
DPG

MC-to-MC embedding

ALICE • Data

Software and Computing Week, 03 April 2019

Preparation

C. Zampolli for DPG

Group

MC productions for Pb-Pb: embedding

- Technique which allows to add a **MC signal event on top of a previously generated MC background event**
 - To allow to save **CPU** → MCs ready faster
 - To allow to save **disk** space → always a critical issue
- Two techniques:
 - ➔ – **Local embedding**: generate 1 background event and reuse it for consecutive events
 - Tested by LF (**ongoing**), HF (**validated**), DQ (**validated**)
 - **Global embedding**:
 - Transfer 1 BKG event over the network and reuse it for several signal events
 - Test was performed, under check the analysis of the output with Lego trains
 - Performance comparison between local and global approach:
 - Disk space comparable (but in global mode, Digits have to stay till we don't want to reuse the BKG production anymore)
 - CPU in global is lower in case the BKG production is re-used several times, otherwise they are equivalent (ignoring the transfer of the BKG event over the network)

→ Preference depends on whether we can profit from a common BKG production (not for now)
- Further optimization (in terms of extra disk space saved):
 - **Extraction of signal-only information** to be stored in the ESDs - **signal filtering**
 - Possibility to save decays/V0s with at least one track from background - **mixed mode**

Current use of embedding

- HF productions anchored to PbPb 2018 used embedding (LHC18c[1, 2, 3][a, b])
 - Central cycles: background event reused 4x
 - Semi-central cycles: background reused 12x
- Sizes:
 - LHC19c2c** (standard D2H production without MC-to-MC embedding, 30-50%, Pythia (Lc → K0s+p) injected in HIJING) - 12 events per job
 - LHC19d1b** (production with **MC-to-MC embedding**, signal filtering, no mixed mode, 30-50%, Pythia (Lc → K0s+p) injected in HIJING, background reused 12x (equivalent to 19c2c configuration))

} Limited by running time and CPU consumption

	#averaged files	Kine (bkg)	Kine (signal)	Kine (bkg + signal)	ESDs	ESDfriends	AODs	AOD VertexingHF	AOD Muons	AOD GConv
LHC19c2c (CPU: 49 min/ev)	2419	-	-	102 MB	49 MB	2.4 MB (?)	31 MB	3.3 MB	1.8 MB	1.6 MB
LHC19d1b (CPU: 18 min/ev)	538	8.2 MB (on 2067 files)	5.4 MB	13.6 MB	8.1 MB	2.9 MB (?)	3. MB	155 kB	323 kB	42 kB

1 HIJING semi-central event

PYTHIA events (number depends on b)

12 HIJING semicentral + PYTHIA events (number depends on b)

signal filtering does not affect ESDfriends

Software-related implications of MC-to-MC embedding

- **Recommendation:** Access to Monte Carlo information in ESDs should not be done via Stack and TParticle
 - users must use the **AliMCEvent::GetTrack(i)** method, which returns an AliVParticle (to be casted to AliMCParticle or AliAODMCParticle for ESD or AOD respectively)
 - users must use **AliMCParticle** to get the mother/daughter
- Possibility to know if a particle comes from background **available**
- Possibility to count how many times an event has been used (to avoid to reuse background events, as they will naturally be present several times if you don't filter them out - which is an option of the embedding) **available**
- Helper methods to retrieve from AliMCEvent the mother of a particle, its label, and the labels of the daughter (without going through the Stack) **available**

Signal filtering / Mixed mode

- Calo information not filtered (upon request of EMCAL experts)
 - When using signal-filtered MCs, since we (currently) remove all reference to background, the access to the background event will not work
- Signal filtering does not change the “global” properties of the event (tracklets, V0...)
 - Centrality, Multiplicity, primary vertex are “untouched”
 - *BUT: with signal filtering, we lose the information on the background event generator → currently, AliMultSelectionTask does not auto-detect correctly the OADB to be used - fix ongoing*
- Mixed mode = possibility to keep tracks built from at least one particle from the background
 - To have this mode, we need to keep the Kinematics.root from the background for the analysis that use ESDs → reduced gain in space saving

Conclusions

- MC-to-MC procedure validated and used
- Impact on analysis communicated to the analyzers, e.g.:
 - need to change the way the MC truth is accessed when using ESDs
 - OADB for AliMultSelectionTask should be (for now) set manually
- **MC-to-MC embedding is strongly encouraged** to save CPU and disk resources