



OPENED DISCUSSION 5TH FCTTC WORKSHOP

Jérôme Lauret

Reminder- The FCTTC “identity”



- Track finding and fitting;
- Finding and reconstruction of vertices and short-lived particles;
- Parallel computer architectures;
- Software implementation for the parallel architectures;
- Software architectures and frameworks.

We have discussed expanding but decided NOT ... The identity is well defined and appreciated & balance deep algorithmic, hardware, technics and other projects seems like the right one. Still no other workshops covering for those topics in a consistent manner

We are opened to other suggestions

5th workshop – goals



- Keep the activities and have status reports from all, sharing experience
- Try to engage more experiment, leveraging the 4th workshop dynamic (or at least keep some)
- Maintain participation from other communities (Geant, ROOT, ...)
- Document, communicate via mailing lists, code repository
- Make sure we have a sustainable path for Vc and promote its usage
- Create a common package and start with KFParticles

5th workshop – Impressions

- Many thanks to all contributors for their time, efforts, thoughts, questions, debate, ...
- Many thanks to our host, all went flawlessly with good “*discussion helper*” at the end of the day 😊
- Pretty good workshops and a success ...
 - Nice turn out with many new faces (we hope to see again)
 - Many previous contributors still coming speaks for the usefulness of the workshop and its “friendly” structure
 - The “usual” experiments are there – ALICE, CBM, STAR
 - AND we have PANDA and LHCb participating
- Activities
 - From the “core” has made much progress – HLT, CA, KFParticles
 - HLT clearly a success in many running experiments and generating new ideas
 - New experiments are at an ideal Phase to consider the successful approach and models from others

5th workshop – also ...

- We need to continue to maintain the balance that has been successful
 - Keep focused on past engagements: we “nearly” missed OpenLab (workshop scheduling overlap) – Many thanks to Andrzej Nowak for finding a solution
 - Keep an eye on the experimental paradigm and what this means (streaming events a-la-CBM, ...)
- We also need to focus on our past goals
 - ... can't say we have wrapped up the common package goal ...
 - ... we need to communicate more on a common list ... like a collaboration would ...

Feedbacks from the last workshop: ATLAS / CMS / LHCb

- ATLAS: these workshops are useful as a forum to discuss tracking across experiments, which is something the community lacks overall
- ATLAS: It is a good forum to show and express what has worked for us and what not. Sharing the ground breaking work (math libs...) is clearly interesting for us.
- ATLAS: The replacement of CLHEP with math functions that directly support vectorization could be a community push and activities reported at those workshops
- ATLAS:/CMS Already have a vertexing solution and unlikely going to leverage it in the near term. Technology aspects for tracking applications is the most useful.
- CMS: workshop useful as it allows considering new aspects for parallelization, vectorization beyond the pure technical aspect of it (that is, at algorithmic level)
- CMS: some interest to try CA especially as it seems promising
- CMS: We find that keeping an eye on and being informed of the hardware and evolution is interesting and fascinating . The workshops should preserve this aspect (if not only for an educational point of view)
- CMS: Workshops is a good medium to get in touch with experts
- LHCb: The workshops are a medium to create productive and collaborative activities with other experiments and exchange experience – we do not have similar activities in the community.
- LHCb: Interest in many aspects, from Geant to tracking many aspects seem applicable and represents building “bricks” for the future

Feedbacks from the last workshop - CBM

- Workshop gives all participants the chance to profit from the exchange of information, ideas and experiences.
- Of particular interests, the direct connection to the hardware developments and connected issues like compilers and tools.
- It is not all clear on how one would further organize a common effort and joint activity as the experimental groups have diverse interests and at different stage of evolution and priorities
- Success of a common package relies on two aspects - the technological possibility and the motivation of the experiment groups to take part in a joint effort. Much natural tendencies to fight: inertia, feel that “our solution is best”, reluctance to adopt black-boxes
- The key issue is to define abstract interfaces to the conditions (material and magnetic field), which is not a straightforward thing to do. A “prototype” package would be more attractive than “I will send the student to implement X in your framework”
- Needed – a few POC from experimental group helping to test a common package – ultimately, the student who have worked on many implementations may be best knowledgeable and equiped to define/propose a common interface
- HLT should be re-emphasized on the context of the LHC upgrade – will have much problems similar to CBM
- Workshops highly biased toward Intel – can we rebalance?

Before opening the floor ...

- It is good to have a “community” engaged and committed to the workshop’s topics – started a “*grassroots effort*”
- Engaged ⇔ please participate and make the activities your own
- I am all for self-assessments and re-adjusting: will poll people again as your feedback is important ...
 - Was the workshop what you expected? What were your initial interests?
 - Did it provide a proper balance of topics?
 - Was it useful to you? What areas were most interesting?
 - What do you think is missing, would be nice to have?
 - New directions & drivers?
- Workshop feedback – jlauret [at] bnl.gov





Notes

- Broadening the hardware scope
 - NVIDIA – Andreas knows more about this (significant group in Germany)
 - Topic pretty much like OpenLab / Intel. New hardware, how does it connect to our field and activities, etc ...
- Common package – KFParticles (best candidate)
 - Deliverable – need priority over extending current functionalities
 - De-coupling from experiment framework and making it a standalone package high priority – would demonstrate we can establish this
 - CBM: Also need a solid common interface – working with multiple communities would achieve this

Notes

- How to develop a support community?
 - Open the source – make it available by many and feedback will come
 - Why only KFParticles? [github] – need to be a priority so feedback comes and the package can be consolidated
 - Worries: opening the source to all may create a support scalability issue (many questions from many community and no sufficient support)
 - Pro: but Emails are not only about “we have a problem” – also “we are expert in X and would like to help” ...
 - How do we handle issues i.e. people who have questions / problems?
 - Cons: reluctance because maturity is needed for the start of a package -> start with KFParticles should be OK
 - Organizational structure has to be thought about

Notes

- Q? How did FairROOT evolve and supported the "core" product.
 - Started with CBM, expanded to PANDA, ... 3 people at first.
 - Got lots of Emails but also patches, suggestions to improve, new features, ...
 - Also, faced communities with different paradigm – enriched the product at the end (more powerful product at the end).
 - 6 + 3x3 external for ~ 10 experiments at different stages of development (exploitation, design phase, etc ...)
- Correlative
 - Would we have at least 1 person for a "core" support for KFParticles
 - We need a firm commitment from the community this is a needed path / common package
- Longer run – could include into a bigger package?
 - Not to decide now but could be easier for the long term
 - SIMDized KF could also be a consideration ... may not have a common interface but may serve as a "template" code, educational and examples for real-implementation

Notes

- Mindset
 - I develop my code for others to see and use – OpenSource approach should be more systematic from the start
 - GSI decided that codes developed at GSI would be publicly available for everyone (GPL)
- Vc in ROOT and beyond
 - ROOT 6 default and 5 (with enable-vc)
 - Final plans there would be known later
 - What about a new “tool” comes along?
 - Sandro’s work gives an idea of how to approach it – with higher level “types”, it may be possible ...
 - Instruction differences would be handled “deep” inside packages ...
 - Could also think of SIMD types pushed into C++ standards
 - Community well connected – work to push some vector type into the
 - C++ standard. But even if this happens, Vc can be re-tooled to use those types (no loss)

Notes

- Vertex
 - Have some problems with finding multiple vertices
 - Solution seem to exists in other experiment – how to leverage the knowledge ?
 - Issue is that KFParticles now finds one primary VF, the rest are secondaries ... How to bring the best of all worlds into one package?
 - Version may exists from Maksym (TBC)
- Online / Offline
 - How to make packages work standalone and portable on all framework (or when there is a lack off)?
 - Data interchange with MQ?
 - Can we define a common format as input of the algo / packages?
 - Software engineering problem
 - Also a question of GPU, not-GPU, ... Is this a coding style issue?
 - If package is opened, people may contribute to the interface for receive/send data

Notes

- Online / offline (cont.)
 - Should one also consider a port for FPGA?
 - Don't really distinguish between online & offline ... +
 - Xeon/Phi, GPU, CPU ... can we consider "anywhere there is a C++ compiler"?
 - Cannot decide in the workshop of what the experiment would do: True but this is a question of support for the future ...
- PANDA
 - Perspective: Cannot build a complete FPGA based online framework ... parts may be FPGA based
 - If one would decide that KFParticles would be worth porting to FPGA, this could be an external contribution providing OpenSource

Notes

- Online “analysis”?
 - Long term future / vision: data selection is already a Physics analysis
 - Would use the online calibration and the “best” knowledge but would make Physics based decisions / selections – trigger? [perhaps not] tagging? [more likely]
 - More like a “pre-analysis”? Tagging and stream for event of interests ...
 - Update could come later
 - “Physics based tagging capabilities” would be a better term
 - Note / comments
 - A fully accomplished physics analysis with online quality is hard
 - Systematic errors are hard to assess in real time
 - Normal way so far – accumulate events, calibrate, align, etc ... then process data (data production) and physics analysis. Eventually, refine.
 - But ... J/Psi identification or other production channel with low cross-section. How to enrich but also, how to “tune” ...
 - Look as it goes that a peak appears – assess online and retuned ... Physics monitor
 - *Note*: before publication, must be reproducible hence data preserved and code can be re-executed
 - BTW: example – anti-He3 and 4 in STAR (HLT + reco & stream selection). The analysis was done offline however
 - More examples along this theme ...