

GammaLib - Action #2715

Add generic CTA response computation caching

11/05/2018 06:35 PM - Knödlseider Jürgen

Status:	Closed	Start date:	11/05/2018
Priority:	Normal	Due date:	
Assigned To:	Knödlseider Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.6.0		

Description

A generic caching of the response computation should be done so that the spatial response for any model that has no free parameters is only evaluated once.

The generic cache should be added to `GCTAResponse::irf(GEvent&, GSource&, GObservation&)` and `GCTAResponseIrf::nroi(GModelSky&, GEnergy&, GTime&, GEnergy&, GTime&, GObservation&)` which only concerns the spatial part of the integration. A method `GModelSpatial::has_free_pars()` should be added that signals whether the spatial model has free parameters. Then, the `GCTAResponse::irf` and `GCTAResponseIrf::nroi` methods should cache the irf and nroi as function of model name, reconstructed energy and true energy if the spatial model has no free parameters.

A generic caching class `GCTAResponseCache` should be implemented that holds the cache results for a list of sky models and observation IDs (in principle there should be only a single observation ID since each observation has a dedicated response). The `GCTAResponseCache` should contain a list of `GCTAResponseCacheEreco` elements, which hold the response information as function of the reconstructed energy. In case that energy dispersion is used, there may be a vector of true energies associated to each reconstructed energy, hence another container class `GCTAResponseCacheEtrue` is needed to store the response for each true energy value.

```
class GCTAResponseCache : public GContainer {
    std::vector<std::string> m_names;
    std::vector<GCTAResponseCacheEreco> m_entries;
}
class GCTAResponseCacheEreco : public GContainer {
    std::vector<GEnergy> m_ereco;
    std::vector<GCTAResponseCacheEtrue> m_values;
}
class GCTAResponseCacheEtrue : public GContainer {
    std::vector<GEnergy> m_etrue;
    std::vector<double> m_values;
}
```

History

#1 - 11/06/2018 02:06 AM - Knödlseider Jürgen

- Status changed from New to In Progress
- Assigned To set to Knödlseider Jürgen
- % Done changed from 0 to 10

It turns out that the cache can be implemented using a single class making use of nested `std::map` constructs:

```
typedef std::map<GEnergy,double> GCTAResponseCacheEtrue;
typedef std::map<GEnergy,GCTAResponseCacheEtrue> GCTAResponseCacheEreco;
typedef std::map<std::string,GCTAResponseCacheEreco> GCTAResponseCacheName;

/*****
 * @class GCTAResponseCache
 *
 * @brief CTA response cache class
 *****/
class GCTAResponseCache : public GBase {
```

```

public:
    // Constructors and destructors
    GCTAResponseCache(void);
    GCTAResponseCache(const GCTAResponseCache& cache);
    virtual ~GCTAResponseCache(void);

    // Operators
    GCTAResponseCache& operator=(const GCTAResponseCache& cache);

    // Methods
    void clear(void);
    GCTAResponseCache* clone(void) const;
    std::string classname(void) const;
    void set(const std::string& name,
            const GEnergy& ereco,
            const GEnergy& etrue,
            const double& value);
    bool contains(const std::string& name,
                const GEnergy& ereco,
                const GEnergy& etrue,
                double* value = NULL) const;
    std::string print(const GChatter& chatter = NORMAL) const;

protected:
    // Protected methods
    void init_members(void);
    void copy_members(const GCTAResponseCache& cache);
    void free_members(void);

    // Protected members
    GCTAResponseCacheName m_cache;
};

```

#2 - 11/06/2018 12:11 PM - Knödlseher Jürgen

- % Done changed from 10 to 50

The cache is now used in GCTAResponse and GCTAResponseIrf. Any spatial model that has no free parameters will benefit from the cache. The former diffuse model cache was removed.

I tested the cache on the H.E.S.S. DR1 for RX J1713.7 and the results are identical to the previous code.

Some more testing should be done using for example the Crab point source or Gaussian model to check whether everything works as expected, also for models that have free spatial parameters.

#3 - 11/06/2018 12:24 PM - Knödseder Jürgen

- % Done changed from 50 to 60

The IRF cache was removed from GCTAEventList since it is no longer used.

#4 - 11/06/2018 03:10 PM - Knödseder Jürgen

- % Done changed from 60 to 70

It turned out that also the event's instrument direction needs to be cached, since eventually this may change at the client side and will result in a different IRF value.

I therefore added two nested layers to the cache, one for Right Ascension and the other for Declination of the instrument direction in degrees. There are variants of the GCTAResponseCache::set() and GCTAResponseCache::contains() methods without an instrument direction argument, since for roi computation this is not needed. For these variations, the Right Ascension and Declination values are set to zero.

#5 - 11/06/2018 03:14 PM - Knödseder Jürgen

Here a comparison of the RX J1713.7 run. First a reference run before implementing the cache:

```
2018-11-06T09:22:59: +=====+
2018-11-06T09:22:59: | Maximum likelihood optimisation results |
2018-11-06T09:22:59: +=====+
2018-11-06T09:22:59: === GOptimizerLM ===
2018-11-06T09:22:59: Optimized function value ..: 363465.042
2018-11-06T09:22:59: Absolute precision .....: 0.005
2018-11-06T09:22:59: Acceptable value decrease ..: 2
2018-11-06T09:22:59: Optimization status .....: converged
2018-11-06T09:22:59: Number of parameters .....: 305
2018-11-06T09:22:59: Number of free parameters ..: 167
2018-11-06T09:22:59: Number of iterations .....: 26
2018-11-06T09:22:59: Lambda .....: 1e-23
2018-11-06T09:22:59: Maximum log likelihood ....: -363465.042
2018-11-06T09:22:59: Observed events (Nobs) ...: 38551.000
2018-11-06T09:22:59: Predicted events (Npred) ..: 38550.999 (Nobs - Npred = 0.00100186896452215)
2018-11-06T09:22:59: === GModels ===
2018-11-06T09:22:59: Number of models .....: 16
2018-11-06T09:22:59: Number of parameters .....: 305
2018-11-06T09:22:59: === GModelSky ===
2018-11-06T09:22:59: Name .....: RX J1713.7-3946
2018-11-06T09:22:59: Instruments .....: all
2018-11-06T09:22:59: Test Statistic .....: 719.177990890923
2018-11-06T09:22:59: Instrument scale factors ..: unity
2018-11-06T09:22:59: Observation identifiers ...: all
2018-11-06T09:22:59: Model type .....: DiffuseSource
2018-11-06T09:22:59: Model components .....: "DiffuseMap" * "PowerLaw" * "Constant"
2018-11-06T09:22:59: Number of parameters .....: 5
2018-11-06T09:22:59: Number of spatial par's ...: 1
2018-11-06T09:22:59: Normalization .....: 1 [0.001,1000] (fixed,scale=1,gradient)
2018-11-06T09:22:59: Number of spectral par's ...: 3
2018-11-06T09:22:59: Prefactor .....: 1.65970723538055e-17 +/- 7.07741211142014e-19 [1e-25,infy[ ph/cm2/s/MeV
```

(free,scale=1e-17,gradient)
2018-11-06T09:22:59: Index: -2.35527895717826 +/- 0.0539618573709091 [10,-10] (free,scale=-2,gradient)
2018-11-06T09:22:59: PivotEnergy: 1000000 MeV (fixed,scale=1000000,gradient)
2018-11-06T09:22:59: Number of temporal par's ..: 1
2018-11-06T09:22:59: Normalization: 1 (relative value) (fixed,scale=1,gradient)
2018-11-06T09:22:59:
2018-11-06T09:22:59: Application "ctlike" terminated after 438 wall clock seconds, consuming 418.19 seconds of CPU time.

And here a run with the new response cache:

```
2018-11-06T14:10:41: +=====+
2018-11-06T14:10:41: | Maximum likelihood optimisation results |
2018-11-06T14:10:41: +=====+
2018-11-06T14:10:41: === GOptimizerLM ===
2018-11-06T14:10:41: Optimized function value ...: 363465.042
2018-11-06T14:10:41: Absolute precision .....: 0.005
2018-11-06T14:10:41: Acceptable value decrease ..: 2
2018-11-06T14:10:41: Optimization status .....: converged
2018-11-06T14:10:41: Number of parameters .....: 305
2018-11-06T14:10:41: Number of free parameters ..: 167
2018-11-06T14:10:41: Number of iterations .....: 26
2018-11-06T14:10:41: Lambda .....: 1e-23
2018-11-06T14:10:41: Maximum log likelihood ....: -363465.042
2018-11-06T14:10:41: Observed events (Nobs) ...: 38551.000
2018-11-06T14:10:41: Predicted events (Npred) ..: 38550.999 (Nobs - Npred = 0.00100186896452215)
2018-11-06T14:10:41: === GModels ===
2018-11-06T14:10:41: Number of models .....: 16
2018-11-06T14:10:41: Number of parameters .....: 305
2018-11-06T14:10:41: === GModelSky ===
2018-11-06T14:10:41: Name .....: RX J1713.7-3946
2018-11-06T14:10:41: Instruments .....: all
2018-11-06T14:10:41: Test Statistic .....: 719.177990890923
2018-11-06T14:10:41: Instrument scale factors ...: unity
2018-11-06T14:10:41: Observation identifiers ...: all
2018-11-06T14:10:41: Model type .....: DiffuseSource
2018-11-06T14:10:41: Model components .....: "DiffuseMap" * "PowerLaw" * "Constant"
2018-11-06T14:10:41: Number of parameters .....: 5
2018-11-06T14:10:41: Number of spatial par's ...: 1
2018-11-06T14:10:41: Normalization .....: 1 [0.001,1000] (fixed,scale=1,gradient)
2018-11-06T14:10:41: Number of spectral par's ..: 3
2018-11-06T14:10:41: Prefactor .....: 1.65970723538055e-17 +/- 7.07741211142014e-19 [1e-25,infy[ ph/cm2/s/MeV
(free,scale=1e-17,gradient)
2018-11-06T14:10:41: Index .....: -2.35527895717826 +/- 0.0539618573709091 [10,-10] (free,scale=-2,gradient)
2018-11-06T14:10:41: PivotEnergy .....: 1000000 MeV (fixed,scale=1000000,gradient)
2018-11-06T14:10:41: Number of temporal par's ..: 1
2018-11-06T14:10:41: Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
2018-11-06T14:10:41:
2018-11-06T14:10:41: Application "ctlike" terminated after 431 wall clock seconds, consuming 427.073 seconds of CPU time.
```

Results are identical!

#6 - 11/06/2018 04:54 PM - Knödseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 70 to 100

I also confirmed on the Crab data that the cache is now working for energy dispersion. Below the reference results for a fixed Crab position without using the cache:

```
2018-11-06T15:24:34: +=====+
2018-11-06T15:24:34: | Maximum likelihood optimisation results |
2018-11-06T15:24:34: +=====+
2018-11-06T15:24:34: === GOptimizerLM ===
2018-11-06T15:24:34: Optimized function value ..: 98213.461
2018-11-06T15:24:34: Absolute precision ..: 0.005
2018-11-06T15:24:34: Acceptable value decrease ..: 2
2018-11-06T15:24:34: Optimization status ..: converged
2018-11-06T15:24:34: Number of parameters ..: 86
2018-11-06T15:24:34: Number of free parameters ..: 46
2018-11-06T15:24:34: Number of iterations ..: 10
2018-11-06T15:24:34: Lambda ..: 1e-07
2018-11-06T15:24:34: Maximum log likelihood ..: -98213.461
2018-11-06T15:24:34: Observed events (Nobs) ..: 9675.000
2018-11-06T15:24:34: Predicted events (Npred) ..: 9674.987 (Nobs - Npred = 0.0130850824170921)
2018-11-06T15:24:34: === GModels ===
2018-11-06T15:24:34: Number of models ..: 5
2018-11-06T15:24:34: Number of parameters ..: 86
2018-11-06T15:24:34: === GModelSky ===
2018-11-06T15:24:34: Name ..: Crab
2018-11-06T15:24:34: Instruments ..: all
2018-11-06T15:24:34: Test Statistic ..: 1997.058246759
2018-11-06T15:24:34: Instrument scale factors ..: unity
2018-11-06T15:24:34: Observation identifiers ..: all
2018-11-06T15:24:34: Model type ..: PointSource
2018-11-06T15:24:34: Model components ..: "PointSource" * "PowerLaw" * "Constant"
2018-11-06T15:24:34: Number of parameters ..: 6
2018-11-06T15:24:34: Number of spatial par's ..: 2
2018-11-06T15:24:34: RA ..: 83.6331 deg (fixed,scale=1)
2018-11-06T15:24:34: DEC ..: 22.0145 deg (fixed,scale=1)
2018-11-06T15:24:34: Number of spectral par's ..: 3
2018-11-06T15:24:34: Prefactor ..: 4.13862354697789e-17 +/- 2.00142048983869e-18 [1e-25,infy[ ph/cm2/s/MeV
  (free,scale=1e-17,gradient)
2018-11-06T15:24:34: Index ..: -2.73611491813232 +/- 0.0704069630437431 [10,-10] (free,scale=-2,gradient)
2018-11-06T15:24:34: PivotEnergy ..: 1000000 MeV (fixed,scale=1000000,gradient)
2018-11-06T15:24:34: Number of temporal par's ..: 1
2018-11-06T15:24:34: Normalization ..: 1 (relative value) (fixed,scale=1,gradient)
2018-11-06T15:24:34:
2018-11-06T15:24:34: Application "ctlike" terminated after 46 wall clock seconds, consuming 45.6971 seconds of CPU time.
```

And here the results with caching enabled:

```
2018-11-06T15:27:08: +=====+
2018-11-06T15:27:08: | Maximum likelihood optimisation results |
2018-11-06T15:27:08: +=====+
2018-11-06T15:27:08: === GOptimizerLM ===
2018-11-06T15:27:08: Optimized function value ..: 98213.461
2018-11-06T15:27:08: Absolute precision ..: 0.005
2018-11-06T15:27:08: Acceptable value decrease ..: 2
2018-11-06T15:27:08: Optimization status ..: converged
2018-11-06T15:27:08: Number of parameters ..: 86
2018-11-06T15:27:08: Number of free parameters ..: 46
2018-11-06T15:27:08: Number of iterations ..: 10
2018-11-06T15:27:08: Lambda ..: 1e-07
2018-11-06T15:27:08: Maximum log likelihood ..: -98213.461
2018-11-06T15:27:08: Observed events (Nobs) ..: 9675.000
2018-11-06T15:27:08: Predicted events (Npred) ..: 9674.987 (Nobs - Npred = 0.0130850824170921)
2018-11-06T15:27:08: === GModels ===
2018-11-06T15:27:08: Number of models ..: 5
2018-11-06T15:27:08: Number of parameters ..: 86
2018-11-06T15:27:08: === GModelSky ===
2018-11-06T15:27:08: Name ..: Crab
2018-11-06T15:27:08: Instruments ..: all
```

2018-11-06T15:27:08: Test Statistic: 1997.058246759
2018-11-06T15:27:08: Instrument scale factors ...: unity
2018-11-06T15:27:08: Observation identifiers: all
2018-11-06T15:27:08: Model type: PointSource
2018-11-06T15:27:08: Model components: "PointSource" * "PowerLaw" * "Constant"
2018-11-06T15:27:08: Number of parameters: 6
2018-11-06T15:27:08: Number of spatial par's ...: 2
2018-11-06T15:27:08: RA: 83.6331 deg (fixed,scale=1)
2018-11-06T15:27:08: DEC: 22.0145 deg (fixed,scale=1)
2018-11-06T15:27:08: Number of spectral par's ...: 3
2018-11-06T15:27:08: Prefactor: 4.13862354697789e-17 +/- 2.00142048983869e-18 [1e-25,infy[ph/cm2/s/MeV
(free,scale=1e-17,gradient)
2018-11-06T15:27:08: Index: -2.73611491813232 +/- 0.0704069630437431 [10,-10] (free,scale=-2,gradient)
2018-11-06T15:27:08: PivotEnergy: 1000000 MeV (fixed,scale=1000000,gradient)
2018-11-06T15:27:08: Number of temporal par's ...: 1
2018-11-06T15:27:08: Normalization: 1 (relative value) (fixed,scale=1,gradient)
2018-11-06T15:27:08:
2018-11-06T15:27:08: Application "ctlike" terminated after 37 wall clock seconds, consuming 37.3492 seconds of CPU time.

They are identical!

#7 - 11/06/2018 06:02 PM - Knödlseider Jürgen

- Status changed from Pull request to Closed

Merged into devel.