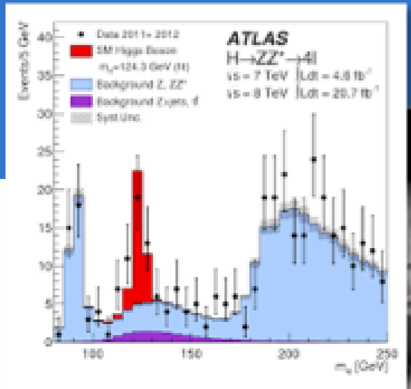
Code modernization: The GeantV project

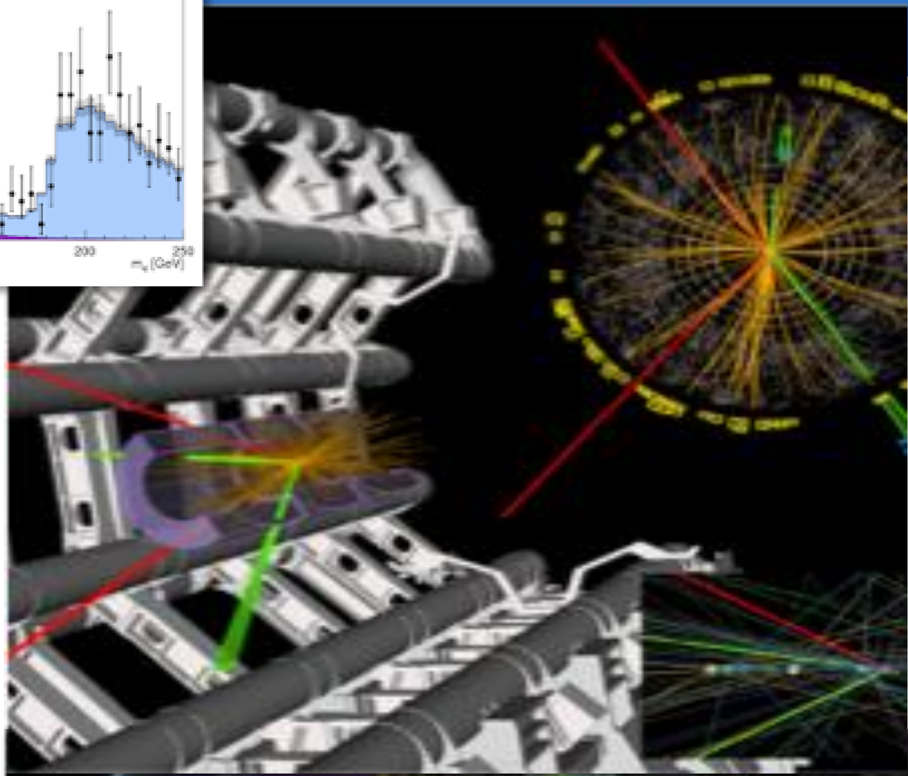
A. Gheata for the GeantV team

Outline

- Introduction
- Geometry benchmarks: vectorization and scalability
- Particle transport improvement
- Sub-node clustering
- Task based approach
- Locality (NUMA), Machine Learning, HPC workloads
- Plans

The problem





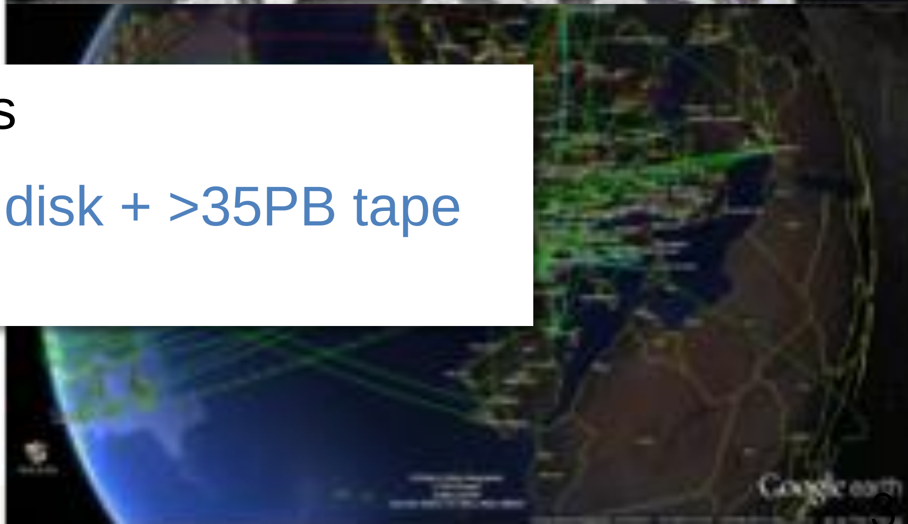
- Detailed simulation of subatomic particles in detectors, essential for data analysis, detector design..
- Complex physics and geometry modeling
- Heavy computation requirements, massively CPU-bound

WLCG
Worldwide LHC Computing Grid

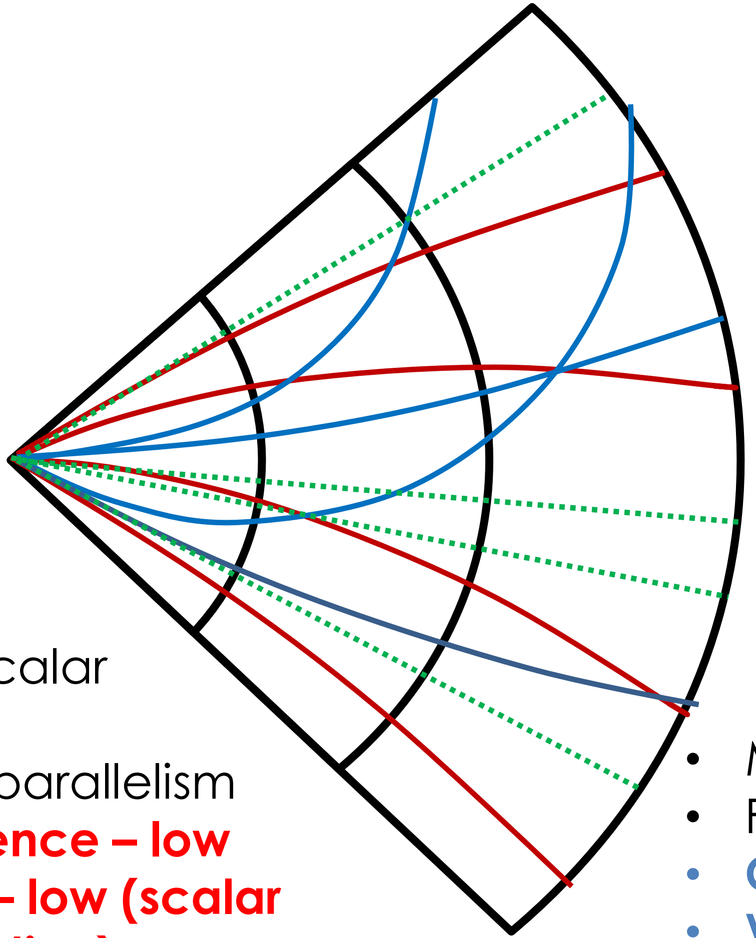
200 Computing centers in 20 countries: > 600k cores

@CERN (20% WLCG): 65k processor cores ; 30PB disk + >35PB tape storage

More than 50% of WLCG power for simulations

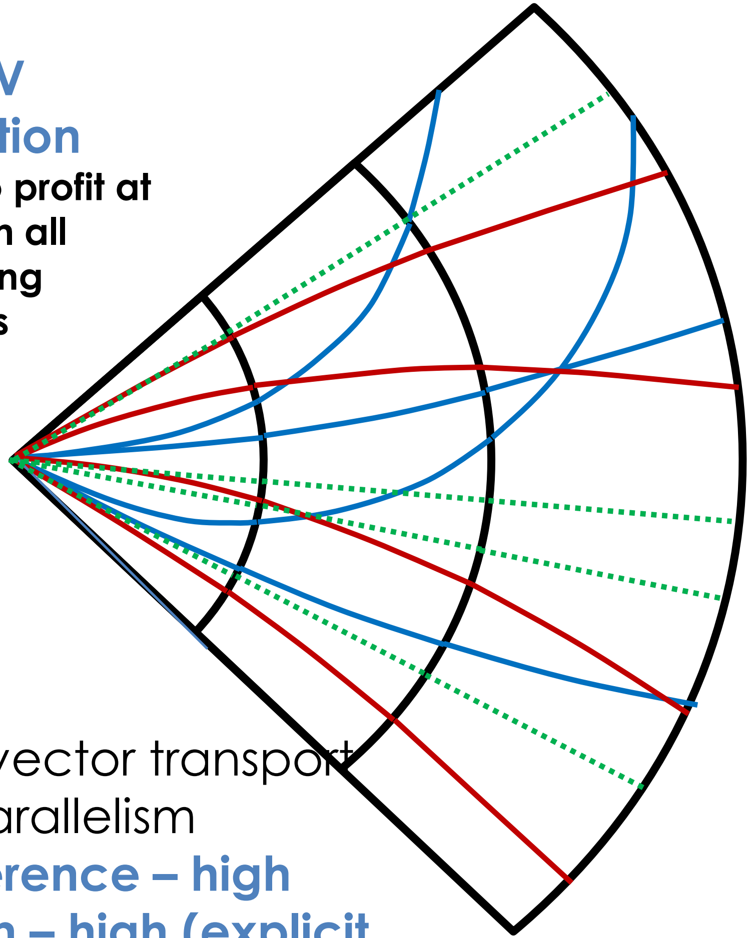


GeantV – Adapting simulation to modern hardware



Classical simulation
hard to approach the full machine potential

- Single event scalar transport
- Embarrassing parallelism
- **Cache coherence – low**
- **Vectorization – low (scalar auto-vectorization)**

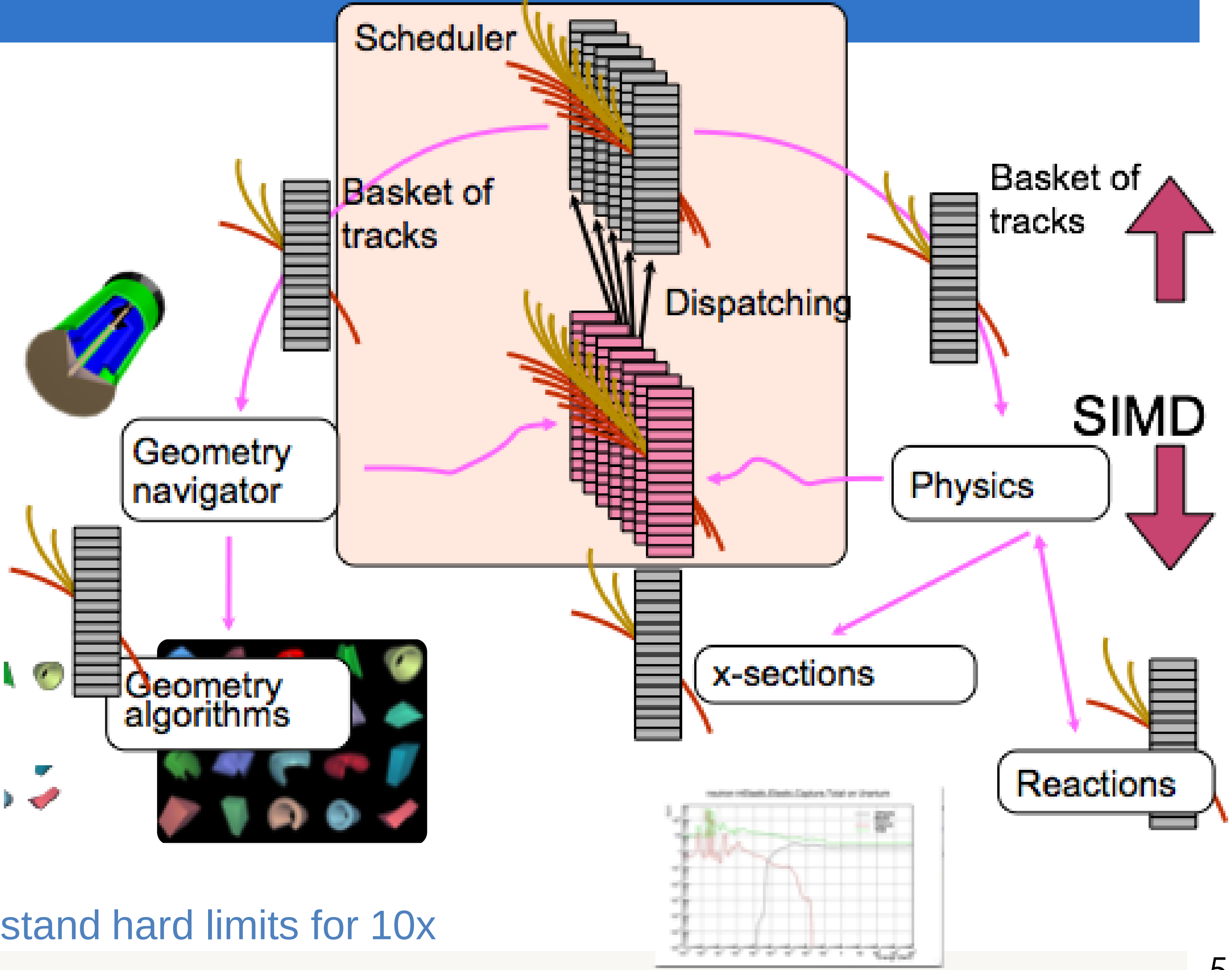


GeantV simulation
needs to profit at best from all processing pipelines

- Multi-event vector transport
- Fine grain parallelism
- **Cache coherence – high**
- **Vectorization – high (explicit multi-particle interfaces)**

GeantV approach

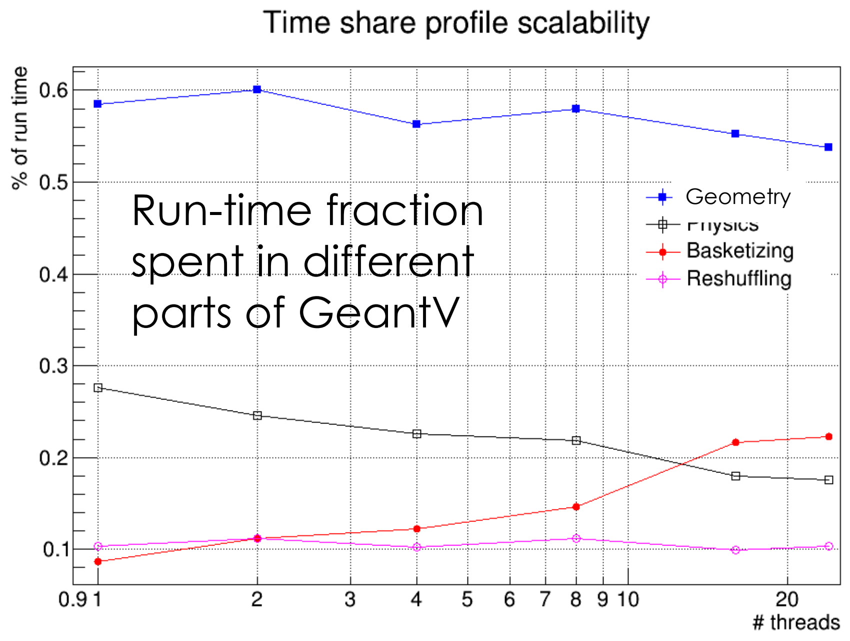
- Transport particles in vectors (“baskets”)
  - Filter by geometry volume or physics process
- Redesign library and workflow to target fine grain parallelism
- Use backends for portability and interface abstraction (vector, scalar)

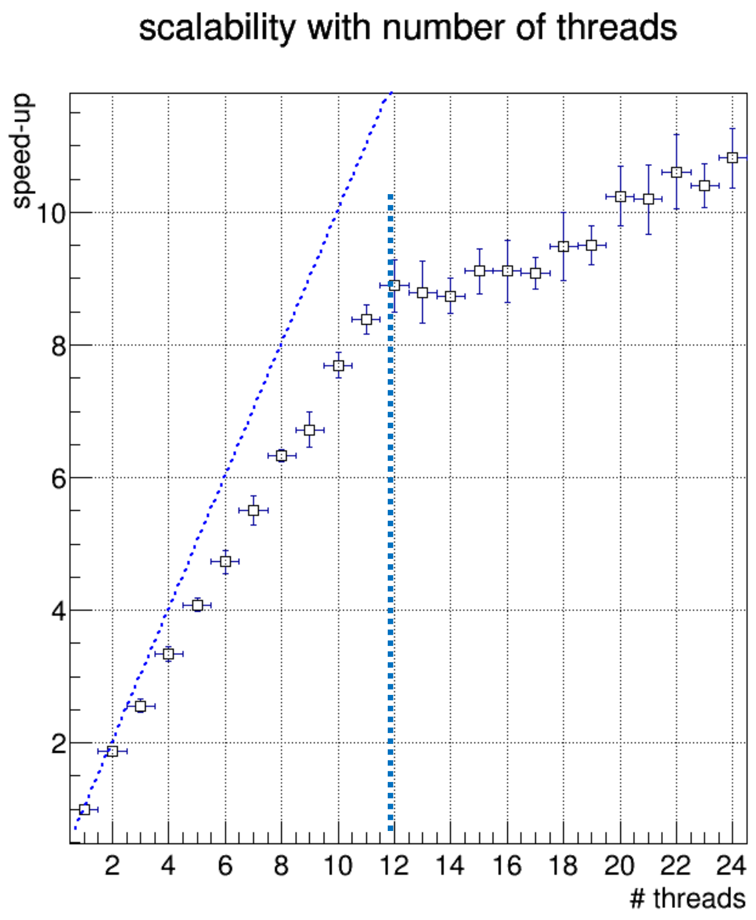


Aim for a 3x-5x faster code, understand hard limits for 10x

Challenges

- No free lunch: need to keep data gathering overheads < vector gains

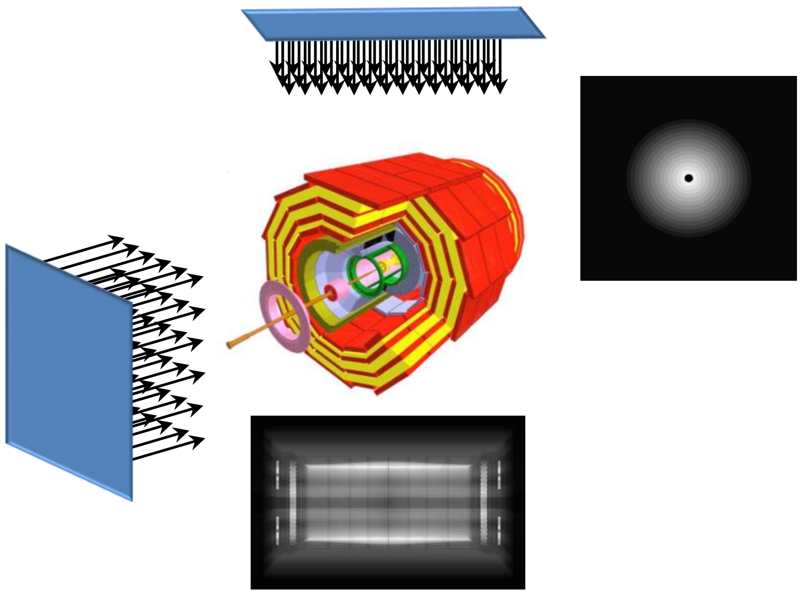


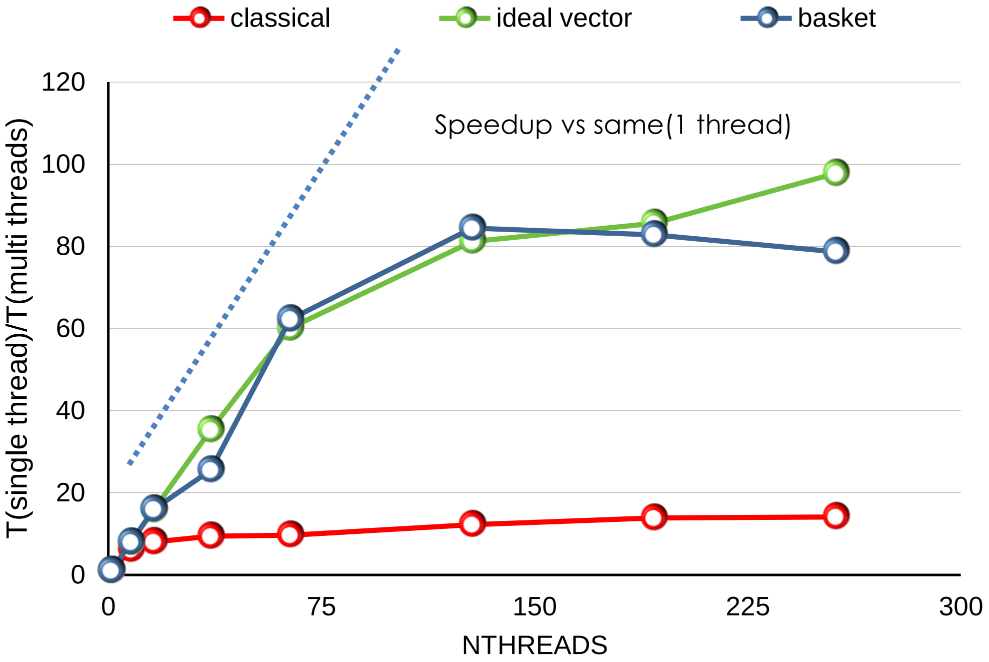


24-core dual socket E5-2695 v2 @ 2.40GHz (HSW).

Geometry navigation on KNL

- X-ray scan of detector volumes
  - Trace a grid of virtual rays through geometry
- Simplified geometry emulating a tracker detector
- Compare GeantV basket approach to
  - Classical scalar navigation (ROOT)
  - An ideal "vector" case (no basketizing overheads)
- AVX512 vectorization enforced by API (UME:SIMD backend)
- ~100x scalability for the ideal and basket versions

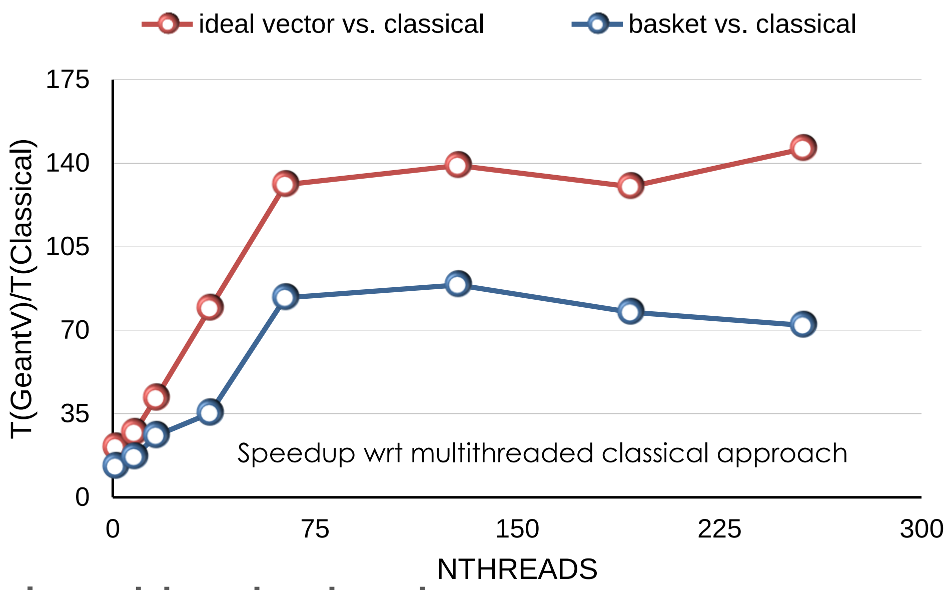




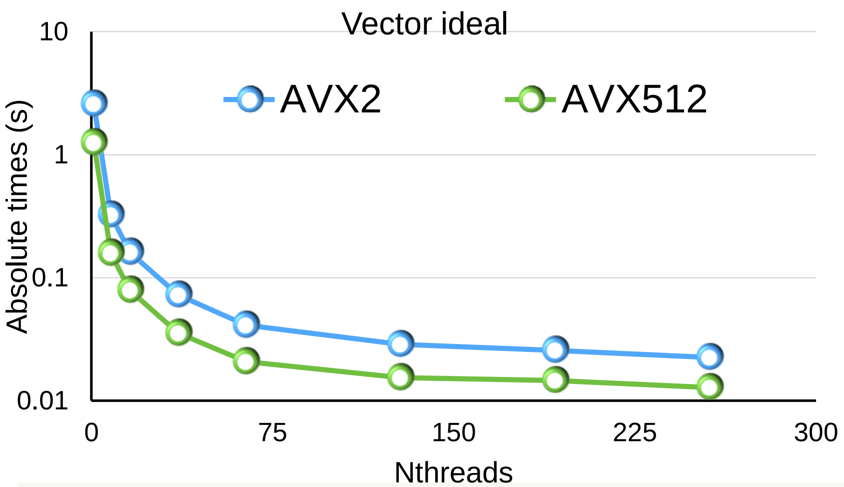
Intel® Xeon Phi™ CPU 7210 @ 1.30GHz

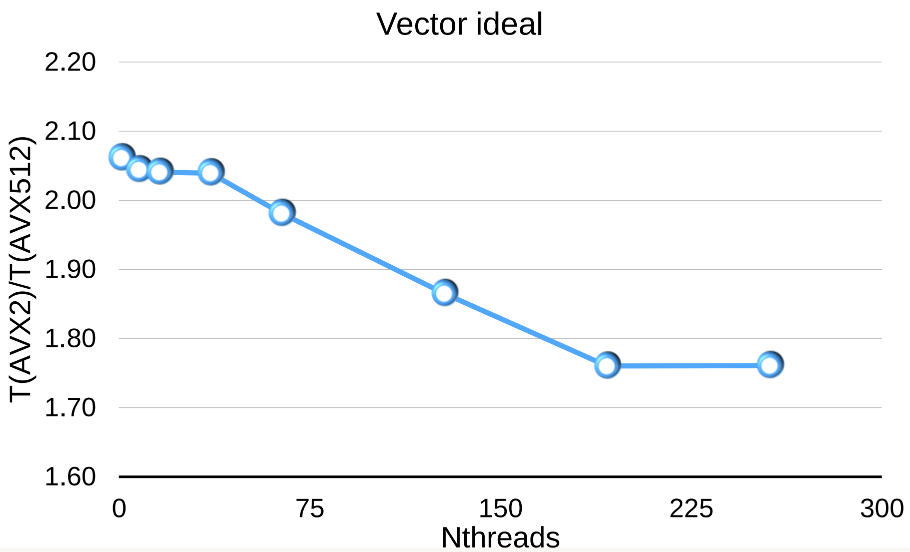
Performance

GeantV gives excellent benefits with respect to ROOT in terms of speedup



- High vectorization intensity achieved for both ideal and basketized cases
  - AVX-512 brings an extra factor of ~2 to our benchmark





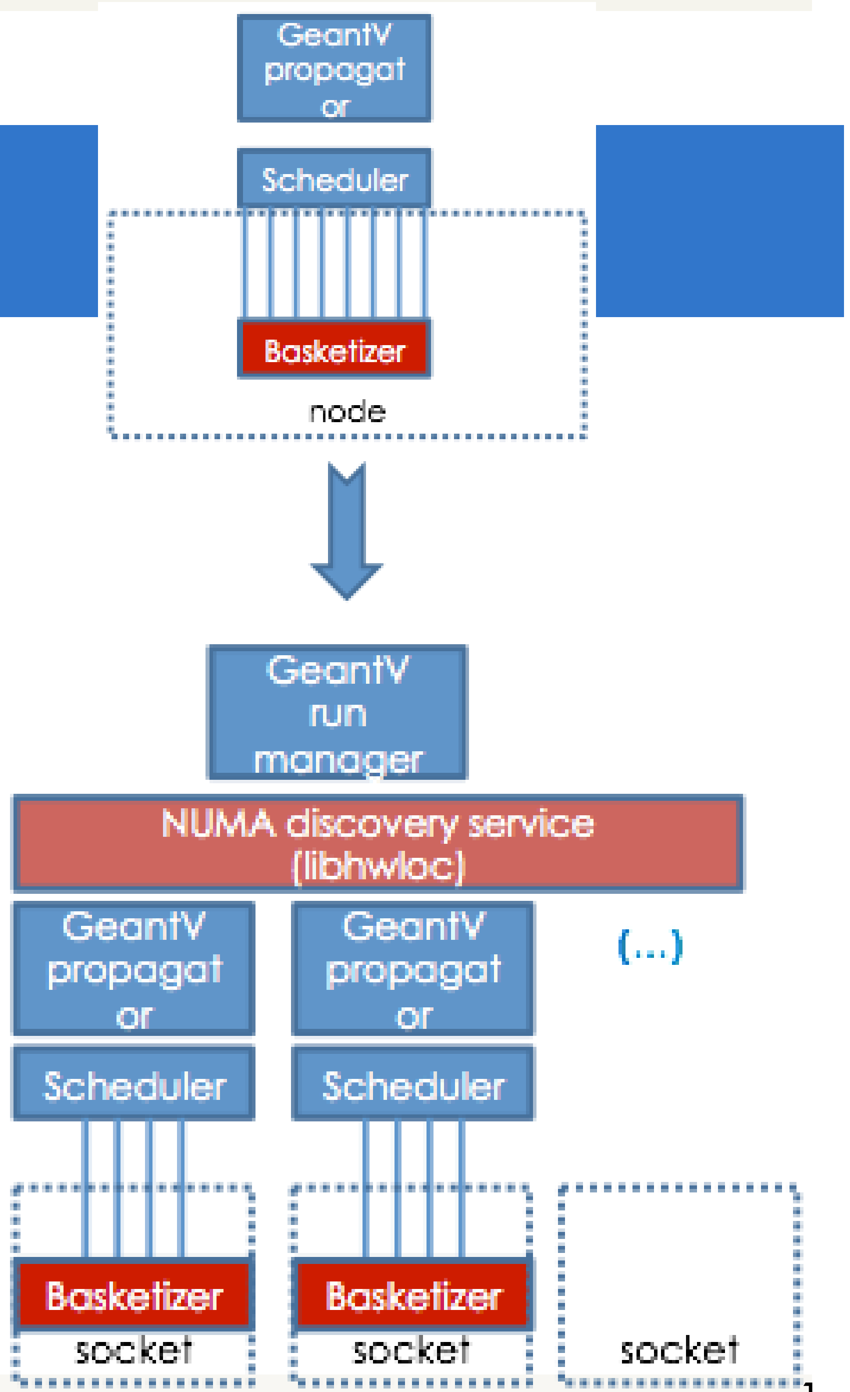
KNL R&D 2016

- Sub-node clustering with multiple propagators
  - Improve data/processing locality and reduce contention
- TBB-based task based version
- Full prototype on KNL (tabulated physics)
- Improved memory management in basketizing procedure

(NUMA awareness)

Sub-node clustering

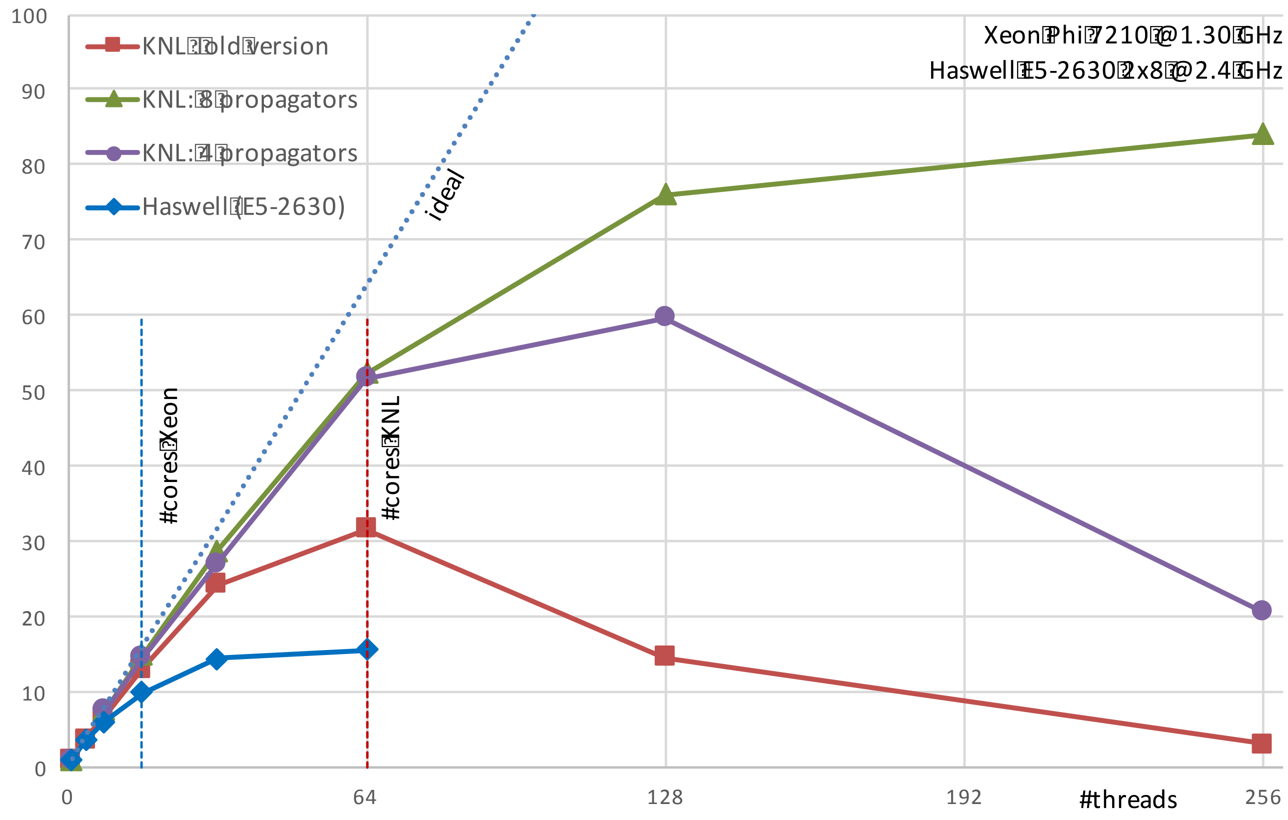
- Known scalability issues of full GeantV due to synchronization in re-basketizing
- New approach deploying several propagators clustering resources at sub-node level
- Objectives: improved scalability at the scale of KNL and beyond, address both many-node and multi-socket (HPC) modes + non-homogenous resources
- Implemented recently and tested on KNL



Multi-propagators prototype

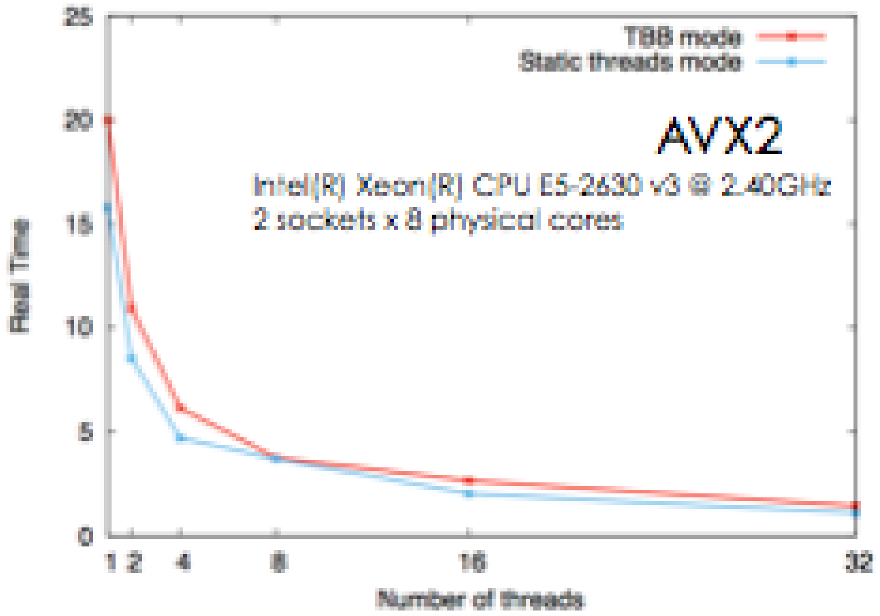
- Full track transport and basketization procedure
- Simplified calorimeter
- Tabulated physics (EM processes + various materials)
- Scalability gets better by increasing number of propagators
- The seed for GeantV core version 3

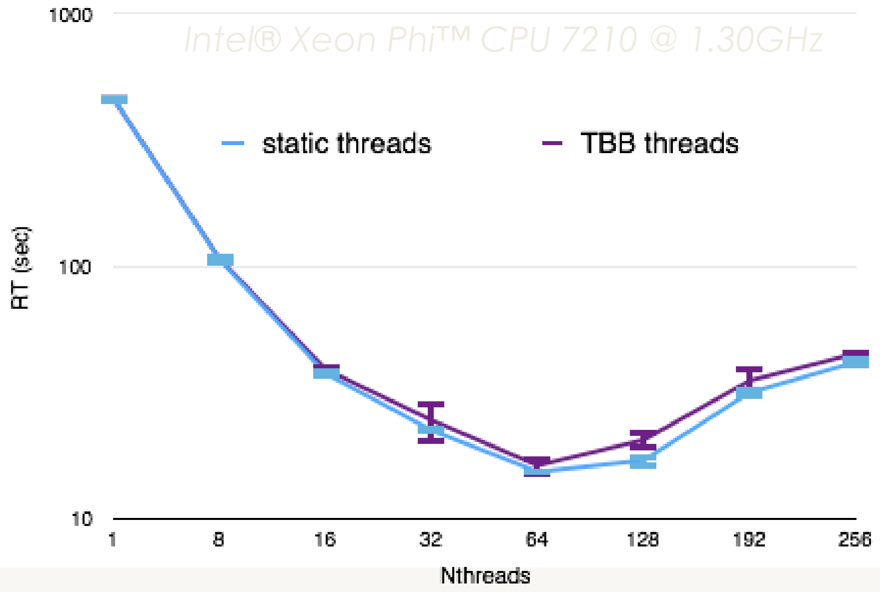
Good scalability up to the number of physical cores



Task based GeantV

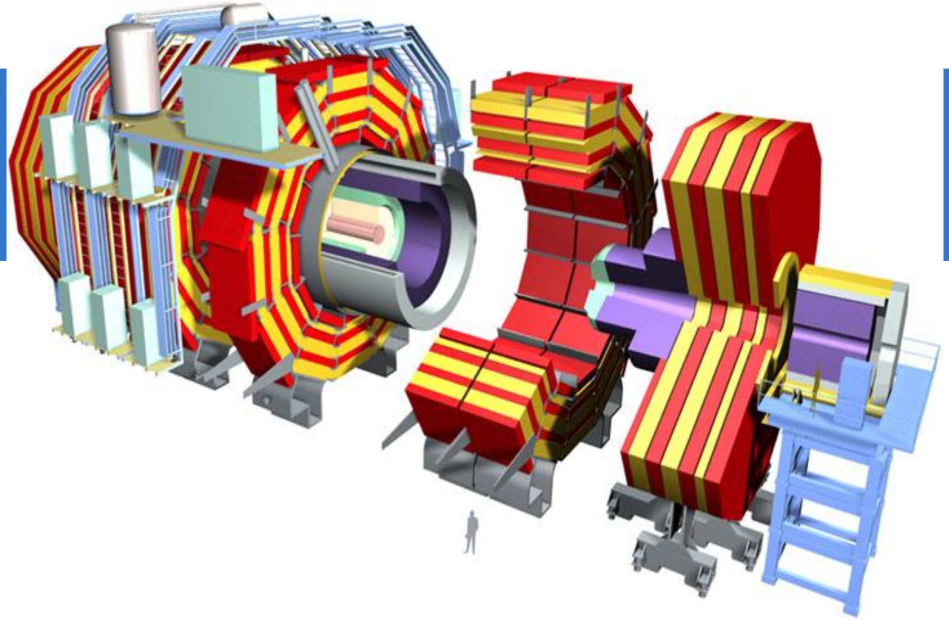
- A first implementation of TBB task-based approach on the full track transport prototype
  - Simplified detector geometry (calorimeter) + tabulated physics
- Some overheads on Haswell/AVX2, not so obvious on KNL/AVX512
- Less than 20% performance loss for the first implementation

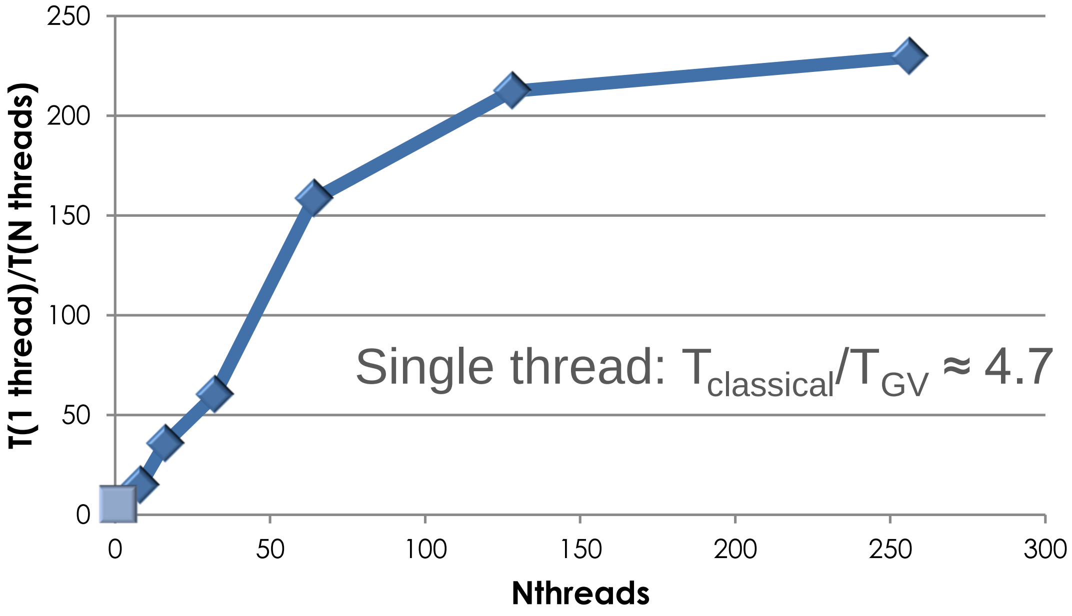




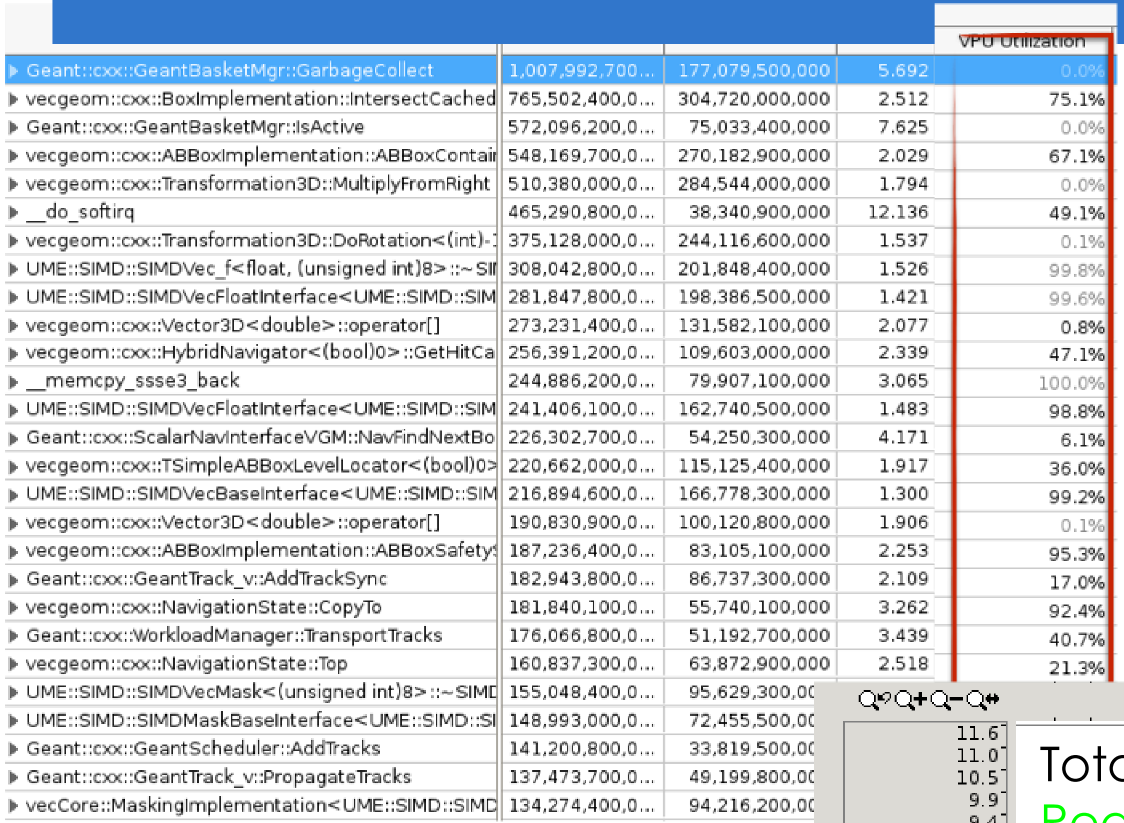
The full prototype

- Exercise at the scale of LHC experiments (CMS)
- Full geometry + uniform magnetic field
- Tabulated physics, fixed 1MeV energy threshold
- Full track transport and basketization procedure
- First results on scalability (comparison to classical approach single-thread)



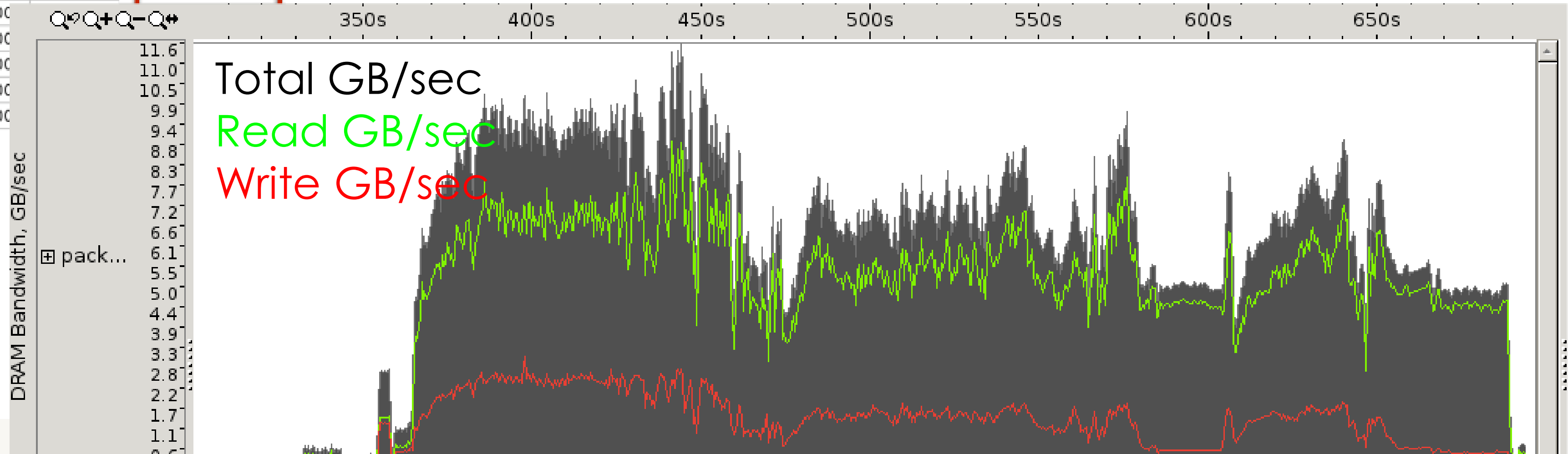


Full prototype performance on KNL



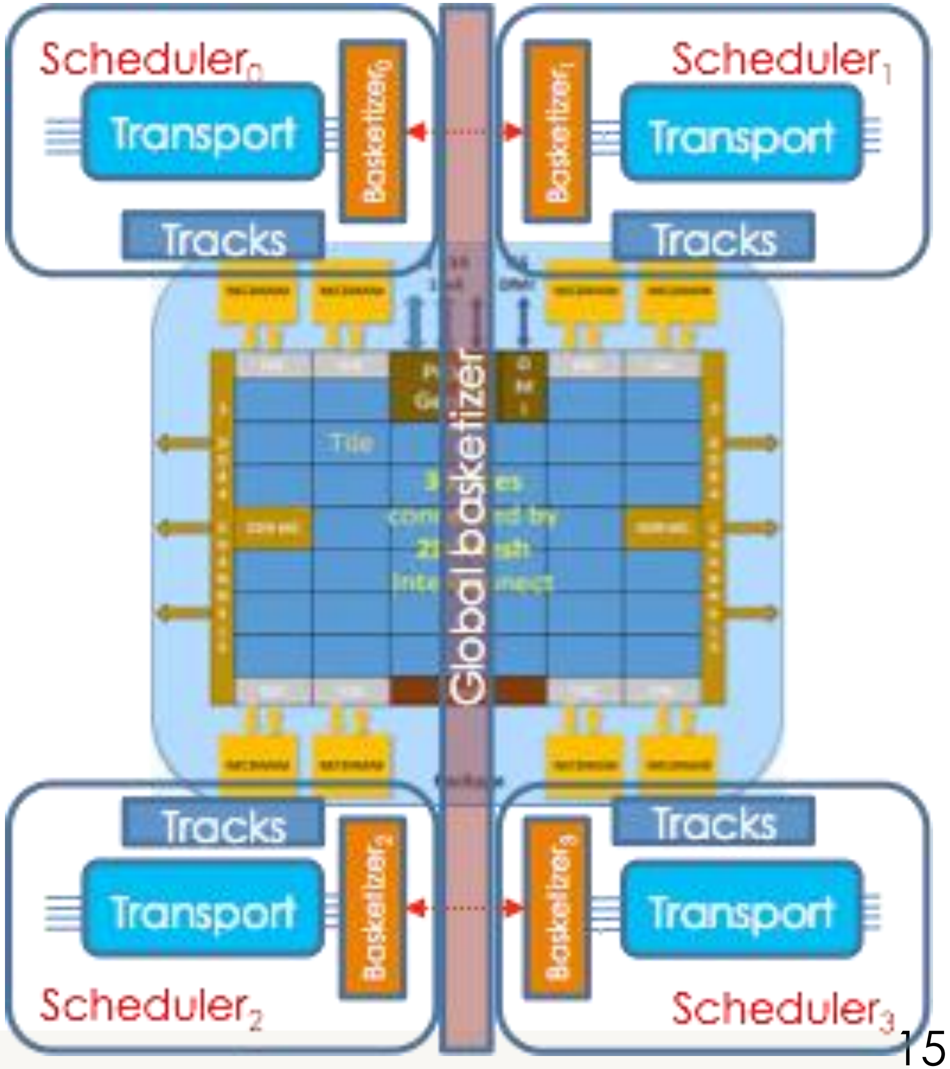
				VPU Utilization
Geant::cxx::GeantBasketMgr::GarbageCollect	1,007,992,700...	177,079,500,000	5.692	0.0%
vecgeom::cxx::BoxImplementation::IntersectCached	765,502,400,0...	304,720,000,000	2.512	75.1%
Geant::cxx::GeantBasketMgr::IsActive	572,096,200,0...	75,033,400,000	7.625	0.0%
vecgeom::cxx::ABBoxImplementation::ABBoxContair	548,169,700,0...	270,182,900,000	2.029	67.1%
vecgeom::cxx::Transformation3D::MultiplyFromRight	510,380,000,0...	284,544,000,000	1.794	0.0%
__do_softirq	465,290,800,0...	38,340,900,000	12.136	49.1%
vecgeom::cxx::Transformation3D::DoRotation<(int)-1	375,128,000,0...	244,116,600,000	1.537	0.1%
UME::SIMD::SIMDVec_f<float, (unsigned int)8>::~SII	308,042,800,0...	201,848,400,000	1.526	99.8%
UME::SIMD::SIMDVecFloatInterface<UME::SIMD::SIM	281,847,800,0...	198,386,500,000	1.421	99.6%
vecgeom::cxx::Vector3D<double>::operator[]	273,231,400,0...	131,582,100,000	2.077	0.8%
vecgeom::cxx::HybridNavigator<(bool)0>::GetHitCa	256,391,200,0...	109,603,000,000	2.339	47.1%
__memcpy_ssse3_back	244,886,200,0...	79,907,100,000	3.065	100.0%
UME::SIMD::SIMDVecFloatInterface<UME::SIMD::SIM	241,406,100,0...	162,740,500,000	1.483	98.8%
Geant::cxx::ScalarNavInterfaceVGM::NavFindNextBo	226,302,700,0...	54,250,300,000	4.171	6.1%
vecgeom::cxx::TSimpleABBoxLevelLocator<(bool)0>	220,662,000,0...	115,125,400,000	1.917	36.0%
UME::SIMD::SIMDVecBaseInterface<UME::SIMD::SIM	216,894,600,0...	166,778,300,000	1.300	99.2%
vecgeom::cxx::Vector3D<double>::operator[]	190,830,900,0...	100,120,800,000	1.906	0.1%
vecgeom::cxx::ABBoxImplementation::ABBoxSafety	187,236,400,0...	83,105,100,000	2.253	95.3%
Geant::cxx::GeantTrack_v::AddTrackSync	182,943,800,0...	86,737,300,000	2.109	17.0%
vecgeom::cxx::NavigationState::CopyTo	181,840,100,0...	55,740,100,000	3.262	92.4%
Geant::cxx::WorkloadManager::TransportTracks	176,066,800,0...	51,192,700,000	3.439	40.7%
vecgeom::cxx::NavigationState::Top	160,837,300,0...	63,872,900,000	2.518	21.3%
UME::SIMD::SIMDVecMask<(unsigned int)8>::~SIMD	155,048,400,0...	95,629,300,00		
UME::SIMD::SIMDMaskBaseInterface<UME::SIMD::SI	148,993,000,0...	72,455,500,00		
Geant::cxx::GeantScheduler::AddTracks	141,200,800,0...	33,819,500,00		
Geant::cxx::GeantTrack_v::PropagateTracks	137,473,700,0...	49,199,800,00		
vecCore::MaskingImplementation<UME::SIMD::SIMD	134,274,400,0...	94,216,200,00		

- Overall we fill VPUs reasonably well
- Memory access analysis shows we are not bandwidth bound: most of the code runs as “low utilisation”(<12 GB/sec)

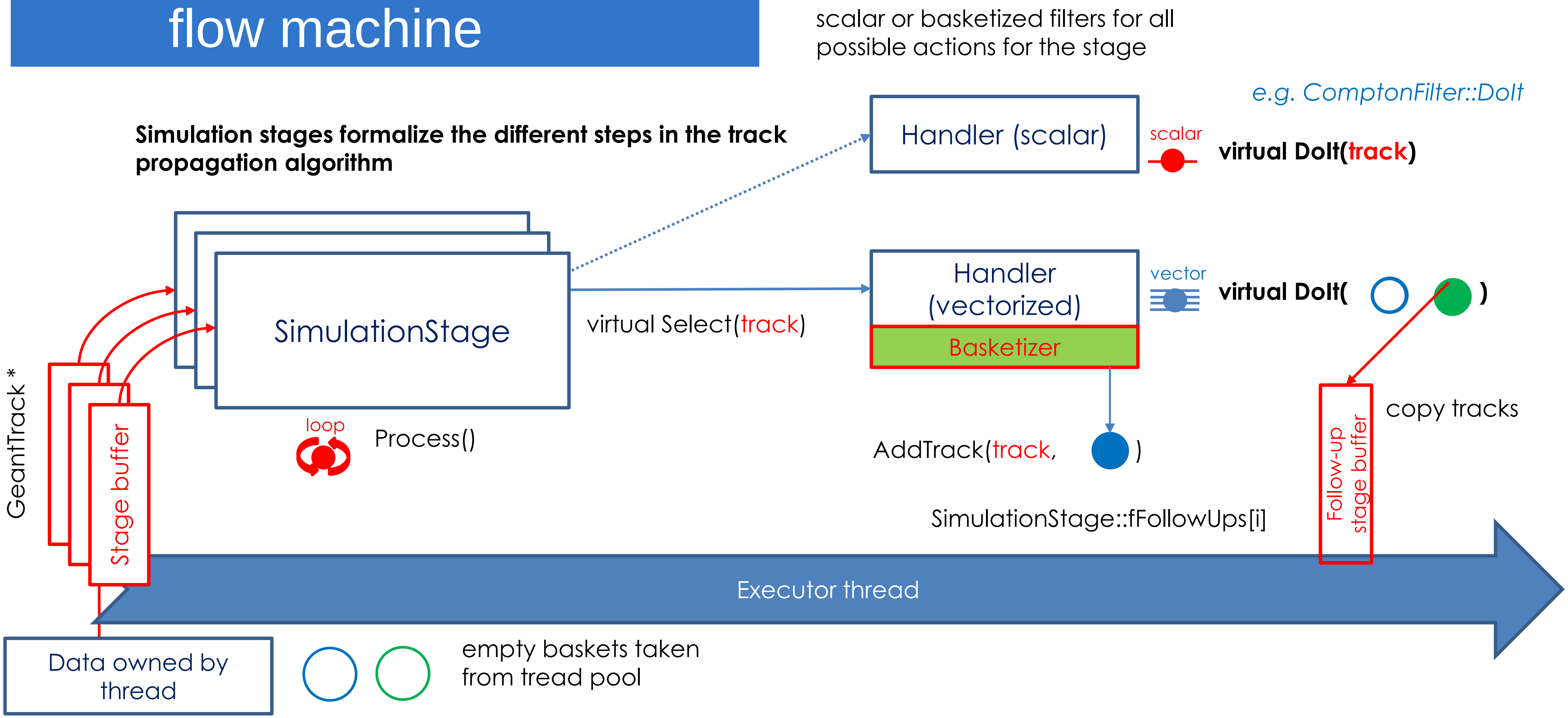


NUMA awareness

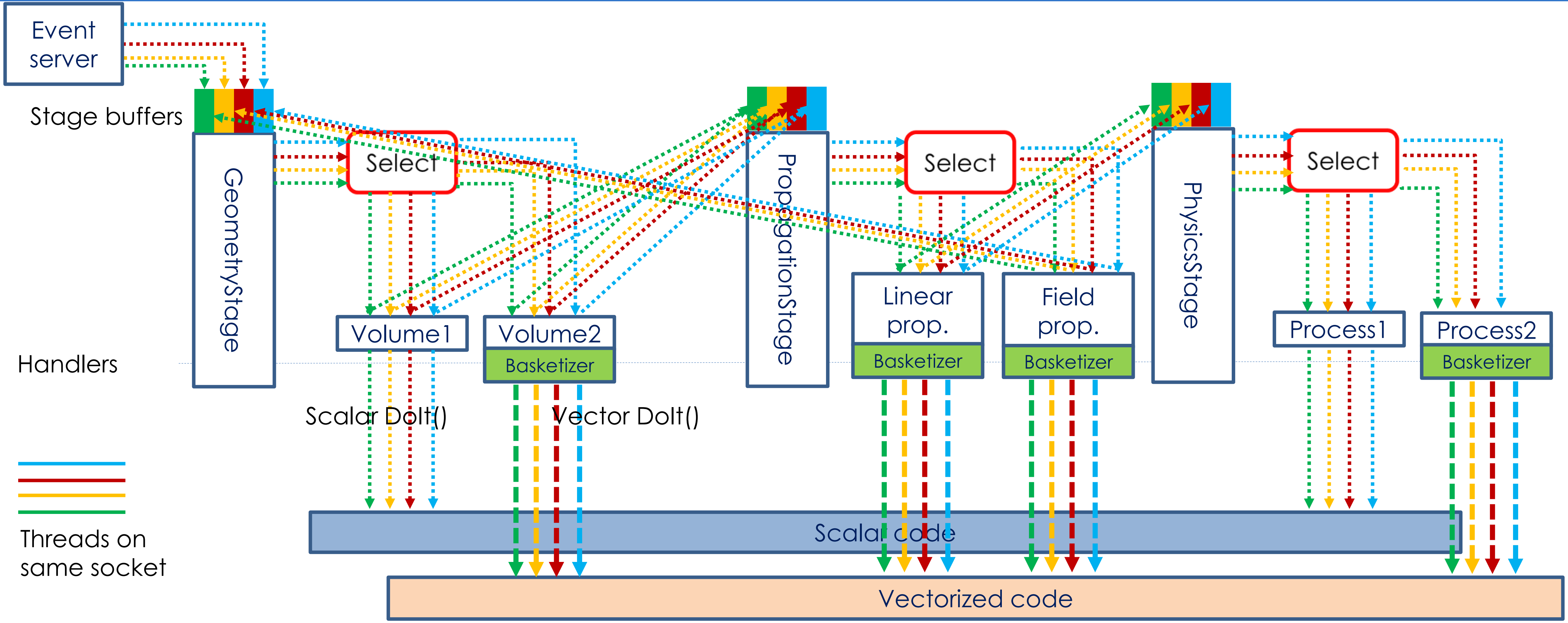
- Replicate schedulers on NUMA clusters
  - One basketizer per NUMA node
  - libhwloc to detect topology
  - Use pinning/NUMA allocators to increase locality
- Multi-propagator mode running one/more clusters per quadrant
  - Loose communication between NUMA nodes at basketizing step
  - Implemented, currently being integrated



V3: A generic vector flow machine



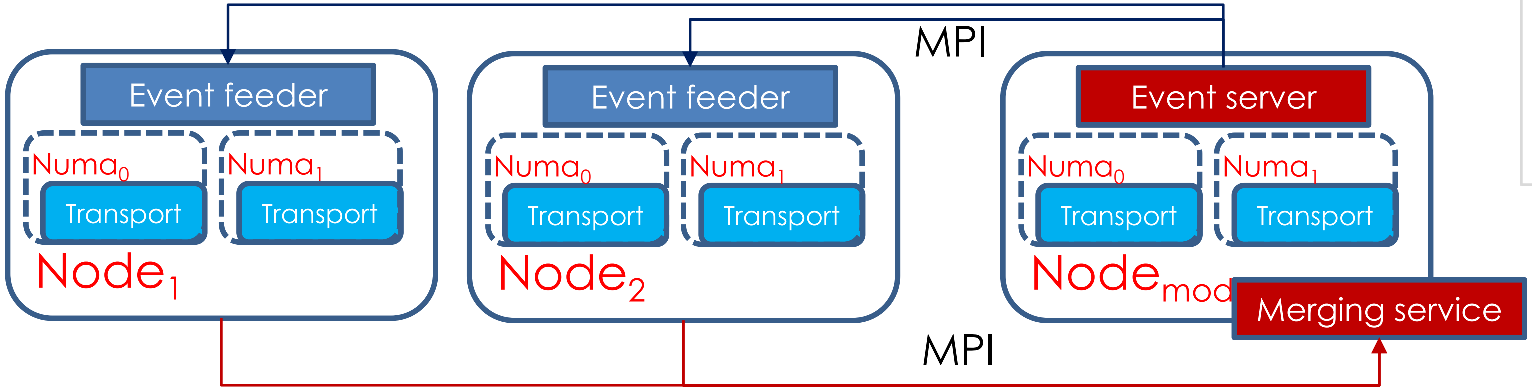
Processing flow per NUMA node

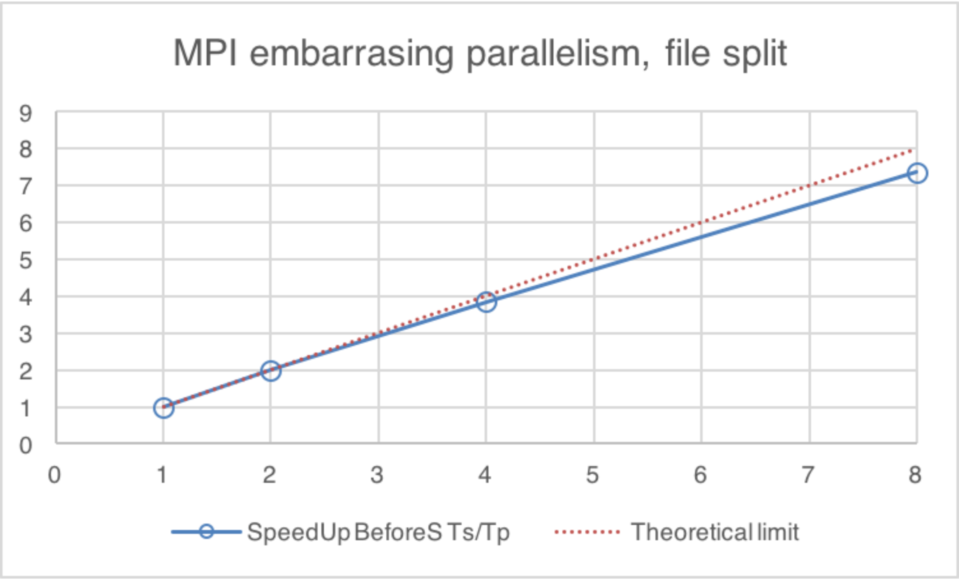


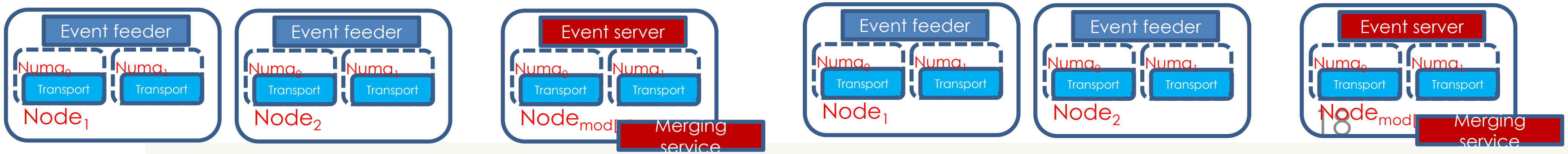
GeantV plans for HPC environments

R&D

- Standard mode (1 independent process per node)
  - Always possible, no-brainer
  - Possible issues with work balancing (events take different time)
  - Possible issues with output granularity (merging may be required)
- Multi-tier mode (event servers)
  - Useful to work with events from file, to handle merging and workload balancing
  - Communication with event servers via MPI to get event id's in common files







Intel® Parallel Computing Centers

Intel® PCC Proposal Presentation

Machine learning based tool for fast particle transport simulation in GeantV

CERN

Principle Investigator:
Dr. Federico Carminati
Federico.Carminati@cern.ch, +41227674949
CH-1211 Geneva 23

Executive Summary:
Increasing need for computing resources has prompted a sustained effort to optimize High Energy Physics (HEP) software and simulation for new computing architectures. A new prototype for particle transport simulation, GEANTV, is being developed to improve physics accuracy and performance on modern SIMD architectures, such as Xeon Phi. A faster approach is to treat simulation as a black-box that can be replaced by a deep learning algorithm trained on different particle types, momentum and position. We aim to develop a machine learning tool to replace traditional Monte Carlo simulation. Several techniques, such as multi-objective regression and data dimensionality reduction, will be applied to improve learning time and preserve correlations between input and output. Our plan is to target highly optimized current and next-generation Intel Xeon/Xeon Phi architectures for deep learning (upcoming 2017 Knights Mill and also the expected Lake Crest and "Knights Crest") by leveraging Intel DAAL libraries, Deep Learning SDK, MKL-DNN. For the application stage we plan to also evaluate the combined Xeon FPGA platform. We expect to achieve a significant speedup (x25) with respect to GeantV full simulation approach. Development of such machine learning simulation tools can further benefit other fields, such as radioactivity protection, environmental modeling and medicine.

Proposed Work Plan

Work Plan – Year 1		
Year 1: project start date is MONTH/DAY/2017 through end date MONTH/DAY/2018.		
Overall Goal For Year 1:		
Deliverables	Success Criteria	Timeframe
Specific actions of work performed	Significant results (performance improvements, Peer reviewed papers), help needed, etc.	Completion Date
• Evaluate machine learning model for multi-objective regression based on predictive clustering trees on simulated single particle dataset • Evaluate adversarial training model (GAN) on same dataset	• Prediction accuracy of detector response to single particles, proper treatment of output correlations, first results in feature evaluation, first timing benchmark of application	Year 1, Q1
• Extend deep learning regression to multiple targets (evaluate DAAL and Neon in this context) • Implement first GeantV fast simulation interface	• Multi-objective deep learning regression prototype • First interface to parametric fast simulation of a simple calorimeter	Year 1, Q2
• Model training and optimization with feature extraction • Evaluate auto-regression model in adversarial training for multiple targets	• Improved accuracy of models after feature extraction • Auto-regression model for multi-objective regression	Year 1, Q3
• Optimized deep learning multi-objective regression model • Extend GeantV interface to include non-parametric machine learning models	• Correct correlations, high accuracy, timing benchmark for standalone model application • Working prototype of non-parametric GeantV interface, first timing performance evaluation	Year 1, Q4

Alpha release of GeantV (Q4, 2017)

- **Version 3 of the scheduler**
  - Low overhead, scalable, AOS basketizing, new interfaces, new memory management (NUMA + shower burners)
  - Design/interfaces cleanup, refactoring of concurrency tools as separate library
  - Demonstrator for EM physics basketizing
  - Task model working with CMSSW
  - Efficient deployment on HPC clusters – R&D
- **Complete user interfaces (discussed with experiments)**
  - Full workflow simulation -> digitizers -> I/O stressing user interface (both standalone GeantV examples and TBB CMSSW)
  - MC truth user hooks defined + most common use case demonstrators
- **Efficient vectorized RK propagator including optimizations (last field value, helix fallback)**
- **Geometry with complete navigation features demonstrating vector gains (2017 release)**
  - Specialized navigators in action, including training/deployment model
- **EM physics: most(?) e+/e-/gamma models in scalar mode + some vector gains**
  - Integration of MSC, development/finalizing of ionization, bremsstrahlung, pair production, Compton, photoelectric
- **Hadronic x-sec from tables, Glauber-Gribov hadronic cross sections, Hadron elastic model, Part I**
- **Fast simulation "hooks" a la G4 demonstrated to work in the basket flow**
  - Formalizing user interface, scope definition R&D, start development of a Multi-Objective regression tool
- **GPU demonstrator capable of doing complete simulation (e.g. CMS, no optimization)**
- **Testing/validation suite and performance demonstrators vs. Geant4**

Beta release of GeantV (Q4, 2018)

- Production-quality scheduling, including error handling at the level of track/event
  - Optimization based on integration with experiment frameworks (user interfaces, digitizers flow)
  - Demonstrator for performance in HPC environments
  - Tuning procedure for scheduling parameters based on ML/GA
- Production-quality geometry (2018 release)
  - Supporting all features of G4/ROOT, full set of shapes, demonstrators for all 4 LHC experiments
  - Extended validation suite, robustness demonstration
- Demonstrator for efficient MC truth usage, based on realistic use cases from experiments
- Full EM shower physics, most CPU-consuming models vectorized
  - Benchmarks demonstrating vector mode and speedup compared to G4 equivalent
- Hadronics – hadronic elastic implemented + QGS part I
  - Complete model-level & application-level tests
- Fast sim demonstrators for most common use cases
  - Integration with experiment frameworks
  - Demonstrator for the full learn/replay procedure – ML standalone tool + performance study for different detectors

Conclusion and insights

- GeantV delivers already a part of the expected performance on KNL
    - Many optimization requirements, now understanding how to handle most of them
- Additional levels of locality (NUMA) available: topology detection already in GeantV, currently being integrated
- Exploring task-based approach: TBB-enabled version is ready
- Next step: V3 core in production, integration with physics and optimization
- 2017 & 2018 – ambitious program of work, aiming to releasing a product having most of the target features to experiments