

GammaLib - Action #1980

Improve computation speed of stacked response computation

03/31/2017 04:40 PM - Knödlseeder Jürgen

Status:	Closed	Start date:	03/31/2017
Priority:	High	Due date:	
Assigned To:	Knödlseeder Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.4.0		
Description			
<p>The computation speed for the stacked response computation should be improved as this is a central to the setup of an analysis. The typical use case for this should be the computation of the stacked response for a large observation definition XML file, such as the one of the GPS KSP.</p> <p>For example, the <code>GCTACubeBackground::fill</code> method should test whether there is an overlap between the background cube and the RoI of an observation. For the moment it is checked whether each bin of an event cube is within an RoI, but by checking first whether an RoI overlaps with the event cube one can considerably speed up the computations.</p> <p>The same is true for the other cube computation methods. What is needed is a method that checks whether a sky map overlaps with an RoI, since all cubes are computed using sky maps.</p> <p>A method could for example be added to <code>GSkyMap</code> to return the bounding circle of a map. This method would supersede some computations in <code>GModelSpatialDiffuseMap::prepare_map</code>. A possible name for this method would be <code>GSkyMap::bounding_circle</code>, the method should return a <code>GSkyRegionCircle</code> object. Or more generally, have <code>GSkyMap::bounding_regions</code> returning a <code>GSkyRegions</code> object which can hold a set of combined regions. The <code>GSkyRegions::overlaps</code> method can then be used to check whether the bounding regions overlap with another region (or set of regions; but for this a new <code>GSkyRegions::overlaps</code> method needs to be added that accepts a region container).</p>			
Related issues:			
Related to GammaLib - Action # 2150: Assure that <code>GSkyMap::overlaps()</code> always r...		New	07/06/2017

History

#1 - 06/06/2017 04:21 PM - Knödlseeder Jürgen

- Target version deleted (1.3.0)

#2 - 06/15/2017 04:21 PM - Cardenzana Josh

- Status changed from New to Pull request

- % Done changed from 0 to 50

I believe the above has implications for the following CTools: `ctbin`, `ctbkcube`, `ctediscube`, `ctexpcube`, `ctmodel`, `ctpscube`, `ctskymap`

I've implemented some code to do a check on each observation inside a supplied XML observation list overlaps the sky map to be generated. The code that does this is located in '`GSkyMap::overlaps(const GSkyRegionCircle&)`'. The check is done by first testing whether the central pointing position of the observation falls within the bounds of the sky map. If this is false, then pixel positions that border the sky map are tested for whether or not they fall into the observation region. The positions tested can be visualized as follows (for a 4x3 bin skymap), where "" marks the positions tested.

```
* * * * *
+---+---+---+
* |0,2|1,2|2,2|3,2| *
+---+---+---+
* |0,1|1,1|2,1|3,1| *
+---+---+---+
* |0,0|1,0|2,0|3,0| *
+---+---+---+
* * * * *
```

Quickly testing the performance for `ctbin` and `ctbkcube` using the latest 1dc data give the following:

ctbin debug=yes inobs=../1dc.south/obs/obs_gps_baseline.xml nxpix=400 nyypix=400 binsz=0.02 xref=0.0 yref=0.0 coordsys=GAL proj=CAR
emin=0.1 emax=100 enumbins=30 ebinalg=LOG outcube=ctbin_test.fits

devel:

* 1011.93 seconds of CPU time
* Number of observed events .: 5530542
1980-skymap_overlap:
* 199.404 seconds of CPU time
* Number of observed events .: 5530542

ctbkgcube

devel:

* 7475.55 seconds of CPU time
1980-skymap_overlap:
* 2320.61 seconds of CPU time

The bottle neck in the computation now appears to be the looping over and computation of the model contribution for each bin. This would also explain why ctbkgcube does not see as much of a speed up as ctbin.

The relevant branches are (note this affects both gammalib and ctools):

joshcardenzana/gammalib: 1980-skymap_overlap
joshcardenzana/ctools: 1980-skymap_overlap

P.S.

I also found a bug where GCTAEventList::dispose() was not actually freeing the memory when it cleared its m_events vector. I've also fixed this which dramatically reduces the amount of memory used by some of the ctools. The run of ctbin using the devel branch above used more than 30 GB of memory. After the changes, the same uses less than 1 GB.

#3 - 07/06/2017 11:09 AM - Knödlseher Jürgen

- Target version set to 1.4.0
- % Done changed from 50 to 90

I'm about to merge in the code.

I did some small modifications to the code since you introduced multiple return statements at various places. Each method should have a single exit point (see the coding conventions), so I modified the code in a way that the code always reached the return statement at the end.

#4 - 07/06/2017 11:40 AM - Knödseder Jürgen

I think there are two problems with the `GSkyMap::overlaps` method:

- if the diameter of the circular region is smaller than the pixel spacing there are cases where the centre of the region is falling outside the map but the circle is overlapping with the map which are not caught (since the method "samples" at pixel locations). It probably would be better to sample locations of the region, but also this is tricky since you can conceive a region that is bigger than the map with the centre lying outside the map.
- the method will not work for HealPix projections

I agree that the method will do the job for the sake of excluding ROIs from the cube, but we need to think a bit more about how to implement a generic and save method for testing map overlap with regions.

The method should also work on all type of regions. Since we only have circular regions for now this is not a problem, but defining the method for a `GSkyRegion` object will make sure that the interface won't change in the future.

#5 - 07/06/2017 02:20 PM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

- *% Done changed from 90 to 100*

Merged into devel.

#6 - 07/06/2017 02:25 PM - Knödseder Jürgen

- *Related to Action #2150: Assure that `GSkyMap::overlaps()` always returns exact response added*

#7 - 07/06/2017 02:25 PM - Knödseder Jürgen

I created issue #2150 to track the fact that the `GSkyMap::overlaps()` needs to be improved.