Concepts for fast large scale Monte Carlo Production for the ATLAS experiment

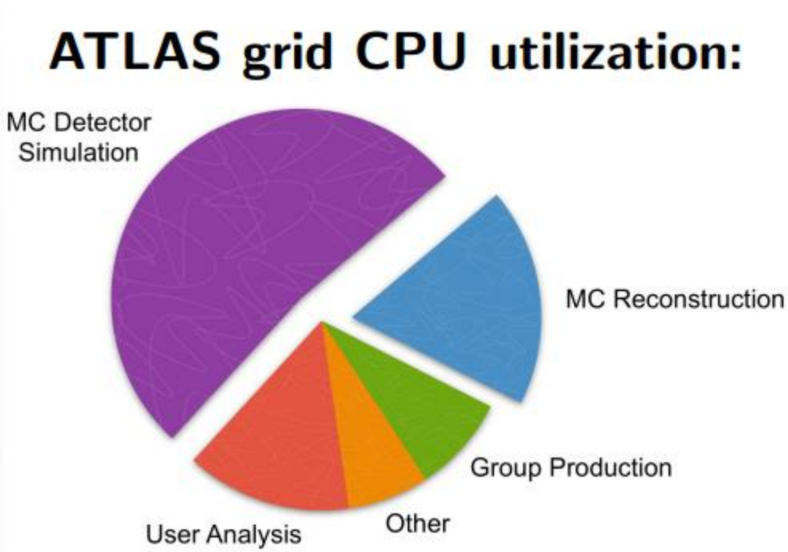
Paper written by C Debenedetti

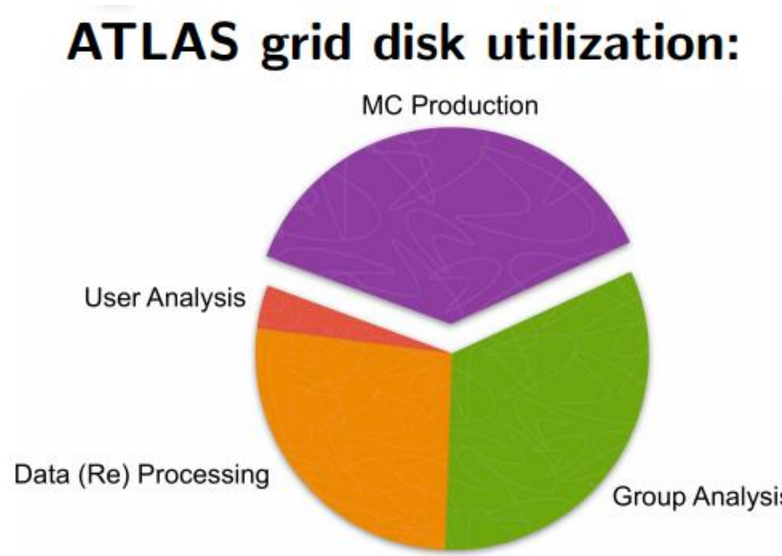
Outline

- Why Fast Simulation Monte Carlo
- Current Monte Carlo Production
- About Fast Simulation
- Integrated Simulation Framework
- Other Steps
- Conclusion

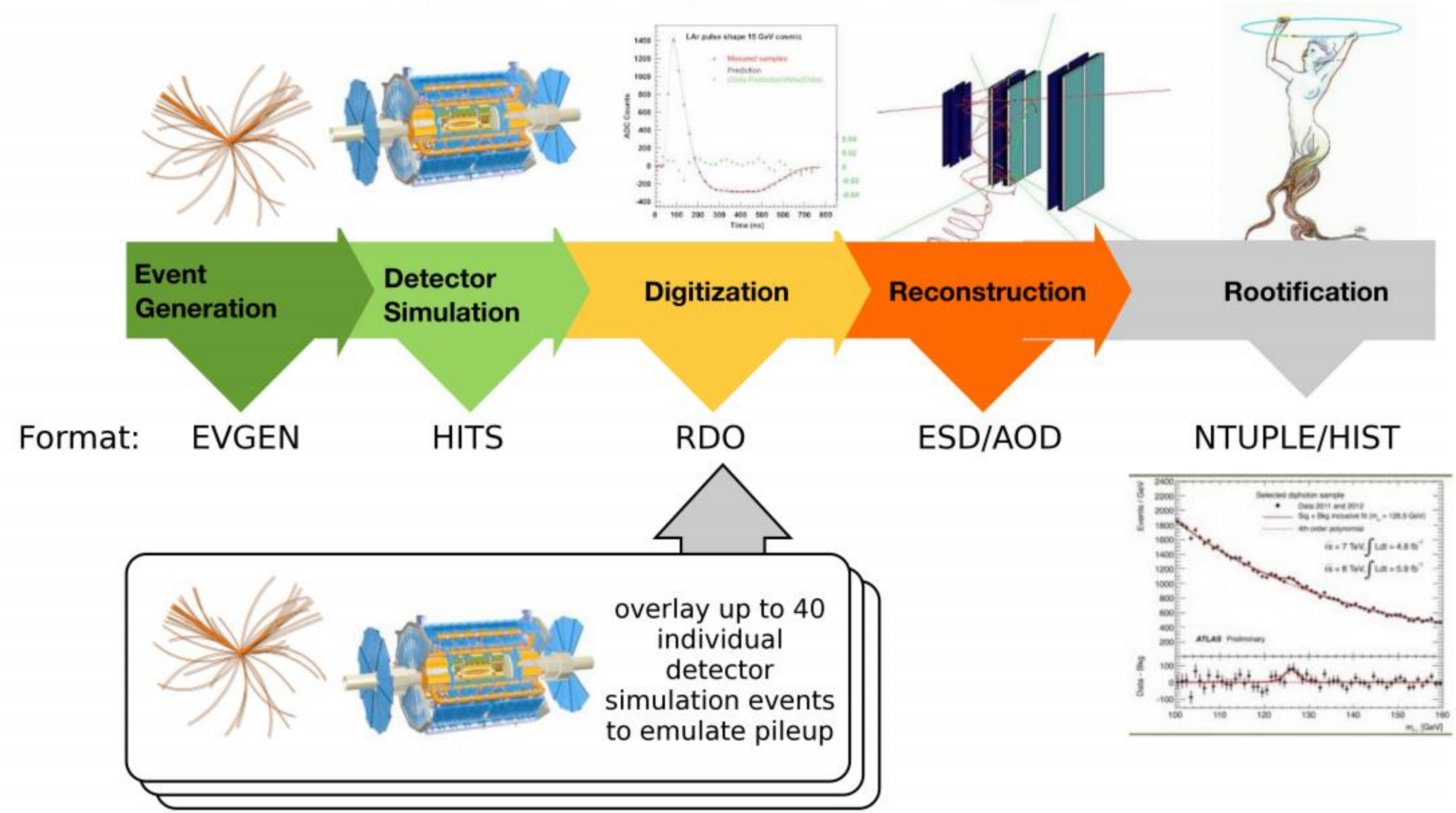
Why Fast Simulation Monte Carlo?

- MC production dominates Grid CPU usage
- Takes up large amounts of disk space on the grid
- Higher luminosity and pileup -> larger MC production needed





Current Production Chain



Detector Simulation

- Geant4
  - Most accurate, complete simulation.
- Frozen Showers
  - Uses pre-computed shower developments
- FastCaloSim
  - Parametrized calorimeter simulation
- Fatras
  - Fast Tracking engine, two orders of Magnitude faster than Geant4
- ATLFAST
  - Fully parametric description of the ATLAS detector
    - Directly produces Physics object output

Integrated Simulation Framework

- Allows for combination of different techniques
- Dynamic
- Allows for focus on certain particles of interest
- Flexibility allows increased speed while maintaining accuracy.

ISF Scheme:

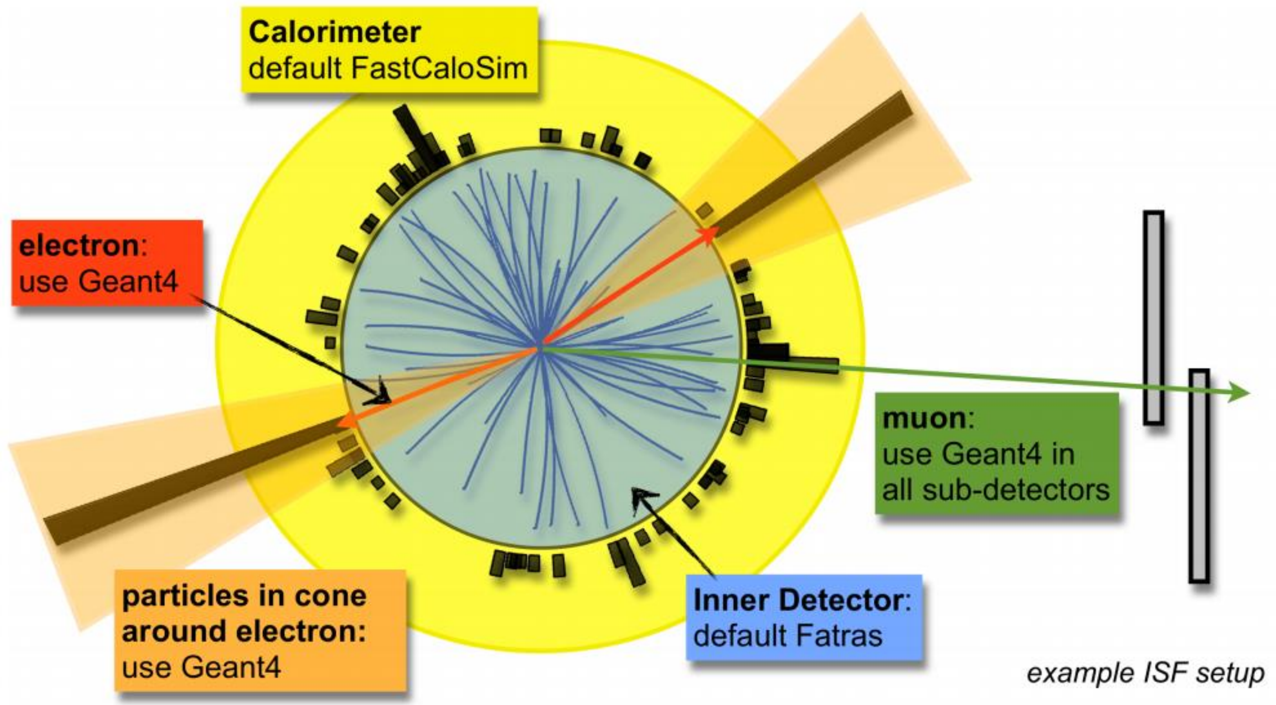


Photo Credit:

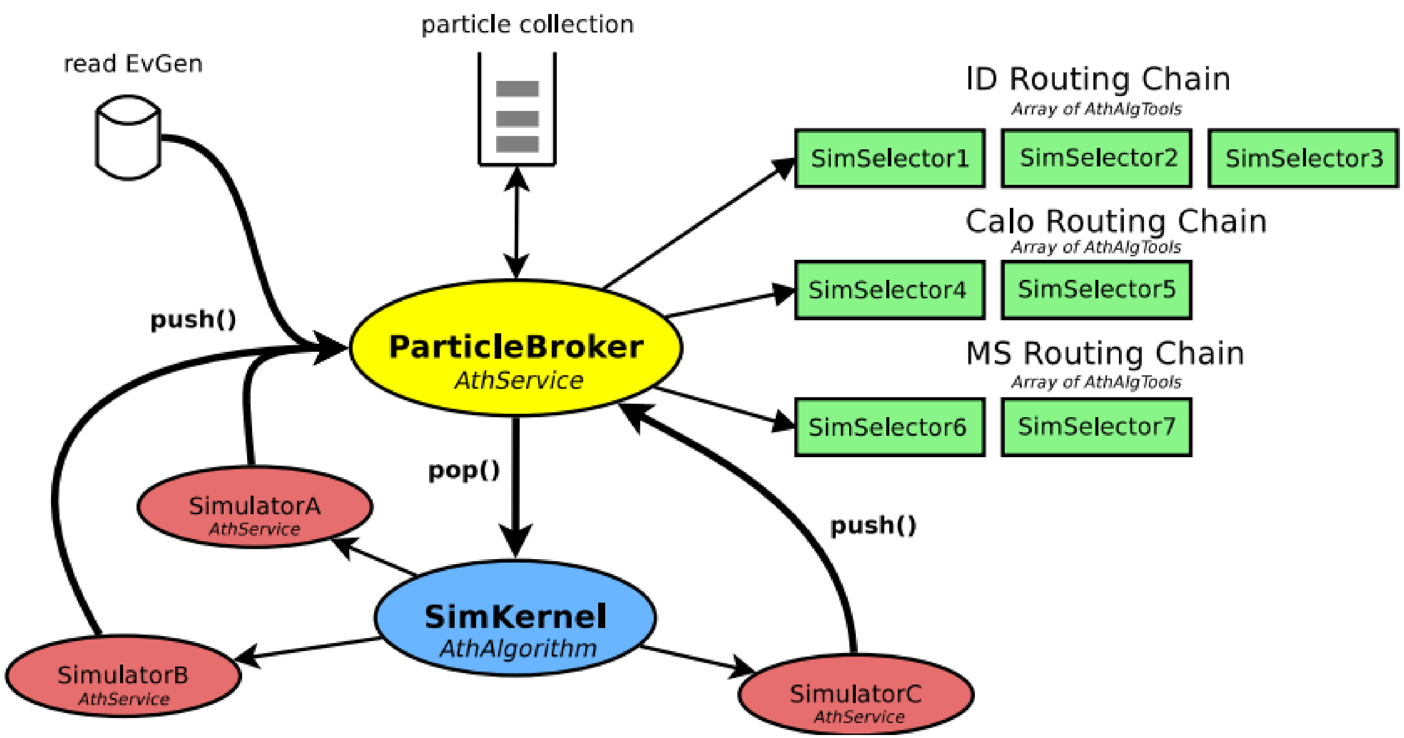
Comparison of Flavors

ISF Simulation Setup	Speedup	Accuracy
Full Geant4	1	Best Possible
Geant4 with FastCaloSim	~25	Approximated calorimeter
Fatras with FastCaloSim	~750	All sub-detector approximated
Fatras with FastCaloSim, only simulating particles inside cones around photons	~3000	All sub-detectors approximated and partial event simulated

$gg \to H \to \gamma\gamma$ no pileup

ISF Details

- SimulationKernel
  - Responsible for Particle loop, sends particle to different simulators
- ParticleBroker
  - Chooses which simulator to use for each particle, and particle storage
    - Uses Routing Chains: One for each subdetector



Routing

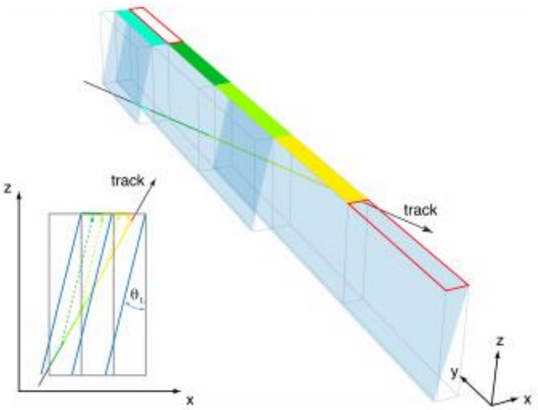
- Needed to keep track of all particle
- Static Routing
  - Based on particle type or kinematic features
- Dynamic Routing
  - Based on specific particle and the surrounding region

Fast Digitization

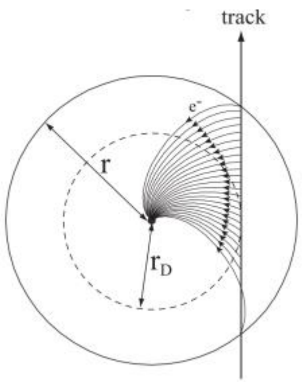
- Main bottle neck for current MC is in detector simulation
  - Digitization and reconstruction are next
- Current Conversion of HITS => Detector Readouts + pileup treatment
  - High accuracy
  - Independent for each sub-detector technology
  - Causes high CPU usage in Inner Detector

Fast Digitization for Silicon and TRT

- Fast approach for silicon:
  - Particle path lengths -> readout signals

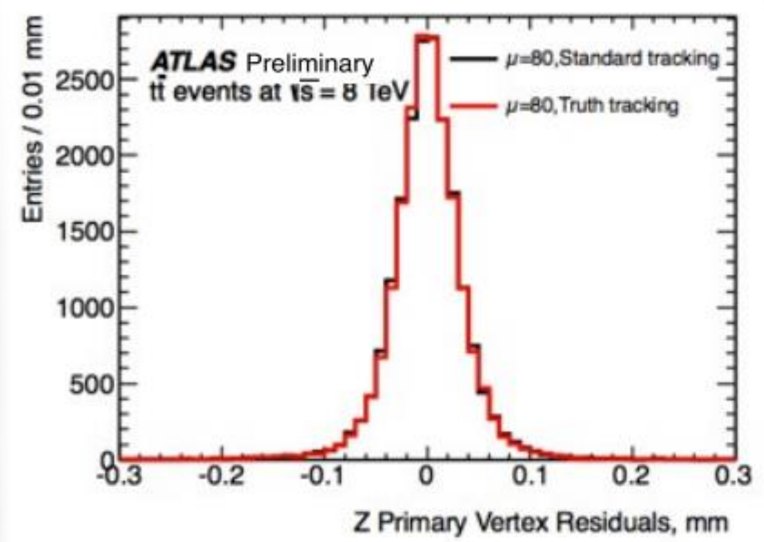


- Transition Radiation Tracker (TRT)
  - Calculated closest approach radius -> drift radius
  - Response in parameterized allowing for particle Identification



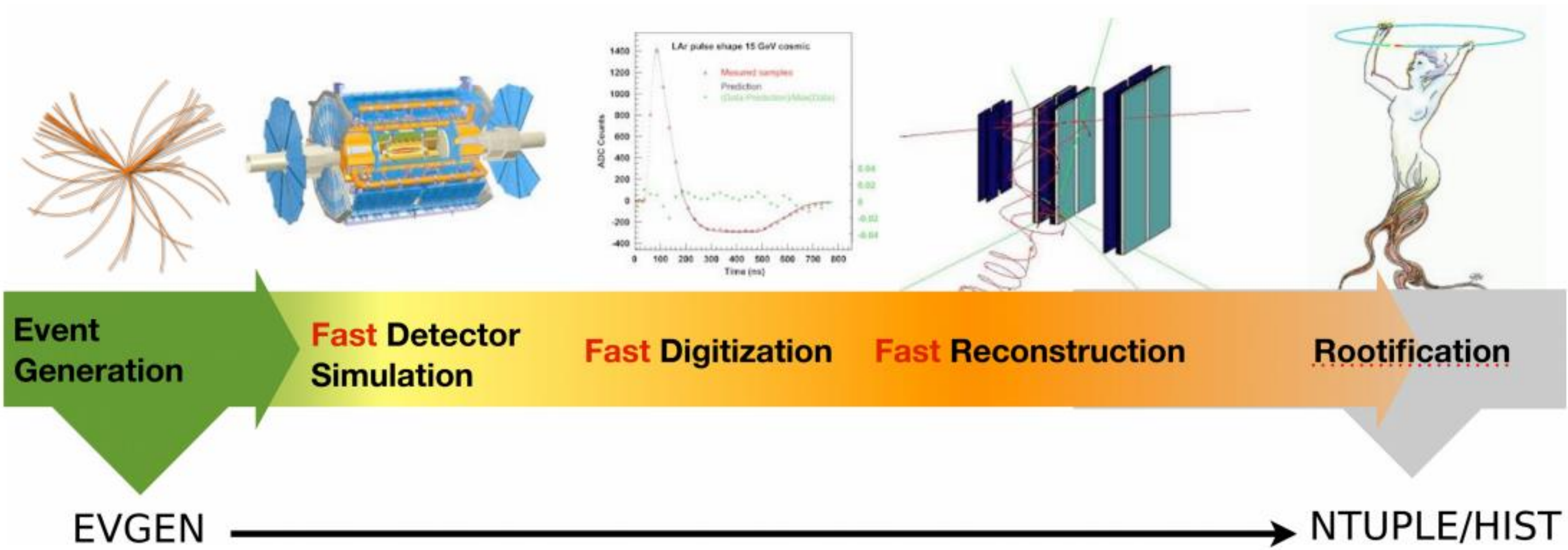
Fast Reconstruction Techniques

- Tracking is the most time consuming part of reconstruction
  - Combinatorics with pattern recognition
- Truth-seeded tracking algorithm
  - Uses truth information from Simulation step
  - Skips most time consuming steps
    - Pattern recognition
    - Track seeding



Comparison of the longitudinal primary vertex resolution using tracks found with the standard tracking and truth tracking at μ=80, where μ represents the average number of collisions per bunch crossing.

Fast MC Production



- Only two file formats
  - EVGEN to ROOT form directly
  - I/O is the next bottleneck for the system
  - No Intermediate output
- Estimated time per event = a few seconds

Conclusion

- Monte Carlo consumes a very large portion of CPU power
  - Fast Simulation decreases CPU power and increases Efficiently
- Integrated Simulation Framework
  - Dynamic use of simulation technology
- Fast digitization for Silicon and transition radiation tracking technology
- Fast reconstruction based in truth-seeded algorithms

Redesigned Fast Production:

- Single Step from Generated events to ROOT files
- Fast and easily manageable simulation
- One a few seconds needed per event

Thank You