# LCFIPlus basics

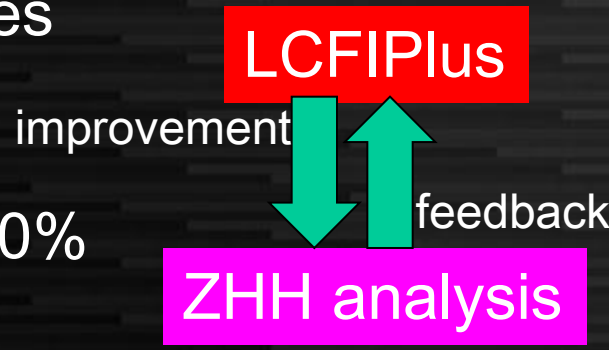
T. Suehara, T. Tanabe
(ICEPP,  U. Tokyo)

# Direction of LCFIPlus development

**LCFIVertex** The first realistic flavor tagging in ILC

- Incorporating modern flavor tagging techniques to obtain reasonable performance
- No other algorithms to be compared…
- Mainly tuned with Z-pole qqbar samples

**LCFIPlus** Our second version




- Clear target: Higgs self-coupling to ~30%
  - ➡ high demand for performance
- Focused on >=4 jet environments
- Including jet clustering (performance driver for 6-jets)
- Trying many ideas for performance improvement

**LCFIPlus is more performance-driven, mainly concentrated on many-jet processes**

# Data/process flow

All in “lcfiplus” namespace

## EventStore

singleton for data pool

vector<Track *>
vector<Neutral *>
vector<MCParticle *>
vector<Vertex *>
vector<Jet *>
any other types

- Automatic type identification
  (Allow one name with multiple types)
- Automatic creation/deletion
  (using ROOT class dictionary)

## LCIO

## LCIOStorer

- Automatic conversion from LCIO to lcfiplus classes (using hook in EventStore)
- Conversion to LCIO is manually invoked by LcfiplusProcessor

configuration

## Algorithm

PrimaryVertex JetVertexRefiner
BuildUpVertex FlavorTag TrainMVA
JetClustering MakeNtuple ReadMVA etc.

- Parameters class used for type-safe configuration

## LcfiplusProcessor

- Marlin processor
- Process Marlin parameters to be passed to Algorithm
- LCIO I/O configuration

Internal algorithms

Independent

Marlin

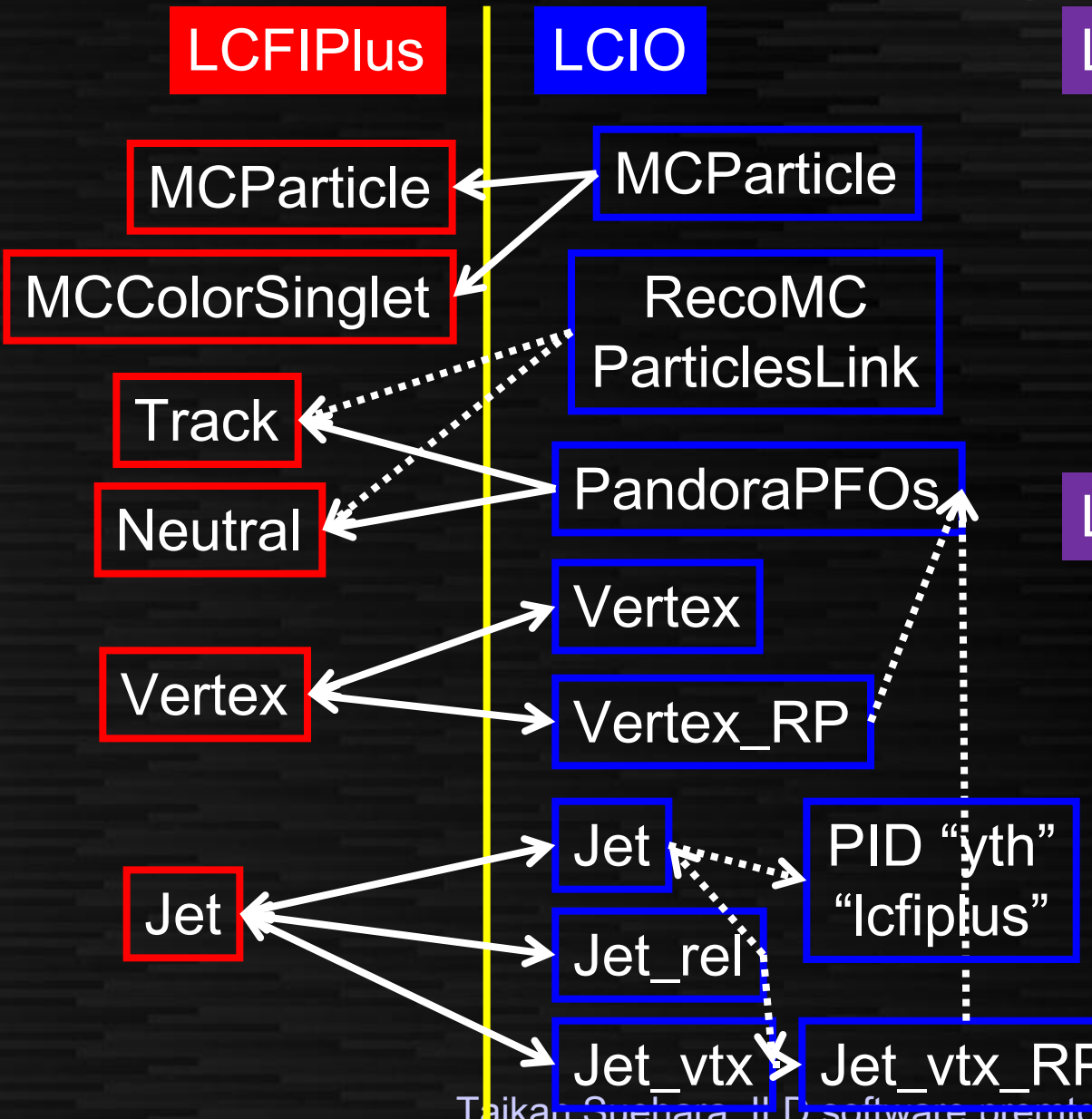
# Outline

- EventStore (not included)
- Algorithm (not included)
- LCIO I/O (LCIOStorer)
- LcfiplusProcessor
- Each algorithm
  - Vertex finder (primary/buildup)
  - Jet clustering
  - JetvertexRefiner
  - Flavor tagging
- Comments

Performance study is not included in this talk: please refer to Tomohiko's talk in the main ILD workshop

Try to provide information to use full functions of LCFIPlus Tomohiko's talk for simple usage

# LCIO input/output (LCIOStorer)



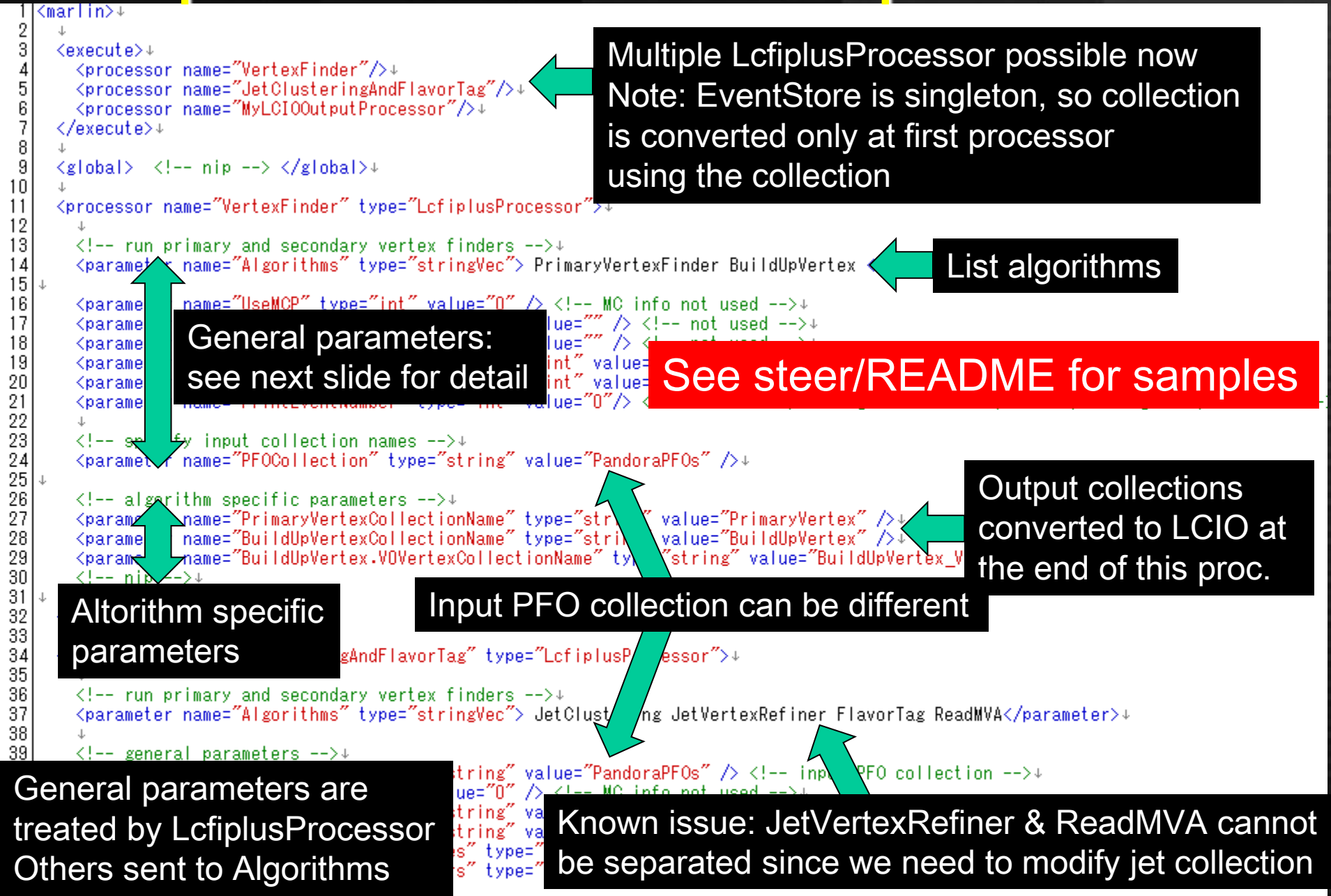

## LCIO -> LCFIPlus

- PFO/MCP conversion at SetEvent() (called by processor)
- Vertex/Jet conversion automatically at request of lcfiplus collection

## LCFIPlus -> LCIO

- Done manually (automatic option available but not used in LcfiplusProcessor)
- Cannot "update" existing LCIO collection – known issue in PID
- Jet – vertex connection is in LCRelation Jet_rel

# LcfiplusProcessor and parameters



# LcfiplusProcessor: global params

To use edep in subdetectors for jet-muon detection: assume ILD so specify 0 for SiD

List algorithms

```
13      <!-- r  primary and secondary vertex finders -->
14      <param  er name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex
15
16      <param  er name="UseMCP" type="int" value="0"
17      <param  er name="MCPCollection" type="string" value
18      <param  er name="MCPFORelation" type="string" value=   /> <!-- not used -->
19      <parameter name="ReadSubdetectorEnergies" type="int" value="1"/> <!-- true for IL
20      <parameter na  "UpdateVertexRPDaughters" type="int" value="0"/> <!-- false for r
21      <parameter na  e="PrintEventNumber" ty  ="int" value="0"/> <!-- 0 for not printing
22
23      <!-- specif  input collection na  s -->
24      <parameter  ame="PFOCollection"  ype= tring" value="PandoraPFOs" />
25
```

UseMCP = 0: do not convert MCParticle

These two are global: do specify the same for all LcfiplusProcessor

Modify “StartVertex” in PFO collection or not
1 for mass production, 0 for user analysis

Print current event number every n events
Good for debug

Input PFO collection: can be different in each LcfiplusProcessor

# Algorithm (1) vertex finders

- VertexFitterSimple
  - Original vertex fitter with Minuit2 in ROOT
  - Do not refit tracks
  - Slow – may need to improve
- PrimaryVertexFinder
  - TearDown method with VertexFitterSimple
  - Use beam vertex (fixed parameters now)
- BuildUpVertex
  - Secondary vertex finder with VertexFitterSimple
  - Tuned for JetClustering (many cuts included)
  - Recent improvement in V0 rejection

# Vertex finder steering file

steer/vertex.xml

```
<processor name="VertexFinder" type="LcfiplusProcessor">
  <!-- run primary and secondary vertex finders -->
  <parameter name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex </parameter>
  <parameter name="ReadSubdetectorEnergies" type="int" value="1"/> <!-- true for ILD -->
  <parameter name="UpdateVertexRPDaughters" type="int" value="0"/> <!-- false for non-updative Pandor
  <parameter name="PrintEventNumber" type="int" value="0"/> <!-- 0 for not printing event number, n f
  <!-- specify input collection names -->
  <parameter name="PFOCollection" type="string" value="PandoraPFOs" />
  <parameter name="PrimaryVertexCollectionName" type="string" value="PrimaryVertex" />
  <parameter name="BuildUpVertexCollectionName" type="string" value="BuildUpVertex" />
  <parameter name="BuildUpVertex.V0VertexCollectionName" type="string" value="BuildUpVertex_V0" />
  <!-- parameters for primary vertex finder -->
  <parameter name="PrimaryVertexFinder.TrackMaxD0" type="double" value="20." />
  <parameter name="PrimaryVertexFinder.TrackMaxZ0" type="double" value="20." />
  <parameter name="PrimaryVertexFinder.TrackMaxInnermostHitRadius" type="double" value=
  <parameter name="PrimaryVertexFinder.TrackMinVtxFtdHits" type="int" value="5" />
  <parameter name="PrimaryVertexFinder.Chi2Threshold" type="double" value="25." />
  <!-- parameters for secondary vertex finder -->
  <parameter name="BuildUpVertex.TrackMaxD0" type="double" value="10." />
  <parameter name="BuildUpVertex.TrackMaxZ0" type="double" value="20." />
  <parameter name="BuildUpVertex.TrackMinPt" type="double" value="0.1" />
  <parameter name="BuildUpVertex.TrackMaxD0Err" type="double" value="0.1" />
  <parameter name="BuildUpVertex.TrackMaxZ0Err" type="double" value="0.1" />
  <parameter name="BuildUpVertex.TrackMinTpcHits" type="int" value="20" />
  <parameter name="BuildUpVertex.TrackMinFtdHits" type="int" value="3" />
  <parameter name="BuildUpVertex.TrackMinVxdHits" type="int" value="3" />
  <parameter name="BuildUpVertex.TrackMinVxdFtdHits" type="int" value="0" />
  <parameter name="BuildUpVertex.PrimaryChi2Threshold" type="double" value="25." />
  <parameter name="BuildUpVertex.SecondaryChi2Threshold" type="double" value="9." />
  <parameter name="BuildUpVertex.MassThreshold" type="double" value="10." />
  <parameter name="BuildUpVertex.MinDistFromIP" type="double" value="0.3" />
  <parameter name="BuildUpVertex.MaxChi2ForDistOrder" type="double" value="1.0" />
  <parameter name="BuildUpVertex.AssocIPTracks" type="int" value="1" />
  <parameter name="BuildUpVertex.AssocIPTracksMinDist" type="double" value="0." />
  <parameter name="BuildUpVertex.AssocIPTracksChi2RatioSecToPri" type="double" value="2.0" />
  <parameter name="BuildUpVertex.UseV0Selection" type="int" value="1" />
</processor>
```

V0 collection is used in later algorithms

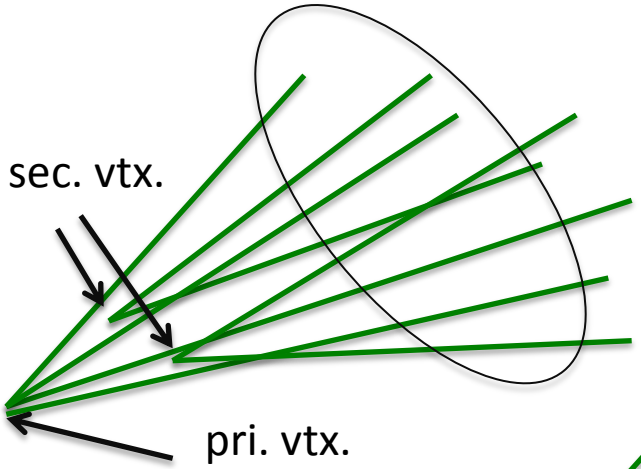
Parameters are highly tuned: please contact us if you need to modify them

# Algorithm(2) JetClustering

- Jet clustering with vertex information (see next slide)
- Using jet muons as vertex with UseMuonID = 1
  - Using edep in muon detector: only for ILD…
- Plane Durham is possible by specifying “0” for InputVertexCollectionName (do not specify “”) & UseMuonID=0
- Multiple output collections can be done such as NJetsRequested = 8 6 4, (must be descending order), OutputJetCollectionName = Jets8 Jets6 Jets4
- Add ParticleID yth with y23, y34,…, y910 parameters for ycuts
  - Issue: yn(n+1) is obtained only if NJetRequested <= n is done

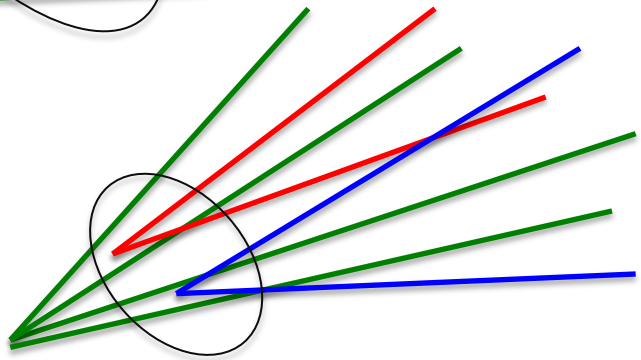
```
<!-- jet clustering parameters -->↓
<parameter name="JetClustering.InputVertexCollectionName" type="string" value="BuildUpVertex" />
<parameter name="JetClustering.OutputJetCollectionName" type="stringVec" value="VertexJets" /> <!
<parameter name="JetClustering.NJetsRequested" type="intVec" value="2" /> <!-- Multiple NJets can
↓
<parameter name="JetClustering.YCut" type="doubleVec" value="0." /> <!-- specify 0 if not used --
<parameter name="JetClustering.UseMuonID" type="int" value="1" /> <!-- jet-muon ID for jet cluste
<parameter name="JetClustering.VertexSelectionMinimumDistance" type="double" value="0.3" /> <!--
<parameter name="JetClustering.VertexSelectionMaximumDistance" type="double" value="30." /> <!--
<parameter name="JetClustering.VertexSelectionKOMassWidth" type="double" value="0.02" /> <!-- in
↓
```

# Vertex-Jet Finding Overview

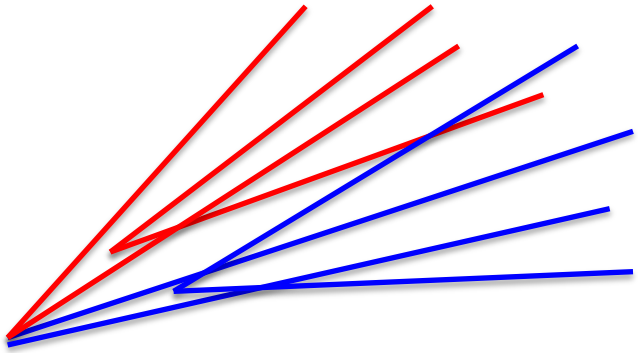



1. Difficult to separate two b-jets which are close. Ordinary kt algorithm tends to merge them.

arXiv:1110.5785



2. To overcome this, find secondary vertices first, and use them as *seeds* for jet finding.



3. This results in an increased chance of correct jet separation.

This effect is particularly pronounced for final states with many b jets, e.g. Zhh

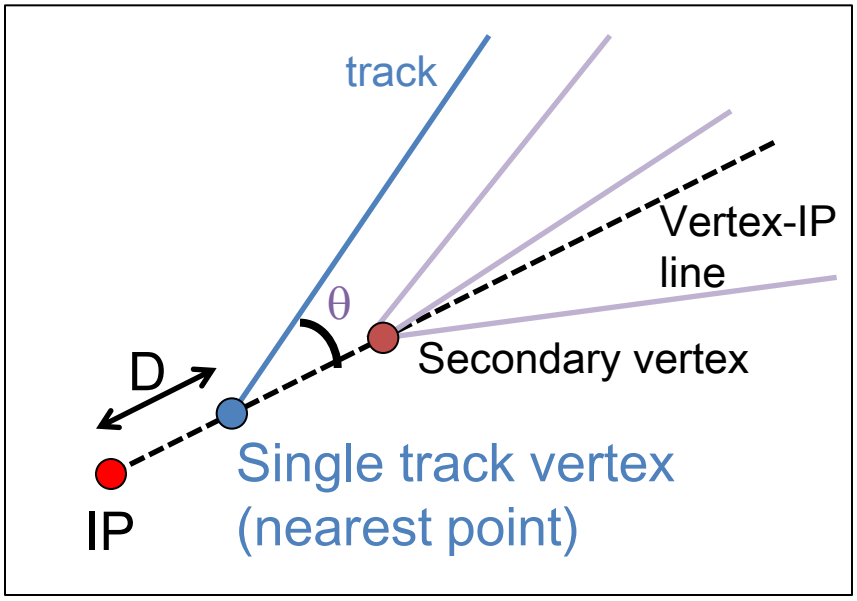
# Algorithm(3) JetVertexRefiner

- Consists of two algorithms
  - SingleTrackVertexFinder & VertexCombiner
- SingleTrackVertexFinder: reconstruct single-track vertices using existing vertex directions (see next slide)
- VertexCombiner: combine vertices into two at most aiming at combining multi+single vertices which are from same b or c – tuned for b/c separation
- Jet & vertex collection are specified separately, so this can be used after other jet clustering method (Durham etc.)

```
<!-- vertex refiner parameters -->
<parameter name="JetVertexRefiner.InputJetCollectionName" type="string" value="VertexJets" />
<parameter name="JetVertexRefiner.OutputJetCollectionName" type="string" value="RefinedJets" />
<parameter name="JetVertexRefiner.PrimaryVertexCollectionName" type="string" value="PrimaryVertex" />
<parameter name="JetVertexRefiner.InputVertexCollectionName" type="string" value="BuildUpVertex" />
<parameter name="JetVertexRefiner.VOVertexCollectionName" type="string" value="BuildUpVertex_VO" />
<parameter name="JetVertexRefiner.OutputVertexCollectionName" type="string" value="RefinedVertex" />

<parameter name="JetVertexRefiner.MinPosSingle" type="double" value="0.3" />
<parameter name="JetVertexRefiner.MaxPosSingle" type="double" value="30." />
<parameter name="JetVertexRefiner.MinEnergySingle" type="double" value="1." />
<parameter name="JetVertexRefiner.MaxAngleSingle" type="double" value="0.5" />
<parameter name="JetVertexRefiner.MaxSeparationPerPosSingle" type="double" value="0.1" />
<parameter name="JetVertexRefiner.mindOsigSingle" type="double" value="5." />
<parameter name="JetVertexRefiner.minzOsigSingle" type="double" value="5." />
<parameter name="JetVertexRefiner.OneVertexProbThreshold" type="double" value="0.001" />
<parameter name="JetVertexRefiner.MaxCharmFlightLengthPerJetEnergy" type="double" value="0.1" />

```

Parameters are highly tuned

# Single Track Selection




- Normal vertex finder needs > 2 tracks
  -> loose many vertices
- Single track vertex can be found by using other vertex direction
- Improves b-tagging performance

| Event | 0 vtx | 1 vtx | >= 2 vtx |
|---|---|---|---|
| bb normal | 322 | 1052 | 426(24%) |
| bb +single | 322 | 459 | 1019(57%) |
| cc normal | 1003 | 779 | 18(1.0%) |
| cc +single | 1003 | 715 | 82(4.6%) |

# Algorithm(4) flavor tagging

- Based on TMVA Boosted Decision Trees
  - MLP neural net under development
  - Four categories: #vtx = 0, 1, 1+singletrack, 2
- Algorithms
  - FlavorTag: obtain input variables
  - MakeNtuple: making ROOT ntuple for training
  - TrainMVA: training MVA with b/c/s ntuples
  - ReadMVA: obtain BTag/CTag variables with weight file
- Procedure
  1. FlavorTag + MakeNtuple for each training sample
  2. TrainMVA with all ntuples (output: weight file)
  3. FlavorTag + ReadMVA with the weight file

# Algorithm(4) input variables

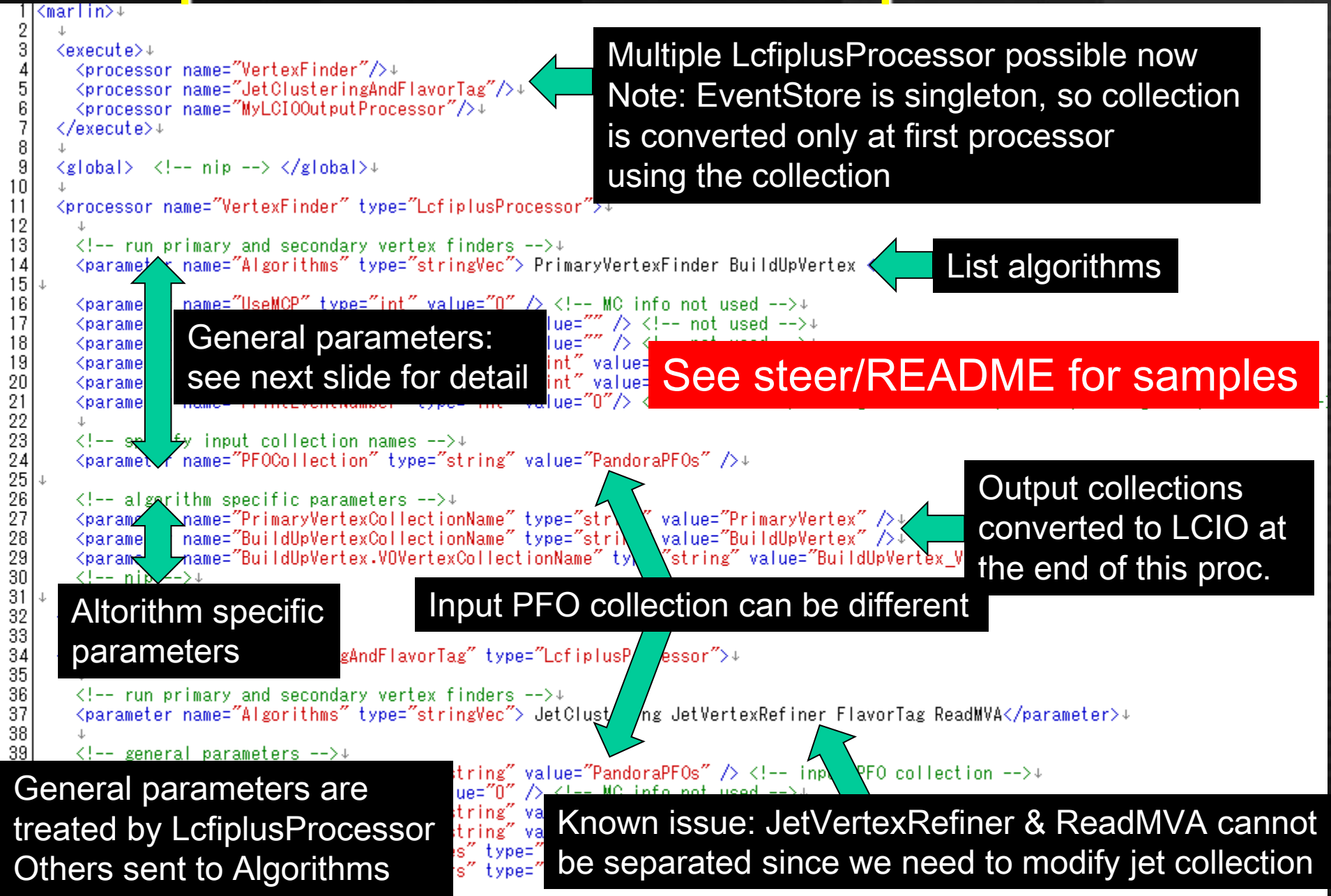
```
<parameter name="FlavorTag.CategoryDefinition1" type="string">nvtx==0</parameter>
<parameter name="FlavorTag.CategoryVariables1" type="stringVec">
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz
</parameter>
<parameter name="FlavorTag.CategorySpectators1" type="stringVec">
  aux nvtx
</parameter>

<parameter name="FlavorTag.CategoryDefinition2" type="string">nvtx==1&&nvtxall==1</parameter>
<parameter name="FlavorTag.CategoryVariables2" type="stringVec">
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob
</parameter>
<parameter name="FlavorTag.CategorySpectators2" type="stringVec">
  aux nvtx
</parameter>

<parameter name="FlavorTag.CategoryDefinition3" type="string">nvtx==1&&nvtxall==2</parameter>
<parameter name="FlavorTag.CategoryVariables3" type="stringVec">
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob
  1vtxprob 2ndflt vtxmassall
</parameter>
<parameter name="FlavorTag.CategorySpectators3" type="stringVec">
  aux nvtx
</parameter>

<parameter name="FlavorTag.CategoryDefinition4" type="string">nvtx>=2</parameter>
<parameter name="FlavorTag.CategoryVariables4" type="stringVec">
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob
  vtxlen2_jete vtxsig2_jete vtxdirang2_jete vtxmom2_jete vtxmass2 vtxmult2
  vtxlen12_jete vtxsig12_jete vtxdirang12_jete vtxmom_jete vtxmass vtxmult
</parameter>
<parameter name="FlavorTag.CategorySpectators4" type="stringVec">
  aux nvtx
</parameter>
```

May not be latest…

# Algorithm(4) misc

- Output in LCIO
  - PID in jet “lcfiplus” (can be specified in steering)
  - Parameter names “BTag”,”CTag”,”Category”
  - Values 0 – 1 (can be combined for all categories)
- Available trained weight files
  - Z-pole qqbar (bb / cc / (uu & dd & ss) )
  - 6-jet sample
    bbbbbb / cccccc / (uuuuuu & dddddd & ssssss)
    mainly from ZZZ production (not Higgs)
- Issue in ReadMVA to obtain PID in LCIO jet
  Need to be in the same Lcfiplus processor with JetVertexRefiner (which writes jet collection to LCIO)

# LcfiplusProcessor and parameters



# Documentation & Feedback

- Doxygen class reference
- User feedback + documentation system has been setup @ CERN (J. Strube):
  - Documentation wiki hosted at SLAC (thanks N. Graf)
    - bug tracker (JIRA) also available
  - https://confluence.slac.stanford.edu/display/ilc/LCFIPlus
    - some documentation present, will be described in more detail later
- Early bug reports (Thanks: J. Engels, F. Gaede, J. Strube, A. Sailer)
- Nightly builds and check input variables (J. Strube)
- Feedback and support from LC community has been very helpful. Will continue working with SiD for a smooth transition from LCFIVertex to LCFIPlus.

# Comments

- Many issues or strange “specification” remain
  - yth, Jet PID output, etc.
  - Many unknown issues (maybe)
    - Information very welcome
- Short in documentation!
  - This slides may be a little help…
- Top priorities
  - Code till vertex finder at least should be fixed before mass reconstruction
  - Better performance needed for ZHH

# おわり。

# Vertex finder performance

| (a) $ZHH \rightarrow qqbbbb$ | Track origin | | | |
|---|---|---|---|---|
| | Primary | $b$ hadron | $c$ hadron | Other |
| Number of all reconstructed tracks | 67575 | 12912 | 15246 | 4087 |
| Number of tracks used by ZVTOP | 1162 | 8534 | 10404 | 999 |
| ...in *good* vertices | - | 8248 | 10103 | - |
| Number of tracks used by our original vertex finder | 617 | 8717 | 10529 | 358 |
| ...in *good* vertices | - | 8551 | 10333 | - |

| (b) $t\bar{t} \rightarrow bbqqqq$ | Track origin | | | |
|---|---|---|---|---|
| | Primary | $b$ hadron | $c$ hadron | Other |
| Number of all reconstructed tracks | 74504 | 8945 | 12602 | 4219 |
| Number of tracks used by ZVTOP | 920 | 5999 | 8353 | 1024 |
| ...in *good* vertices | - | 5830 | 8137 | - |
| Number of tracks used by our original vertex finder | 420 | 6161 | 8447 | 341 |
| ...in *good* vertices | - | 6060 | 8279 | - |

good vertex: all tracks are from the same heavy hadron
(note: b & its daughter c are recognized as the same)
reconstruction before LCFIPlus was done with ilcsoft v01-11

Better than LCFIVertex vertex finder in ZHH/tt sample!