Objective

Demonstrate advanced skills in HFT strategy development, system design, and quantitative analysis by expanding on the initial market-making strategy and implementing sophisticated features.

Data Source

Continue using the sample tick data provided at https://reach.stratosphere.capital/data/usdm/ for your implementation.

Task Description

Building on your initial implementation, enhance your HFT system with the following components:

1. Multi-Asset Strategy:

  - Extend your strategy to handle at least two correlated trading pairs from the provided data source.
  - Implement cross-asset signals or arbitrage opportunities.

2. Machine Learning Integration:

  - Implement and train an XGBoost model for ultra-short-term price movement prediction.
  - Integrate the model into your market-making strategy.

3. Advanced Latency Modeling:

  - Implement a custom adaptive latency model that adjusts based on market conditions.
  - Analyze how different latency scenarios affect your strategy's performance.

4. Sophisticated Order Execution:

  - Implement smart order routing across multiple simulated venues.
  - Develop a custom order fill probability model and integrate it with HftBacktest.

5. Risk Management:

  - Implement a real-time risk management module with VaR calculations.
  - Develop dynamic position and order size adjustments based on market volatility.

6. Market Regime Detection:

  - Implement a market regime detection algorithm.
  - Adapt your market-making strategy to different identified regimes.

7. Performance Enhancements:

  - Optimize more components in Rust or C++ for ultra-low latency.
  - Implement parallel processing for multi-asset handling.

Extended Written Component

Expand your report (additional 3-4 pages) to include:
1. Detailed explanation of your enhanced multi-asset strategy.
2. Analysis of the machine learning model's effectiveness in the HFT context.
3. Discussion of your advanced latency modeling and its impacts.
4. Explanation of your risk management approach and its effectiveness.
5. Analysis of market regime detection and strategy adaptation.
6. Comprehensive performance analysis, including HFT-specific metrics.
7. Discussion on potential real-world deployment challenges and solutions.

Additional Requirements

1. Extensive use of Rust or C++ for performance-critical components.
2. Implementation of unit and integration tests.
3. Design document outlining system architecture and data flow.

Time Allocation

1-2 days (in addition to the initial task)

Additional Evaluation Criteria

1. Sophistication and innovation in HFT strategy design
2. Effectiveness of machine learning integration in HFT context
3. Quality of system design and architecture
4. Depth of quantitative analysis and performance optimization
5. Consideration of real-world HFT challenges and solutions

Submission

1. Updated GitHub repository with your enhanced code
2. Extended written report (PDF format)
3. System architecture and data flow design document
4. Updated README file with instructions for running your advanced implementation