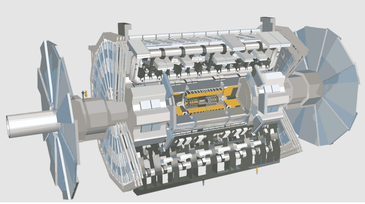
Charged Particle Track Reconstruction Algorithms for Massively Parallel Systems

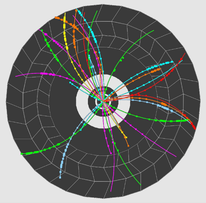
Stephen Nicholas Swatman adv. Ana-Lucia Varbanescu, Andy Pimentel, Andreas Salzburger, Attila Krasznahorkay

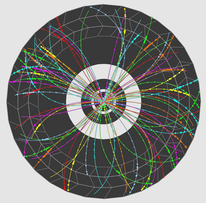
University of Amsterdam

Introduction

- **High-energy physics** experiments record **collision events** at rates of around **40 MHz** (at the **Large Hadron Collider** at CERN)
- **Particles** are identified based on their trajectories or **tracks**
- We aim to reconstruct **continuous** particle trajectories from **discrete** measurement points: **track reconstruction**
- We tackle both **real-time** and **non-real-time** processing of data: high performance required in both cases
- The **pipeline** consists of **structurally different** algorithms
- Upcoming upgrades will increase **data volume**: CPU solutions will not suffice – **how can we run on massively parallel GPUs**?

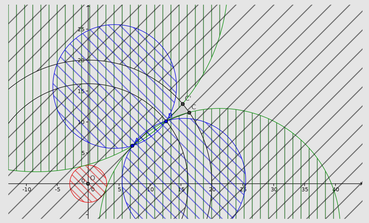


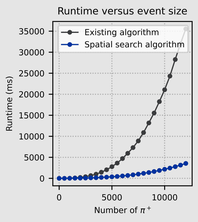




Seed Finding Using *k*d Trees

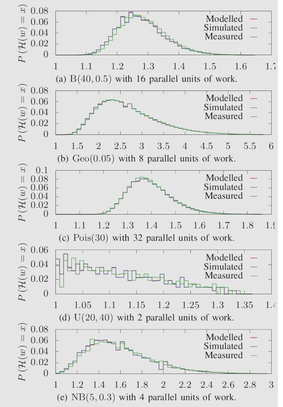
- **Seed finding** is a **combinatorial** problem: form **triplets** of points, n^3 combinations where $10{,}000 \leq n \leq 100{,}000$
- SOTA reduces combinatorics using **spatial grouping** into bins
  - Leads to **imbalanced bins** and **inflexibility**
- We propose a **novel method** using ***k*d trees** with $\mathcal{O}(n^{2/3})$ search
  - Allows **dynamic search regions** and is **efficient** on **GPUs**
  - Requires analytical **translation** from **compatibility criteria** to **axis-aligned search queries**
  - Significant **performance improvements** in **CPUs** and **GPUs**



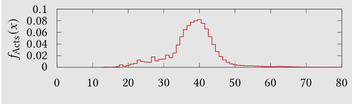


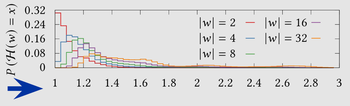
Modelling Thread Divergence

- **Thread divergence** is an important consideration in SIMT application design
- Thread **coarsening** and **refinement** can reduce divergence; **no models are available** for the impact on performance
- We propose a new **statistical** model for the performance loss in SIMT applications with **stochastic workloads**
  - Enables **analytical** evaluation of the efficacy of optimizations
  - Requires only **a priori** knowledge of the workload **distribution**, which can be measured!



(a) B(40, 0.5) with 16 parallel units of work.
(b) Geo(0.05) with 8 parallel units of work.
(c) Pois(30) with 32 parallel units of work.
(d) U(20, 40) with 2 parallel units of work.
(e) NB(5, 0.3) with 4 parallel units of work.





Hit Clustering
Structured grids, graph traversal, sparse data

Spacepoint Formation
Map-reduce, dense linear algebra

Seed Finding
N-body, graph traversal

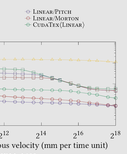
Track Parameter Est.
Map-reduce, dense linear algebra

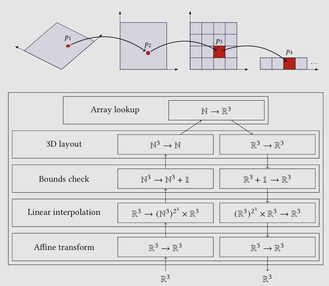
Combinatorial Kalman
Branch-and-bound, structured grids

Track Fitting
Structured grids, dense linear algebra

Design Space Exploration of Vector Fields

- Strong reliance on **structured vector fields**: multi-dimensional vector data. In-memory layout affects **performance**!
- There is a large **design space** across hardware and software
  - Including e.g. GPU **texture units**
- We propose methods for automated exploration of this space using **compile-time composition**
  - Data layouts, interpolation methods, bounds checking, etc.
  - Allows automated exploration with **native performance**

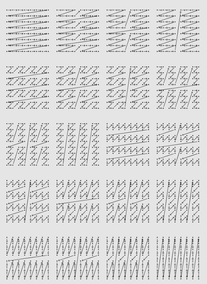




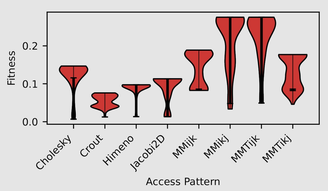
(a) AMD EPYC 7402P (48 threads)
(a) NVIDIA A100
(b) Intel Xeon E5-2630 v3 (2 sockets, 32 threads)
(b) NVIDIA A6000

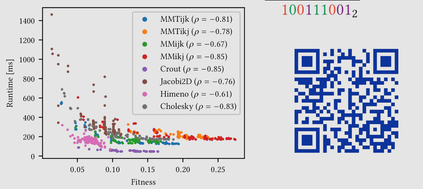
Morton-Like Layouts for Cache Efficiency

- **Morton layouts** provide balanced spatial locality
  - Implemented using **bit interleaving**
- Allowing **arbitrary** interleavings gives **very large** family of layouts with **novel cache properties**, but there are **too many** layouts to explore **exhaustively**
- Explore space using **evolutionary algorithms**
  - We are able **quickly** to find layouts with **superior cache behaviour** over canonical row- and column-major layouts in *some* applications
  - Can improve performance by up to **10x**!



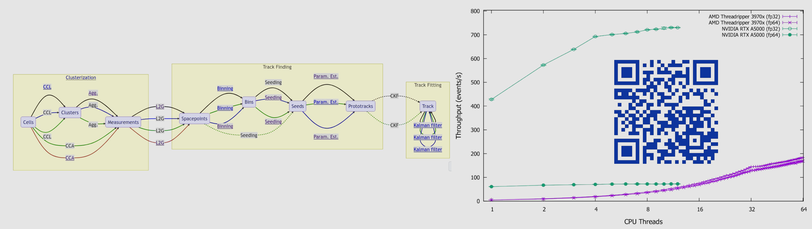
$$f(100_2, 011_2, 101_2) = \begin{array}{r} 100000000_2 \\ 000011000_2 \\ \vee\ 000100001_2 \\ \hline 100111001_2 \end{array} = 313_{10}$$





Conclusion

- We need **very fast track reconstruction** for future high-energy physics
- State-of-the-art track reconstruction does **not** translate well to GPUs: workloads are too **imbalanced** with poor **accesses patterns**
- We tackle **different pipeline stages** with **specific, novel solutions**
  - New algs. for seed finding and others that **reduce imbalance**
  - **Model of divergence** to evaluate **refinement** and **coarsening**
  - Systematic exploration of **multi-dimensional data storage**
  - **Evolutionary approach** where design space is too large
- We demonstrate the **first GPU-enabled track finding software** which **runs efficiently on GPUs** (>5x speedup over CPU solutions)



- **Stephen Nicholas Swatman**, et al., 2022. "Modelling Performance Loss due to Thread Imbalance in Stochastic Variable-Length SIMT Workloads". In *Proceedings of the 2022 30th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Nice, France. DOI: 10.1109/MASCOTS56607.2022.00026.
- **Stephen Nicholas Swatman**, et al., 2023. "Systematically Exploring High-Performance Representations of Vector Fields Through Compile-Time Composition". In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE)*, Coimbra, Portugal. DOI: 10.1145/3578244.3583723
- **Stephen Nicholas Swatman**, et al., 2023. "Finding Morton-Like Layouts for Multi-Dimensional Arrays Using Evolutionary Algorithms", *under review at PPoPP 2024*, Edinburgh, United Kingdom.

Contact me at **me@stephenswatman.nl**!

The work presented in this poster was supported by the CERN doctoral student programme and by the University of Amsterdam.