Towards Real Time Radiotherapy Simulation

Nils Voss, Peter Ziegenhein, Lukas Vermond, Joost Hoozemans, Oskar Mencer, Uwe Oelfke, Wayne Luk and Georgi Gaydadjiev

Imperial College London

ICR The Institute of Cancer Research

MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

The 30th IEEE International Conference on Application-specific Systems, Architectures and Processors
16. July 2019

Outline

- Problem Description
- Programming Model
- Architecture
- Evaluation
- Conclusions and Future Work

Problem Description

- Radiotherapy common treatment for cancer
  - Idea: Kill cancer using high radioactive dose
  - It is important to limit damage of healthy tissue
- Monte Carlo Simulations are used as common tool for dose simulation
  - Need to simulate huge number of particles to achieve statistical meaningful results
  - Computationally expensive / long runtimes

Performance Target

- Dose simulation at real time
  - Simulate representative dose in less than 1 sec
    - Requires simulation of 100 million particles
- This will enable adaptive treatment
  - New state of the art machines combine MRI scanning with linear accelerator
  - In most cases the cancerous organ moves within the body between CT and therapy
    - Or even during the therapy session
  - -> Less sessions required, more targeted treatments with less damage to the healthy tissue

Monte Carlo Dose Simulation

- Patient body data represented as cube
  - Cube is discretised in voxels
- Particles are sent into the cube
  - Each particle has energy and fuel values, determining travel distance as well as a direction vector
  - When fuel runs out
    - particle interaction of a certain kind is selected
    - the voxel dose is aggregated
    - a new particle direction is randomly set
  - When energy runs out the particle gets absorbed / removed

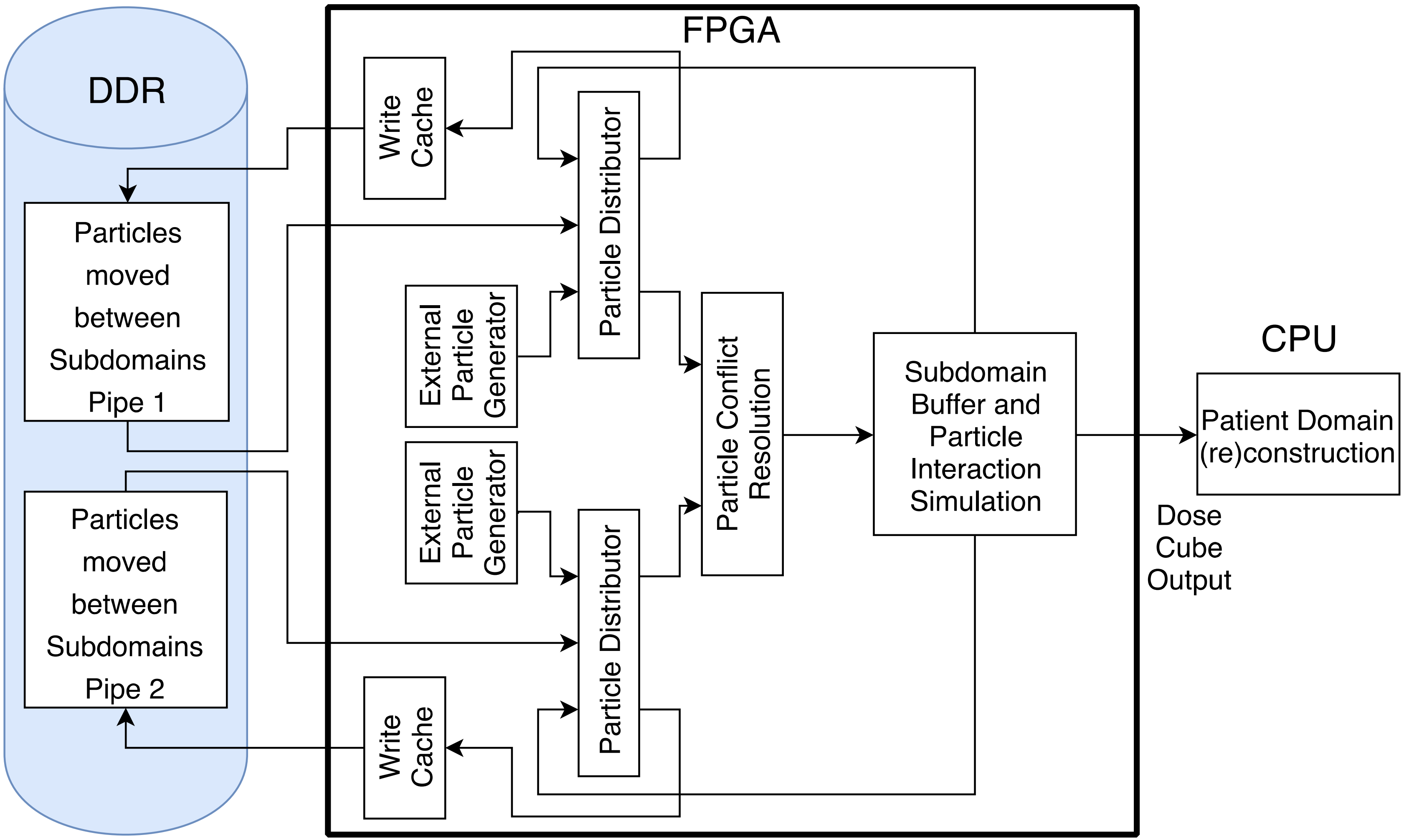
FPGA Implementation Challenges

- The path through the patient cube is random
- Patient cube size is much bigger than FPGA on-chip memory
  - -> Decompose big cube into smaller sub domains
- Particles are processed in a loop
  - Final result is needed to start next loop iteration
  - Direct data dependency between loop iterations interferes with pipelined execution
  - -> Reorder compute to process only one iteration per particle incrementally, to break dependency

Programming Model: Dataflow

- Describe your computation as a directed graph
- Data flows through the graph, and gets processed while traversing through the nodes
- Maxeler's MaxCompiler tool chain maps onto dataflow engines (DFEs)
- Current generation DFE: MAX5C
  - based on VU9P FPGA from Xilinx
  - ~7,000 multipliers
  - ~40 MB on chip memory
  - 48 GB DDR4 memory
  - ~50 GB/s memory bandwidth

Our Architecture



MAXELER Technologies MAXIMUM PERFORMANCE COMPUTING

Evaluation

- Targeting Maxeler MAX5C DFEs using MaxCompiler 2018.3.1 and Vivado 2018.2
- CPU platform 2x Xeon E5-2643 v4 (6 cores each @ 3,4 GHz)

Frequency	Design Count	Kernel Parallelism	Cube Size	Subdomain Size	LUT	FF	DSP	BRAM	URAM
250	3	2	128	64	543,642 (45.98%)	1,069,421 (45.23%)	2,457 (35.92%)	2,916 (67.5%)	384 (40%)

Cards	FPGA Time [ms]	Total Time [ms]
1	3,267	3,435
2	1,173	1,342
3	810	988

- 4.1x speedup compared to CPU
- 8x speedup compared to GPU
  - Due to random memory access and different interactions problem with SIMD execution

MAXELER Technologies MAXIMUM PERFORMANCE COMPUTING

Conclusion and Future Work

- Conclusions
  - Presented architecture for FPGA based Monte Carlo radiotherapy simulation
  - Detailed performance model (in the paper)
  - Met real time requirement of simulating 100 million particles in less than one second using three FPGA cards
  - 4x speedup over CPU and 8x over GPU
- Future Work
  - Implement additional particle interaction (bremsstrahlung)
  - More particle types and materials
  - Integration into latest radiotherapy systems

Questions?

MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

10.7.2019