Profile of CMS Geometry in GeantV

Guilherme Amadio

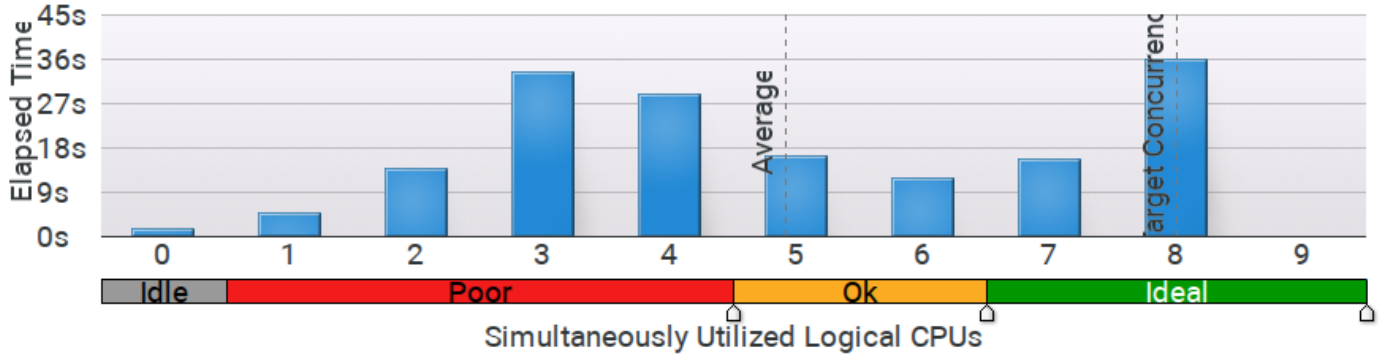
April 7, 2015

Overview

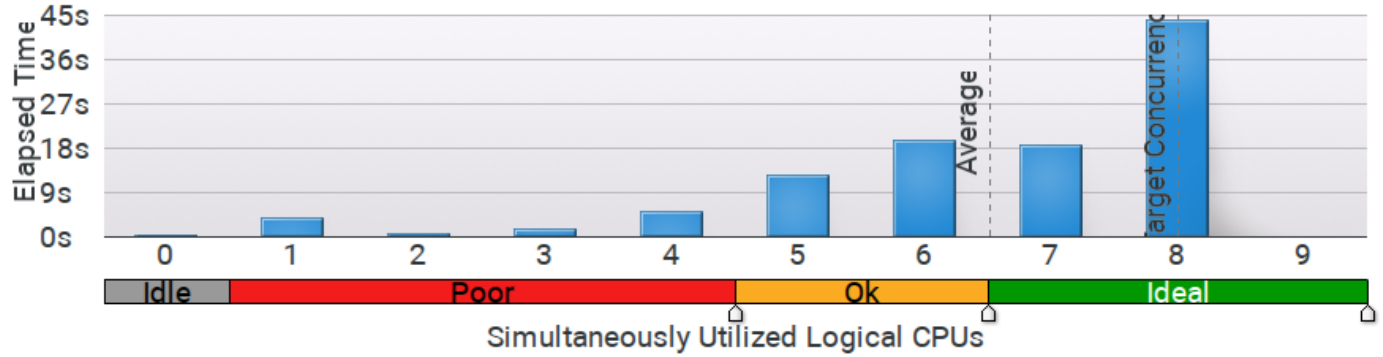
- CMS Application in GeantV with ROOT for geometry
  - VecGeom is not yet ready to run in GeantV
- Release builds of ROOT and VecGeom
  - No call stack information for now
  - No kernel profiling
- Running on Core i7 4710HQ (2.5 GHz, 16GB RAM)
  - Use 8 threads to take advantage of hyper-threading
  - Set max memory to 12GB (never reaches the threshold)
  - Simulate 10 events, 5 buffered at a time
  - Using pp14TeVminbias.root with HepMC
  - No graphics monitoring when profiling

CPU Usage

General Exploration Analysis



Advanced Hotspots Analysis



General Exploration Analysis Summary

Elapsed Time: 163.652s

Instructions Retired: 1,438,574,157,858
CPI Rate: 1.462
The CPI may be too high. This could be caused by issues such as memory stalls, instruction starvation, branch misprediction or long latency instructions. Explore the other hardware-related metrics to identify what is causing...
CPU Frequency Ratio: 0.992
Paused Time: 0s

- **CPU Time: 849.898s**
  - Spin Time: 7.472s
  - Overhead Time: 0s
  - **Effective Time: 842.427s**

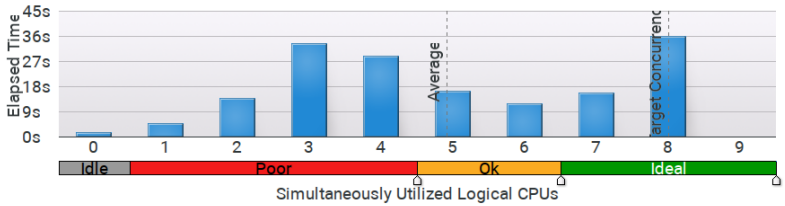
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	CPU Time
[vmlinux]	102.526s
TList::LinkAt	65.041s
__ieee754_log_avx	61.494s
__memcpy_avx_unaligned	49.231s
TGeoHMatrix::Multiply	40.739s
[Others]	530.868s

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



Collection and Platform Info

This section provides information about this collection, including result set size and collection platform data.

Application Command Line:	root "-b" "-q" "runCMS.C"
Operating System:	3.19.0-gentoo Gentoo Base System release 2.2
Computer Name:	antares
Result Size:	878 MB
Collection start time:	15:06:24 02/04/2015 UTC
Collection stop time:	15:09:08 02/04/2015 UTC

- **CPU**
  - Name: 4th generation Intel(R) Core(TM) Processor family
  - Frequency: 2.5 GHz
  - Logical CPU Count: 8

Elapsed Time: 163.652s

Clockticks: 2,103,737,155,601
Instructions Retired: 1,438,574,157,858
CPI Rate: 1.462
The CPI may be too high. This could be caused by issues such as memory stalls, instruction starvation, branch misprediction or long latency instructions. Explore the other hardware-related metrics to identify what is causing...
MUX Reliability: 0.999
Paused Time: 0s

- **Filled Pipeline Slots:**
  - **Retiring: 0.189**
  - **Bad Speculation: 0.022**
- **Unfilled Pipeline Slots (Stalls):**
  - **Back-End Bound: 0.660**

    Identify slots where no uOps are delivered due to a lack of required resources for accepting more uOps in the back-end of the pipeline. Back-end metrics describe a portion of the pipeline where the out-of-order scheduler dispatches ready uOps into their respective execution units, and, once completed, these uOps get retired according to program order. Stalls due to data-cache misses or stalls due to the overloaded divider unit are examples of back-end bound issues.
    - **Memory Bound: 0.441**

      This metric shows how memory subsystem issues affect the performance. Memory Bound measures a fraction of cycles where pipeline could be stalled due to demand load or store instructions. This accounts mainly for incomplete in-flight memory demand loads that coincide with execution starvation in addition to less common cases where stores could imply back-pressure on the pipeline.
      - **L1 Bound: 0.381**

        This metric shows how often machine was stalled without missing the L1 data cache. The L1 cache typically has the shortest latency. However, in certain cases like loads blocked on older stores, a load might suffer a high latency even though it is being satisfied by the L1.
        - DTLB Overhead: 0.124

          A significant proportion of cycles is being spent handling first-level data TLB misses. As with ordinary data caching, focus on improving data locality and reducing working-set size to reduce...
        - Loads Blocked by Store Forwarding: 0.000
        - Split Loads: 0.000
        - 4K Aliasing: 0.010
      - **L3 Bound:**
        - Contested Accesses: 0.025
        - Data Sharing: 0.015
        - LLC Hit: 0.109
      - **DRAM Bound:**
        - LLC Miss: 0.060
      - **Store Bound: 0.061**
    - **Core Bound: 0.349**

      This metric shows how core non-memory issues limit the performance when you run out of OOO resources or are saturating certain execution units (for example, using FP-chained long-latency arithmetic operations).
      - **Port Utilization:**
        - Cycles of 0 Ports Utilized: 0.512

          The number of cycles during which no port was utilized.
        - Cycles of 1 Port Utilized: 0.251

          The number of cycles during which only 1 port was utilized.
        - Cycles of 2 Ports Utilized: 0.139
        - Cycles of 3+ Ports Utilized: 0.092
  - **Front-end Bound: 0.129**
    - **Front-End Latency: 0.085**
      - ICache Misses: 0.054

        A significant proportion of instruction fetches are missing in the instruction cache. Use profile-guided optimization to reduce the size of hot code regions. Consider compiler options to reorder functions...
      - ITLB Overhead: 0.016
      - Branch Resteers: 0.023
      - DSB Switches: 0.016
      - Length Changing Prefixes: 0.000
      - Assists: 0.044
    - **Front-End Bandwidth: 0.043**

General Exploration Time Breakdown

Process / Module / Class / Function / Call Stack	CPU Time: Effective Time by Utilization (Idle / Poor / Ok / Ideal / Over)	Spin Time	Overhead Time	Instructions Retired	CPI Rate	CPU Frequency Ratio
root.exe	99.1%	7.472s	0s	100.0%	1.462	0.992
libGeom.so.6.03	23.1%	0s	0s	33.4%	1.012	0.993
libGeant_v.so	17.8%	0s	0s	13.2%	1.961	0.990
libXsec.so	15.3%	0s	0s	15.0%	1.497	0.995
libm-2.20.so	13.7%	0s	0s	19.2%	1.045	0.993
vmlinux	12.1%	0s	0s	5.1%	3.454	0.993
libCore.so.6.03	9.5%	0s	0s	6.6%	2.096	0.993
libc-2.20.so	6.1%	0s	0s	5.3%	1.696	0.987
libpthread-2.20.so	0.0%	7.472s	0s	0.3%	4.310	0.983
libMathCore.so.6.03	0.5%	0s	0s	0.8%	0.845	0.993
libGeantExamples.so	0.4%	0s	0s	0.4%	1.682	0.993
ld-2.20.so	0.4%	0s	0s	0.2%	2.491	0.968
libEG.so.6.03	0.1%	0s	0s	0.1%	1.253	0.907
libCling.so.6.03	0.1%	0s	0s	0.2%	0.799	1.346
libstdc++.so.6.0.20	0.0%	0s	0s	0.0%	5.301	0.976
libRIO.so.6.03	0.0%	0s	0s	0.1%	0.419	1.392
libz.so.1.2.8	0.0%	0s	0s	0.0%	0.536	1.182
libHepMC.so	0.0%	0s	0s	0.0%	1.300	1.625
libThread.so.6.03	0.0%	0s	0s	0.0%		0.000

Advanced Hotspots Time Breakdown

Process / Module / Class / Function / Call Stack	CPU Time			Instructions Retired	CPI Rate	CPU Frequency Ratio
	Effective Time by Utilization Idle Poor Ok Ideal Over	Spin Time	Overhead Time			
root.exe	98.9%	8.378s	0s	100.0%	1.321	1.003
libGeom.so.6.03	25.3%	0s	0s	34.3%	0.974	1.002
libGeant_v.so	18.5%	0s	0s	13.3%	1.824	0.998
libXsec.so	15.8%	0s	0s	15.2%	1.373	1.000
libm-2.20.so	15.2%	0s	0s	19.5%	1.040	1.012
libCore.so.6.03	9.8%	0s	0s	6.6%	1.965	1.006
vmlinux	6.5%	0s	0s	3.4%	2.505	1.007
libc-2.20.so	6.1%	0s	0s	5.4%	1.517	1.009
libpthread-2.20.so	0.0%	8.378s	0s	0.4%	3.722	0.962
libMathCore.so.6.03	0.5%	0s	0s	0.8%	0.820	0.991
libGeantExamples.so	0.5%	0s	0s	0.4%	1.524	0.958
ld-2.20.so	0.4%	0s	0s	0.2%	2.510	1.030
libEG.so.6.03	0.1%	0s	0s	0.2%	1.029	0.986
libCling.so.6.03	0.1%	0s	0s	0.2%	0.768	1.378
libstdc++.so.6.0.20	0.0%	0s	0s	0.0%	3.404	0.861
libRIO.so.6.03	0.0%	0s	0s	0.1%	0.390	1.389
libz.so.1.2.8	0.0%	0s	0s	0.0%	0.551	1.265
libHepMC.so	0.0%	0s	0s	0.0%	2.333	1.167
libHist.so.6.03	0.0%	0s	0s	0.0%		0.000
x86_64-pc-linux	0.0%	0s	0s	0.0%	0.810	1.417
cc1plus	0.0%	0s	0s	0.0%	1.000	1.000
root	0.0%	0s	0s	0.0%	1.000	1.000
sh	0.0%	0s	0s	0.0%	1.000	1.000
sh	0.0%	0s	0s	0.0%	1.000	1.000
sh	0.0%	0s	0s	0.0%	1.000	1.000
sh	0.0%	0s	0s	0.0%	1.000	1.000
ldd	0.0%	0s	0s	0.0%	0.667	2.000
awk	0.0%	0s	0s	0.0%	1.000	
x86_64-pc-linux	0.0%	0s	0s	0.0%	1.000	

Breakdown of Time Spent in Top Functions

Function Stack	CPU Time: Total Effective Time by Utilization Idle Poor Ok Ideal Over	CPU Time: Self Effective Time by Utilization Idle Poor Ok Ideal Over	Instructions Retired: Total	Instructions Retired: Self	CPI Rate: Total	CPI Rate: Self	Module
Total	99.1%	0s	1,438,574,157,858	0	1.462		
[vmlinux]	12.1%	102.526s	73,540,110,310	73,540,110,310	3.453	3.453	vmlinux
TList::LinkAt	7.7%	65.041s	68,856,103,284	68,856,103,284	2.326	2.326	libCore.so.6.03
__ieee754_log_avx	7.2%	61.494s	157,878,236,817	157,878,236,817	0.963	0.963	libm-2.20.so
__memcpy_avx_unaligned	5.8%	49.231s	70,554,105,831	70,554,105,831	1.717	1.717	libc-2.20.so
TGeoHMatrix::Multiply	4.8%	40.739s	131,286,196,929	131,286,196,929	0.768	0.768	libGeom.so.6.03
TTabPhysMgr::SampleFinalStates	2.8%	23.595s	36,666,054,999	36,666,054,999	1.607	1.607	libXsec.so
TGeoNodeCache::CdDown	2.6%	21.691s	62,004,093,006	62,004,093,006	0.867	0.867	libGeom.so.6.03
__dubsin	2.4%	20.470s	38,748,058,122	38,748,058,122	1.314	1.314	libm-2.20.so
TFinState::SampleReac	2.2%	18.512s	19,278,028,917	19,278,028,917	2.334	2.334	libXsec.so
GeantTrack_v::AddTrackSync	2.0%	17.105s	19,794,029,691	19,794,029,691	2.140	2.140	libGeant_v.so
TGeoNavigator::CdDown	1.8%	15.484s	43,038,064,557	43,038,064,557	0.891	0.891	libGeom.so.6.03
GeantTrack_v::PropagateInVolumeSingle	1.7%	14.115s	31,496,047,244	31,496,047,244	1.100	1.100	libGeant_v.so
TMXsec::Range	1.5%	12.752s	27,002,040,503	27,002,040,503	1.156	1.156	libXsec.so
GeantTrack_v::AddTracks	1.4%	11.994s	13,572,020,358	13,572,020,358	2.149	2.149	libGeant_v.so
TGeoHMatrix::CopyFrom	1.4%	11.616s	34,034,051,051	34,034,051,051	0.836	0.836	libGeom.so.6.03
GeantScheduler::AddTracks	1.3%	11.275s	10,710,016,065	10,710,016,065	2.596	2.596	libGeant_v.so
WorkloadManager::TransportTracks	1.3%	11.255s	8,602,012,903	8,602,012,903	3.285	3.285	libGeant_v.so
TMXsec::Eloss	1.3%	10.782s	17,790,026,685	17,790,026,685	1.473	1.473	libXsec.so
TPXsec::SampleReac	1.3%	10.747s	16,034,024,051	16,034,024,051	1.665	1.665	libXsec.so
TMXsec::ProposeStep	1.3%	10.675s	16,100,024,150	16,100,024,150	1.646	1.646	libXsec.so
GeantTrack_v::ComputeTransportLengthSingle	1.2%	10.431s	16,286,024,429	16,286,024,429	1.564	1.564	libGeant_v.so
TGeoTrap::Safety	1.2%	10.424s	29,086,043,629	29,086,043,629	0.883	0.883	libGeom.so.6.03
TObject::SetBit	1.2%	10.066s	16,806,025,209	16,806,025,209	1.502	1.502	libCore.so.6.03
__sin_avx	1.2%	10.006s	22,992,034,488	22,992,034,488	1.064	1.064	libm-2.20.so
TGeoBranchArray::UpdateNavigator	1.1%	9.682s	15,358,023,037	15,358,023,037	1.525	1.525	libGeom.so.6.03
TMXsec::SampleInt	1.0%	8.653s	17,764,026,646	17,764,026,646	1.155	1.155	libXsec.so
TPFstate::SampleReac	1.0%	8.420s	17,684,026,526	17,684,026,526	1.205	1.205	libXsec.so
GeantBasketMgr::AddTrack	1.0%	8.411s	4,034,006,051	4,034,006,051	5.129	5.129	libGeant_v.so
__ieee754_atan2_avx	1.0%	8.324s	14,672,022,008	14,672,022,008	1.388	1.388	libm-2.20.so
TGeoNavigator::Safety	1.0%	8.282s	18,938,028,407	18,938,028,407	1.086	1.086	libGeom.so.6.03

Breakdown of Time Spent in Top Functions

Function Stack	CPU Time: Self				Instructions Retired: Total	CPI Rate: Total	CPI Rate: Self	Module
	Effective Time by Utilization							
	Idle	Poor	Ok	Ideal				
__ieee754_log_avx	0.012s	3.222s	15.135s	41.966s	11.2%	0.948	0.948	libm-2.20.so
TGeoHMatrix::Multiply	0.011s	1.766s	9.981s	28.664s	9.5%	0.765	0.765	libGeom.so.6.03
__memcpy_avx_unaligned	0.012s	2.144s	11.344s	29.194s	5.0%	1.507	1.507	libc-2.20.so
TList::LinkAt	0.016s	3.019s	16.202s	41.797s	4.9%	2.202	2.202	libCore.so.6.03
TGeoNodeCache::CdDown	0.002s	0.947s	5.232s	14.429s	4.4%	0.829	0.829	libGeom.so.6.03
[vmlinux]	0.017s	7.041s	26.429s	14.881s	3.4%	2.458	2.458	vmlinux
TGeoNavigator::CdDown	0.005s	0.762s	3.753s	10.551s	3.2%	0.856	0.856	libGeom.so.6.03
TTabPhysMgr::SampleFinalStates	0.005s	0.955s	5.315s	14.663s	2.6%	1.445	1.445	libXsec.so
TGeoHMatrix::CopyFrom	0.001s	0.548s	3.039s	8.669s	2.5%	0.852	0.852	libGeom.so.6.03
__dubsin	0s	0.834s	4.631s	13.257s	2.5%	1.316	1.316	libm-2.20.so
GeantTrack_v::PropagateInVolumeSingle	0.003s	0.620s	3.472s	9.742s	2.2%	1.088	1.088	libGeant_v.so
TGeoTrap::Safety	0.005s	0.407s	2.405s	7.496s	2.1%	0.895	0.895	libGeom.so.6.03
TMXsec::Range	0.004s	0.572s	3.066s	8.306s	1.9%	1.133	1.133	libXsec.so
__sin_avx	0.001s	0.429s	2.393s	6.469s	1.5%	1.068	1.068	libm-2.20.so
GeantTrack_v::AddTrackSync	0.007s	0.641s	3.916s	10.485s	1.4%	1.856	1.856	libGeant_v.so
TFinState::SampleReac	0.003s	0.706s	3.740s	9.762s	1.4%	1.822	1.822	libXsec.so
TGeoNavigator::Safety	0.001s	0.358s	2.053s	5.632s	1.4%	1.036	1.036	libGeom.so.6.03
TPFstate::SampleReac	0.002s	0.391s	1.957s	5.509s	1.3%	1.146	1.146	libXsec.so
TMXsec::Eloss	0.001s	0.442s	2.478s	7.310s	1.3%	1.433	1.433	libXsec.so
TMXsec::SampleInt	0.001s	0.306s	1.998s	5.836s	1.2%	1.159	1.159	libXsec.so
TObject::SetBit	0.003s	0.420s	2.334s	6.049s	1.2%	1.251	1.251	libCore.so.6.03
TMXsec::ProposeStep	0.006s	0.466s	2.566s	6.736s	1.2%	1.444	1.444	libXsec.so
TGeoBranchArray::UpdateNavigator	0.002s	0.379s	2.076s	5.514s	1.2%	1.210	1.210	libGeom.so.6.03
GeantTrack_v::ComputeTransportLengthSingle	0.002s	0.408s	2.367s	6.144s	1.1%	1.373	1.373	libGeant_v.so
TPXsec::SampleReac	0.004s	0.424s	2.401s	6.376s	1.1%	1.476	1.476	libXsec.so
__cos_avx	0s	0.319s	1.698s	4.518s	1.1%	1.048	1.048	libm-2.20.so
do_cos.isra.2	0.001s	0.177s	1.162s	3.192s	1.1%	0.746	0.746	libm-2.20.so
__ieee754_atan2_avx	0.001s	0.387s	2.137s	5.527s	1.0%	1.424	1.424	libm-2.20.so
GeantTrack_v::AddTracks	0.001s	0.613s	2.948s	6.377s	1.0%	1.813	1.813	libGeant_v.so

Preliminary Conclusions

- Large amount of time spent in system
  - Kernel accounts for 3.4% of instructions, but 12.1% of time
  - Possible reason: threads waiting add to this time?
- Second largest amount of time is spent at TList::LinkAt
  - Need to learn what the list is being used for
  - Can we substitute it for a vector in some places?
- Memory alignment may be an issue
  - __memcpy_avx_unaligned with high amount of time
- Significant time spent multiplying matrices
  - Quaternions might help if those are from chained coordinate transformations
- Switch to VecGeom may improve alignment problems
- Logarithm function taking significant amount of time
  - Can we parameterize tables using $\log_2$? That might help.