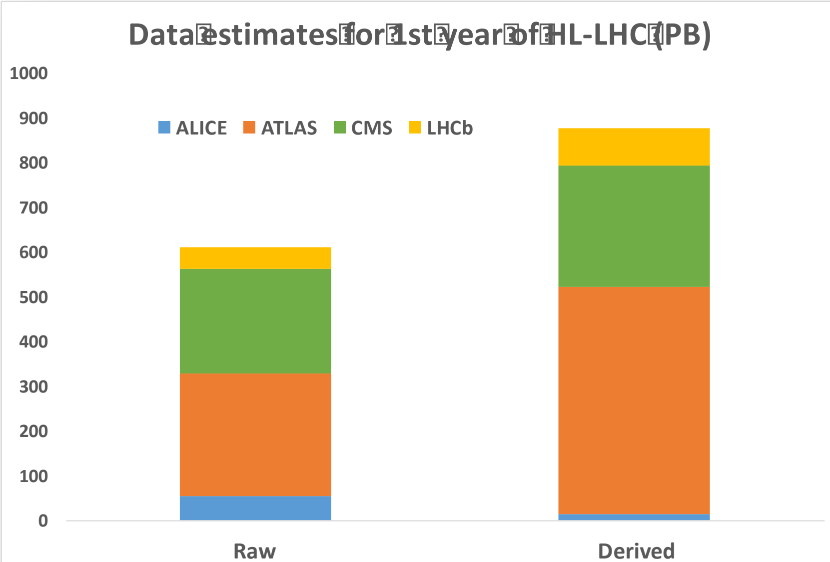
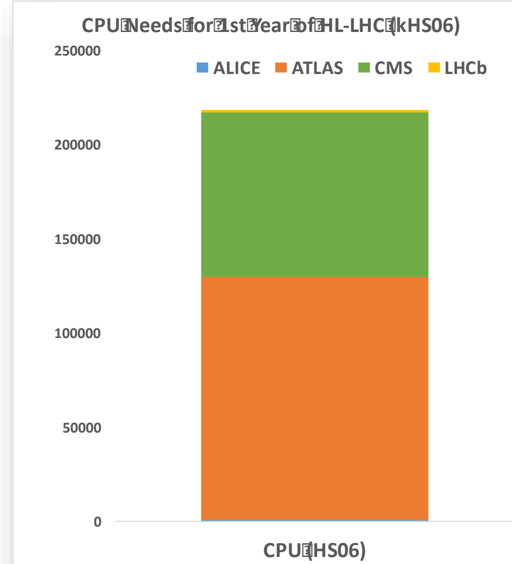
GeantV – An overview

Federico Carminati (CERN) for the GeantV development team

Estimates of resource needs for HL-LHC





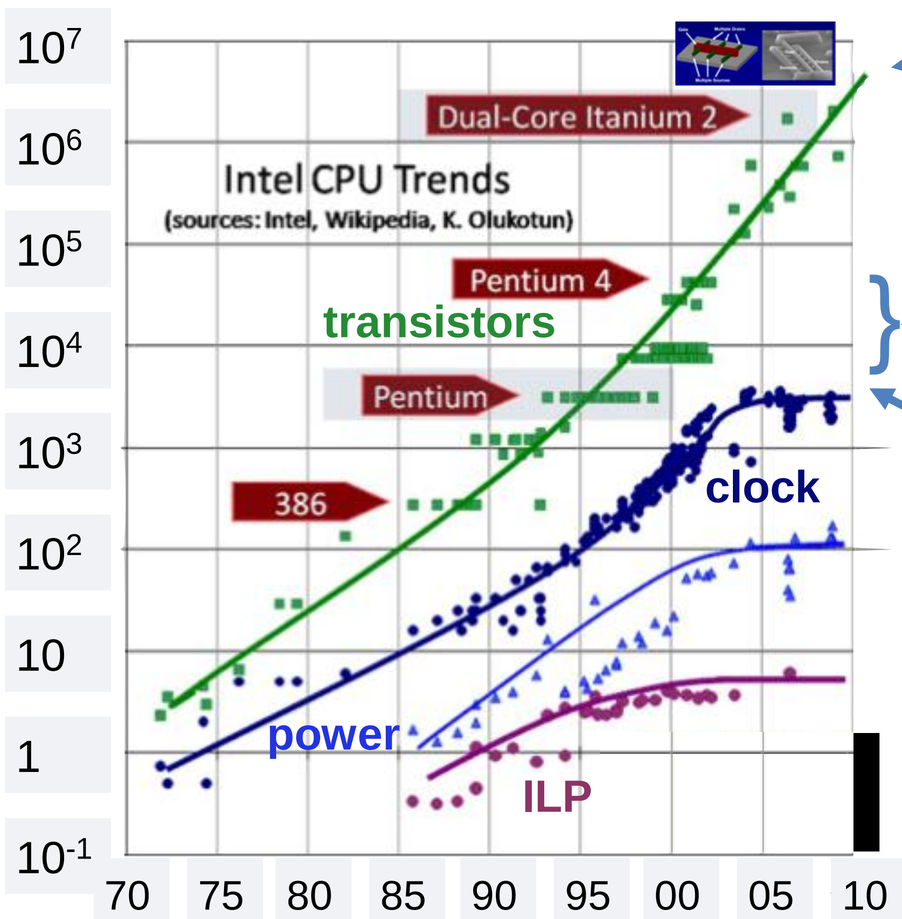
Data:
- Raw 2016: 50 PB → 2027: 600 PB
- Derived (1 copy): 2016: 80 PB → 2027: 900 PB

CPU:
- x60 from 2016

Technology at ~20%/year will bring x6-10 in 10-11 years

- Simple model based on today's computing models, but with expected HL-LHC operating parameters (pile-up, trigger rates, etc.)
- At least x10 above what is realistic to expect from technology with reasonably constant cost

Motivations
(even if you are familiar with them)



The above is true only if we are here

We are now probably here

We used to be here

The Eight dimensions

- The “dimensions of performance”
  - Vectors
  - Instruction Pipelining
  - Instruction Level Parallelism (ILP)
  - Hardware threading
  - Clock frequency
  - Multi-core
  - Multi-socket
  - Multi-node

Fine-grain parallelism: gain in throughput and in time-to-solution

Very little gain to be expected and no action to be taken

Gain in memory footprint and time-to-solution but not in throughput

Possibly running different jobs as we do now is the best solution

Expected limits on performance scaling

	SIMD	ILP	HW THREADS
THEORY	8	4	1.35
OPTIMISED	6	1.57	1.25
HEP	1	0.8	1.25

Expected limits on performance scaling (multiplied)

	SIMD	ILP	HW THREADS
THEORY	8	32	43.2
OPTIMISED	6	9.43	11.79
HEP	1	0.8	4 1

Why is it so difficult?

- No clear kernel
- C++XX code generation / optimisation evolving rapidly
- Most of the technology coming out now
- Lack of standards
- Technological risk
- Physicist coders
- Fast evolving code
- Loose control on hardware acquisition

Why simulation?

- Simulation accounts for 50% of the WLCG cycles
- A large share of the simulation code is common to all users
- A single program dominates the scene for full detector simulation: GEANT4
- Parasitic time availability on supercomputers can be best exploited by simulation
- I/O problem can be factorized out
  - That is NOT to say that it can be ignored!
- Strong encouragement by CERN management to start this project

YEARS/ANS CERN

FNAL, BARC, UNESP and INAF (and SLAC)

- Strong encouragement from FNAL management and US DOE to start a project to re-engineer Geant4 for modern computing architectures.
  - Having the same goals and sharing the same initial conclusions, we decided to join forces
- UNESP joined via an Intel IPCC
- INAF shares our interest in optimizing the code and has been a partner since the beginning of GEANTV
- BARC has a long history of developing and using simulation codes, and joined the project
- SLAC will collaborate with us to develop an independent module for neutron transport

… openlab & Intel

- openlab & Intel provide strong support for this (first?) large-scale effort to vectorize HEP code
- GEANTV is one of the openlab applications
- Intel has granted GEANTV two IPCC

Intel® Parallel Computing Center at CERN, European Organisation for Nuclear Research

Added December 19, 2014 Translate

Share Tweet Share

Forums

Tools

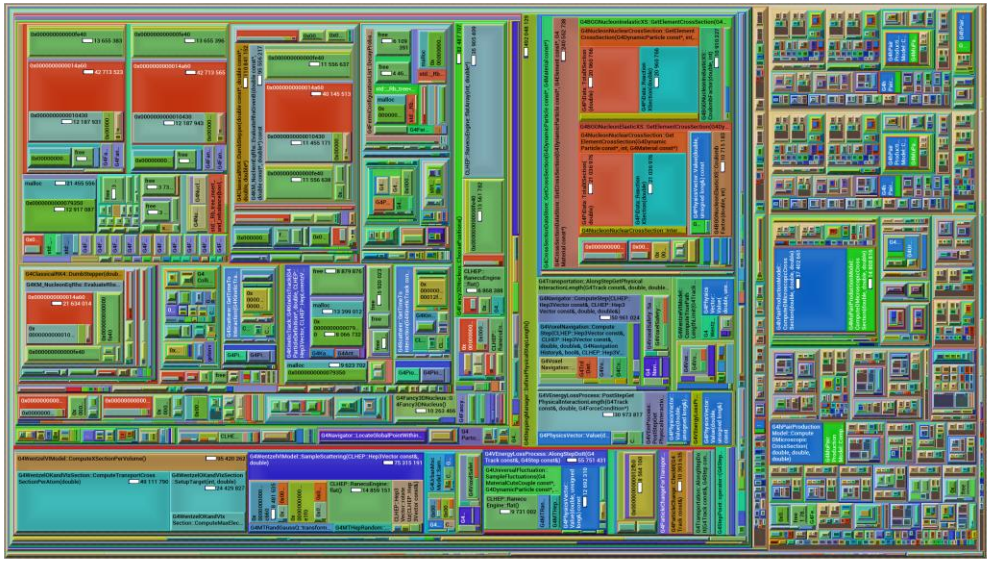
YEARS/ANS CERN

Principal Investigators:

Federico Carminati works at CERN, (Geneva Switzerland) where he leads the detector simulation activities. After his degree in 1981 he worked at Los Alamos and Caltech before coming to CERN, where he was responsible for the CERN Program Library, the worldwide standard High Energy Physics code in the 80's and 90's. From 1994 to 1998 he worked with Nobel Prize Carlo Rubbia at the design of a novel accelerator-driven nuclear power device. From 1998 to 2012 he was Computing Coordinator of the ALICE experiment at LHC. In 2013 he obtained his PhD in physics from the University of Nantes.

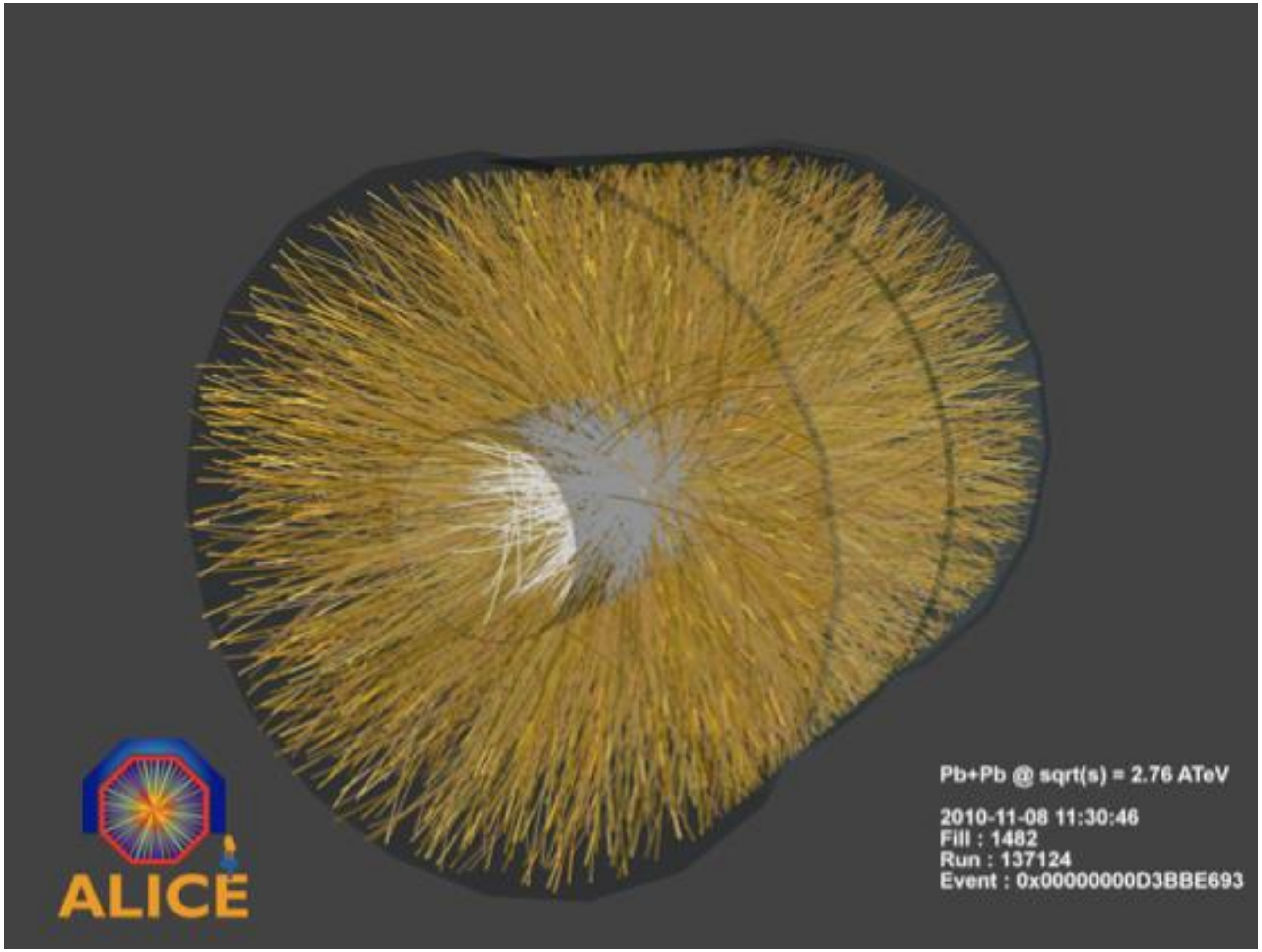
Why not Geant4+?

- No hotspots (!)
- Virtual table structure very deep and complex (1990's style)
- Codebase very large and non-homogeneous
- No criticism, but even the best things age



Extensive R&D convinced us that "vectorisation" of GEANT4 was not achievable without a major rewrite of the code

Parallelism everywhere again… but how to exploit it?



?

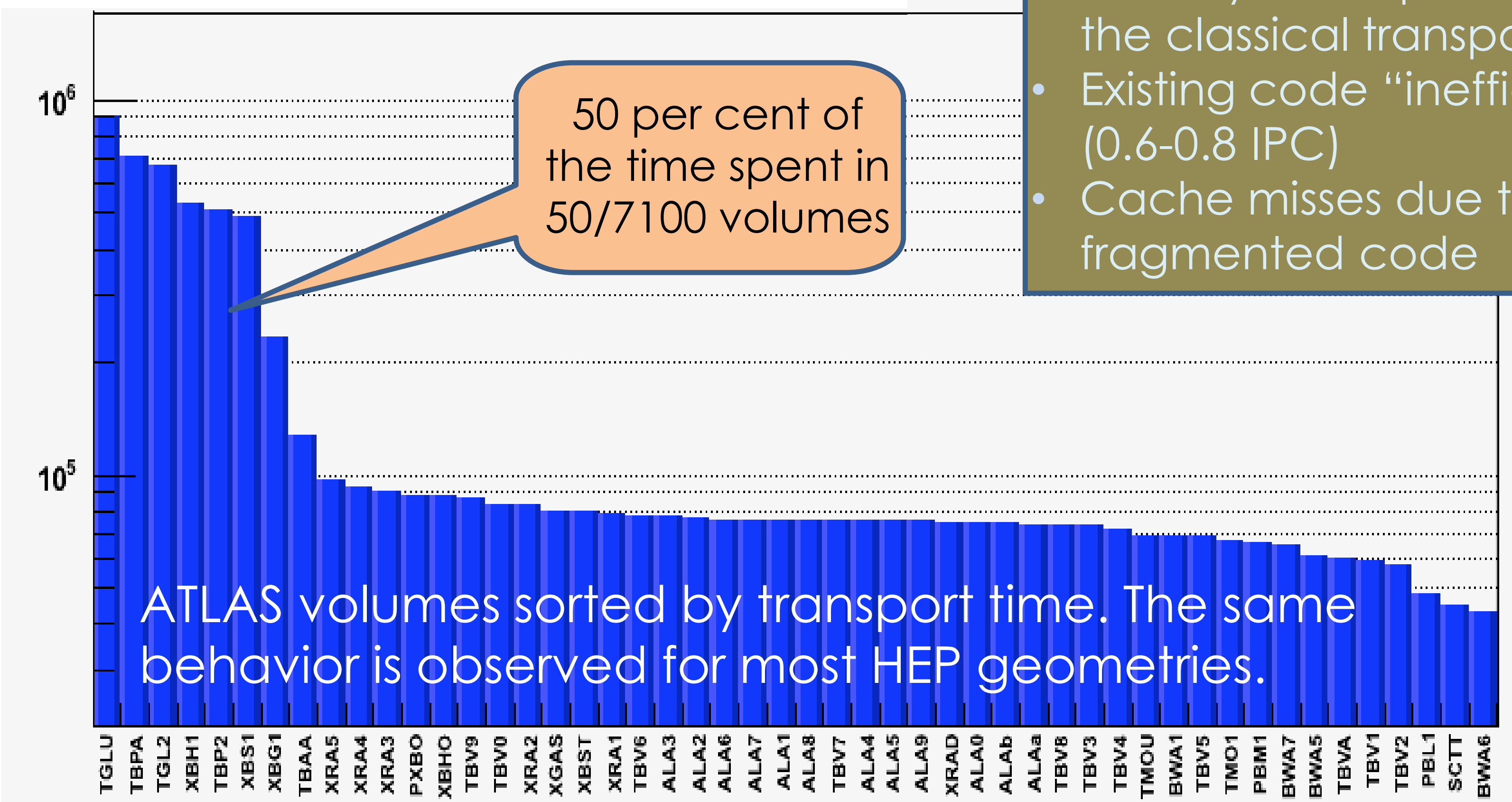
In some sense… but not entirely (see later)

Hotspots?

- F.Carminati: The problem of our code is that there are no hotspots
- S.Jarp (openlab director): Well, create them and then you will be able to optimize them

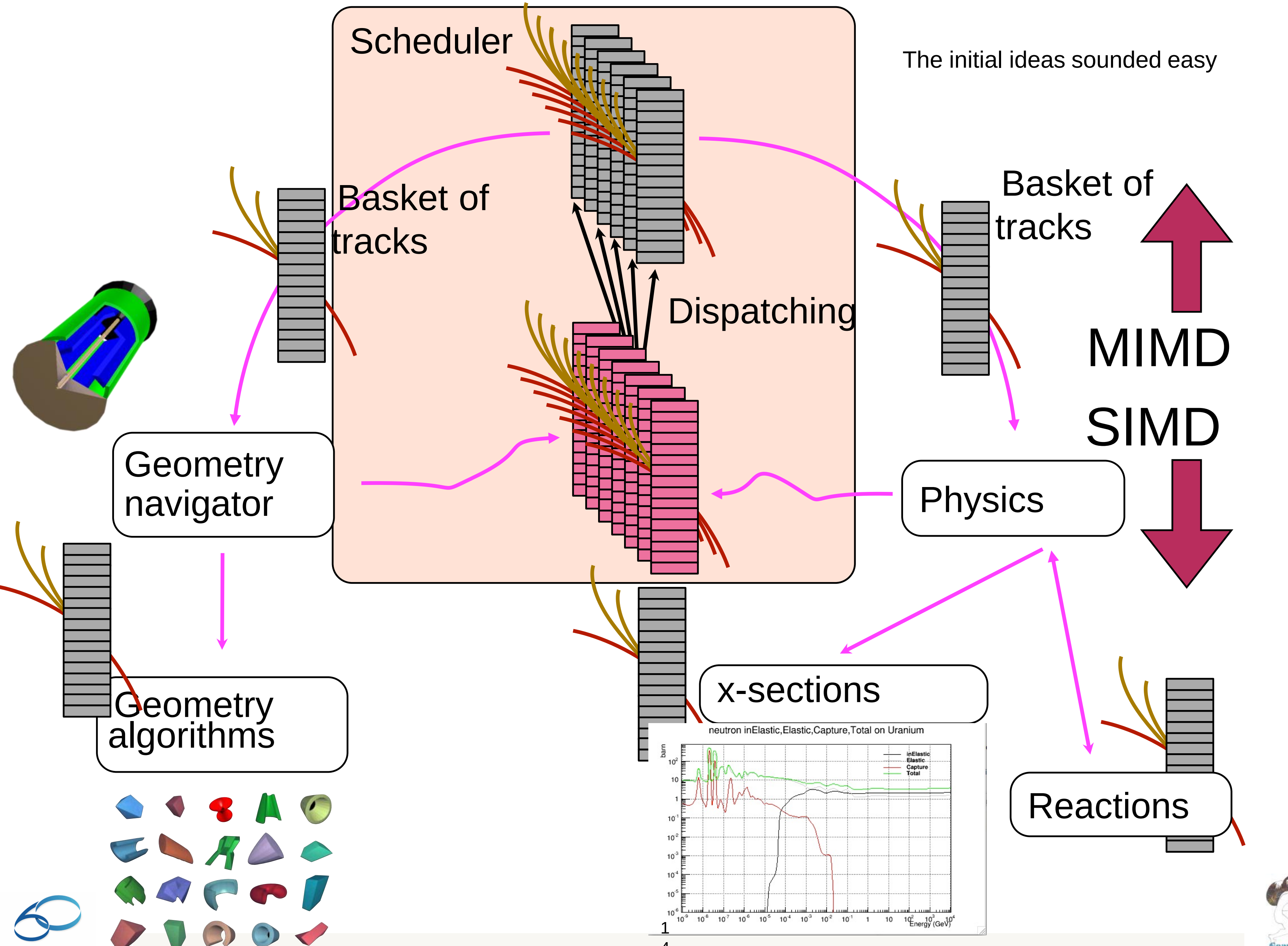
HEP transport is mostly local !

- Locality not exploited by the classical transport
- Existing code “inefficient” (0.6-0.8 IPC)
- Cache misses due to fragmented code



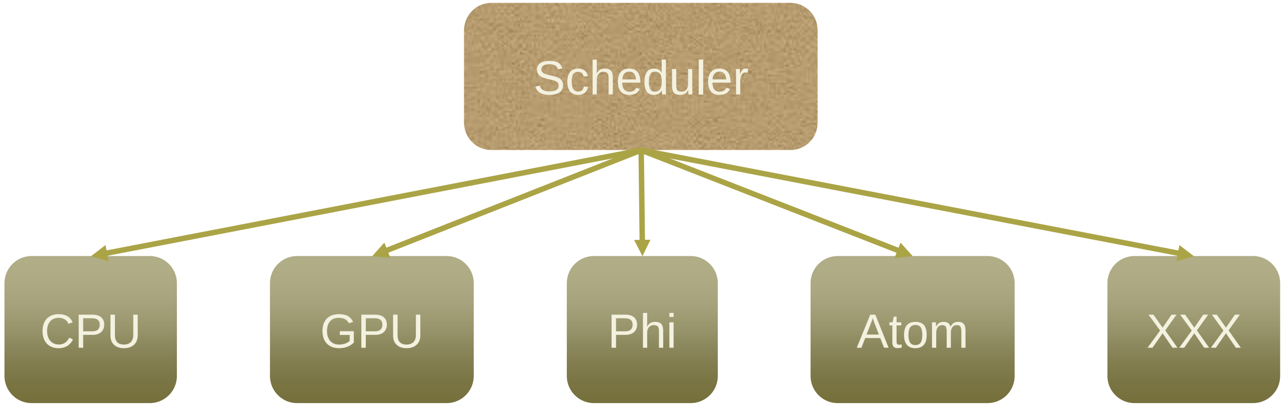
50 per cent of the time spent in 50/7100 volumes

ATLAS volumes sorted by transport time. The same behavior is observed for most HEP geometries.


The initial ideas sounded easy
Scheduler
Basket of tracks
Basket of tracks
Dispatching
MIMD
SIMD
Geometry navigator
Physics
Geometry algorithms
x-sections
neutron inElastic,Elastic,Capture,Total on Uranium
inElastic
Elastic
Capture
Total
barn
Energy (GeV)
Reactions

Challenges

- Overhead of particle lists should not offset SIMD gains
- Exploit the metal at its best, while maintaining portability



- Test from the onset on a “large” setup (LHC-like detector)
  - Toy models tell us very little – complexity is the problem

Redesigning simulation

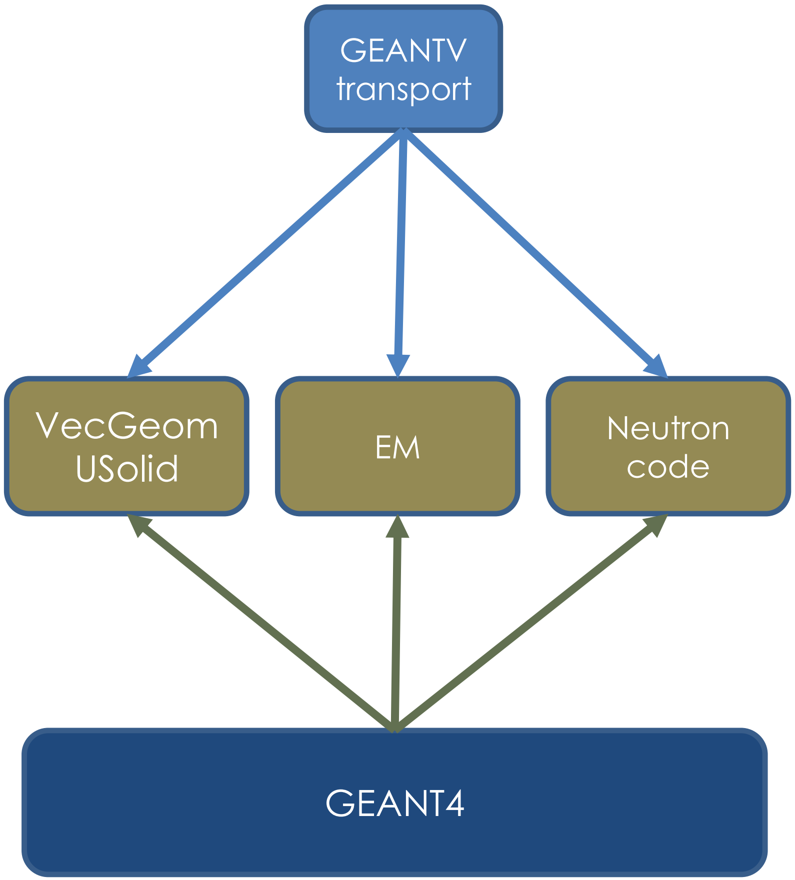
- Maintain minimal coupling between components, in particular kernel and physics
- Re-use as much as possible the design of Geant4 physics
  - Avoid reinventing the wheel
- Streamlining by reducing the inheritance depth
  - Replace whenever possible, virtual polymorphism with static (template) polymorphism
  - Offer *thin* interfaces for direct calls to cross sections and final-state models
- Build into the design provision for biasing and fast simulation

The VecGeom story

- The first thing we set out to develop for GEANTV was a high performance geometry navigator
- Since USolid was not targeting SIMD, we decided to extend USolid with VecGeom
- This allows us to validate VecGeom with GEANT4 and ROOT applications
- … and provided us with an entirely new idea on how develop components for GEANTV

The casual modular designer
AKA fighting against ourselves

- Develop modular codes that can be used by GEANTV, GEANT4 and beyond
- Use GEANT4 to validate them, and, if appropriate, make them part of GEANT4
- This ensures a continuous benefit for the community coming from the project
- And this continues to move the goalpost for us ☹



Redesigning simulation

- Design components that can be used by different simulation codes
  - Highest performance when used in vector mode
  - Portable on CPUs and accelerators thanks to well identified code kernels (do NOT spread ifdefs in the code!)
  - Make sure they can be used by GEANT4 and GEANTV at least
- Progressively build a set of tests as granular as possible taking the data from a database
  - Introduce automatic alarms for failing tests
- Perform continuous integration and testing
  - Nightly
  - At every pull request

Timescales

- The lifecycle of a simulation package is decades
- Introducing changes requires close coordination with the customers
- In the case of LHC, anything new has to be delivered before 2019 to be even considered before the end of Run 3
  - Need LS2 (2019-2020) for commissioning
- This requires planning & stable resource commitment

Planning & (wo)manpower

- We are indebted to funding agencies for the support and encouragement as early as 2012
- We have adjusted planning according to the skills we have at our disposal
  - This in order to maximize the results of the "proof of concept"
- The "modular development" and staged approach, and the "common house structure" (GEANT4 & GEANTV) at CERN and FNAL (soon SLAC – neutron package) allows us to leverage GEANTV effort for the good of GEANT4 and reciprocally
- More on personpower and planning at the end…