# Performance Measurement Tools (for parallel application)

Philippe Canal, Daniel Elvira, Krzysztof Genser,
Soon Yung Jun, Jim Kowalkowski, Marc Paterno
*Fermilab*

Annual Concurrency Meeting
February 4-6, 2013
Fermilab

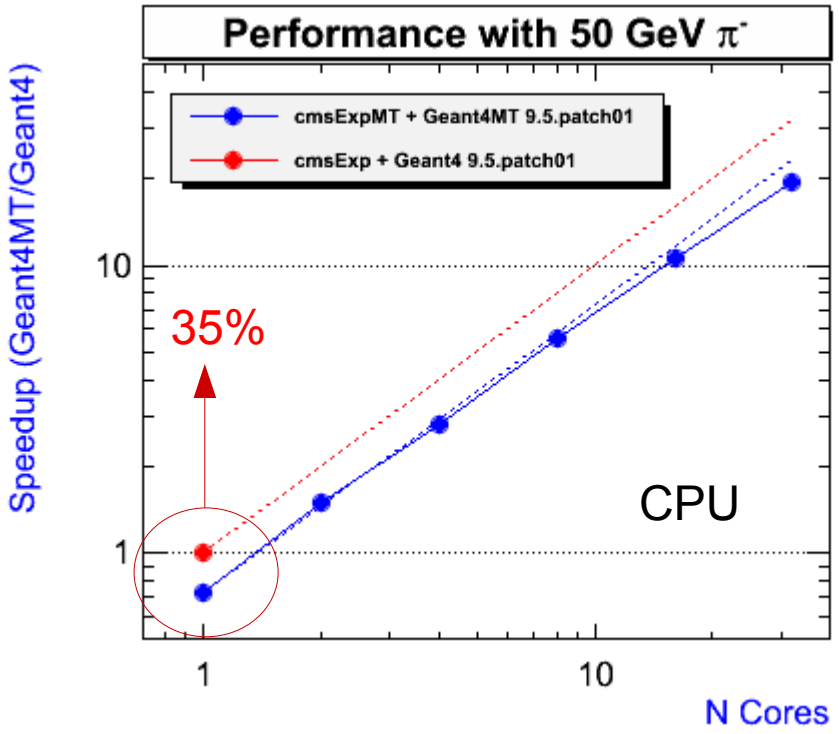
# Introduction

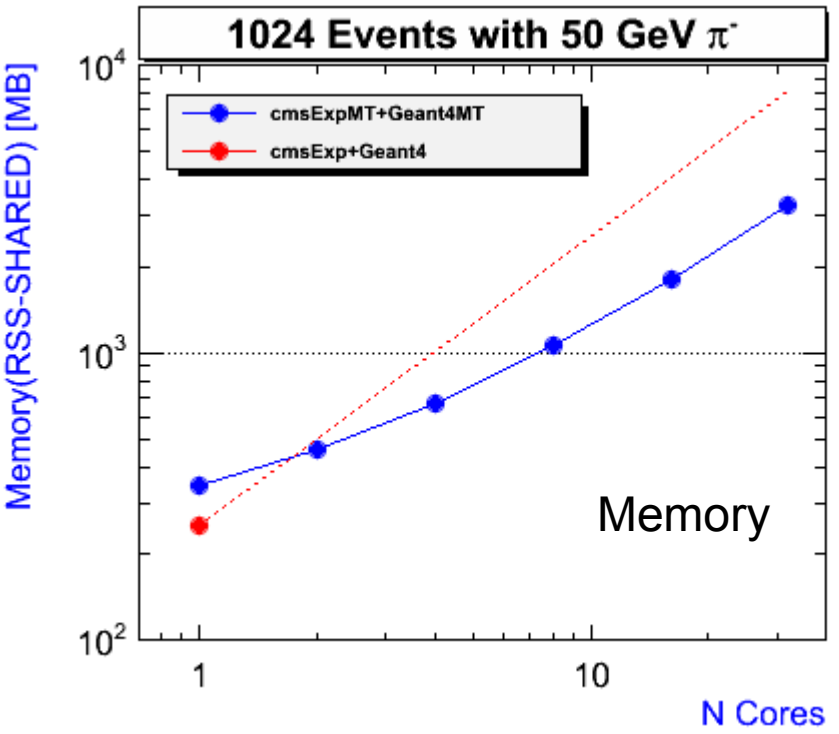
- Necessity of performance analysis
  - benchmarking/monitoring
  - efficient use of resources
  - optimization and tuning
- Emerging parallelism
  - parallel applications
  - heterogeneous systems
- Performance tools are essential
  - complexity of applications
  - diversity of architectures



# Problem Statement

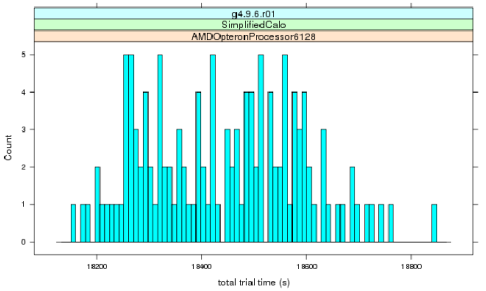
- Search tools and libraries for parallel applications
  - profiling multithreaded applications of Geant4
  - performance evaluation for GPGPU codes
- A user application: cmsExpMT (cms geom + b-field map)

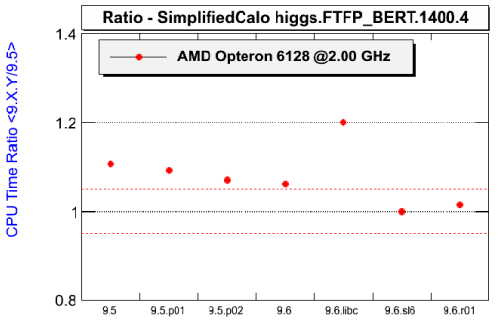


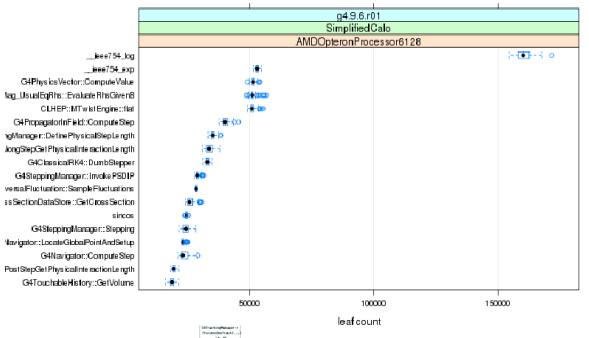



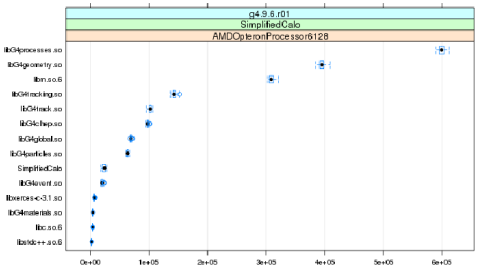
# Review of Geant4 Performance Profiling

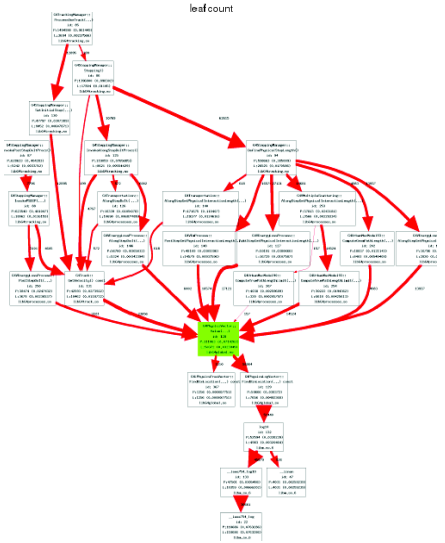
- Tools
  - FAST (CPU)
  - IgProf (memory)
- Metrics
  - time (version, physics)
  - leaf counts (funs, libs)
  - call path analysis
  - heap (live, max, total)
  - statm (vsize/rss)
  - number of tracks and steps

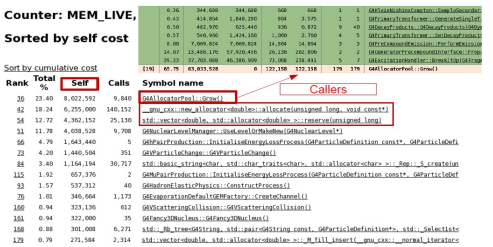





| Name | Particle | Mean | Sigma | Low | 1/4Quad | Median | 3/4Quad | High |
|---|---|---|---|---|---|---|---|---|
| 1 | Nstep e- | 1.9214e+07 | 6.1294e+05 | 1.7891e+07 | 1.8846e+07 | 1.9196e+07 | 1.9663e+07 | 2.0736e+07 |
| 2 | Nstep e+ | 2.6696e+06 | 1.3309e+05 | 2.4400e+06 | 2.5659e+06 | 2.6553e+06 | 2.7668e+06 | 3.0220e+06 |
| 3 | Nstep gamma | 1.6601e+07 | 5.0588e+05 | 1.5485e+07 | 1.6303e+07 | 1.6598e+07 | 1.6969e+07 | 1.7838e+07 |
| 4 | Nstep N | 7.0410e+06 | 7.5337e+05 | 5.0027e+06 | 6.5169e+06 | 7.2666e+06 | 7.5037e+06 | 8.0701e+06 |
| 5 | Nstep other | 3.9299e+05 | 4.3769e+04 | 2.7979e+05 | 3.5370e+05 | 4.0674e+05 | 4.2379e+05 | 4.5151e+05 |
| 6 | Nstep p | 3.2549e+05 | 3.5179e+04 | 2.3170e+05 | 3.0530e+05 | 3.3502e+05 | 3.4787e+05 | 3.7748e+05 |
| 7 | Nstep pi- | 4.2773e+04 | 4.6553e+03 | 3.1341e+04 | 3.9691e+04 | 4.4302e+04 | 4.6106e+04 | 5.0434e+04 |
| 8 | Nstep pi+ | 4.2990e+04 | 4.7467e+03 | 2.9472e+04 | 3.9956e+04 | 4.4154e+04 | 4.6512e+04 | 4.9173e+04 |
| 9 | Ntrack e- | 1.4359e+07 | 4.3076e+05 | 1.3392e+07 | 1.4117e+07 | 1.4360e+07 | 1.4676e+07 | 1.5402e+07 |
| 10 | Ntrack e+ | 3.6313e+05 | 1.3851e+04 | 3.3542e+05 | 3.5306e+05 | 3.6146e+05 | 3.7300e+05 | 3.9866e+05 |
| 11 | Ntrack gamma | 4.4820e+06 | 1.5189e+05 | 4.1621e+06 | 4.3837e+06 | 4.4732e+06 | 4.5919e+06 | 4.8610e+06 |
| 12 | Ntrack N | 2.1883e+05 | 2.3542e+04 | 1.5538e+05 | 2.0211e+05 | 2.2634e+05 | 2.3322e+05 | 2.5154e+05 |
| 13 | Ntrack other | 2.1537e+05 | 2.2964e+04 | 1.5382e+05 | 1.9906e+05 | 2.2243e+05 | 2.2942e+05 | 2.4733e+05 |
| 14 | Ntrack p | 8.5583e+04 | 9.1407e+03 | 6.0811e+04 | 7.9987e+04 | 8.8626e+04 | 9.1229e+04 | 9.8422e+04 |
| 15 | Ntrack pi- | 3.0006e+03 | 3.3441e+02 | 2.1540e+03 | 2.7600e+03 | 3.1005e+03 | 3.2040e+03 | 3.4680e+03 |
| 16 | Ntrack pi+ | 3.1720e+03 | 3.3850e+02 | 2.2400e+03 | 2.9900e+03 | 3.2775e+03 | 3.4060e+03 | 3.6460e+03 |

# Metrics for Parallel Applications

- Multithreaded applications
  - speedup, scalability
  - memory (cache miss, TLB miss, coherence)
  - communication
  - I/O
- Platform dependence
  - NUMA
  - distributed memory models
- GPU applications
  - memory throughput
  - arithmetic intensity
  - occupancy vs. latency
  - ILP vs. TLP
  - locality (spatial/temporal)

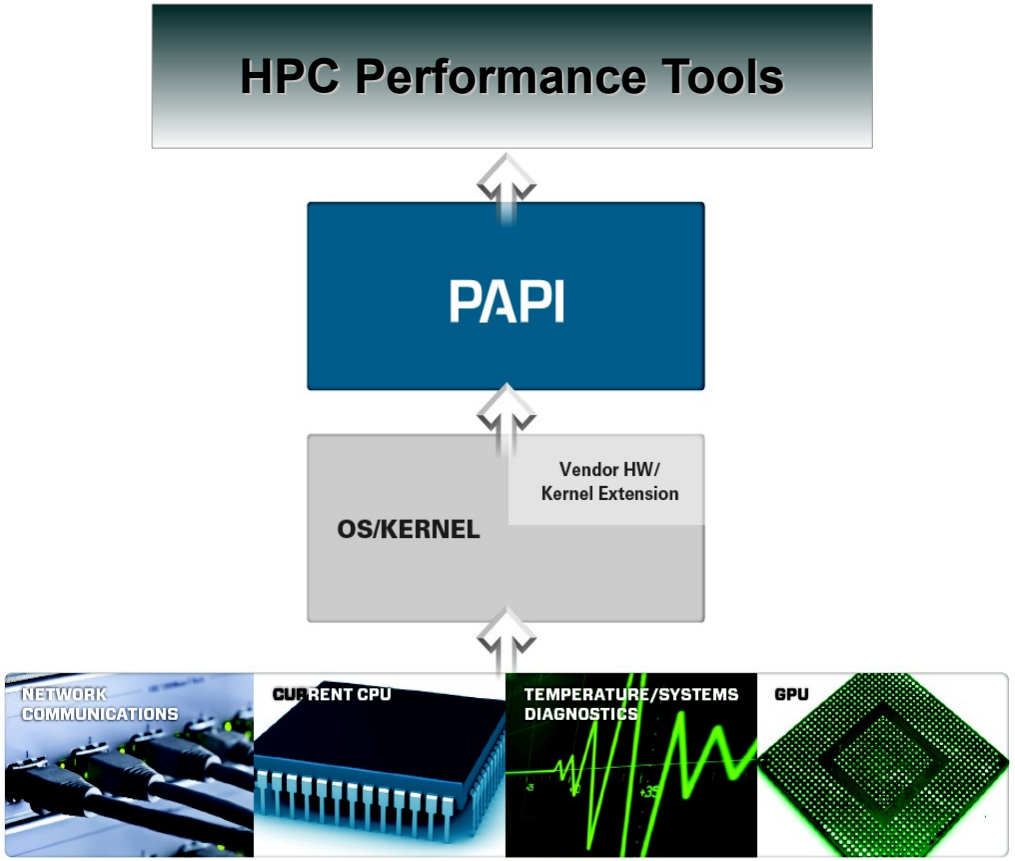
# Performance Tool Requirement

- Support parallelism and multi-threaded applications
- Open source (support Linux)
- Transparent instrumentation (applicable on binaries)
- Light time overhead and precision of measurement
- Advanced analysis (tracing, callgraph)
- Easy to use, but extensible
- Documentation and support

# Performance Tools: A Short List

- Community infrastructures:
  - PAPI and MuMMI (UTK)
  - Paradyn (DynInst, MRNet) (UW, UMD)
- Integrated tool kits:
  - HPCToolkit (Rice)
  - Open|SpeedShop (Krell)
  - TAU (Oregon)
  - nvvp (NVIDIA)
- Tracing tools:
  - Jumpshot (ANL)
  - Scalasca (Jülich)

# PAPI (Performance API)

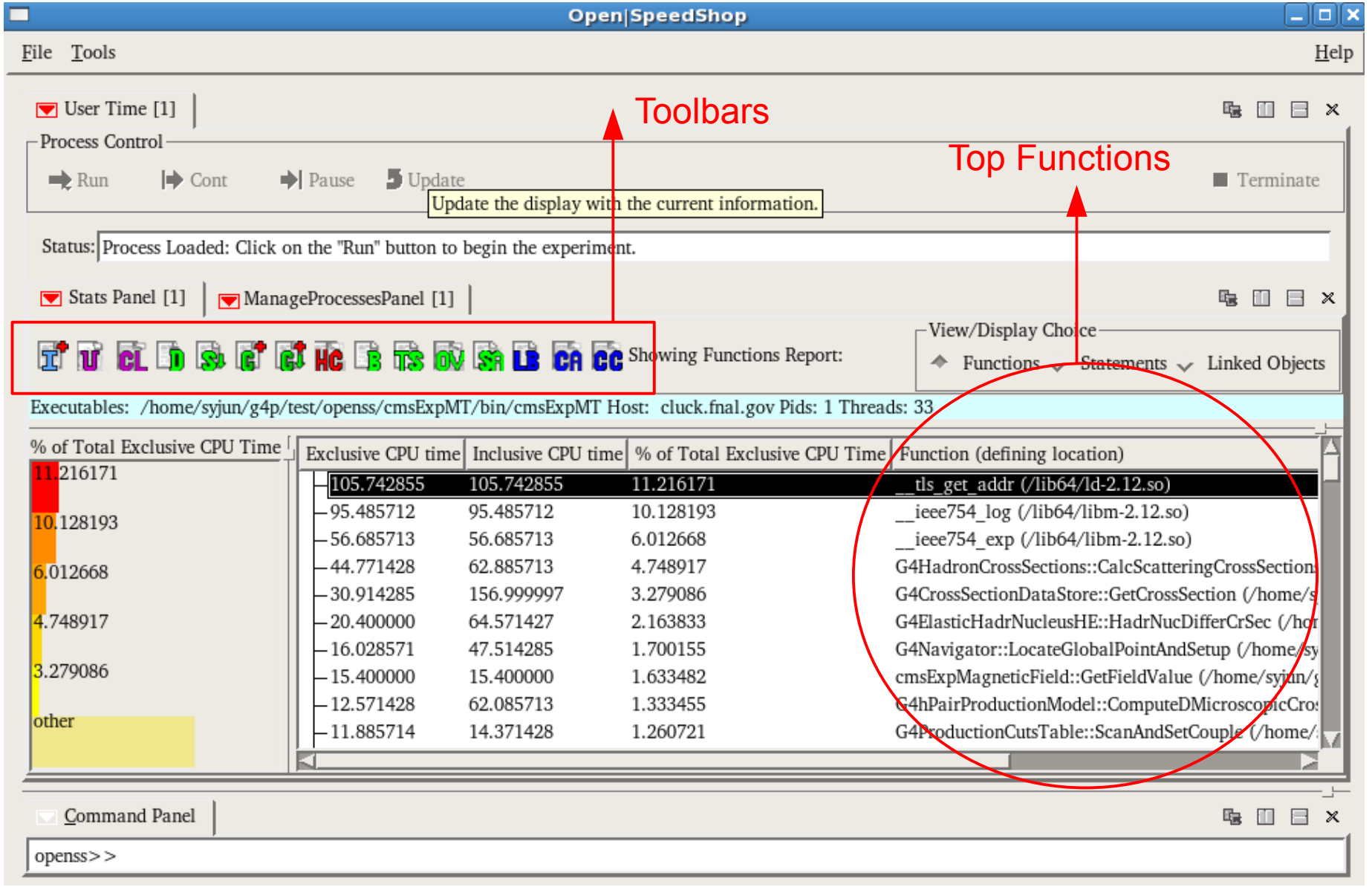



- A standard API to access hardware performance counters
- Relation between software performance and processor events
- Event metrics : platform specific metrics, cache hit/miss, Flops, power consumption (MuMMI)

# Open|Speedshop

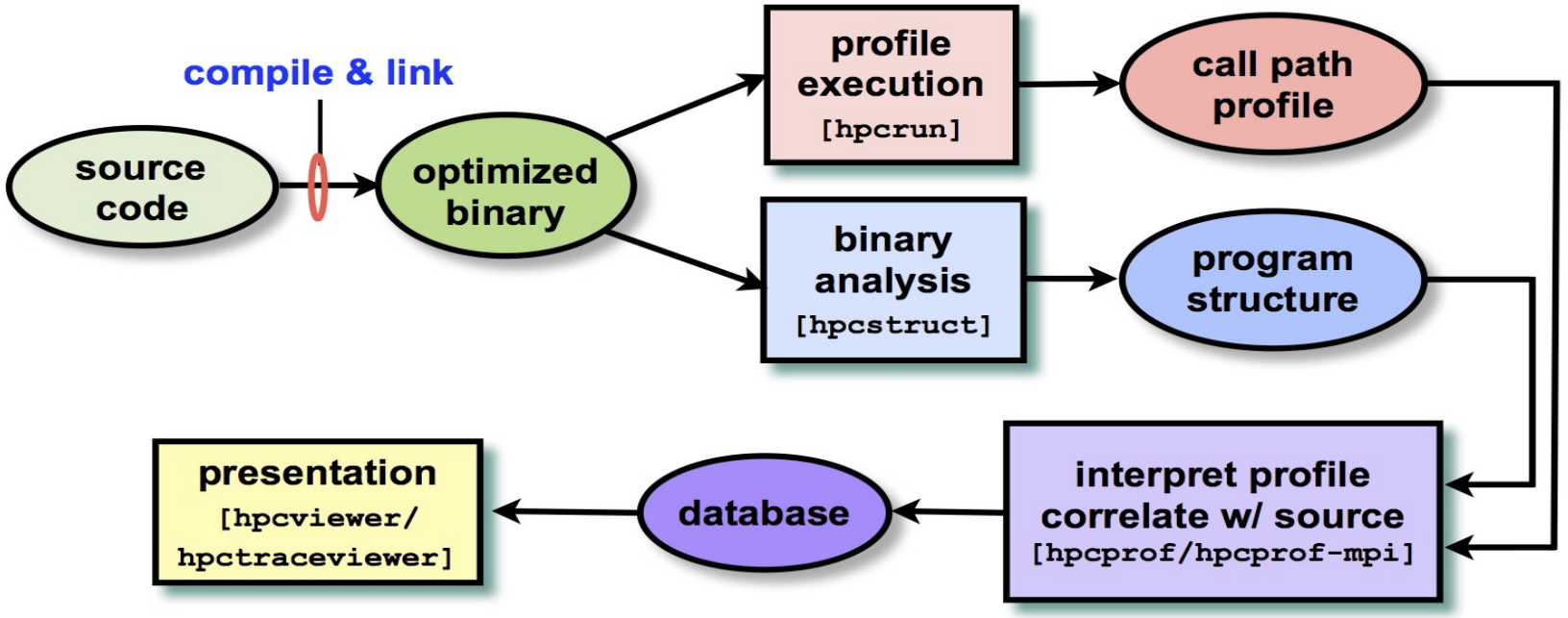
- Comprehensive performance analysis for sequential, multithreaded, and MPI applications
- The base functionality includes
  - sampling experiments
  - support callstack analysis
  - hardware performance counters
  - multi-threaded, MPI profiling and tracing
  - floating point exception analysis
- GUI and CLI (command line instruction)
- Almost ready to support GPU (95%) and MIC (if funded)

# Default View and Stats Panel



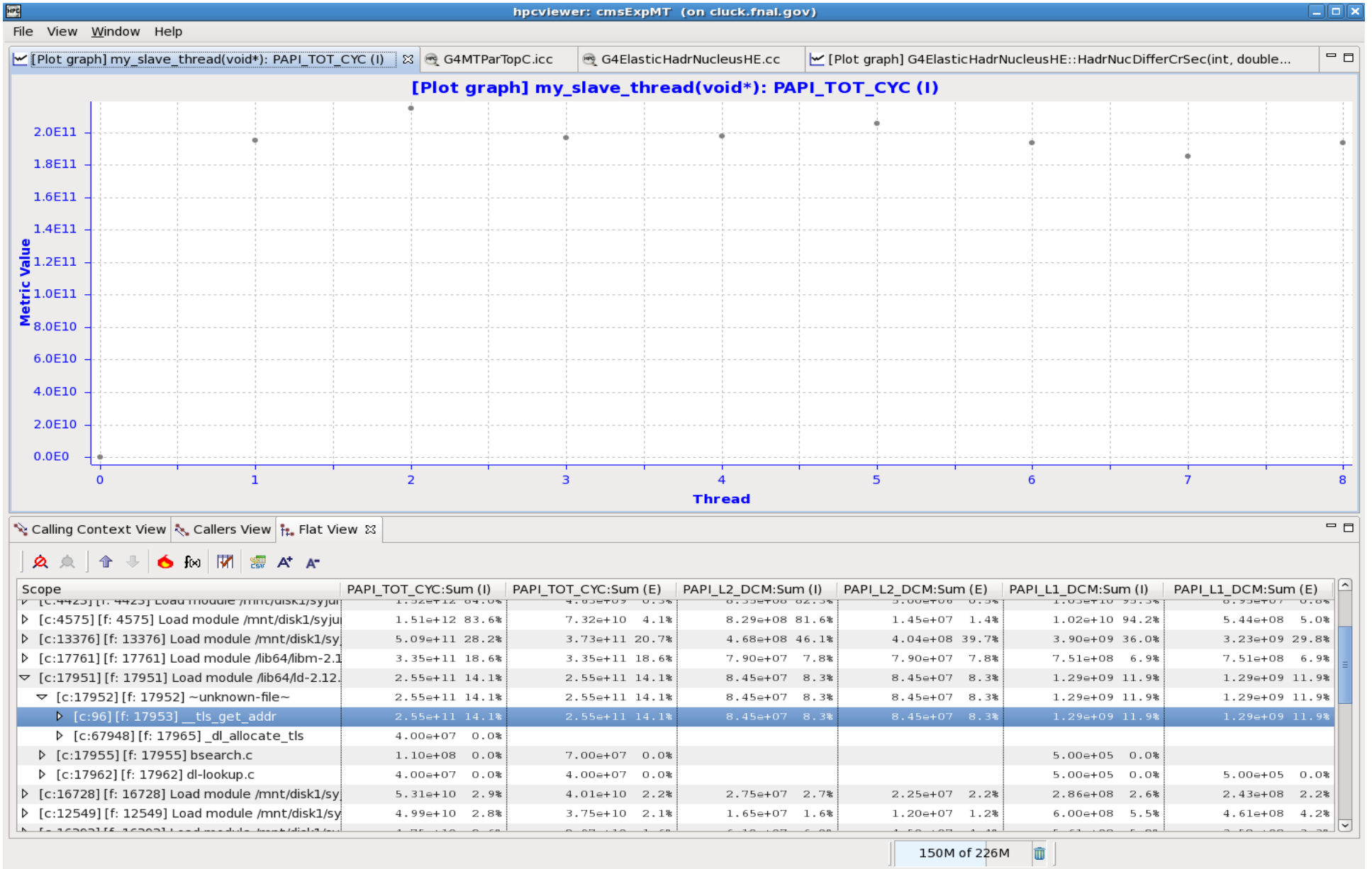
# HPCToolkit

- Overview of HPCTOOLKIT tool's work flow (from manual)




- Code centric view, GUI and text-base flat profile
- Supporting performance analysis of heterogeneous architecture (hybrid CPU/GPU)

# HPCToolkit: hpcviewer

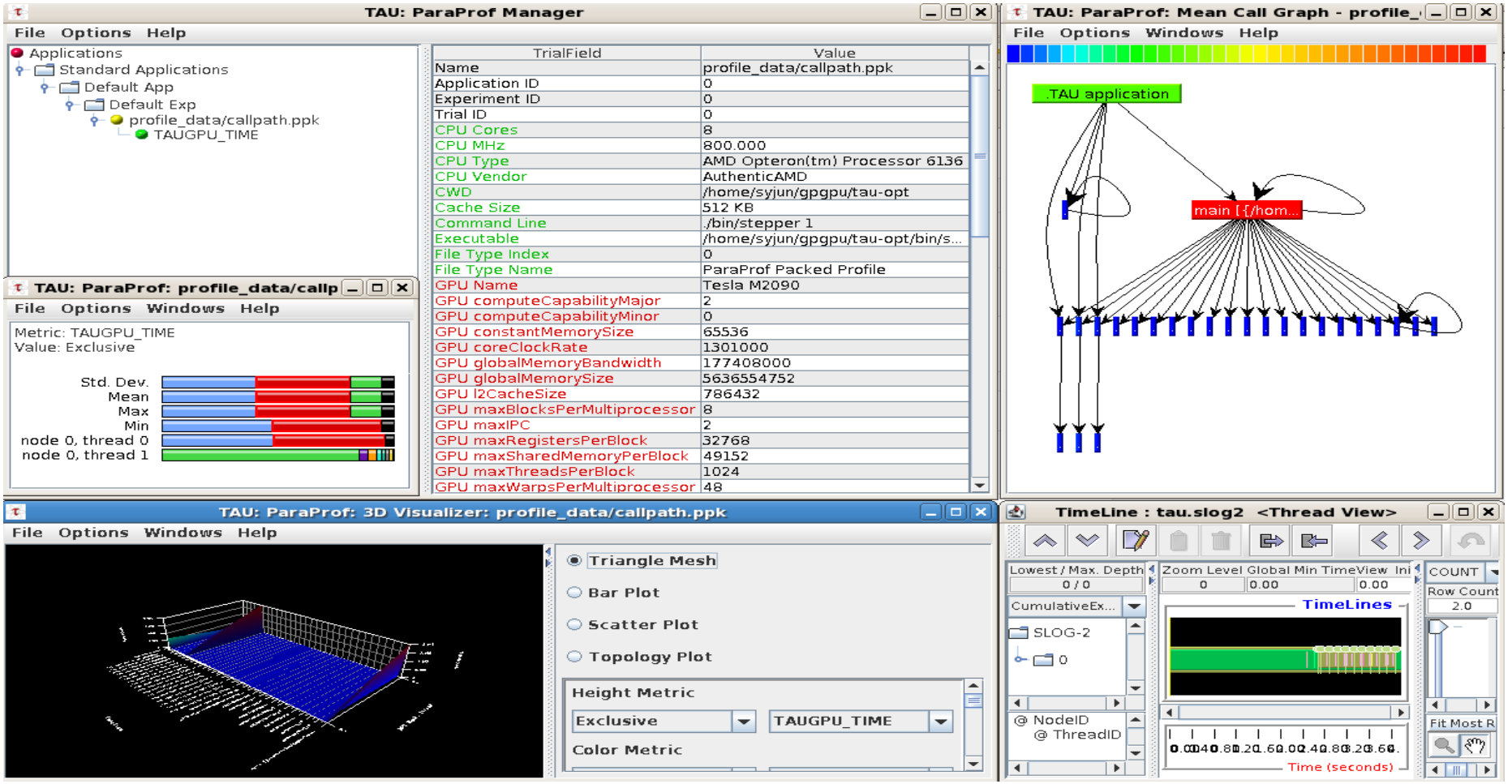


# TAU (Tuning Application Utilities)

- Dynamic, compiler based, source based Instrumentation
- Analysis tools
  - ParaProf
  - PerfExplorer
  - Tracer (Jumpshot, vampir)
- Various built-in graphical presentations
- Capable to measure performance of GPU with cuda/openCL
- Disadvantage: compiler/source-based instrumentation

# TAU for CUDA/openCL?

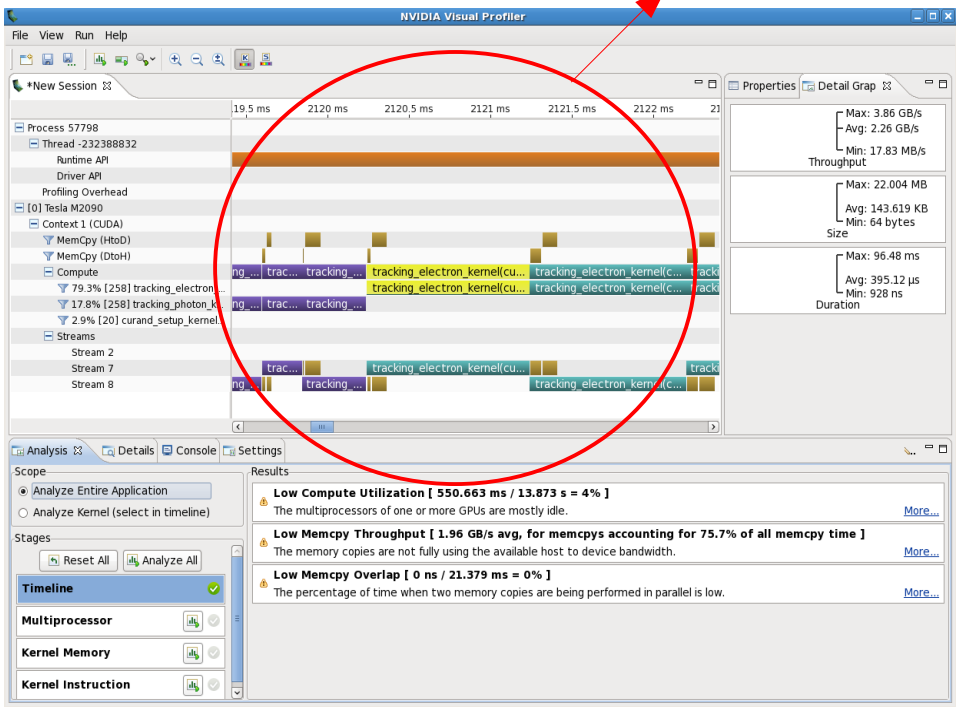
- Test tau with cupti on a Runga-Kutta stepping algorithm

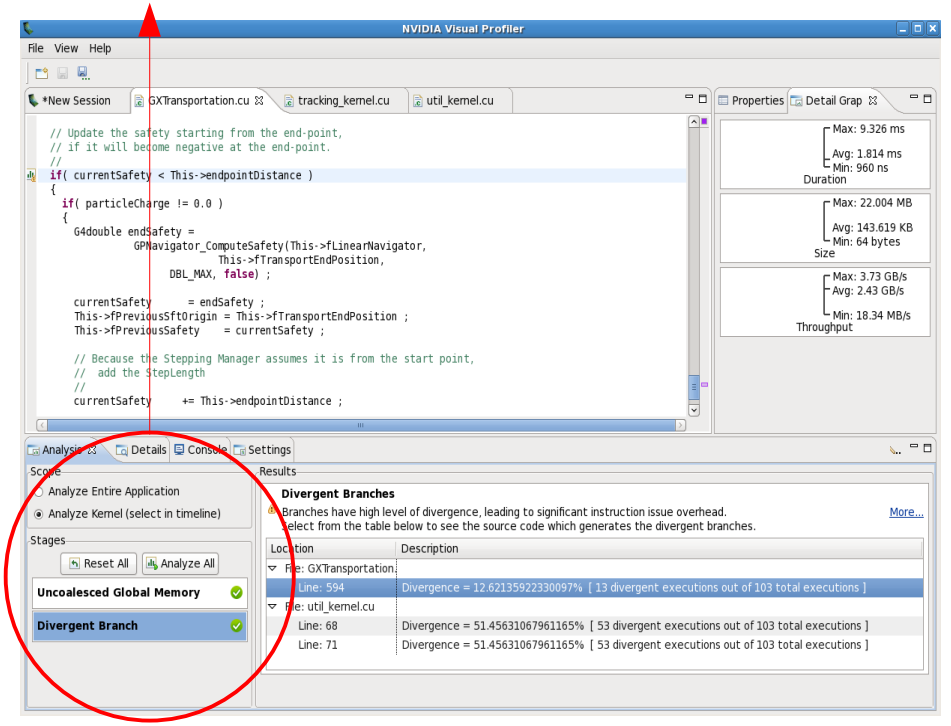



Provides general profiling information (H2D, kernel, D2H), but no details for 'inside' device codes

# NVIDIA GPU Tools

- CUPTI, NVML, occupancy calculator
- nvvp (nvprof) : timeline, kernel analysis with source




- No good way to measure exclusive time for each device function

# Other Tools: Another Short List

- Community Infrastructure and libraries
  - CBTF (Component Based Tool Framework) (Krell)
  - GPTL (General Purpose Timing Library) (ORNL)
  - gperftools (Google Peformance Tools)
- Integrated Tools
  - Intel: VTune Amplifer, Cluster/Parallel Studio (License)
  - AMD: CodeAnalyst, APP Profiler/Kernel Analyzer (openCL)
  - IBM: HPMToolkit
- Tracing Tools
  - KOJAK (OPARI,EPILOG,EARL,EXPERT/CUBE)
  - Vampir (License)

# Summary

- Profiling serial and parallel codes is a critical step in assessing the efficiency of software development
- Various tools and libraries are available for performance profiling and analysis
- Performance analysis is domain specific

  (domain knowledge, architecture, programing models)
- Collaborating with ASCR institutes
  - Bob Lucas, Pedro Diniz (ISI)
  - Rob Fowler, Paul Ruth (RENCI)
  - Boyana Norris (ANL)