# Simulations: A case study

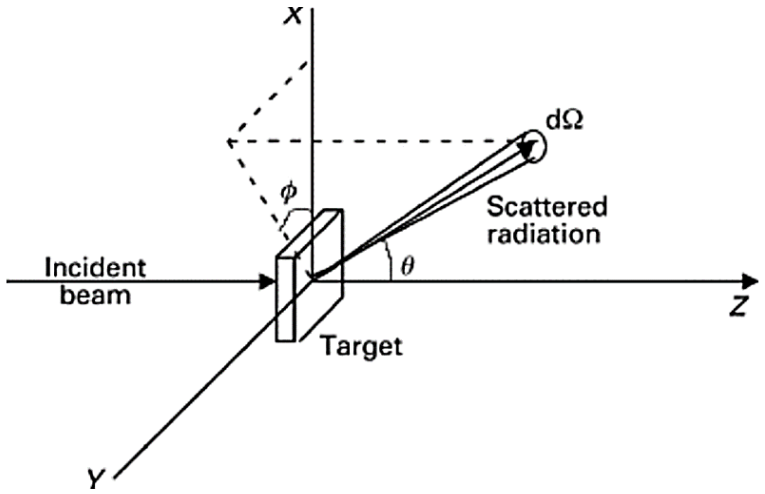
## FPGA accelerated Monte Carlo

Marco Barbone
m.barbone19@imperial.ac.uk

Imperial College London

# Elastic scattering

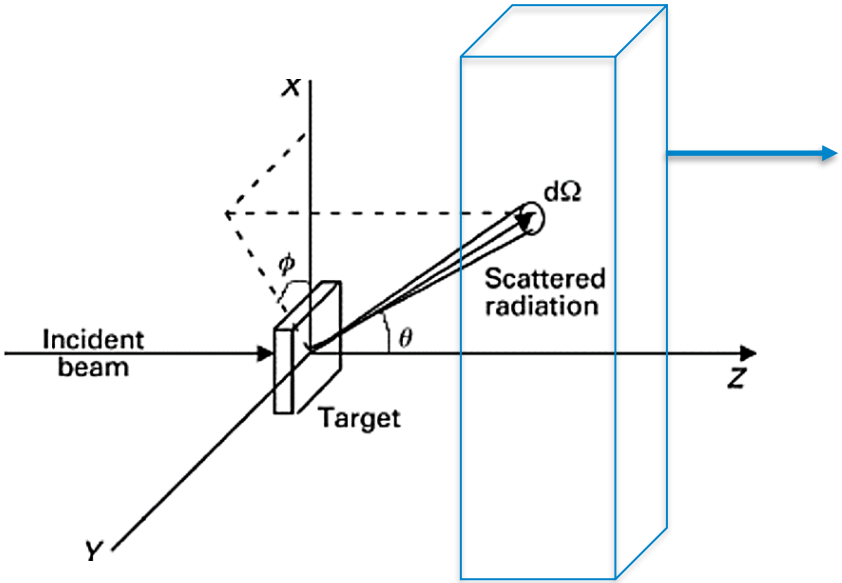
A simulation of e-/e+ transport considering only elastic scattering as possible interactions described by scattering of spin-less e-/e+ on an exponentially screened Coulomb potential




Source: Kotlarchyk M. Scattering theory. Encyclopedia of Spectroscopy and Spectrometry, 1999.

Imperial College London

# Monte Carlo Simulation



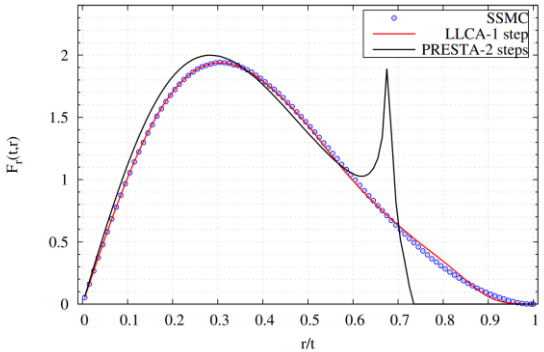

Scattered radiation final position

Many particles are simulated to achieve statistical significance

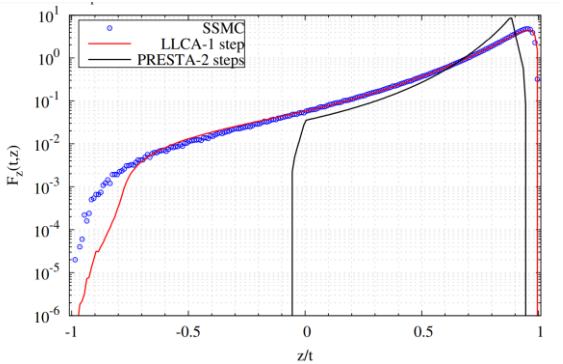
Source: Kotlarchyk M. Scattering theory. Encyclopedia of Spectroscopy and Spectrometry, 1999.

Imperial College London

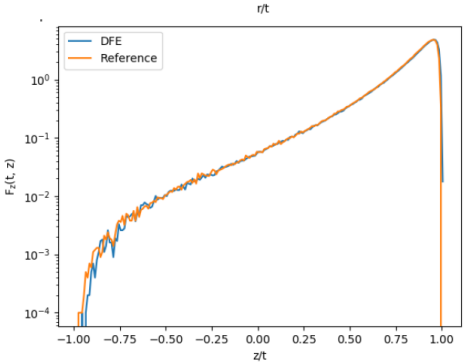
# Histograms




Traverse distribution




Longitudinal distribution




FPGA and Reference

Imperial College London

# Test configurations

Hardware:

- AMD Ryzen 5900x 12-cores 3.7GHz (4.8GHz) CPU
- Xilinx's Alveo U200 Data Center accelerator card Xilinx VU9P FPGA

Toolchain:

- MaxCompiler 2021.1
- Vivado 2019.2
- OpenMP

Imperial College London

# Test configurations (continued)

Simulation configuration:

- 100M histories
- 6 MeV beam
- Water

The FPGA clock frequency was limited to 200MHz

Imperial College London

# Performance measurements

The FPGA accelerated version is:

- ?? faster than single-core implementation
- ?? faster than multi-core implementation (12-cores)
- ?? cost-equivalent speedup

Imperial College London

# FPGA acceleration

Imperial College London

# Workload selection

Not all workloads benefits from FPGA acceleration!

FPGA have higher throughput than CPUs (and GPUs) if the workload:

- Has many branch mispredictions
- Has many cache faults
- (optional) Embarrassingly parallel
- (optional) Independency

Note: in the case of L1 trigger, latency is more important than throughput

Imperial College London

# Methodology

No agile development! Compilation might take days

A methodology is needed to minimize unneeded compilations:

1. Application analysis
2. Performance and resource modelling
3. Decide the acceleration target
4. Model the target in software
5. FPGA programming

Imperial College London

# Workload partitioning -- Decide the acceleration target

Determine the acceleration potential accounting for the major factors that dictate performance when using compute co-processors:

- Bandwidths and latencies of the various interconnects
- The capacities of different memories
- Impact of data movements
- Accelerator specific properties, e.g., for FPGA
  - Hardware/arithmetic operations space
  - Clock frequency
  - Fine-grain tunable numeric operations and their impact
  - Other custom computing approaches, e.g., compression

Accelerator compute capabilities

Imperial College London

# FPGA Performance

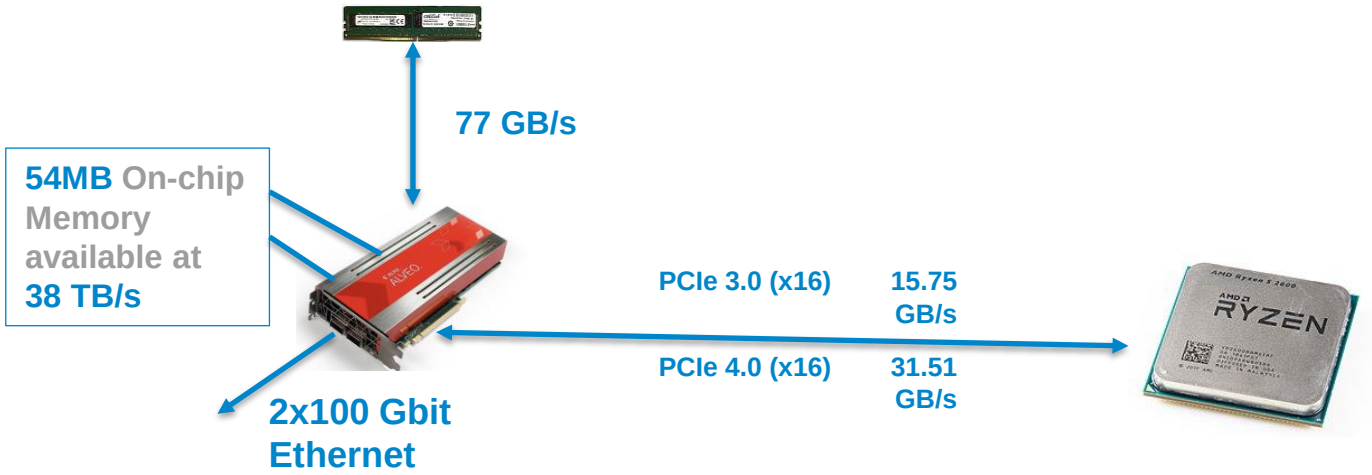
- FPGA Performance is predictable
- There is no context switch, garbage collector or any background process
- The bitstream will be executed the same number of clock cycles every time
- The number of clock cycles needed can be computed easily

Further read: Nils Voss et al. (2021), On Predictable Reconfigurable System Design

https://doi.org/10.1145/3436995

Imperial College London

# System Architecture




https://en.wikipedia.org/wiki/DDR4_SDRAM https://en.wikipedia.org/wiki/List_of_AMD_Ryzen_processors https://www.xilinx.com/products/boards-and-kits/alveo/u250.html

Imperial College London

# Monte Carlo

Imperial College London

# Workload selection

Not all workload benefits from FPGA acceleration!

FPGA have higher throughput than CPUs (and GPUs) if the workload:

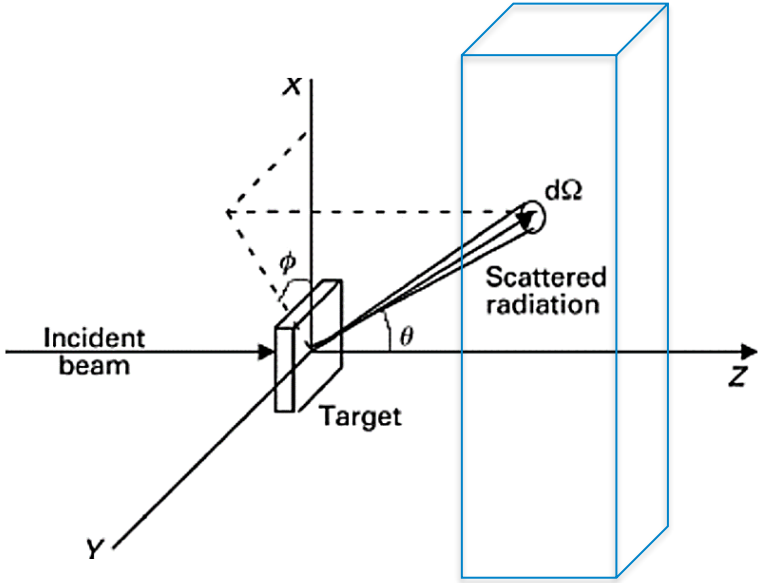
- Has many branch mispredictions
- Has many cache faults
- (optional) Embarrassingly parallel
- (optional) Independency

→ Monte Carlo fits all!

Note: in the case of L1 trigger, latency is more important than throughput

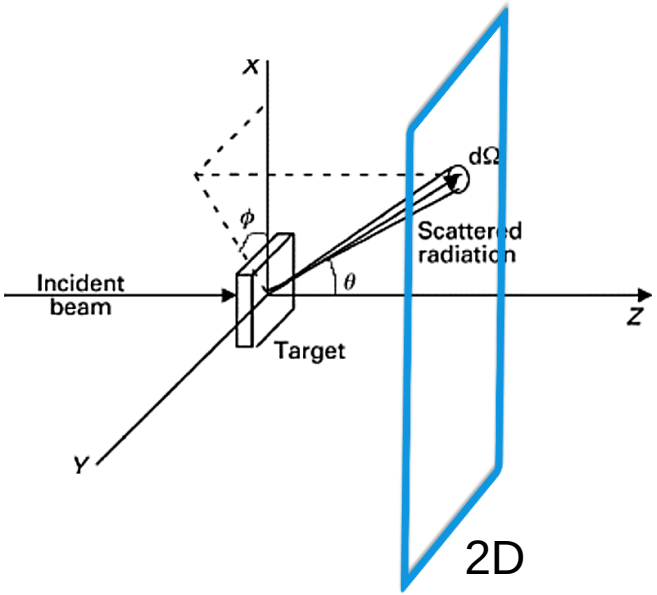
Imperial College London

# Minimize data storage




3D

Reduction

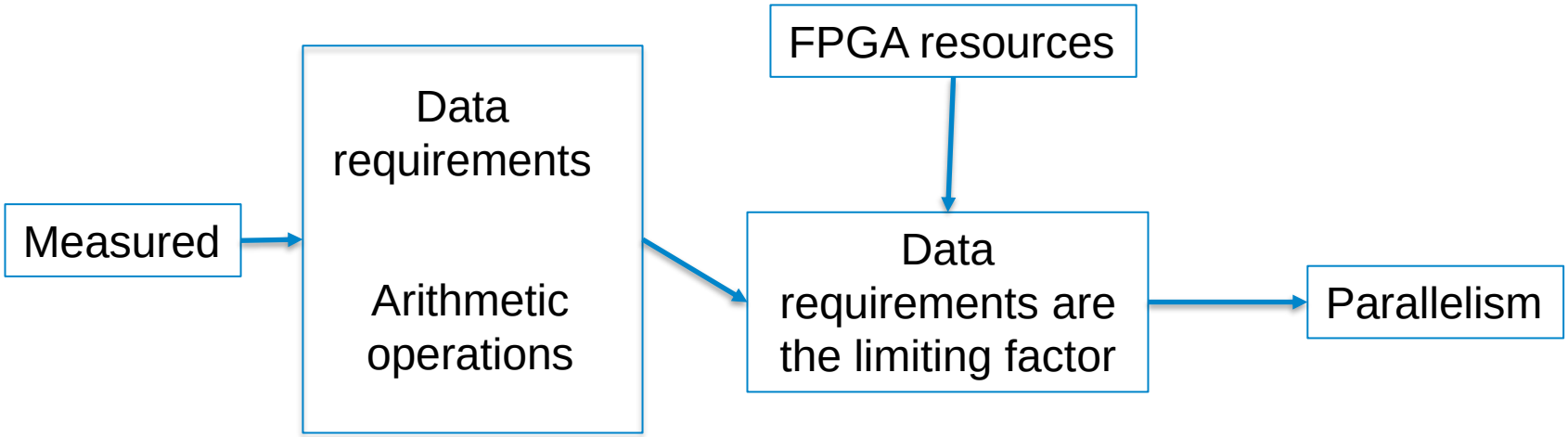



2D

Source: Kotlarchyk M. Scattering theory. Encyclopedia of Spectroscopy and Spectrometry, 1999.

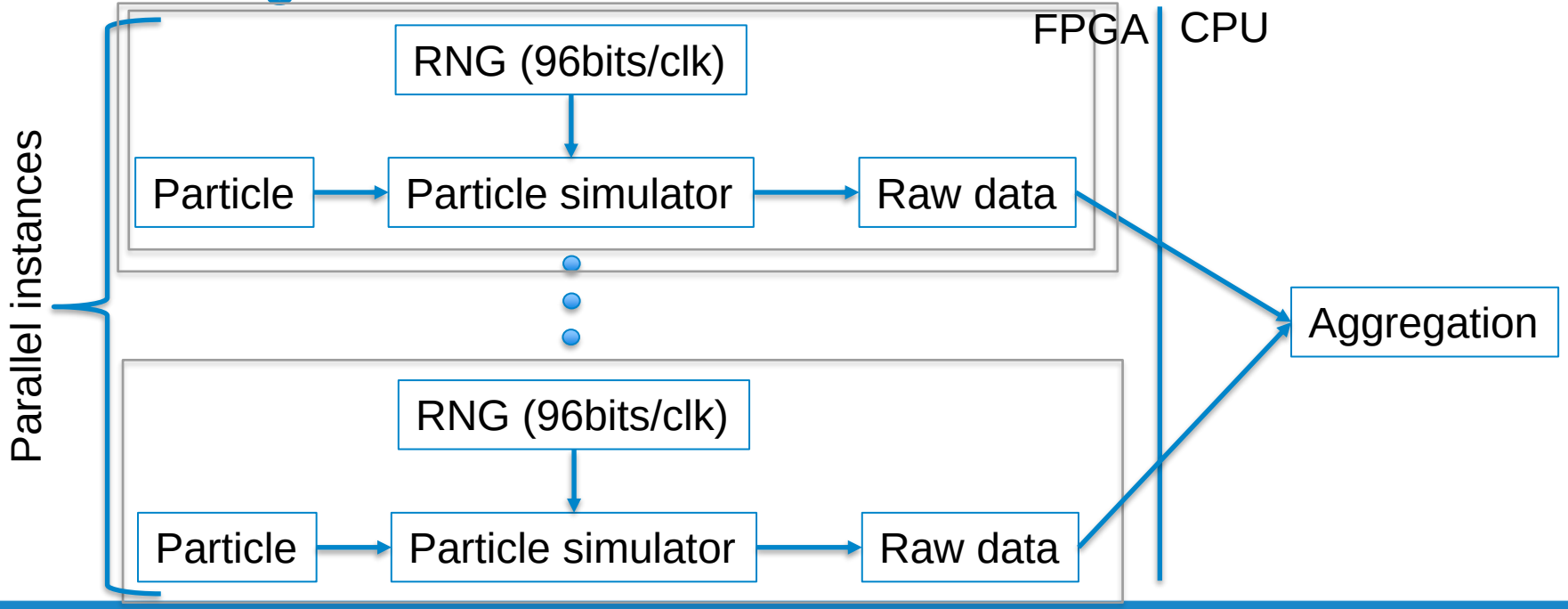
Imperial College London

# Modelling



Imperial College London

# Resulting architecture



Imperial College London

# Conclusions

FPGAs can be multiple orders of magnitude faster than CPUs and GPUs when accelerating suitable workloads

Due to their stochastic nature Monte Carlo simulations greatly benefit from FPGA acceleration

Preliminary results shows over 100x speedup compared to high-end multicore consumer CPUs