

ctools - Action #2189

ctmodel - only loop on models overlapping observations

08/30/2017 05:12 PM - Cardenzana Josh

<b>Status:</b>	Closed	<b>Start date:</b>	08/30/2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Cardenzana Josh	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	1.00 hour
<b>Target version:</b>	1.5.0		
<b>Description</b>			
In ctmodel, the evaluation loop each model is looped over and evaluated regardless of whether or not it actually overlaps with the current observation. The convolution of the model with the various IRFs is expensive, so screening out models that dont overlap would yield a significant speedup in computation time when many models are involved.			

History

#1 - 09/01/2017 02:41 PM - Cardenzana Josh

- Status changed from New to Pull request
- % Done changed from 0 to 90

This has been implemented now. I introduced a new method "ctmodel::trim\_models(const GModels&, const GCTARoi&)" which takes in a model object representing every model in the passed xml file. It then picks only the models that fall within the region of interest and returns a new model container with these models. Running this on a sample of detected sources in the 1DC EGS survey consisting of about 35 models (lots of models, but every observation only overlaps 1-2 sources), the speedup is about a factor of 10. The updated code runs in about 12 minutes, when parallelized over 4 cores.

In terms of the region of interest, there are a couple of notes regarding how this is handled:

- A small buffer of 0.5 degrees is added to the circular region of interest. This is done since the region definitions for point sources have a radius of 0.0 degrees. Without a buffer, all point sources falling just outside the observation region would be excluded, which is not what we want.
- When the passed 'inobs' is an events cube, then the region of interest is not valid. In that case every model would end up being cut from the model list. Instead, when this happens no models are cut from the model list.

Pull branch  
Josh Cardenzana / ctools: 2189-ctmodel\_trim\_models

#2 - 09/15/2017 02:57 PM - Cardenzana Josh

Since this falls along the same guidelines of the original post and I dont believe this change has been merged into the devel branch yet, I'm going to tack an additional change on here.

There are two more updates to ctmodel that are relevant to increasing it's speed:

1. The original parallelization of the method resulted in creating an additional copy of the GModels container. However, if a diffuse cube model was included the model would have to be reloaded for each observation. Loading a large diffuse cube (like the diffuse background cube in the 1DC) was a huge computational overhead. The copy is now made before looping over the observations so that it is evaluated only once.
2. The 'trim\_models' method now removes models outside of an observation and re-adds them after evaluating for a given observation. This was necessary for the above update. There is also a check on whether the model is a diffuse map, in which case the model is always left in the list of observations as it's assumed (for now) that the copying of the cube back and forth is a larger overhead than just evaluating the model.

Pull branch  
Josh Cardenzana / ctools: 2189-ctmodel\_trim\_models

**#3 - 10/11/2017 01:44 PM - Knödlseider Jürgen**

- *Status changed from Pull request to Closed*
- *Target version set to 1.5.0*
- *% Done changed from 90 to 100*

Merged into devel