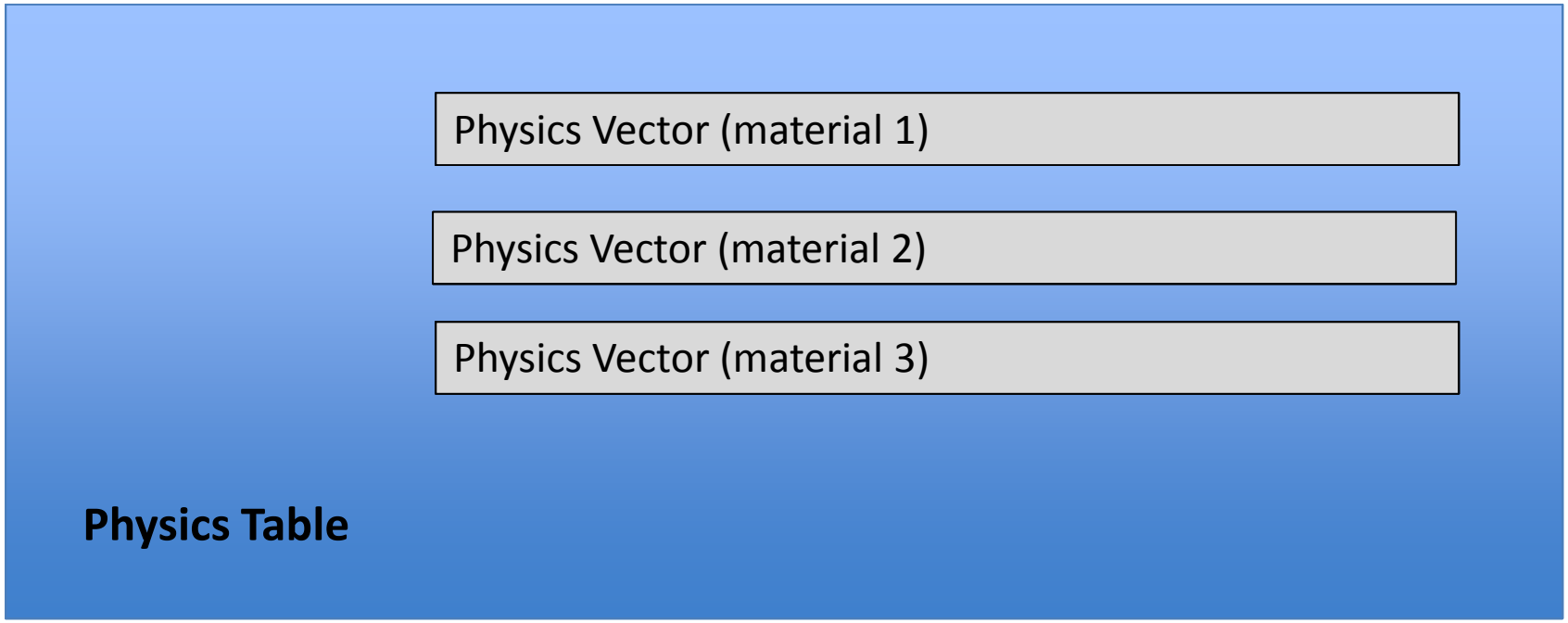
Physics tables and multi-core

J. Apostolakis

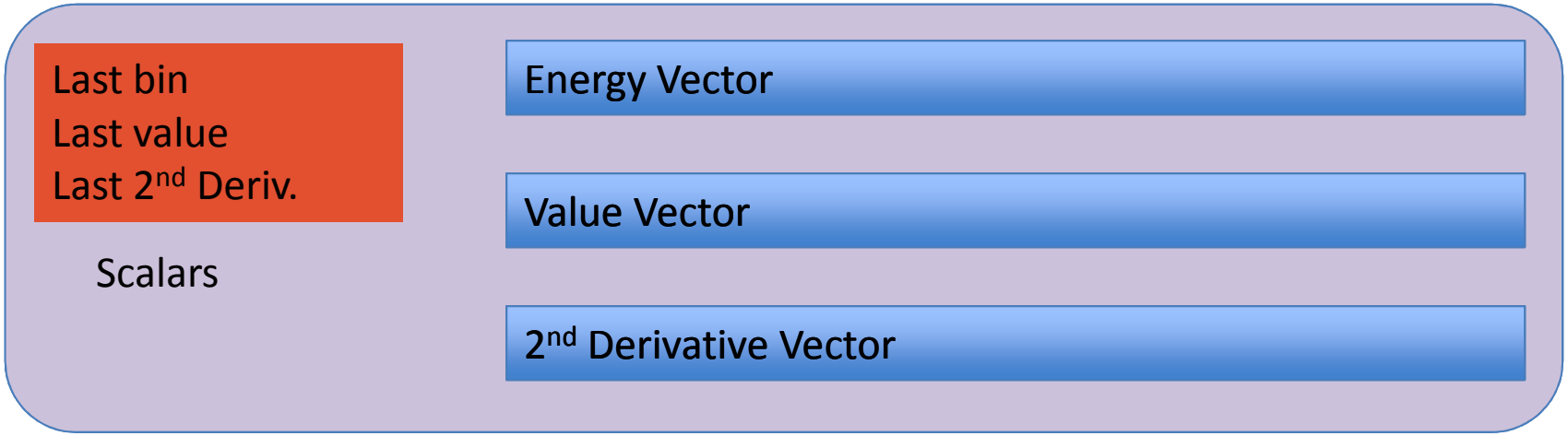
Motivation

- Limited reuse of memory in Multi-Processing
  - A forked process shares all pages of memory which read-only (called 'Copy-on-write'=COW)
  - With Geant4 less than 30% can be shared
- Leading reasons:
  - Caching in physics tables
  - Replicas' copy numbers.
- With simple changes could increase reuse

Inside a Physics Table



Inside a Physics Vector



Cache: Read/Write

Read-only after filling (initialization)

3 doubles

Typically 70-150 double values each

Typically created in consecutive areas of the heap. The result is:
By writing the 3 doubles (cache) a process creates copy of page(s) which containing ~ 300 doubles

Multiprocessing and ‘timing’

- BeamOn is called
- Initialization of geometry
- Initialization of physics processes
- First event is processed
- Worker processes (or threads) are created

- Fork waits until the physics tables are initialized!

Requirement for multiprocessing

- Use separate areas of memory
  - One for the scalars (which are rewritten)
  - A different one for the vectors
- Fork waits until the tables are initialized & shared.

Complications?

Other large arrays used by physics processes

- Static arrays in Brems, pair production, Goldsmith-Saunderson MSc
- C-arrays in
  - Hadron Elastic (no caching)
  - CHIPS Elastic (?caching?)

Outcome

- Revision of design of physics vector
  - To separate areas of memory for scalars and vectors
  - Hisaya was present – he maintains phys. vector