CEPC Software Framework

Ziyan Deng, Xingtao Huang, Gang Li, Weidong Li,
Tao Lin, Shengsen Sun, Manqi Ruan, Xiaomei Zhang, Jiaheng Zou.

huangxt@sdu.edu.cn

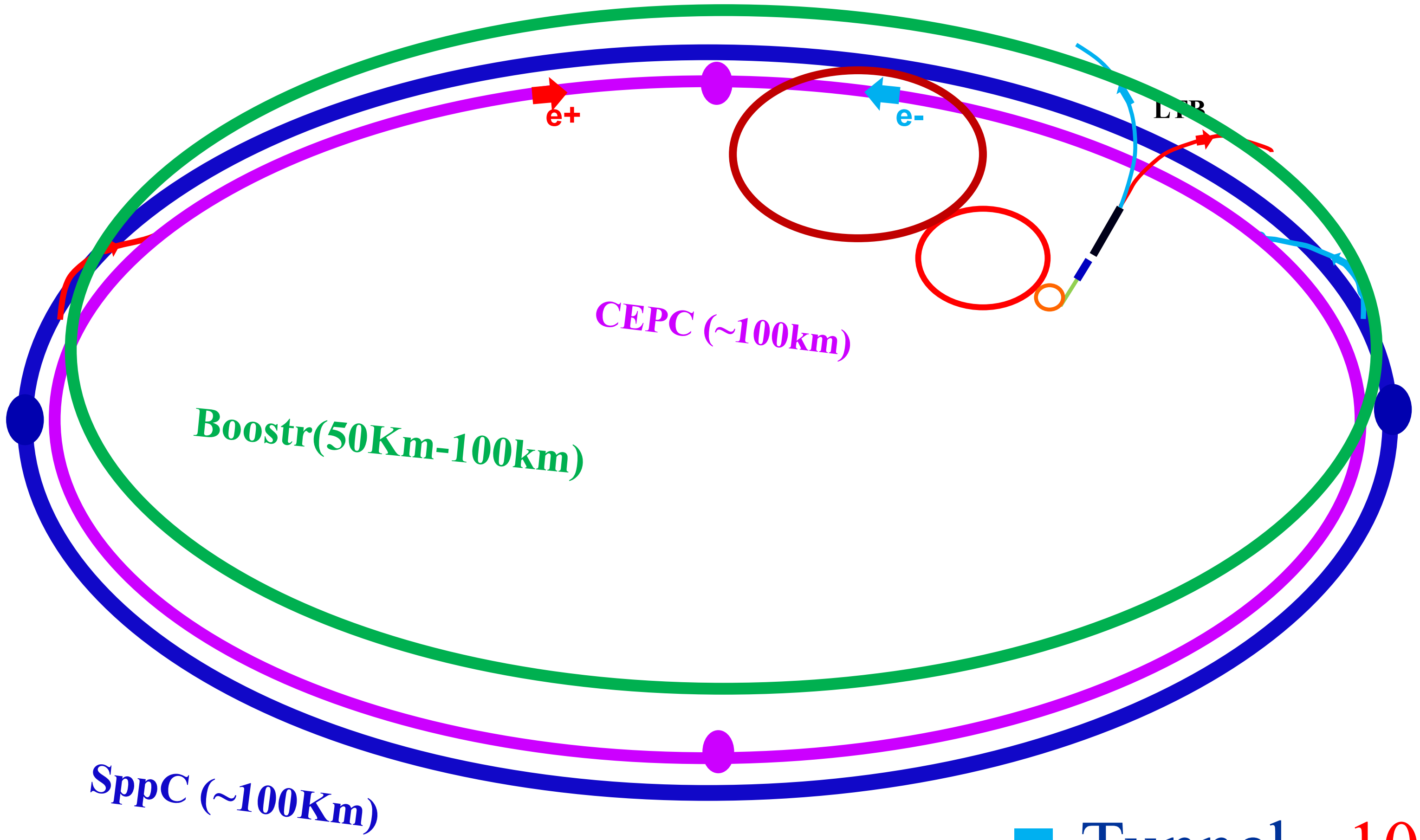
June.12-13, 2019
Future Collider Software Workshop
Bologna， Italy

Outline

- About CEPC
- Current CEPC Software
- Challenges of New Framework
- Design of New Framework
- Investigation and Concerns of Gaudi and Marlin
- New Framework Prototype Plan
- Summary

CEPC and SppC

- Circular Electron-Positron Collider (CEPC)
- Supper Proton-Proton Collider (SppC)



- Tunnel~ 100 km

Science at CEPC and SppC

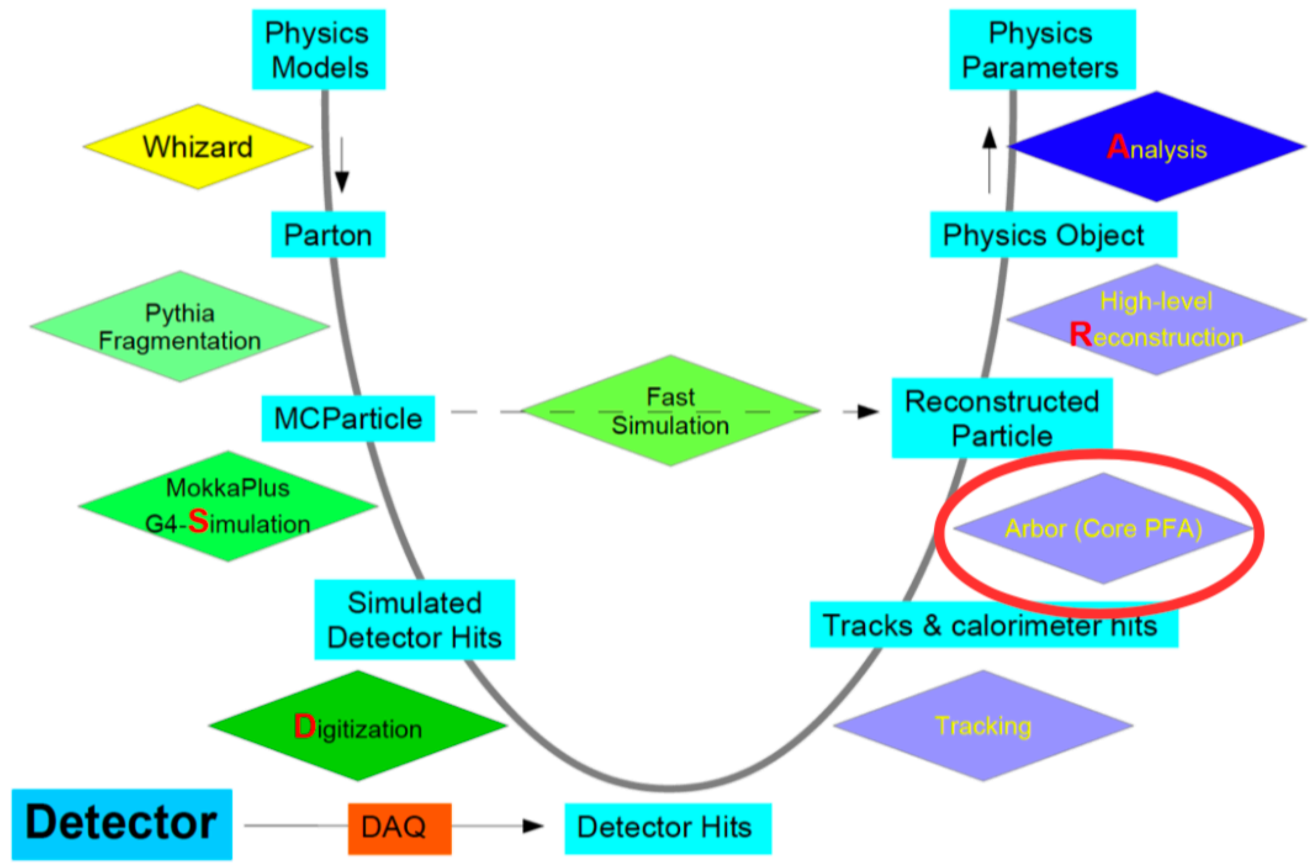
- **CEPC (90-250 GeV)**
  - ⇨ Higgs factory: 1M Higgs boson
    - Absolute measurements of Higgs boson width and couplings
    - Searching for exotic Higgs decay modes (New Physics)
  - ⇨ Z and W factory: 100B-1T Z boson
    - Precision test of the SM
    - Rare decay
  - ⇨ Flavor factory: b, c, tau and QCD studies
- **SppC (~ 100 TeV)**
    - Direct search for new physics
    - Precision test of SM
    - Complementary Higgs measurements to CEPC g(HHH), g(Htt)

Huge Data Volume

- Read out in DAQ from CDR
  - Maximum event rate: ~100 KHz at Z pole
  - Data rate to trigger : ~2 TB/s
- Event Size from simulation
  - Size of signal event:
    - ~500 KB/event for Z and ~ 1 MB/event for Higgs
  - Signal+background
    - 5~10 MB/event for Z and 10~20 MB/event for Higgs
- Data Storage in disk
  - Higgs/W factory
    - ~ 10^8 events for 8 year and 1.5~ 3 PB/year
  - Z factory (2 years)
    - 10^{11} ~10^{12} events for 2 years and 0.5~ 5 EB/year

Overview of Current CEPC Software

From Manqi



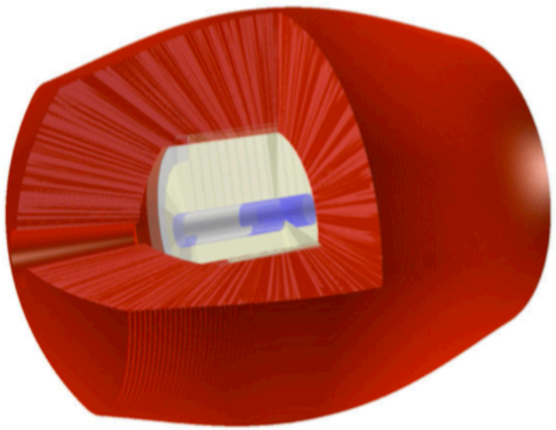
CEPC-SIMU-2017-001,
CEPC-SIMU-2017-002,
(DocDB id-167, 168, 173)

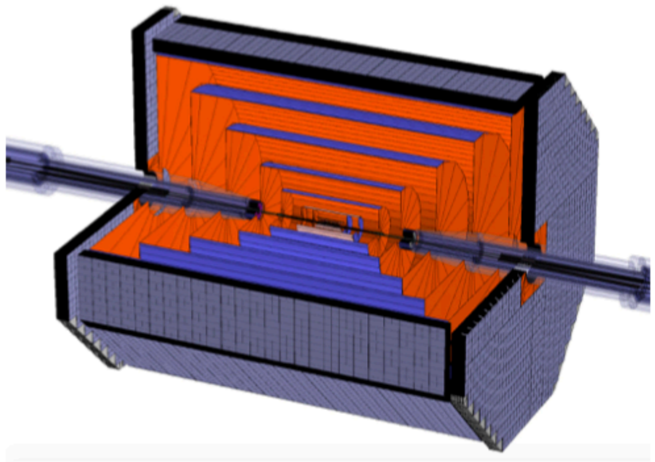
- Generators (Whizard & Pythia)
- Data format & management (LCIO & Marlin)
- Simulation (MokkaC)
- Digitizations
- Tracking
- PFA (Arbor)
- Single Particle Physics Objects Finder (LICH)
- Composed object finder (Coral)
- Tau finder
- Jet Clustering (FastJet)
- Jet Flavor Tagging (LCFIPLus)
- Event Display (Druid)
- General Analysis Framework (FSClasser)
- Fast Simulation (Delphes + FSClasser)

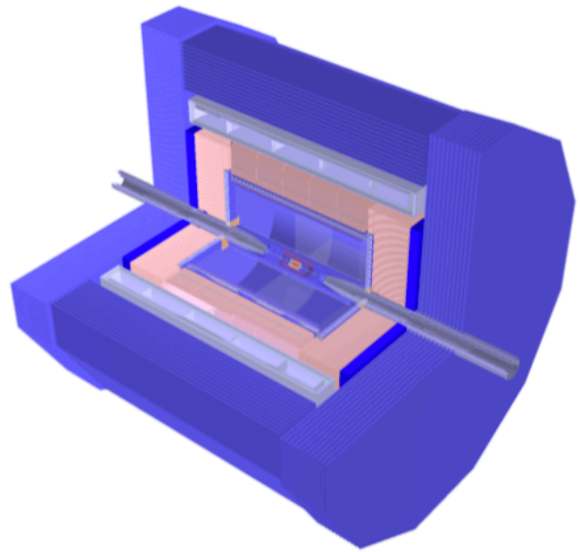
General Software | ILCSoft | ILCSoft + Development | Developments

2

Detector Performance Study

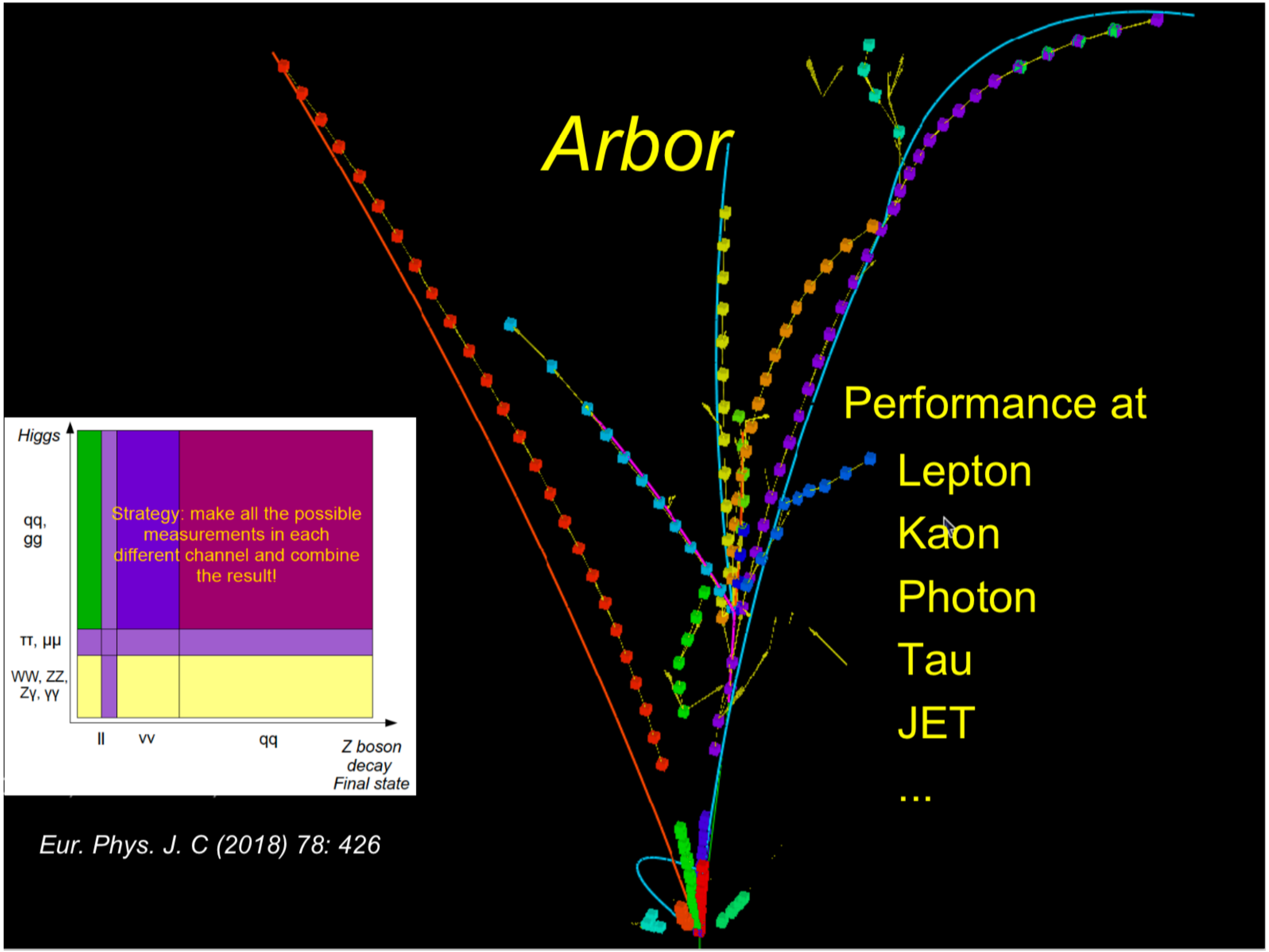






	Geant4-Simulation	Digitization	Reconstruction	Performance-Object	Performance-Benchmark
IDEA					
Full-Silicon					
APODIS					

Reconstruction Algorithm Study



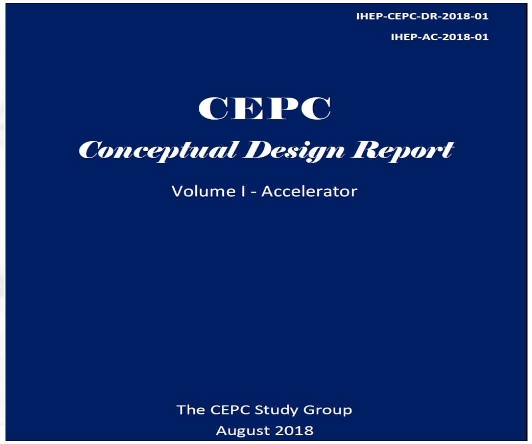
Short Summary of Current CEPC Software

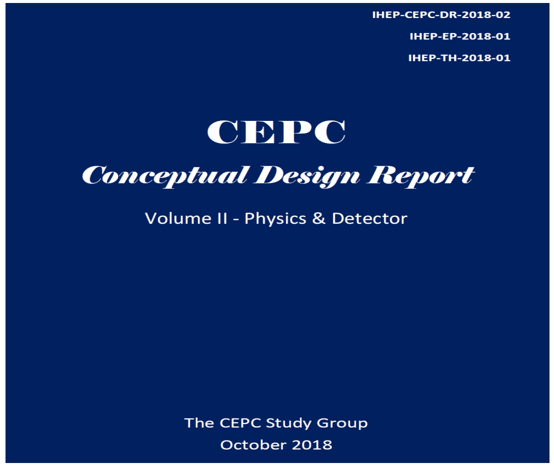
- CEPCSW started from ilcsoft (Many Thanks)
  - LCIO: ILC event data model and format
  - Marlin: data management for reconstruction and analysis
  - Some Tracking and flavor-tagging algorithms

- New developments according to CEPC
  - MokkaC for detector simulation: Extension on Mokka
  - Arbor PFA (Particle Flow Algorithm)
  - Some high-level physics object reconstruction….

- The whole M.C. Chain has been setup and played an important role for the entire CDR Study
  - Optimize Detector geometry and performance
  - Efficiently reconstruct all key physics objects
  - Physics potential study

CEPC Timeline





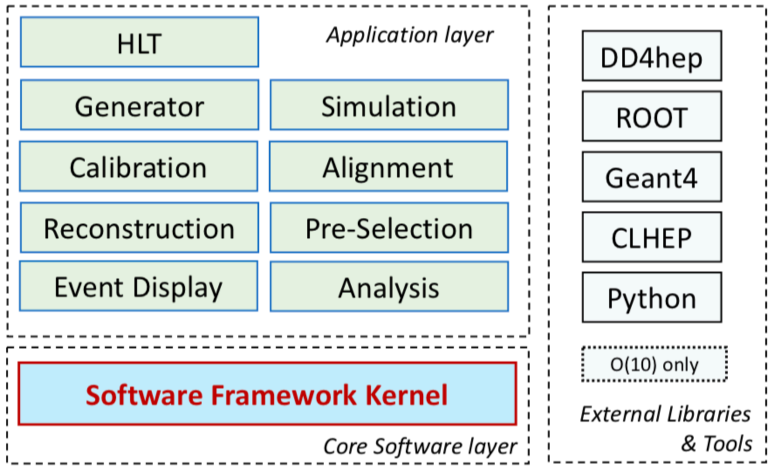
- Finished CDR in Nov, 2018
- Now entering Key Tech. R&D stage
  - Accelerator
  - Detector
  - Software
  - ……
- So a new framework for TDR is under investigation!

Challenges of New Framework

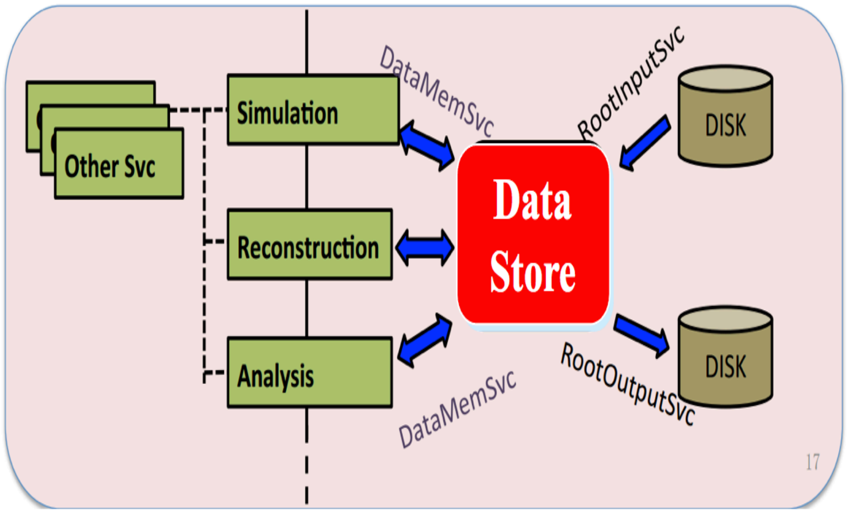
- Huge Data Volume
  - O(~EB) for CEPC running at Z pole
  - Management of Event Data and even non-Event Data
- Heterogeneous Architectures
  - CPU,GPU, FPGA, HPC, Cloud…
  - Portable and flexible
- Parallelization
  - Levels: algorithm, intra-event and inter-event
  - Technologies: OpenCL, CUDA, TBB, MPI…
  - Re-coding or redesign of existing and successful serial algorithms
- Interfaces to novel tools and software
  - Application of Machine learning, Deep Learning and Big Data into HEP
- Friendly user interfaces
  - Hiding new techniques from physicists
  - Support flexible analysis with/without framework , interactive web analysis

Design of New Framework (I)

- **Architecture**
  - C++ and Python
  - Object-oriented, modular, configurable, load dynamically, easy integration with applications
  - Lightweight, less dependencies and portable on different hardware
  - Easily maintained

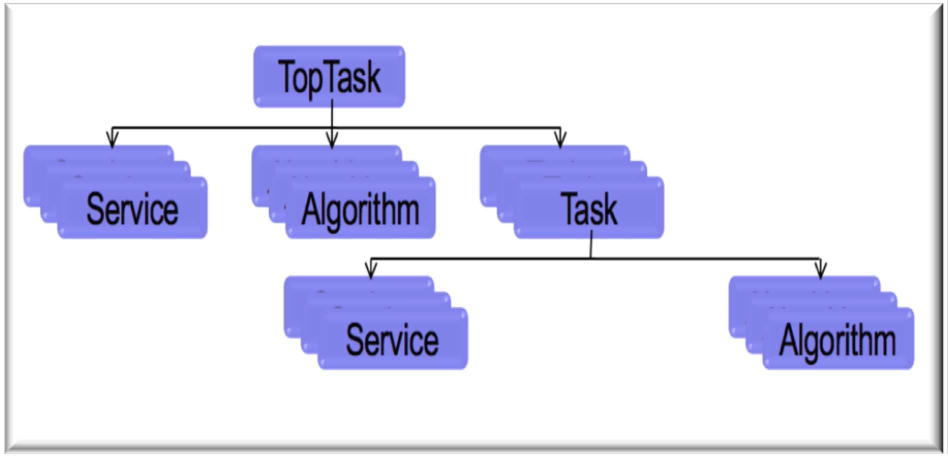


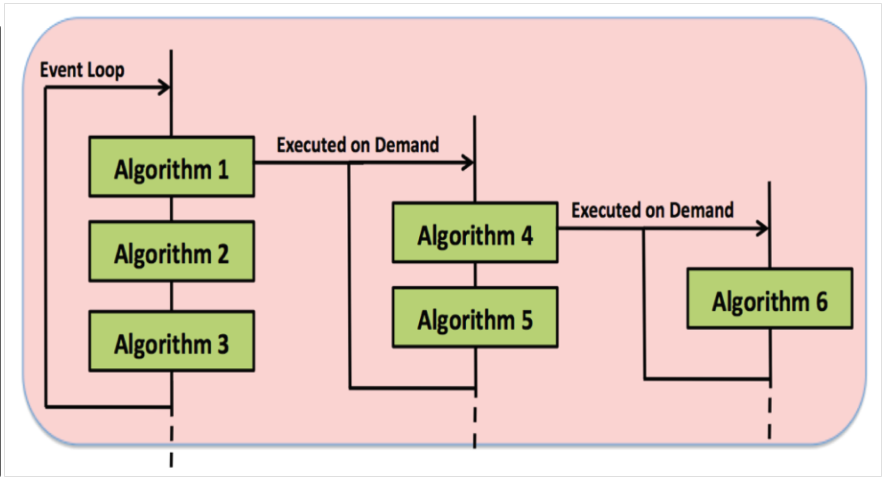
- **Event Data Management**
  - Data Store serves as the central data management place
  - Objects in Data Store should be generic and flexible
  - Objects in Data Store and access to Objects should be thread-safe
  - Support data from disk, tape and network



Design of New Framework (II)

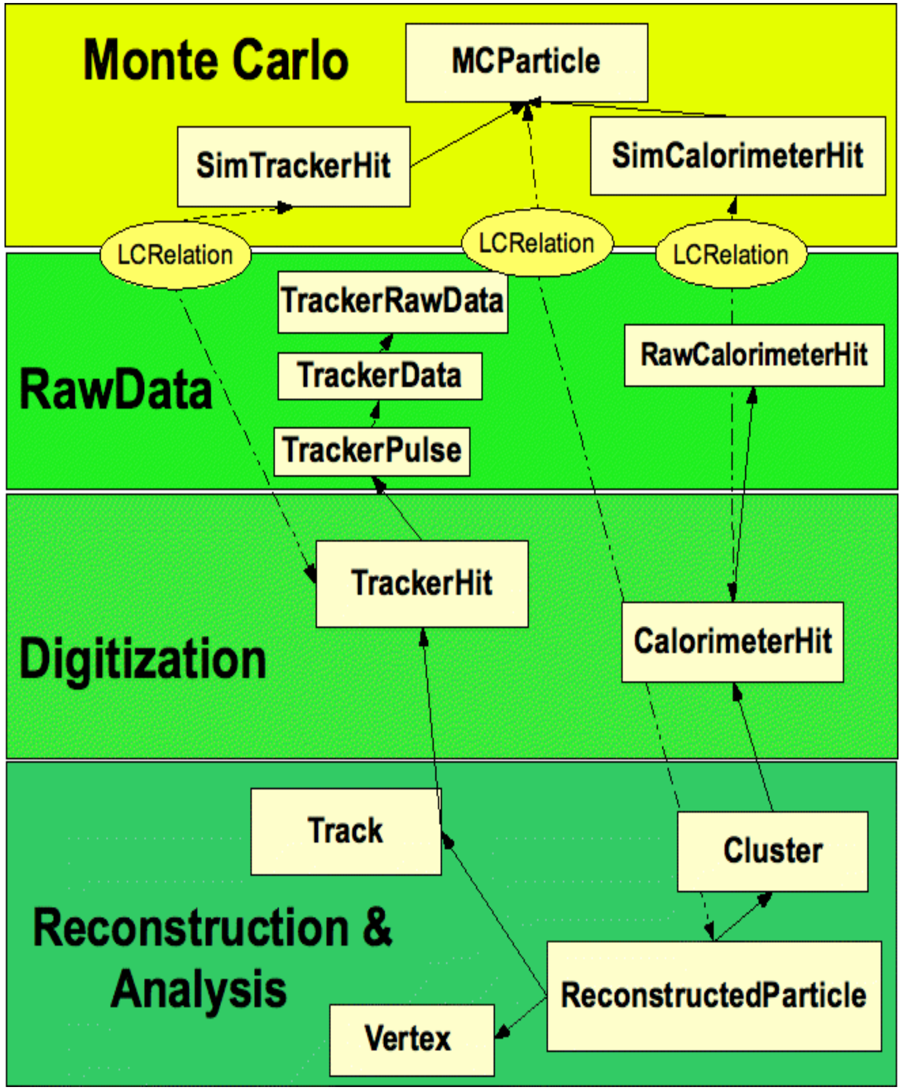
- Data processing
  - Flexible workflow management to support event loop & event filter
  - Three key components: Algorithm, Service and Task
  - Task manages its algorithms, services and subtasks
  - Multi-task provides the intrinsic interface for parallelization





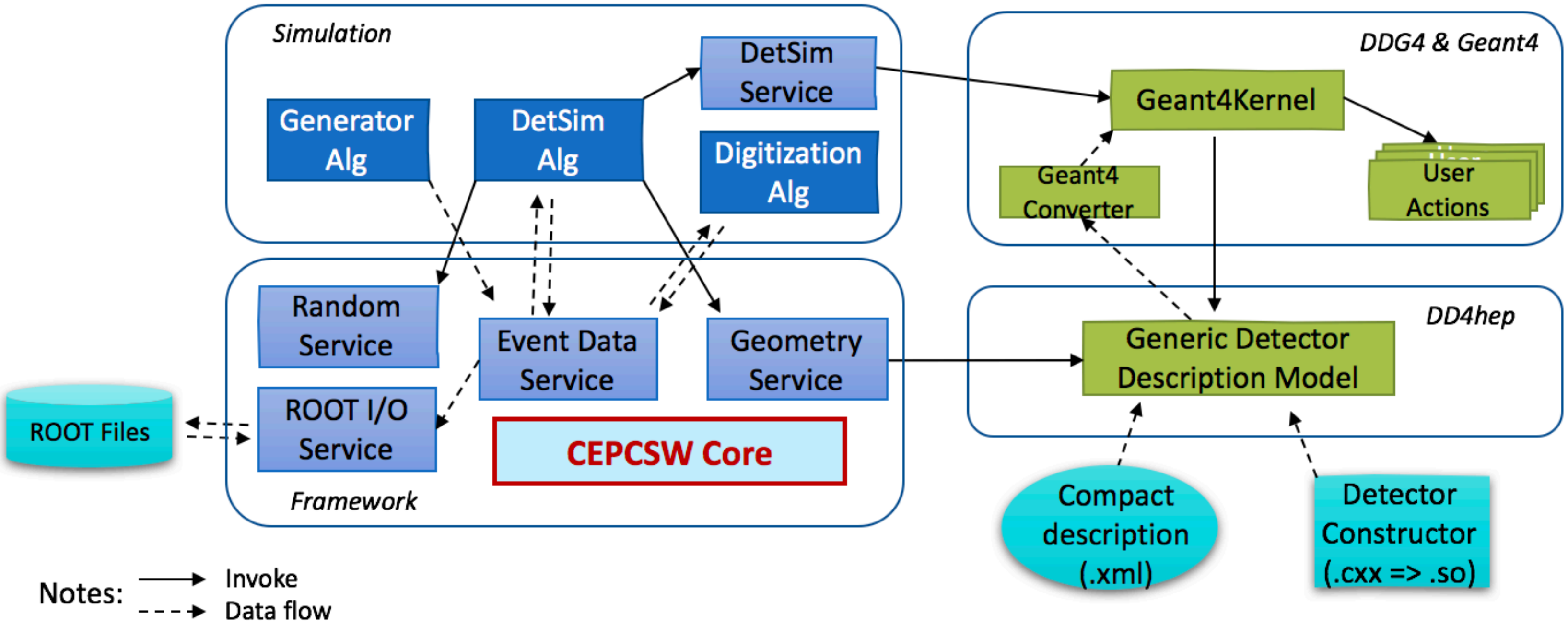
Design of New Framework (III)

- **Event Data Model**
  - LCIO defines a common Event Data Model for Collider Experiments, including
    - Monte Carlo Data, Raw Data
    - Digitization, Reconstruction and Analysis Data
    - Relations between different data objects
  - Will be adopted for CEPC with ROOT extension
    - Combine LCIO with ROOT
    - Put LCIO object inheriting from TObject into Data Store
    - Write Data Store objects into ROOT files
    - Convert Data Store objects into other format (such as NumPy for machine learning)



Design of New Framework (IV)

- Detector Simulation
  - Integration with generators (for example Whizard and Pythia)
  - Adopt DD4hep for detector geometry description
  - Support running different detector designs and comparison between them.
  - Integrate DD4hep with framework
  - Mixing of fast Simulation and full Simulation



Design of New Framework (V)

- Integration of the existing, successful packages and algorithms
  - Reconstruction algorithms
    - Tracking, Vertex-finding，Jet-tagging, for example
      - ACTS (A Common Tracking Software)
      - Arbor particle Flow Algorithm
  - Algorithms developed in offline could be used online.
  - Algorithm development and physics pre-study

- Integration with New technologies
  - Parallel computing: Multithreading, MPI, GPU
  - Machine learning: Tesorflow, Blocks, Keras etc.
  - Big data techniques: spark and dataframe
  - Supercomputers with different computing architectures

CEPC New Framework

- Above are requirements analysis and design consideration for a new Framework

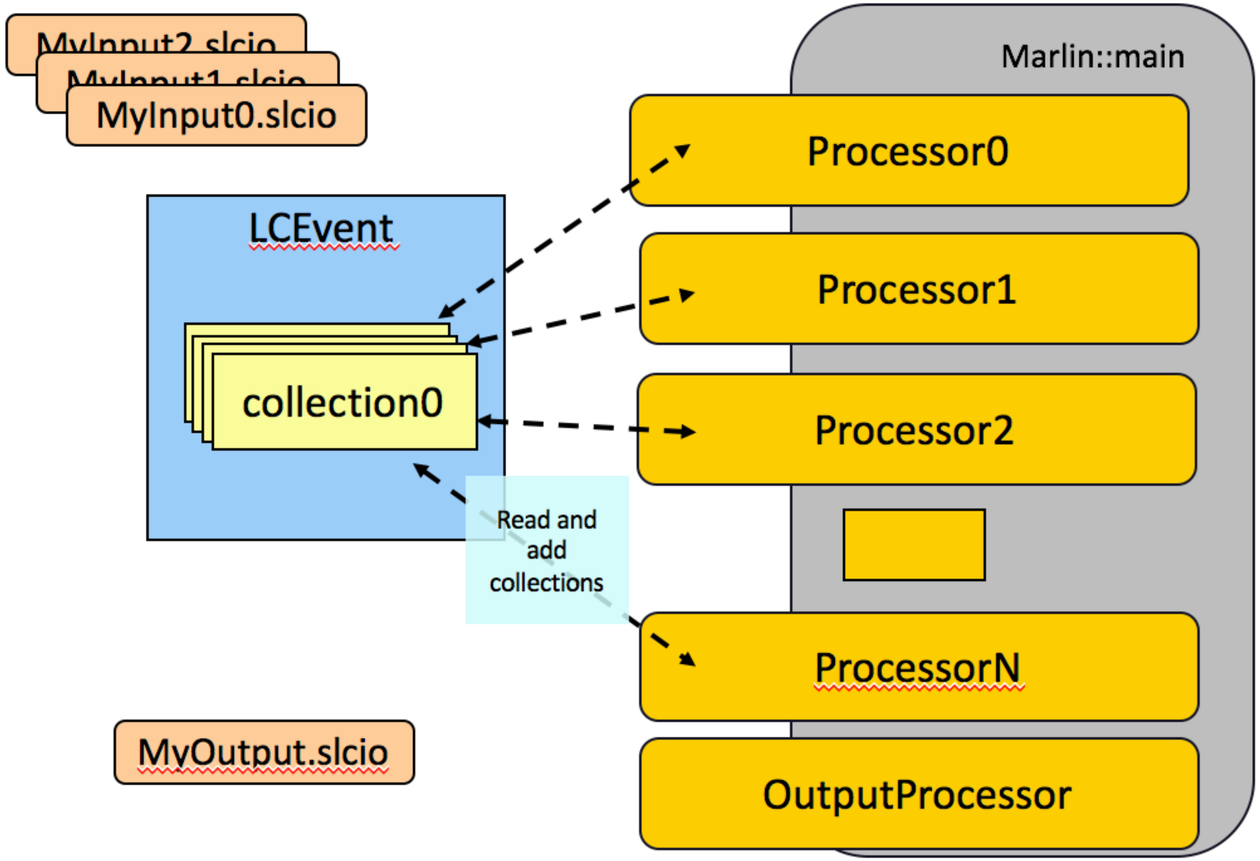
- Honestly speaking, some features and interfaces, especially integration with new technologies, is under study and investigation.

- We urgently hope and need helps/collaboration from other experiments

- But how to do it
  - Start from current popular one?
  - Start basically from scratch?

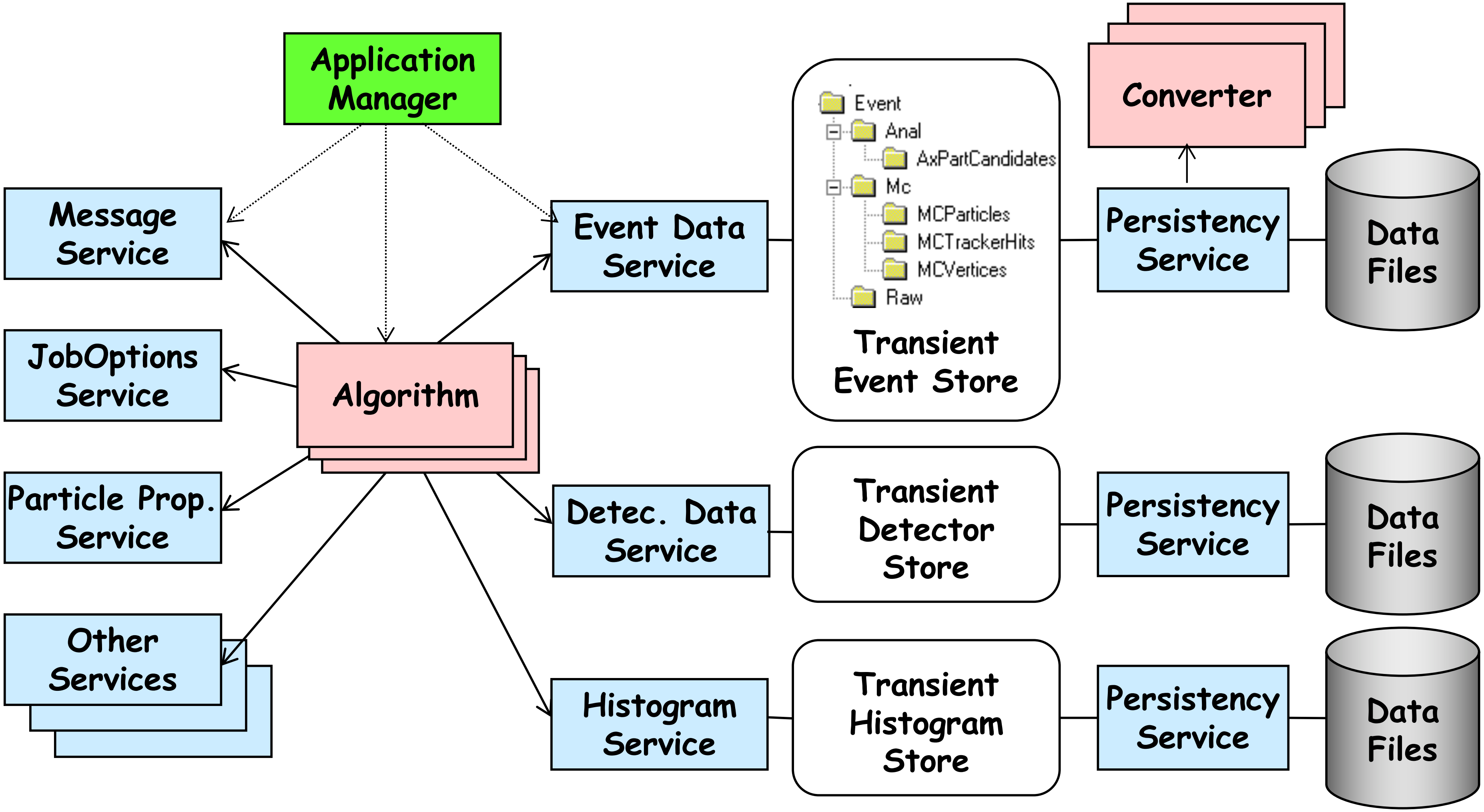
Marlin

- Developed by ILC, used for Reconstruction & Analysis.
- A simple framework based on LCIO
- Used by CEPC-CDR
- Only used in R&D
- Parallelization not yet supported



Gaudi

- Developed by LHCb, became CERN standalone project.
- Already used by BESIII and Daya Bay Experiment
- Good design and very powerful.
- Support Algorithm-level parallelization



Our Concerns with Gaudi

- Less dependencies
  - Decrease the number and the size of external libraries
  - Replace inactive and unpopular libraries
  - Possibility to make it lightweight and convenient to distribute and deploy
- Be portable on various computing resources
  - Cloud, supercomputer, volunteer computing, and etc.
  - Non-x86 hosts, such as ARM
- Integration with other software and tools
  - General purpose software integration, such as DD4HEP
  - Experiment-specific software integration, such as database
    - Expect advices on interfaces, considerations, good and bad experiences from LHC experiments

Our Concerns with Gaudi

- Parallel Computing
  - Complexity: understandable and maintainable
  - How to implement a reentrant algorithm
  - Migration cost of existing codes
  - Smooth switching between parallel and serial mode
- Event Store and Data I/O services
  - Any data types other than DataObject in TES
  - Unified data structure for transient and persistent data
    - Save the costs of type conversions while I/O
  - Management of condition data
  - Thread-safety and execution efficiency
    - Possibility of parallel data structures

New Framework Prototype Plan

- Investigate and design Kernel part of CEPC new Framework
  - Based one of current one or not?
- Develop ROOT-based Event Data Model & I/O
  - Similar interface as LCIO EDM.
- Develop Backward Compatible Input System
  - Read existing simulated samples during the migration.
- Develop Unified Geometry System for Sim/Rec/Ana
  - DD4hep
- Integrate and migrate the existing codes
  - Simulation/Reconstruction/Analysis algorithms
- Design integration interfaces with new technologies
- Conduct performance testing and give advices for the final CEPC framework

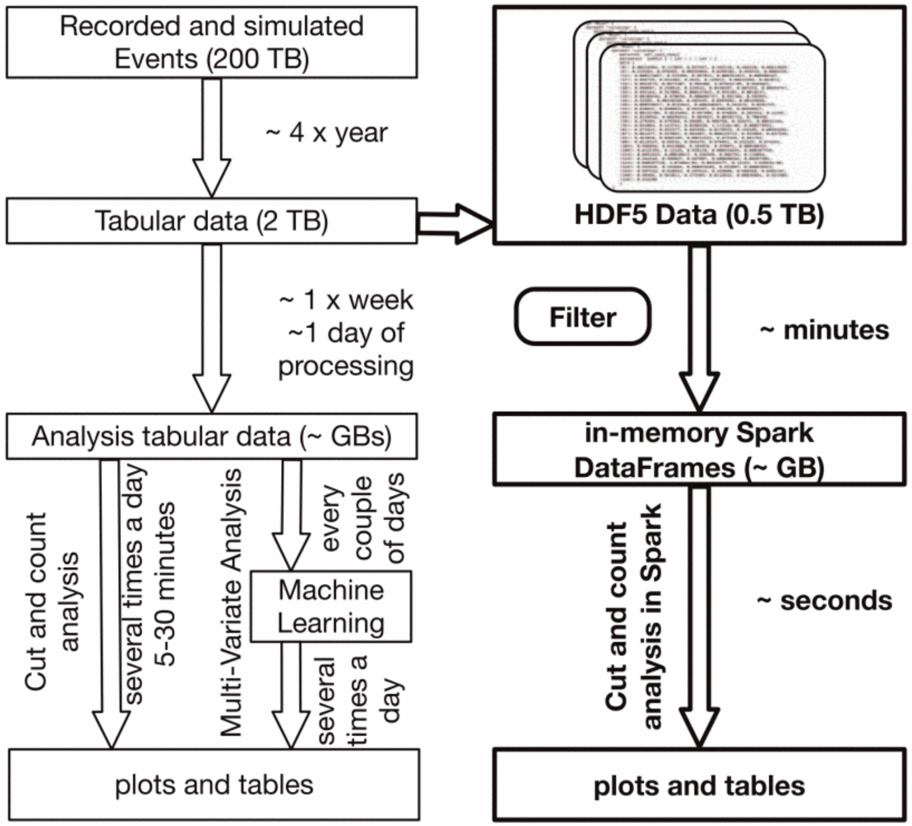
Summary

- Current Software system played an important role for Detector design, physics benchmarks and CDR.
- Based on the new challenges, design consideration of new framework, and our concerns on current popular frameworks are expressed
- We hope to get advices/suggestions/comments from the experts during this meeting
- International collaboration on CEPC new framework is mostly welcome
- HSF is the right platform for this kind of collaboration

Thanks a lot!

Interactive Analysis with "Big Data"

- Motivation: physics analysis via a web interface with dynamically allocated computing resource.
  - ⇨ Jupyter based web interface.
- "Big data" technology: in-memory analysis
  - ⇨ Read once; Analysis multiple times
  - ⇨ Example: in-memory Spark, DataFrames.
- Challenges: integration with new technologies, such as Spark
  - ⇨ Access event data in different formats.
  - ⇨ Distributed computing with Spark.



Saba Sehrish *et al.*, Spark and HPC for High Energy Physics Data

CEPC Distributed Computing (DC)

- DC will be the main way to organize resources for CEPC
- DIRAC-based DC system has been built up to support CEPC R&D in 2015
  - ⇨ Already integrate resource: Cloud, Cluster, Grid
  - ⇨ Extensible to use more heterogeneous resources
- Several supports for software parallelism in DC are ready
  - ⇨ Multi-core workload scheduling
  - ⇨ Seamless integration of HPC resources
  - ⇨ Singularity supports in pilots