Compass SPMD: a SPMD vectorized tracking algorithm

Placido Fernandez Declara on behalf of LHCb RTA project
University Carlos III of Madrid and CERN

Introduction

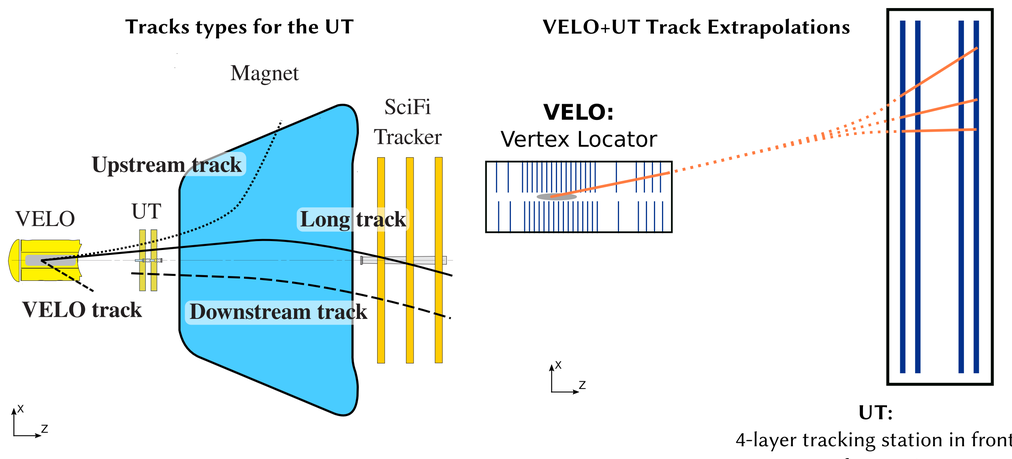
Compass is a SPMD (Single Program Multiple Data) tracking algorithm for the upcoming LHCb upgrade in 2021 [1]. 40 Tb/s need to be processed in real-time to select events. Alternative frameworks, algorithms and architectures are being tested to cope with the deluge of data.
Allen is a R&D project aiming to run the full HLT1 on GPUs. Allen's architecture focuses on data-oriented layout and algorithms. GPUs already proved to exploit the framework efficiently. We explore opportunities for the SIMD (Single Instruction Multiple Data) paradigm through the Compass algorithm. We use the Intel SPMD Program Compiler (ISPC) to achieve good readability and performance, writing "GPU-like" source code, preserving the algorithm design.

ISPC sample

```
1 typedef float<3> float3;
2
3 export void chi2_ispc(
4   uniform const size_t N, uniform float chi2 [],
5   uniform float3 x [], uniform float3 y [],
6   uniform const float m, uniform const float q) {
7     foreach(i = 0 ... N) {
8         varying float3 expected_y = m * x[i] + q;
9         chi2[i] = (y[i].x - expected_y.x) * (y[i].x - expected_y.x) +
10                  (y[i].y - expected_y.y) * (y[i].y - expected_y.y) +
11                  (y[i].z - expected_y.z) * (y[i].z - expected_y.z);
12    }
13 }
```

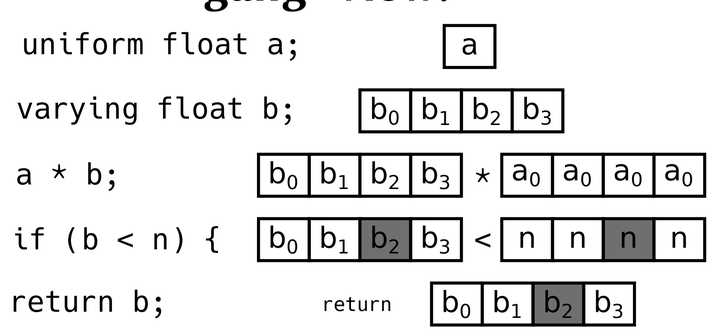
UT tracking

- Reconstruct charged particle trajectories that cross the UT
- Match a VELO track with UT hits to create a VELO+UT track
- Tracks are slightly bent by the magnetic field, introducing more multiplicity to the calculations.

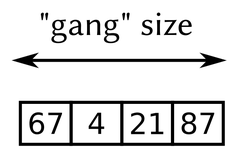


ISPC: Intel SPMD Program Compiler

"gang" view:

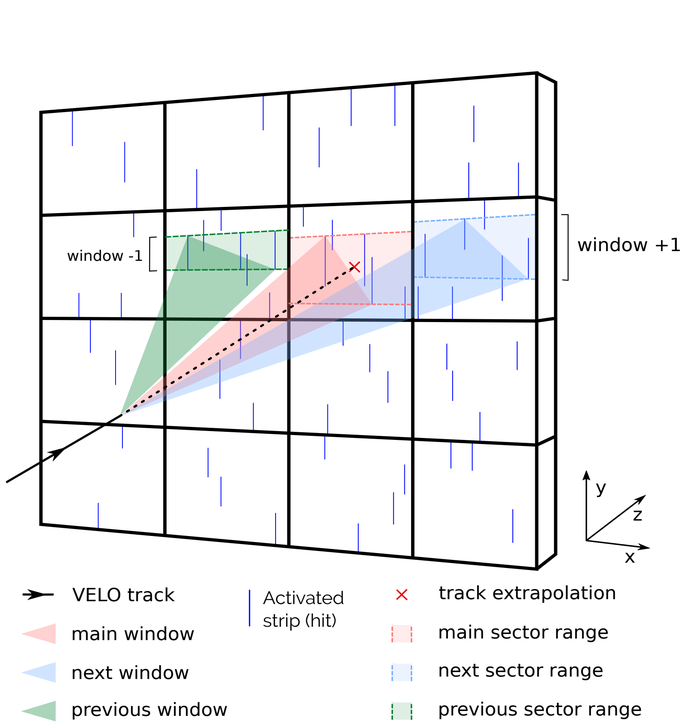


A "gang" is a group of *program instances* that run in parallel through the vector units. Similar to a CUDA "warp".

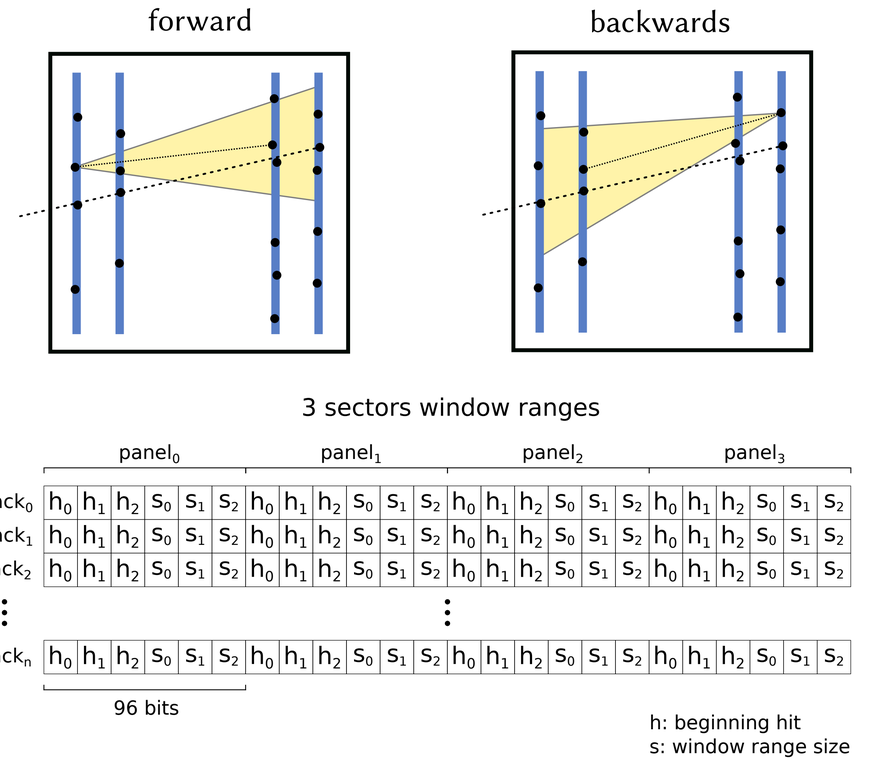


- C-based language with additions: foreach, foreach_tiled, foreach_active, cif and others.
- Program C-like sequential algorithms, execute in parallel (SPMD) in a "gang" of program instances that map to vector lanes. Easier to reason about vectorization.
- Memory alignment and access pattern remains critical.
- Explicit scalar and vector types: varying and uniform.
- Uniform are scalar types, same value accross a "gang".
- Varying are vector types, different across a "gang".
- Interoperable with C/C++: SPMD algorithms are mixed with "regular" algorithms in Allen framework.
- Features SSE, AVX1, AVX2, AVX-512 and NEON instructions.

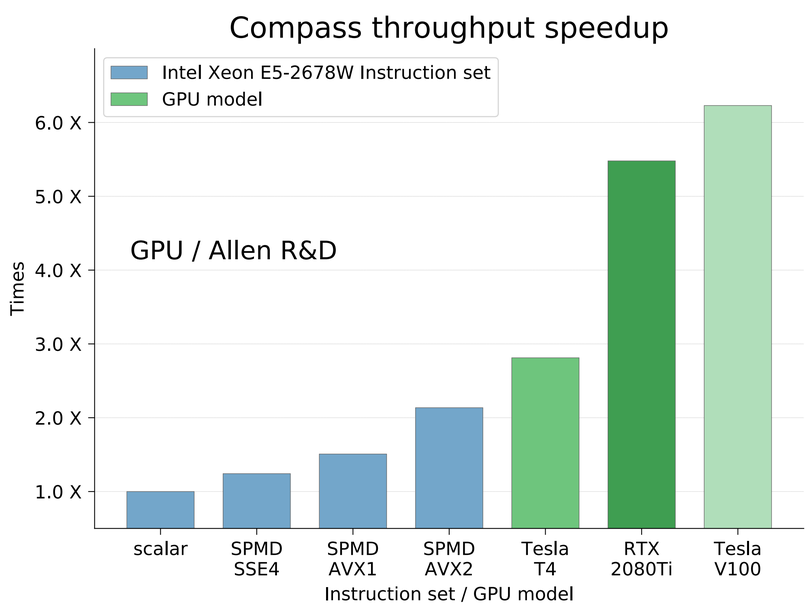
SPMD Compass tracking



- Hits are arranged into sectors. Inside sectors, hits are locally arranged by Y coordinate.
- Binary search the regions by extrapolating VELO track and define a "window range" where to later search for hits.
- A total of 5 window ranges (3 in the image) per track is used to account for bending tracks.
- Minimize branching, filter non-valid tracks first for optimal usage of vector lanes, cache window ranges for efficient access.
- Search for triplet/quadruplets with combined forward-backward loop to reduce divergence.
- Lower vector usage due to tracks being filtered at the beginning and the end of the algorithm.
- Physics efficiency differs < 1% from Allen baseline (GPU).



Performance comparison



- All instances run same number of events and threads. CPU is Intel(R) Xeon(R) CPU E5-2687W.
- More than 2X speedup is achieved with AVX2 target instruction set.
- Due to early track filtering and multiple branching, vectorization efficiency lowers.
- Candidate finding instroduces many conditional clauses that lower the "gang" usage.
- There is room for improvement, not all "kernels" are running with SPMD nicely.
- GPUs still vectorize better by writting "equivalent" source code.

Future work & code

A nice speedup is achieved writing CUDA-like code, but better speedup and vector lane usage was expected. Further optimizations could be introduced, and changes to access pattern. Some "kernels" still run faster not vectorized.

Allen source

Allen SPMD source

[1] P.F. Declara *et al.*, *A parallel-computing algorithm for high-energy physics particle tracking and decoding using gpu architectures.* IEEE Access 7 (2019) 91612