Computing & Software
for the Super-Charm-Tau factory detector project

Andrey Sukharev

Budker Institute of Nuclear Physics, Novosibirsk, Russia

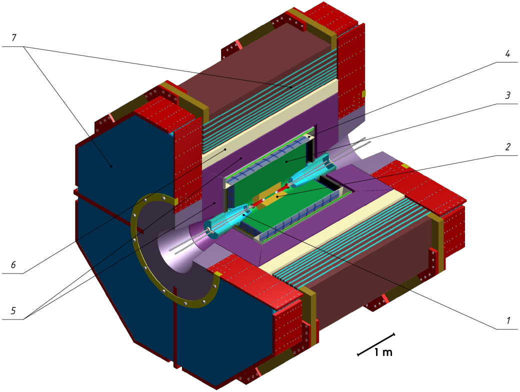
13 December 2021

BINP

Detector overview

Requirements:

- Occupancy 300 kHz
- Good energy and momentum resolution
- High detection efficiency of soft tracks
- Best possible π/K and π/μ separations
- Minimal CP detection asymmetry



	subsystem	options		subsystem	options
1	Beam pipe	beryllium	**2**	Inner tracker	TPC, cGEM, Si-strip
3	Main tracker	drift chamber	**4**	PID system	FARICH, ASHIPH
5	Calorimeter	CsI, LYSO, LXe	**6**	Magnet	thin coil?
7	Muon system	Scintillators, RPC, . . .			

Computing infrastructure for the project

- The immediate goal is to design the detector
  - need the simulation
  - ...and where to run it
- Actual computing infrastructure is to be finalized as late as possible
  - outlined in the CDR (to be updated soon)
- The existing Budker INP/General Computing Facility (BINP/GCF) is available
  - local computing farm of about 2k CPU cores
  - various storage subsystems
  - VM servers (about 100 CPU cores)
  - IB/10GbE/40GbE local interconnects
  - access to remote resources

...shared with other groups

Local Resources at BINP/GCF

- Local computing farm

- BINP/GCF storage

- VM servers

- Centralized UPS solution has been installed recently
- The computing environment based on Scientific Linux 7

Local Resources at BINP/GCF

- Local computing farm → **improving**
  two EPYC 7702P installed recently, more to come soon,
  phasing out older machines
- BINP/GCF storage
- VM servers
- Centralized UPS solution has been installed recently
- The computing environment based on Scientific Linux 7

Local Resources at BINP/GCF

- Local computing farm → **improving**
  two EPYC 7702P installed recently, more to come soon,
  phasing out older machines
- BINP/GCF storage
  → **upgrading, moving from disk arrays to CEPH**
  $\sim$ 500 TB @ HDD + $\sim$ 50 TB @ SSD (raw)
- VM servers

- Centralized UPS solution has been installed recently
- The computing environment based on Scientific Linux 7

Local Resources at BINP/GCF

- Local computing farm $\rightarrow$ **improving**
  two EPYC 7702P installed recently, more to come soon, phasing out older machines
- BINP/GCF storage
  $\rightarrow$ **upgrading, moving from disk arrays to CEPH**
  $\sim$ 500 TB @ HDD + $\sim$ 50 TB @ SSD (raw)
- VM servers
  partially **upgraded** recently, almost doubling CPU power
- Centralized UPS solution has been installed recently
- The computing environment based on Scientific Linux 7

Local Resources at BINP/GCF

- Local computing farm → **improving**
  two EPYC 7702P installed recently, more to come soon, phasing out older machines
- BINP/GCF storage
  → **upgrading, moving from disk arrays to CEPH**
  ∼ 500 TB @ HDD + ∼ 50 TB @ SSD (raw)
- VM servers
  partially **upgraded** recently, almost doubling CPU power
- Centralized UPS solution has been installed recently
- The computing environment based on Scientific Linux 7
  → **migration to CentOS 8 cancelled**

Resources available via BINP/GCF

- Computing resources of Novosibirsk Scientific Center
  - NUSC & SSCC supercomputers
    - mostly GPU...
    - ...but still several thousands CPU
    - good experience of collaboration
  - ICT SB RAS storage
    - > 500 TB
  - ...connected with isolated 10GbE network (NSC/SCN)
- Dedicated network link to Moscow (KIAE)
  - 2 Gbps presently
  - direct access to LHCone network

SCT Project at BINP/GCF

BINP/GCF provides the SCT detector project with

- general login servers,
- computing nodes & batch system,
- storage area,
- gitlab server,
- wiki & web servers,
- mail list,
- ...

SCT Project at BINP/GCF

BINP/GCF provides the SCT detector project with

- general login servers,
- computing nodes & batch system,
- storage area,
- gitlab server,
- wiki & web servers,
- mail list,
- ...

No lack of computing resources for the present stage of the detector project

Human resources

- BINP base software development team
  - design & develop general framework
  - provide users with examples and basic package version
  - implement modules when subsystem experts are unavailable
  - prepare build, test, visualization etc tools
  - BINP/GCF administration
- Parameterized simulation team
- Subsystem experts
  - expected to prepare comprehensive parts of the code concerned with corresponding subsystem(s)
  - framework usage and feedback
- Physics & Analysis experts
  - event generation and analysis tools implementation
  - framework usage and feedback

Software for the project

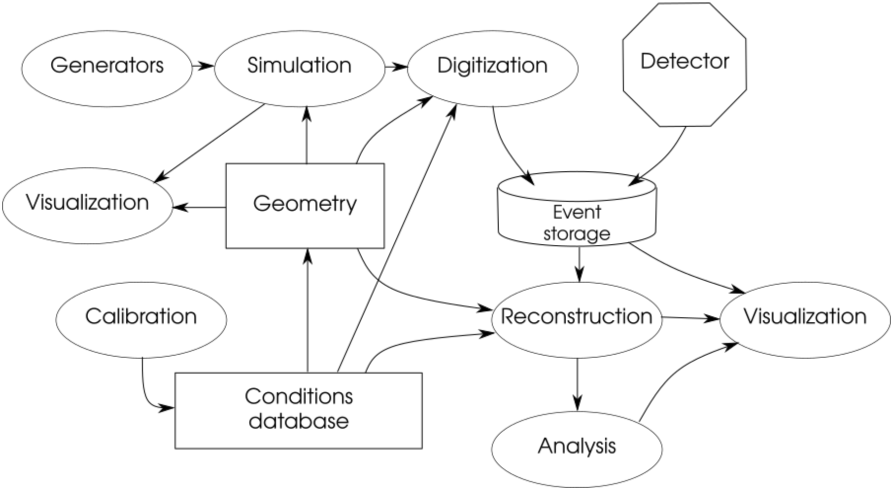
A HEP software framework

A typical HEP experiment requires a complete stack of relevant software:

- event generators,
- parametric and full detector simulation,
- event reconstruction algorithms,
- online event interpretation for trigger decisions,
- event data model (EDM),
- I/O interface to conditions data base,
- I/O interface to data storage,
- offline data analysis algorithms,
- build system and release management software.

Software for the project

Framework elements and data flows



The Aurora framework

- Based on Gaudi (de-facto HEP standard)
  - allows to develop software components in a convinient way (Algorithms, Tools, Services)
  - mixing C++ and Python code
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4...
  - Key4HEP (DD4hep, PODIO...)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages
- Nightly builds
- Standard computing environment is Scientific Linux 7 x86_64, GCC8 + Python2&3

The Aurora framework

- Based on Gaudi (de-facto HEP standard)
  - allows to develop software components in a convinient way (Algorithms, Tools, Services)
  - mixing C++ and Python code
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4...
  - Key4HEP (DD4hep, PODIO...)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages $\rightarrow$ **spack?**
- Nightly builds
- Standard computing environment is Scientific Linux 7 x86_64, GCC8 + Python2&3

The Aurora framework

- Based on Gaudi (de-facto HEP standard)
  - allows to develop software components in a convinient way (Algorithms, Tools, Services)
  - mixing C++ and Python code
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4...
  - Key4HEP (DD4hep, PODIO...)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages → **spack?**
- Nightly builds → **CI**
- Standard computing environment is Scientific Linux 7 x86_64, GCC8 + Python2&3

The Aurora framework

- Based on Gaudi (de-facto HEP standard)
  - allows to develop software components in a convinient way (Algorithms, Tools, Services)
  - mixing C++ and Python code
- Conventional and recently emerged HEP software tools:
  - ROOT, Geant4...
  - Key4HEP (DD4hep, PODIO...)
- Other experiments software
  - Belle II, ILC, FCCSW...
- Build & configuration system inspired by ATLAS Athena
- `lcgcmake` system to build external packages → **spack?**
- Nightly builds → **CI**
- Standard computing environment is Scientific Linux 7 x86_64, GCC8 + Python2&3 → **GCC9, pure Python3**

Standalone studies

- Parametric simulation tool for quick estimations of the detector response
- Background simulations with Fluka
- Gas mixture studies and electric field simulations with Garfield for TPC and DC
- CERN team develops TCP variant & adopts track finding algorithms from iLCSoft

Standalone studies

- Parametric simulation tool for quick estimations of the detector response
  $\rightarrow$ **recently incorporated into the framework**
- Background simulations with Fluka
- Gas mixture studies and electric field simulations with Garfield for TPC and DC
- CERN team develops TCP variant
  & adopts track finding algorithms from iLCSoft
  $\rightarrow$ **have a prototype, now incorporating into the framework**

Event Generators

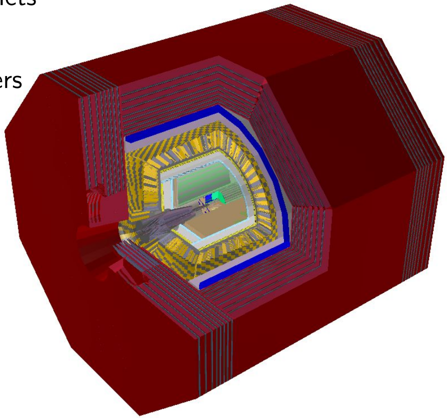
The conventional set of event generators available

- Exclusive decays of hadrons and tau lepton
  - EvtGen, Tauola, PHOTOS, Pythia
- Inclusive generators for $e^+e^- \to$ hadrons
  - preliminary solution based on Pythia
- Generators for luminosity measurements and calibrations
  - MCGPJ, BabaYaga, BBBREM, KKMC...

Status of the software

Geometry in Aurora

- Subsystems implemented to the moment:
  - Beam pipe & final focus magnets
  - Inner tracker (three options)
  - Advanced DC with StereoLayers
  - Particle ID
  - Crystal calorimeter
  - Simplified s/c coil
  - Muon system & yoke
- Geometry testing tools (overlaps, material scans. . .)
- Simplified magnetic field



We have geometry for at least one option for each subsystem

Status of the software

Digitization & Reconstruction

- refactoring the code base to fit Digi/Reco model
- some subsystems miss separate Digitization stage yet (“integrated” into reconstruction)
    - based on standalone studies
    - in preparation by subsystem groups
- 1st stage Reconstruction: individual subsystem level
    - in preparation by subsystem groups
    - Calorimeter and DC most advanced at the moment
- 2nd stage Reconstruction: combining subsystems, PID...
- the Event Data Model is to be finalized

Status of the software

Digitization & Reconstruction

- refactoring the code base to fit Digi/Reco model
- some subsystems miss separate Digitization stage yet ("integrated" into reconstruction)
  - based on standalone studies
  - in preparation by subsystem groups
- 1st stage Reconstruction: individual subsystem level
  - in preparation by subsystem groups
  - Calorimeter and DC most advanced at the moment
- 2nd stage Reconstruction: combining subsystems, PID...

  $\rightarrow$ scheduled for next year
- the Event Data Model is to be finalized

Status of the software

Digitization & Reconstruction

- refactoring the code base to fit Digi/Reco model
- some subsystems miss separate Digitization stage yet (“integrated” into reconstruction)
  - based on standalone studies
  - in preparation by subsystem groups
- 1st stage Reconstruction: individual subsystem level
  - in preparation by subsystem groups
  - Calorimeter and DC most advanced at the moment
- 2nd stage Reconstruction: combining subsystems, PID...

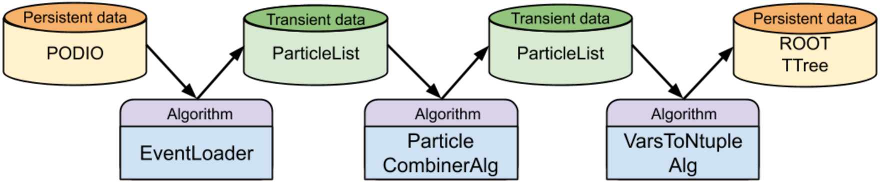
  $\rightarrow$ scheduled for next year
- the Event Data Model is to be finalized

  $\rightarrow$ eventually re-basing to EDM4hep

Status of the software

Data Analysis

- Adopting Belle II recipes and solutions for analysis
- Base set of analysis algorithms ready:



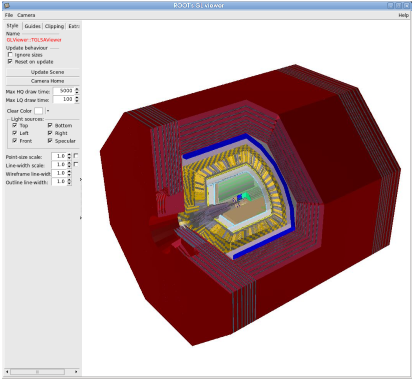
- final particles processing
- kinematic cuts

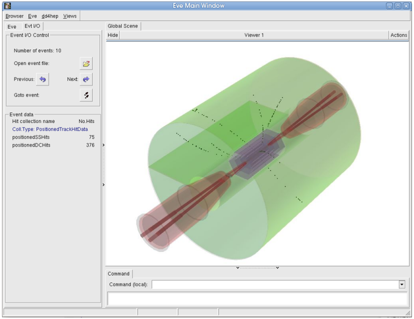
- reconstructuion of decayed particle
- kinematic cuts

- kinematic parameters calculation
- particle selection

Status of the software

Detector/Event Display

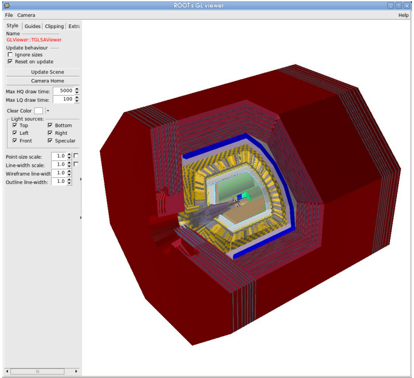


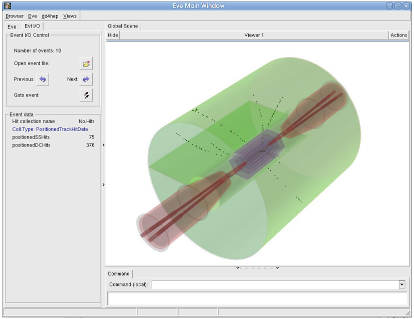


- Geometry Display tool is ready
- Event Display (DDEve-based) available, lots of things to improve

Status of the software

Detector/Event Display

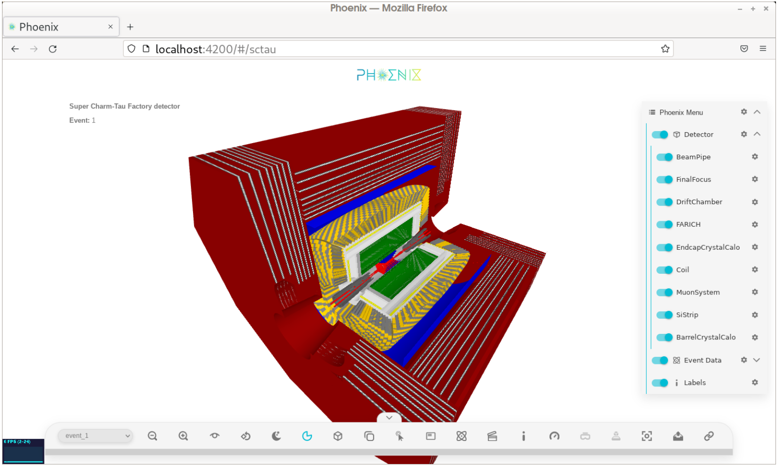




- Geometry Display tool is ready
- Event Display (DDEve-based) available, lots of things to improve ? to be replaced with →

Status of the software

Web-based Detector/Event Display



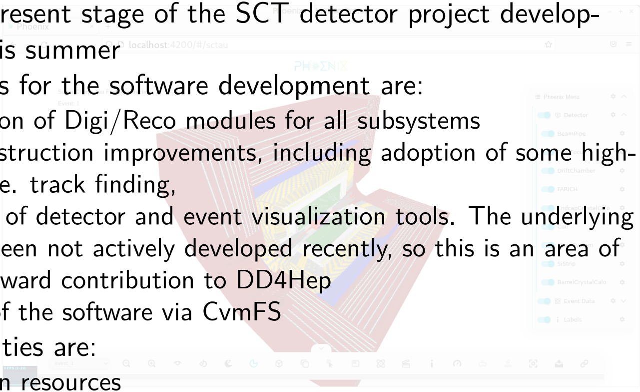
- based on Phoenix project
- development started recently

Conclusions

- There are sufficient computing resources for the SCT detector project at Budker INP
- The Aurora framework v 1.0.1 containing all components minimally required at the present stage of the SCT detector project development released this summer
- The nearest goals for the software development are:
  - implementation of Digi/Reco modules for all subsystems
  - further reconstruction improvements, including adoption of some high-level tools, i. e. track finding,
  - improvement of detector and event visualization tools. The underlying DDEve has been not actively developed recently, so this is an area of possible backward contribution to DD4Hep
  - distribution of the software via CvmFS
- The main difficulties are:
  - lack of human resources
  - Key4HEP tools are not finalized yet

Conclusions

- There are sufficient computing resources for the SCT detector project at Budker INP
- The Aurora framework v 1.0.1 containing all components minimally required at the present stage of the SCT detector project development released this summer
- The nearest goals for the software development are:
  - implementation of Digi/Reco modules for all subsystems
  - further reconstruction improvements, including adoption of some high-level tools, i. e. track finding,
  - improvement of detector and event visualization tools. The underlying DDEve has been not actively developed recently, so this is an area of possible backward contribution to DD4Hep
  - distribution of the software via CvmFS
- The main difficulties are:
  - lack of human resources
  - Key4HEP tools are not finalized yet



Thank you for attention