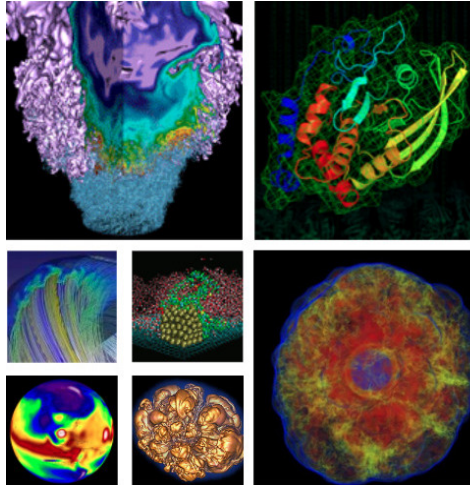


Geant4 + timemory

Updates since Lund



National Energy Research
Scientific Computing Center



U.S. DEPARTMENT OF
ENERGY

Office of
Science



Jonathan R. Madsen

✉ jrmadsen@lbl.gov

National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

September 25, 2019

- Last year, presented introduction of **timemory** into Geant4
 - Lightweight “in-situ” package for automating performance analysis
 - Designed to be very easy to use
 - Used macro `TIMEMORY_AUTO_TIMER(“”)` macro at beginning of certain high-level functions in Geant4
 - When not compiled with timemory support \Rightarrow empty macro
- Capabilities were limited: wall/user/sys/cpu/cpu-util timing + peak/page memory in 3 possible bundles (timing, memory, timing and memory)

- Library essentially underwent a complete re-write using CRTP
- The capabilities have increased **significantly**
 - HW counters, Roofline, gperftools, [rusage](#), many more...
 - Still uses `TIMEMORY_AUTO_TIMER("")` but more available
 - Provides fully capabilities from [Caliper](#) and [TAU](#)
- Library is now header-only for all the previous capabilities
- Measurements can be arbitrarily assembled into any combination
- Always-on bundles (tuple), run-time optional bundles (list), and pair of those bundles (hybrid)
- GitHub: github.com/NERSC/timemory
- Documentation: timemory.readthedocs.io

Always-on

- wall_clock
- system_clock
- user_clock
- cpu_util
- current_rss
- peak_rss

Run-time Optional

- caliper
- papi_array_t
- cuda_event
- nvtx_marker
- cupti_activity
- cupti_counters
- cpu_roofline_flops
- cpu_roofline_sp_flops
- cpu_roofline_dp_flops
- gpu_roofline_flops
- gpu_roofline_hp_flops
- gpu_roofline_sp_flops
- gpu_roofline_dp_flops
- gperf_cpu_profiler
- gperf_heap_profiler

Listing 1: Sample wall-clock Text Output

```

> [cxx] cmsExpMT : 330.661 sec real, 1 laps
> [cxx] |_BeamOn@'G4RunManager.cc':267 : 319.992 sec real, 2 laps
> [cxx] |_RunInitialization@'G4RunManagerKernel.cc':660 : 167.113 sec real, 2 laps
> [cxx] |_BuildPhysicsTables@'G4RunManagerKernel.cc':791 : 97.510 sec real, 1 laps
> [cxx] |_RunInitialization@'G4RunManager.cc':319 : 0.000 sec real, 2 laps
> [cxx] |_DoEventLoop@'G4RunManager.cc':376 : 1.242 sec real, 2 laps
> [cxx] |_InitializeEventLoop@'G4MTRunManager.cc':259 : 1.242 sec real, 2 laps
> [cxx] |_DoWork@'G4WorkerRunManager.cc':608 : 2103.221 sec real, 12 laps
> [cxx] |_BeamOn@'G4RunManager.cc':267 : 1819.951 sec real, 24 laps
> [cxx] |_RunInitialization@'G4WorkerRunManager.cc':138 : 0.322 sec real, 24 laps
> [cxx] |_RunInitialization@'G4RunManagerKernel.cc':660 : 0.278 sec real, 23 laps
> [cxx] |_BuildPhysicsTables@'G4RunManagerKernel.cc':791 : 0.213 sec real, 10 laps
> [cxx] |_RunInitialization@'G4RunManagerKernel.cc':660 : 0.025 sec real, 1 laps
> [cxx] |_BuildPhysicsTables@'G4RunManagerKernel.cc':791 : 0.024 sec real, 1 laps
> [cxx] |_DoEventLoop@'G4WorkerRunManager.cc':221 : 1692.912 sec real, 22 laps
> [cxx] |_ProcessOneEvent@'G4WorkerRunManager.cc':266 : 1692.886 sec real, 122 laps
> [cxx] |_GenerateEvent@'G4WorkerRunManager.cc':279 : 0.034 sec real, 121 laps
> [cxx] |_TerminateOneEvent@'G4RunManager.cc':423 : 0.002 sec real, 100 laps
> [cxx] |_DoEventLoop@'G4WorkerRunManager.cc':221 : 0.000 sec real, 1 laps
> [cxx] |_ProcessOneEvent@'G4WorkerRunManager.cc':266 : 0.000 sec real, 1 laps
> [cxx] |_GenerateEvent@'G4WorkerRunManager.cc':279 : 0.000 sec real, 1 laps

```

Listing 2: Sample page RSS JSON Output

```
"type": "page_rss",
"description": "resident set size of memory pages",
"unit_value": 1000000,
"unit_repr": "MB",
"graph": [
  {
    "hash": 7014545444386178527,
    "prefix": "> [cxx] cmsExpMT",
    "depth": 0,
    "entry": {
      "cereal_class_version": 0,
      "is_transient": true,
      "laps": 1,
      "display": 973.774848,
      "repr_data": 973.774848,
      "value": 1011580928,
      "accum": 973774848
    }
  },
  {
    "hash": -180666760766870264,
    "prefix": "> [cxx] |_BeamOn@'G4RunManager.cc':267",
    "depth": 1,
    "entry": {
      "is_transient": true,
      "laps": 2,
      "display": 762.085376,
      "repr_data": 762.085376,

```

Listing 3: `auto_timer`

```
1 void foo()  
2 {  
3     TIMEMORY_AUTO_TIMER("");  
4     // ...  
5 }
```

Listing 4: Custom specification

```
1 using always_t = component_tuple < wall_clock,   cpu_clock >;  
2 using option_t = component_list  < system_clock, peak_rss >;  
3 using hybrid_t = auto_hybrid    < always_t,    option_t >;  
4  
5 void bar()  
6 {  
7     TIMEMORY_MARKER(hybrid_t, "");  
8     // ...  
9 }
```

- There is no restrictions on nesting \Rightarrow start and stop in any order
- This enables timemory to be used within Geant4 in *very* non-traditional performance analysis contexts
 - For the Geant ECP Project, attached timemory to all [G4Track](#) instances \Rightarrow post-processed JSON to analyze per-particles processing time w.r.t. number of instances

```
1 G4Track::G4Track(G4DynamicParticle* dynParticle,  
2                 G4double aValueTime,  
3                 const G4ThreeVector& aValuePosition)  
4 {  
5     auto pname = dynParticle->GetDefinition()->GetParticleName();  
6     trackTimer = new G4AutoTimer(pname, 0);  
7 }
```


Non-traditional Use Cases

```
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 38
> [cxx]      |_gamma    : 0.000 sec real, 2 laps, depth 37
> [cxx]      |_e-      : 0.000 sec real, 2 laps, depth 37 (exclusive: 30.3%)
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 38 (exclusive: 28.4%)
...
> [cxx]      |_e+      : 0.000 sec real, 1 laps, depth 47
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 47 (exclusive: 23.2%)
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 47 (exclusive: 45.0%)
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_proton   : 0.001 sec real, 1 laps, depth 45
> [cxx]      |_neutron  : 0.001 sec real, 1 laps, depth 46 (exclusive: 25.9%)
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 47
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 47 (exclusive: 87.8%)
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_proton   : 0.001 sec real, 1 laps, depth 46 (exclusive: 6.3%)
> [cxx]      |_Al27     : 0.001 sec real, 1 laps, depth 47 (exclusive: 8.9%)
> [cxx]      |_gamma    : 0.001 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 48
> [cxx]      |_neutron  : 0.001 sec real, 1 laps, depth 43 (exclusive: 35.3%)
> [cxx]      |_gamma    : 0.000 sec real, 1 laps, depth 44 (exclusive: 14.5%)
> [cxx]      |_neutron  : 0.000 sec real, 1 laps, depth 45 (exclusive: 94.7%)
> [cxx]      |_W183     : 0.000 sec real, 1 laps, depth 46
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 44 (exclusive: 66.9%)
> [cxx]      |_e-      : 0.000 sec real, 1 laps, depth 45
> [cxx]      |_proton   : 0.002 sec real, 1 laps, depth 43 (exclusive: 4.4%)
> [cxx]      |_Al27     : 0.001 sec real, 1 laps, depth 44
```

Non-traditional Use Cases

PARTICLE	WALL	CPU	PERC_WALL	PERC_CPU	AGG_WALL	AGG_CPU	LAPS
e-	5092082.068	4523577.476	59.901 %	59.140 %	59.901 %	59.140 %	24999623
gamma	1453282.218	1394334.528	17.096 %	18.229 %	76.996 %	77.369 %	10511498
neutron	470668.394	413414.379	5.537 %	5.405 %	82.533 %	82.774 %	656765
proton	458564.377	400630.414	5.394 %	5.238 %	87.927 %	88.012 %	509989
pi+	279696.042	236695.434	3.290 %	3.094 %	91.218 %	91.106 %	8407
pi-	231377.604	210396.925	2.722 %	2.751 %	93.939 %	93.857 %	9533
anti_proton	164314.328	149861.520	1.933 %	1.959 %	95.872 %	95.816 %	745
pi0	125214.507	111033.410	1.473 %	1.452 %	97.345 %	97.268 %	8280
e+	53113.975	52151.025	0.625 %	0.682 %	97.970 %	97.949 %	935320
016	38601.293	36932.746	0.454 %	0.483 %	98.424 %	98.432 %	207421
Al27	17994.699	14117.837	0.212 %	0.185 %	98.636 %	98.617 %	61435
C12	11972.456	12024.187	0.141 %	0.157 %	98.777 %	98.774 %	65480
Cu63	11202.182	10357.312	0.132 %	0.135 %	98.908 %	98.909 %	85695
eta	10178.519	10093.700	0.120 %	0.132 %	99.028 %	99.041 %	438
Fe56	10167.581	7785.565	0.120 %	0.102 %	99.148 %	99.143 %	49170
kaon+	5225.957	5197.097	0.061 %	0.068 %	99.209 %	99.211 %	472
kaon0L	5001.100	4928.734	0.059 %	0.064 %	99.268 %	99.276 %	366
mu+	4949.642	4260.876	0.058 %	0.056 %	99.326 %	99.331 %	3332
mu-	4887.122	4202.955	0.057 %	0.055 %	99.384 %	99.386 %	646
kaon0S	4704.240	3966.241	0.055 %	0.052 %	99.439 %	99.438 %	346