

The VELO detector and GPU strategies

GAP meeting - Pisa

M. Corvo

S. Gallorini, A. Gianelle, A. Rugliancich

13 Gennaio 2014

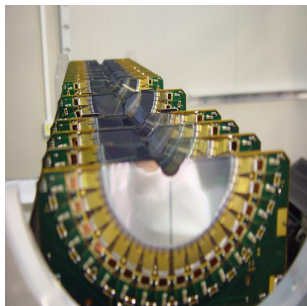


Outline

- Current VELO algo
- GPU approach
- Issues

The Velo detector

- The current VELO algorithm to detect particle tracks is named FastVelo
- Basically is split into two main functions
 - Find Quadruplets searches for seeding quadruplets of hits in the RZ projection
 - Make Space Tracks adds the information of the Phi sensor to provide 3D extrapolation



Half Velo sensors

GPU strategies I

- We started following a minimal perturbative approach, that is let's keep as much as possible of the original code
- Major limitations are nested for loops and the presence of tagging, that is keeping already used hits outside of future iterations
- We applied some "common sense" ideas, that is parallelize over sensors' zones and over quadruplets, avoiding tagging (although performance degraded significantly as many more tracks appeared) and
- Information copied to the device is kept at a minimum: light structures with hit coordinates and few more things
- Another approach exploits Hough transform to detect tracks. Completely different algo which needs to be tested and verified against data and current "trusted" algos

GPU strategies II

- Liverpool group adopted a different strategy and algo
- Events are fully copied to the device and 3D space hits are built accordingly
- Track fit is (being) accomplished by a Kalman Filter algo
- Some efforts are already being made to implement on GPU new algos for the VeloPixel, the evolution of the current Velo for the upgrade

GPU strategies issues

- Computation must be offloaded to the device
 - Currently there's a client-server system to dispatch the workload
 - Good in a computing environment where nodes with attached GPU are just a handful
 - Less interesting (imho) in one where every node is equipped with an APU (realistic future)
- GPU are made to solve simple computational problems
 - Not the best environment for OOP
- Organization of data matters
 - Complex structures prevent memory coalescence
 - Prefer SOA instead of AOS