ctools - Feature #1797

Write cscript to create sliced cttsmap commands

06/20/2016 01:48 PM - Mayer Michael

Status:	Closed	**Start date:**	06/30/2016
Priority:	Normal	**Due date:**	
Assigned To:	Mayer Michael	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.1.0		

Description

cttsmap currently supports to compute chunks of a TS map in individual jobs (binmin, binmax parameters). In order to allow a coherent way of splitting the computation into several chunks a cscript could be implemented. The tool could have the following parameters to query:

```
$ cstssplit
Input event list, counts cube or observation definition XML file [selected_obs.xml]
Input model XML file [model.xml]
Test source name [TestSource]
Output Test Statistic map file [tsmap.fits]
Size of the X axis in pixels [100]
Size of the Y axis in pixels [100]
Image scale (in degrees/pixel) [0.02]
First coordinate of image center in degrees (RA or galactic l) (0-360) [83.6331]
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [22.01]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [CEL]
Projection method (AIT|AZP|CAR|MER|MOL|STG|TAN) [TAN]
Number of bins per chunk [10]
Compute null hypothesis first? [yes]
Output file holding commands [compute_tsmaps.txt]
```

The first parameters are, obviously, for feeding cttsmap. The last 3 parameters would determine how the jobs are splitted. The script could compute the null hypothesis first and pass the outcome value to the individual jobs (thus each job wouldnt need to compute the same thing again).
Ultimately, the commands could be written into an ASCII file which looks like this:

```
cttsmap inobs=selected_obs binmin=0 binmax=9 outmap=tsmap_0.fits ...
cttsmap inobs=selected_obs binmin=10 binmax=19 outmap=tsmap_1.fits ...
...
```

The user would simply have to submit those commands to a batch system to parallelise the computation to the grid she/he has access to. Does this make sense, or is this too specific to go in ctools?

History

#1 - 06/21/2016 03:36 PM - Knödlseder Jürgen

What is the advantage of having the commands in an ASCII file over implementing OpenMP support for cttsmap?

#2 - 06/21/2016 03:44 PM - Mayer Michael

I agree that OpenMP support is something good to implement. Nevertheless, when computing a TS map with large observation containers, one might want to split the computation into 1000 jobs or more (since one fit might take an hour or more). The parameter optimisation already uses OpenMP within a job to achieve the maximum speed.
When running a TS map on 100x100 pixels with 10 bins computed in job, I don't think OpenMP will help here a lot since all threads will be taken by the observation optimisation. Of course you never know what kind of CPUs we will have at hand in the future but for now 1000 jobs need to be executed on different machines to gain speed.

#3 - 06/30/2016 04:34 PM - Mayer Michael

- Status changed from New to Pull request

- Assigned To set to Mayer Michael

- Target version set to 1.1.0

- Start date set to 06/30/2016

- % Done changed from 0 to 100

I have created and added cstssplit on branch *1797-cstssplit*
I have also added the necessary documentation and unit tests which run smoothly.

#4 - 07/01/2016 03:39 PM - Knödlseder Jürgen

I'm about to integrate the script. I have renamed the script to cstsmapsplit since it is for TS map computation (this name is also consistent with cstsmapmerge).

I assume that it would also be useful that the script will produce an ASCII file that includes all TS maps and that can be put as input file into cstsmapmerge. Otherwise the user will have to create such a file manually.

The integrated script is so far in the integration branch, will move to devel once I fixed some minor problems with the unit test.

#5 - 07/01/2016 03:51 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Merged into devel.