Massively Parallel Fourier-Space Cross-Correlation: Analyzing Highly Dimensional Time Series Databases

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

November 8th 2010  ADASS XX Boston
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

\[ d = \text{Dist}(\text{VP3}, \text{Query}) \]
Tree Structure Search:

Bonsai Tree

```
+---+---+---+
|   |   |   |
+---+---+---+
     |     |
+---+---+---+
| VP₀ | VP₁ | VPₙ |
+---+---+---+
      |     |
+---+---+---+
| 54  | 123 | 223 |
| 0.0 | 0.021 | 0.115 |
| 1245 | 15248 | 15248 |
+---+---+---+
```

```
+---+---+---+
|   |   |   |
+---+---+---+
     |     |
+---+---+---+
| 54  | 123 | 223 |
| 0.0 | 0.021 | 0.115 |
| 1245 | 15248 | 15248 |
+---+---+---+
```
GPU Speed-Up:
GPU naïve vs CPU with indexing: Parallel vs Serial

GPU (naïve) vs CPU (with PDI)
Time to search Light Curves

Time taken in ms

Number of Light Curve Compared

1.747413 3.89744 27.62073 42.78502 69.16129 119.0859 165.0271

November 8th 2010 ADASS XX Boston
GPU Speed-Up:
GPU naïve vs GPU with indexing

![Graph showing GPU speed-up comparison between naive and with indexing.]
Demonstration.

Search Setting
- Searching method: VPT, PDL, GPU
- Survey: ASAS, OGLE2
- Open in new window.

Target Lightcurve
- Use a File: Use a time series data from your local machine to search.
- Draw It: Draw the curve you would like to search for.

6 Tap Matches | Search Type: GPU
- 17551 (0.06 | 230)
  - survey: OGLE2
  - varid: CEPH
  - ra: 16.721458293
  - dec: -72.584827778
  - ADS related info: Ref
- 132771 (0.07 | 247)
  - survey: OGLE2
  - varid: CEPH
  - ra: 72.6778075
  - dec: -68.859472222
  - ADS related info: Ref
- 174900 (0.08 | 77)
  - survey: OGLE2
  - varid: CEPH
  - ra: 23.940918667
  - dec: -65.727410067
  - ADS related info: Ref

Search
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?