Massively Parallel Fourier-Space Cross-Correlation:

Analyzing Highly Dimensional Time Series Databases

Repurposing Gaming Hardware for Science

Matthias Alexander Lee

Wentworth Institute of Technology

Harvard University Initiative in Innovative Computing

In collaboration with:

Pavlos Protopapas
Harvard University CfA

&

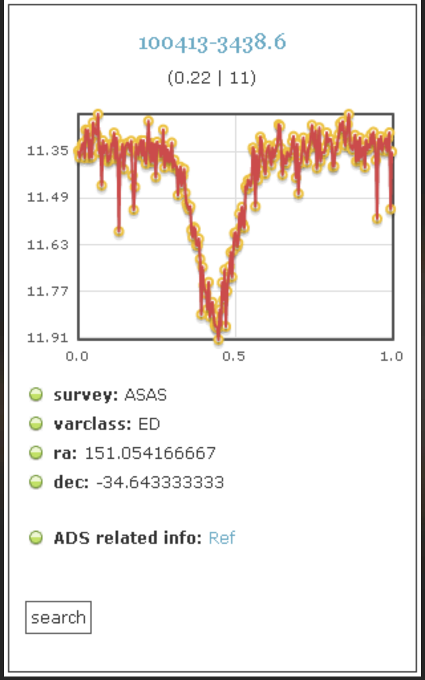
Patrick Ohiomoba
Harvard University IIC

Partially funded by NSF REU Grant # CDI-0835713

Overview

Massively Parallel Fourier-Space Cross-Correlation

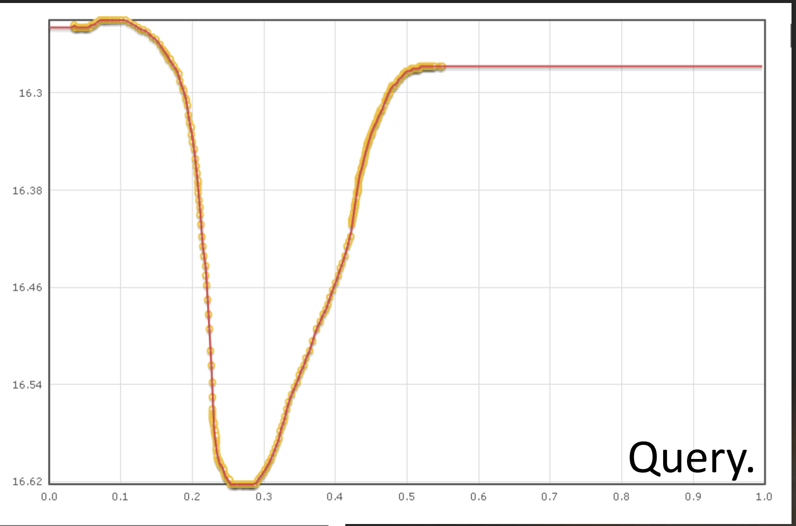
- Background
- Our Approach
  - General Structure
  - Graphics Processors (GPUs)
  - Searching
  - Indexing
  - Tree Structure
- Results!
- Demo

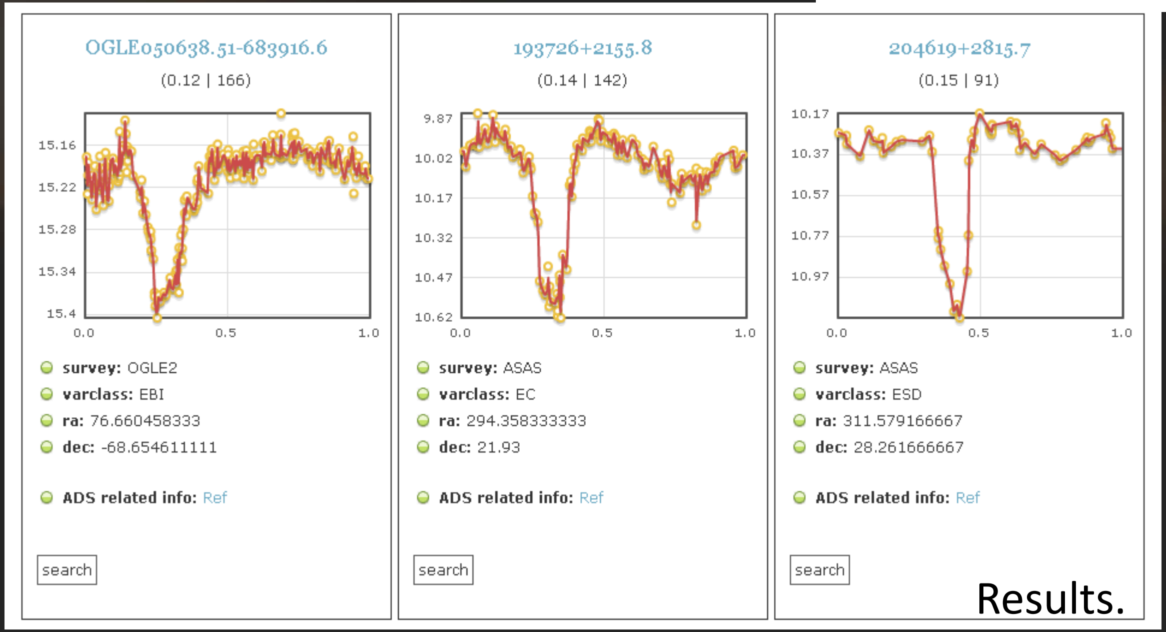


Real-time Time Series DB Searching

The Problem.

- Problem.
  - Searching Light Curves
  - 100-1000 Data-points





Real-time Time Series DB Searching

The Problem.

- Problem.
  - Searching Light Curve Databases
    - Light Curve = brightness over time
    - Millions of Time Series
    - High interval resolution
    - Computationally expensive
  - **Is Real-time searching possible?**

Real-time Time Series DB Searching

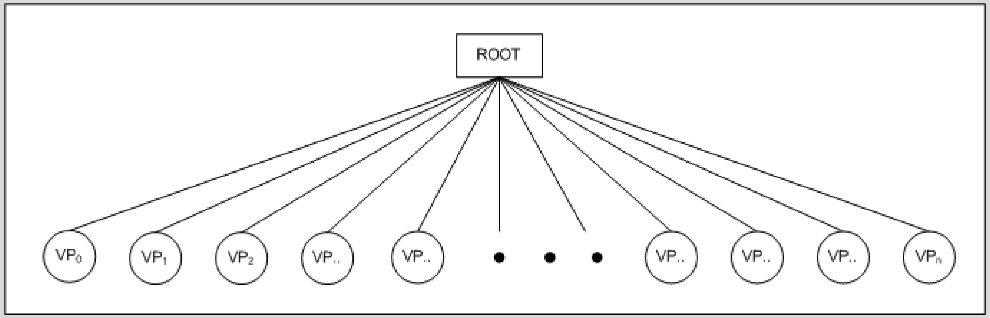
The Solution.

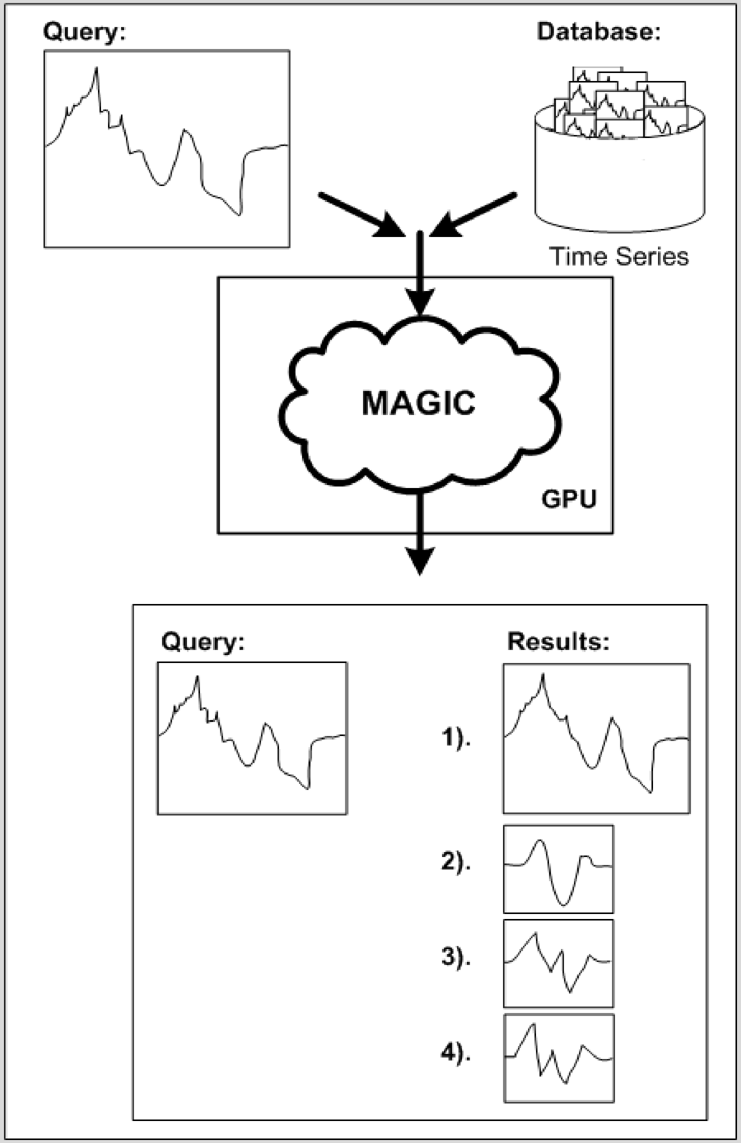
- Solution
  - Offline Preprocessing
  - Using Fast Fourier Transform
  - Extremely Wide Tree Structure
  - Parallel Processing Platform

Approach in a **NutShell.**

FFT Cross Correlation - O(n log n)

- Comparison:
  - FFT for a Cross Correlation
    - O(n log n) efficiency
- Tree Structure:
  - Extremely Wide Tree
    - 10,000+ Branches
    - Bonsai Tree





Graphics Processors GPUs

Gaming Hardware for Science. . . (quick intro)

Pros:

- High FLOP/s (per GPU)
  - 1.35 TFLOP/s – Nvidia
  - 2.73 TFLOP/s – ATI
- 1000s of Threads
- Cheap Performance!
  - 300-600$
- Fast Memory
  - +160GB/s

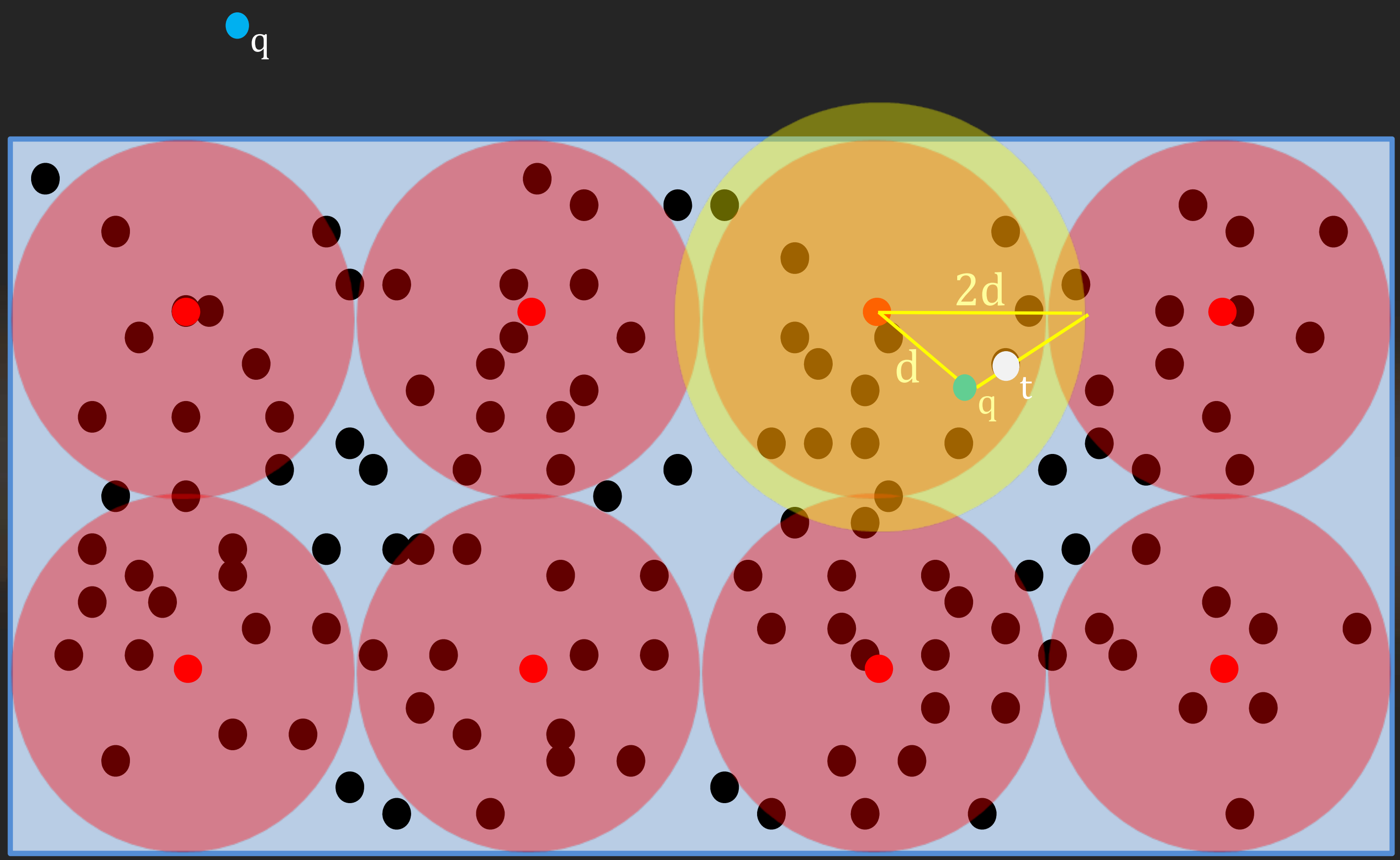
Cons:

- Limited Memory
  - 1GB - 4GB
- Small Cache
  - 16Kb/SIMD
- Limited Threads

Indexing and Vantage Points

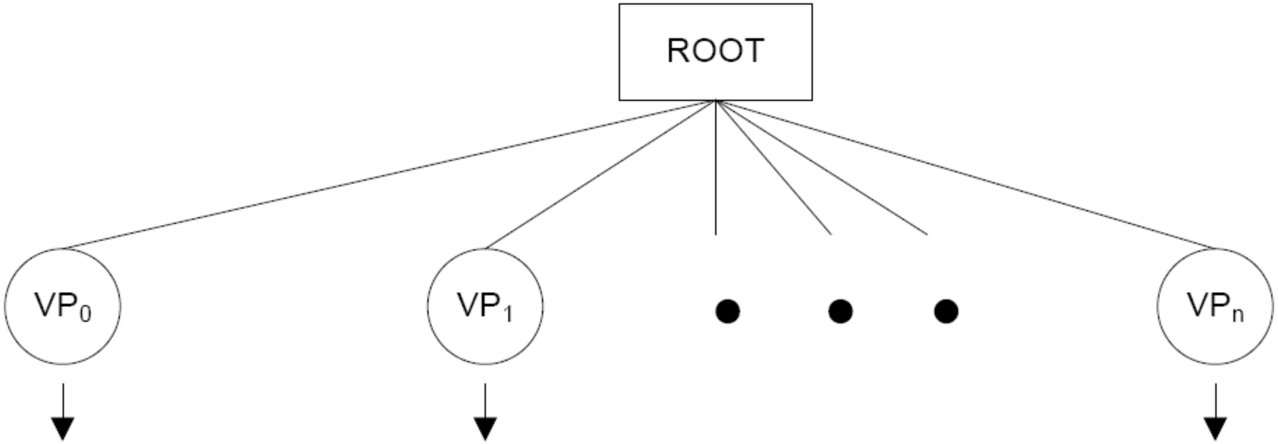
Good Vantage point distribution is essential

d = Dist(VP3, Query)



Tree Structure Search:

Bonsai Tree



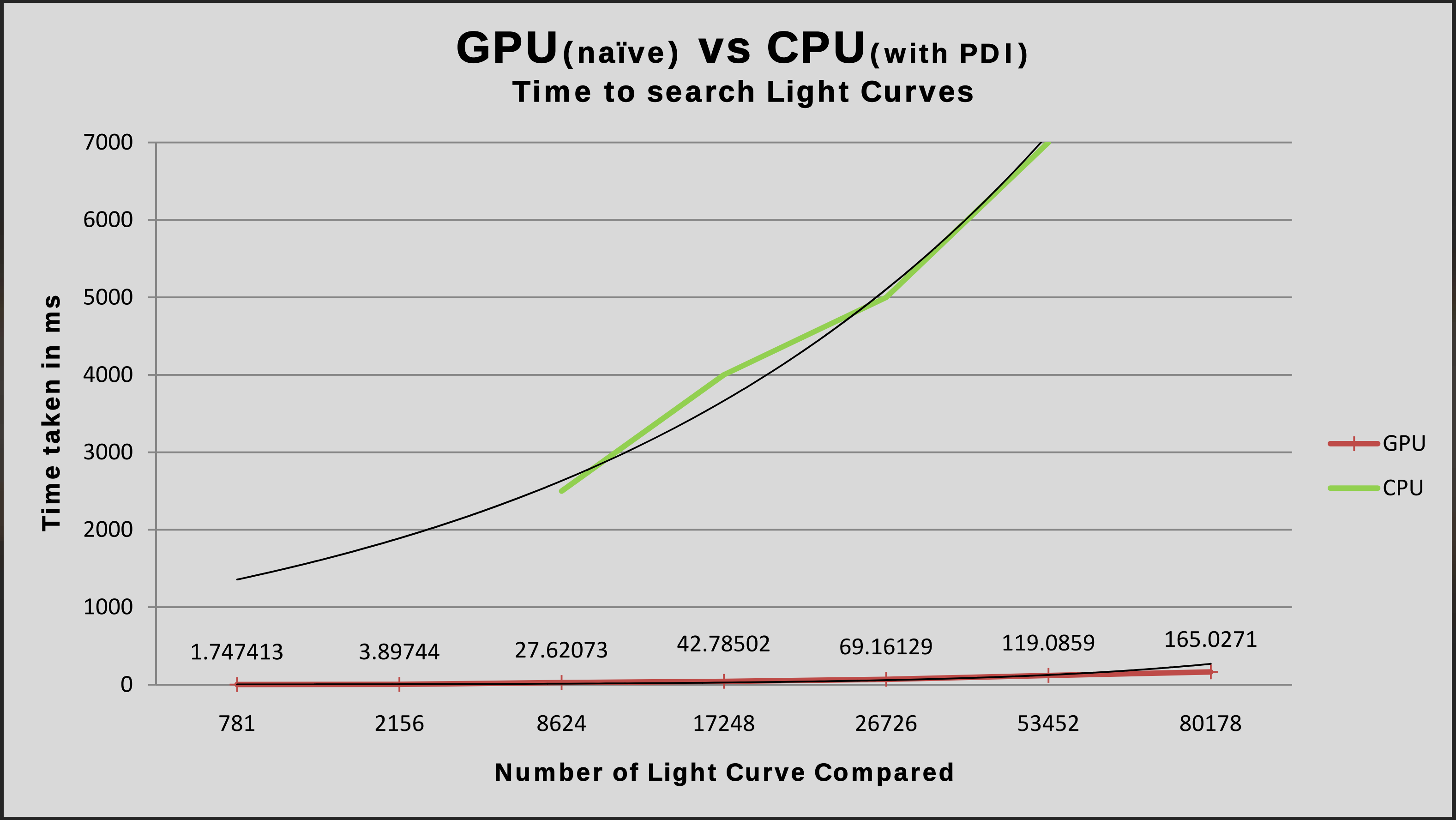
LC_{id}	Corr
54	0.0
123	0.021
223	0.115
1245	0.122
15248	0.195

LC_{id}	Corr
15248	0.0
251	0.001
23	0.235
585	0.256
12	0.378

LC_{id}	Corr
1373	0.0
69	0.145
327	0.146
31337	0.182
1988	0.214

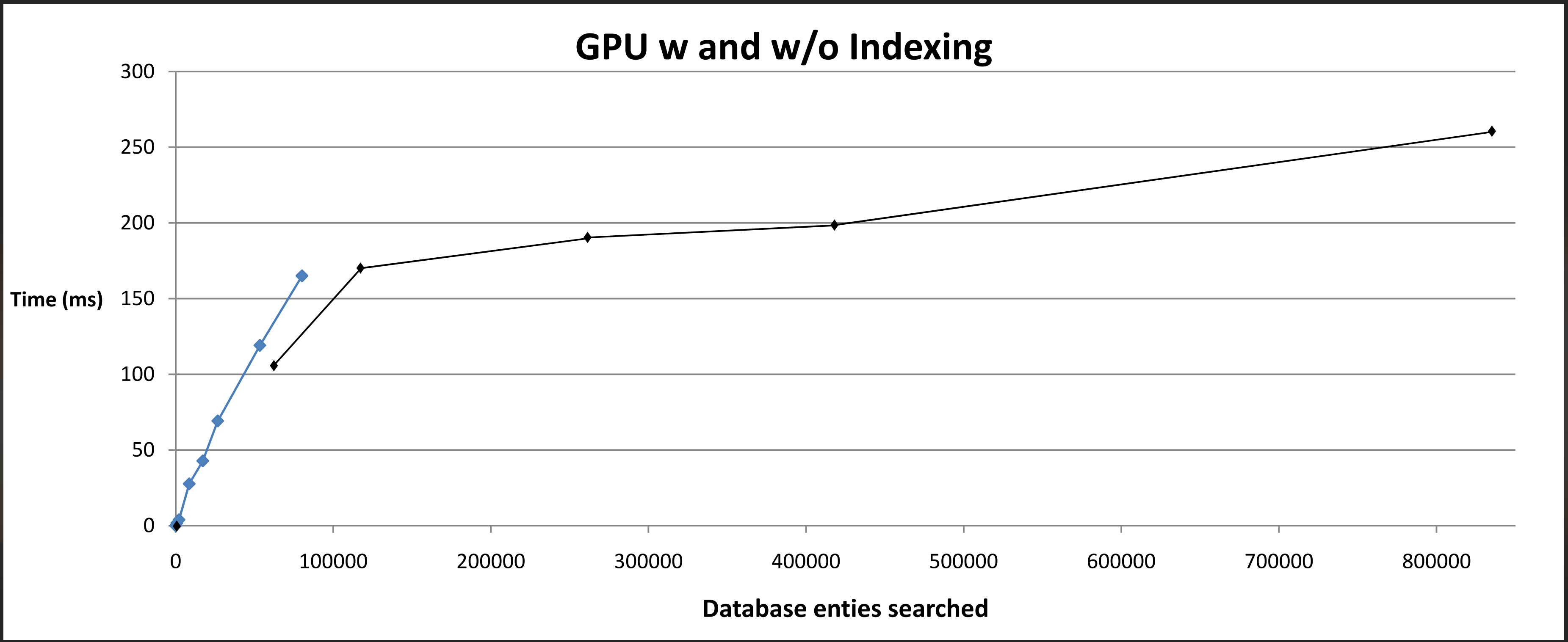
GPU Speed-Up:

GPU naïve vs CPU with indexing: Parallel vs Serial



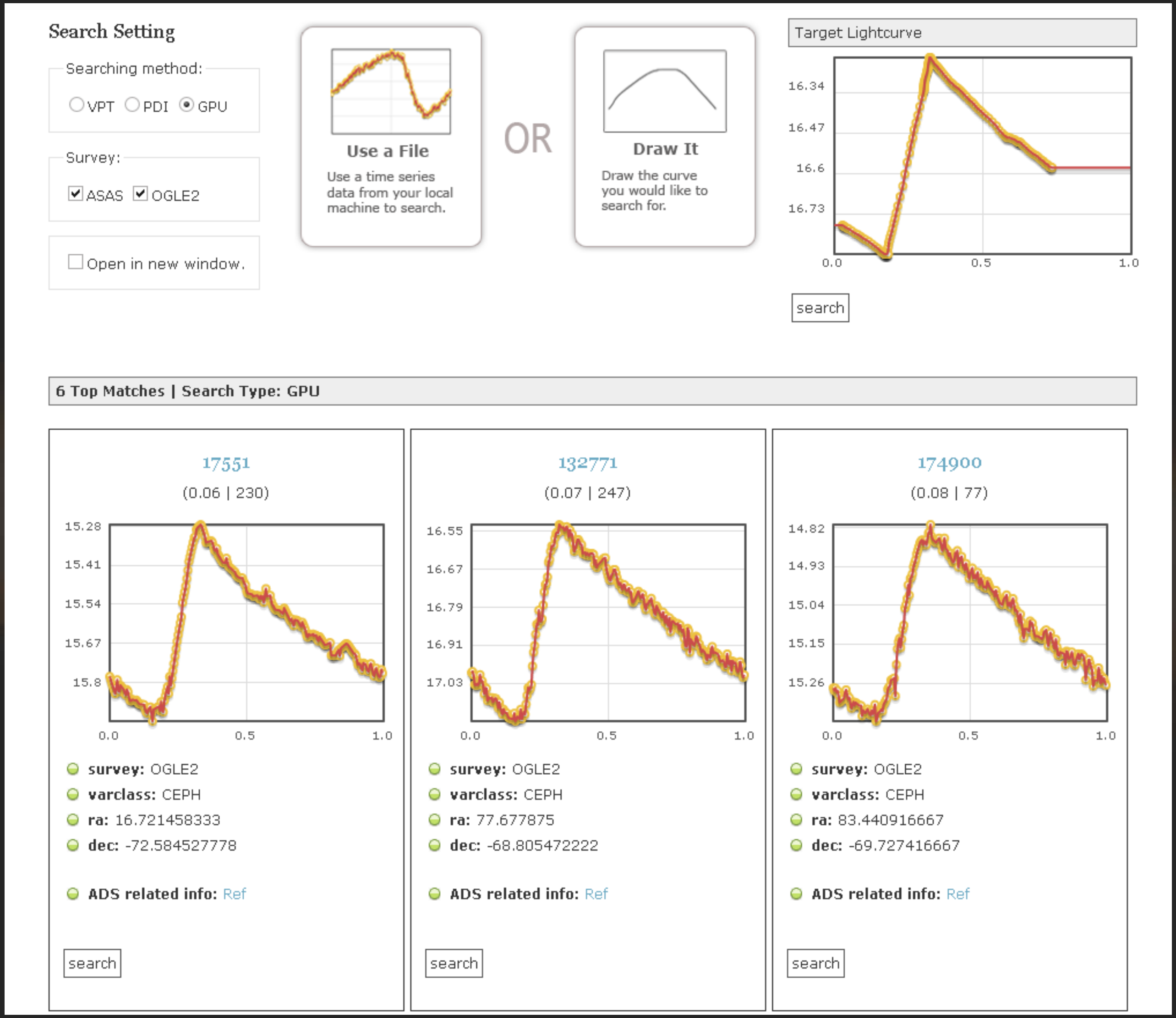
GPU Speed-Up:

GPU naïve vs GPU with indexing



= GPU without Indexing
= GPU with Indexing

Demonstration.



Conclusion.

Searching Massive Time Series Databases

- GPU-aided Searching is fast
  - Faster than CPU methods
  - Highly parallelizable
- Vantage Point placement is crucial
  - Better VP placement == Reliably fast searching

Questions ?