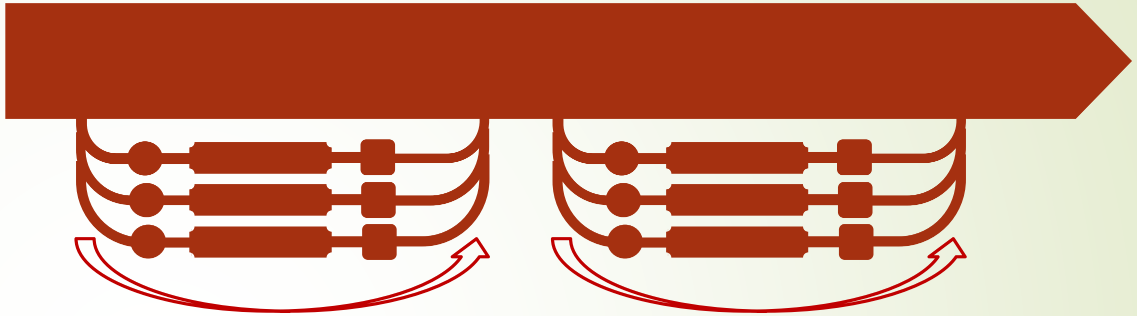


VectorFlow

A subscription-based vector data-shipping service

andrei.gheata@cern.ch

The context

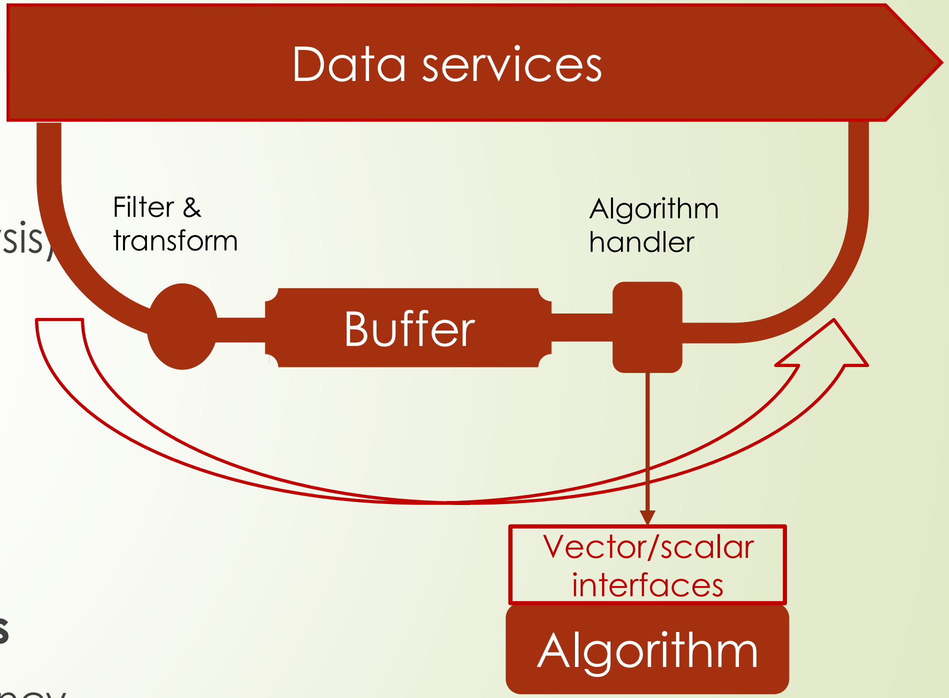
- Extensive R&D work on vectorization was done both in SFT (ROOT/VecCore, GeantV, VecGeom) and experiments during the last years
  - Vectorization is good (when it happens) but difficult to control and extend
  - More and more examples on how to do it, helper libraries (Vc, VecCore), tools to control data formats (e.g. SOAContainer) and larger knowledge base
- Getting the data to the vector registers is tough…
  - Compilers offer very limited help with our kind of code
    - **revolution in compilers?**
  - Important code transformations required for explicit vectorization
    - **support in C++ standard ? Meanwhile we have VecCore…**
  - Not all algorithms benefit of nice internal vectorizable loops. Data transformations needed
    - **we may be able to do something about this one…**

The idea

- Most of the algorithms that work today on scalar data could benefit of vector input
  - A natural loop on data of the same type preempts vectorization
  - Repetitive work increases locality and allows for much better cache coherence
  - Delivering data to algorithms in the appropriate vector form represents a large part of the vectorization effort
- In practice: **move the event loop into to the algorithm**
  - Subscribe to data using **filters** integrated in the main event loop
  - Use a data transformation service to **extract & buffer** views from the input event data AOS to custom SOA relevant for the algorithm
  - Deliver buffered data to the algorithm using either **scalar or vector interface**

VectorFlow

- **On-demand service complementing the functionality of existing frameworks**
  - Non-intrusive (adapt to existing frameworks/event loops)
- **Potential benefits:**
  - Extend vectorization potential to more areas (such as analysis)
  - Better performance due to increased code/data locality & vectorization
  - Allows scalar and vectorized algorithms to coexist (smooth migration)



- **Many related R&D opportunities**
  - Performance studies, concurrency
  - Extending type-based vectorization approach to new areas
  - Integration with experimental frameworks (e.g. CMSSW, Gaudi)
  - Extension of vector flow to devices (e.g. GPGPU)

andrei.gheata@cern.ch