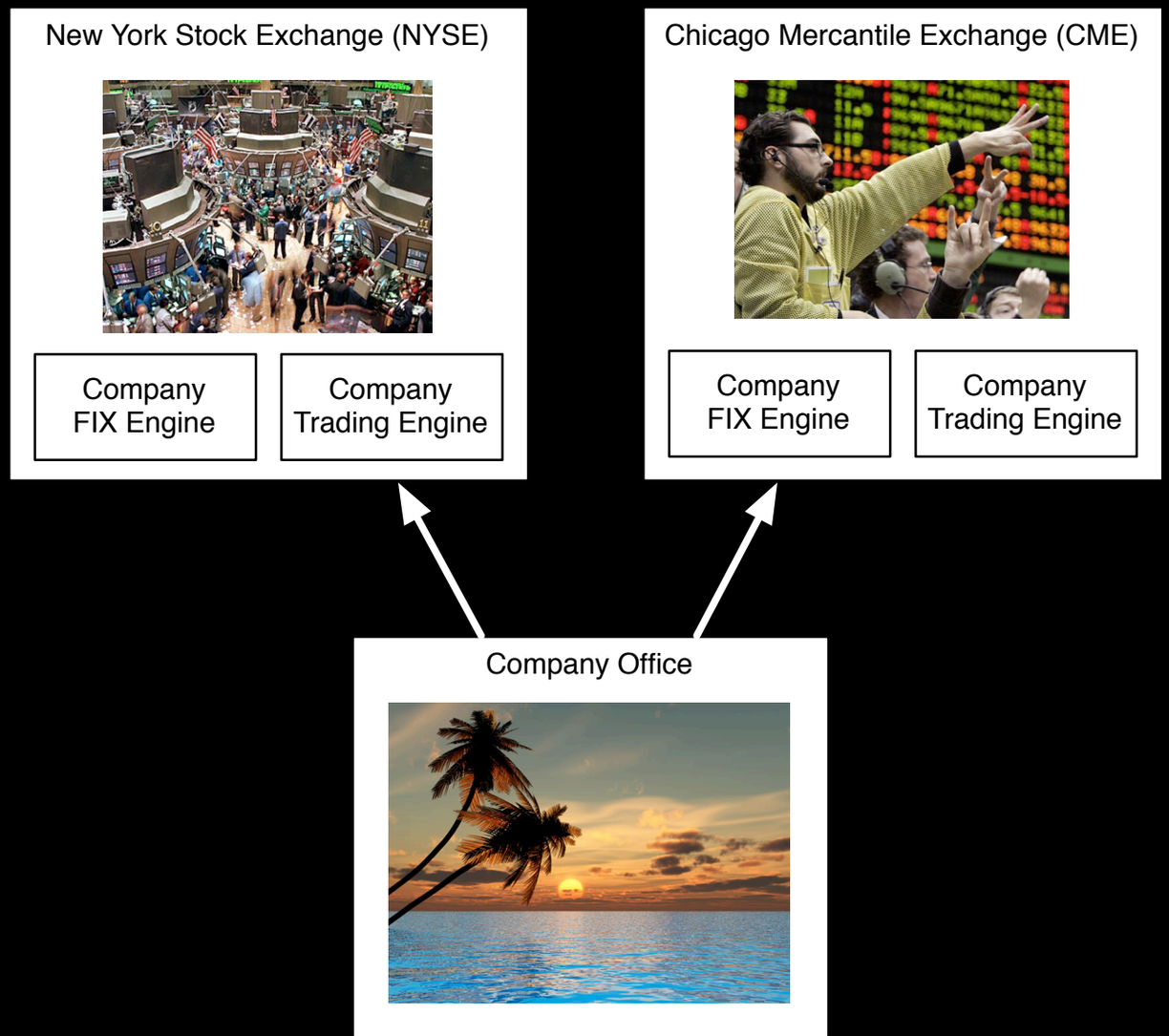
Algo-trading architecture

- A typical trading engine architecture...



The Enterprise Architecture

- It's not exactly complicate from an enterprise view point
  - NB: Not every exchange uses FIX so connectivity is not universally FIX
- Offices are often in "nice" locations
  - Of course it's nothing to do with the favourable tax
- Most of the infrastructure runs in co-located boxes
  - Co-hosting costs money and limits what you can do



Storing "tick" data

- Tick data is the raw price feed from the exchange
  - I.e. MSFT BID=24.77 ASK=24.79

- 8=FIXT.1.1^A9=0^A35=X^A49=CME^A34=2127825^A52=20100120150049656^A1128=8^A268=1^A279=0^A269=0^A48=109291^A22=8^A270=115060^A271=1^A273=150049000^A336=2^A346=1^A83=27750^A1023=2^A75=20081117^A10=000^A

- This needs to be stored for legal reasons as you have to be able to demonstrate "best execution" for clients
- We've seen the volumes earlier, we don't need everything on the exchange but certainly everything we're trading
- Typically these are tens of gigabytes per day per exchange

- An interesting solution is to ship it up to EC2 and store it on EBS and S3

Shipping it up to EC2

- There are two mechanisms to get data up onto Amazon's EC2
  - Streaming and batch

- Batch is more efficient, we compress the data hourly and "scp" it up to an EC2 box

```
-rw-r--r--  archive  onix  12710903  7 Dec  2008 OrderBooksRepository_Channel_7_summary_20081125-14_00_54_154.gz
-rw-r--r--  archive  onix  19452739  7 Dec  2008 OrderBooksRepository_Channel_7_summary_20081125-15_00_54_785.gz
-rw-r--r--  archive  onix  27549005  7 Dec  2008 OrderBooksRepository_Channel_7_summary_20081125-16_00_55_417.gz
```

- Streaming adds the interesting advantage of having near-realtime data on EC2
  - Not to mention an EC2 box doing little more than writing to disk

- So we started to add monitoring processes to the data
  - Triggers, statistics, filters, aggregators etc.

Terabytes of data

- Several 10s of Gigabytes per exchange results in 10s, even 100s of Terabytes per year (and increasing)
- Fortunately the data compresses at well over 10:1
  - Incidentally Amazon charges $150/month per TB
- To retrieve data we simply fire up a machine instance, mount the drive, de-compress and filter out what we need
- Retrieval time can be as low as 10 seconds for simple requests, longer if more machines are needed

- With access to so much data we can also run a FIX engine and emulate an exchange from EC2 based on any date in the past (data permitting)

Back-testing

- Being able to emulate an exchange or download samples of historic data is vital to being able to back-test algorithms
- Traders can test "what-if" scenarios on data they wouldn't normally work with

- This lead us to think about moving the algo engine into Amazon EC2
- This way we can fire up more engines and test closer to the data

- What if the algo trading engines were to work on the near-realtime data being streamed up from the exchange though?

Trading from the cloud

- Most trading algorithms are latency critical which excludes them from the cloud but a few interesting options remain...

1. Non latency-critical trading algorithms can run in the cloud, what's more, we can run lots of them
   - Cooperation between "what-if" scenarios being run in the cloud and latency-critical algos run on co-hosted boxes can improve trading strategies

2. Since we can create a small exchange in the cloud ultimately this may become more common place
   - The cloud will then be the only place to run trading platforms

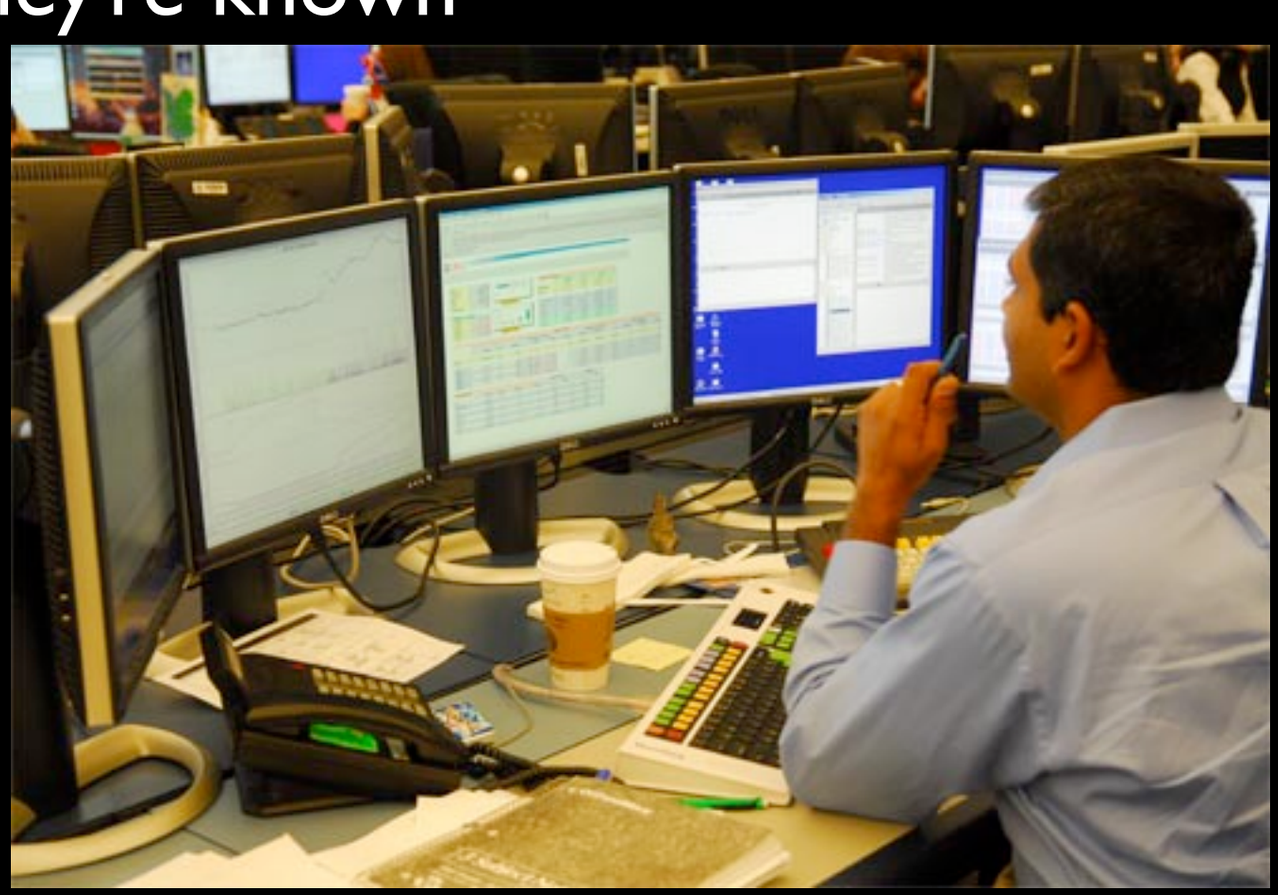
3. Many "what if" scenarios, too complex to run locally can run in the background - in the cloud

Seeing what's going on

- Letting computers' decide what to trade is good but it's not good to leave them unattended

- It's not the computers that are the problem it's the programmers - Quants as they're known

- Left to their own devices some algorithms can dump stock at an incredible rate taking down not just exchanges but also entire companies - in seconds



Getting away from the screens

- The following quote explains nicely how important it is to be able to see what's going on...

  "... personally I carry a laptop with me even to bed, but most of our clients would be very interested in being able to monitor their individual accounts"

- So why not provide most of what people already have on a mobile device?

Getting away from the screens

- The following quote explains nicely how important it is to be able to see what's going on...

  "... personally I carry a laptop with me even to bed, but most of our clients would be very interested in being able to monitor their individual accounts"

- So why not provide most of what people already have on a mobile device?

Thank you

John.Davies@Onixs.biz

Twitter: @jtdavies