Geant4 Computing Performance Task with OpenlSpeedshop

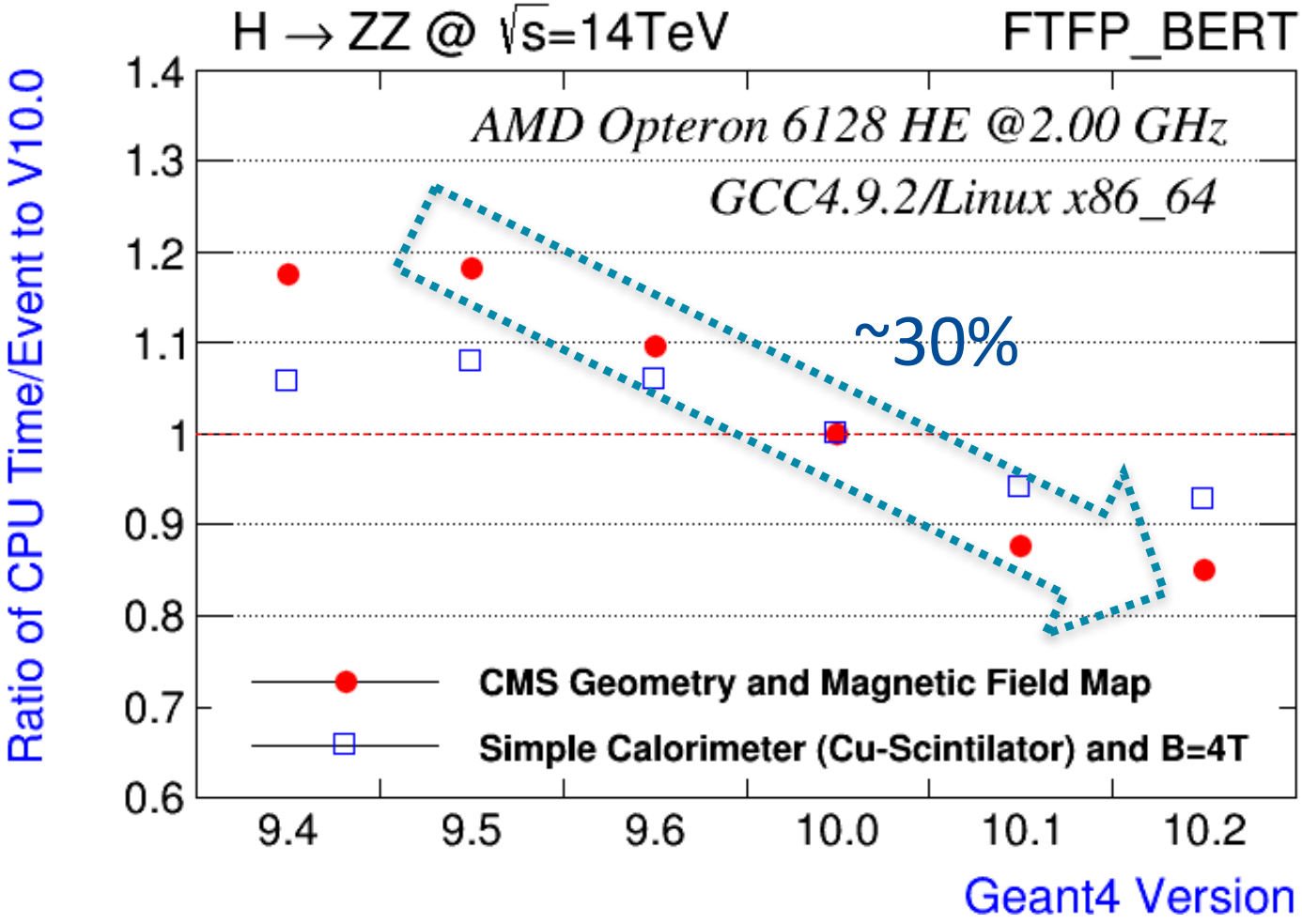
Soon Yung Jun, Krzysztof Genser, Philippe Canal (Fermilab)

21st Geant4 Collaboration Meeting, Ferrara, Italy

Sept. 12 - 16, 2016

Geant4 Computing Performance (Review 1/4)

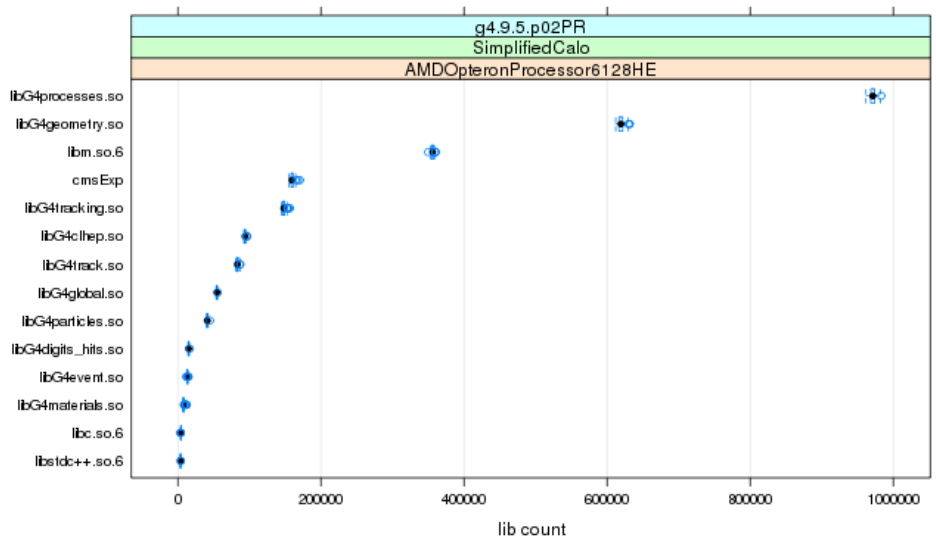
- Progressive improvements over the last five years

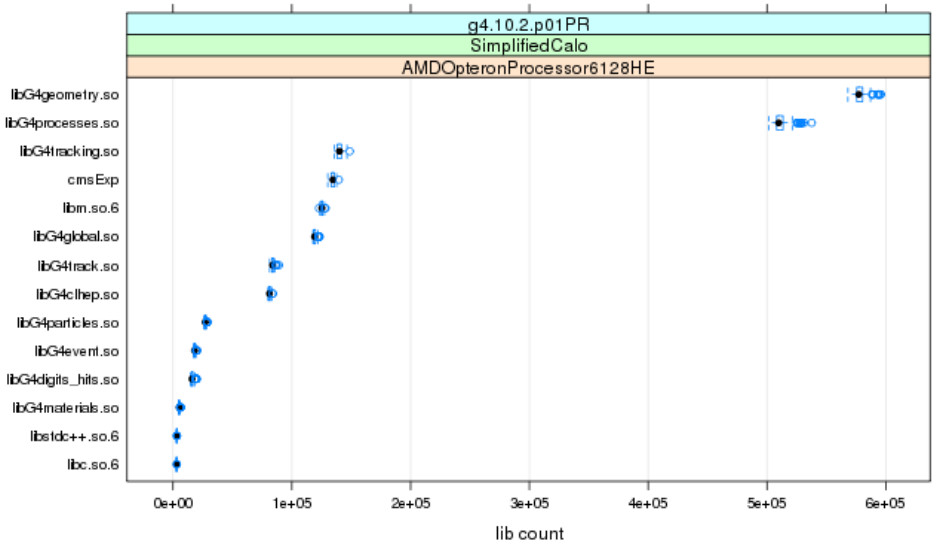


(more information at https://g4cpt.fnal.gov/g4p/prplots/cpu_by_version.html)

Geant4 Computing Performance (Review 2/4)

- Changes in linked objects (libraries): from 9.5 to 10.2



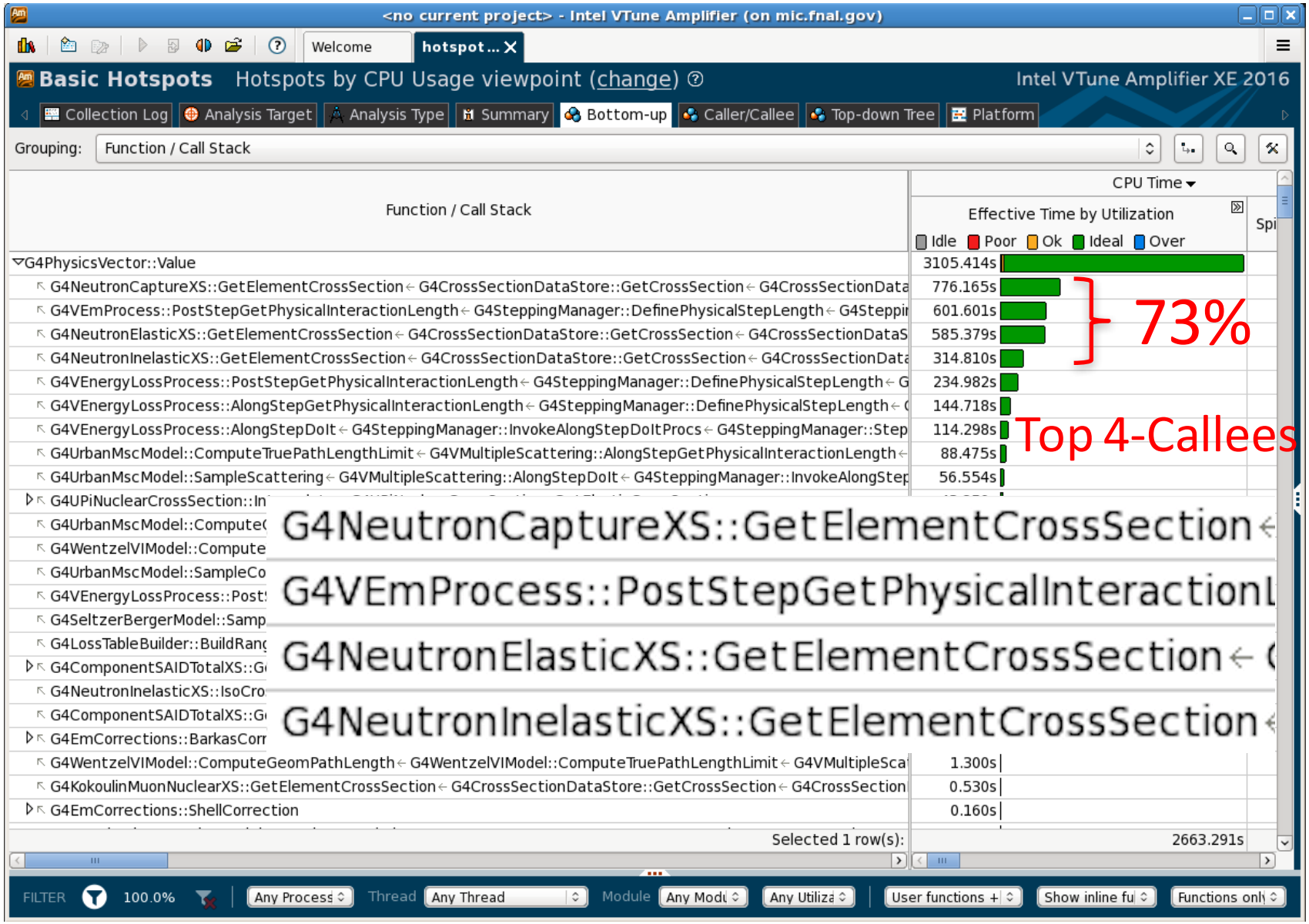


Library	9.5	10.2	9.5/10.2
libG4processes	971348	577240	1.68
libG4geometry	619304	509816	1.21
libm.so	356072	125249	2.90
libG4tracking	148078	140128	1.06

- Also many changes in hot functions from 9.5 to 10.2

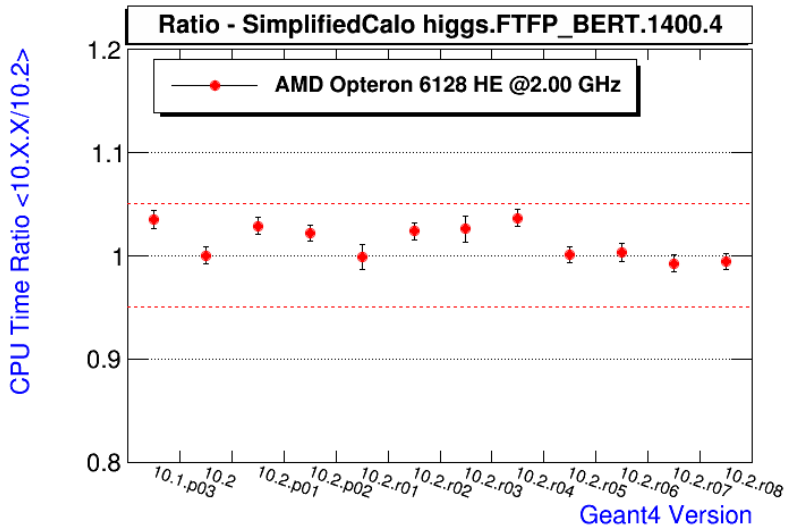
Geant4 Computing Performance (Review 3/4)

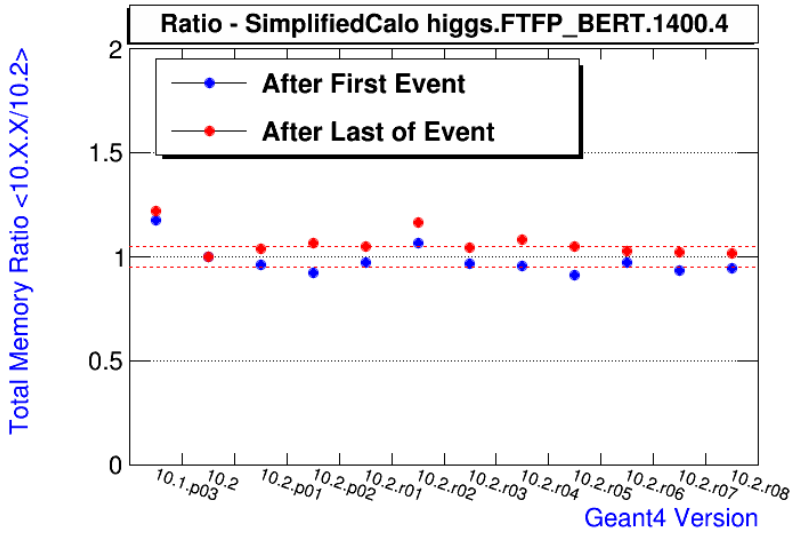
- The current hot spot: G4PhysicsVector::Value (~6%)



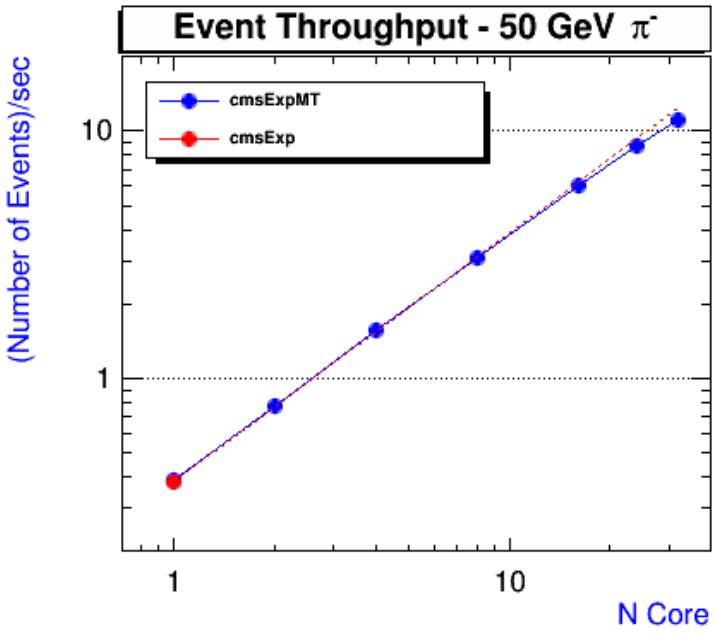
Geant4 Computing Performance (Review 4/4)

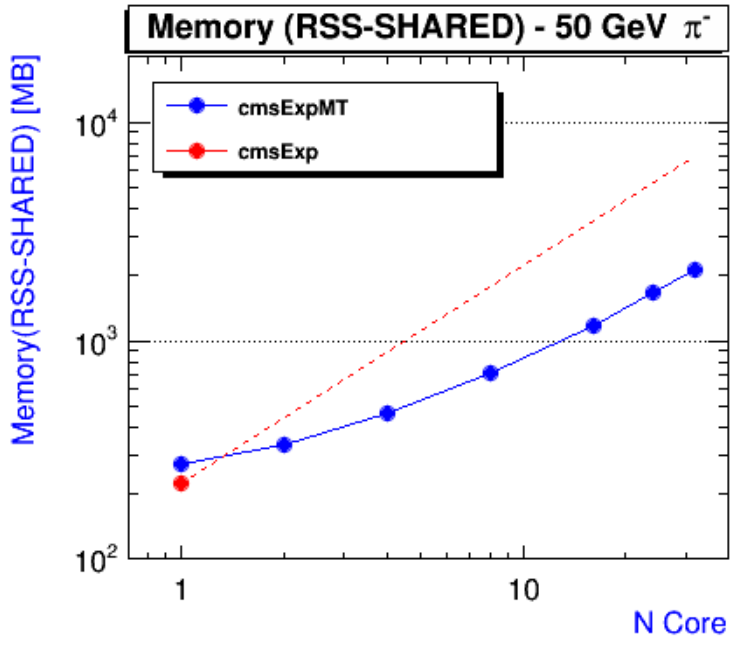
- 10.2 cycle: CPU/mem under control while improving physics





- Geant Multi-threading: good scalability and memory reduction





Current Profiling Tools and Primary Metrics

- Tools
  - FAST (CPU Profiling)
  - IgProf (Memory profiling)
  - OpenlSpeepshop (Geant4 MT profiling)
- Metrics (Observables)
  - Timing
    - Event
    - Function (exclusive and inclusive)
    - Library
  - Memory
    - Footprint (live, total)
    - statm (top)
  - Multi-threaded applications
    - Event throughput and memory reduction

Challenges and New Requirements

- Flat profile
  - No obviously reducible hot spots or bottlenecks
  - Need more specific metrics
- Measurement uncertainty
  - Fixed rate for sampling and call path collections (100Hz)
  - Different optimization paths when recompiling Geant4 libs ?
  - Rerun reference for each release (4 nodes x 32 jobs)
- Limited extensibility
  - Not all tools are capable for profiling multi-threaded applications
  - Not capable for profiling codes on new hardware architectures
- Identify the better profiling tool(s) for G4CPT
  - Pick a right (integrated) tool to meet all new requirements
  - Surveyed several tools and selected OpenlSpeedshop

Introduction to OpenlSpeedshop

- Open source (the Krell institute, https://openspeedshop.org/)
- Sampling Experiments (Light weighted)
- Support for Call Stack Analysis
- Hardware Performance Counters
- Memory Function Tracing
- MPI Profiling and Tracing
- I/O Profiling and Tracing
- Floating Point Exception Analysis
- POSIX Thread Function Tracing (Multithreading-capable)
- Flexible analysis options (GUI, command line, online)
- Support Intel MIC architecture and NVIDIA CUDA
- Well documented and supported

Sampling Experiments in OSS

- pcsamp (periodic sampling the program counters)
  - Low overhead overview of time distribution
- usertime (call path profiling)
  - Inclusive and exclusive timing data
  - Call paths, caller and callee relationships
- hwcsamp (periodic sampling hardware counters)
  - Profile of hardware counter events (PAPI events)
- mem (memory tracing)
  - Call paths for memory related function call events
  - Aggregate and individual rank, thread, or processing timings
- pthreads (POSIX thread tracing)
- io (I/O tracing)
- Many other useful experiments

Measurement Overheads and Output Size

- pcsamp: exclusive time - insensitive to sampling frequency (default 100Hz, set to 100000 Hz for G4CPT)

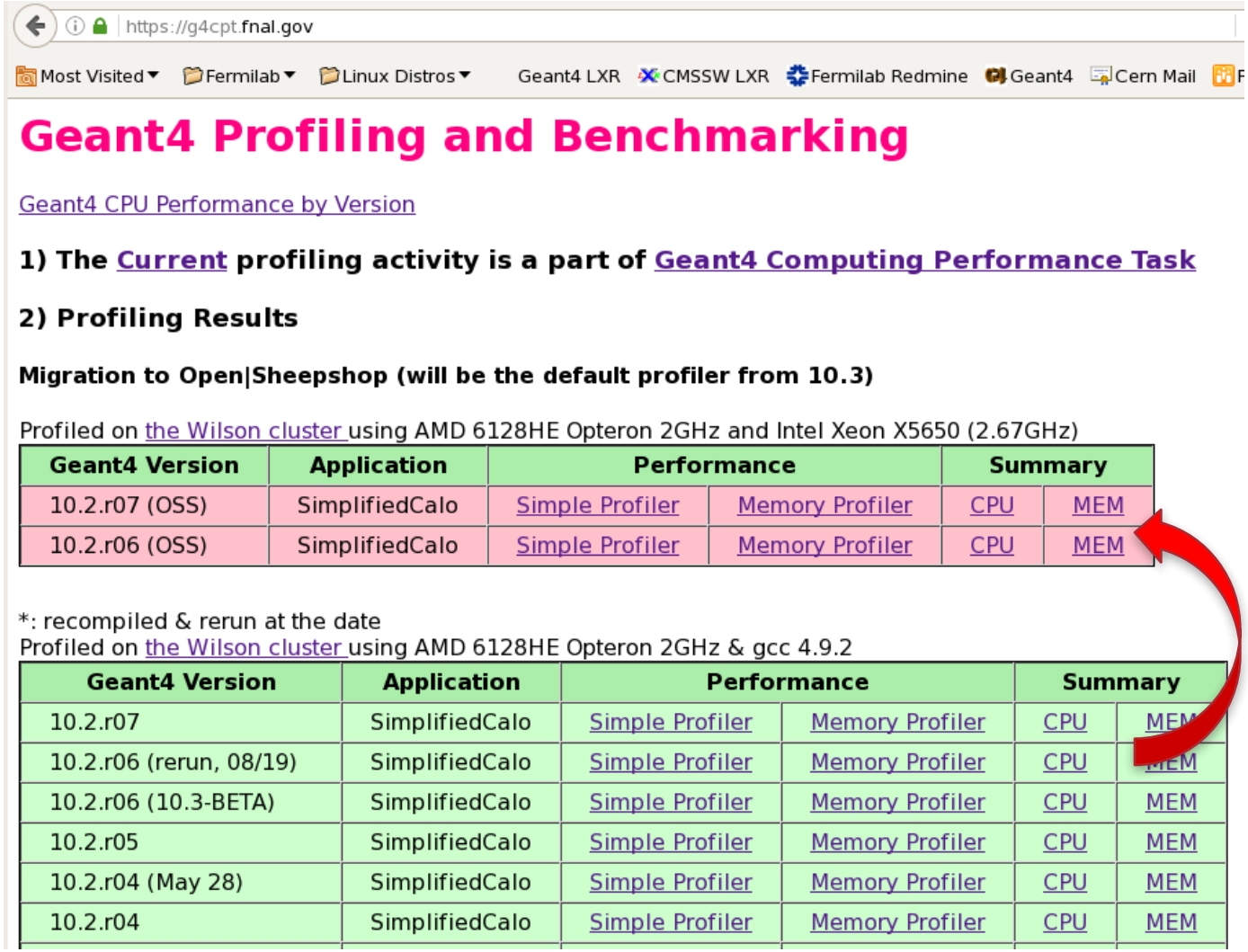
Frequency	Time(sec)	OverHead(%)	DB size(MB)
base :	52.20	-	
50 Hz:	52.27	0.13	0.376832
100 Hz:	52.62	0.80	0.486400
200 Hz	52.36	0.31	0.607232
500 Hz	52.98	1.49	0.811008
1000 Hz:	52.65	0.86	0.971776
10000 Hz:	52.76	1.07	1.012736

- usertime: inclusive time and call paths – large overhead (default 35, set to 100Hz): similar overhead for hwcsamp

Frequency	Time(sec)	OverHead(%)	DB size(MB)
base :	52.80	-	
35 Hz:	53.89	2.06	1.087488
50 Hz:	54.33	2.90	1.430528
100 Hz:	56.21	6.46	2.355200
200 Hz	60.25	14.11	4.208640
1000 Hz:	92.84	75.83	18.725888

Migration Plan and Status:

- Replace SimpleProfiler (FAST) with OpenlSpeedshop (OSS)
- All information provided by old tools are reproduced with OSS and results have been posted since 10.2.r06



https://g4cpt.fnal.gov

Most Visited ▾ Fermilab ▾ Linux Distros ▾ Geant4 LXR CMSSW LXR Fermilab Redmine Geant4 Cern Mail

Geant4 Profiling and Benchmarking

Geant4 CPU Performance by Version

1) The Current profiling activity is a part of Geant4 Computing Performance Task

2) Profiling Results

Migration to Open|Sheepshop (will be the default profiler from 10.3)

Profiled on the Wilson cluster using AMD 6128HE Opteron 2GHz and Intel Xeon X5650 (2.67GHz)

Geant4 Version	Application	Performance		Summary	
10.2.r07 (OSS)	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r06 (OSS)	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM

*: recompiled & rerun at the date
Profiled on the Wilson cluster using AMD 6128HE Opteron 2GHz & gcc 4.9.2

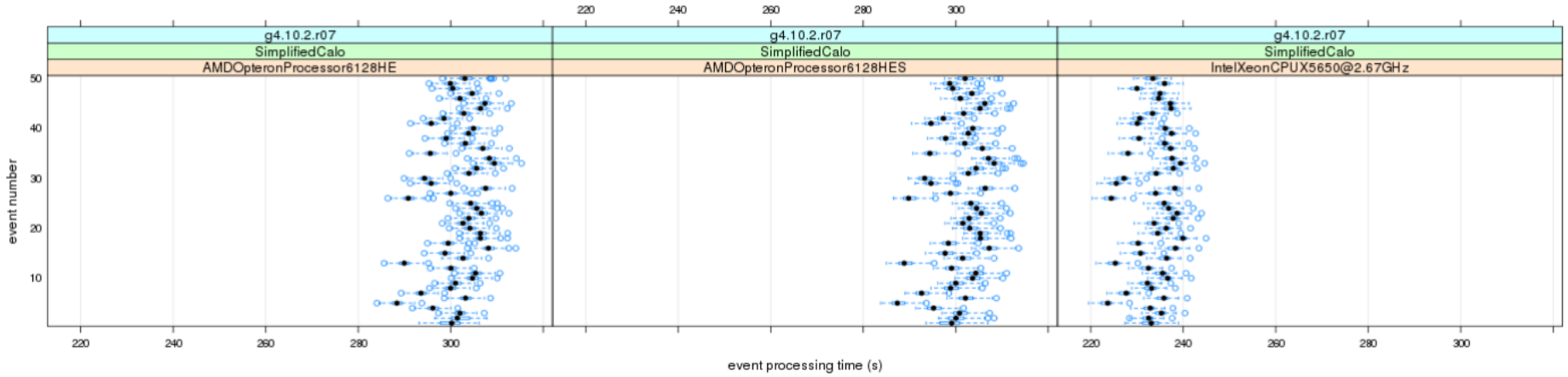
Geant4 Version	Application	Performance		Summary	
10.2.r07	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r06 (rerun, 08/19)	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r06 (10.3-BETA)	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r05	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r04 (May 28)	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM
10.2.r04	SimplifiedCalo	Simple Profiler	Memory Profiler	CPU	MEM

Results: Average CPU Time

- Triple-cross-checks (2 sets x 128 AMD nodes, 1set x 48 Intel nodes) - example: H →ZZ (SimplifiedCalo)

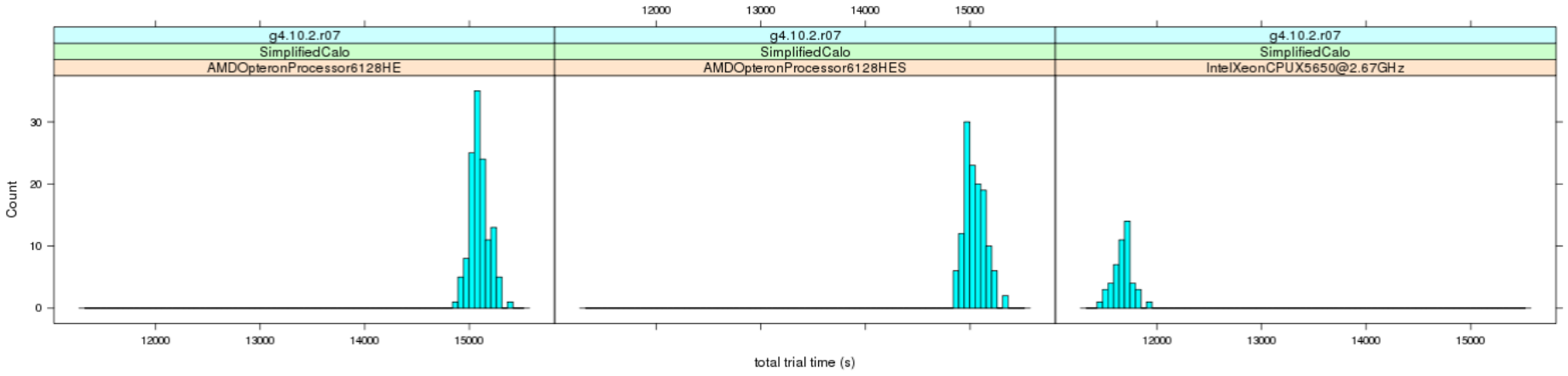
prof_100_events_plot.png:

Event Time



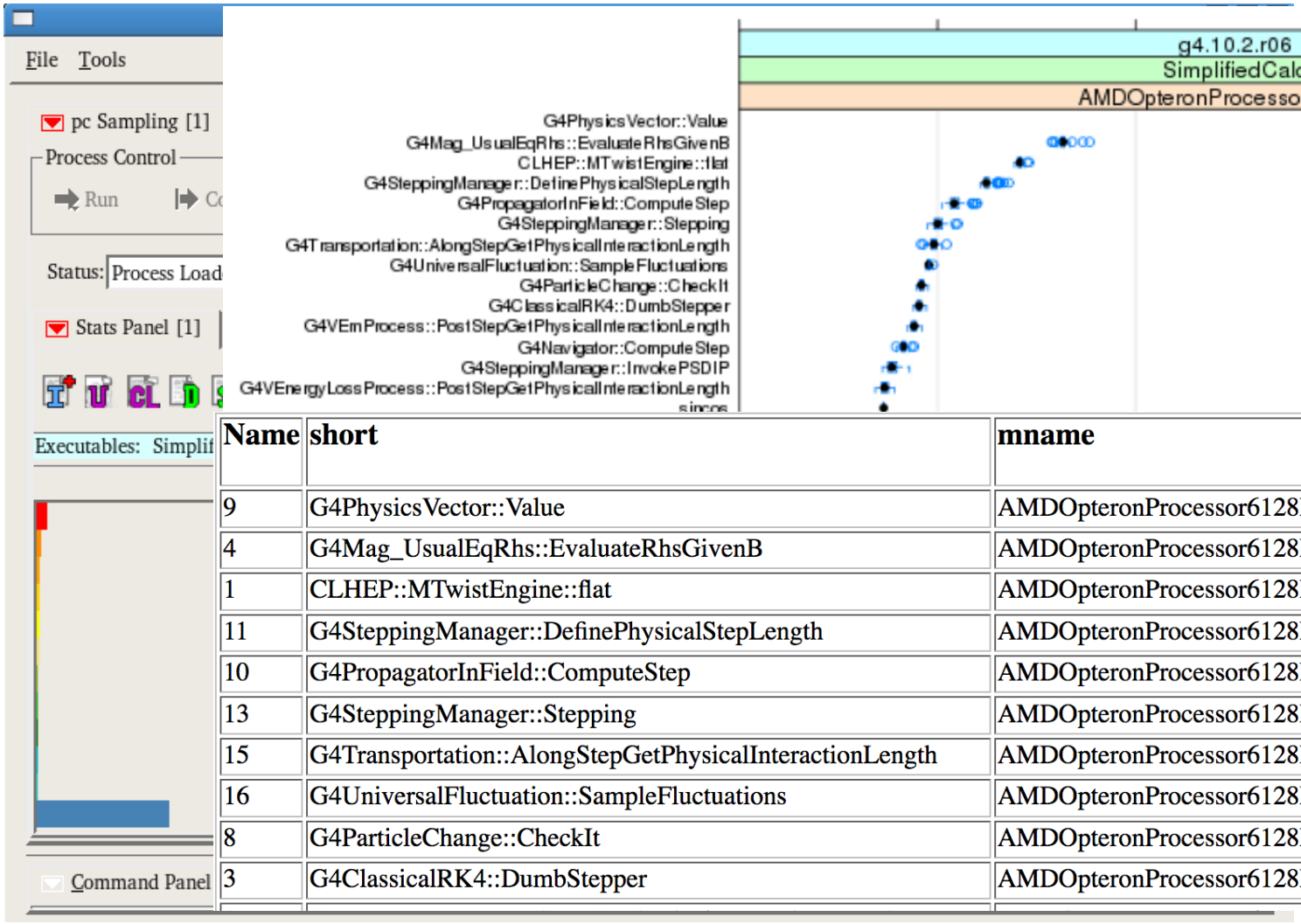
prof_basic_trial_times_histogram.png:

Total Time



OSS pcsamp (program counter sampling) Experiment

- OSS GUI example: Function view → g4cpt web plot/table



Name	short	mname	max.leaf.frac.median
9	G4PhysicsVector::Value	AMDOpteronProcessor6128HE	0.06152750
4	G4Mag_UsualEqRhs::EvaluateRhsGivenB	AMDOpteronProcessor6128HE	0.03266805
1	CLHEP::MTwistEngine::flat	AMDOpteronProcessor6128HE	0.02831920
11	G4SteppingManager::DefinePhysicalStepLength	AMDOpteronProcessor6128HE	0.02491585
10	G4PropagatorInField::ComputeStep	AMDOpteronProcessor6128HE	0.02167700
13	G4SteppingManager::Stepping	AMDOpteronProcessor6128HE	0.02000040
15	G4Transportation::AlongStepGetPhysicalInteractionLength	AMDOpteronProcessor6128HE	0.01962175
16	G4UniversalFluctuation::SampleFluctuations	AMDOpteronProcessor6128HE	0.01911510
8	G4ParticleChange::CheckIt	AMDOpteronProcessor6128HE	0.01836950
3	G4ClassicalRK4::DumbStepper	AMDOpteronProcessor6128HE	0.01812575

- Other views: linked object (libraries), function paths and so on

Comparison between FAST and OSS: Top Ten Functions

- Exclusive time fraction (10.2.r06, H→ZZ, SimplifiedCalo)

FAST (100 Hz) OSS (10 KHz)

short	.frac.median	l.frac.median	short
G4PhysicsVector::Value	0.06152750	0.061318050	G4PhysicsVector::Value
G4Mag_UsualEqRhs::EvaluateRhsGivenB	0.03266805	0.032771450	G4Mag_UsualEqRhs::EvaluateRhsGivenB
CLHEP::MTwistEngine::flat	0.02831920	0.027021450	CLHEP::MTwistEngine::flat
G4SteppingManager::DefinePhysicalStepLengtl	0.02491585	0.025940300	G4SteppingManager::DefinePhysicalStepLength
G4PropagatorInField::ComputeStep	0.02167700	0.024555300	G4PropagatorInField::ComputeStep
G4SteppingManager::Stepping	0.02000040	0.021670850	G4Transportation::AlongStepGetPhysicalInteract
G4Transportation::AlongStepGetPhysicalInterac	0.01962175	0.019675500	G4Navigator::ComputeStep
G4UniversalFluctuation::SampleFluctuations	0.01911510	0.018696050	G4UniversalFluctuation::SampleFluctuations
G4ParticleChange::CheckIt	0.01836950	0.018662900	G4ClassicalRK4::DumbStepper
G4ClassicalRK4::DumbStepper	0.01812575	0.018078150	G4SteppingManager::Stepping

- Also consistent profiling results for exclusive timing (call paths) and linked objects

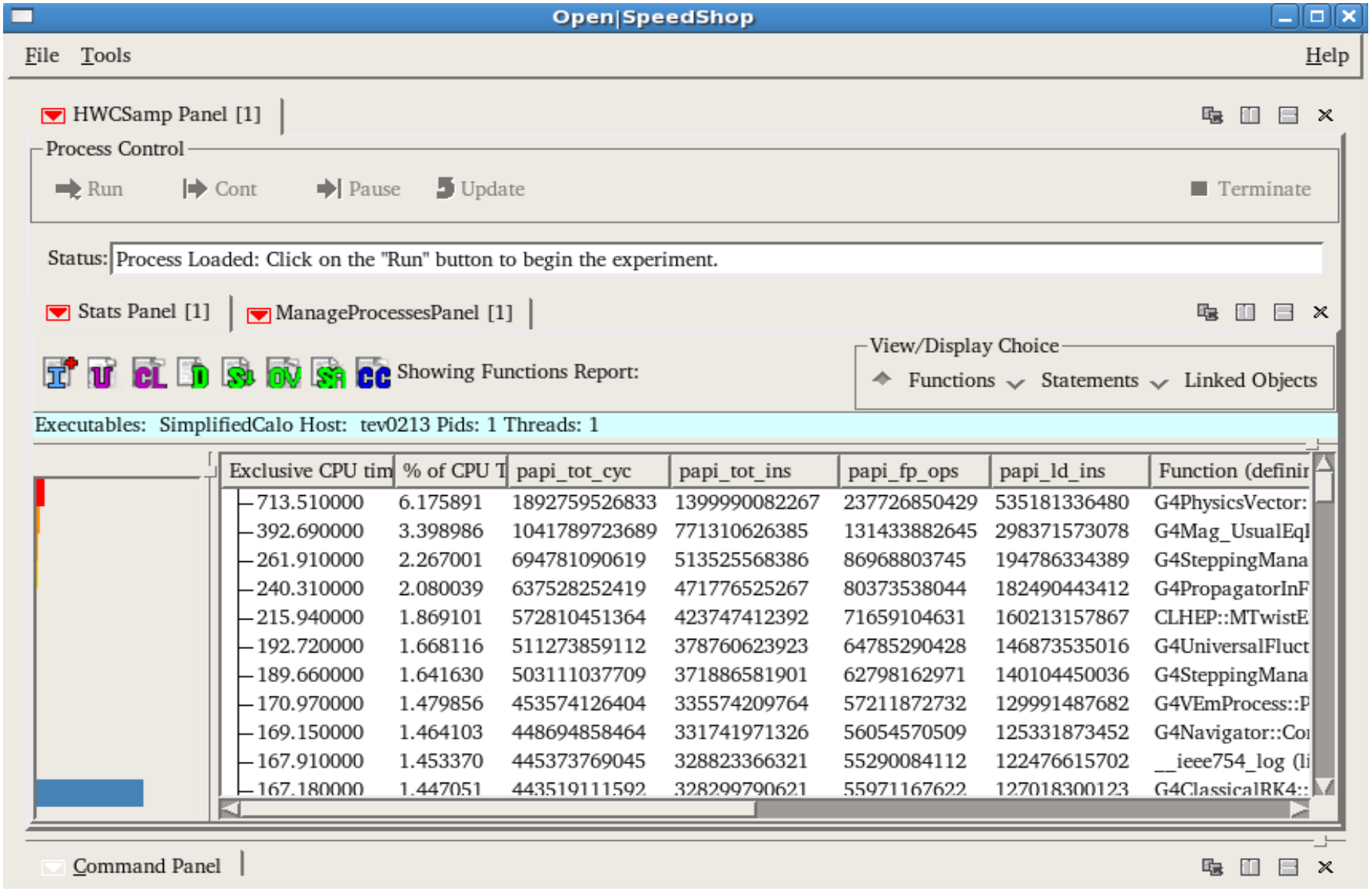
OpenISpeedshop: Experiments with Hardware Counters

- Periodic sampling hardware counters (hwcsamp)
- Supports both derived and non-derived PAPI presets
- A list of some possible hardware counter combinations

For **Xeon** processors:	
PAPI_FP_INS, PAPI_LD_INS, PAPI_SR_INS	Load store info, memory bandwidth needs
PAPI_L1_DCM, PAPI_L1_TCA	L1 cache hit/miss ratios
PAPI_L2_DCM, PAPI_L2_TCA	L2 cache hit/miss ratios
LAST_LEVEL_CACHE_MISSES, LAST_LEVEL_CACHE_REFERENCES	L3 cache info
MEM_UNCORE_RETIRED:REMOTE_DRAM, MEM_UNCORE_RETIRED:LOCAL_DRAM	Local/nonlocal memory access
For **Opteron** processors:	
PAPI_FAD_INS, PAPI_FML_INS	Floating point add multiply
PAPI_FDV_INS, PAPI_FSQ_INS	Square root and divisions
PAPI_FP_OPS, PAPI_VEC_INS	Floating point and vector instructions
READ_REQUEST_TO_L3_CACHE:ALL_CORES, L3_CACHE_MISSES:ALL_CORES	L3 cache

OSS hwcsamp (hardware counters) Experiment

- hwcsamp example: TOT_CYS, TOT_INC, FP_OPS, LD_INS (10.2.r06, H→ZZ, SimplifiedCalo)



Code Performance by Hardware Counter Metrics

- Derivatives: examples

Hardware Counter Metrics Derivatives	Performance
IPC (Instruction/Cycle)	Large values suggest good balance with minimal stalls.
FPC (FLOPS/Cycle)	Large values for floating point intensive codes suggests efficient CPU utilization
FMO (FLOPS/Memory Ops)	Good data locality, Computational Intensity
LPC (Loads/Cycle)	Useful for calculating FMO, may indicate good stride through arrays.
SPC (Stores/Cycle)	Useful for calculating FMO, may indicate good stride through arrays.

- Example: Higgs → ZZ (10.2.r06) with SimplifiedCalo
  - IPC = 0.74 (relatively small)
  - FMO = 0.30

More OSS Post Analysis

- Profiling information at the function statement (line) level
  - Will provide the top-100 list from 10.3
- Multi-threading capability (see the next talk)
- Compare two experiments (osscompare): examples to use
  - two releases
  - Two experiments with the different numbers of threads
- Call path analysis based on DB

Work Plan

- Migration to OpenIspeedshop (partially done)
  - Have been tested with recent releases since 10.3.beta
  - Reproduced all performance data that we used to measure
  - More information are ready to be added
- Current set of experiments
  - pcsamp (exclusive time)
  - usersamp (inclusive time and call path)
  - hwcsamp (PAPI hardware counters)
- Extension
  - mem (memory tracing)
  - pthreads (POSIX thread tracing)
  - io (I/O tracking)

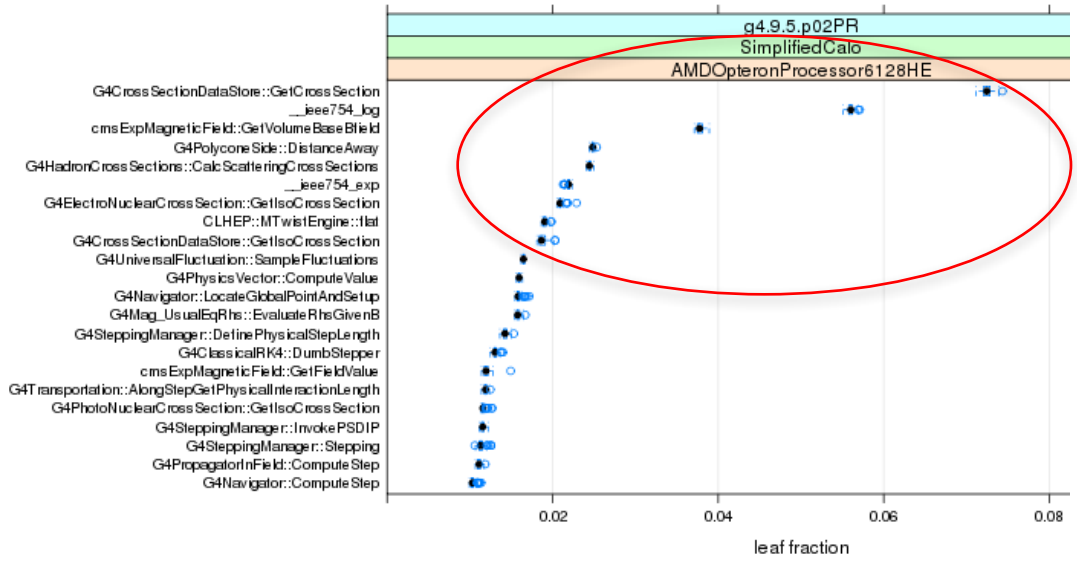
Summary

- Reviewed Geant4 computing performance for recent releases
- Challenges and new requirement in G4CPT have been addressed
- OpenIspeedshop is selected as the default profiler for the future computing performance task
- Migration is done and computing performance profiling and benchmark are ready to be extended

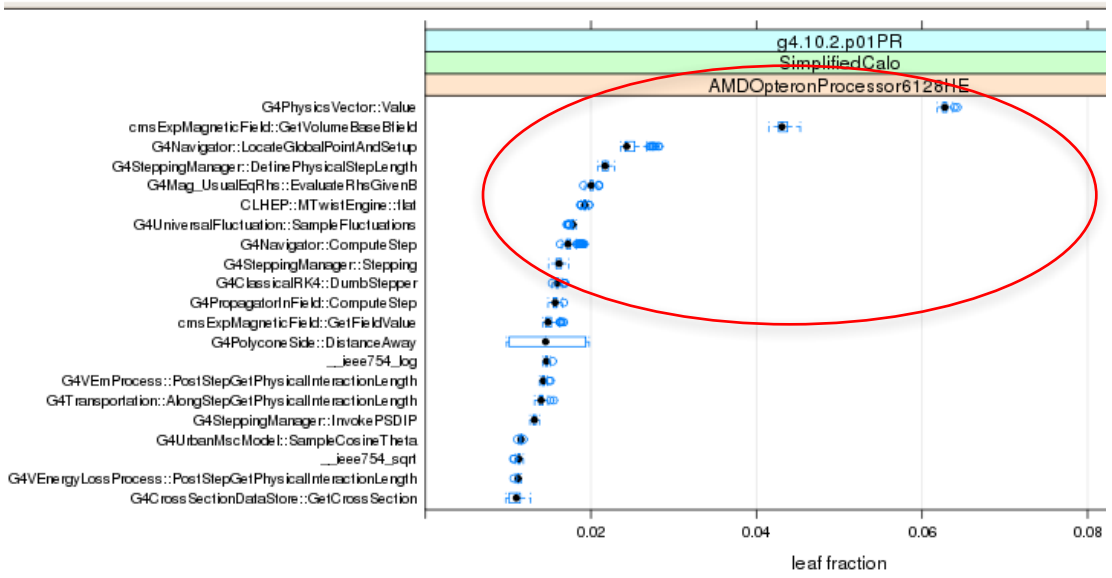
Backup Slides

Geant4 Computing Performance: Hot Functions

- Changes in hot functions from 9.5 to 10.2



G4CrossSectionDataStore::GetCrossSection
__ieee754_log
cmsExpMagneticField::GetVolumeBaseBfield
G4PolyconeSide::DistanceAway
G4HadronCrossSections::CalcScatteringCrossSections
__ieee754_exp
G4ElectroNuclearCrossSection::GetIsoCrossSection
CLHEP::MTwistEngine::flat
G4CrossSectionDataStore::GetIsoCrossSection
G4UniversalFluctuation::SampleFluctuations



G4PhysicsVector::Value
cmsExpMagneticField::GetVolumeBaseBfield
G4Navigator::LocateGlobalPointAndSetup
G4SteppingManager::DefinePhysicalStepLength
G4Mag_UsualEqRhs::EvaluateRhsGivenB
CLHEP::MTwistEngine::flat
G4UniversalFluctuation::SampleFluctuations
G4Navigator::ComputeStep
G4SteppingManager::Stepping
G4ClassicalRK4::DumbStepper

Results: OSS linked Objects

- OSS GUI example: Linked Objects view → g4cpt web plot