Parallelization improvements

MT and MPI

Andrea Dotti (adotti@slac.stanford.edu) ; SLAC/SD/EPP/Computing

Geant4 21st Collaboration Meeting – Ferrara, 12-16 September 2016

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

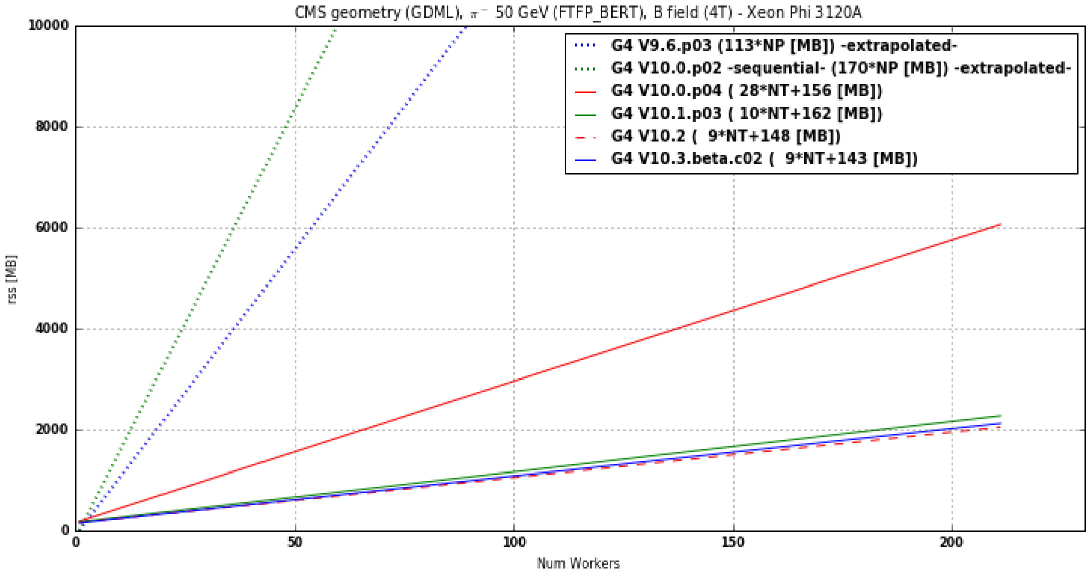
Parallelization options in G4

Multi-threading (shared memory model) and MPI (distributed memory model) are both mature and feature complete

MPI extension, depending on external library, is not part of the kernel, but is provided in the example category

- A library libG4mpi.so is created together with examples
- We could move it to the kernel (intercoms category) as an optional component (like GDML support for example), but it is not critical

MT Memory



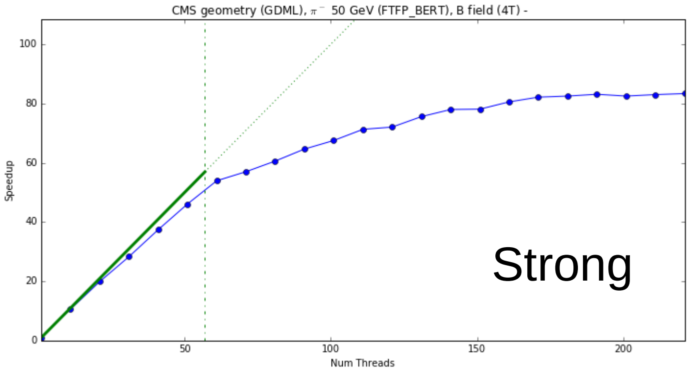
Memory well under control and better than our goal since 1.2

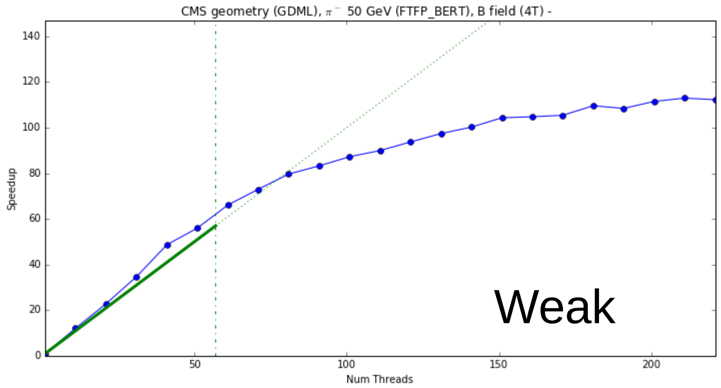
MT Scaling

We measure scaling in two ways: strong and weak

- Strong scaling: keep constant the total number of events measure the time between first and last event: $$S^{strong}_{threads} = \frac{\sum_{e \in events} T_{1,e}}{\max_{t \in threads} \sum_{e \in events} T_{t,e}} \leq 1$$
- Weak scaling: keep constant the number of events per thread, measure time for each event and: $$S^{weak}_{threads} = \frac{\sum_{e \in events} T_{1,e}/N_{1,e}}{\sum_{e \in events} T_t, e/N_{t,e}} \approx \frac{1}{t} \frac{\sum_{e \in events} T_{1,e}}{\sum_{e \in events} T_{t,e}}$$

It is possible that in weak scaling we get >1 slope

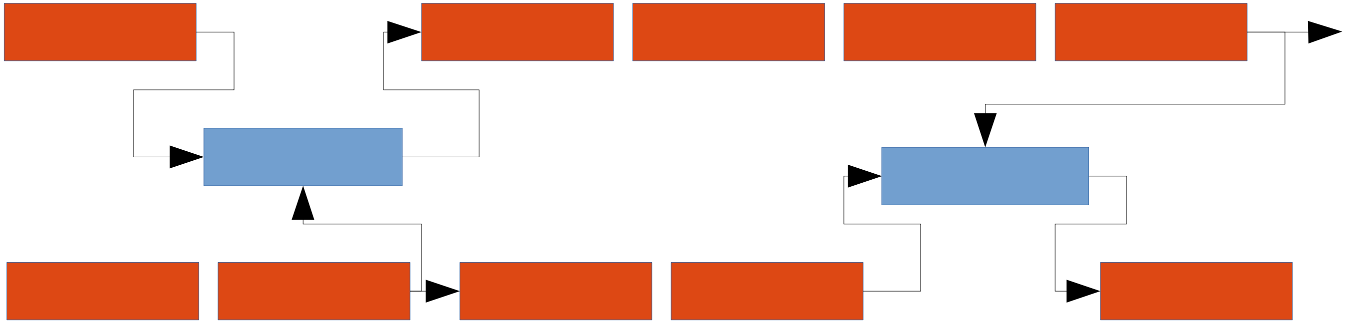




Explanation

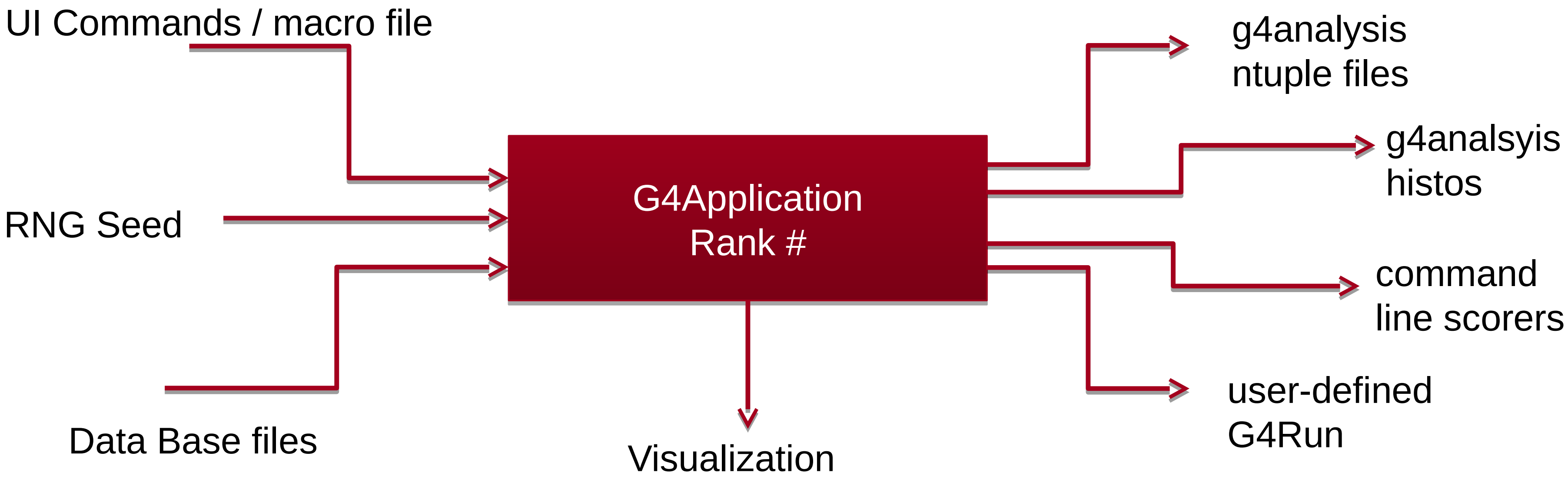
Sequential

MT

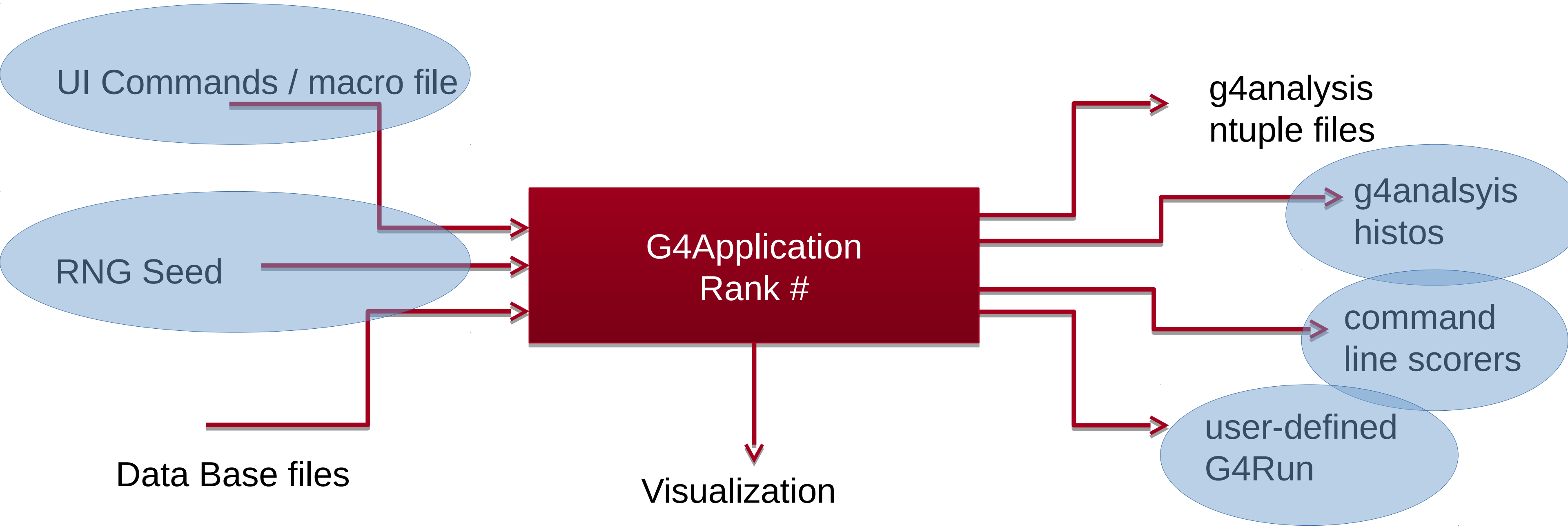


MT collaborative building data structures

Geant4 applications from MPI point of view



Geant4 applications from MPI point of view



At this meeting we will discuss ntuple output and start discussion about DB

Demonstrate speed-up w/ multi-threading and MPI

Performances measured with “Bertini cascade validation” application

Linearity of speedup well demonstrated up to large number of total workers

Measured on Tachyon 2 supercomputer at KISTI (South Korea)

- degradation of linearity for Nthreads>2000 is partially due to sub-optimal merging of histogram files via mpi (to be improved in 2016) and physical I/O

Number of nodes	Total number of threads	Speedup (strong scaling)	Speedup (strong-scaling) w.r.t. sequential
10	80	1	79
20	160	2	158
40	320	4	317
80	640	7.9	626
160	1280	15.8	1251
320	2560	29	2297
640	5120	45	3555

Section	Specs.
Model	SUN Blade 6275
Blade Nodes	3176 Compute Nodes, 300 TFlops (Rpeak)
CPU	Intel Xeon x5570 Nehalem 2.93GHz, 8 cores per node, Total 25408 cores
Memory	24 GB (per node)
Storage	1125 TB (Disk) 2112 TB (Tape)
Interconnect Network	Infiniband 4x QDR

Need to improve strategy for merging of results

Demonstrate speed-up w/ multi-threading and MPI

SLAC

Performances measured with "Bertini cascade validation" application

Linearity of speedup well demonstrated up to large number of total workers

Measured on Ta ... orea)

- degradatio ... to sub-optimal me ... d in 2016) and physical I/(

See discussion later on "G4 at extreme scales" for further notes on scalability

Number of nodes	Total number of threads	Speedup (strong scaling)	Speedup (strong-scaling) w.r.t. sequential
10	80	1	79
20	160	2	158
40	320	4	317
80	640	7.9	626
160	1280	15.8	1251
320	2560	29	2297
640	5120	45	3555

Section	Specs.
Model	SUN Blade 6275
Blade Nodes	3176 Compute Nodes, 300 TFlops (Rpeak)
CPU	Intel Xeon x5570 Nehalem 2.93GHz, 8 cores per node, Total 25408 cores
Memory	24 GB (per node)
Storage	1125 TB (Disk) 2112 TB (Tape)
Interconnect Network	Infiniband 4x QDR

Need to improve strategy for merging of results

Merging of results in MT/MPI

Results are merged into master thread in a MT application

If MPI is enabled master sends merged results to rank #0 for further reduction

- MPI ranks tend to finish ~ at the same time: bottleneck appears since ranks are serialized in communication with rank #0

New: binary tree merging of results to use in parallel network

- Not done yet for histogram merging

Binary Tree Mering

Binary Tree Mering

HepExpMT

An improved version of ParFullCMS has been created and made publicly available:

- Interest from users and even some companies
- To be used as a "public candle" for Geant4 performance measurement
- Some optional features (e.g. MPI) and I/O testing

To simplify application compilation a script is provided that:

1) Downloads G4
2) Configure G4 and Application
3) Compiles G4 and Application in a coherent environment

Check it out at: https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4HepExpMTBenchmark


HepExpMT brought to you by:
G4
Across the Bay

Workspace re-use and c++11 migration

Thread private code is now owned/managed by "workspaces" (one per relevant category)

- Possibility of re-using workspaces (if streams < #threads) is coded, but neither tested or debugged: is this still requested? Move to next year?

Thread starting and initialization has been further reviewed: more granular methods to ease integration in experimental frameworks

Migration from pthreads to c++11: missing manpower for 10.3

- Main use-case is support for Windows
- We have "volunteers" assigned to that, but we are very late!