GammaLib - Action #3404

Implement a sparse vector cache in GResponse

10/17/2020 12:17 AM - Knödlseder Jürgen

Status:	Closed	**Start date:**	10/17/2020
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0.0		

Description

The GVector GResponse::irf_spatial(const GModelSky& model, const GObservation& obs, GMatrix* gradients) has so far no response cache for spatial models without free parameters.

For these models, a sparse vector cache should be implemented that stores efficiently the model.

Related issues:

Related to GammaLib - Action # 3316: Implement a specific response cache for ...	**Closed**	**08/07/2020**
Related to GammaLib - Feature # 3312: Implement a source distance cache in GC...	**Closed**	**08/06/2020**

History

#1 - 10/20/2020 02:25 PM - Knödlseder Jürgen

- Subject changed from Implement a spare vector cache in GResponse to Implement a sparse vector cache in GResponse

- Status changed from New to In Progress

- % Done changed from 0 to 50

I implemented a new class GResponseVectorCache that handles a sparse vector cache and integrated this method in the GResponse class. Using some debug code, the following output is produced in model fitting:

```
2020-10-20T12:20:48: +==================================+
2020-10-20T12:20:48: | Maximum likelihood optimisation |
2020-10-20T12:20:48: +==================================+
GResponseVectorCache::contains(000001:Crab,800000): no entry found.
GResponseVectorCache::set(000001:Crab,800000): no entry found. Require 223392 elements.
2020-10-20T12:20:52:  >Iteration   0: -logL=-118103.118, Lambda=1.0e-03
GResponseVectorCache::contains(000001:Crab,800000): entry found at index 0 with 223392 elements.
2020-10-20T12:20:53:  >Iteration   1: -logL=-118105.707, Lambda=1.0e-03, delta=2.589, step=1.0e+00, max(|grad|)=14.159134 [Index:8]
GResponseVectorCache::contains(000001:Crab,800000): entry found at index 0 with 223392 elements.
2020-10-20T12:20:54:  >Iteration   2: -logL=-118105.707, Lambda=1.0e-04, delta=0.000, step=1.0e+00, max(|grad|)=0.039299 [Index:8]
GResponseVectorCache::contains(000001:Crab,800000): entry found at index 0 with 223392 elements.
2020-10-20T12:20:55:
2020-10-20T12:20:55: +========================================+
2020-10-20T12:20:55: | Maximum likelihood optimisation results |
2020-10-20T12:20:55: +========================================+
2020-10-20T12:20:55: === GOptimizerLM ===
2020-10-20T12:20:55:  Optimized function value ..: -118105.707
2020-10-20T12:20:55:  Absolute precision .....: 0.005
2020-10-20T12:20:55:  Acceptable value decrease .: 2
2020-10-20T12:20:55:  Optimization status .....: converged
2020-10-20T12:20:55:  Number of parameters ......: 11
2020-10-20T12:20:55:  Number of free parameters .: 4
2020-10-20T12:20:55:  Number of iterations ......: 2
2020-10-20T12:20:55:  Lambda .....................: 1e-05
2020-10-20T12:20:55:  Maximum log likelihood ....: 118105.707
2020-10-20T12:20:55:  Observed events  (Nobs) ...: 784404.000
2020-10-20T12:20:55:  Predicted events (Npred) ..: 784404.001 (Nobs - Npred = -0.00138538656756282)
2020-10-20T12:20:55: === GModels ===
2020-10-20T12:20:55:  Number of models ..........: 2
2020-10-20T12:20:55:  Number of parameters ......: 11
2020-10-20T12:20:55: === GModelSky ===
2020-10-20T12:20:55:  Name .......................: Crab
2020-10-20T12:20:55:  Instruments ................: all
2020-10-20T12:20:55:  Observation identifiers ...: all
```

2020-10-20T12:20:55: Model type ....: ExtendedSource
2020-10-20T12:20:55: Model components ....: "RadialGaussian" * "PowerLaw" * "Constant"
2020-10-20T12:20:55: Number of parameters ......: 7
2020-10-20T12:20:55: Number of spatial par's ...: 3
2020-10-20T12:20:55:   RA ....: 83.6331 [-360,360] deg (fixed,scale=1,gradient)
2020-10-20T12:20:55:   DEC ....: 22.0145 [-90,90] deg (fixed,scale=1,gradient)
2020-10-20T12:20:55:   Sigma ....: 0.2 [0.01,10] deg (fixed,scale=1,gradient)
2020-10-20T12:20:55: Number of spectral par's ..: 3
2020-10-20T12:20:55:   Prefactor ....: 5.66555268538365e-16 +/- 3.03971964494305e-18 [1e-23,1e-13] ph/cm2/s/MeV
(free,scale=1e-16,gradient)
2020-10-20T12:20:55:   Index ....: -2.47400882335043 +/- 0.00462307199887204 [-5,-0] (free,scale=-1,gradient)
2020-10-20T12:20:55:   PivotEnergy ....: 300000 [10000,1000000000] MeV (fixed,scale=1000000,gradient)
2020-10-20T12:20:55: Number of temporal par's ..: 1
2020-10-20T12:20:55:   Normalization ....: 1 (relative value) (fixed,scale=1,gradient)
2020-10-20T12:20:55: Number of scale par's .....: 0
2020-10-20T12:20:55: === GCTAModelCubeBackground ===
2020-10-20T12:20:55: Name ....: BackgroundModel
2020-10-20T12:20:55: Instruments ....: CTA, HESS, MAGIC, VERITAS
2020-10-20T12:20:55: Observation identifiers ...: all
2020-10-20T12:20:55: Model type ....: "PowerLaw" * "Constant"
2020-10-20T12:20:55: Number of parameters ......: 4
2020-10-20T12:20:55: Number of spectral par's ..: 3
2020-10-20T12:20:55:   Prefactor ....: 0.996990953510219 +/- 0.00232146317917687 [0.01,100] ph/cm2/s/MeV (free,scale=1,gradient)
2020-10-20T12:20:55:   Index ....: -0.00238033065721312 +/- 0.00133939959499067 [-5,5] (free,scale=1,gradient)
2020-10-20T12:20:55:   PivotEnergy ....: 1000000 MeV (fixed,scale=1000000,gradient)
2020-10-20T12:20:55: Number of temporal par's ..: 1
2020-10-20T12:20:55:   Normalization ....: 1 (relative value) (fixed,scale=1,gradient)
2020-10-20T12:20:55:
2020-10-20T12:20:55: +===============+
2020-10-20T12:20:55: | Save results |
2020-10-20T12:20:55: +===============+
2020-10-20T12:20:55: Model definition file .....: crab_results_test.xml
2020-10-20T12:20:55: Covariance matrix file ....: NONE
2020-10-20T12:20:55:
2020-10-20T12:20:55: Application "ctlike" terminated after 7 wall clock seconds, consuming 6.78244 seconds of CPU time.

A reference run done without the cache produced the same results. Note that the reference run consumed

2020-10-20T09:23:14: Application "ctlike" terminated after 17 wall clock seconds, consuming 16.5433 seconds of CPU time.

hence a speed-up by a factor of 2.4 was achieved.

#2 - 10/20/2020 04:24 PM - Knödlseder Jürgen

- % Done changed from 50 to 80

I implemented a unit test and a Python interface. The compilation of the Python interface leads to the following warning:

GResponseVectorCache.i:41: Warning 453: Can't apply (GVector *OUTPUT). No typemaps are defined.

I merged the code into devel but keep the feature open to fix the Python warning.

#3 - 10/21/2020 09:21 AM - Knödlseder Jürgen

- Related to Action #3316: Implement a specific response cache for diffuse models and binned data added

#4 - 10/21/2020 09:26 AM - Knödlseder Jürgen

- Related to Feature #3312: Implement a source distance cache in GCTAResponseCube added

#5 - 10/22/2020 05:30 PM - Knödlseder Jürgen

The ctools test failed in relation with the vector cache:

```
Test ctlike from Python: ...../test_python_ctools.sh: line 25: 18386 Segmentation fault: 11  ./test_python_ctools.py
Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0   libgamma.8.dylib                0x00000001064b0432 GResponseVectorCache::contains(std::__1::basic_string<char, std::__1::char_traits<char>,
 std::__1::allocator<char> > const&, GVector*) const + 130 (GResponseVectorCache.cpp:337)
1   libgamma.8.dylib                0x00000001064aa502 GResponse::irf_spatial(GModelSky const&, GObservation const&, GMatrix*) const + 178
 (GResponse.cpp:687)
2   libgamma.8.dylib                0x00000001064a986f GResponse::eval_probs(GModelSky const&, GObservation const&, GMatrixSparse*) const +
 191
3   libgamma.8.dylib                0x00000001064a84cc GResponse::convolve(GModelSky const&, GObservation const&, GMatrixSparse*) const + 412
4   libgamma.8.dylib                0x00000001064c2236 GModelSky::eval(GObservation const&, GMatrixSparse*) const + 54 (GModelSky.cpp:603)
5   libgamma.8.dylib                0x0000000106498282 GObservation::model(GModels const&, GMatrixSparse*) const + 402
6   libgamma.8.dylib                0x000000010649a162 GObservation::likelihood_poisson_unbinned(GModels const&, GVector*, GMatrixSparse*,
 double*) const + 274
7   libgamma.8.dylib                0x0000000106497443 GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const + 595
8   libgamma.8.dylib                0x0000000106494b8f GObservations::likelihood::eval(GOptimizerPars const&) + 735
9   libgamma.8.dylib                0x000000010647382c GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) + 1084
10  libgamma.8.dylib                 0x0000000106492c63 GObservations::optimize(GOptimizer&) + 51
11  libctools.7.dylib               0x000000010804b02f ctlike::optimize_lm() + 831 (ctlike.cpp:483)
12  libctools.7.dylib               0x0000000108049c46 ctlike::run() + 214 (GObservations.hpp:359)
13  libctools.7.dylib               0x00000001080320a6 ctool::execute() + 22 (ctool.cpp:267)
```

The issue occurs when ctlike is re-executed:

```
    # Execute copy of ctlike tool again, now with a higher chatter
    # level than before
    cpy_like['outmodel'] = 'ctlike_py2.xml'
    cpy_like['logfile']  = 'ctlike_py2.log'
    cpy_like['chatter']  = 3
    cpy_like.logFileOpen()  # Needed to get a new log file
    cpy_like.execute()
```

#6 - 10/22/2020 05:42 PM - Knödlseder Jürgen

There was a typo in GResponseVectorCache::copy_members() that led to a bad copy of the cache, which was relevant for the case that a tool is copied. I fixed the typo. The modified code was merged into devel.

#7 - 11/08/2020 06:34 PM - Knödlseder Jürgen

Still need to fix the Python warning (see above)

#8 - 11/18/2020 11:00 PM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 80 to 100

The Python warning was fixed by issue #3324.