Faculty of Physics

WARSAW UNIVERSITY OF TECHNOLOGY

A novel algorithm of event mixing for ALICE Run 3

Maja Kabus

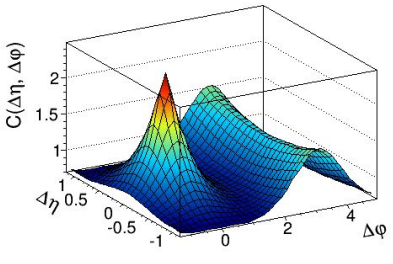
Quark Matter, 08.04.2022

A novel algorithm of event mixing for ALICE Run 3

Angular and femtoscopic correlations: analyzing QGP initial state and thermalization mechanisms

Correlation function: $S(\Delta\eta, \Delta\phi) = \frac{d^2N^{signal}}{\Delta\eta\Delta\phi}$ $B(\Delta\eta, \Delta\phi) = \frac{d^2N^{mixed}}{\Delta\eta\Delta\phi}$

$$C(\Delta\eta, \Delta\phi) = \frac{N^{mixed}}{N^{signal}} \frac{S(\Delta\eta, \Delta\phi)}{B(\Delta\eta, \Delta\phi)}$$



Event mixing: **pairs of tracks** (V0s/cascades/…) from **2 different collisions** from the **same bin,** e.g., multiplicity and z-vertex intervals.

Run 2: sort collisions into **a vector of mixing buffers**, at the same time select pairs in a double loop

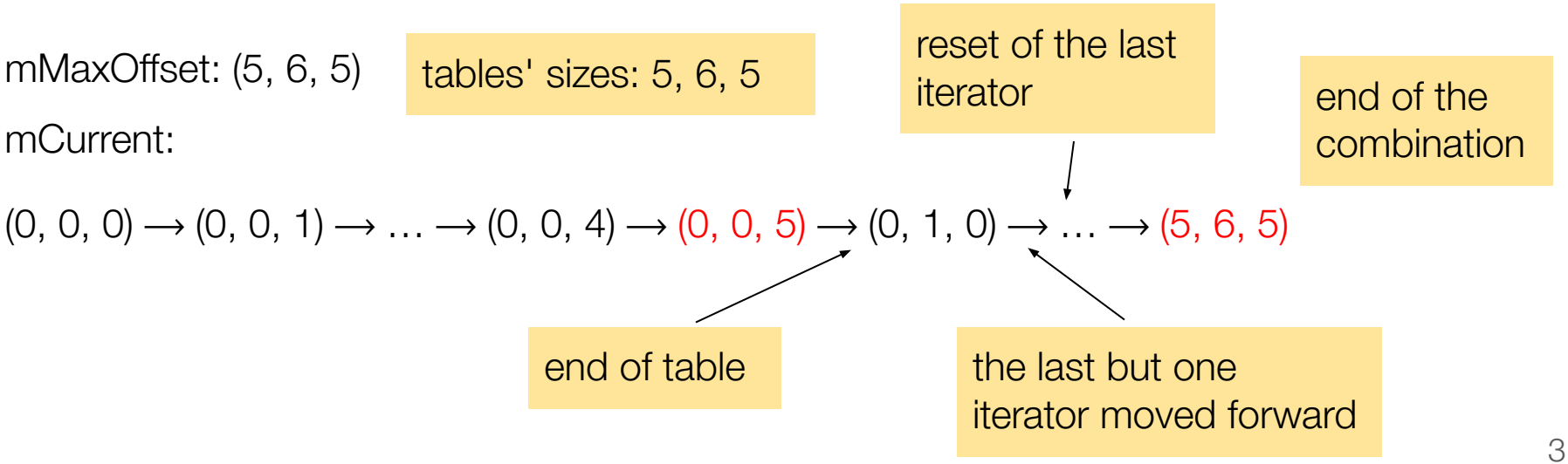
In this poster:
Run 3: **many more** collisions → **big** memory and time **overhead**
Idea: **lazy** generation (one at time) of combinations of elements, **without data copies**
mixed-event pairs: **binned combinations** of collisions + full track combinations
Universal – any n-tuple, any table. Time and memory performance **promising**

How to implement combinations **effectively**?

combinations – pairs, triples, … of elements from a table or different tables

Memory to store all tuples: **O(n!)** where n is the table size –> **too much!**
–> **Lazy** generation – one tuple by one

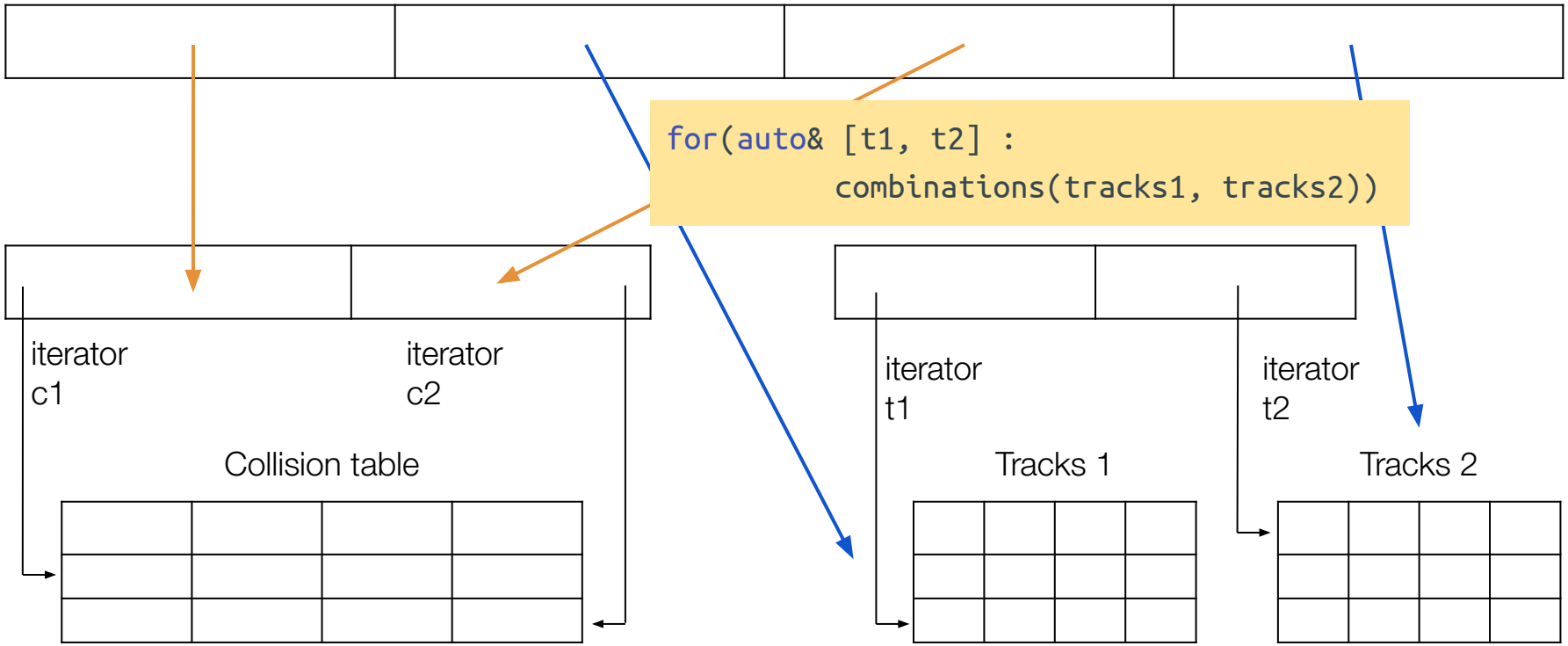
iterator – refers to a certain row in a table



Event mixing

```
BinningPolicy<collision::PosX, collision::PosY> binning{{xBins, yBins}};
SameKindPair<aod::Collisions, aod::Tracks> pair{binning};
for (auto& [c1, tracks1, c2, tracks2] : pair)
```

`tracks1` (`tracks2`) contains only tracks from the collision `c1` (`c2`)



How much time does it take?

- naive looping: AliPhysics algorithm re-implemented in O2 benchmark with O2 tables
- both algorithms have **linear complexity** w.r.t. number of collision pairs
- 8 x Intel® Core™ i5-8250U CPU
- on average 1000 tracks / collision
- 10 benchmark repetitions
- event mixing buffer size of 5

Same or **shorter** processing time but with **many more** functionalities:

- **any tuple**, not only a pair/triple
- **any input tables**, not just collisions and tracks
- **different elements** in a tuple, e.g., tracks-V0s
- user can supply his own **customized binning** class
- various combination policies

More code optimization ongoing.