HF O2 hackathon: skim creator, size estimation, triggering

ALICE

Fabrizio Grosa, Nima Zardoshti
CERN

HF-O2 hackathon
9 December 2021

fgrosa@cern.ch
nzardosh@cern.ch

Motivation

- Analysis code has to cope with much larger data size in Run3 compared to Run2
  - ➡ pp: x100 higher luminosity expected (~5 pb^{-1} MB@5.5 TeV, ~1 pb^{-1} MB@13.5TeV, ~200 pb^{-1} triggered@13.5TeV)
  - ➡ Pb–Pb: x50 higher luminosity expected (~5 nb^{-1} @5.5TeV)

- Implication:
  - ➡ Analysis code should be faster to be able to process all the collected statistics in a reasonable time
  - ➡ Reduction of data size: partly from O2 itself, but effort from HF side needed as well
  - ➡ In pp, event trigger on "rare" signals since storage is limited

What we discuss today?

- Strategy for creation and storage of charm-hadron candidates for further analysis
  - ➡ https://github.com/AliceO2Group/O2Physics/blob/master/PWGHF/TableProducer/HFTrackIndexSkimsCreator.cxx
- Strategy for event filtering (trigger) for the high-energy pp program
  - ➡ https://github.com/AliceO2Group/O2Physics/blob/master/EventFiltering/PWGHF/HFFilter.cxx

Creation of charm-hadron candidates in AliPhysics (Run2)

fgrosa@cern.ch
nzardosh@cern.ch

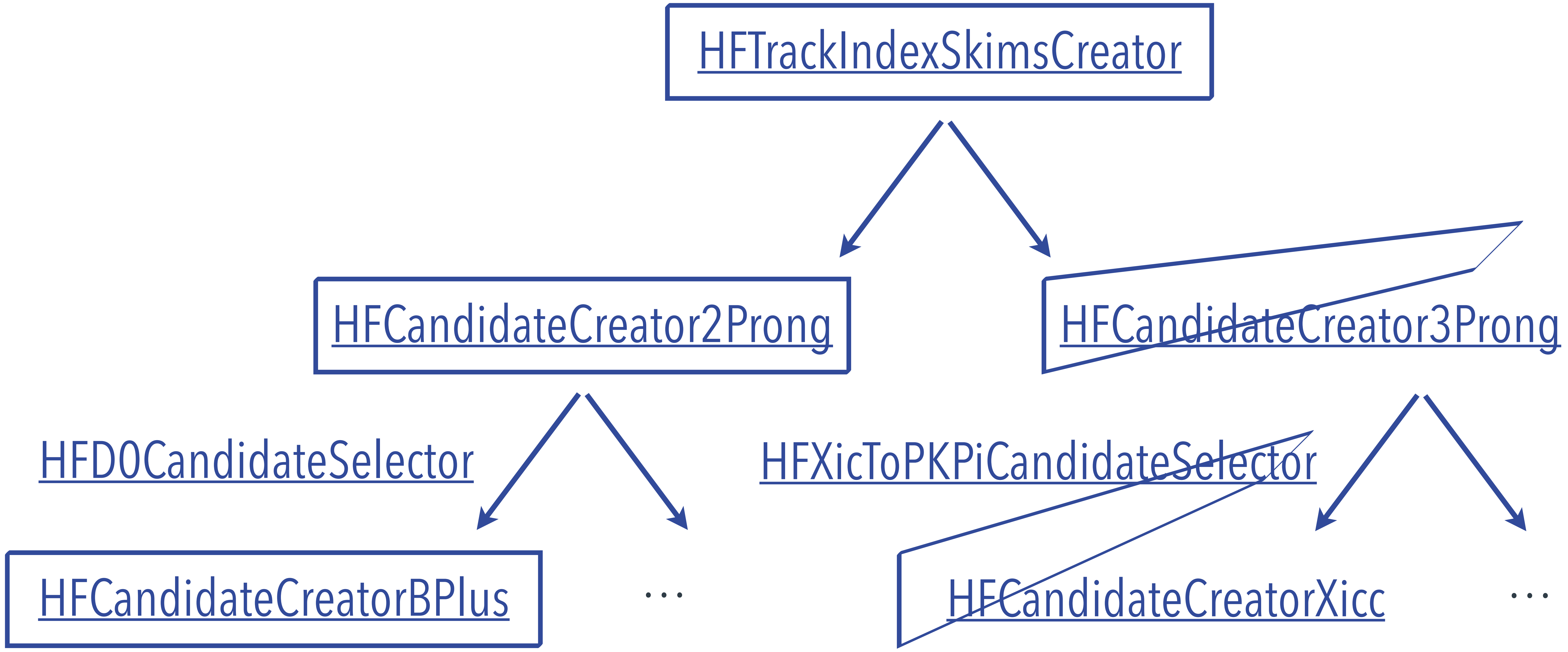
- Charm-hadron candidates created at the AOD filtering level
  - ➡ https://github.com/alisw/AliPhysics/blob/master/PWGHF/vertexingHF/AliAnalysisVertexingHF.cxx
  - ➡ Combinatorial of 2 and 3 tracks and vertex reconstruction
  - ➡ Some preselections applied to reduce output size and reduce CPU time of further analyses that will loop over the arrays of charm hadrons

- Storage of charm-hadron candidates in "delta AOD" files (AliAOD.VertexingHF.root)
  - ➡ AliAOD.VertexingHF.root files read together the standard AliAOD.root files (friend TTrees)
    - Necessity to match events in AliAOD.root and AliAOD.VertexingHF.root to assign the charm hadrons to the correct event
  - ➡ pp collisions: full information stored (i.e. objects containing all the kinematic and topological information)
    - pros: faster analysis (no need to recompute any quantity at the analysis task level)
    - cons: larger data size
  - ➡ Pb–Pb collisions: only doublets/triplets of track indices stored for each candidate
    - pros: smaller data size
    - cons: slower analysis (necessity to recompute the secondary vertex and associated quantities in the analysis task)

Creation of charm-hadron candidates in O2 (Run3 and beyond)

fgrosa@cern.ch
nzardosh@cern.ch

- Two steps:
  1. Combinatorial of 2 and 3 tracks and production of tables with doublets/triplets of track indices stored based on prefiltering on 2/3 prong candidates (HFTrackIndexSkimsCreator)
  2. Creation tables of full candidates with full kinematic and topological information (HFCandidateCreator*)



Inputs: tables of collisions, tracks
Outputs: tables with track-indices

Inputs: tables with track-indices
Outputs: tables with candidates

Inputs: tables with candidates, collisions
Outputs: tables with more complex candidates

Creation of charm-hadron candidates in O2 (Run3 and beyond)

fgrosa@cern.ch
nzardosh@cern.ch

- Two steps:
  1. Combinatorial of 2 and 3 tracks and production of tables with doublets/triplets of track indices stored based on prefiltering on 2/3 prong candidates (HFTrackIndexSkimsCreator)
  2. Creation tables of full candidates with full kinematic and topological information (HFCandidateCreator*)

HFTrackIndexSkimsCreator

Inputs: tables of collisions, tracks
Outputs: tables with track-indices

DISCUSSED IN THIS PRESENTATION

HFCandidateCreator2Prong

HFCandidateCreator3Prong

Inputs: tables with track-indices
Outputs: tables with candidates

HFD0CandidateSelector

HFXicToPKPiCandidateSelector

HFCandidateCreatorBPlus

…

HFCandidateCreatorXicc

…

Inputs: tables with candidates, collisions
Outputs: tables with more complex candidates

Skims-creator: event selection

fgrosa@cern.ch
nzardosh@cern.ch

- First step: tag "good" collisions HfTagSelCollision
  - ➡ Consumes collision table and, if enabled, event selection table

```cpp
/// Event selection with trigger selection
void processTrigSel(soa::Join<aod::Collisions, aod::EvSels>::iterator const& collision)
```

```cpp
/// Event selection without trigger selection
void processNoTrigSel(aod::Collision const& collision)
```

  - ➡ Tests if collisions are good or not based on the configured selections

```cpp
    // trigger selection
    if (!collision.alias()[triggerClass]) {
      SETBIT(statusCollision, EventRejection::Trigger);
      if (fillHistograms) {
        registry.fill(HIST("hEvents"), 3 + EventRejection::Trigger);
      }
    }

    // vertex selection
    selectVertex(collision, statusCollision);
```

  - ➡ Produces a table with a flag for each collision

```cpp
  Produces<aod::HFSelCollision> rowSelectedCollision;
```

```cpp
    // fill table row
    rowSelectedCollision(statusCollision);
```

Skims-creator: track selection

fgrosa@cern.ch
nzardosh@cern.ch

- Second step: tag “good” tracks HfTagSelTracks
  - ➡ Consumes track and collision tables

```cpp
  void process(aod::Collisions const& collisions,
               MY_TYPE1 const& tracks
```

  - ➡ Tests if tracks are good or not based on the configured selections (quality cuts, p_{T}, η, impact parameter)

```cpp
      // DCA cut
      array<float, 2> dca{track.dcaXY(), track.dcaZ()};
      if ((debug || statusProng > 0)) {
        if ((debug || TESTBIT(statusProng, CandidateType::Cand2Prong)) && !isSelectedTrack(track, dca, CandidateType::Cand2Prong)) {
          CLRBIT(statusProng, CandidateType::Cand2Prong);
          if (debug) {
            //cutStatus[CandidateType::Cand2Prong][3] = false;
            if (fillHistograms) {
              registry.fill(HIST("hRejTracks"), (nCuts + 1) * CandidateType::Cand2Prong + iDebugCut);
            }
          }
        }
```

  - ➡ Produces a table with a flag for each track

```cpp
  Produces<aod::HFSelTrack> rowSelectedTrack;

      // fill table row
      rowSelectedTrack(statusProng, track.px(), track.py(), track.pz());
```

Skims-creator: combinatorial for 2-prong and 3-prong candidates

fgrosa@cern.ch
nzardosh@cern.ch

- Third step: compute combinatorial [HfTrackIndexSkimsCreator]
  - ➡ Consumes filtered collisions and tracks
  - ➡ Loop over pairs (triplets) of tracks and apply preselections before vertex and after vertex
  - ➡ Produces a table with indices of daughter tracks for each selected candidate)

```
Filter filterSelectCollisions = (aod::hf_selcollision::whyRejectColl == 0);
Filter filterSelectTracks = aod::hf_seltrack::isSelProng > 0;
```

```
void process(
  SelectedCollisions::iterator const& collision,
  aod::BCs const& bcs,
  SelectedTracks const& tracks)
```

```
for (auto trackPos1 = tracks.begin(); trackPos1 != tracks.end(); ++trackPos1) {
  if (trackPos1.signed1Pt() < 0) {
    continue;
  }
```

```
for (auto trackNeg1 = tracks.begin(); trackNeg1 != tracks.end(); ++trackNeg1) {
  if (trackNeg1.signed1Pt() > 0) {
    continue;
  }
```

```
// 2-prong preselections
is2ProngPreselected(trackPos1, trackNeg1, cutStatus2Prong,
```

```
// 2-prong selections after secondary vertex
is2ProngSelected(pVecCandProng2, secondaryVertex2,
```

```
Produces<aod::HfTrackIndexProng2> rowTrackIndexProng2;
```

```
rowTrackIndexProng2(trackPos1.globalIndex(),
                    trackNeg1.globalIndex(), isSelected2ProngCand);
```

Skims-creator: data size estimation

fgrosa@cern.ch
nzardosh@cern.ch

- Content of a standard AO2D.root

- AO2D.root
  - DF_2449180042955382502;1
    - O2collision;1
    - DbgEventExtra;1
    - O2track;1
    - O2trackcov;1
    - O2trackextra;1
    - O2fwdtrack;1
    - O2fwdtrackcov;1
    - O2calo;1
    - O2calotrigger;1
    - O2zdc;1
    - O2fv0a;1
    - O2fv0c;1
    - O2ft0;1
    - O2fdd;1
    - O2v0;1
    - O2cascade;1
    - O2bc;1
    - O2run2bcinfo;1
    - O2origin;1
    - O2hmpid;1

- Content of a derived AO2D.root with the tables produced by the [HfTrackIndexSkimsCreator]() and the [HFCandidateCreator2Prong]() and [HFCandidateCreator3Prong]()

- AO2D_HF.root
  - DF_2449180042955382500;1
    - O2hfselcollision;1
    - O2hftrackidxp2;1
    - O2hfcandp2ext;1
    - O2hfcandp2base;1
    - O2hfseltrack;1
    - O2hftrackidxp3;1
    - O2hfcandp3ext;1
    - O2hfcandp3base;1

- Using Pb-Pb Run2 converted data (few files) and applying Run2 like filtering selections

Size estimates	Ratio to AO2D (uncompressed)	Ratio to AO2D (compressed)
O2hftrackidxp2	0.03	0.02
O2hftrackidxp3	0.30	0.17
O2hfcandp2base+ext	0.36	0.48
O2hfcandp3base+ext	3.18	3.97

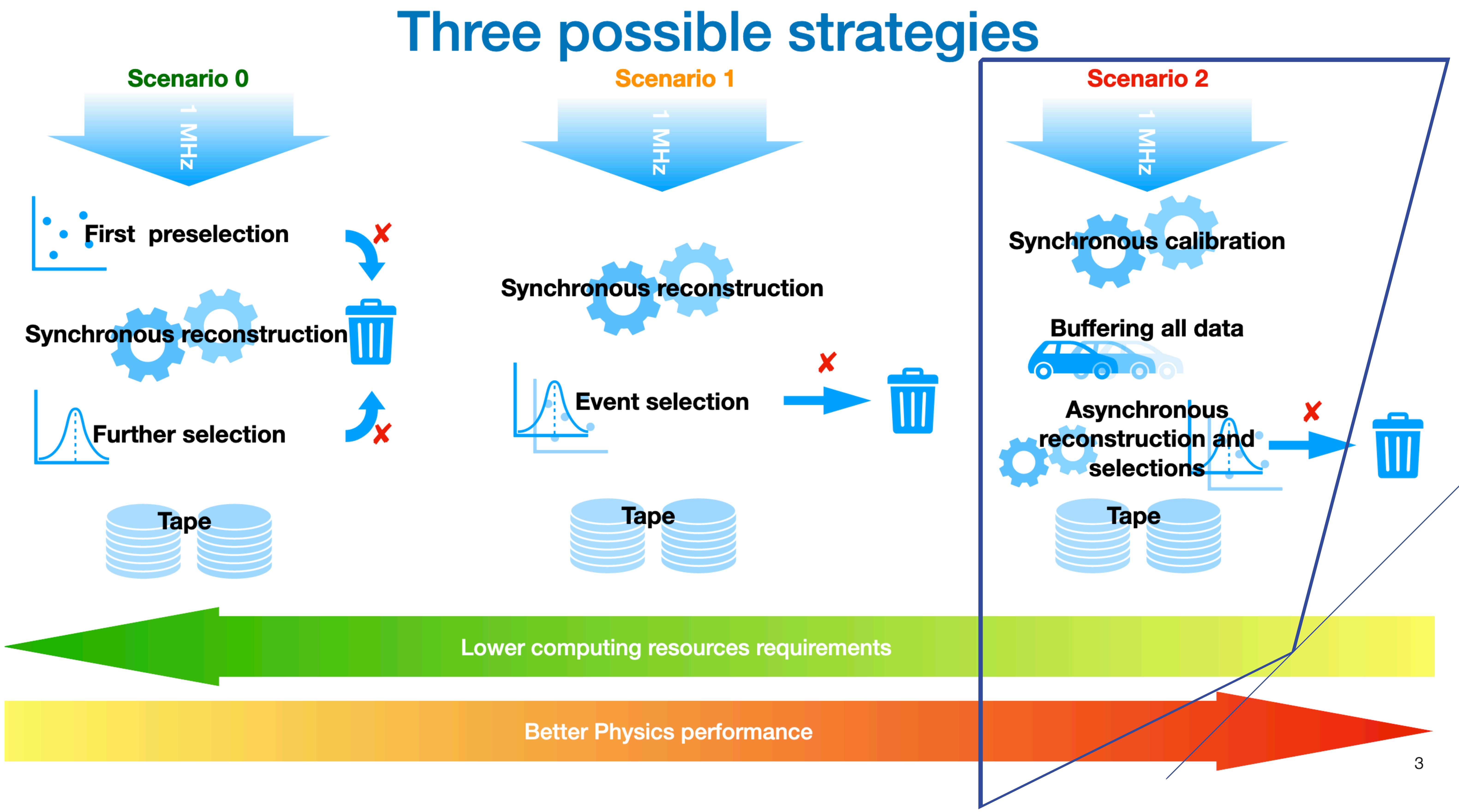
➡ About 20% of the AO2D.root size if we consider the table with the daughter-track indices

➡ More than the AO2D.root size when considered the candidate tables

➡ In both cases, HF tables should be joined with tables in the standard AO2Ds

- Two possible solutions: "delta" files or self contained skimmed AO2Ds

fgrosa@cern.ch
nzardosh@cern.ch



- Scenario 2 chosen as strategy for pp triggers
  - ➡ Synchronous calibration, buffer the data, asynchronous reconstruction and trigger
  - ➡ Code for the pp triggers in the analysis framework (O2Physics/EventFiltering) can make use of all the information after the reconstruction

- Expected interaction rate: 1MHz
- Target selection rate: 100 Hz
  - ➡ Rejection factor 10^4 → only triggers on "rare" probes can be defined

Physics channels and strategy decided in the trigger meetings with the trigger coordination

HF trigger general strategy

fgrosa@cern.ch
nzardosh@cern.ch

HFTrackIndexSkimsCreator

↓

HFFilters

- Use HFTrackIndexSkimsCreator to build 2-prong and 3-prong charm candidates as in the analysis
  ➡ Same code by design
- HFFilters consumes collision, track, and 2-prong and 3-prong track indices tables

```
void process(aod::Collision const& collision,
             HfTrackIndexProng2withColl const& cand2Prongs,
             HfTrackIndexProng3withColl const& cand3Prongs,
             BigTracksWithProtonPID const& tracks)
```

Events with high-p_T charm hadrons

Events with beauty hadrons

Events with p–(charm hadron) pairs with low relative momentum

Events with more than one charm-hadron candidate

- Other than the presence of at least a charm-hadron candidate associated to a collision, it tests other conditions and it tags the event to be kept or not

```
tags(keepEvent[kHighPt], keepEvent[kBeauty], keepEvent[kFemto], keepEvent[kDoubleCharm]);
```

HF trigger example: beauty hadrons

fgrosa@cern.ch
nzardosh@cern.ch

- Idea: look for events that contain at least one beauty-hadron candidate in the following decay channels:
  - ➡ $B^+ \rightarrow \bar{D}^0\pi^+$, $B^0 \rightarrow D^-\pi^+$, $B_s^0 \rightarrow D_s^-\pi^+$, $\Lambda_b^0 \rightarrow \Lambda_c^+\pi^-$

- Loop over 2-prong and 3-prong charm-hadron candidate

```
      for (const auto& cand2Prong : cand2Prongs) { // start loop over 2 prongs

        if (!TESTBIT(cand2Prong.hfflag(), o2::aod::hf_cand_prong2::DecayType::D0ToPiK)) {
          continue;
        }
```

- For each candidate, loop over tracks and apply selection criteria (quality, p_T, impact parameter)
  - ➡ If track selected compute invariant mass with charm hadron and test if it's compatible with the corresponding b hadron

```
      for (const auto& track : tracks) { // start loop over tracks

        if (track.globalIndex() == trackPos.globalIndex() || track.globalIndex() == trackNeg.globalIndex()) {
          continue;
        }

        std::array<float, 3> pVecThird = {track.px(), track.py(), track.pz()};

        if (!keepEvent[kBeauty]) {
          if (isSelectedTrackForBeauty(track, kBeauty3Prong)) {
            auto massCandB = RecoDecay::M(std::array{pVec2Prong, pVecThird}, std::array{massD0, massPi});
            if (std::abs(massCandB - massBPlus) <= deltaMassBPlus) {
              keepEvent[kBeauty] = true;
              if (activateQA) {
                hMassB[kBplus]->Fill(massCandB);
              }
            }
          }
        }
```

- As soon as the first b-hadron candidate is found, the event is tagged as good for b hadrons and no candidates are searched anymore

Summary

fgrosa@cern.ch
nzardosh@cern.ch

- HFTrackIndexSkimsCreator
  - ➡ Code that tags 2-3 prong charm-hadron candidates in O2
  - ➡ Discussion on which information to save on disk to avoid computation of combinatorial every analysis ongoing
  - ➡ Optimisation of the code (memory management, execution time) ongoing
  - ➡ Missing pieces in selections under development
- HFFilters
  - ➡ Triggers defined and basic implementation in O2
  - ➡ Optimisation of the code (memory management, execution time) ongoing
  - ➡ Development of more refined selection strategies under development

- Working group parallel session: Skim creator, size estimation, triggering
  - ➡ Starts at 14:30 (CERN time)
  - ➡ Material: https://codimd.web.cern.ch/s/GfPxZBVxh#
  - ➡ Zoom link: https://cern.zoom.us/u/cbjuchiXvt

15:00 → 17:00 **Working groups, parallel sessions: Skim creator, size estimation, triggering**

Conveners: Fabrizio Grosa (CERN), Nima Zardoshti (University of Birmingham (GB))

CodiMD link ZOOM link

ADDITIONAL SLIDES

Rejection factors for b-hadron triggers (AliPhysics with improver)

fgrosa@cern.ch
nzardosh@cern.ch