Session 3B: Computing Performance

(co-chaired by Gunter Folger and Soon Yung Jun)

CMS: Remark on computing performance	*IVANTCHENKO, Vladimir*		
ATLAS: remarks on computing performance	*APOSTOLAKIS, John*		
Reduction of runtime memory in Geant4 using compressed sensing	*MADSEN, Jonathan*		
Performance profiling and benchmark for medical applications	*CHO, Kihyeon*	*Aula Magna, Ferrara*	14:40 - 15:00
G4CPT with Open|Speedshop	*JUN, Soon Yung*	*Aula Magna, Ferrara*	15:00 - 15:15
Geant4 MT Performance	*JUN, Soon Yung*	*Aula Magna, Ferrara*	15:15 - 15:30

CMS: Remark on computing performance

V. Ivanchenko, CERN & Geant4 Associates International

ATLAS Geant4 Performance

Geant4 Collaboration Week
September 2016

Elmar Ritsch (CERN)
for the ATLAS Simulation Team

Presented by John Apostolakis

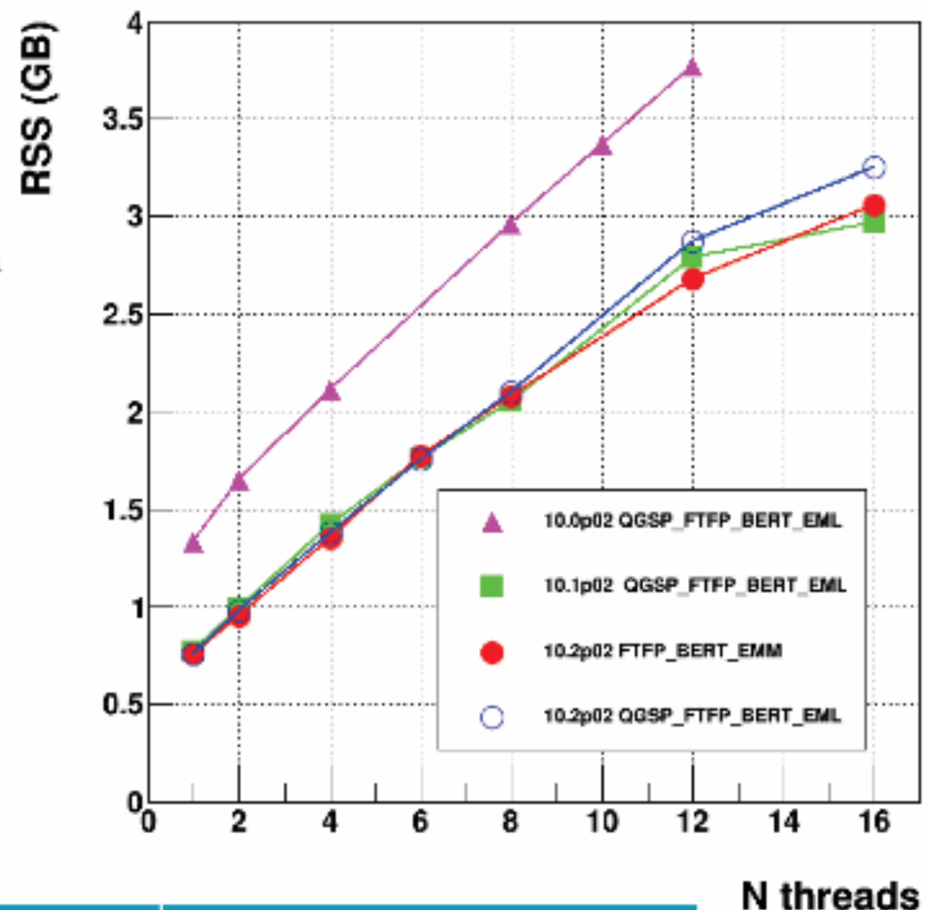
Memory for Geant4 10.2p02

Current development versions of Geant4 in CMSSW

Default physics list: FTFP_BERT_EMM

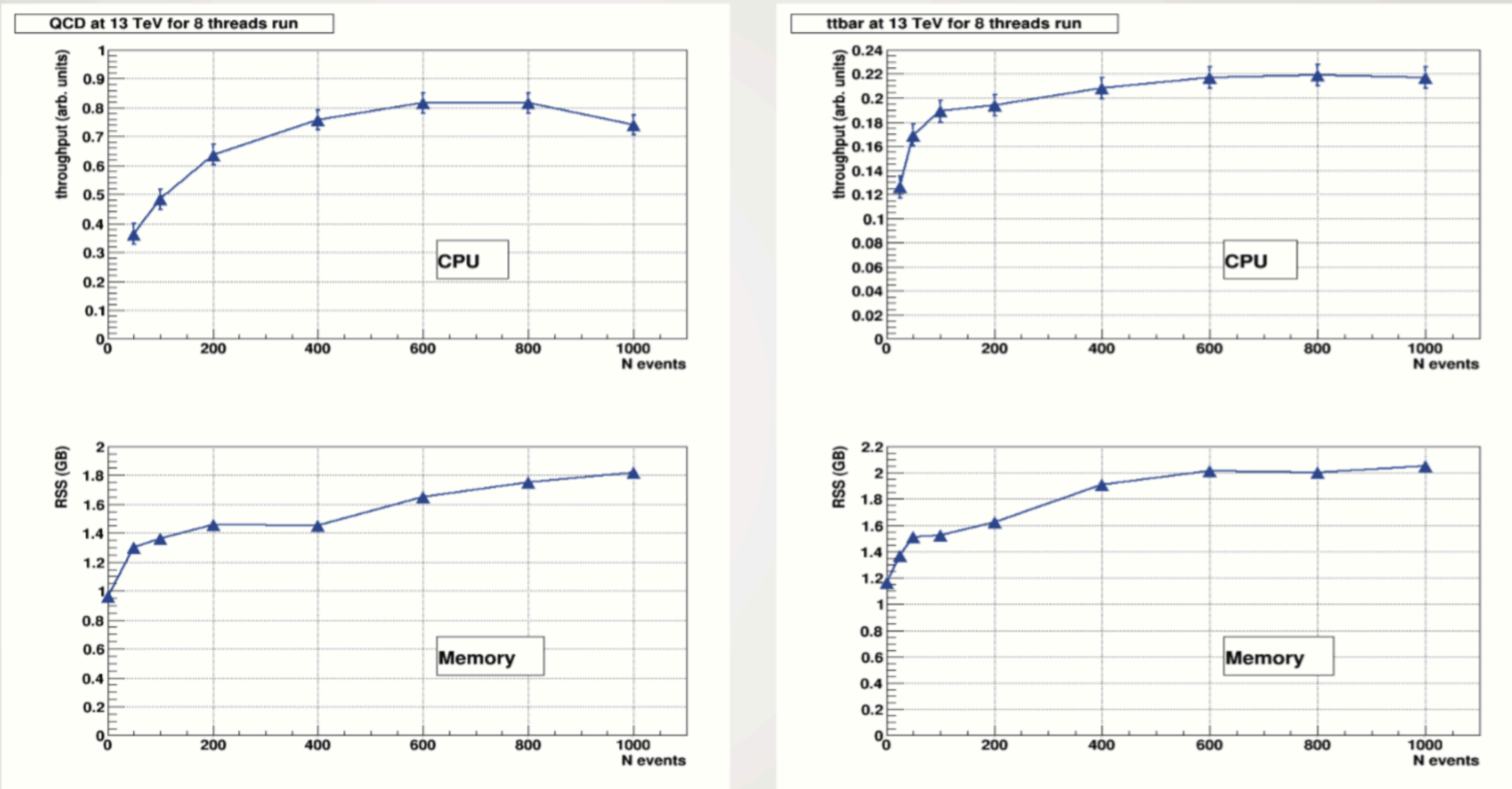
- A node with 12 Intel cores was used to study memory utilisation
- 13 TeV hard scattering event were simulated
  - Results after 1000 events are shown
  - CMS private patches to 10.0 include backports of fixes of memory leak and memory optimisation
  - Results for 10.1 and 10.2 are practically the same
  - No dependency on Physics List
- No problems to run CMS SIM production in the MT mode

Memory for ttbar events



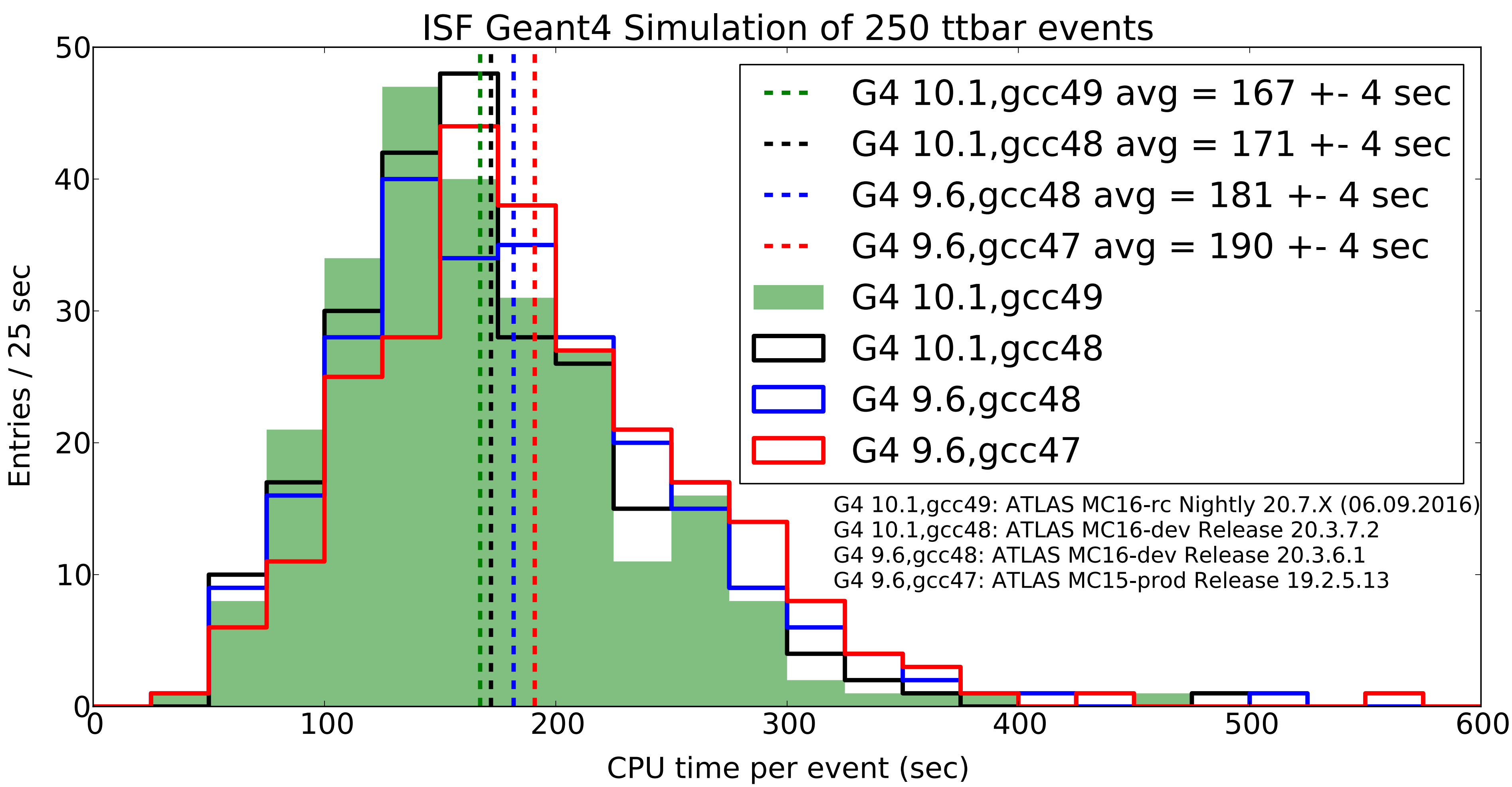
Release	1st thread (GB)	Delta per thread (GB)
10.0p02+CMS patches	1.33	0.23
10.2p02	0.76	0.19

Dynamic of CPU and RSS for 13 TeV CMS simulation run in MT mode with 8 threads, Geant4 10.2p02



- Dynamic of effective CPU and RSS memory are similar for events of different type
- Maxmum CPU efficiency is achieved after simulation ~500 events
  - The shape be explain by influence of Geant4 initialisation before 1st event and lazy initialization of hadron physics classes

ATLAS Event Time Measurement



Impressive 15% speedup between ATLAS MC15 and MC16, attributed to: 9.6→10.1, gcc47→gcc49 and code cleanup

CPU Profile Findings: G4

- **No crucial G4 hotspots**, neither in our MC15 (G4 9.6, gcc47) nor MC16 (G4 10.1, gcc49) simulation ☺
- **Small G4 hotspot disappeared** from G4 9.6 to 10.1
  - `G4NavigationHistory`: call to `std::__find` removed ☺ (~1.5% job time)

Reduction of Runtime Memory in Geant4 Using Compressed Sensing

Jonathan R. Madsen

Department of Nuclear Engineering
Texas A&M University
College Station, TX, USA 77843

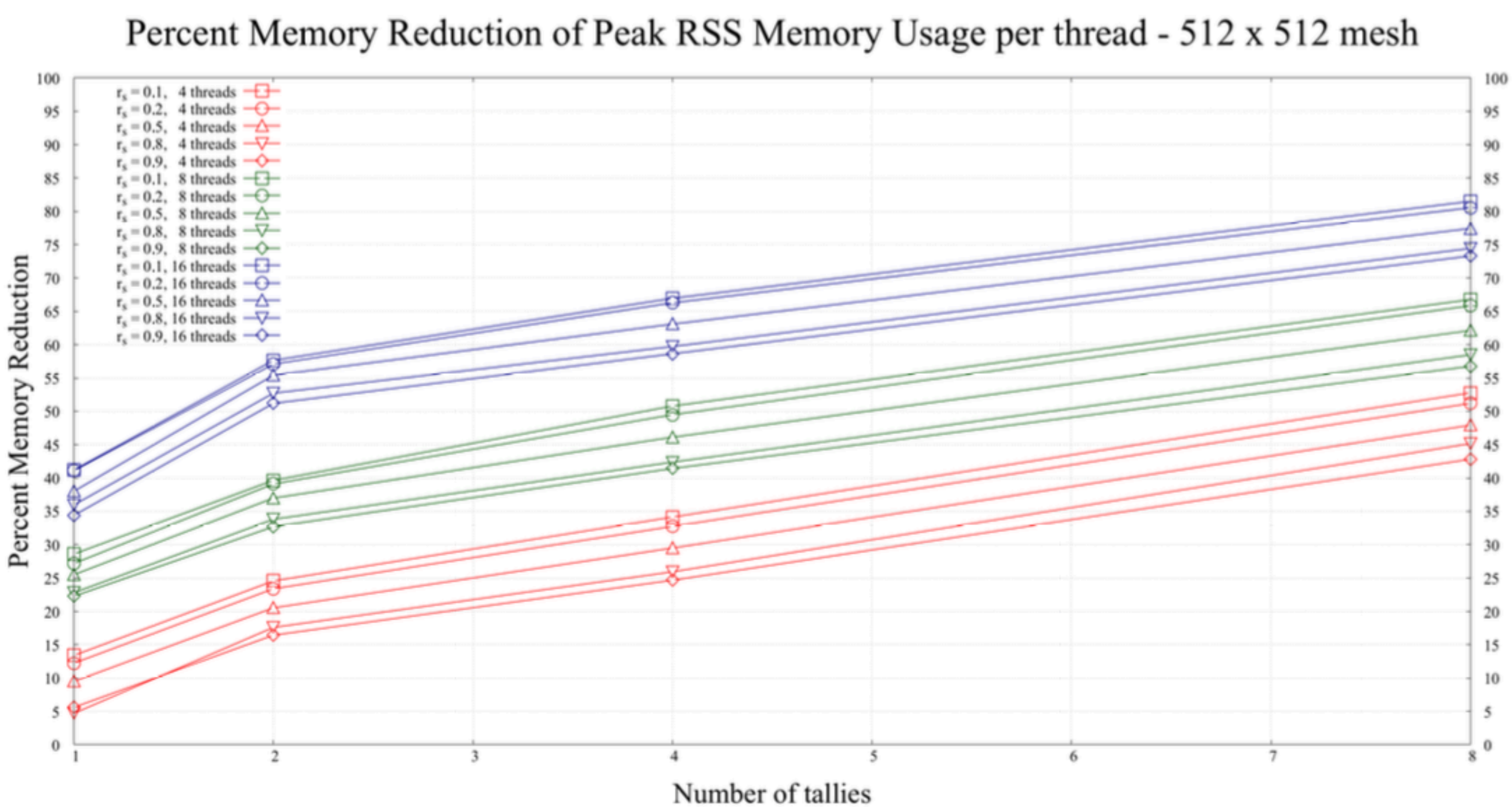
jonathanrmadsen@tamu.edu

NUCLEAR ENGINEERING
TEXAS A&M UNIVERSITY

MCNP6 Calculation of Thermal Flux of TRIGA reactor

- Compressed sensing used is used
  - to reduce number of primaries to be tracked
  - to reduce memory usage when using many threads or high number of quantities scored by using lossy compression
- Reconstruction from compressed format preserves peaks

Memory Reduction Results per thread



Performance Profiling and Benchmark for Medical Physics

Kihyeon Cho and Wonqook Choi (KISTI)

Computing Performance Task with OpenISpeedshop

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

MT Performance

Soon Yung Jun (Fermilab)

Profiling and Benchmark for Brachytherapy

- Goal
  - Extending computing performance profiling for low energy physics and other user cases (other than HEP applications)
  - Profile and benchmark CPU/memory usage and other performance metrics
- Status
  - Successfully adopted the FNAL protocol and modified for the KISTI platform/environment
  - Profiled CPU/memory with advanced/brachytherapy example and produce other monitoring data
  - Also study other dependencies (e.g. mesh size)
- Future work
  - Monitoring performance changes by each release
  - Scalability for new computing architecture (KISTI supercomputer)

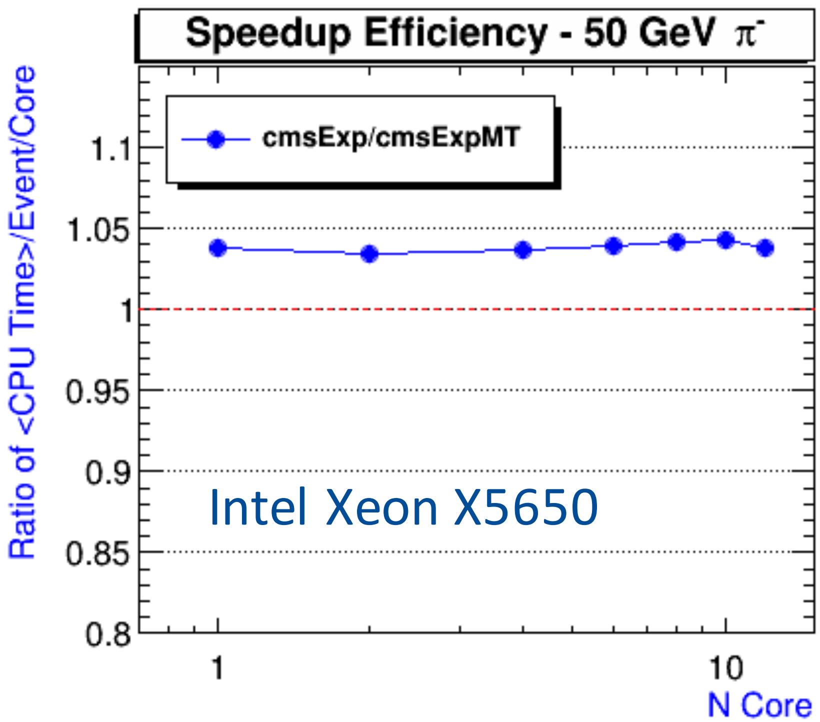
Geant4 Computing Performance Profiling and Benchmark

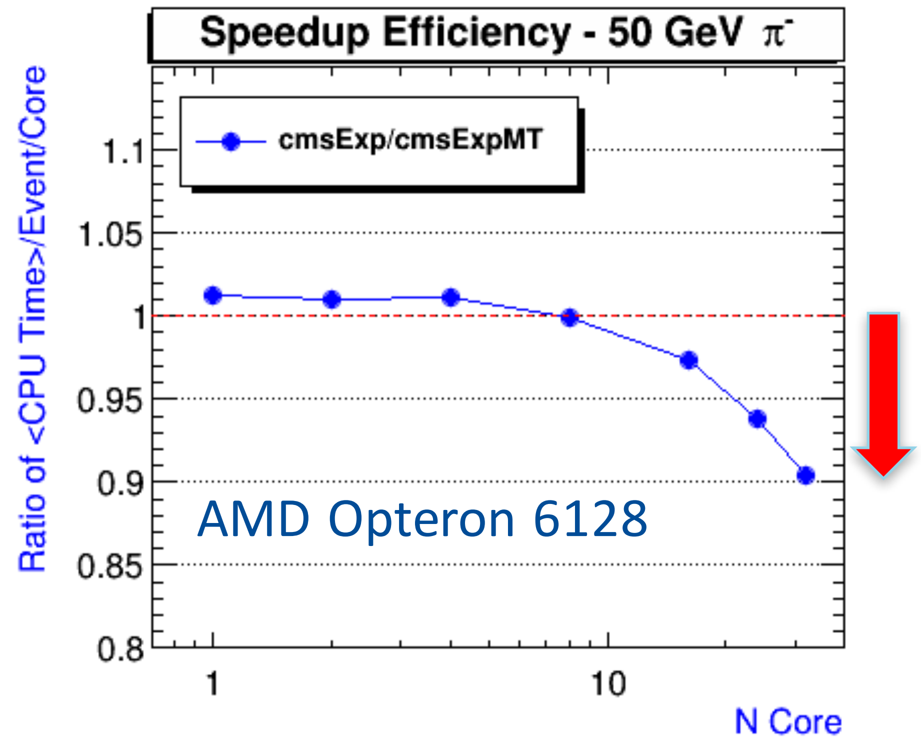
- Challenges and new requirements
  - Geant4 CPU profile is very flat (no hot spots): hard to trace small changes in performance
  - Reduce measurement uncertainties
  - Lack of extensibility of current tools for multi-threaded applications
- Adopt OpenISpeedshop (OSS) to meet new requirements
  - Light weight and easy to use
  - Support call stack analysis, memory tracing and hardware counter experiments
  - Multithreading-capable (POSIX thread tracing)
- Status
  - Migration is done and OSS has been tested since 10.3.beta
  - Will replace the current protocol from 10.3

MT Performance on General Purpose CPUs: Intel vs. AMD

- Event throughput = the number of event processed/time

Speedup efficiency: $\epsilon(Nthreads) = \frac{Throughput(Sequential)}{Throughput(Nthreads)} \times Nthreads$

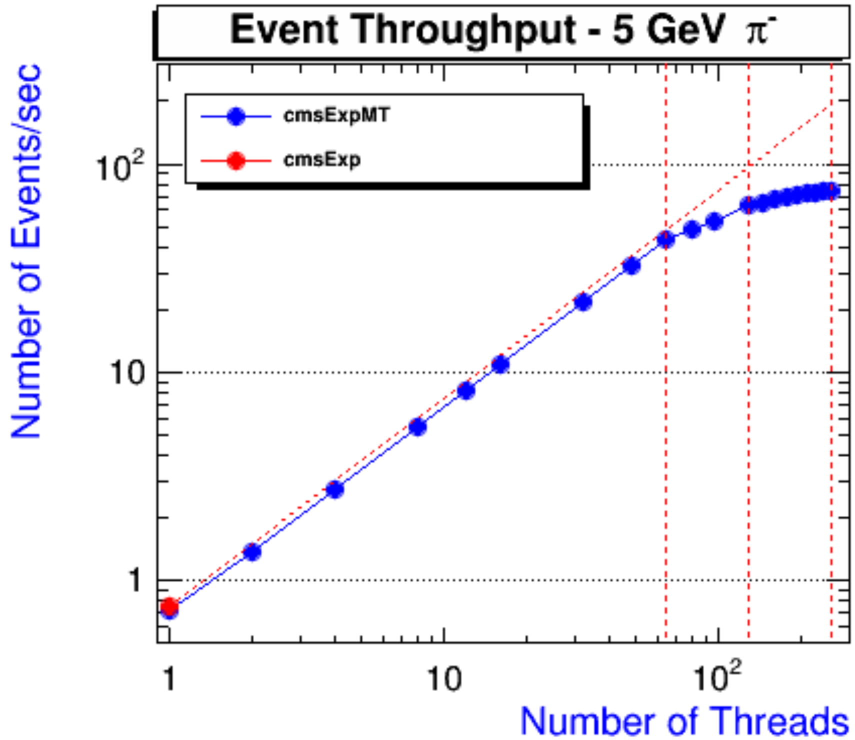


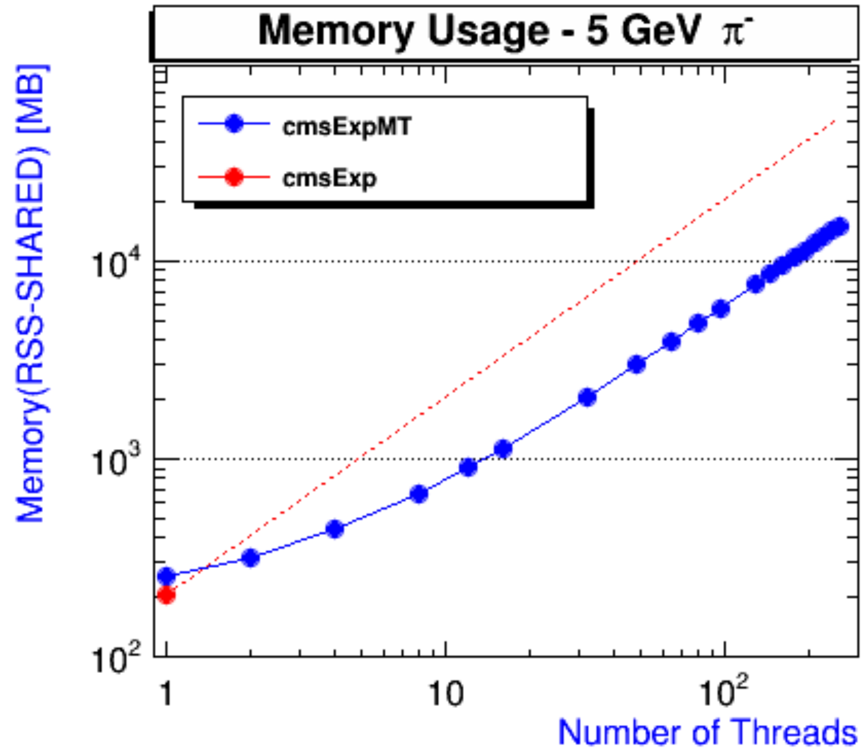


- MT application is being profiled with OpenISpeedshop
  – Degradation seen in AMD is under investigation (profiling data are being examined, but more cross-checks are underway)

Geant4 Multi-threading Performance on KNL

- Performance on Intel Xeon Phi Processor (Knight's Landing)
  - Geant4 10.2.p02 + standalone CMS simulation with -xMIC-AVX512 (N-threads x1028 events, N-threads up to 256)





  - Almost-linear scalability up to 64 cores and significant memory reduction as the number of threads increases