Parallel computing with (GP)GPUs
and its applications to astrophysics

Denis Bastieri
Mimesis HPC - Università di Padova
October 20th 2011 @ICRA
Outline

- Why GPGPU
- Fermi LAT Standard Analysis
  - \textit{gpul}tcube: standalone tool
  - \textit{gpulike}: GPU-aware tool
- Shower simulations
- HEP Triggering
- Conclusions

NVIDIA S2050: 1U 4*(448 cores, 3GB)
Urania: host (2U) + 2*S2050
Why (GP)GPU?

• GPU in abstract on the entire arxiv:
  – 69 (-> 2009), 90 (2010), 108 = 72/2*3 (2011)

• CPU versus GPU
  – Intel Core i7 980 XE (6 cores): 109 GFLOPS
  – NVIDIA C2050 (448 cores): 515 GFLOPS

• Cheap parallel computing!
  – FLOPS/$ and FLOPS/W
CPU versus GPU: GFLOP/s
GFLOPs/W, GFLOPs/$

Bloomberg: Bond Pricing

48 GPUs
$144K
$31K / year

42x Lower Space
28x Lower Cost
38x Lower Power Cost

2000 CPUs
$4 Million
$1.2 Million / year
Why not earlier?

• Computations were not always correct.
  – 512 cores (VG cards) -> 448 cores + ECC (Tesla/Fermi)

• Good for integer computation
  – GPGPU 8×ALU standard VG-GPU

• Programming: Not an easy task
  – from Cg to CUDA (and thrust)

• Not enough memory
  – now: 3 to 6 GB/card. UVA: 4 cards on the same PCI-e
http://mimesis.pd.infn.it/

Computing facilities

Urania  2 x Xeon E5620  2 x NVIDIA S2050  3584 GPU cores - 24 Gb RAM
... men who have been instructed of her she raises aloft to heaven (ouranos), for it is a fact that imagination and the power of thought lift men's souls to heavenly heights ...

Clio  2 x Xeon E5620  4 x NVIDIA GTX 580  1920 GPU cores - 12 Gb RAM
... the praise which poets sing in their encomia bestows great glory (kleos) upon those who are praised ...

Euterpe  2 x Xeon E5620  1 x NVIDIA S2050  1792 GPU cores - 24 Gb RAM
... she gives to those who hear her sing delight (terpein) in the blessings which education bestows ...
LAT Analysis: overview

• Standard Maximum Likelihood approach:
  – Create a model
  – Compute the likelihood of your model with respect to the detected gamma rays

• Steps (standard analysis):
  – Selection: data stored in FITS format.
    *gtselect*: apply required selection
  – Evaluate likelihood: could be factorized.
LAT Analysis: steps

The unbinned likelihood

0. prepare the (XML) model
1. gtselect: apply desired cuts
2. gtltcube: compute the *livetime cube*.
3. gtexpmap: compute the *exposure map*.
4. gtlike: compute the likelihood

BUT, it takes time! gtltcube 1h/2yr
Ultrafast Robotic Interface for Extended Likelihood

Denis Bastieri: astro
Andrea Pigato: web/daemon
Giorgio Urso: CUDA
The sw/hw architecture

1. Allot one GPU card for ltcube: FT2 (spacecraft) data resident in memory, output to...

2. ... the same GPU card allotted for expmap: FT2 data resident in memory, output ltcube and expmap to...

3. ... three different cards on the same PCI-e storing FT1 data, where gpulike will select data and compute the likelihood of the model(s) fed.
gpultcube: standalone tool

- Livetime cube computation
  - number of seconds under which a given direction of the sky is observed under a given angle ($\theta$)
  - Using a HEALPix of 1° (=12×64×64 = 48k)
  - 40 bins in cos($\theta$) (4 $\phi$-bins foreseen)
  - Use 30s - FT2 file (every rows describing the spacecraft in a given interval)
  - Take care of GTIs (and FT2) (#include <thrust/unique.h>)

D. Bastieri - GPUs and Astrophysics - ICRA
HEALPix

- Hierarchical Equal Area isoLatitude Pixelization of a sphere
- Order 64
  => width = 90°/64 = 1.4°
- 12×64×64 pixels (48k)
- 40 HEALPix matrices (float)
- Read all FT2 for every elements -> threads
• FT2 entries: [start, stop], ra\(_z\), dec\(_z\), lt, wlt
• For every GTI (10k)
  – GTI couples \textit{cached} in shared
  – flag FT2 in GTI
    \texttt{kGTI}<<<\texttt{nGTI/512+1, 512>>> \\
• For every lt-bin (HPix, \cos(\theta))
  – Compute the angle between HPix and FT2 entry (entirely loaded into GPU memory).
  – Update the proper \cos(\theta):
    • \texttt{cosbin} = 40*\sqrt{1-\text{dot}};
    • \texttt{Ltcube}[threadID, cosbin] += lt;
      \texttt{wltcube}[threadID, cosbin] += wlt;
    \texttt{kEval2}<<<96, 512>>>

The kernels

October 20th, 2011 D. Bastieri - GPUs and Astrophysics - ICRA 14
Likelihood ratio test

• Evaluate the *free* parameters in order to maximize the likelihood ratio with & without source.

• gtselect: 10-yr mission = 2 Gγ = 200 GB !!

• New in Cuda 4.0: unified access: 40 bit => 1 TB

• Use `newminuit` to compute ratio.

• Feed `newminuit` with the actual *value* of the likelihood and its *gradient*.

• Report the values and compute the *TS* value of the input model
• Typically: 1 to 10 $M\gamma$/year inside ROI

• For every detected photon $\gamma$ (oν $\Gamma\Pi\Upsilon$)
  – Compute the probability that it is originated from the model.
  For every source in the model:
    • $\delta = \text{dist}[(RA_{\gamma}, \text{dec}_{\gamma}), (RA_{\text{src}}, \text{dec}_{\text{src}})]$
    • loglike $+= \text{efficiency}$
      * $\text{spectrum}_{\text{src}}(E)$
      * $f(\delta, \text{PSF}[5](\gamma))$
      * $\text{Aeff}(\gamma)$

• In parallel (on CPU)
  – add the Npred from every source
  – add the prior contributions

• Synchronize everything
Files in output from URIEL

- feed.py (gtselect, gtmktime, gtbin): name.{ft1.fits, ft2.fits, cmap.fits, cmap.png}
- GPU: name.expCube.fits
- GPU: name.expMap.fits
- GPU: likelihood evaluation
- wrapup.py: name.results.fits
  - EXT#0: input info (+ Minimiser outputs)
  - EXT#1,2,3: Counts, Fluxes, Ebounds
  - EXT#4 ?: results.dat
- name.model.xml, name.model.yml
- name.results.png (fluxes, counts, residuals)
DB Wrap™

• gtselect: 10-yr mission = 2 $G\gamma = 200$ GB !!

• New in Cuda 4.0: unified access: 40 bit => 1 TB

• **DB Wrap™**: SQL-complying, DB-agnostic, middle-layer, GPU<->DB interface.

• Performs SQL-select launching a **kernel** on each selected **entries**
DB Wrap: implementation

- 1 TB too expensive & not needed for LAT
- Reduce row size (time, ra, dec, energy, quality)  
  \[ \Rightarrow 92 \text{ B/entry} \rightarrow 20 \text{ B/entry} \rightarrow 4\text{GB/year} \]
- Select only good quality data (0.1\times)  
  \[ \Rightarrow (20 \text{ GB})/(10 \text{ year}) \text{ fits in } 6 \text{ (or 4) GPU cards} \]
- Everything driven by a single host!
URIEL: Performance

- feed.py:
  -> ft1, ft2, cmap (.png)
  FT2: already in GPU
- Mkn 180 for 2 years
- ltcube:
  - CPU: 01:10:37
  - GPU: 00:00:43 (single card)
- gtexpmap: testing
- adding support for likelihood evaluation
Suspend

- Fermi data downlinked and delivered to SLAC every 3h:
  - Fermi put 800 cores in the general batch queues
  - We use them for <1h every 3h:
    - The rest of the time you can use them
  - But when we need them we suspend any job already running on our cores.
  - Why?
Level1Proc: Main requirements

• Kalman filtering
  – Or other image processing techniques (even better)

• Classification trees
  – CUDA repository has examples (even RF)
  – Good performance
    • Tree instruction
    • Tree traversing
MC for CTA

Cherenkov Telescope Array: flagship experiment in Europe, also US/China/Japan. US: Stanford & SLAC. Average production time for a shower: 40s on CPU (data provided by the collaboration). GPU first trial << 1s!! To be checked.
Conclusions

GPU dominated by CPU <-> GPU <-> CPU

1) Combinatorics in GPU ?
2) More GPUs ?
3) Streaming capabilities ?
CUDA Spotlight: GPU-Accelerated Astronomy

This week's spotlight is on Dr. Denis Bastieri of the University of Padua, Italy and co-founder of Mimesis HPC. Dr. Bastieri leads the Fermi Large Area Telescope (LAT) team for the National Institute of Nuclear Physics (INFN) in Padua.

NVIDIA: Denis, Tell Us About The Fermi Space Telescope.
Denis: The Fermi mission is part of NASA's focus on the theme of "Structure and Evolution of the Universe." I specifically work with one of the two instruments aboard the Fermi spacecraft -- the Large Area Telescope (LAT), which observes gamma rays, the electromagnetic radiation with the highest energy.

The subjects of the Fermi mission are no ordinary stars. The closest star to Earth, the Sun, cannot emit gamma rays at high enough energies to be detected by the LAT (unless a bright solar flare is occurring). What we are observing are collapsed stars like black holes and pulsars, stars as heavy as our Sun but with a typical size of only a few kilometers (smaller than the SF Golden Gate Bridge!), spinning around their axes many times per second, as well as far away galaxies and some Gamma Ray Bursts (GRBs), the most energetic explosions (second only to the Big Bang).
Conclusion

• Data analysis or MC (data are quasi-independent)
  --> acceleration > 100×

• Best CUDA 4 features:
  UVA: implemented for FT1 big files (8GB on 3 cards)
  CURAND: uniform, normal
  thrust: sorting, “uniquify”

• Future:
  minimiser?
  streaming?