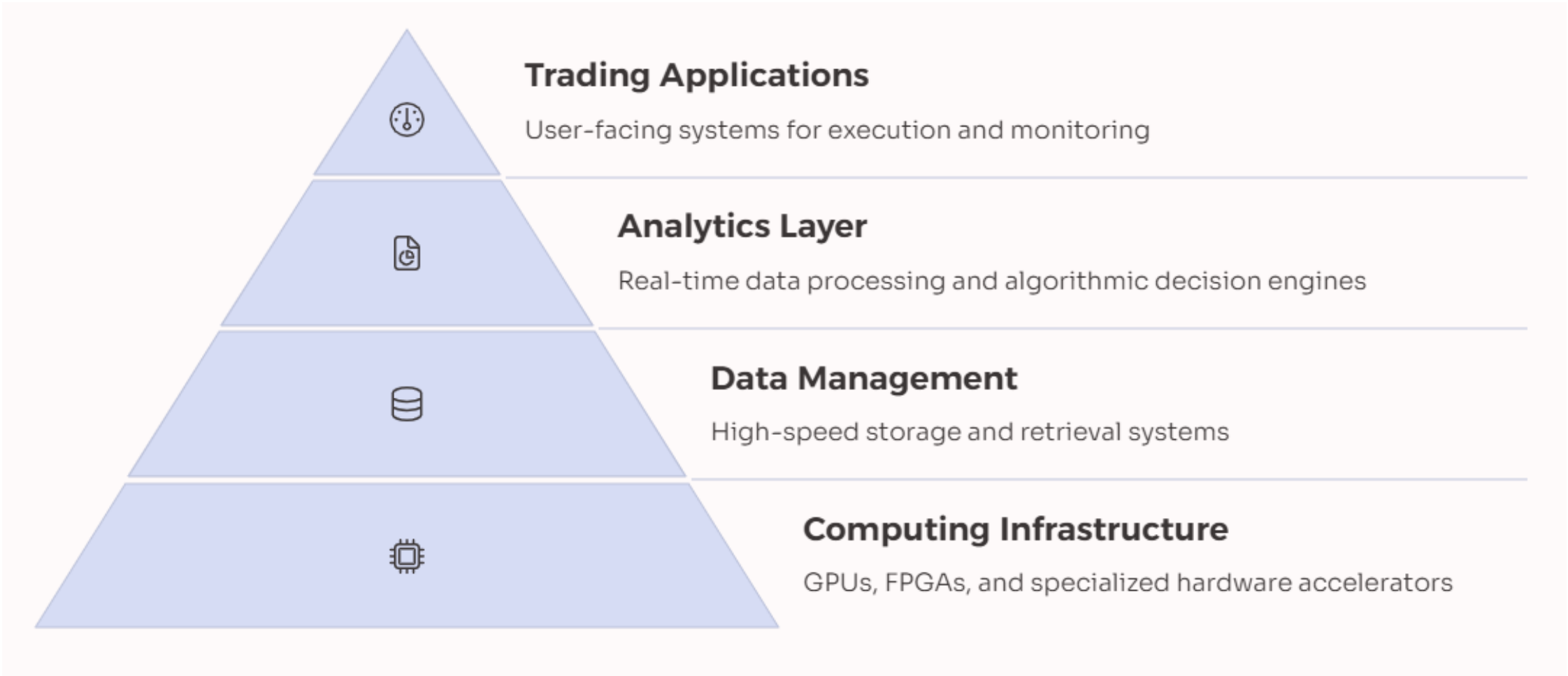
HPC in Finance

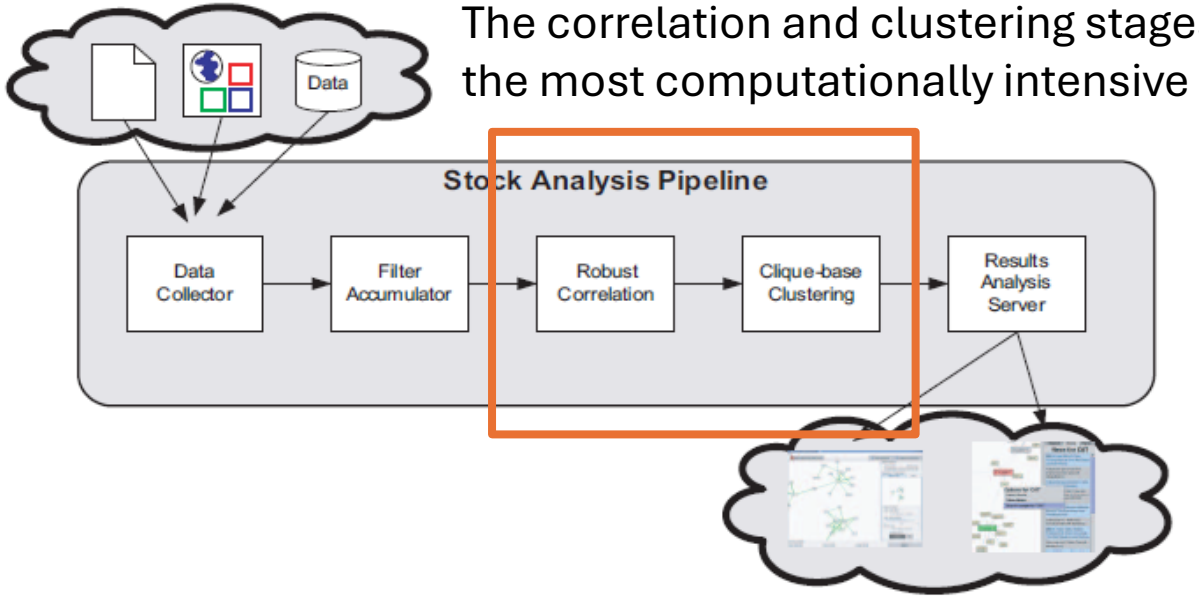
HPC in Real-Time Stock Price Prediction

Jiaxin Lu

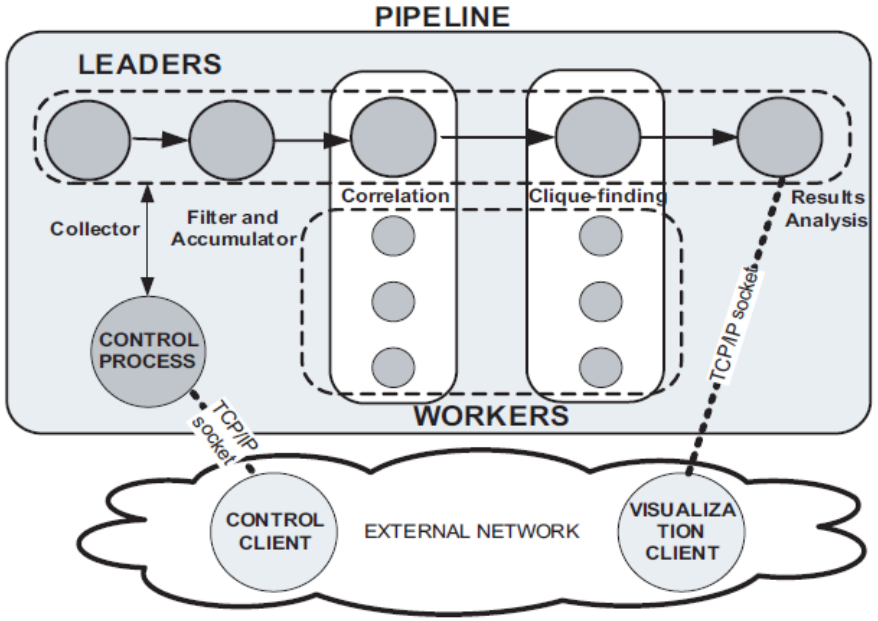
HPC Architecture for Finance



A Parallel Workflow of Real-time Correlation and Clustering



The correlation and clustering stages are the most computationally intensive stages.



The control process uses an intra-communicator to send messages to the leader group, which is composed of one process from each stage of the pipeline.

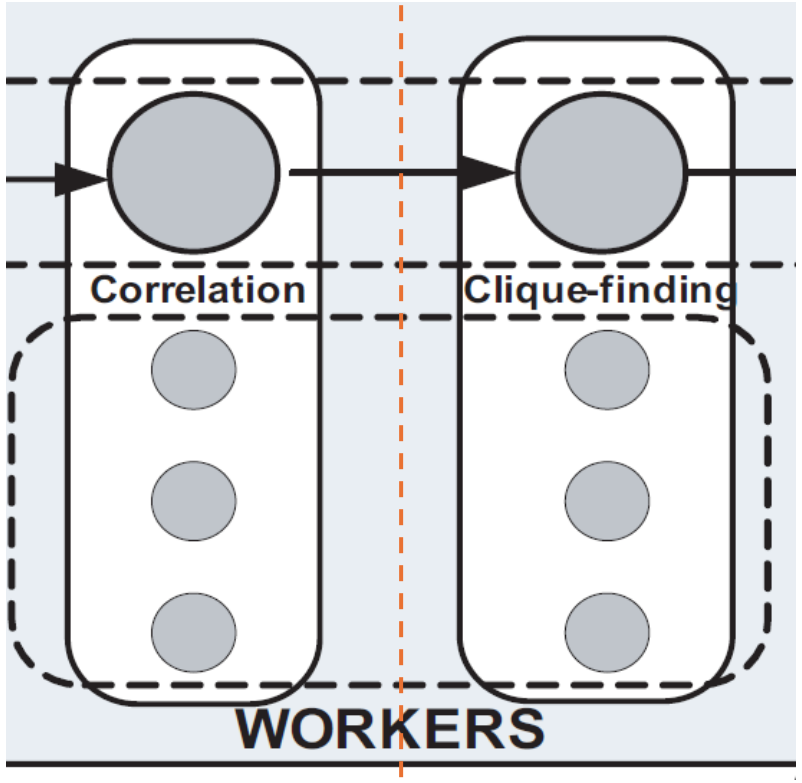
Correlation Calculation

Manager-Worker Architecture using MPI:

1. The manager assigns to each worker a batch of jobs to compute

2. Upon completion the worker sends back the batch of results and subsequently asks for another batch.

This continues until the correlation computation is complete.



Clique-based Clustering

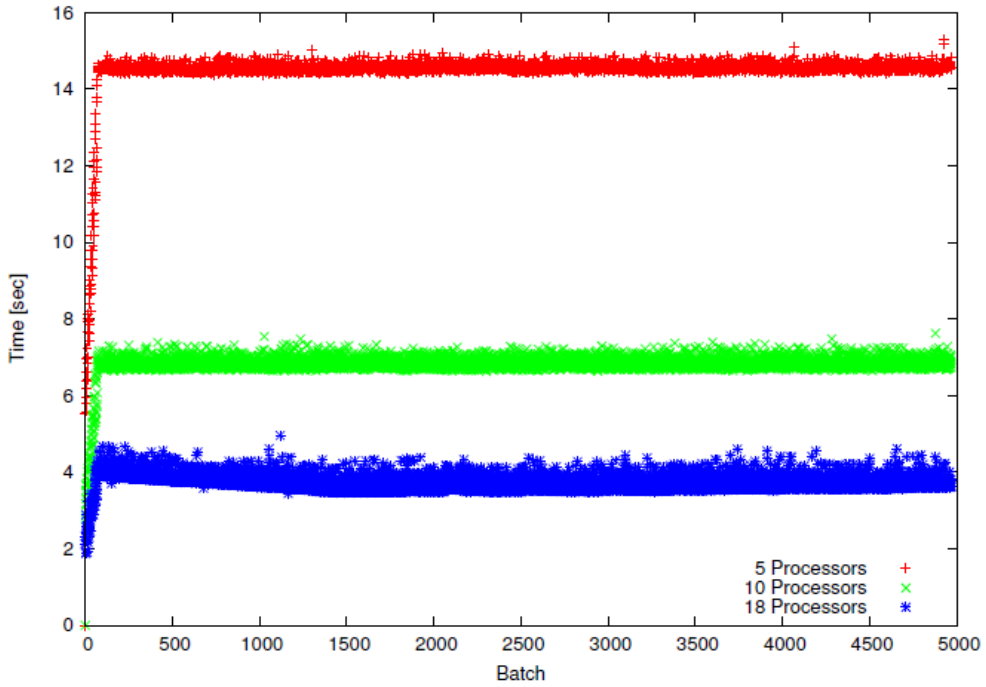
Parallel PLS:

1. Run multiple PLS instances in parallel (independent random searches) for speedup

Use trajectory continuation to update the search state when the graph changes, instead of restarting from scratch.

Modified PLS to output a set of maximal cliques (multiple clusters), not just the single largest clique

Performance



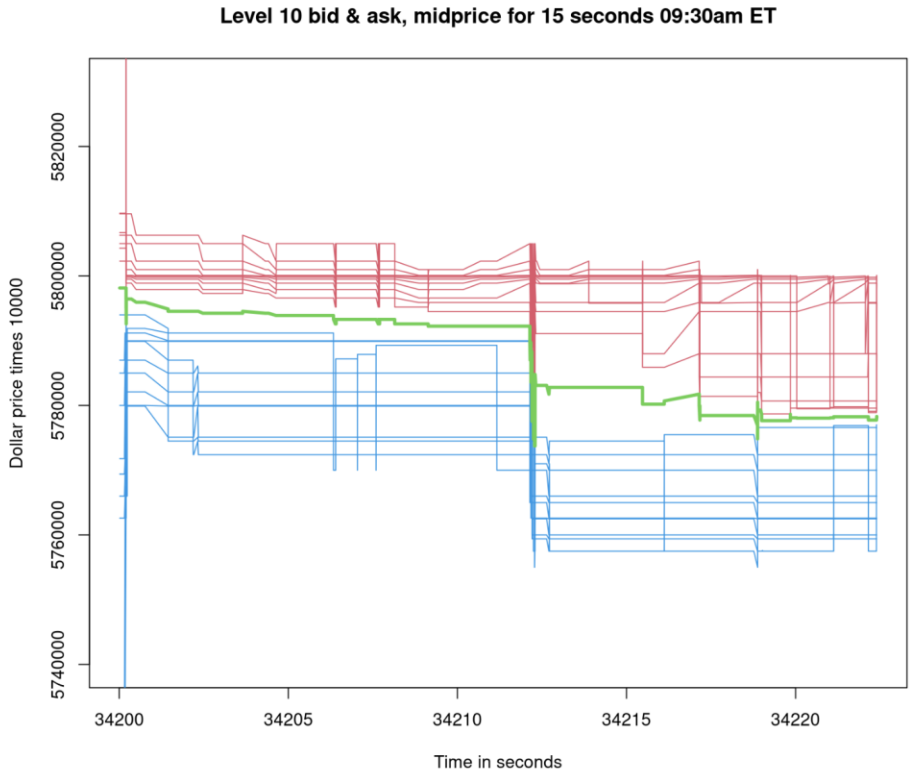
Number of Processors	Avg. response time (seconds)	Speedup (w.r.t. 5)
5	14.49	1
10	6.83	2.12
18	3.74	3.87

GPU-Accelerate Algorithmic Trading Simulations

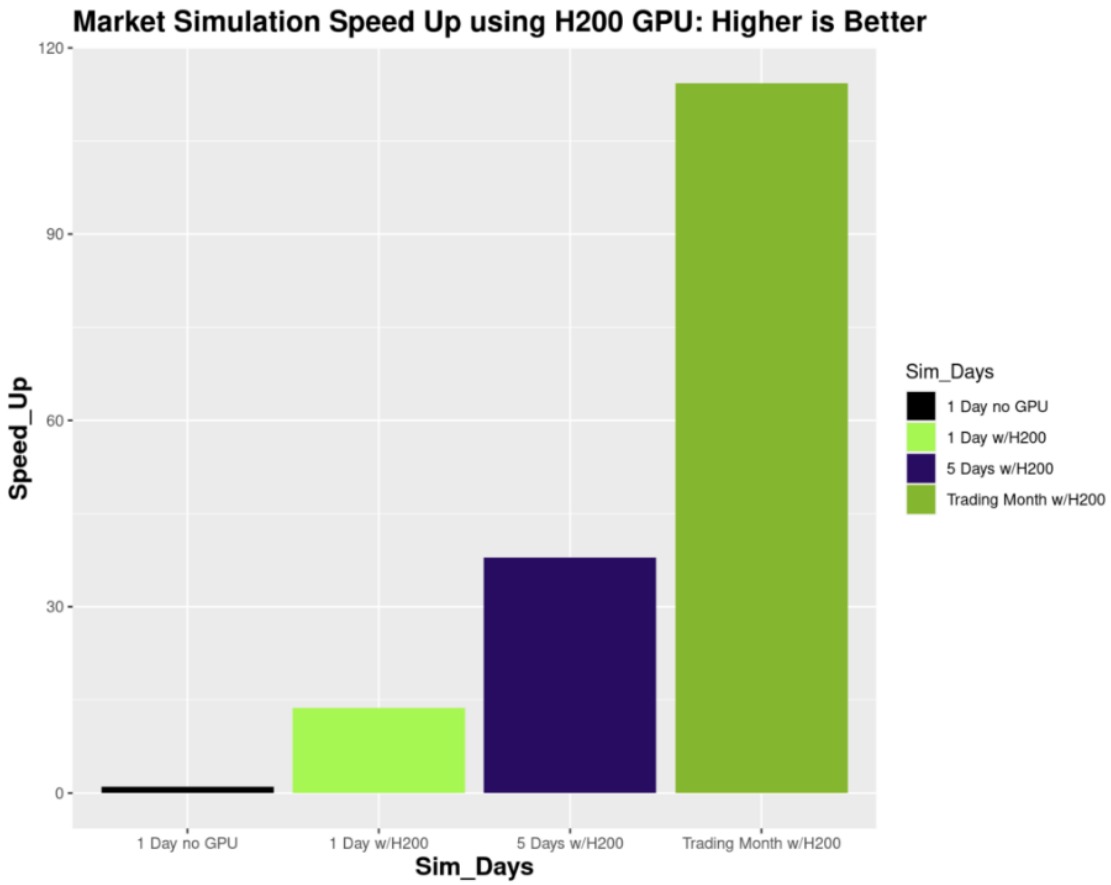
- **Challenge 1: Time-Dependent Computation**
  Solution: Parallelize Over Simulation Paths. Instead of one timeline, simulate many timelines in parallel.

- **Challenge 2: Complex Order Book State**
  Solution: Flatten and Optimize the Order Book Structure

- **Challenge 3: Random Number Generation**
  Solution: Use cuRAND or Batched Transfer



Performance



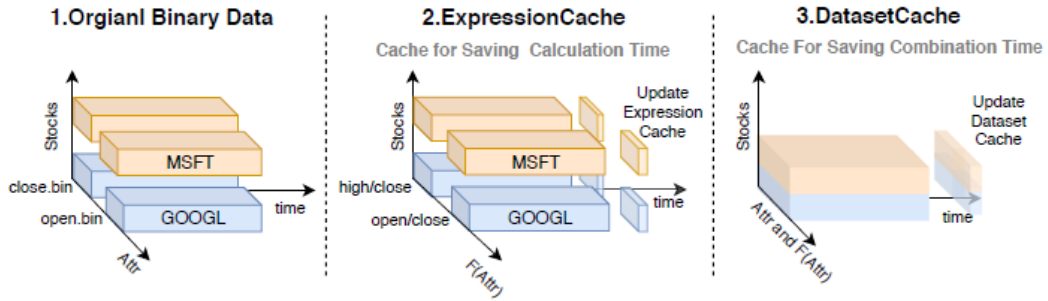
The speedup provided by the GPU comes from the geometry of the algorithm being 2D. In the first dimension, time from 0 to the end T in seconds or fractions of seconds, there is no opportunity for GPU speedup because of the sequential nature of the stochastic differential equation time simulation. The stock price value at the time just past the time point called s_path(t+Δt) depends on s_path(t), the prior value, continuing down the timeline.

Qlib : An AI-oriented Quantitative Investment Platform

- **Challenge 1**: **Workflow Complexity and Flexibility**
  Solution: Modularized, parallel execution of tasks across multiple CPUs/GPUs

- **Challenge 2: Infrastructure Bottlenecks for Massive Data**
  Solution: Qlib introduces a flat-file, time-series optimized database designed for speed and low-latency queries

- **Challenge 3: Low Signal-to-Noise Ratio in Financial Data**
  Solution: Large-scale Monte Carlo simulations and ensemble learning pipelines can be executed across cores or nodes.



In performance tests, Qlib + cache achieved **7.4s** total preprocessing time vs. **365s** for MySQL — a massive improvement driven by compact storage and in-memory computing.

Reference

- [1] Camilo Rostoker, Alan Wagner, Holger H. Hoos. (2007). A Parallel Workflow for Real-time Correlation and Clustering of High-Frequency Stock Market Data. In Proc. of IEEE IPDPS.
- Mark J. Bennett. (2025). GPU-Accelerate Algorithmic Trading Simulations by over 100x with Numba. NVIDIA Technical Blog, March 4, 2025.
- Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, Tie-Yan Liu. (2020). Qlib: An AI-oriented Quantitative Investment Platform. arXiv:2009.11189.

Thank You!