Prospects of GPGPU in the Auger Offline Software Framework

Marvin Gottowik, Julian Rautenberg and Tobias Winchen, for the Pierre Auger Collaboration

GPU Computing in High Energy Physics
Pisa, September 2014

winchen@uni-wuppertal.de
Cosmic Ray Energy Spectrum

- Galactic (SNR): 10,000 per m² and second
- Extragalactic: 1 per m² and year
- 1 per km² and century
Cosmic Ray Induced Air Showers

- Particle Cascade
  - \( \sim 10^{10} \) particles \( 10^{19} \text{ eV} \)
  - Extend over km scale
- Electrons excite air molecules which emit fluorescence light
- Shower geometry and particle content allows conclusions on energy, direction and nature of primary particles
The Pierre Auger Observatory

Surface Detector
1660 Water Cherenkov stations
1.5 km spacing
3000 km² covered area

Fluorescence Detector
27 telescopes at
4 sites with 180° view
Radio Emission from Cosmic Ray Air Showers

Geomagnetic Emission

Charge Excess
Auger Engineering Radio Array (AERA)

124 Stations with 2 antennas (NS, EW)

Different Antenna Types

Bandwidth 30 – 80 MHz
Digitizing with 200 MHz

Science Goals:
Evaluate Radio Technology
Understand Radio Emission Composition Measurement

(...)
Super Hybrid Events

Surface Detector

Fluorescence Detector

Radio Detector
The Auger Offline Framework

Configuration

Detector Description
- Observatory
  - Fluorescence
  - Surface
  - Atmosphere
  - Radio

Algorithms
- Module A
- Module B
- Module G

Event Data
- Event
  - Fluorescence
  - Surface
  - Radio
  - Air Shower

Utilities
Radio Integration in Offline
Reconstruction of Radio Events

- N Stations
- Noise Filters + Signal Enhancing
- Voltage Traces
  - Estimate Peak Timings
  - Update Peak Timings
- Envelope + Peak Time
  - Calculate Hilbert Transform
  - Interpolate Antenna Pattern
- Shower Direction
- Shower Properties
- E-Field Vector E(t)
  - Fold Antenna + Signal
- Directional Antenna Properties

GPGPU in the Auger Offline Software Framework
Tobias Winchen for the Pierre Auger Collaboration
Profiling

Tools

Google-perftools + kCachegrind
Valgrind + kCachegrind
Intel VTune
Linux kernel profiler (perf)

Notes

Free
Free, Slow
Proprietary
Free

No difference in conclusions in this application

Top Hotspots

FFT ~ 15 %

Interpolation of Antenna patterns ~ 25 %

Other (max 5%)

→ Minimum invasive Approach: Move individual Hotspots on GPU
Offline FFT Data Container:
- Stores data in time and frequency domain
- Lazy evaluation of FFT to update time (frequency) after modification of frequency (time)

1D FFT, 2D FFT, ...

FFT++
(Interface to FFTW)

CuFFT++
(Interface to CuFFT)
Hilbert Envelope

Envelope is squared sum of signal and its Hilbert Transform

\[ E(t) = \sqrt{x^2(t) + H^2(x(t))} \]

Hilbert Transform is \((- (+)\) 90 degree phase shift for first (second) half of spectrum

\[ H(\omega) = -i \text{sgn}(\omega - \omega_{\text{mid}}) x(\omega) \]
Hilbert Envelope

Envelope is squared sum of signal and its Hilbert Transform

\[ E(t) = \sqrt{x^2(t) + H^2(x(t))} \]

Hilbert Transform is - (+) 90 degree phase shift for first (second) half of spectrum

\[ H(\omega) = -i \, \text{sgn}(\omega - \omega_{\text{mid}}) \, x(\omega) \]

=> Time spend in FFT negligible in Cuda - Version
Interpolation of Antenna Patterns

Get Efield from Voltage Traces: \[ U = \mathbf{H} \cdot \mathbf{E} \]

\[ \mathcal{E}_\theta(\omega) = \frac{\mathcal{V}_1(\omega) \mathcal{H}_{2,\phi}(\omega) - \mathcal{V}_2(\omega) \mathcal{H}_{1,\phi}(\omega)}{\mathcal{H}_{1,\theta}(\omega) \mathcal{H}_{2,\phi}(\omega) - \mathcal{H}_{1,\phi}(\omega) \mathcal{H}_{2,\theta}(\omega)} \]

\[ \mathcal{E}_\phi(\omega) = \frac{\mathcal{V}_2(\omega) - \mathcal{H}_{2,\theta}(\omega) \mathcal{E}_\theta(\omega)}{\mathcal{H}_{2,\phi}(\omega)} \]

\( \mathcal{H}_{\ldots} \in \mathbb{C} \)
Interpolation of Antenna Patterns

- Few (~6) independent Patterns
- 2 Channels / Pattern
- ~ 80 frequencies, 180 x 90 angles
- Theta / Phi Component Complex Numbers
- Linear interpolation

- Bind Antenna Patterns as textures on GPU
- Use texture interpolation
- > 100x Speedup
Test Systems

Cluster

- 24x Intel Xeon X5650, 2.67GHz
- 48 GB Ram
- 4x Tesla M2090
- Debian GNU/Linux (stable)
- Cuda 4.2

Desktop

- 1x AMD A8-6600K, 3.9 GHz
- 8 GB Ram
- 1x GeForce 750 Ti
- Debian GNU/Linux (stable)
- Cuda 6.0
Performance Overview

Total Speedup
~ 1.5x on Cluster with Intel Xeon X5650 @ 2.7 GHz / Tesla M2090, Cuda 4.2
~ 1.9x on Desktop with AMD A8-6600K / GeForce 750 Ti, Cuda 6.0

Top hotspots have been eliminated
Conclusions on GPGPU in Auger Offline

- Implementation of GPU versions for selected bottlenecks in parallel to existing CPU version with minimum modifications of the code possible:
  - Replacement of FFTW with CuFFT
  - Interpolation of Antenna patterns as textures
- Implementation not optimal, but minimum invasive
- GPU implementations eliminate two main hotspots: Speedup ~ 1.9x on Desktop PC
- High benefit from GPU on Desktop with entry level GPU