Denis Bastieri: astro
Andrea Pigato: web/daemon
Giorgio Urso: CUDA

Friday SCA Meeting
August 5th, 2011
SLAC

Ultrafast
Robotic Interface
for Extended Likelihood

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.

YAML: input for URIEL

```
uriel:
    fermi_lat_catalog_name:  string
    source_name:             string   # req’d
    source_ra:               number
    source_dec:              number
    source_model:            string
    roi_radius:              number
    source_region_radius:    number
    free_source_radius:      number
    t_start:                 number
    t_stop:                  number
    e_min:                   number
    e_max:                   number
    irf:                     string
    z_max:                   number
    event_class:             number
    conversion_layer:        number
    ts_upper_limits:         number
    n_refit:                 number
    n_refit_ts_min:          number
```

Files in output from URIEL

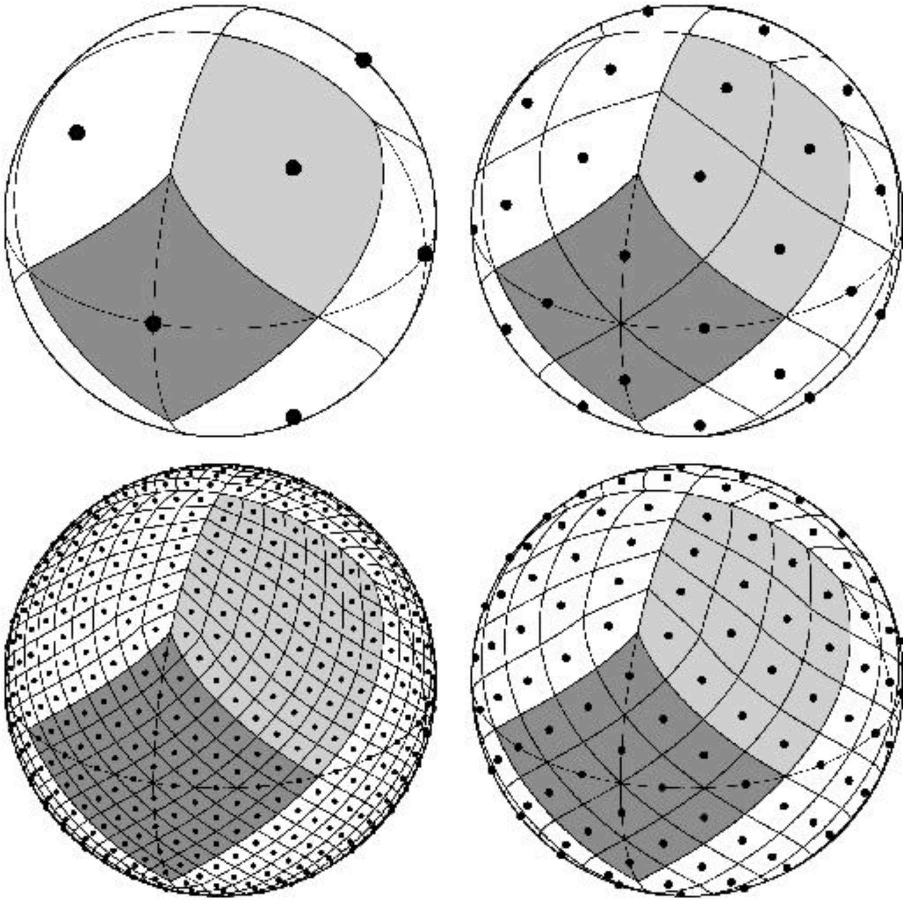
- feed.py (gtselect, gtmktime, gtbin): name.{ft1.fits, ft2.fits, cmap.fits, cmap.png}
- GPU: name.expCube.fits
- GPU: name.expMap.fits
- GPU: likelihood evaluation
- wrapup.fits: name.results.fits
  - EXT#0: input info (+ Minimiser outputs)
  - EXT#1,2,3: Counts, Fluxes, Ebounds
  - EXT#4 ? : results.dat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

Factorizing elements

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

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 -> *thread*



The kernels

- FT2 entries: [start, stop], ra_z, dec_z, lt, wlt
- For every GTI (10k)
  - GTI couples *cached* in shared
  - flag FT2 in GTI

```
kGTI<<<g_currentDbSize/512+1, 512>>>
```

- For every lt-bin (HPix, cos(θ))
  - Compute the angle between HPix and FT2 entry (entirely loaded into GPU memory).
  - Update the proper cos(q):
    - `cosbin = ceil(40E-6*rintf(1E6*sqrt(1-dot)));`
    - `ltcube[ += lt; wltcube += wlt;`

```
kEval2<<<96, 512>>
```

URIEL: Performance

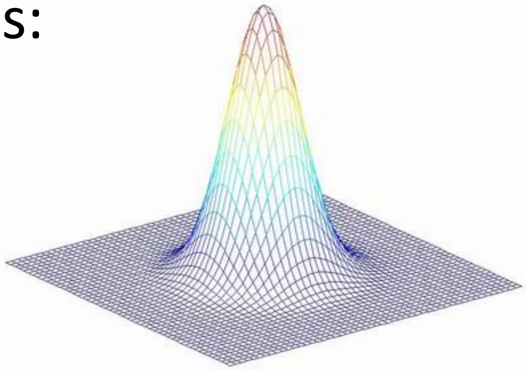
- feed.py:
  -> ft1, ft2, cmap (.png)
  FT2: already in GPU
- Mkn 180 for 2 years
- ltcube:
  - **CPU: 01:10:37**
  - **GPU: 00:00:43 (single card)**
- gtexpmap: testing
- adding support for likelihood evaluation

YAML ROI modeling

```
roi:
  - name: 2FGLJ1801.7+4405
    spatialModel:
      type: pointlike
      ra:    270.434
      dec:   44.0856
    spectrum:
      type: PowerLaw2
      integral:   {free: true,  scale: 1e-06, value: 0.00440}
      index:      {free: true,  scale: -1,    value: 2.66203}
      LowerLimit: {free: false, scale: 1,     value: 100}
      UpperLimit: {free: false, scale: 1,     value: 300000}
  - name: iso_p7v6source
    spatialModel:
      type: isotropic
    spectrum:
      type: file
      file: $GLAST_EXT/diffuseModels/v2r0/isotrop_2year_P76_source_v0.txt
      normalization: {free: true, min: 1e-05, scale: 1, value: 1}
  - name: gal_2yearp7v6_v0
    spatialModel:
      type: cube
      file: $GLAST_EXT/diffuseModels/v2r0/ring_2year_P76_v0.fits
    spectrum:
      type:     PowerLaw2
      integral: {free: true,  scale: 1e-06, value: 0.00440}
      index:    {free: true,  scale: 1,     value: 0}
```

The kernel

- For every detected photons
  - Compute the probability that it is originated from the model
  - Easy to do for point sources:
    PSF ~ 2D-Gauß
    $\delta = \text{dist}[(RA_\gamma, dec_\gamma),$
    $(RA_{src}, dec_{src})]$
    $prob(RA_\gamma, dec_\gamma) = \exp(-\delta^2)$



  - Multiply all *p* together
    (actually *sum*: better using (−)log)
- Typically: 1Mγ/year

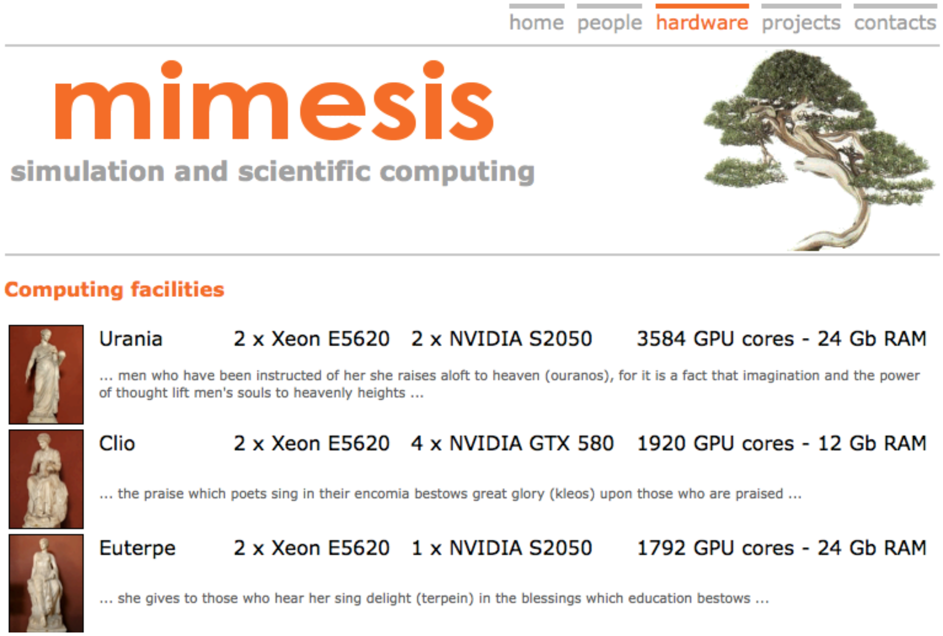
Likelihood ratio test

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

Conclusions

- Uriel is alive!
- ✓ daemon + YAML
- ✓ feed.py (!png)
- ✓ ltcube
- expmap (some checks)
- ✓ minuit interface
- actual kernel
- wrapup.py

http://mimesis.pd.infn.it/



Computing facilities

Urania	2 x Xeon E5620	2 x NVIDIA S2050	3584 GPU cores - 24 Gb RAM
Clio	2 x Xeon E5620	4 x NVIDIA GTX 580	1920 GPU cores - 12 Gb RAM
Euterpe	2 x Xeon E5620	1 x NVIDIA S2050	1792 GPU cores - 24 Gb RAM

Urania: ... 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: ... the praise which poets sing in their encomia bestows great glory (kleos) upon those who are praised ...

Euterpe: ... she gives to those who hear her sing delight (terpein) in the blessings which education bestows ...