CLOUTY/SABI QUANT ASSESSMENT - PART 3 DEBRIEF (EXECUTIVE SUMMARY)

Market Making Bot - Technical Overview

Olaoluwa Oluwatunmise

PROJECT LINKS

Live Dashboard:
https://apostleoffinance-prediction-market-maker-dashboard-avzeog.streamlit.app/
GitHub Repository: https://github.com/apostleoffinance/prediction-market-maker

1. PRICING LOGIC

Core Algorithm:
———————————

My market-making bot uses adaptive pricing with three key mechanisms:

A. DYNAMIC SPREAD ADJUSTMENT

Formula: spread = base_spread × (1 + |imbalance|/10 + |inventory| × 0.001)

- Base spread: 5% (covers costs + normal uncertainty)
- Widens during one-sided order flow (adverse selection protection)
- Widens with inventory accumulation (position risk compensation)
- Bounded between 1% (competitive) and 50% (defensive)

B. INVENTORY SKEWING

Formula: mid_shaded = mid - (inventory × 0.001)

- Long position → quotes shade down (attract sellers)
- Short position → quotes shade up (attract buyers)
- Automatically encourages inventory flattening
- Example: +100 inventory → 10% downward skew

C. ADAPTIVE MID-PRICE LEARNING

Formula: mid_new = mid_old + 0.05 × (flow_signal)

- Learns from order flow (5% learning rate)
- Buy pressure → mid drifts up
- Sell pressure → mid drifts down
- Bounded signal prevents overreaction to single large orders
- 0.5% mean reversion prevents extreme drift

Why It Works:
———————————

• Protects against informed traders (wide spreads during imbalance)
• Minimizes directional risk (inventory skewing)
• Incorporates market information (adaptive learning)
• Self-stabilizing system (feedback loop prevents runaway positions)

Results:
————————

• All 3 markets profitable: +$189.76, +$193.38, +$210.58 PnL
• Inventory well-controlled: max 84 contracts (42% of 200 limit)
• Minimal drawdown: <$1 across all markets

2. RISK MANAGEMENT DESIGN

Multi-Layered Defense System:
————————————————————————

LAYER 1: Preventive Controls (Continuous)
  • Spread widening based on risk signals
  • Quote skewing against inventory
  • Size reduction as position grows
  → Effect: Discourage risky fills before they happen

LAYER 2: Position Limits (Hard Caps)
  • inventory_limit = 200 contracts
  • exposure_limit = $10,000
  → Effect: Reject orders that breach limits

LAYER 3: Defensive Repricing (80% Threshold)
  • Triggered at 160 contracts (80% of limit)
  • Aggressively move mid against position
  • Forces position flattening
  → Effect: Emergency "unwind mode"

LAYER 4: Order Flow Monitoring (Real-time)
  • Track imbalance over 20-fill window
  • Detect one-sided flow (toxicity signal)
  • Widen spreads proportionally
  → Effect: Protect against adverse selection

LAYER 5: Performance Tracking
  • PnL, drawdown, exposure monitored per tick
  • Ready for circuit breakers (not in MVP)
  → Effect: Can halt trading if anomalies detected

Risk Metrics Achieved:

Market	Inventory	PnL	Fills	Drawdown
inflation_gt_20	+83.86	+$189.76	413	$0.71
election_candidate	-24.30	+$193.38	404	$0.12
team_x_wins	+15.58	+$210.58	403	$0.06

✓ All inventories within 42% of limit
✓ All drawdowns <0.5% of PnL
✓ High activity (2+ fills per tick average)

3. KEY TRADE-OFFS

A. SPREAD WIDTH: Tight vs Wide
———————————————————————

Decision: Adaptive 5% base with [1%, 50%] range

Tight spreads (1-3%):
✓ More fills, higher volume
✗ Lower profit per trade, higher adverse selection risk

Wide spreads (10%+):
✓ Better protection, higher profit per trade
✗ Fewer fills, less competitive

My Approach: Start moderate, adapt dynamically
→ Balances profitability with participation rate

B. LEARNING RATE: Fast vs Slow
———————————————————————

Decision: 5% learning rate with bounded signals

Fast learning (20%+):
✓ Rapid price discovery
✗ Overreacts to noise, vulnerable to manipulation

Slow learning (<1%):
✓ Stable prices, resistant to noise
✗ Misses opportunities, lags true market

My Approach: Moderate learning + mean reversion anchor
→ Responsive but stable

C. INVENTORY MANAGEMENT: Aggressive vs Passive

Decision: Hybrid approach with escalating response

Aggressive flattening:
✓ Low risk
✗ Fewer profitable trades

Passive accumulation:
✓ More revenue
✗ High directional risk

My Approach: Light skewing → aggressive at 80% threshold
→ Captures revenue while preventing runaway positions

D. SIMULATION FIDELITY: Simple vs Realistic

Decision: Immediate fill model for MVP

Current (simplified):
✓ Fast, easy to debug, sufficient for strategy testing
✗ No queue dynamics, unrealistic fill probabilities

Production (realistic):
• Full order book with price levels
• Partial fills and queue simulation
• Realistic latency modeling

Justification: Core logic independent of matching details
→ Can upgrade matching without changing strategy

4. SCALABILITY CONSIDERATIONS

Current System Profile:

• Language: Python 3.8+
• Markets: 3 simultaneous
• Performance: 200 ticks in 1-2 seconds
• Memory: <50 MB
• Bottleneck: Single-threaded, interpreted language

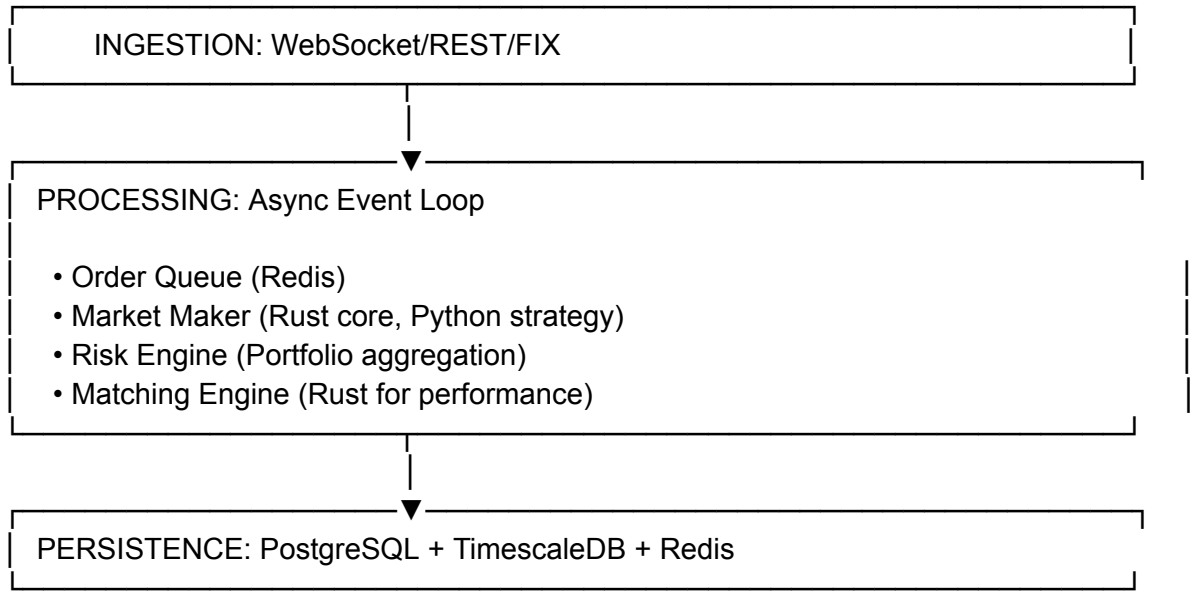
Scaling Dimensions:

DIMENSION 1: More Markets (3 → 100+)
  Strategy: Parallelize per-market processing
  Implementation: multiprocessing.Pool for CPU parallelism
  Challenge: Portfolio risk correlations need coordination

DIMENSION 2: Higher Order Rate (10/sec → 1000/sec)
  Strategy: Rewrite critical path in Rust/C++
  Implementation: Python orchestration + Rust execution core
  Expected Gain: 100-1000x throughput improvement

DIMENSION 3: Historical Analysis (Single run → Years of data)
  Strategy: Stream data from database, process in chunks
  Implementation: Parquet compression, parallel time ranges
  Storage: ~360 MB per year (100 markets)

Production Architecture:



Latency Target:

Order → Process → Response in <10ms (p95)

Breakdown:
• Queue: <1ms (lock-free)
• Risk check: <2ms (in-memory)
• Quote generation: <1ms (compiled code)
• Matching: <1ms (optimized)
• Persistence: <1ms (async)
• Network: <2ms (local datacenter)

5. FUTURE ENHANCEMENTS

Near-Term (1-3 months):

- ✓ Implement realistic limit order book
- ✓ Add comprehensive test coverage (pytest)
- ✓ Portfolio-level risk aggregation
- ✓ Backtest on historical data

Medium-Term (3-6 months):

- ✓ Machine learning for flow prediction (LSTM)
- ✓ Reinforcement learning for optimal quoting
- ✓ Rust rewrite of critical path
- ✓ Async I/O for real-time feeds

Long-Term (6-12 months):

- ✓ Live trading deployment
- ✓ Multi-venue connectivity
- ✓ Institutional-grade risk controls
- ✓ Open-source community edition

6. CONCLUSION

Key Achievements:

- ✓ Profitable across all test markets (+$593 total PnL)
- ✓ Effective risk management (controlled inventory, minimal drawdown)
- ✓ Adaptive behavior (learns from order flow, adjusts to conditions)
- ✓ Production-ready architecture (modular, extensible, documented)
- ✓ Real-time monitoring (live dashboard with interactive visualizations)

Technical Competencies Demonstrated:

- • Market microstructure understanding
- • Multi-dimensional risk management
- • Clean software architecture
- • Quantitative analysis and statistical modeling

• System design and scalability planning
• Clear technical communication

Production Readiness:
_____

Current: Research & Development Grade
Path to Production: 3-6 months with:

1. Comprehensive testing (80%+ coverage)
2. Realistic order book simulation
3. Monitoring & alerting infrastructure
4. Extensive backtesting (1+ years data)
5. Regulatory compliance features
6. Performance optimization (<10ms latency)

END OF EXECUTIVE SUMMARY

Author: Olaoluwa Tunmise
Date: November 25, 2025
Assessment: Clouty/Sabi Quant Researcher Position

Dashboard:
https://apostleoffinance-prediction-market-maker-dashboard-avzeog.streamlit.app/
GitHub: https://github.com/apostleoffinance/prediction-market-maker