GammaLib - Action #3501

Implement vectorised response computation for COMPTEL

01/15/2021 08:57 AM - Knödlseder Jürgen

Status:	Closed	**Start date:**	01/15/2021
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0.0		

Description

History

#1 - 01/15/2021 01:29 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 50

I implemented a vectorised response computation in GCOMResponse::irf_ptsrc() making use of the characteristics of the COMPTEL data space. Specifically, phigeo is now only computed once for the Chi/Psi pixel.

Initial testing using a five energy-band Crab fit reduced the ctlike execution time from 23.23 seconds to 12.48 seconds, hence leading to a speed-up of a factor of 2. More testing is needed to verify for example that the code also works for TS map computation. Also, a valgrind analysis may be done to further speed-up the analysis.

#2 - 01/15/2021 02:04 PM - Knödlseder Jürgen

- % Done changed from 50 to 60

There is basically no speed-up in the spectral analysis, which is normal since the spatial parameters are fixed in the spectral analysis. Consequently, the cached values will be used.

#3 - 01/15/2021 02:41 PM - Knödlseder Jürgen

- % Done changed from 60 to 70

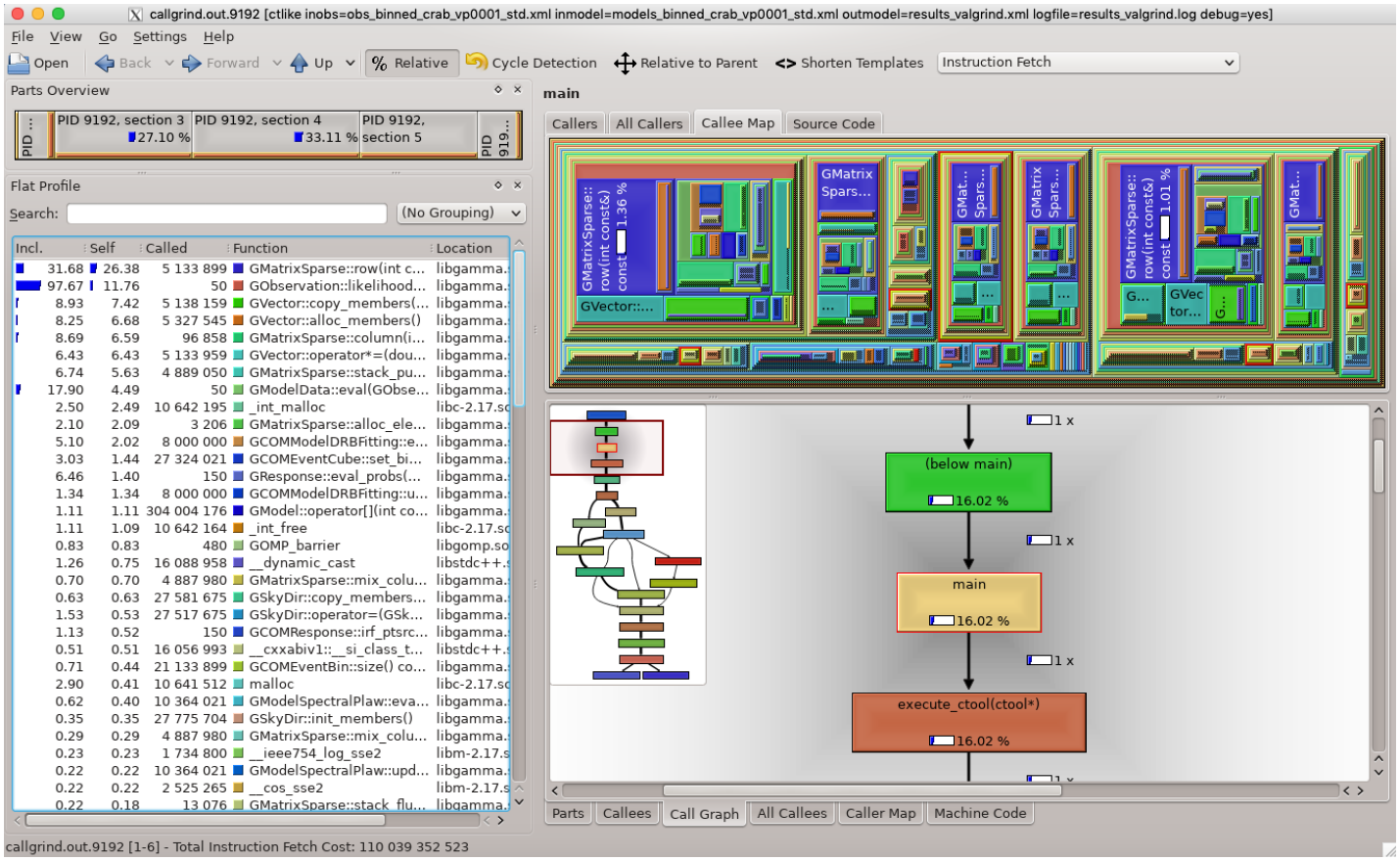
The same holds for the TS map computation. No speed-up because the spatial parameters are fixed.

#4 - 01/15/2021 03:32 PM - Knödlseder Jürgen

- File kcachegrind.png added

Here is a valgrind analysis of the 5 standard band ctlike run on viewing period 1. Obviously a significant fraction of time is spent in the matrix and vector handling.

Specifically, the GMatrixSparse::row method is called many times in GObservation::likelihood_poisson_binned which uses quite some computing time. The reason for this is that the matrices are stored column-wise, hence row access needs quite some computations. It should be checked whether a transposition of the matrix and a column access residuals in actually faster code.


callgrind.out.9192 [ctlike inobs=obs_binned_crab_vp0001_std.xml inmodel=models_binned_crab_vp0001_std.xml outmodel=results_valgrind.xml logfile=results_valgrind.log debug=yes]
File View Go Settings Help
Open Back Forward Up Relative Cycle Detection Relative to Parent Shorten Templates Instruction Fetch
Parts Overview
PID 9192, section 3 27.10 %
PID 9192, section 4 33.11 %
PID 9192, section 5
Flat Profile
Search:
(No Grouping)
Incl. Self Called Function Location
31.68 26.38 5 133 899 GMatrixSparse::row(int c... libgamma.
97.67 11.76 50 GObservation::likelihood... libgamma.
8.93 7.42 5 138 159 GVector::copy_members(... libgamma.
8.25 6.68 5 327 545 GVector::alloc_members() libgamma.
8.69 6.59 96 858 GMatrixSparse::column(i... libgamma.
6.43 6.43 5 133 959 GVector::operator*=(dou... libgamma.
6.74 5.63 4 889 050 GMatrixSparse::stack_pu... libgamma.
17.90 4.49 50 GModelData::eval(GObse... libgamma.
2.50 2.49 10 642 195 _int_malloc libc-2.17.so
2.10 2.09 3 206 GMatrixSparse::alloc_ele... libgamma.
5.10 2.02 8 000 000 GCOMModelDRBFitting::e... libgamma.
3.03 1.44 27 324 021 GCOMEventCube::set_bi... libgamma.
6.46 1.40 150 GResponse::eval_probs(... libgamma.
1.34 1.34 8 000 000 GCOMModelDRBFitting::u... libgamma.
1.11 1.11 304 004 176 GModel::operator[](int co... libgamma.
1.11 1.09 10 642 164 _int_free libc-2.17.so
0.83 0.83 480 GOMP_barrier libgomp.so
1.26 0.75 16 088 958 __dynamic_cast libstdc++.
0.70 0.70 4 887 980 GMatrixSparse::mix_colu... libgamma.
0.63 0.63 27 581 675 GSkyDir::copy_members... libgamma.
1.53 0.53 27 517 675 GSkyDir::operator=(GSk... libgamma.
1.13 0.52 150 GCOMResponse::irf_ptsrc... libgamma.
0.51 0.51 16 056 993 __cxxabiv1::__si_class_t... libstdc++.
0.71 0.44 21 133 899 GCOMEventBin::size() co... libgamma.
2.90 0.41 10 641 512 malloc libc-2.17.so
0.62 0.40 10 364 021 GModelSpectralPlaw::eva... libgamma.
0.35 0.35 27 775 704 GSkyDir::init_members() libgamma.
0.29 0.29 4 887 980 GMatrixSparse::mix_colu... libgamma.
0.23 0.23 1 734 800 __ieee754_log_sse2 libm-2.17.s
0.22 0.22 10 364 021 GModelSpectralPlaw::upd... libgamma.
0.22 0.22 2 525 265 __cos_sse2 libm-2.17.s
0.22 0.18 13 076 GMatrixSparse::stack_flu... libgamma.
main
Callers All Callers Callee Map Source Code
GMatrixSparse:: row(int const&) const 1.36 %
GVector::...
GMatrix Spars...
GMatrixSparse:: row(int const&) const 1.01 %
1 x
(below main)
16.02 %
main
16.02 %
execute_ctool(ctool*)
16.02 %
Parts Callees Call Graph All Callees Caller Map Machine Code
callgrind.out.9192 [1-6] - Total Instruction Fetch Cost: 110 039 352 523

#5 - 01/15/2021 04:48 PM - Knödlseder Jürgen

- % Done changed from 70 to 80

Transposing the matrix in GObservation::likelihood_poisson_binned and accessing it through the GMatrixSparse::column method leads to faster code, using 10.21 instead of 12.50 seconds for the 5-band ctlike fit of viewing period 1. For the TS map, the computation needed 3671.03 seconds instead of 5017.22 seconds. Overall, a speed-up of about 20-25% is achieved.

I also checked the speed-up for fitting the Crab to simulated CTA data (fixed position, only prefactor and spectral index are free). Below the computation times and speed-up for the various analysis methods:

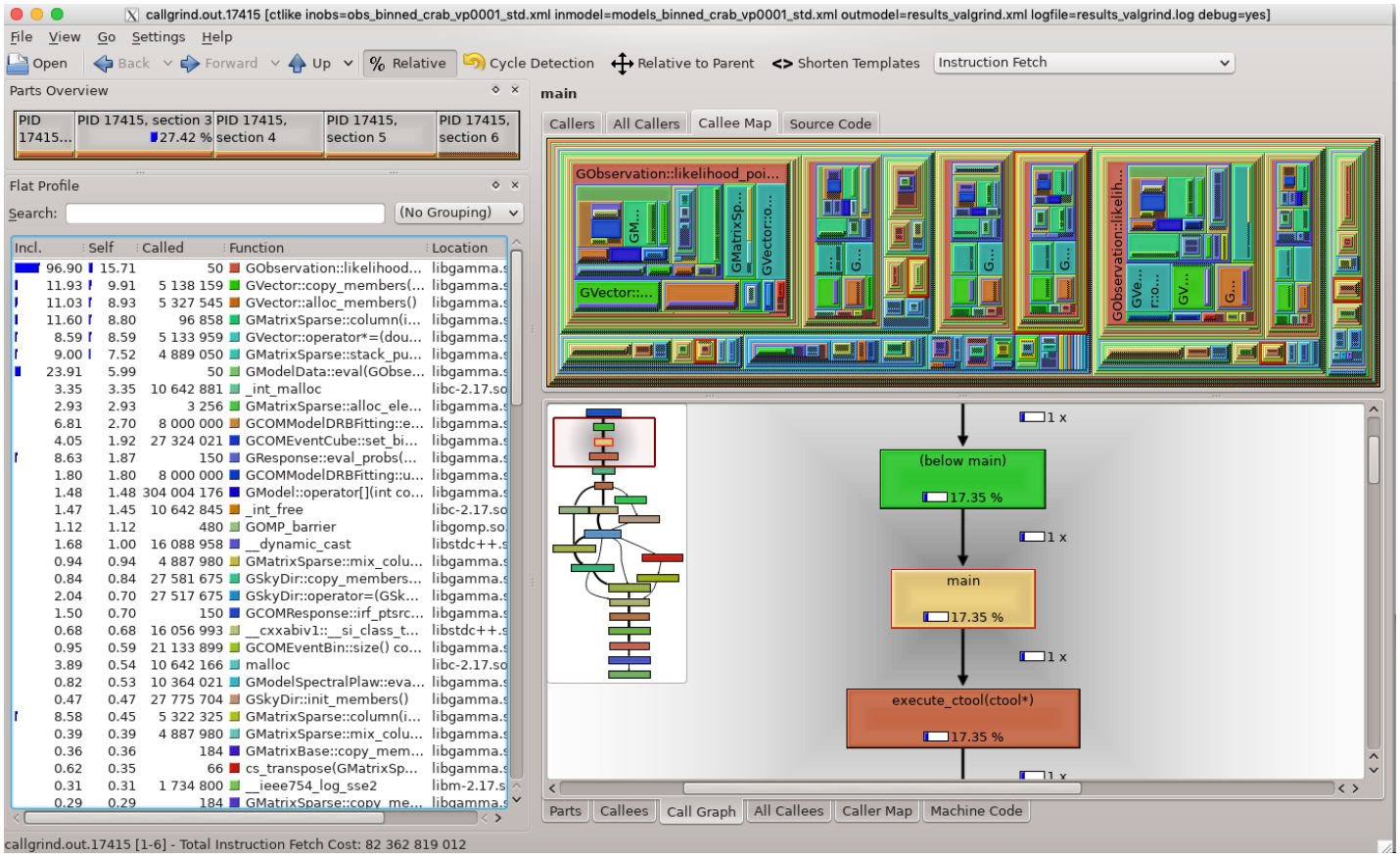
Method	**Original**	**Transposed**	**Speed-up**
Unbinned	7.95 s	7.34 s	8%
Binned	5.97 s	5.62 s	6%
Stacked	3.68 s	3.31 s	10%

In all cases the code is faster. I will therefore implement the matrix transformation.

#6 - 01/15/2021 04:55 PM - Knödlseder Jürgen

- File kcachegrind-transpose.png added

Below the valgrind analysis for the transposed matrix. As expected, matrix access is no longer a bottleneck (the GMatrixSparse::column method uses less than 1% of the time).


callgrind.out.17415 [ctlike inobs=obs_binned_crab_vp0001_std.xml inmodel=models_binned_crab_vp0001_std.xml outmodel=results_valgrind.xml logfile=results_valgrind.log debug=yes]
File View Go Settings Help
Open Back Forward Up Relative Cycle Detection Relative to Parent Shorten Templates Instruction Fetch
Parts Overview
PID 17415...
PID 17415, section 3
27.42 %
PID 17415, section 4
PID 17415, section 5
PID 17415, section 6
Flat Profile
Search:
(No Grouping)
Incl. Self Called Function Location
96.90 15.71 50 GObservation::likelihood... libgamma.s
11.93 9.91 5 138 159 GVector::copy_members(... libgamma.s
11.03 8.93 5 327 545 GVector::alloc_members() libgamma.s
11.60 8.80 96 858 GMatrixSparse::column(i... libgamma.s
8.59 8.59 5 133 959 GVector::operator*=(dou... libgamma.s
9.00 7.52 4 889 050 GMatrixSparse::stack_pu... libgamma.s
23.91 5.99 50 GModelData::eval(GObse... libgamma.s
3.35 3.35 10 642 881 _int_malloc libc-2.17.so
2.93 2.93 3 256 GMatrixSparse::alloc_ele... libgamma.s
6.81 2.70 8 000 000 GCOMModelDRBFitting::e... libgamma.s
4.05 1.92 27 324 021 GCOMEventCube::set_bi... libgamma.s
8.63 1.87 150 GResponse::eval_probs(... libgamma.s
1.80 1.80 8 000 000 GCOMModelDRBFitting::u... libgamma.s
1.48 1.48 304 004 176 GModel::operator[](int co... libgamma.s
1.47 1.45 10 642 845 _int_free libc-2.17.so
1.12 1.12 480 GOMP_barrier libgomp.so
1.68 1.00 16 088 958 __dynamic_cast libstdc++.s
0.94 0.94 4 887 980 GMatrixSparse::mix_colu... libgamma.s
0.84 0.84 27 581 675 GSkyDir::copy_members... libgamma.s
2.04 0.70 27 517 675 GSkyDir::operator=(GSk... libgamma.s
1.50 0.70 150 GCOMResponse::irf_ptsrc... libgamma.s
0.68 0.68 16 056 993 __cxxabiv1::__si_class_t... libstdc++.s
0.95 0.59 21 133 899 GCOMEventBin::size() co... libgamma.s
3.89 0.54 10 642 166 malloc libc-2.17.so
0.82 0.53 10 364 021 GModelSpectralPlaw::eva... libgamma.s
0.47 0.47 27 775 704 GSkyDir::init_members() libgamma.s
8.58 0.45 5 322 325 GMatrixSparse::column(i... libgamma.s
0.39 0.39 4 887 980 GMatrixSparse::mix_colu... libgamma.s
0.36 0.36 184 GMatrixBase::copy_mem... libgamma.s
0.62 0.35 66 cs_transpose(GMatrixSp... libgamma.s
0.31 0.31 1 734 800 __ieee754_log_sse2 libm-2.17.s
0.29 0.29 184 GMatrixSparse::copy_me... libgamma.s
main
Callers All Callers Callee Map Source Code
GObservation::likelihood_poi...
GVector::...
GM...
GMatrixSp...
GVector::o...
GObservation::likelih...
1 x
(below main)
17.35 %
1 x
main
17.35 %
1 x
execute_ctool(ctool*)
17.35 %
1 x
Parts Callees Call Graph All Callees Caller Map Machine Code
callgrind.out.17415 [1-6] - Total Instruction Fetch Cost: 82 362 819 012

#7 - 01/15/2021 04:56 PM - Knödlseder Jürgen

- Status changed from In Progress to Pull request

- % Done changed from 80 to 90

#8 - 01/15/2021 08:28 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

- % Done changed from 90 to 100

Merged into devel.

Files

kcachegrind.png	491 KB	01/15/2021	Knödlseder Jürgen
kcachegrind-transpose.png	519 KB	01/15/2021	Knödlseder Jürgen