Track Summary Event Processing, Simulation and Analysis

Peter Elmer (Princeton University)
Rolf Seuster (TRIUMF)
Florian Uhlig (GSI)

Statistics

CHEP 2013

59(53) contributions

31(27) poster

28(26) talks

1 Vidyo presentation

CHEP 2012

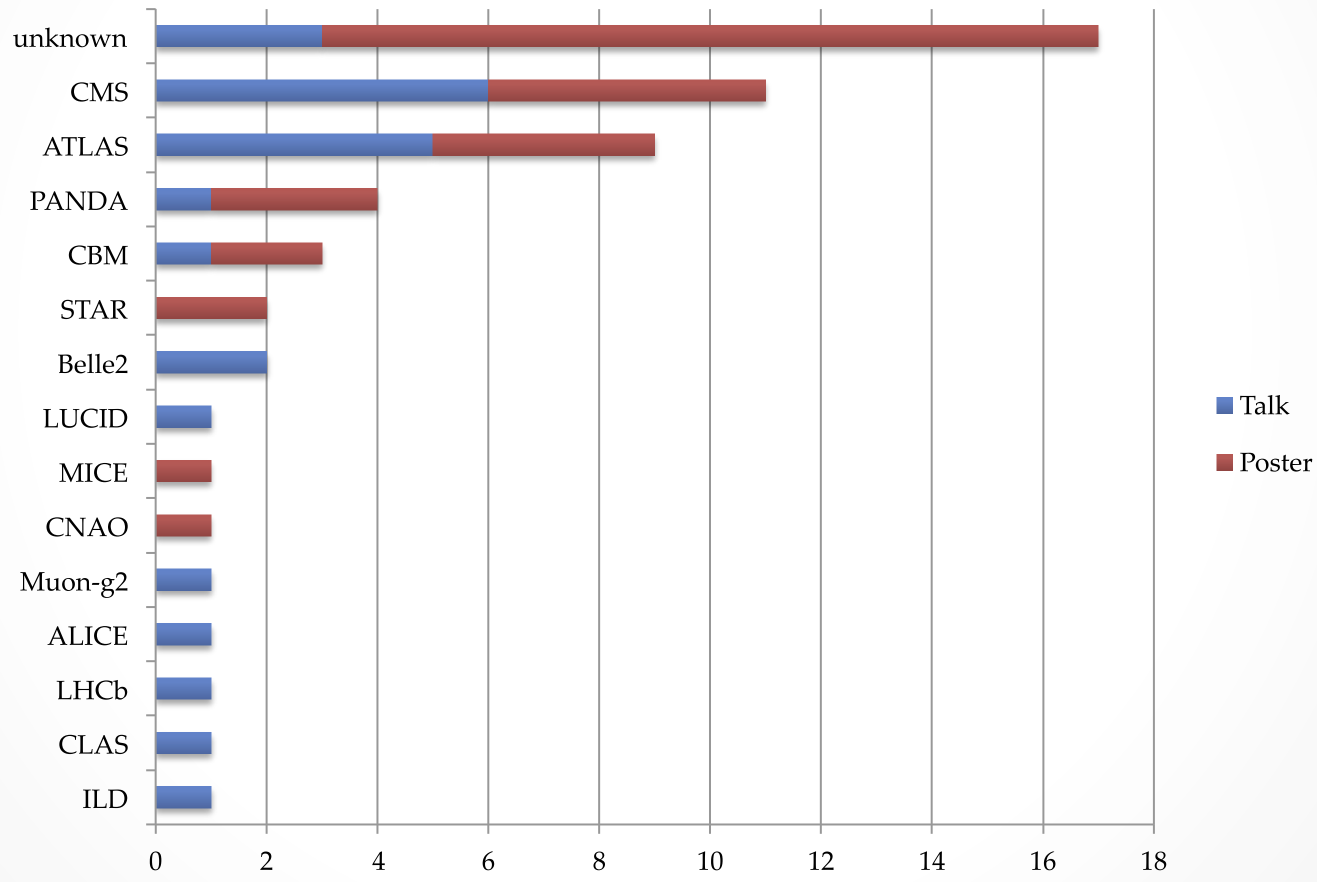
84 contributions

64 posters

20 talks

- Between 20 and 50 participants in the different sessions
  - Much less then in previous CHEPs
  - Many people probably in track 5: Software Engineering, Parallelism & Multi-Core
- Many people jumped between the different tracks
- In the presentation I will focus on the talks and try to give an overview over the topics not the talks

Which experiments are represented?



What is this all about?

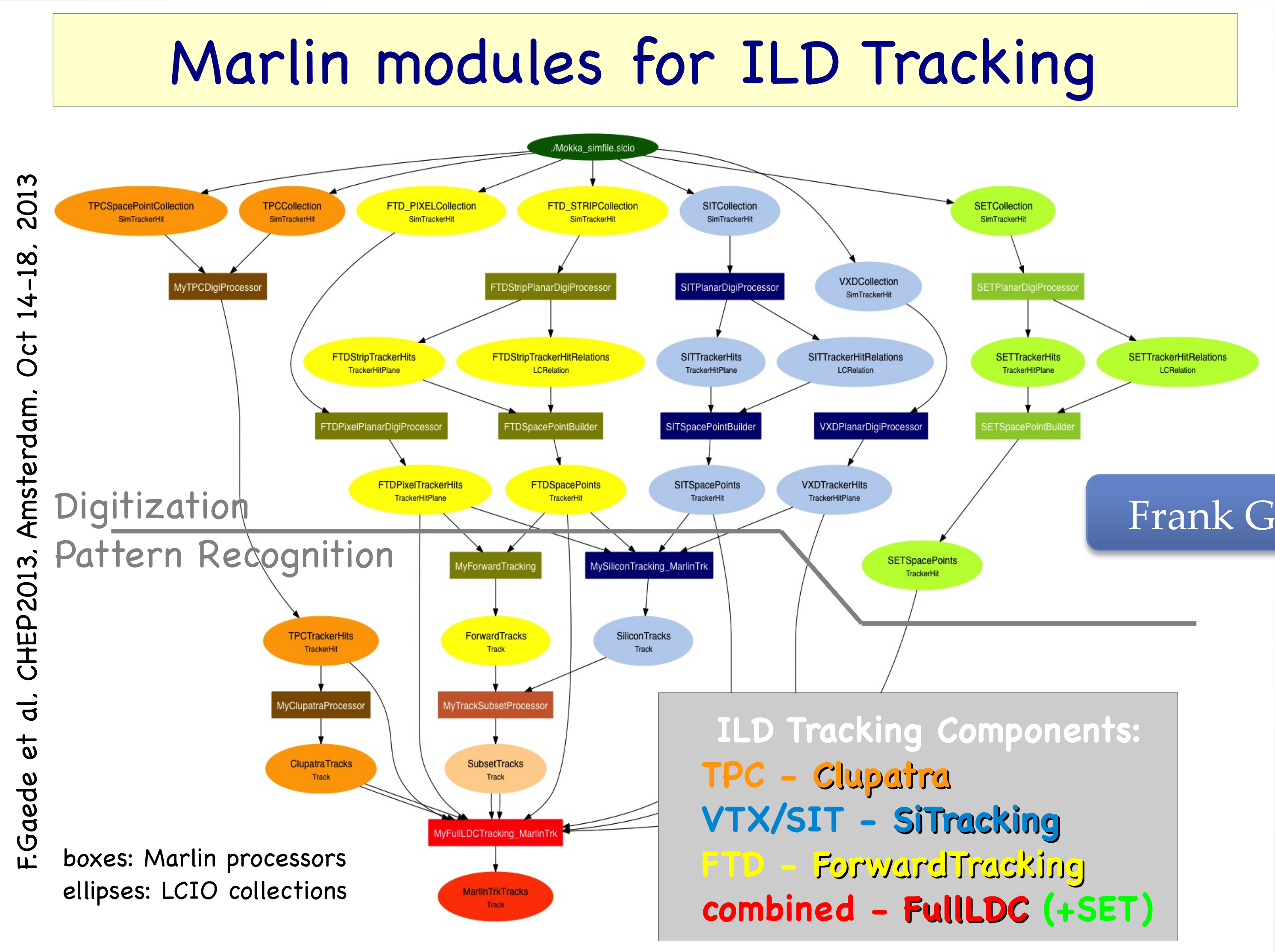
Outline

- Common Frameworks
- Concurrency
- Algorithms
- Pileup Simulation
- Future Simulation for LHC
- Everything else

Common Frameworks

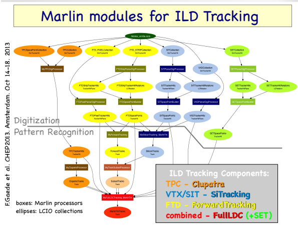
- Many different frameworks presented
  - For sure the big and well known ones ATLAS(Gaudi) LHCb(Gaudi), CMS
  - Many others
    - International Large Detector@ILC

Common Frameworks



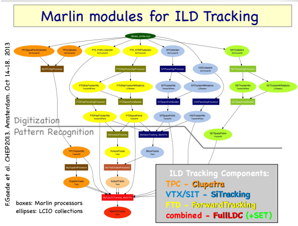
Frank Gaede

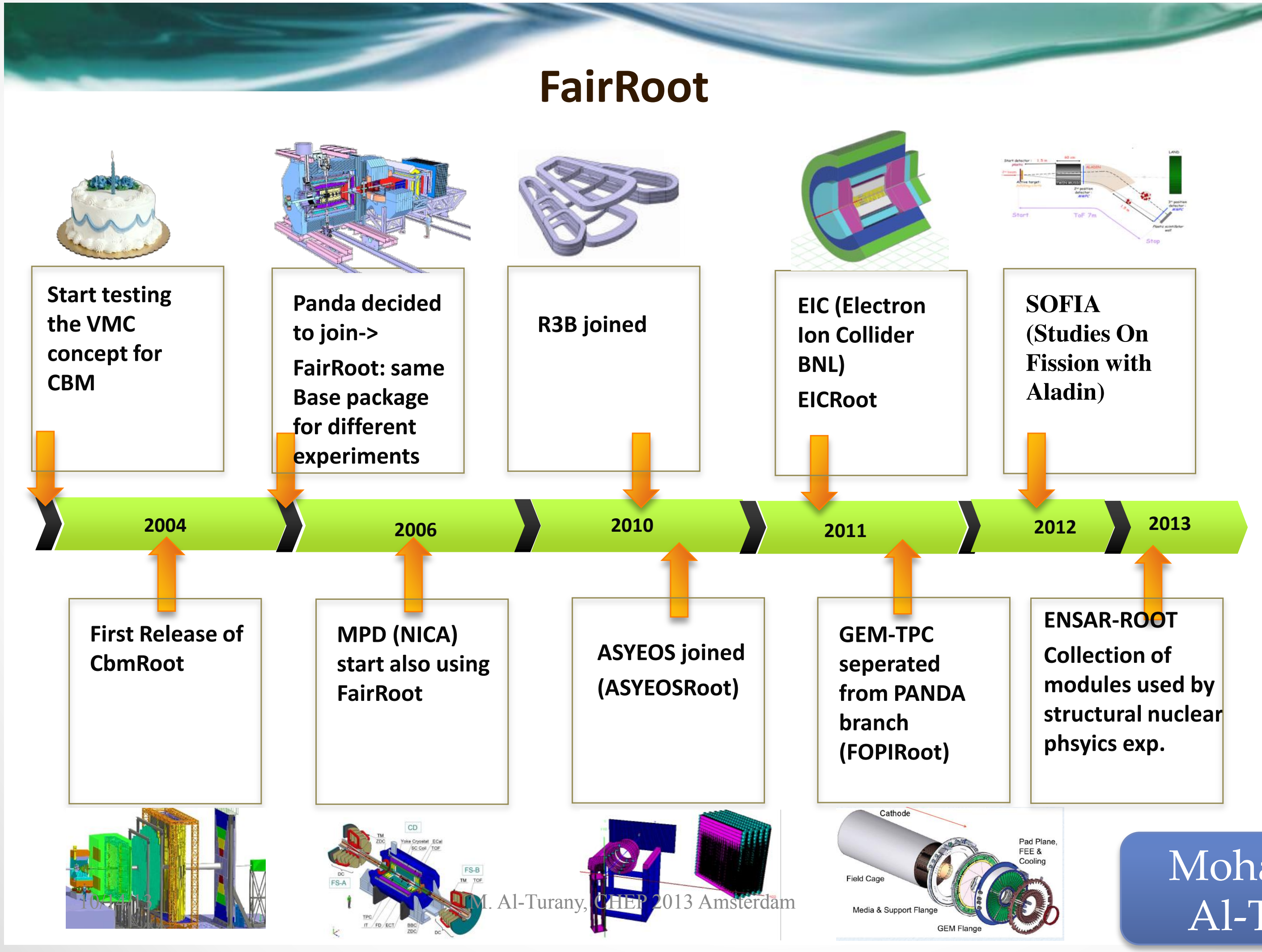
Common Frameworks



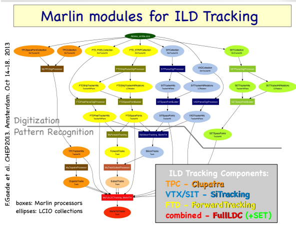
- Many different frameworks presented
  - For sure the big ones ATLAS(Athena), LHCb(Gaudi), CMS
  - Many smaller ones
    - International Large Detector@ILC(Marlin)
    - CBM, Panda (FairRoot)
      - Many other experiments meanwhile

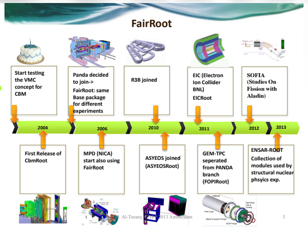
Common Frameworks





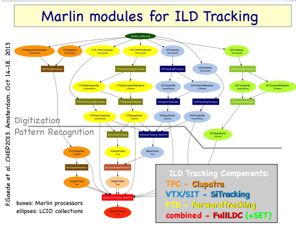
Common Frameworks

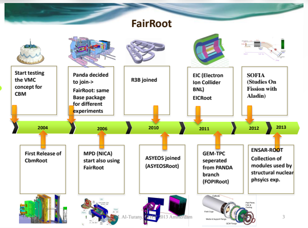


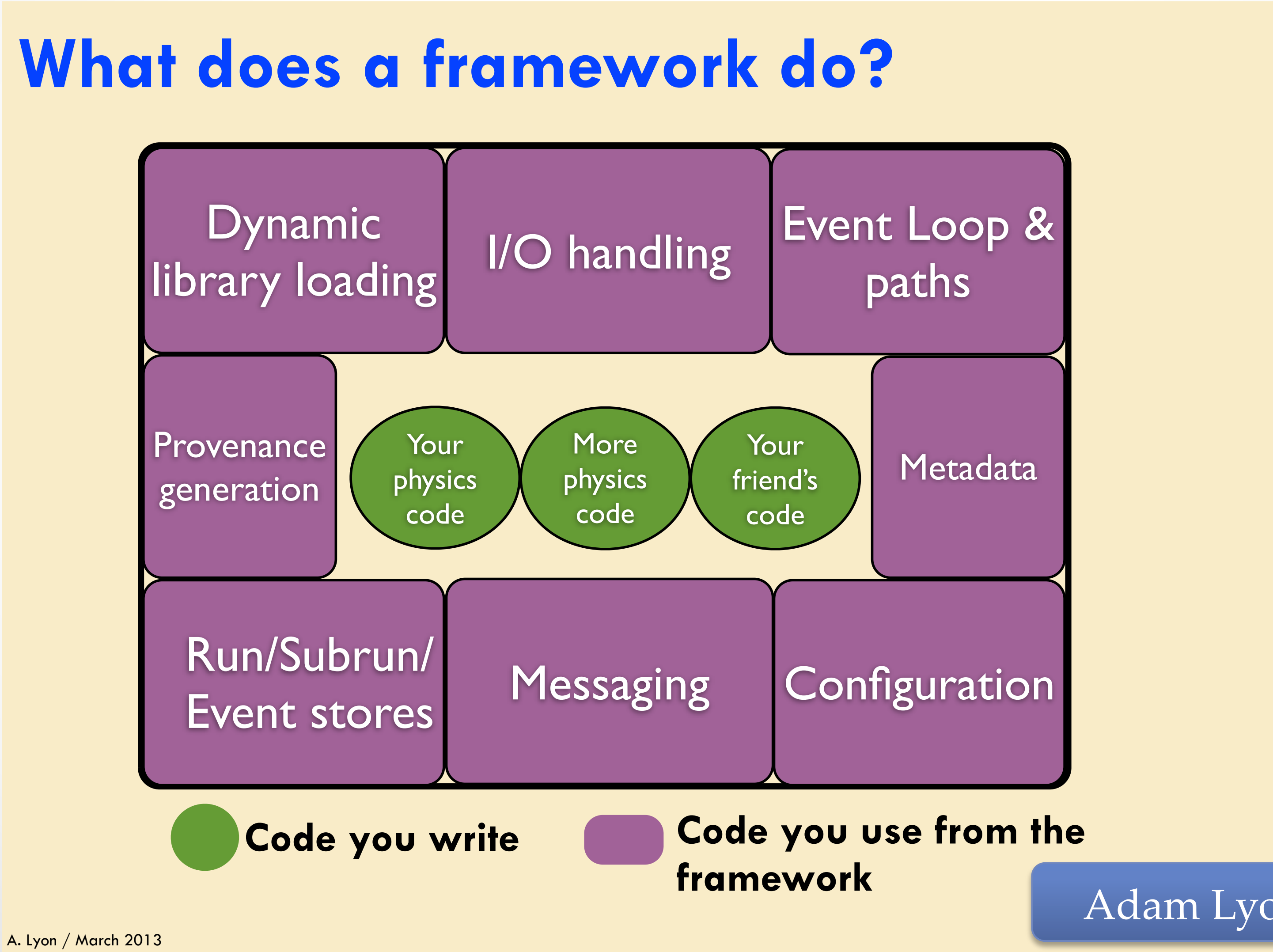


- Many different frameworks presented
  - For sure the big ones ATLAS(Athena), LHCb(Gaudi), CMS
  - Many smaller ones
    - International Large Detector@ILC(Marlin)
    - CBM, Panda (FairRoot)
      - Many other experiments meanwhile
    - Muon g-2 (ArtG4 based on Art, lite fork from CMS)
      - NOvA, *Mu2e*, MicroBoone, LBNE, …

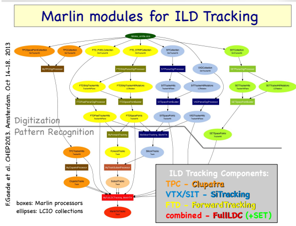
Common Frameworks

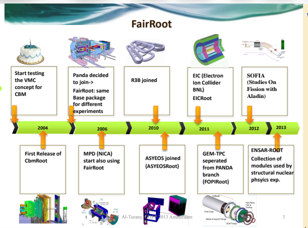


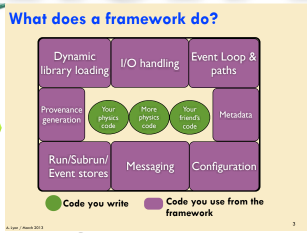




Common Frameworks

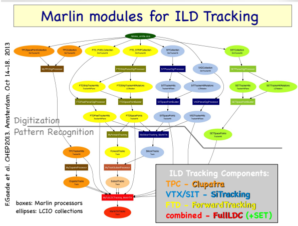


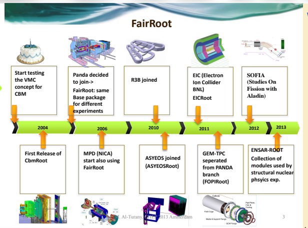


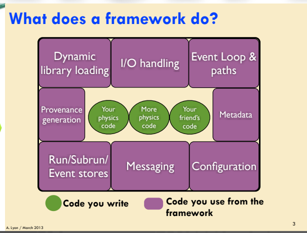


- Many different frameworks presented
  - For sure the big ones ATLAS(Athena), LHCb(Gaudi), CMS
  - Many smaller ones
    - International Large Detector@ILC(Marlin)
    - Muon g-2 (ArtG4 based on Art, lite fork from CMS)
      - NOvA, *Mu2e*, MicroBoone, LBNE, …
    - CBM, Panda (FairRoot)
      - Many other experiments meanwhile
- Other frameworks
  - Geant4

Common Frameworks







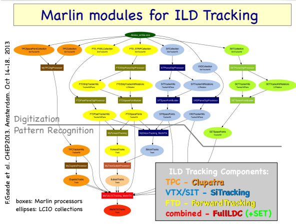
Multi-threading

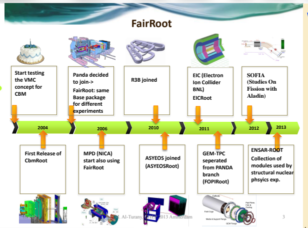
Porting applications …

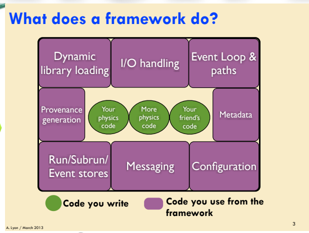
- Few changes needed in user code:
  1. Change **`main()`** to use **`G4MTRunManager`** – **one line**
  2. Create Sensitive Detector & Field in a new method
  3. Adapt to **per-event RNG seeding** (potential change)
  4. Check User 'Action' classes (Step, Track, Event)
- Choice - handling Output: per thread or accumulate ?
  - Geant4 automatically performs reductions (accumulation) if using scorers or **`G4Run`** derived classes
- Testing
  - Check output of runs – MT vs 1-thread vs Sequential

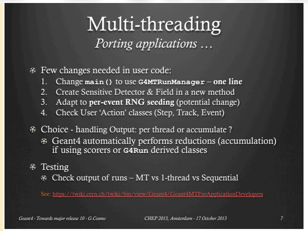
See: https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4MTForApplicationDevelopers

Common Frameworks



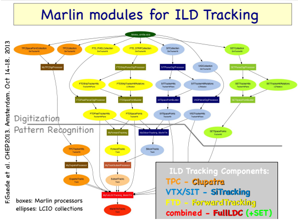


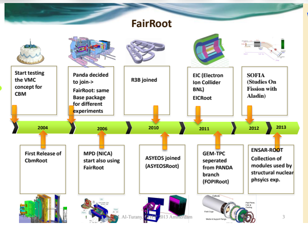


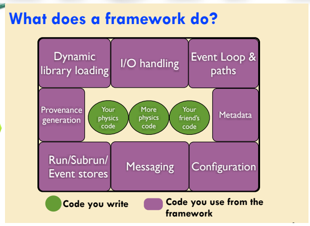


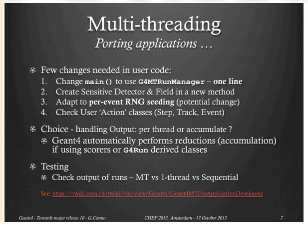
- Many different frameworks presented
  - For sure the big ones ATLAS(Athena), LHCb(Gaudi), CMS
  - Many smaller ones
    - International Large Detector@ILC(Marlin)
    - Muon g-2 (ArtG4 based on Art, lite fork from CMS)
      - NOvA, *Mu2e*, MicroBoone, LBNE, …
    - CBM, Panda (FairRoot)
      - Many other experiments meanwhile
- Other frameworks
  - Geant4
  - RooStats, RooFit

Common Frameworks





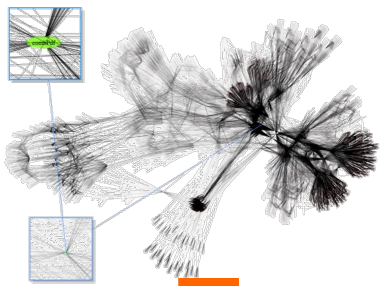




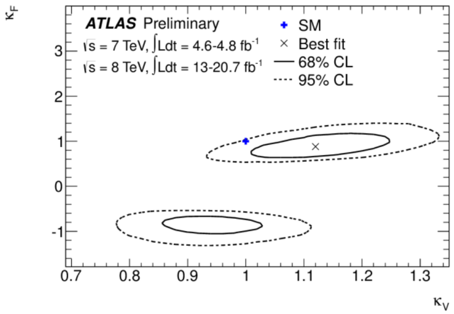
An excursion – Collaborative analyses with workspaces

- Workspaces allow to share and modify very complex analyses with very little technical knowledge required
- Example: Higgs coupling fits

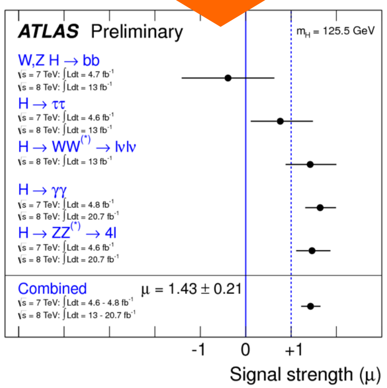
Full Higgs model



Confidence intervals on Higgs fermion, boson couplings



Signal strength in 5 channels



$$\sigma(gg \to H) * \mathrm{BR}(H \to \gamma\gamma) \sim \frac{\kappa_F^2 \cdot \kappa_\gamma^2(\kappa_F, \kappa_V)}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2}$$

$$\sigma(qq' \to qq'H) * \mathrm{BR}(H \to \gamma\gamma) \sim \frac{\kappa_V^2 \cdot \kappa_\gamma^2(\kappa_F, \kappa_V)}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2}$$

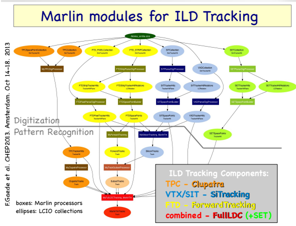
$$\sigma(gg \to H) * \mathrm{BR}(H \to ZZ^{(*)}, H \to WW^{(*)}) \sim \frac{\kappa_F^2 \cdot \kappa_V^2}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2}$$

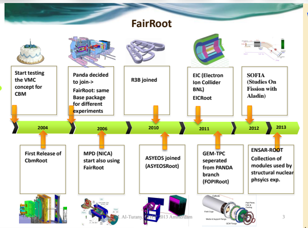
$$\sigma(qq' \to qq'H) * \mathrm{BR}(H \to ZZ^{(*)}, H \to WW^{(*)}) \sim \frac{\kappa_V^2 \cdot \kappa_V^2}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2}$$

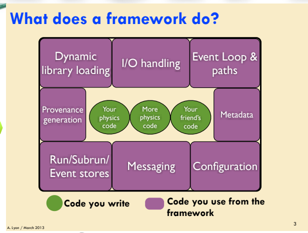
$$\sigma(qq' \to qq'H, VH) * \mathrm{BR}(H \to \tau\tau, H \to b\bar{b}) \sim \frac{\kappa_V^2 \cdot \kappa_F^2}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2},$$

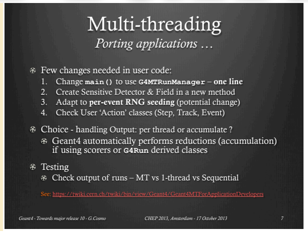
Reparametrize in terms of fermion, boson scale factors

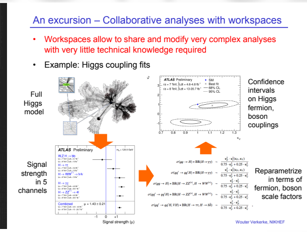
Common Frameworks









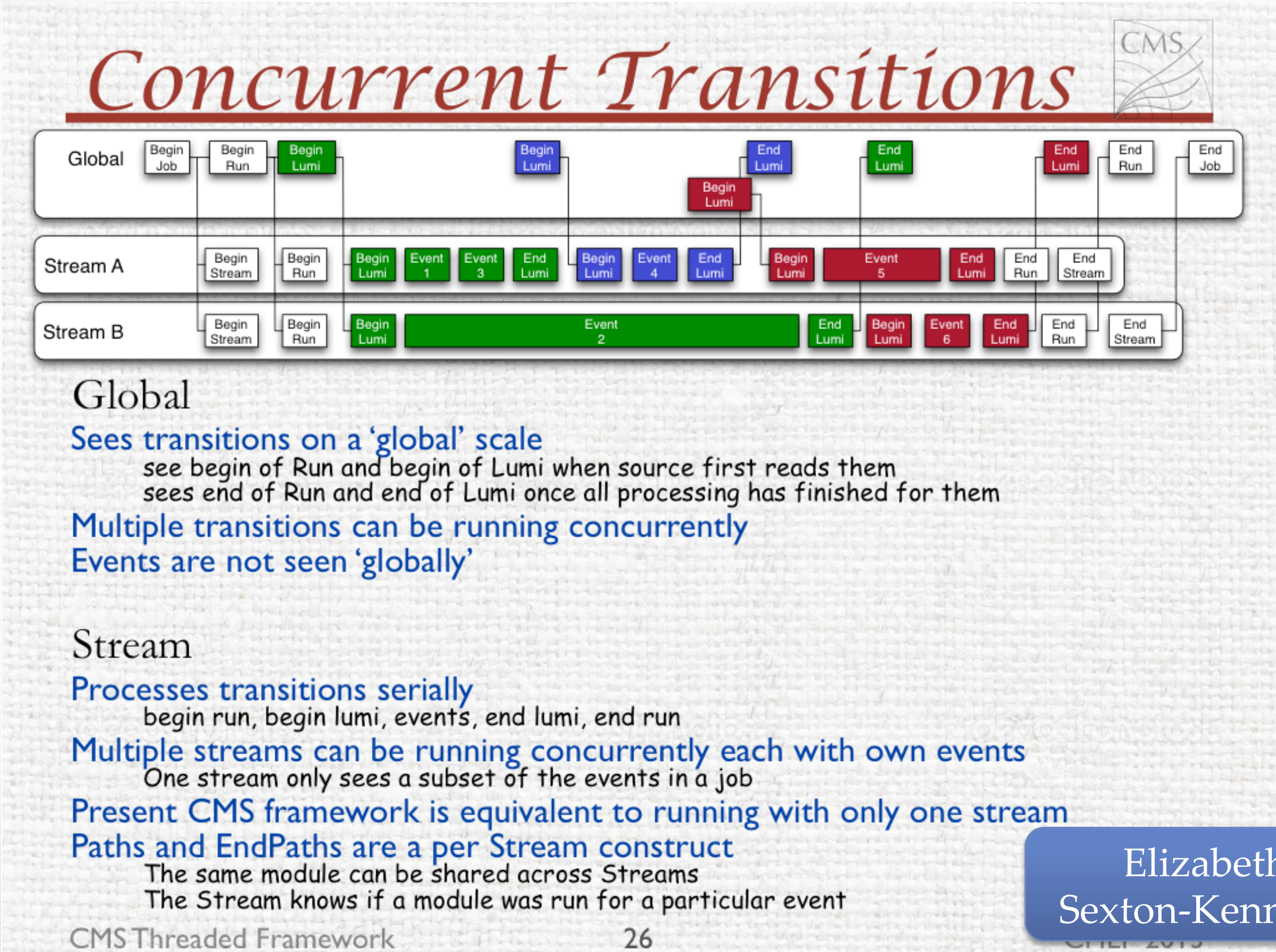


- Many different frameworks presented
  - For sure the big ones ATLAS(Athena), LHCb(Gaudi), CMS
  - Many smaller ones
    - International Large Detector@ILC(Marlin)
    - Muon g-2 (ArtG4 based on Art, lite fork from CMS)
      - NOvA, *Mu2e*, MicroBoone, LBNE, …
    - CBM, Panda (FairRoot)
      - Many other experiments meanwhile
- Other frameworks
  - Geant4
  - RooStats, RooFit
- DRY, Code reuse, Consolidation
  - Make better use of your resources (manpower, money, …)
  - More help from other users
  - Benefit from improvements done by your colleagues

Concurrency

- CHEP2013 Prediction: Lots of reports about success of deep parallelization of algorithms (Adam Lyon)
- CHEP2013: Different approaches to solve the problem
- CMS:
  - Run multiple events in parallel, within one event run multiple modules in parallel, and within one module run multiple tasks in parallel
  - Use Intel Threaded Building Blocks (TBB) for all the parallelization

Concurrency



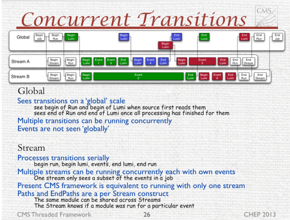
cess

on)

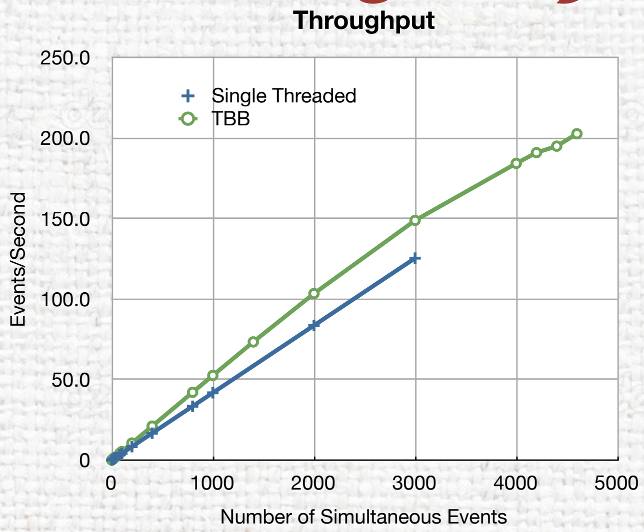
es in

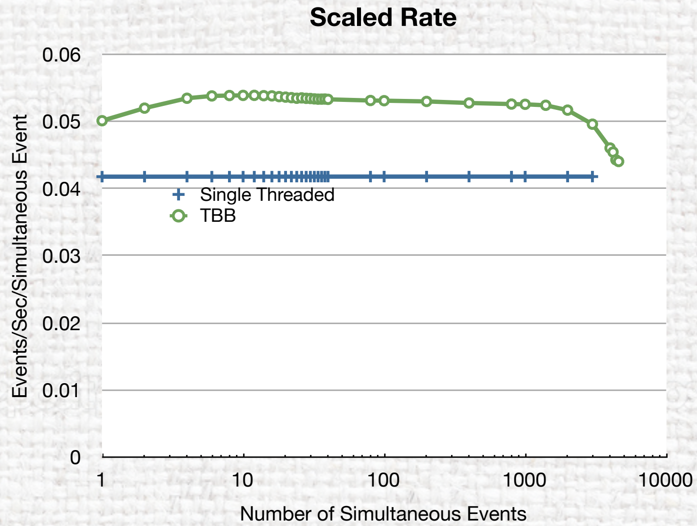
Elizabeth Sexton-Kennedy

Concurrency



Scaling: Infinite Cores





32 core AMD Opteron Processor 6128 w/ 64GB RAM

All modules are calling usleep

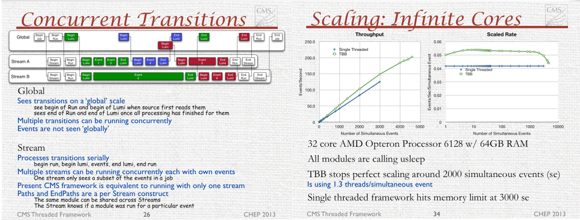
TBB stops perfect scaling around 2000 simultaneous events (se)

Is using 1.3 threads/simultaneous event

Single threaded framework hits memory limit at 3000 se

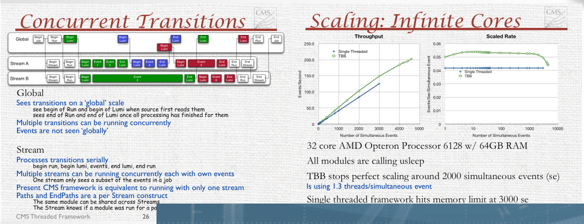
ess
n)
es in

Concurrency



- CHEP2013 Prediction: Lots of reports about success of deep parallelization of algorithms (Adam Lyon)
- CHEP2013: Different approaches to solve the problem
- CMS:
  - Run multiple events in parallel, within one event run multiple modules in parallel, and within one module run multiple tasks in parallel
  - Use Intel Threaded Building Blocks (TBB) for all the parallelization
- ATLAS:
  - Use scheduler to start task when input data is ready
  - New scheme is implemented using TBB

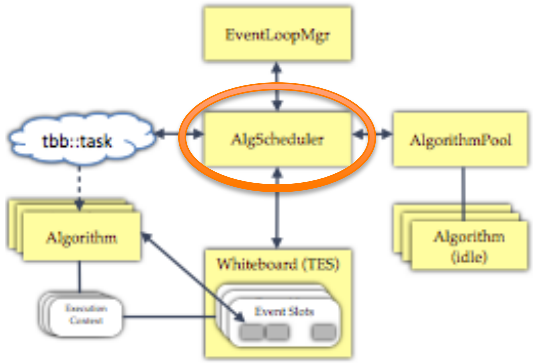
Concurrency

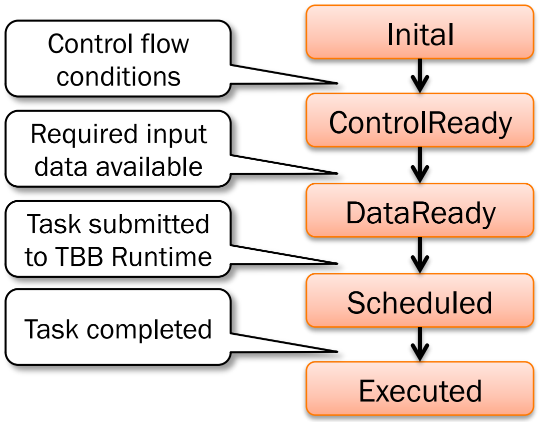


The Forward Scheduler

Keeps the state of each algorithm for each event

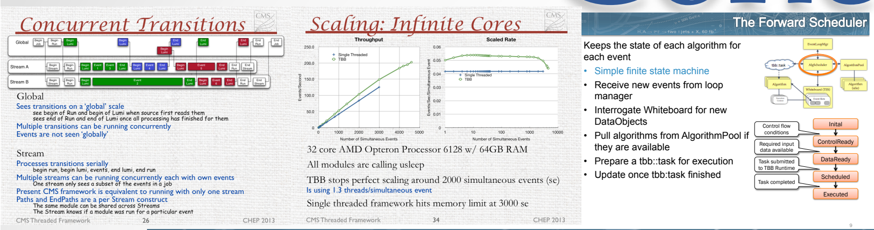
- Simple finite state machine
- Receive new events from loop manager
- Interrogate Whiteboard for new DataObjects
- Pull algorithms from AlgorithmPool if they are available
- Prepare a tbb::task for execution
- Update once tbb:task finished





cess
on)
les in

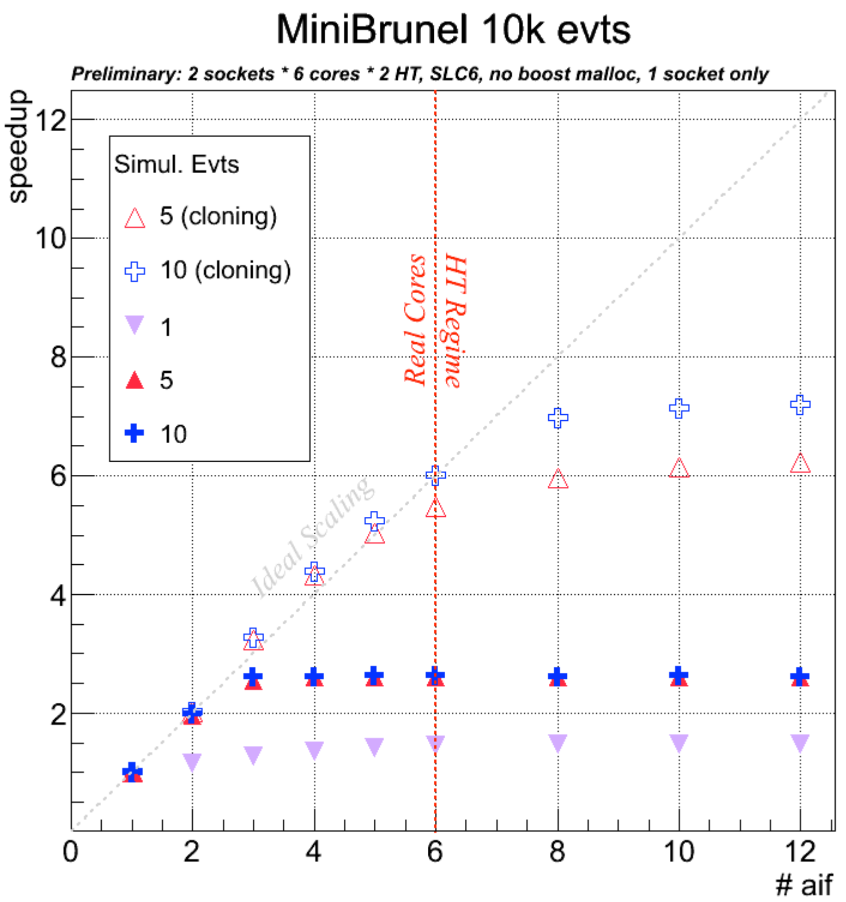
Concurrency



ccess
yon)

dules in

Scaling on One Processor

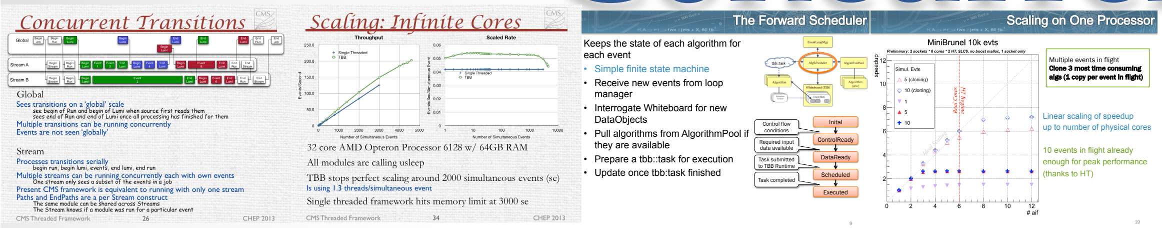


Multiple events in flight
Clone 3 most time consuming algs (1 copy per event in flight)

Linear scaling of speedup up to number of physical cores

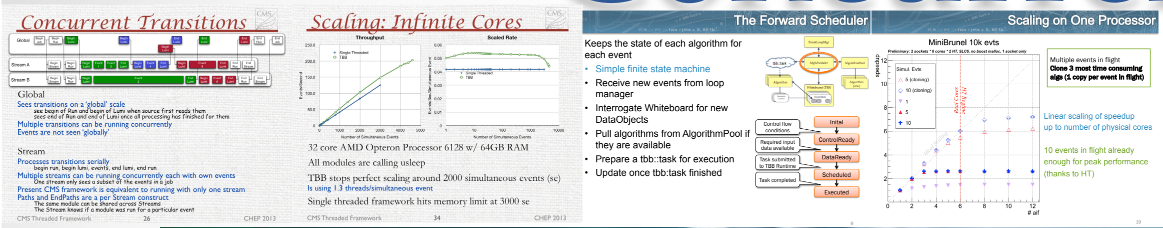
10 events in flight already enough for peak performance (thanks to HT)

Concurrency



- CHEP2013 Prediction: Lots of reports about success of deep parallelization of algorithms (Adam Lyon)
- CHEP2013: Different approaches to solve the problem
- CMS:
  - Run multiple events in parallel, within one event run multiple modules in parallel, and within one module run multiple tasks in parallel
  - Use Intel Threaded Building Blocks (TBB) for all the parallelization
- ATLAS:
  - Use scheduler to start task when input data is ready
  - New scheme is implemented using TBB
- FairRoot
  - Use Multi-Process instead of Multi-Threading
  - Communication and synchronization through message (data) exchange

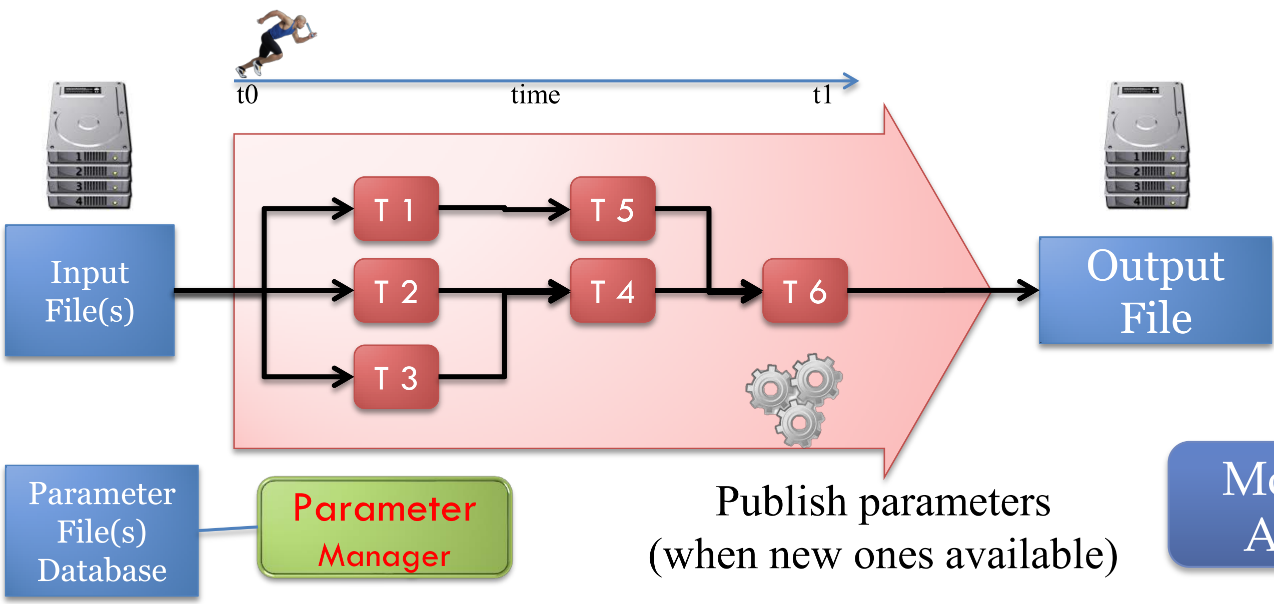
Concurrency



- s of
- blem
- in

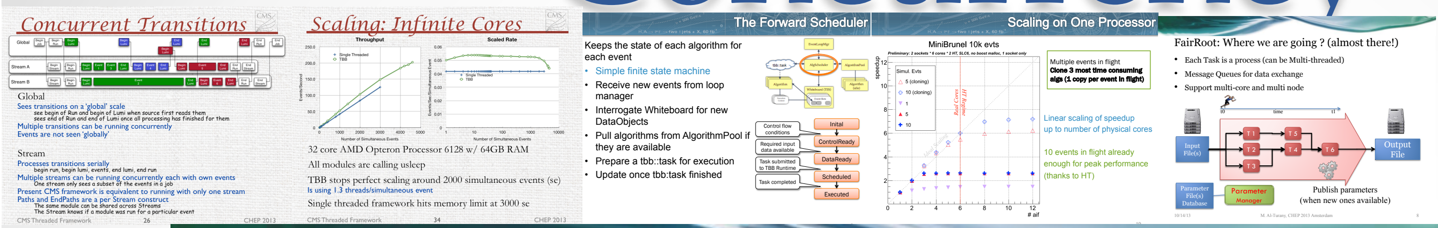
FairRoot: Where we are going ? (almost there!)

- Each Task is a process (can be Multi-threaded)
- Message Queues for data exchange
- Support multi-core and multi node



Mohammad Al-Turany

Concurrency



s of

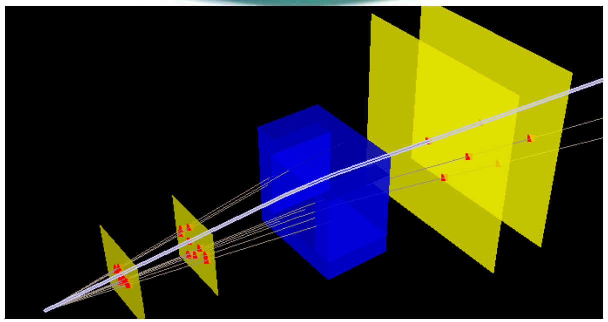
lem

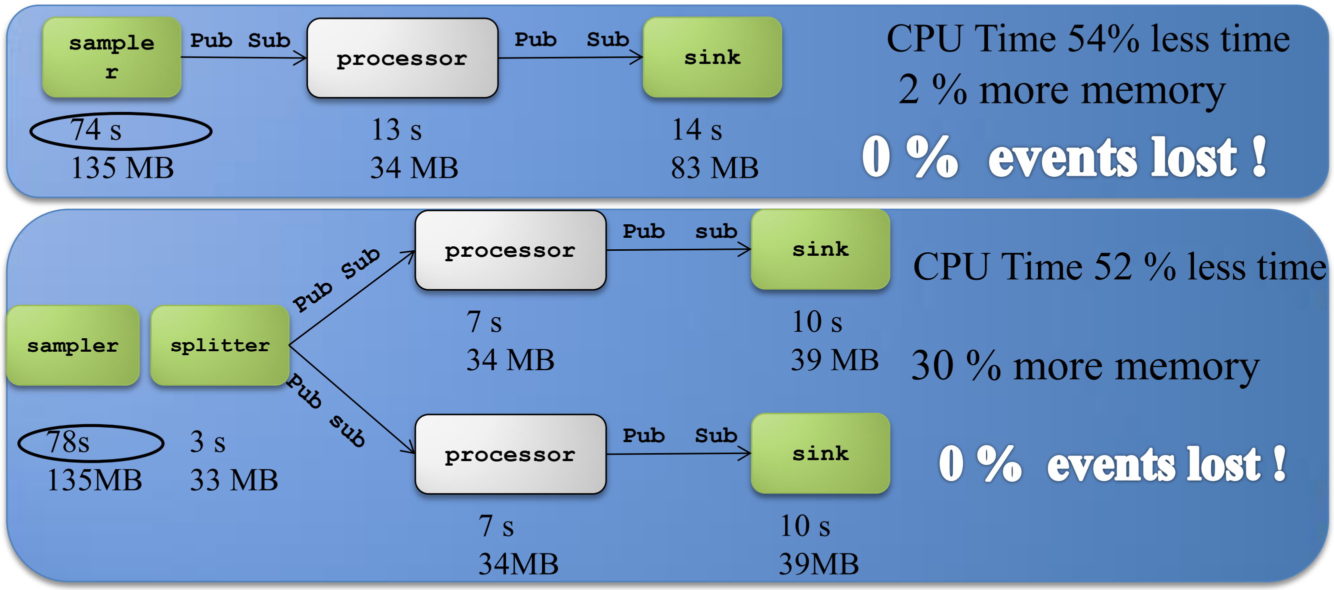
in

nge

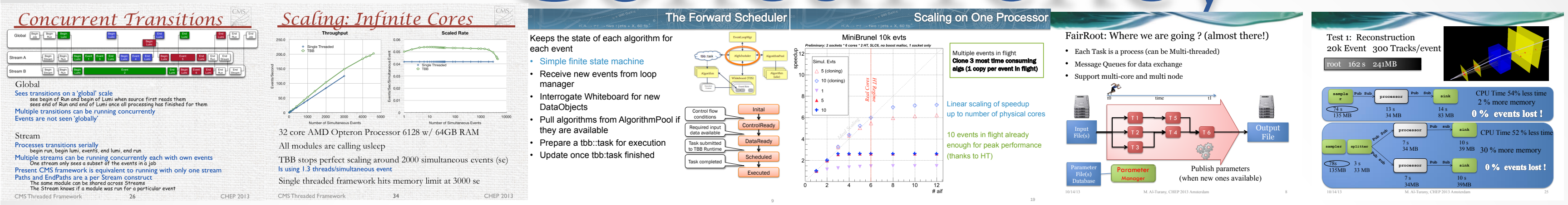
Test 1: Reconstruction
20k Event 300 Tracks/event

root 162 s 241MB





Concurrency



- CHEP2015 Prediction: Lots of reports about success of parallelization of the frameworks
- It will be interesting to compare the different implementations when they are production ready

Algorithms

- Many talks from different collaborations
- Many algorithms are very specific designed for one experiment
  - CBM: Selected event reconstruction algorithms
  - Belle II: Track extrapolation using Geant4E
  - ....
- There are also developments which should be usable for a larger user community
  - CLAS: Bayesian Data Analysis in Baryon Spectroscopy
  - PANDA: Common Partial Wave Analysis Framework
  - ....
- How to find such developments which could be (re)used?
  - Database with information?
  - Web page?
- How can we come to a situation like with common frameworks?

Pileup Simulation

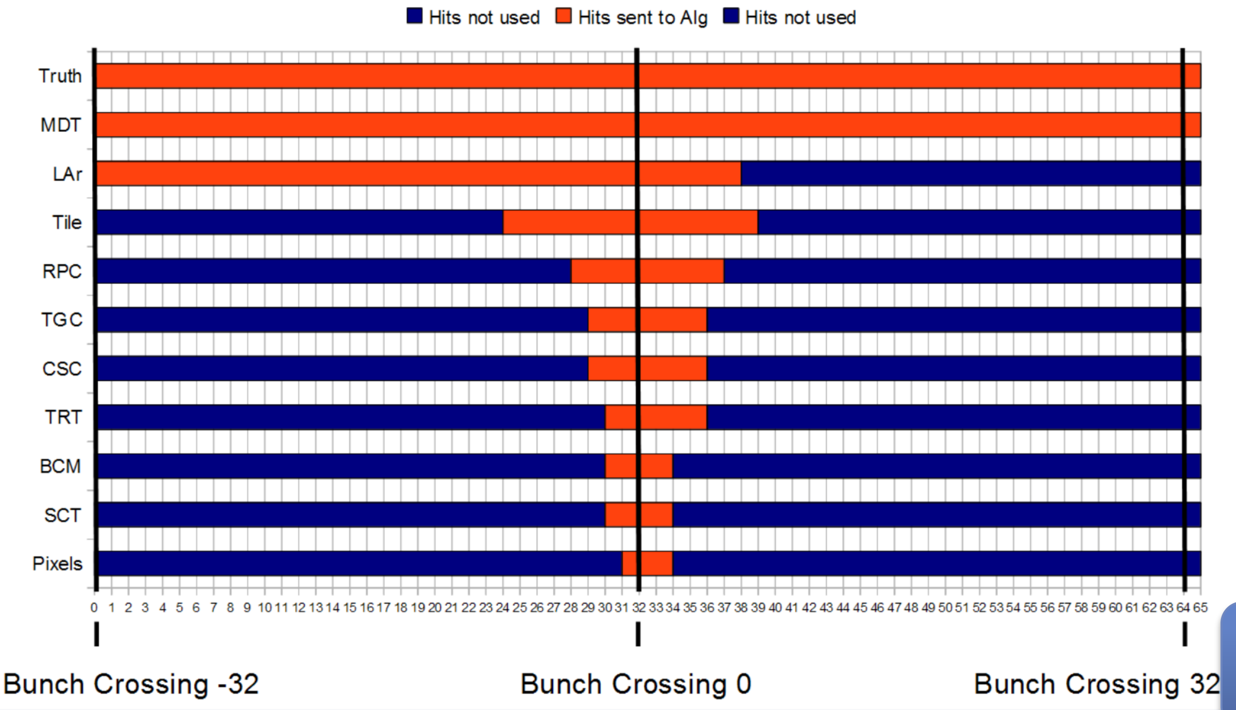
- LHC exceeded expectations of pileup (PU)up to 40 interactions / crossing (design 23)
- Simulation has to keep up
- Geant 4 predictions reached enormous precision, at cost of high CPU consumption → improve its usage
- Overlay: use data for pileup 'simulation'
- other measures, e.g. use only those out-of-time pileup events to which detectors are sensitive

Pileup Simulation

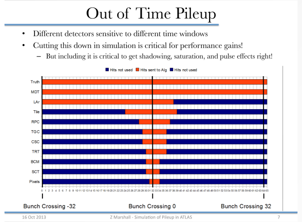
- to 40
- ion, at
  sage
- e pileup

Out of Time Pileup

- Different detectors sensitive to different time windows
- Cutting this down in simulation is critical for performance gains!
  - But including it is critical to get shadowing, saturation, and pulse effects right!

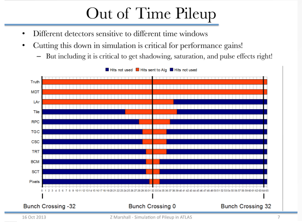


Pileup Simulation



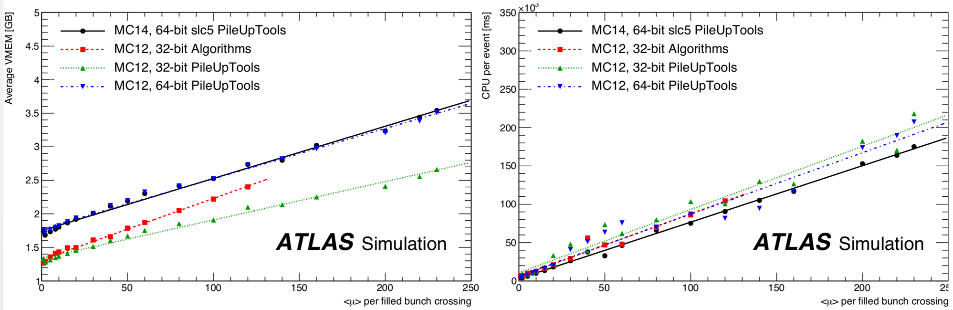
- LHC exceeded expectations of pileup (PU)up to 40 interactions / crossing (design 23)
- Simulation has to keep up
- Geant 4 predictions reached enormous precision, at cost of high CPU consumption → improve its usage
- Overlay: use data for pileup 'simulation'
- other measures, e.g. use only those out-of-time pileup events to which detectors are sensitive
- Simulations limited by CPU and/or memory
- need new ideas to reduce consumptions

Pileup Simulation



Computing Performance

- Obvious trade-off between CPU and memory
  - For high luminosity, we spend the CPU on I/O to avoid serious memory limitations ("Algorithms" → "PileUpTools")
  - For low luminosity it's possible to pay with some memory and save some CPU (32-bit → 64-bit, slc5 → slc6)
  - Memory shows much more regular growth; normal non-linear effects on CPU like changing from active memory to swap



ions of pileup (PU)up to 40
esign 23)
)
hed enormous precision, at

- simulations limited by CP
- need new ideas to reduc
-

CPU/Memory Performance

- Some timing/performance results:
  - Single neutrino events with 8 TeV Pythia minbias events simulating individual interactions
  - Events processed merely through pileup addition and digitization

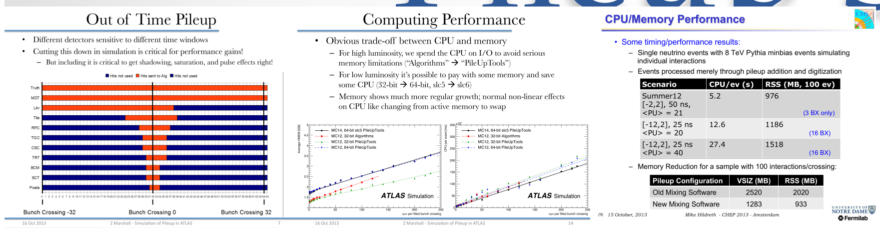
Scenario	CPU/ev (s)	RSS (MB, 100 ev)
Summer12 [-2,2], 50 ns, <PU> = 21	5.2	976 (3 BX only)
[-12,2], 25 ns <PU> = 20	12.6	1186 (16 BX)
[-12,2], 25 ns <PU> = 40	27.4	1518 (16 BX)

  - Memory Reduction for a sample with 100 interactions/crossing:

Pileup Configuration	VSIZ (MB)	RSS (MB)
Old Mixing Software	2520	2020
New Mixing Software	1283	933

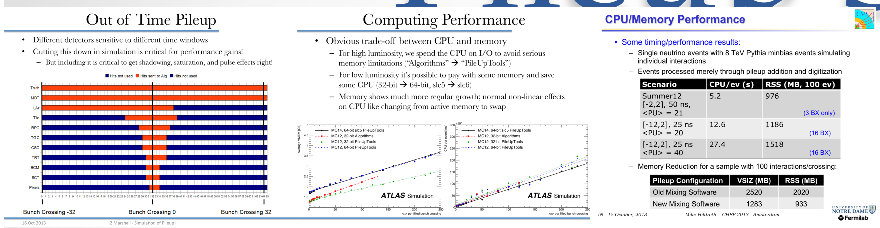
University of Notre Dame
Fermilab

Pileup Simulation



- LHC exceeded expectations of pileup (PU)up to 40 interactions / crossing (design 23)
- Simulation has to keep up
- Geant 4 predictions reached enormous precision, at cost of high CPU consumption → improve its usage
- Overlay: use data for pileup 'simulation'
- other measures, e.g. use only those out-of-time pileup events to which detectors are sensitive
- Simulations limited by CPU and/or memory
- need new ideas to reduce consumptions
- Premixing of events

Pileup Simulation

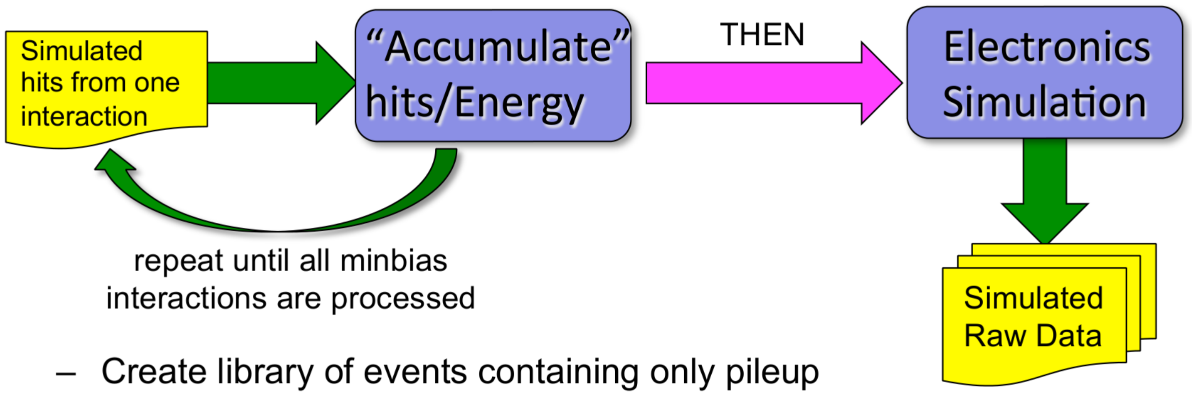


Simulation meets Computation

Even if the events are read sequentially, it still will require more than 2000 minbias events to produce a single MC event with appropriate pileup at sLHC luminosities

- nightmare for computing infrastructure if huge minbias event files have to be made available to each compute note for MC production

- Potential Solution: "Pre-Mixing"
  - For the pure minbias pileup simulation,



  - Create library of events containing only pileup contributions, following pre-determined luminosity profile to calculate how many interactions to include

to 40

sion, at

usage

e pileup

Mike Hildreth

Simulation at LHC in Future

- ATLAS/CMS in run-1 produced several billion MC events, even more will be needed in run-2 with higher luminosity & trigger output rates
- ATLAS / CMS investigated in speeding up simulation of their detectors
- frozen showers; fast parameterized simulations
- ATLAS also worries about simulating events under old conditions (trigger simulation)
- One way to reduce CPU time: simulate not all particles: Russian Roulette
    - Now also employed for calorimeters

Simulation at LHC in Future

Russian Roulette CPU Usage

- Comparison of CPU performance between 8 TeV and 14 TeV Simulation:

Events	Energy (TeV)	No RR	RR=10	Energy (TeV)	No RR	RR=10
MinBias	8	19.3s	15.2s 78.5%	14	21.5s	16.1.s 74.2%
Z→e+e-	8	50.9s	33.4s 65.6%	14	116.9s	92.3s 78.7%
ttbar	8	87.1s	52.8s 60.6%	14	115.8s	74.3s 62.4%

← % of default

Only n and γ are biased in ECAL and HCAL; RR Factor 10 is used

At 14 TeV, Zee becomes compatible in CPU with TTbar. Similar RR effects on

CMS software version CMSSW_6_2_0

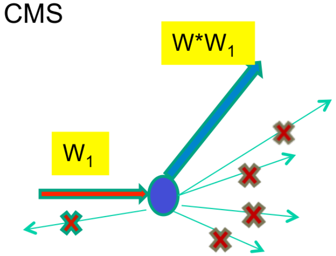
ced several billion MC
needed in run-2 with
output rates
in speeding up simulation

old conditions (trigge

- One way to reduce C
particles: Russian Rou
  - Now also employed for calc

Russian Roulette (RR) in CMSSW

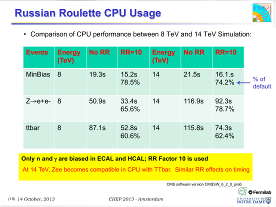
- Method used in neutron shielding calculations for many years
  - Not necessary to track all low-energy particles in a shower
- Some fraction of low-energy particles are killed but remainder get higher weight
  - not suited for tracker, muon systems
  - direct CPU savings (for calorimeter simulation)
  - geometry independent
- RR may be enabled separately per particle type and detector region
  - n, γ - allow significant CPU savings for CMS
  - p, e⁻ - no visible effect so far
- Two parameters per particle
  - RR factor (1/W)
  - Upper energy limit



Mike Hildreth

Fermilab
UNIVERSITY OF NOTRE DAME

Simulation at LHC in Future



Russian Roulette CPU Usage

- Comparison of CPU performance between 8 TeV and 14 TeV Simulation:

Events	Energy (TeV)	No RR	RR=10	Energy (TeV)	No RR	RR=10
MinBias	8	19.3s	15.2s 78.5%	14	21.5s	16.1.s 74.2%
Z→e+e-	8	50.9s	33.4s 65.6%	14	116.9s	92.3s 78.7%
ttbar	8	87.1s	52.8s 60.6%	14	115.8s	74.3s 62.4%

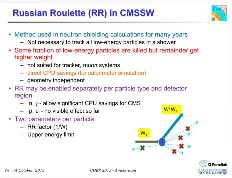
% of default

Only n and γ are biased in ECAL and HCAL; RR Factor 10 is used

At 14 TeV, Zee becomes compatible in CPU with TTbar. Similar RR effects on timing

CMS software version CMSSW_6_2_0_pre6

(18) 14 October, 2013 CHEP 2013 - Amsterdam



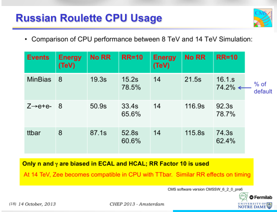
Russian Roulette (RR) in CMSSW

- Method used in neutron shielding calculations for many years
  - Not necessary to track all low-energy particles in a shower
- Some fraction of low-energy particles are killed but remainder get higher weight
  - not suited for tracker, muon systems
  - direct CPU savings (for calorimeter simulation)
  - geometry independent
- RR may be enabled separately per particle type and detector region
  - n, γ - allow significant CPU savings for CMS
  - p, e- - no visible effect so far
- Two parameters per particle
  - RR factor (1/W)
  - Upper energy limit

(9) 14 October, 2013 CHEP 2013 - Amsterdam

- ATLAS/CMS in run-1 produced several billion MC events, even more will be needed in run-2 with higher luminosity & trigger output rates
- ATLAS / CMS investigated in speeding up simulation of their detectors
- frozen showers; fast parametrized simulations
- ATLAS also worries about simulating events under old conditions (trigger simulation)
- One way to reduce CPU time: simulate not all particles: Russian Roulette
  - Now also employed for calorimeters
- Idea of a Integrated Simulation Framework
  - CPU can be reduced by up to factor of 3000
  - Then digitization and reconstruction becomes bottleneck → need also fast digi + reco !

Simulation at LHC in Future



Russian Roulette CPU Usage

- Comparison of CPU performance between 8 TeV and 14 TeV Simulation:

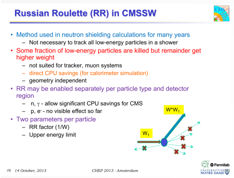
Events	Energy (TeV)	No RR	RR=10	Energy (TeV)	No RR	RR=10
MinBias	8	19.3s	15.2s 78.5%	14	21.5s	16.1.s 74.2% ← % of default
Z→e+e-	8	50.9s	33.4s 65.6%	14	116.9s	92.3s 78.7%
ttbar	8	87.1s	52.8s 60.6%	14	115.8s	74.3s 62.4%

Only n and γ are biased in ECAL and HCAL; RR Factor 10 is used

At 14 TeV, Zee becomes compatible in CPU with TTbar. Similar RR effects on timing

CMS software version CMSSW_6_2_0_pre6

(18) 14 October, 2013 CHEP 2013 - Amsterdam



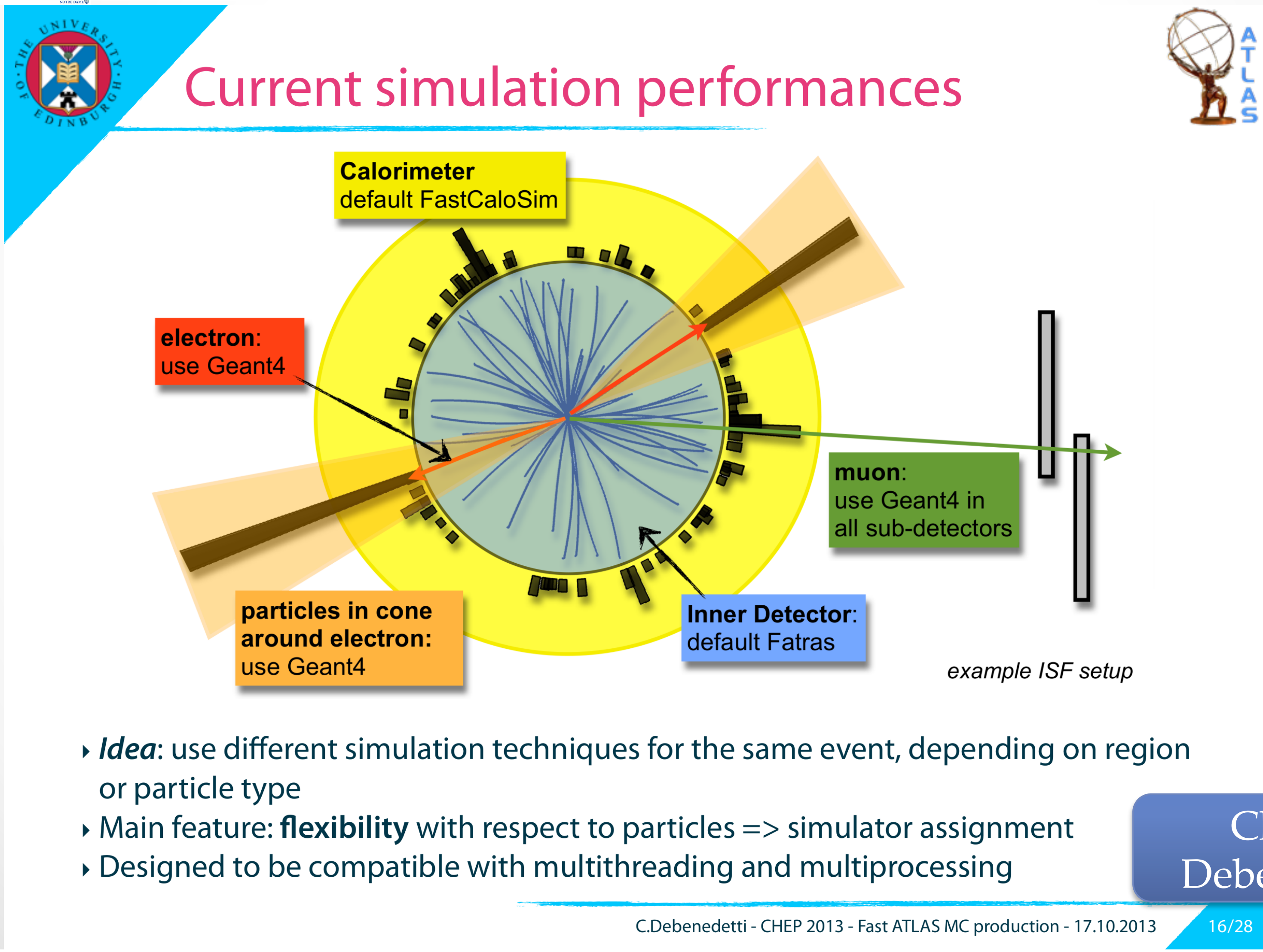
Russian Roulette (RR) in CMSSW

- Method used in neutron shielding calculations for many years
  - Not necessary to track all low-energy particles in a shower
- Some fraction of low-energy particles are killed but remainder get higher weight
  - not suited for tracker, muon systems
  - direct CPU savings (for calorimeter simulation)
  - geometry independent
- RR may be enabled separately per particle type and detector region
  - n, γ - allow significant CPU savings for CMS
  - p, e⁻ - no visible effect so far
- Two parameters per particle
  - RR factor (1/W)
  - Upper energy limit

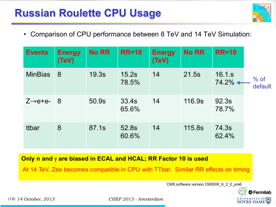
W*W₁

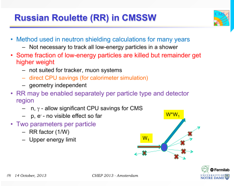
W₁

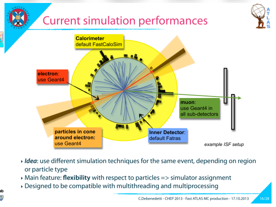
(9) 14 October, 2013 CHEP 2013 - Amsterdam



Simulation at LHC in Future







ISF performance: H -> gamma gamma

ISF simulation setup	Speedup	Accuracy
Full Geant4	1	best possible
Geant4 with FastCaloSim	~25	approximated calorimeter
Fatras with FastCaloSim	~750	all subdetectors approximated
Fatras with FastCaloSim simulate only particles in cones around photons	~3000	all subdetectors approximated event simulated only partially

gg→H→γγ no pileup

- Use of fast simulation => **significant speedup**
- Speed increased even further thanks to partial event simulation
  - helps in reducing output size

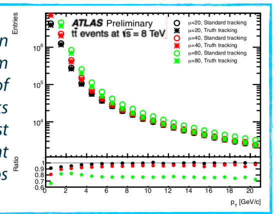
...ced several billion MC events,
...l in run-2 with higher luminosity

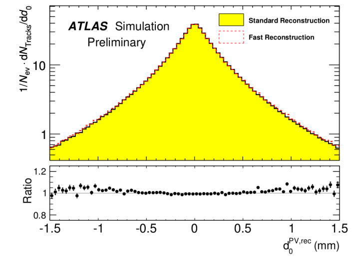
...n speeding up simulation of

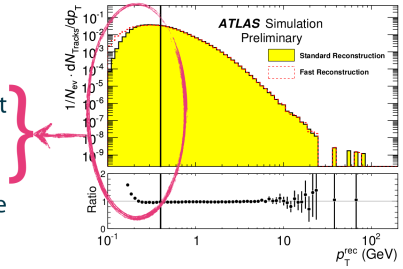
- Now also employed for calorimete...
- Idea of a Integrated Simul...
  - CPU can be reduced by up to fac...
  - Then digitization and reconstructic... + reco !
-

Fast reconstruction: performance

8 TeV tt Simulation Transverse momentum spectrum of reconstructed tracks with standard and fast tracking for different pileup scenarios







8 TeV minimum bias Simulation

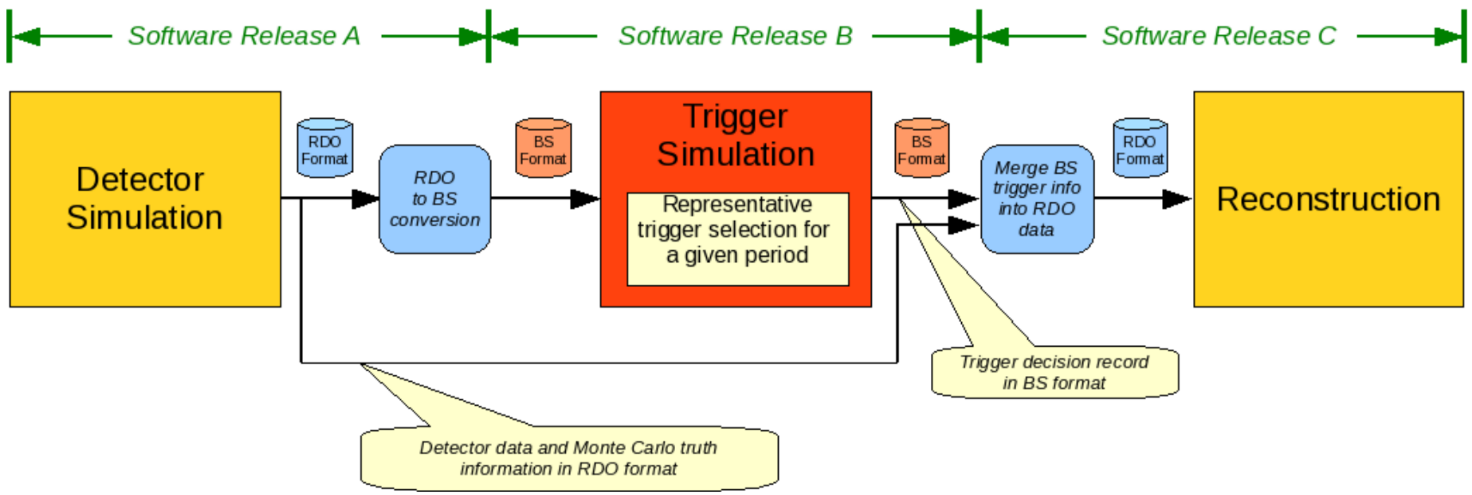
- Good agreement with standard Reconstruction
- Significant speedup
- Difference at low momentum not significant
  - p_T>400 MeV for standard ATLAS data and MC processing
- Fast reconstruction with better performance
  - inefficiency factor taken into account for low p_T particles

Everything else

- What about simulating old data?

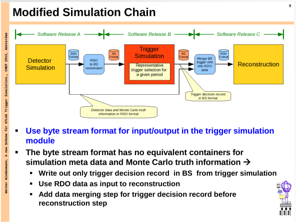
Everything else

Modified Simulation Chain



- **Use byte stream format for input/output in the trigger simulation module**
- **The byte stream format has no equivalent containers for simulation meta data and Monte Carlo truth information →**
  - **Write out only trigger decision record in BS from trigger simulation**
  - **Use RDO data as input to reconstruction**
  - **Add data merging step for trigger decision record before reconstruction step**

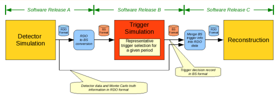
Everything else



- What about simulating old data?
- Speeding up the reconstruction.

Everything else

Modified Simulation Chain



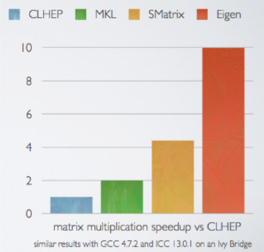
- **Use byte stream format for input/output in the trigger simulation module**
- **The byte stream format has no equivalent containers for simulation meta data and Monte Carlo truth information →**
  - **Write out only trigger decision record in BS from trigger simulation**
  - **Use RDO data as input to reconstruction**
  - **Add data merging step for trigger decision record before reconstruction step**

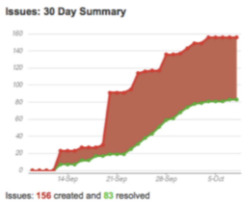
ATLAS, Technische Universität München

…ng old data?
…onstruction.

Efforts: Library Change

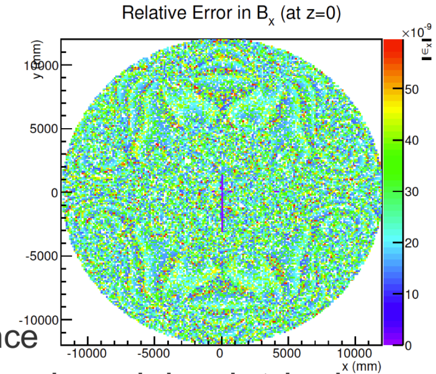
- Change from CLHEP to Eigen
  - Huge software migration
    - O(1000) packages affected
  - Eigen library functions can vectorize if compiled accordingly
- Exchanging the allocator
- Exchanging GNU libm
  - Under investigation: VDT, libimf





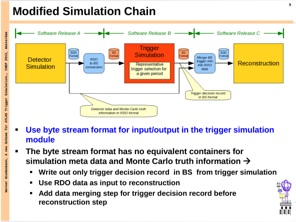
Efforts: Magnetic Field

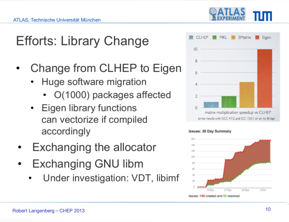
- Change from Fortran77 to C++
  - Code a lot more readable now
  - Reduced function call depth
- Adding field value cache
  - Greatly affects performance as particles are traced along their trajectory
- Unit Conversion Minimization
  - Affects accuracy and performance
- Make code autovectorizable and applying intrinsics
- Speed-up of 20% (reco) up to 60 % (single particle simulation)

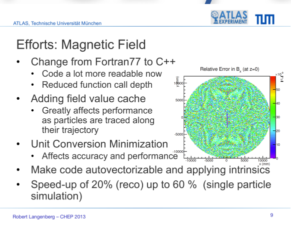


Robert Langenberg

Everything else

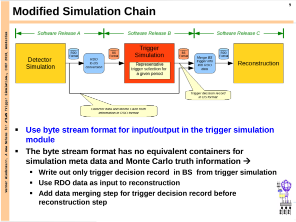


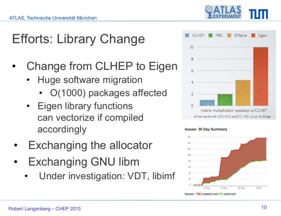


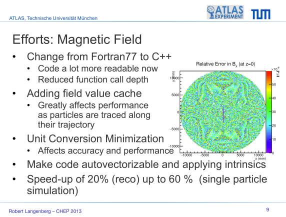


- What about simulating old data?
- Speeding up the reconstruction.
- Geant4 in hadron therapy

Everything else

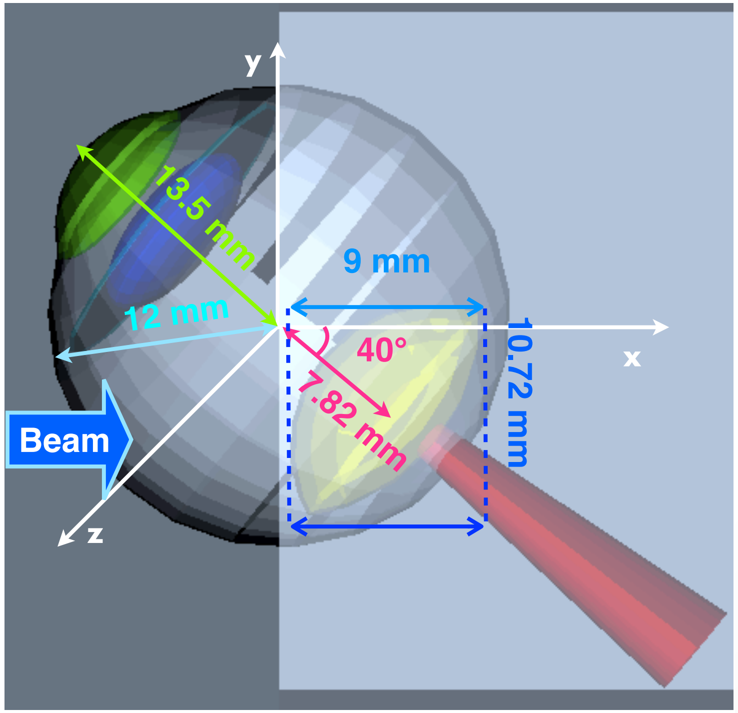




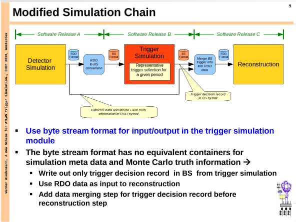


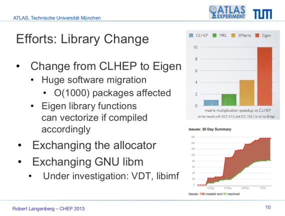
The eye detector

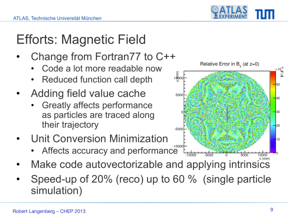
- Eye anatomy deeply studied and a geometric schematization realized
- Accurate reproduction of all eye-components in the G4 simulation
- Dimensions parameterised as a function of the sclera radius
- Rotation possible to misalign tumour and sensitive sub-components

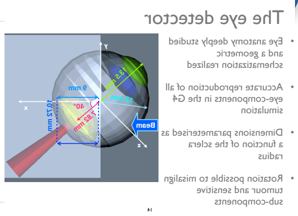


Everything else



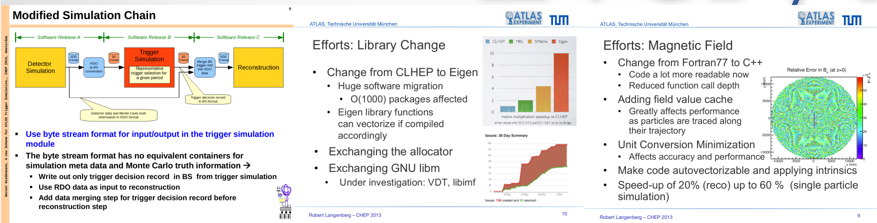






- What about simulating old data?
- Speeding up the reconstruction.
- Geant4 in hadron therapy
- LUCID

Everything else



data?

tion.

A brief history of LUCID

In 2008, the **Simon Langton Grammar School for Boys** entered a satellite experiment design competition run by the British National Space Centre (now **UK Space Agency**) and **Surrey Satellite Technology Limited** (SSTL).

- The Langton Ultimate Cosmic ray Intensity Detector (LUCID) would use Timepix detectors, developed by the Medipix Collaboration, to measure the space radiation environment in Low Earth Orbit.
- Designed by students, built by SSTL, now due to launch in February 2014.
- LUCID now part of CERN@school, supported by UK Science and Technology Facilities Council (STFC) Large Award ST/J000256/1.

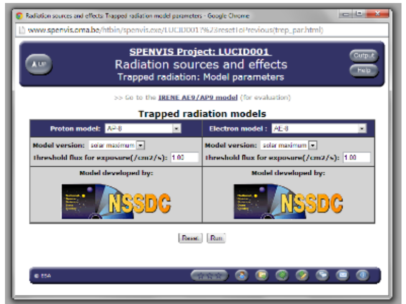
Particle source(s)

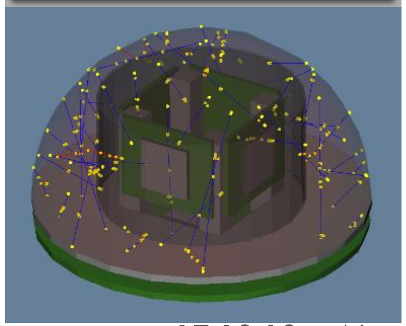
SPENVIS

- ESA-backed "Space Environment Information System" web portal.
- *Spacecraft coordinate generators*:
  - Input LUCID orbit details.
- *Trapped radiation models*:
  - AP-8 for protons and electrons;
  - Int. and diff. flux spectra.

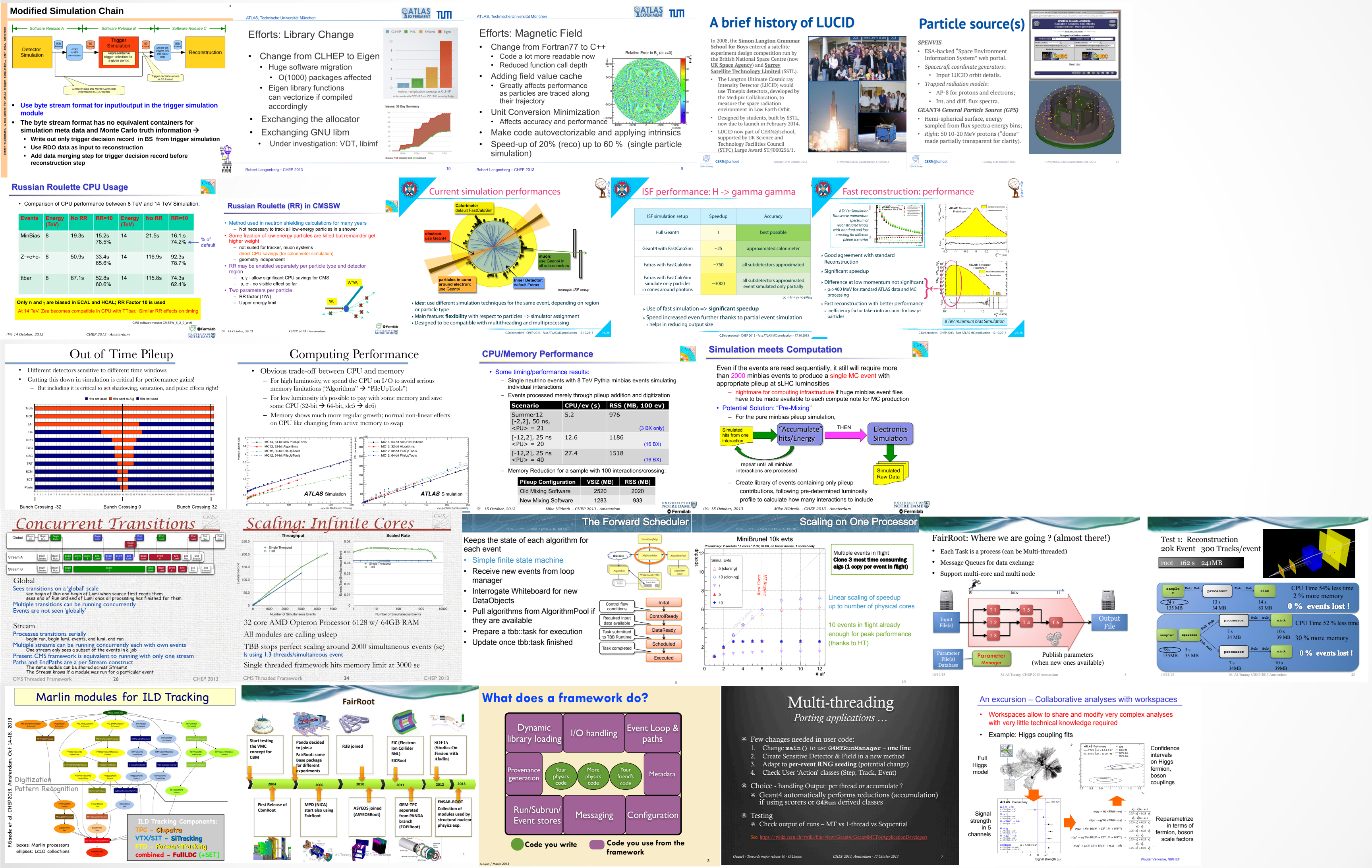
GEANT4 General Particle Source (GPS)

- Hemi-spherical surface, energy sampled from flux spectra energy bins;
- *Right*: 50 10-20 MeV protons ("dome" made partially transparent for clarity).





Thank you


Modified Simulation Chain
Efforts: Library Change
Efforts: Magnetic Field
A brief history of LUCID
Particle source(s)
Russian Roulette CPU Usage
Russian Roulette (RR) in CMSSW
Current simulation performances
ISF performance: H -> gamma gamma
Fast reconstruction: performance
Out of Time Pileup
Computing Performance
CPU/Memory Performance
Simulation meets Computation
Concurrent Transitions
Scaling: Infinite Cores
The Forward Scheduler
Scaling on One Processor
FairRoot: Where we are going ? (almost there!)
Marlin modules for ILD Tracking
FairRoot
What does a framework do?
Multi-threading
Porting applications …
An excursion – Collaborative analyses with workspaces