GTC 2014
San Jose, California

# An Approach to Parallel Processing of Big Data in Finance for Alpha Generation and Risk Management

*Yigal Jhirad and Blay Tarnoff*
*March 26, 2014*

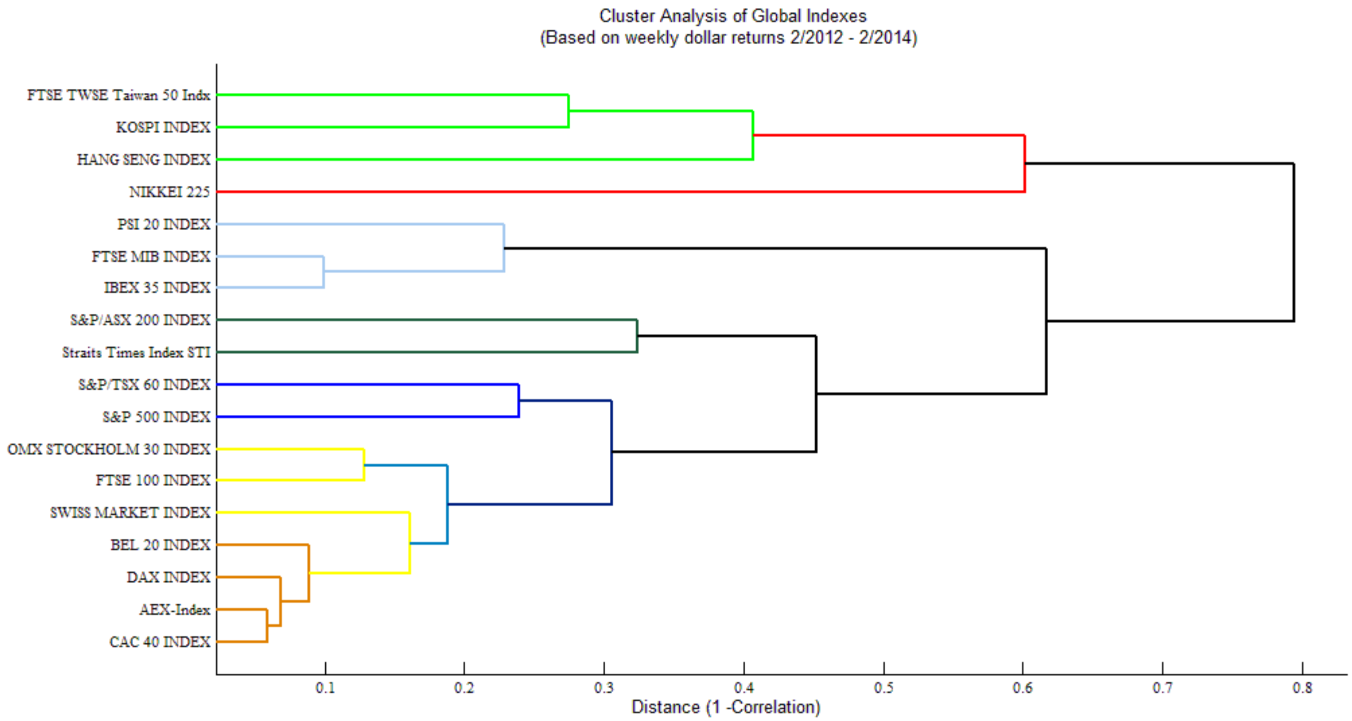
# GTC 2014: Table of Contents

# GTC 2014: The Need For Speed

- **Portfolio/Risk Management analytics demand increasing amounts of computing speed**
  - Big Data: Time Series Data, Interday and Intraday
  - Capture Trends, Patterns, and Signals
  - Coherency and Group Membership
  - Alpha Generation, Risk Management, Market Impact
- **Parallel Processing: APL/CPU and CUDA/GPU physics based modeling exploit hardware efficiently**
  - Array Based Processing paradigm Matrix/Vector thought process is key
  - APL is a programming language whose quantum data object is an array, which is fundamental to parallel processing and can leverage parallel processing across CPU's
  - CUDA leverages GPU Hardware
- **Application in Econometrics and Applied Mathematics**
  - Monte Carlo Simulations
  - Principal Components
  - Cluster Analysis
  - Fourier Analysis
  - Optimization
  - Cointegration
- **Neural Networks**
  - Rapid application development and testing of idea thesis and innovation

# Cluster Analysis

- **Cluster Analysis: A multivariate technique designed to identify relationships and cohesion**
  - Factor Analysis, Risk Model Development
- **Correlation Analysis: Pairwise analysis of data across assets. Each pairwise comparison can be run in parallel.**
  - Use Correlation as primary input to cluster analysis
  - Apply proprietary signal filter to remove selected data and reduce spurious correlations



# Cluster Analysis: Correlation

## Application example: Correlation function removing N/A values

- Correlation measures the direction and strength of a linear relationship between variables. The Pearson product moment correlation between two variables X and Y is calculated as:

$$\frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n(\sum x_i^2) - (\sum x_i)^2}\ \sqrt{n(\sum y_i^2) - (\sum y_i)^2}}$$

- For N assets there are $\frac{\mathrm{N} \times (\mathrm{N}-1)}{2}$ unique correlation pairs

- Given an N x M matrix A in which each row is a list of returns for a particular equity, return an N x N matrix R in which each element is the scalar result of correlating each row of A to every other
  - Each element of A may be an N/A value
  - When processing a pair of rows, the calculation must include neither each N/A value nor the corresponding element in the other row. This requires evaluating each pair separately.
  - As a result the increased computational demand is more effectively implemented through a parallel processing solution. As the matrix size increases the benefits of parallel processing become more significant.

# Optimization: Mixed Integer Optimization

- **Multi-Phase Optimization Analysis**
  - Identify a target portfolio of stocks that is trending consistently over consecutive periods using specialized, possibly time-sensitive, optimization algorithms
  - Establish a portfolio of stocks that is performing in a cohesive way
- **Identifying and Assessing factors driving outperformance**
  - Optimize a basket of factors to track target portfolio
  - Look at factors such as Value vs. Growth, Large Cap vs. Small Cap

Optimized Factor Attribution of Targeted Portfolio

| | Relative Ranking |
|---|---|
| Cash/Assets | 1 |
| Capex/Assets | 2 |
| Dividend Yield | 3 |
| Dividend Growth (1 and 5 Year) | 4 |
| Market Cap (High - Low) | 5 |
| Dividend Payout | 6 |
| ROIC | 7 |
| E/P | 8 |

Indicative factor attribution of target portfolio

Period: 2nd Half of 2013

# Mixed Integer Optimization: Coherency and Pattern Identification using Monte Carlo

**Application example: Maximize the number of Runs of Outperformance**

$$\max_{w_i} \; k \;\; s.t. \left(\prod_{t=q}^{q-k} 1\left[\left(R_p^t > R_I^t\right)\right]\right) > 0 \;\; \text{where} \;\; R_p^t = \sum w_i \times r_i^t$$

**Given a set of assets and periodic prices for each asset, identify sets of long and short weights for those assets that result in portfolios that maximize the number of consecutive positive incremental returns through the most recent period**

- Load matrix of price indexes (assets x periods) into global memory
- Run independent simulations
  - Generate random long and short weights for a subset of assets
  - Evaluate the portfolio and count the outperforming final periods
  - Store the final count and corresponding random seed in global memory
- Sort the final counts to identify the best performers
- Reconstitute the weights for the best performers

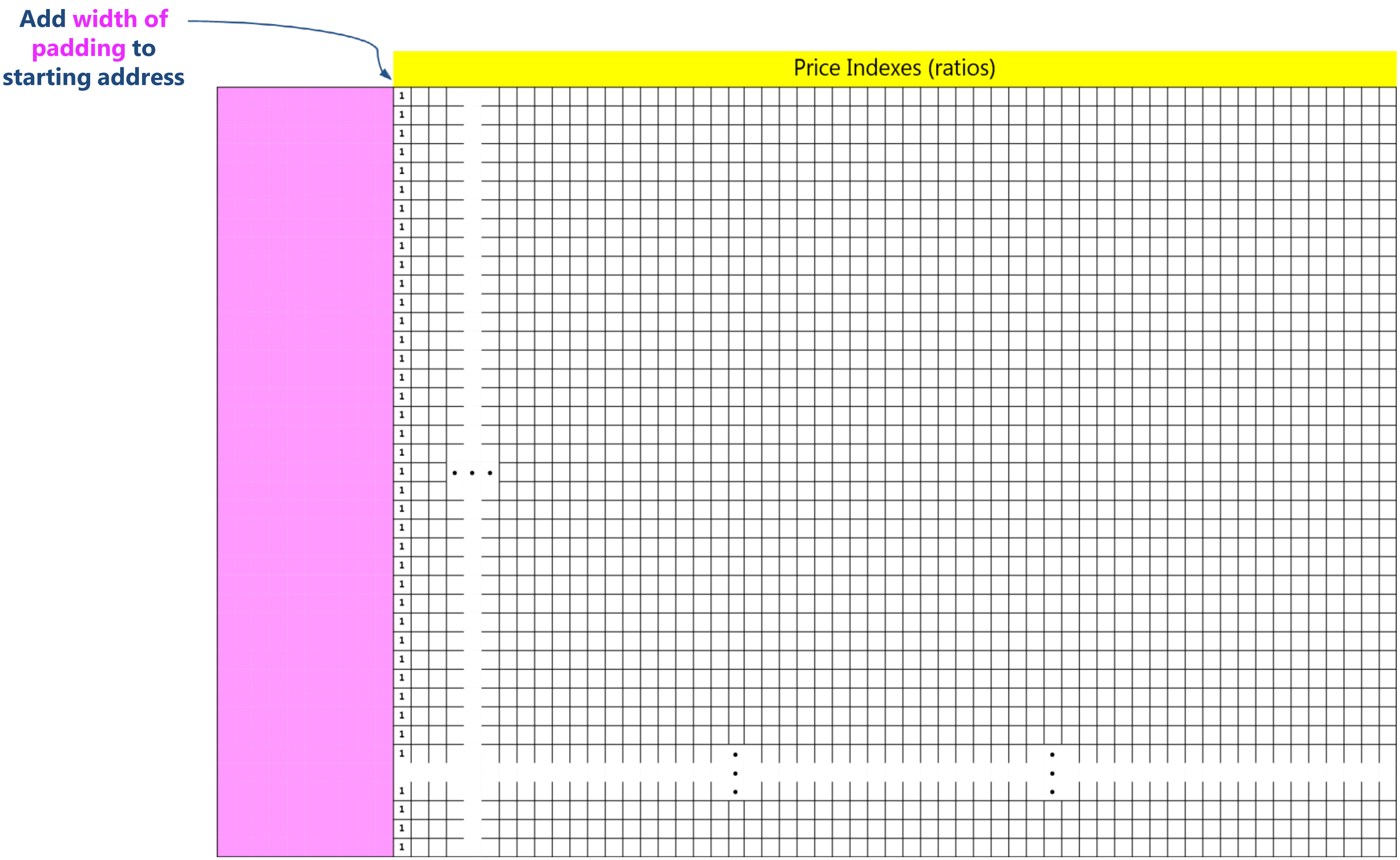
# Mixed Integer Optimization: Configuration

**Tesla C1060**

- **Hardware Constraints**
  - Compute capability: 1.3
  - Max threads per multiprocessor: 1024
  - Max blocks per multiprocessor: 8
  - Number of shared memory banks: 16
  - Coalescence capacity (4-byte words): 64 bytes = 16 contiguous words
  - Number of streaming multiprocessors: 30
- **Software Response – Portfolio Evaluation**
  - Run 16 independent simulations per block for output coalescence
  - Read 16 price indexes per simulation for input coalescence
  - Thread block configuration: 16 x 16 = 256 threads
  - Max blocks per multiprocessor: 1 - 4, depending on occupancy
  - Max contemporaneous blocks: 30 - 120, depending on occupancy
  - For sort, maximize number of blocks: 8 => 128 threads per block

# Mixed Integer Optimization: Price Indexes Input Matrix

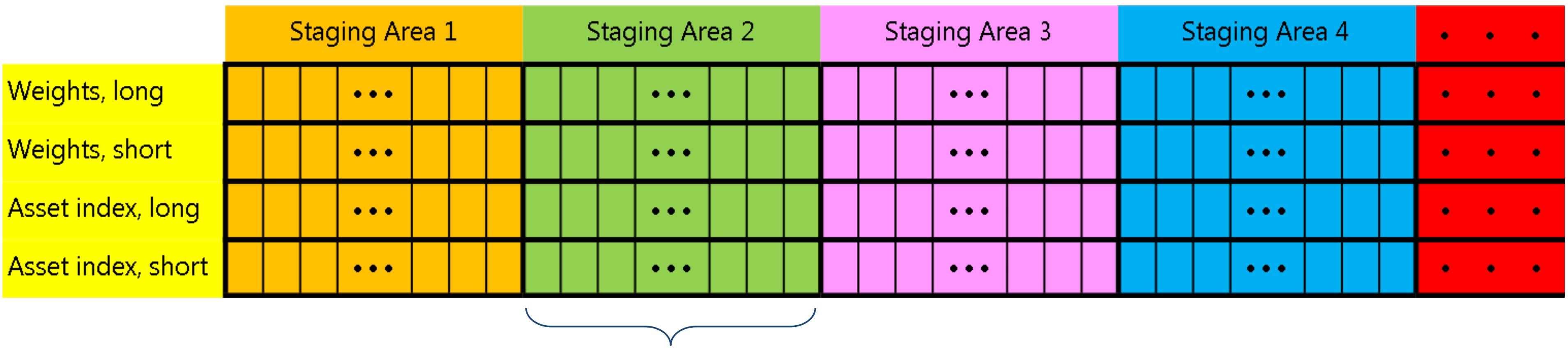
**Process right to left**



# Mixed Integer Optimization: Weights Staging Area

**Global Memory**

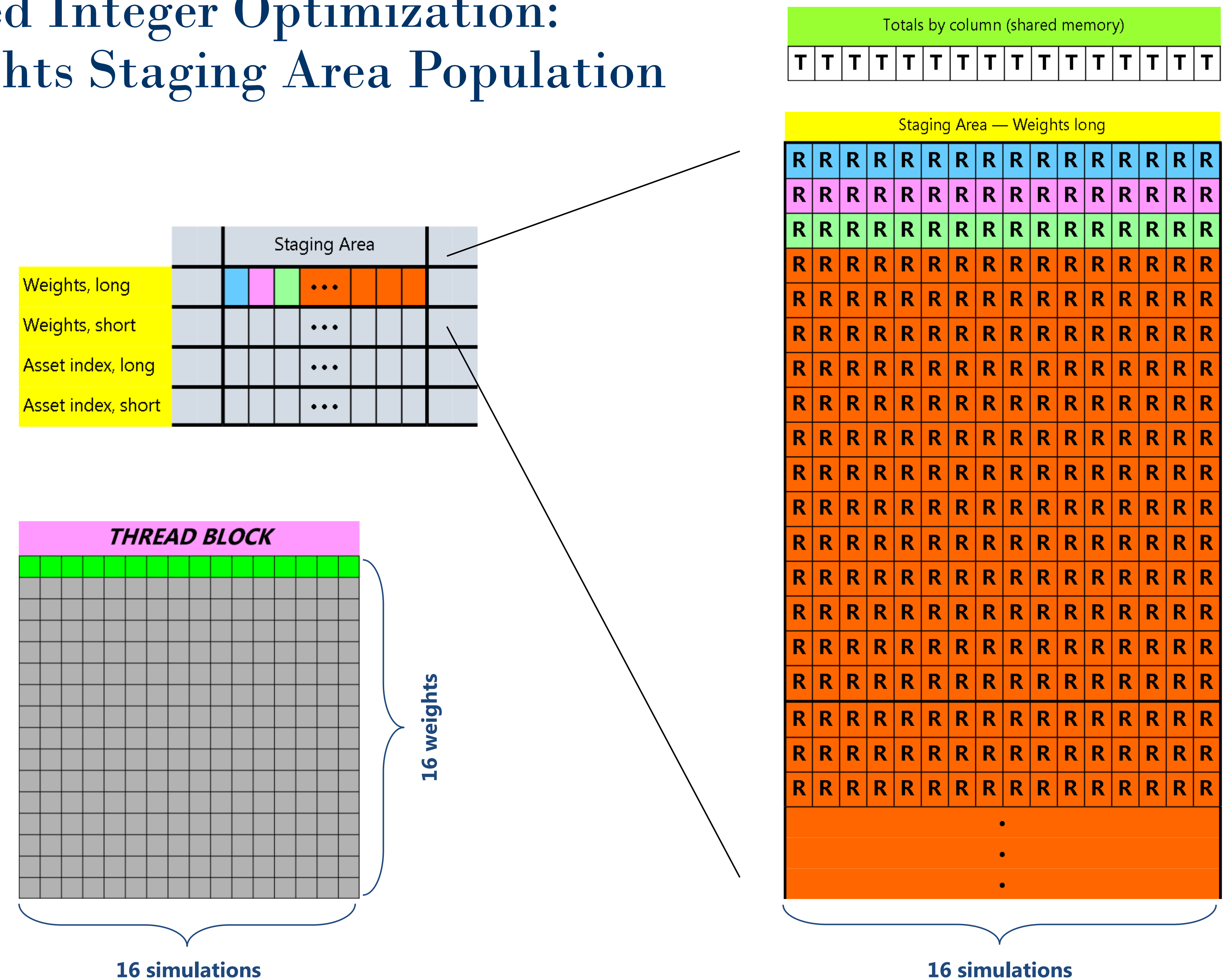
- **Temporary storage for weights generated, used and discarded during the life of a block**




**Width = max # of longs/shorts (area not necessarily rectangular)**

- Number of staging areas = # of multiprocessors on card (multiProcessorCount) * max # of blocks per multiprocessor (4 or less, depending on occupancy) -- Creates enough staging areas to accommodate all blocks that may execute simultaneously
- Each staging area holds data for 16 simulations
- Each cell represents 16 * 4-byte floats or integers (one per simulation)
- Each area populated in three loops: long weights one at a time, short weights one at a time, both asset indexes one asset at a time
- Accompanied by an array of 16 * 4-byte integers = 512 bits, each indicating whether the corresponding staging area is currently available (atomicExch with 0 to capture array and mark all areas “not free” while searching, identify and flip lowest “true” bit, atomicOr array back in)

# Mixed Integer Optimization: Weights Staging Area Population



# Mixed Integer Optimization: Weights Staging Area Population

## Asset Indexes

- **How *not* to do it**

Staging Area — Asset index long

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 1 | 3 | 0 | 0 | 3 | 2 | 7 | 1 | 3 | 7 | 0 | 0 | 3 | 1 |
| 1 | 5 | 2 | 7 | 3 | 6 | 6 | 4 | 8 | 5 | 5 | 9 | 2 | | 7 | 2 |
| 2 | 6 | 4 | 8 | 4 | | 7 | 6 | 9 | 7 | 6 | | 4 | | 8 | 7 |
| 5 | 9 | | 9 | 7 | | | 7 | | 9 | | | | | | 9 |
| 8 | | | | | | | 9 | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

⋮

Staging Area — Asset index short

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 0 | 0 | 5 | 1 | 2 | 3 | 1 | 0 | 0 | 1 | 3 | 2 | 0 | 5 |
| 4 | 7 | 3 | 1 | 6 | 3 | 4 | 8 | | 4 | 1 | 2 | 5 | 3 | | 6 |
| 9 | | 6 | 2 | 9 | 4 | 8 | | | 8 | 4 | 5 | 6 | 6 | | |
| | | 8 | 6 | | 8 | | | | | 7 | 6 | 9 | 9 | | |
| | | 9 | | | 9 | | | | | 9 | 8 | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

⋮

# Mixed Integer Optimization: Weights Staging Area Populated

- **Long Values, same for shorts**

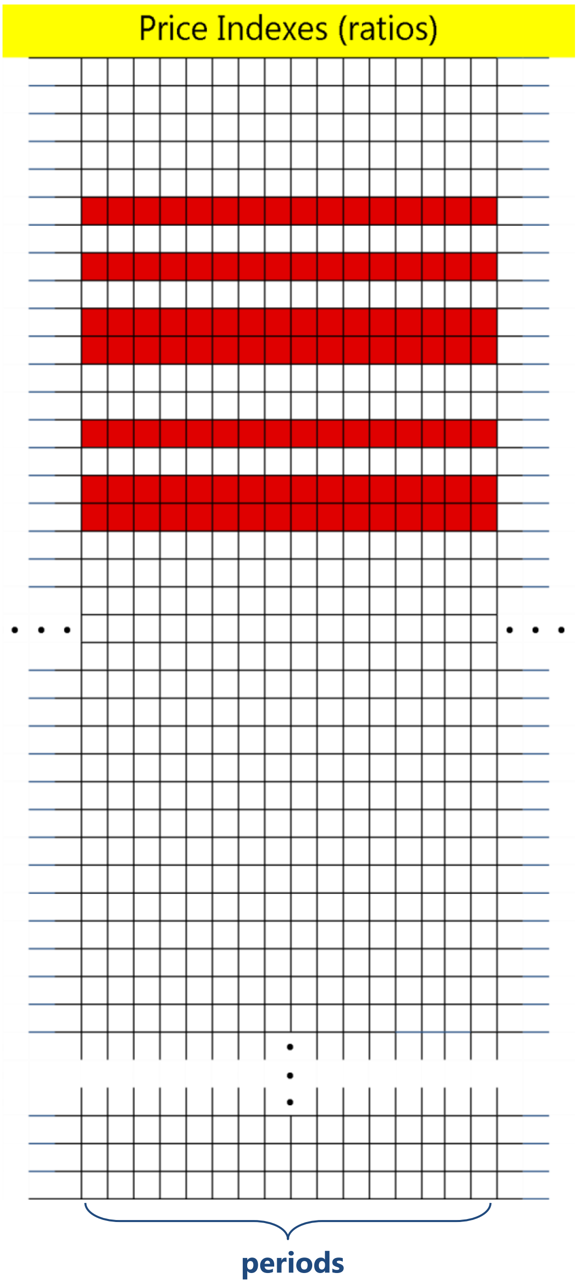
| Totals by column (shared memory) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T |

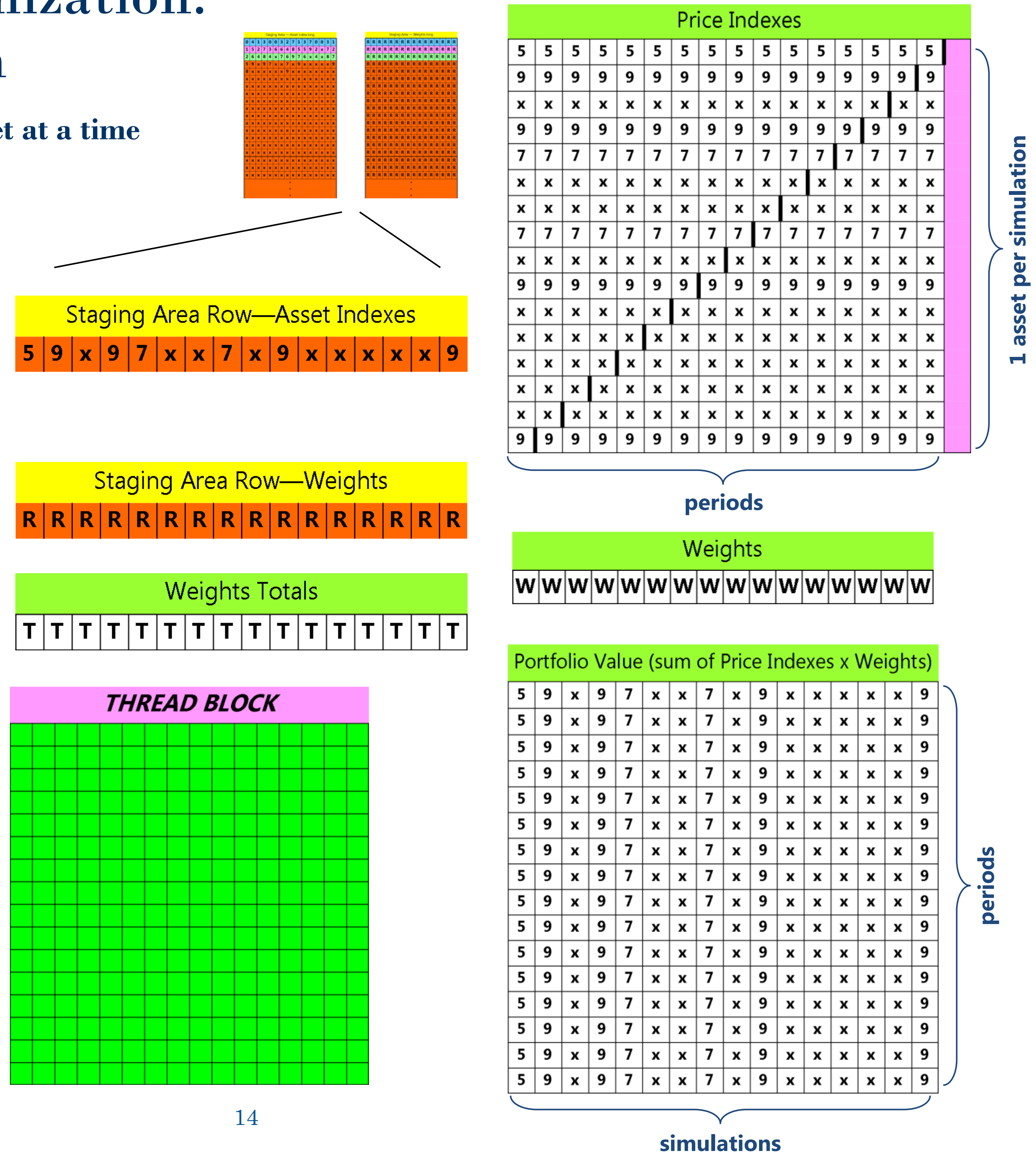
| Staging Area — Weights long | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| . | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | |

| Staging Area — Asset index long | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 1 | 3 | 0 | 0 | 3 | 2 | 7 | 1 | 3 | 7 | 0 | 0 | 3 | 1 |
| 1 | 5 | 2 | 7 | 3 | 6 | 6 | 4 | 8 | 5 | 5 | 9 | 2 | x | 7 | 2 |
| 2 | 6 | 4 | 8 | 4 | x | 7 | 6 | 9 | 7 | 6 | x | 4 | x | 8 | 7 |
| 5 | 9 | x | 9 | 7 | x | x | 7 | x | 9 | x | x | x | x | x | 9 |
| 8 | x | x | x | x | x | x | 9 | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| . | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | |
| . | | | | | | | | | | | | | | | |

# Mixed Integer Optimization: Portfolio Evaluation

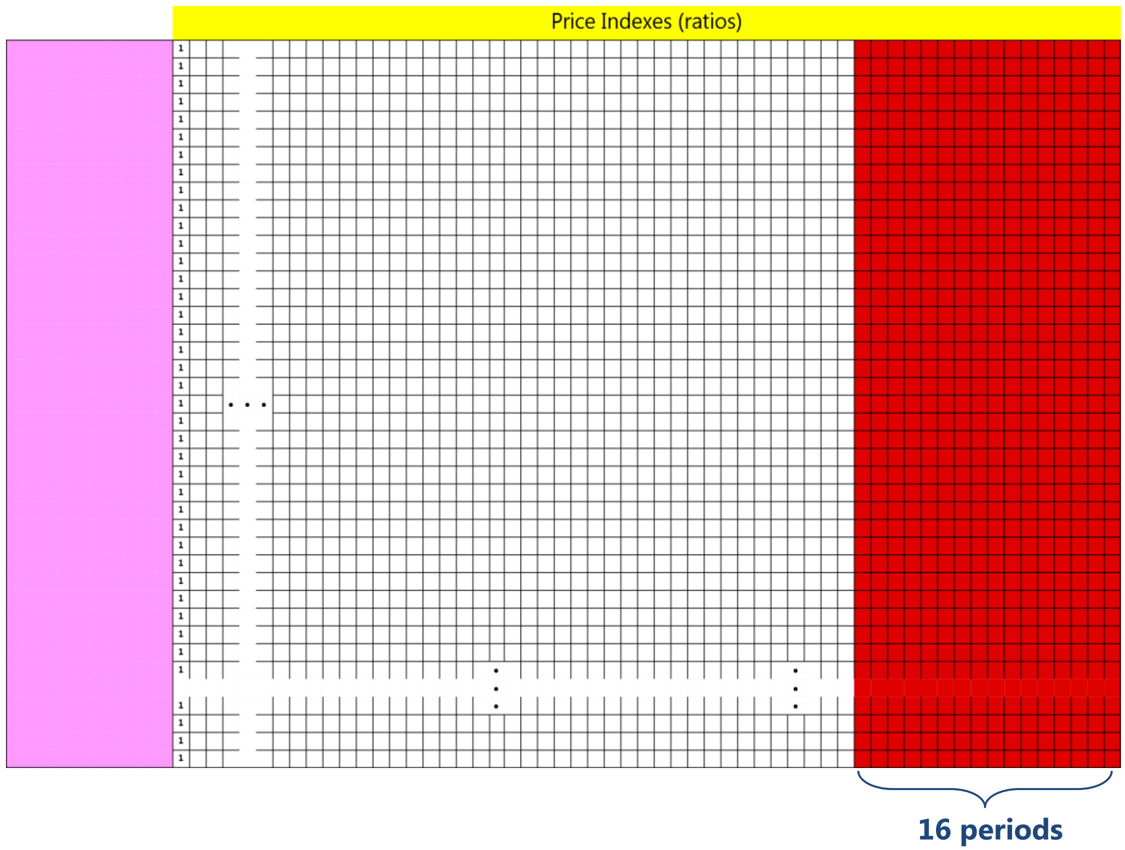
## Loop by row of staging area: one asset at a time

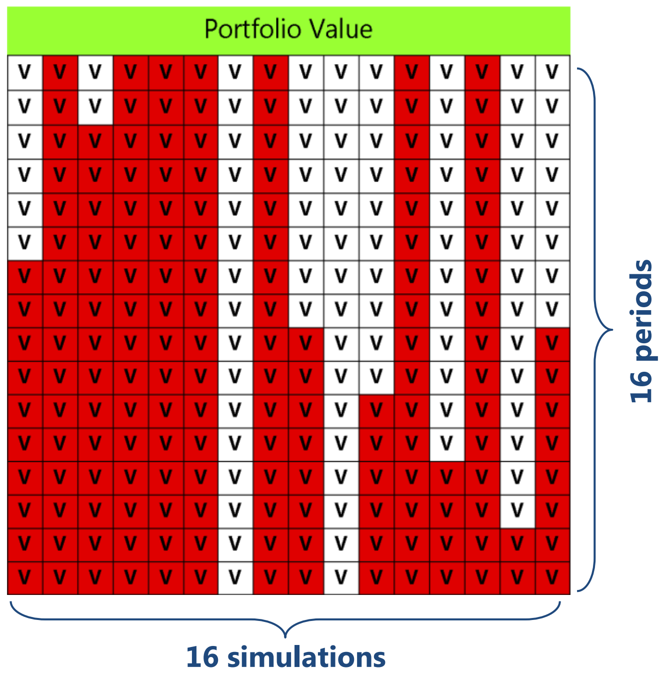


# Mixed Integer Optimization: Overall Portfolio Evaluation

## Process right to left




| Results | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 16 | 14 | 16 | 16 | 16 | 0 | 16 | 8 | 0 | 6 | 16 | 4 | 16 | 2 | 8 |

# Mixed Integer Optimization: Sort and Store Results

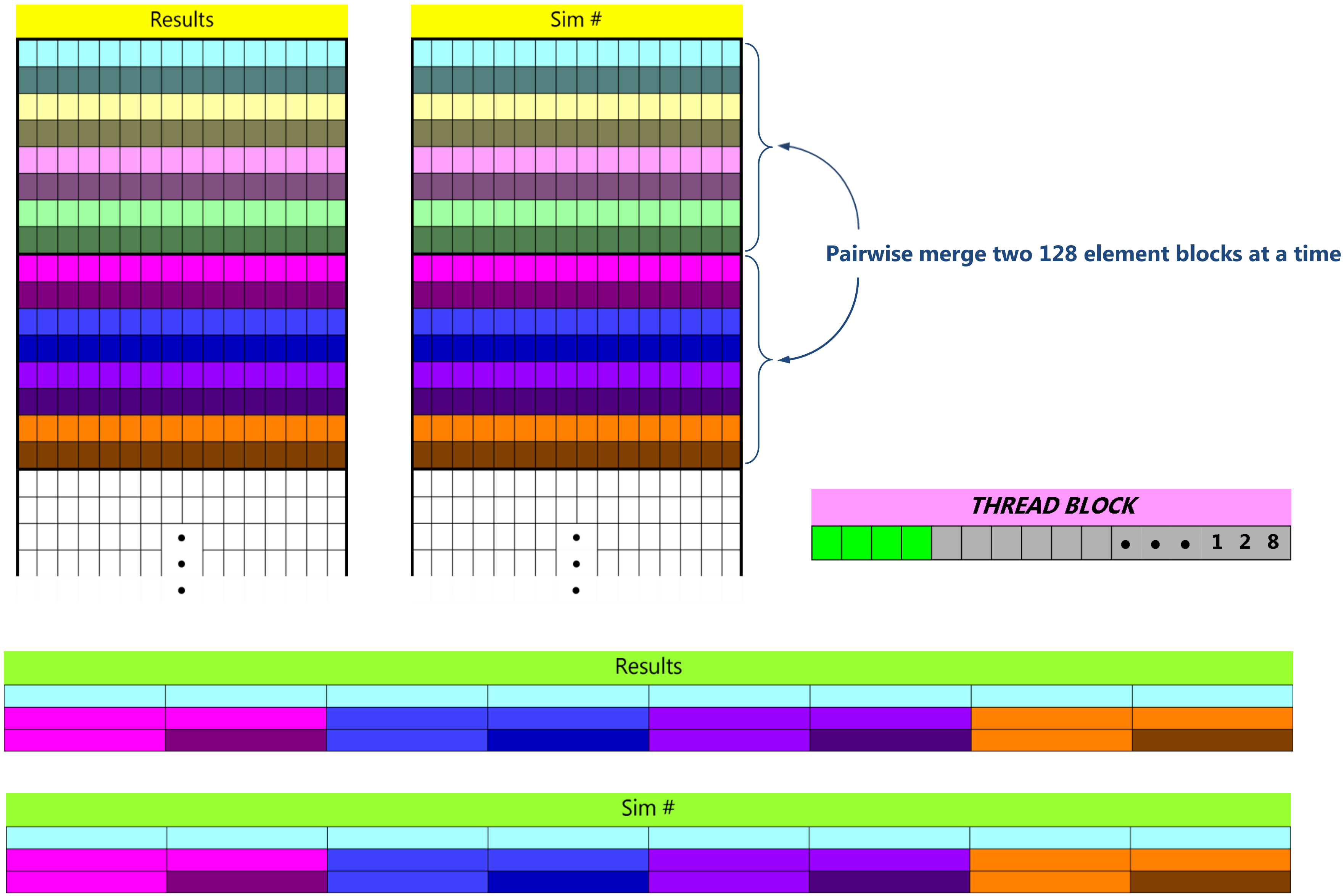
| Results | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 29 | 14 | 19 | 19 | 40 | 0 | 26 | 8 | 0 | 6 | 48 | 4 | 34 | 2 | 8 |

| Rank | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 3 | 7 | 5 | 6 | 1 | 14 | 4 | 9 | 15 | 11 | 0 | 12 | 2 | 13 | 10 |
| 6 | 3 | 6 | 5 | 6 | 1 | 7 | 4 | 6 | 7 | 7 | 0 | 7 | 2 | 7 | 6 |
| 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 4 | 1 | 0 | 2 | 0 | 2 | 1 |
| 3 | 1 | 3 | 2 | 3 | 1 | 4 | 1 | 3 | 4 | 4 | 0 | 4 | 1 | 4 | 3 |
| 1 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 2 | 1 | 0 | 1 | 0 | 2 | 1 |
| 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 1 | 2 | 2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 2 | 2 | 0 | 2 | 0 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |

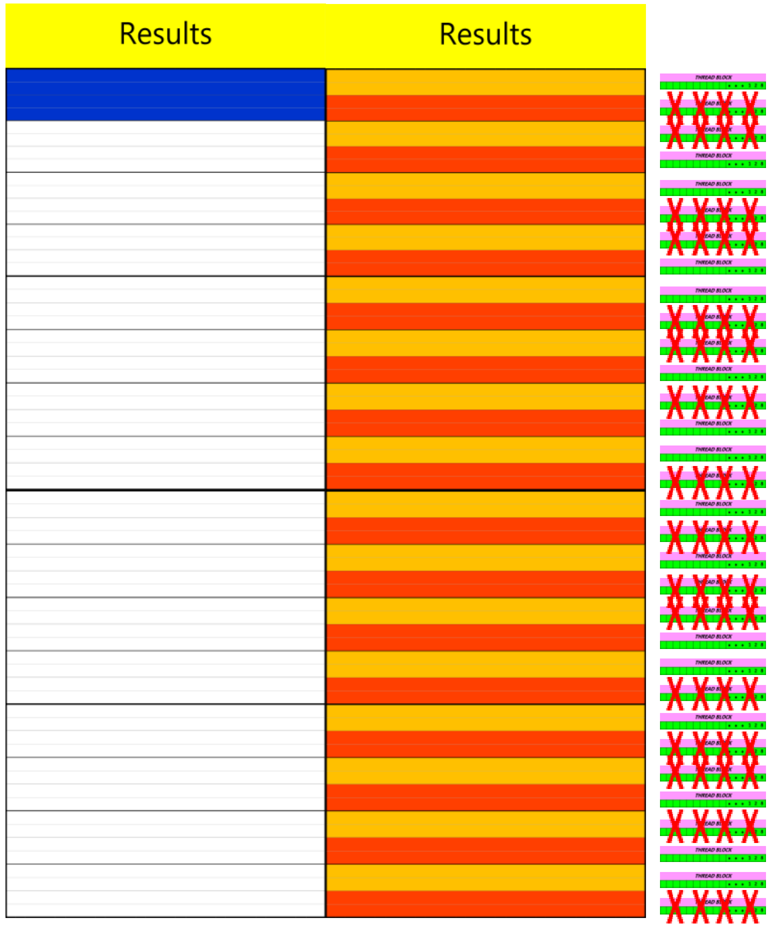
| Results | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | • • • | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 48 | 40 | 34 | 29 | 26 | 19 | 19 | 14 | 10 | 8 | 8 | 6 | 4 | 2 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| | | | | | | | • • • | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| Sim # | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | • • • | | | | | | | | |
| | | | | | | | | | | | | | | | |
| ...11 | ...5 | ...13 | ...1 | ...7 | ...3 | ...4 | ...2 | ...0 | ...8 | ...15 | ...10 | ...12 | ...14 | ...6 | ...9 |
| | | | | | | | | | | | | | | | |
| | | | | | | | • • • | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

# Mixed Integer Optimization: Merge-Sort Results



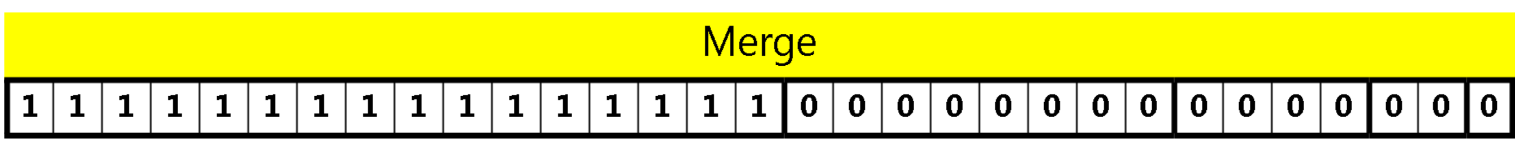
# Mixed Integer Optimization: Final Merge-Sort




Once it has written its final row, each block will atomicSwap the value 1 to the first Merge cell

The first block to finish finds a 0 in that cell, meaning its counterpart block has not yet written its final row, so it quits

The second block to finish finds a 1 that cell, meaning its counterpart block has already written its final row, so it is the one that can continue on to perform the next level of sort



# Summary

- **Parallel processing capabilities provide a powerful way to analyze large amounts of financial time series data in a Big Data framework**
  - The need for speed calls for parallel CPU and GPU solutions
  - Need to have not just nice to have
- **Our approach is based on time sensitive implementation of Cluster Analysis and Mixed Integer Optimization Analysis**
  - Capture Trends, Patterns, and Signals
  - Identify Coherency and Group Membership
  - Evolutionary algorithm with Time-Urgent objective maximization
- **Applications in Portfolio Management, Risk Management, and Trading/Execution**
  - Alpha Generation
  - Risk Analysis
  - Market Impact

# Author Biographies

- **Yigal D. Jhirad,** Senior Vice President, is Director of Quantitative Strategies and a Portfolio Manager for Cohen & Steers' options and real assets strategies. Mr. Jhirad heads the firm's Investment Risk Committee. He has 26 years of experience. Prior to joining the firm in 2007, Mr. Jhirad was an executive director in the institutional equities division of Morgan Stanley, where he headed the company's portfolio and derivatives strategies effort. He was responsible for developing, implementing and marketing quantitative and derivatives products to a broad array of institutional clients, including hedge funds, active and passive funds, pension funds and endowments. Mr. Jhirad holds a BS from the Wharton School. He is a Financial Risk Manager (FRM), as Certified by the Global Association of Risk Professionals.

- **Blay A. Tarnoff** is a senior applications developer and database architect. He specializes in array programming and database design and development. He has developed equity and derivatives applications for program trading, proprietary trading, quantitative strategy, and risk management. He is currently a consultant at Cohen & Steers and was previously at Morgan Stanley.