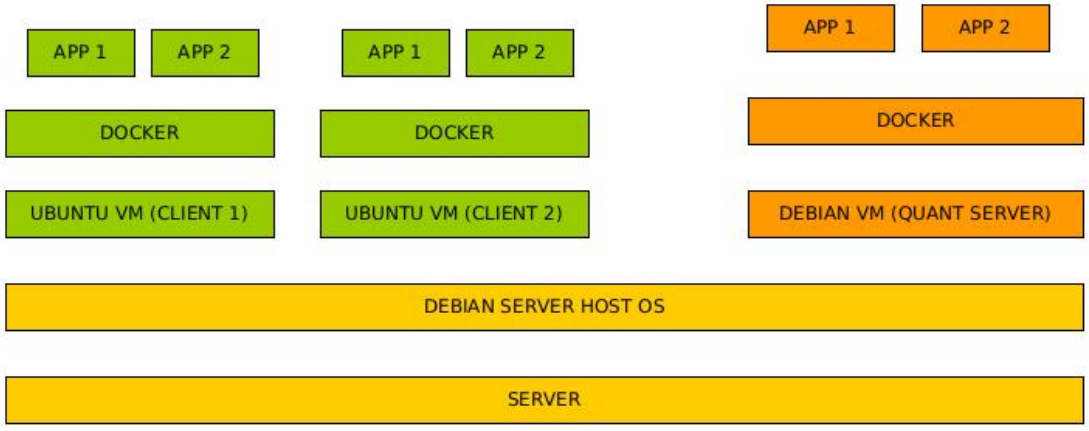
portfolio

April 23, 2022

Swapnil Sonawane
Quantitative Trader South Delhi, Delhi, India

https://www.linkedin.com/in/swapnil-sonawane-53763116/

1 VIRTUALIZATION



I designed the quant trading infra from the scratch using open source software at a prop desk.

On premise custom built desktop.

- CPU : AMD RYZEN 5950x 32 Threads @ 4.9Ghz max
- MOBO : Asus Prime X570-p
- RAM : 64GB DDR4 RAM @ 3200Mhz
- HDD : 2 X 4TB
- Cooling : Deep Cool L360RGB Liquid cooler
- Power : Deep Cool 850W Gold SMPS

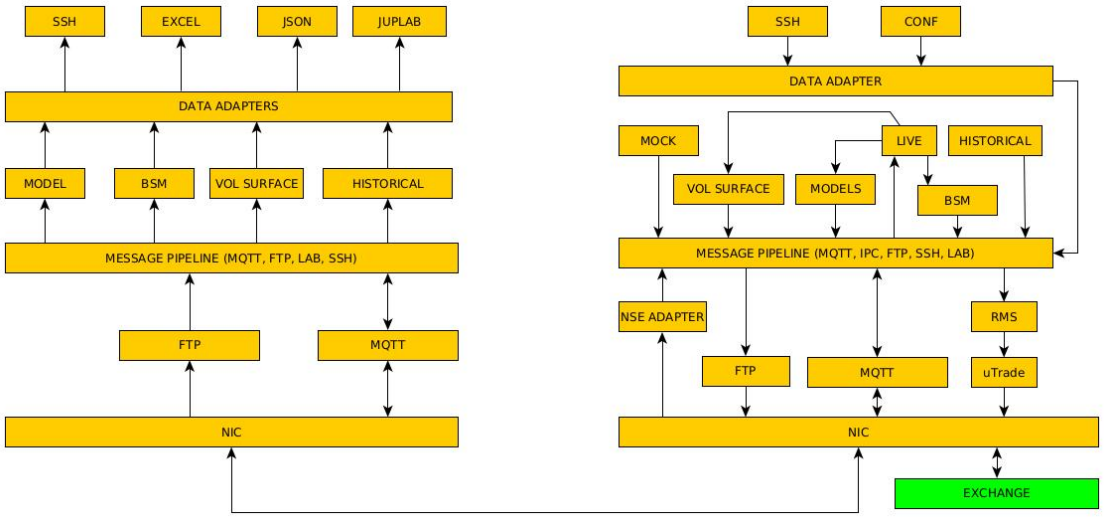
The CPU uses debian server as host operating system and KVM for virutalization. It is designed to be a portable quant research lab.

QUANT CLIENT VM (green) are accessed by traders for model building and monitoring production environment.

QUANT SERVER VM (orange) acts as the central brain of the operation. All data crunching, algo products, market data feed and order execution systems sit here.

2 INFRASTRUCTURE

LEFT QUANT CLIENT VM <=====> RIGHT QUANT SERVER VM



- NIC : Network Interface Card
- Message Pipeline : Communication protocols available
- Data Adapter : User Interface to the system
- Exchange : Market data and order execution venue
- MQTT : Lightweight messaging protocol
- FTP : File Transfer Protocol
- RMS : Firmwide Risk Management System
- uTrade : Low Latency Order execution API
- NSE Adapter : Reads market data from exchange
- Live : Market data broker
- BSM : Black Scholes Model to price derivatives
- Historical : Historical market data archives
- Vol Surface : Volatility surface for pricing options
- Mock : Mock trading or paper trading service
- Models : In-production Fully automated algo trading models
- SSH : Used to access remote shell
- Conf : Configuration file for different services and job schedulers
- Juplab : Remotely hosted Jupyter Notebooks for research
- Excel : Excel sheets for use as live dashboard

3 OPTION STRUCTURING ENGINE

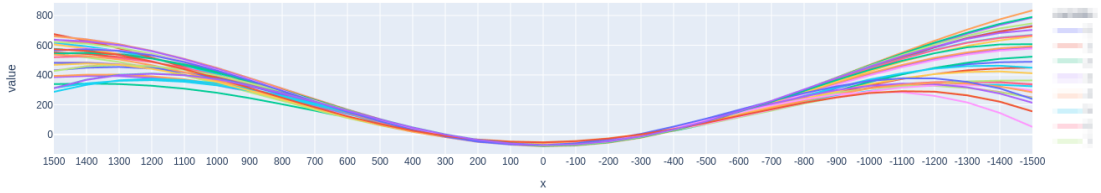
The firm trades a lot of options strats and needs option valuation tools and pricing these baskets as well as simulate these baskets to different market vol regimes

TIMESTAMP 2022-03-06 10:55:17 POS 0

LEVEL	P2	BUY	SELL	Forcast+	Forcast-	IV_chng	IV
500	15200	96.700	95.225	95.5	95.1	5	42.2
450	15250	94.825	92.300	92.7	91.9	5	41.8
400	15300	92.450	91.775	92.5	90.9	5	41.4
350	15350	91.725	90.425	91.4	89.1	5	41.1
300	15400	88.475	87.150	88.4	85.5	5	40.8
250	15450	89.400	87.200	88.8	85.0	5	40.4
200	15500	90.750	89.725	91.6	87.0	5	40.2
150	15550	90.225	88.775	90.9	85.8	5	39.9
100	15600	90.425	88.850	91.1	85.6	5	39.7
50	15650	88.525	86.375	88.6	83.3	5	39.4
0	15700	84.850	82.900	84.9	80.1	5	39.3
-50	15750	86.375	84.550	86.4	82.1	5	39.1
-100	15800	85.075	83.700	85.1	82.0	5	39.0
-150	15850	89.225	86.800	87.9	85.6	5	38.9
-200	15900	92.125	89.425	90.2	88.7	5	38.9
-250	15950	92.100	89.650	90.0	89.6	5	38.8
-300	16000	94.725	92.650	92.8	93.0	5	38.7
-350	16050	96.125	92.225	91.9	93.2	5	38.8
-400	16100	96.875	93.975	93.4	95.3	5	38.9
-450	16150	100.500	93.125	92.2	94.8	5	38.9
-500	16200	103.375	96.600	95.6	98.3	5	39.0

C2=P2=Ratio 16400 = 15700 = 1 -1.5 0.5
C2P2-Cgap-Pgap 700 - 200 - 200

STRDL 465.05 IV 39.15 STRL 36.55 BNF 15856.92 CHNG 13.02



Each color represents the payoff of a multi-leg option basket designed to capture implied and realised variance of the underlying under different market volatility regimes

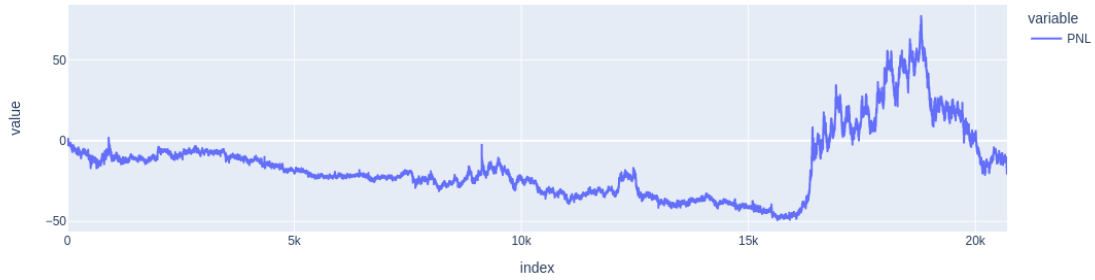
The derivatives structuring engine is used to automatically price and compare payoffs available for different multileg option baskets at different strikes where 0 represents ATM.

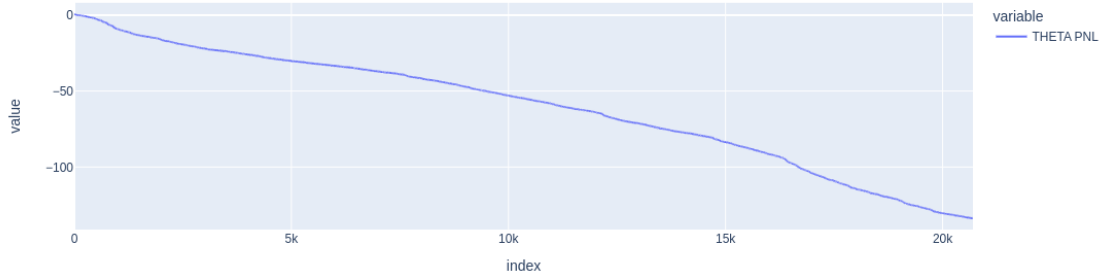
4 GREEK PNL ENGINE

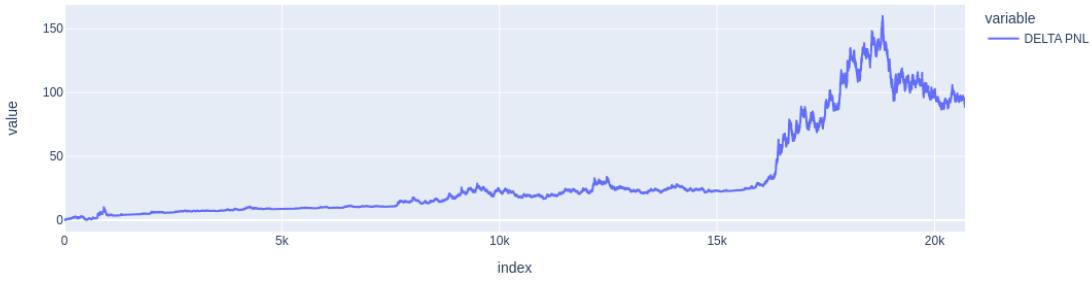
Allows you to understand the PnL performance of the option basket in live markets and historical data.

Tool based on the famous Option Profit and Loss Attribution and Pricing: A New Framework by Peter Carr.

Visualize the options payoff over time based on separate greek exposure.







5 REALTIME GREEK SENSITIVY CALCULATIONS

```
%%time
for i in range(100000):
    option_type = 'put'
    strike = 17200
    underlying_price = 17215.75
    days_to_expiration = 7
    price = 218.60
    dividend = 0
    output = KK_BSM(option_type, strike, underlying_price, days_to_expiration, price, dividend)
```

```
CPU times: user 522 ms, sys: 27 µs, total: 522 ms
Wall time: 522 ms
```

```
output
```

```
{'iv': 0.23820553489073212,
 'delta': -0.4822641239625843,
 'vega': 950.1875004928472,
 'theta': -16.158182736318853,
 'rho': -163.420267559724,
 'gamma': 0.000701779042076819,
 'vanna': -0.01919905809812191,
 'charm': 0.0003139697995495073,
 'speed': -2.622120244966921e-10,
 'zomma': -8.067408416886312e-06,
 'color': -5.0102493144343946e-05,
 'veta': -0.6790381806889784,
 'vomma': 2.0351931456004517,
 'ultima': -0.12013691707887045,
 'dual_delta': 0.49541858521012677}
```

6 MARKET DECODING ADAPTER

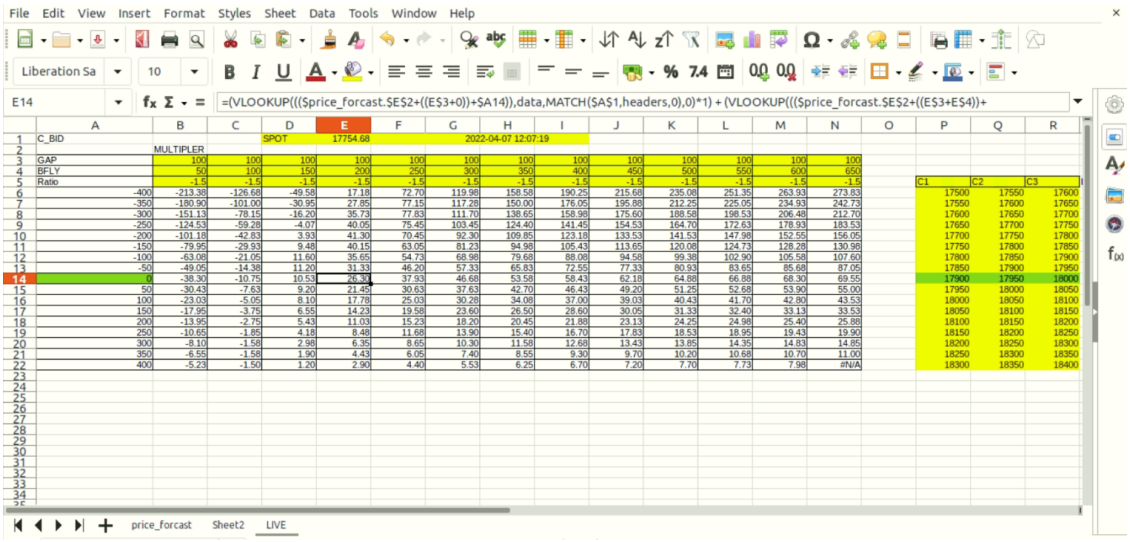
Used to read byte encoded market data from NSE

```
// Structures for decoding trimmed data from exchange
struct BcastPackData
{
    char cPackData [512];
    short iNoPackets;
    short iCompLen;
    char cCompData[512];
    short CompressionLen;
    char BroadcastData [8192];
} packed_data;

struct BCAST_HEADER
{
    long LogTime;
    short TransactionCode;
    short MessageLength;
    char BCAST_HEADER[40];
} bcast_header;

struct BCAST_ONLY_MBP
{
    char BCAST_ONLY_MBP[470];
    short NoOfRecords;
} bcast_mbp;

struct INTERACTIVE_ONLY_MBP_DATA
{
    char INTERACTIVE_ONLY_MBP_DATA[213];
    long Token;
    short BookType;
    short TradingStatus;
    long LastTradedPrice;
    long VolumeTradedToday;
    long Quantity[2];
    long Price[2];
} i_mbp[2];
//
```

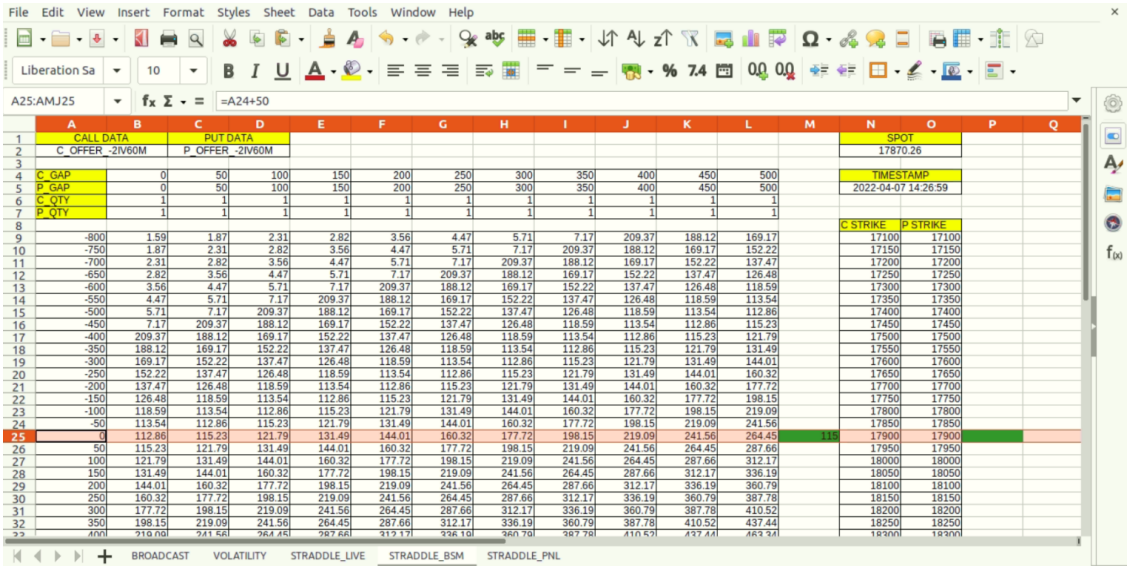
7 BUILDING EXCEL FOR LIVE TRADING

Built excel tools for windows users.

Live excel pricing for trading complex multileg option baskets.

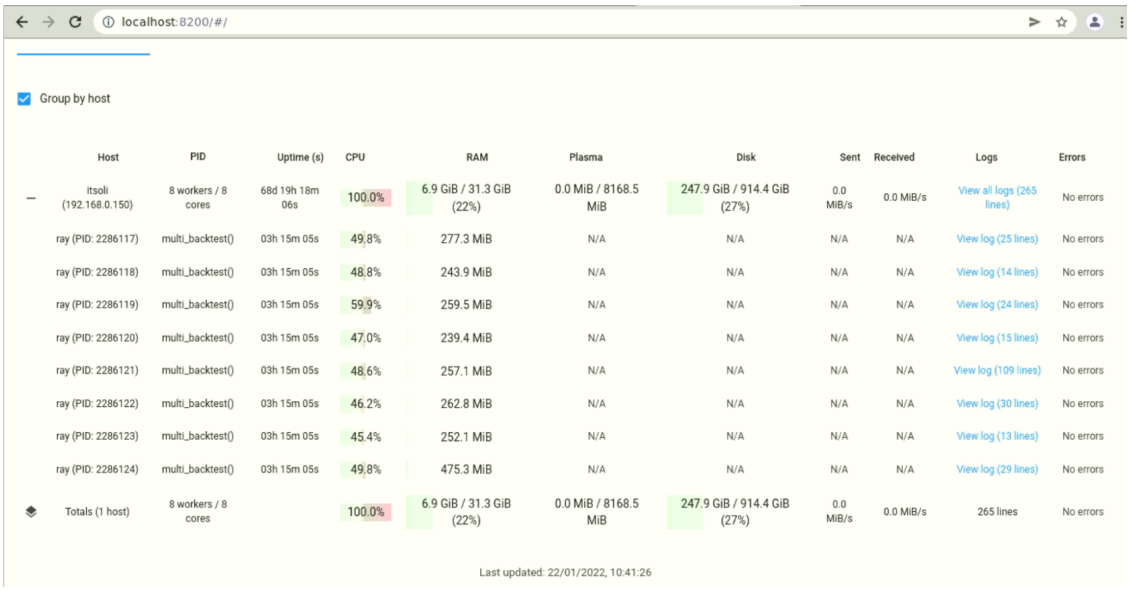


Option strategy simulation inside excel



8 MULTITHREAD BACKTESTING

Expertise in quant research and algo strat development and model validation


localhost:8200/#/
Group by host
Host PID Uptime (s) CPU RAM Plasma Disk Sent Received Logs Errors
itsoli (192.168.0.150) 8 workers / 8 cores 68d 19h 18m 06s 100.0% 6.9 GiB / 31.3 GiB (22%) 0.0 MiB / 8168.5 MiB 247.9 GiB / 914.4 GiB (27%) 0.0 MiB/s 0.0 MiB/s View all logs (265 lines) No errors
ray (PID: 2286117) multi_backtest() 03h 15m 05s 49.8% 277.3 MiB N/A N/A N/A N/A View log (25 lines) No errors
ray (PID: 2286118) multi_backtest() 03h 15m 05s 48.8% 243.9 MiB N/A N/A N/A N/A View log (14 lines) No errors
ray (PID: 2286119) multi_backtest() 03h 15m 05s 59.9% 259.5 MiB N/A N/A N/A N/A View log (24 lines) No errors
ray (PID: 2286120) multi_backtest() 03h 15m 05s 47.0% 239.4 MiB N/A N/A N/A N/A View log (15 lines) No errors
ray (PID: 2286121) multi_backtest() 03h 15m 05s 48.6% 257.1 MiB N/A N/A N/A N/A View log (109 lines) No errors
ray (PID: 2286122) multi_backtest() 03h 15m 05s 46.2% 262.8 MiB N/A N/A N/A N/A View log (30 lines) No errors
ray (PID: 2286123) multi_backtest() 03h 15m 05s 45.4% 252.1 MiB N/A N/A N/A N/A View log (13 lines) No errors
ray (PID: 2286124) multi_backtest() 03h 15m 05s 49.8% 475.3 MiB N/A N/A N/A N/A View log (29 lines) No errors
Totals (1 host) 8 workers / 8 cores 100.0% 6.9 GiB / 31.3 GiB (22%) 0.0 MiB / 8168.5 MiB 247.9 GiB / 914.4 GiB (27%) 0.0 MiB/s 0.0 MiB/s 265 lines No errors
Last updated: 22/01/2022, 10:41:26


File Edit View Run Kernel Tabs Settings Help
Launcher
study11.ipynb
study11-backtest.ipynb
Code
Python 3 (ipykernel)
value
index
variable
return
Daily_drawdown
Max_drawdown

[ ]: