



ПРОГРАММНАЯ ПЛАТФОРМА ДЛЯ
МНОГОЭТАПНОЙ ОБРАБОТКИ ДАННЫХ
ЭКСПЕРИМЕНТА SPD НА ОСНОВЕ
ФРЕЙМВОРКА GAUDI

Студент: Симбирятин Л. Л.

Научный руководитель: Жемчугов А. С.

27.01.2025
Москва

Цели работы

Текущее офлайн ПО SPDRoot:

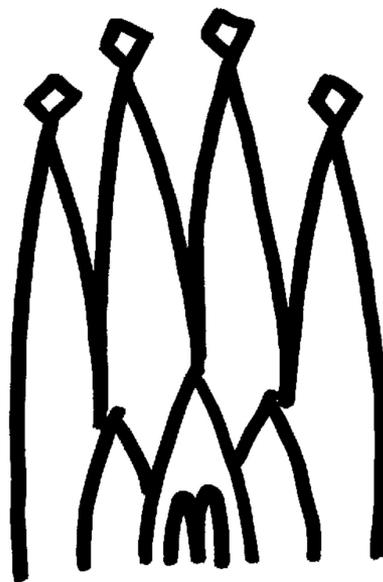
- недостаточно надежное, много рудиментарного код
- наследует проблемы пакета ROOT
- однопоточное

Таким образом, его необходимо заменить.

Gaudi фреймворк в качестве альтернативы:

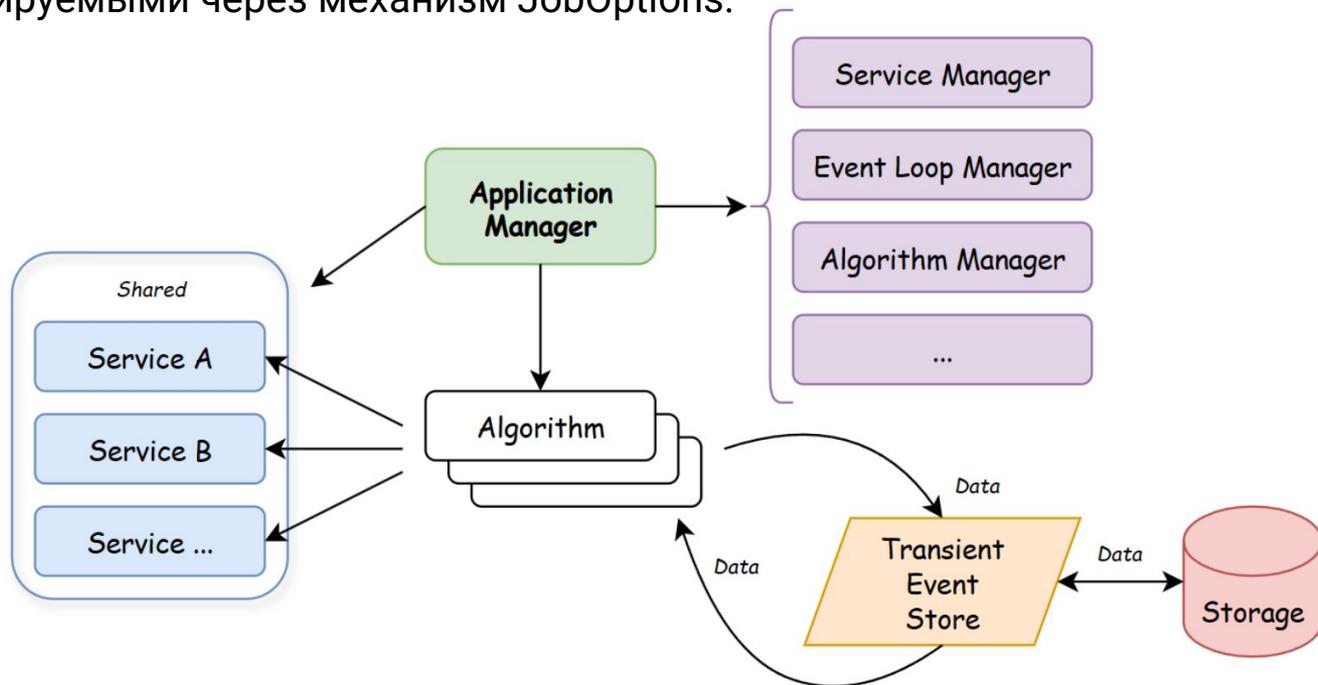
- надежный
- динамически конфигурируемые задачи
- поддержка многопоточности

Gaudi Tutorial: <https://git.jinr.ru/lsimbir/gauditutorial/-/tree/main>



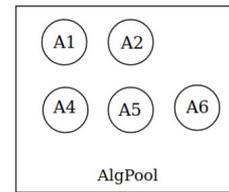
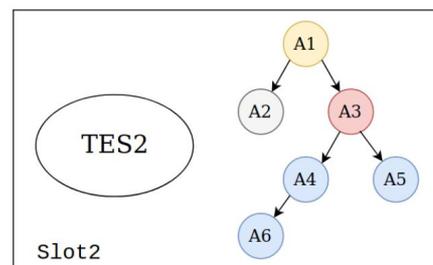
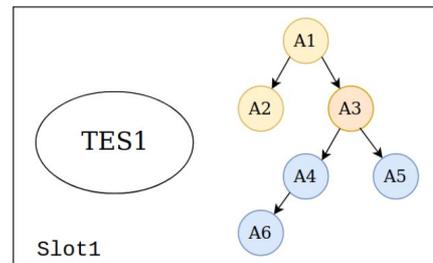
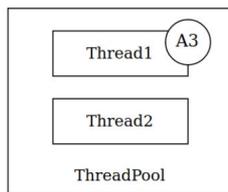
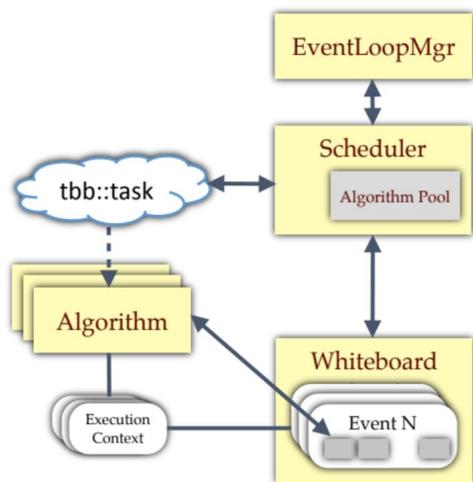
Архитектура Gaudi (однопоточная)

Концепция: Объекты данных обрабатываются алгоритмами, запускаемыми в цикле обработки событий. Алгоритмы, сервисы и инструменты являются динамически конфигурируемыми через механизм JobOptions.

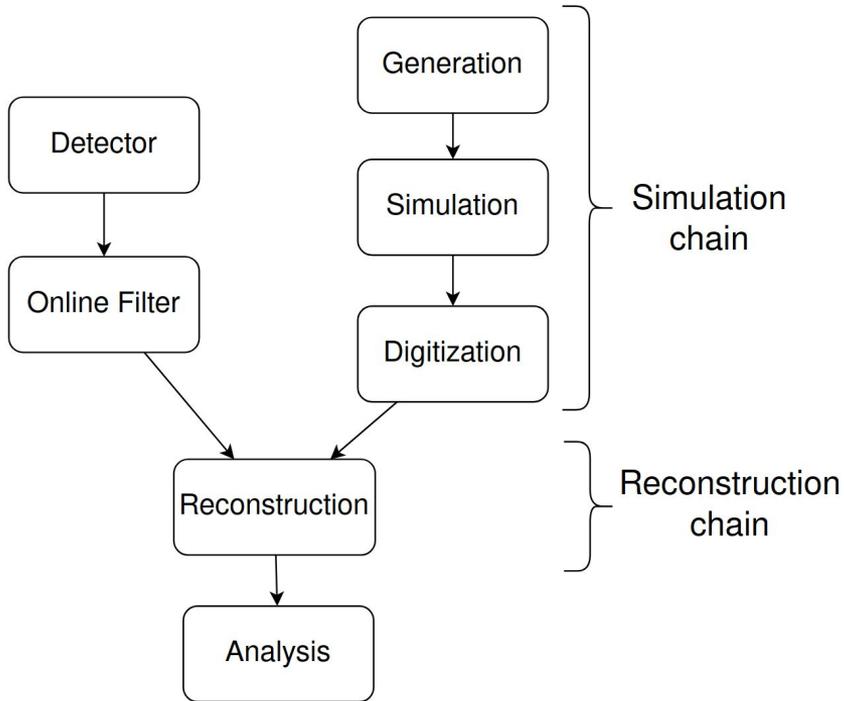


Архитектура GaudiHive (многопоточная)

Концепция: Можно обрабатывать одновременно несколько событий. Task-based параллельность. Планировщик анализирует граф обработки события, находит готовые к запуску алгоритмы, достает их из пула и упаковывает в задачи. Задачи попадают в очередь, отслеживаемую пулом потоков.



Этапы обработки данных



Какие библиотеки используются:

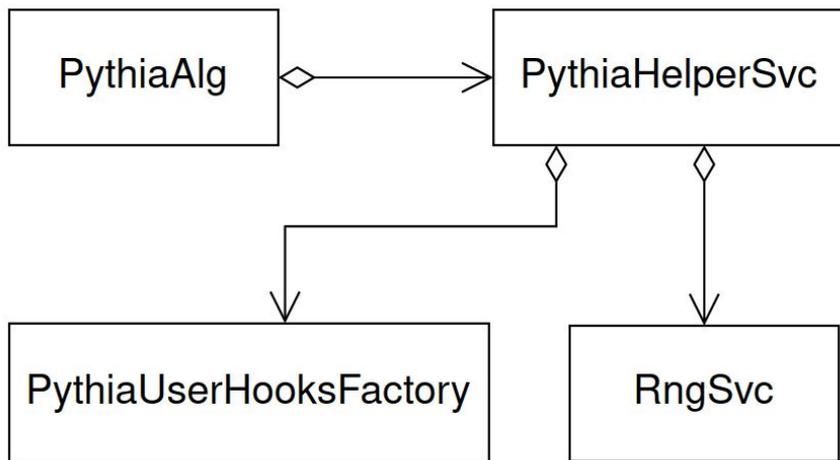
Generation -> Pythia8

Simulation-> Geant4 + GeoModel + HepMC3

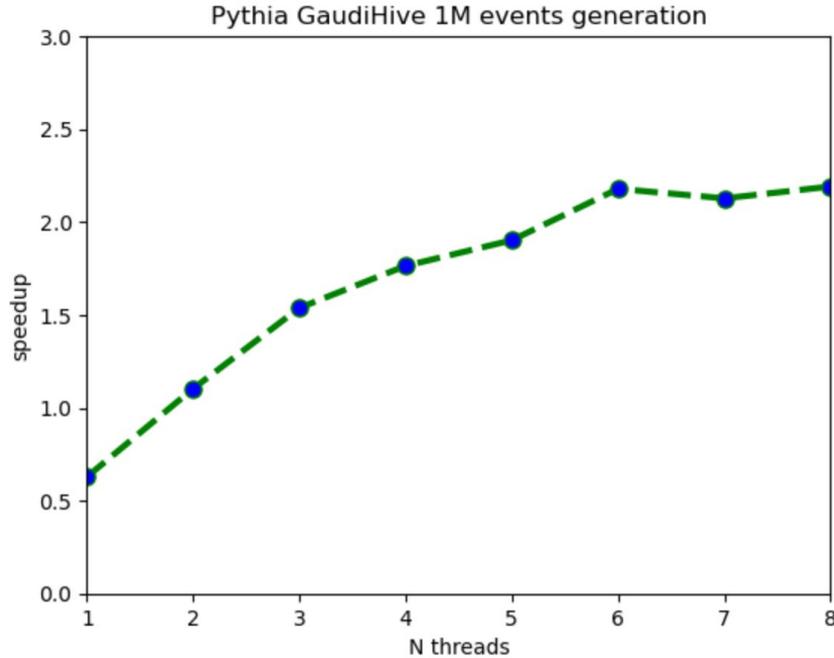
Reconstruction-> ?Acts?

Интеграция Pythia8

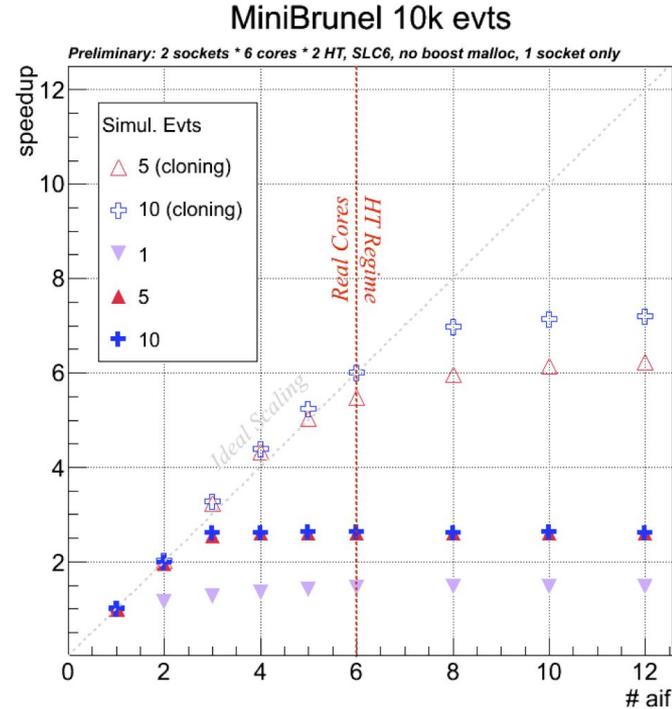
Концепция: PythiaHelperSvc создает объекты класса Pythia8, снабжает их генераторами случайных чисел, юзер-хуками. Pythia8::next() вызывается в алгоритме PythiaAlg.



Pythia в GaudiHive



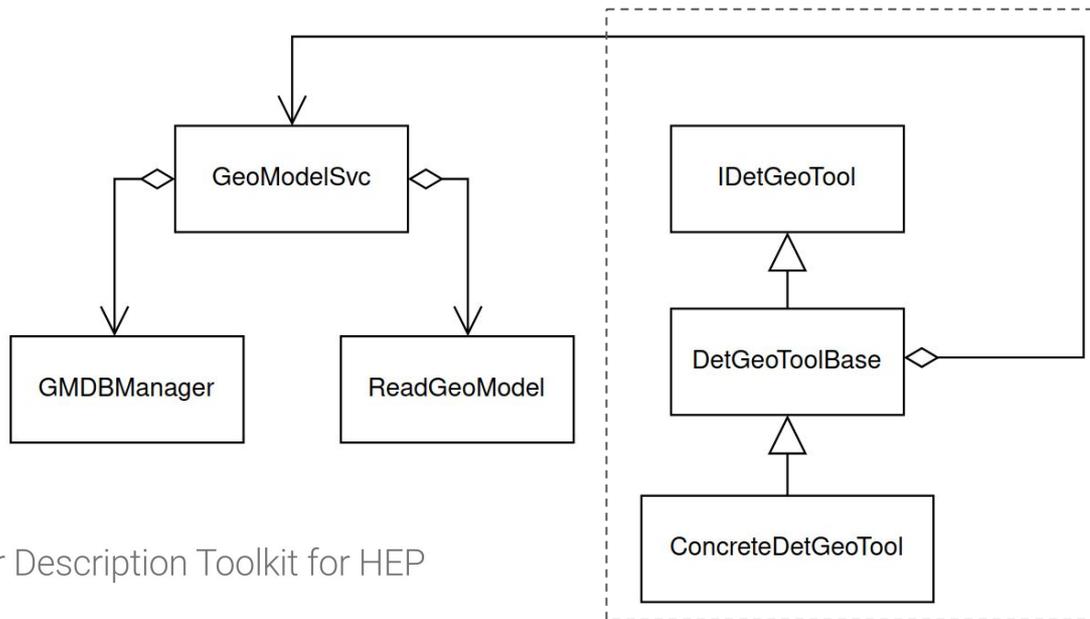
Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, 8 CPUs



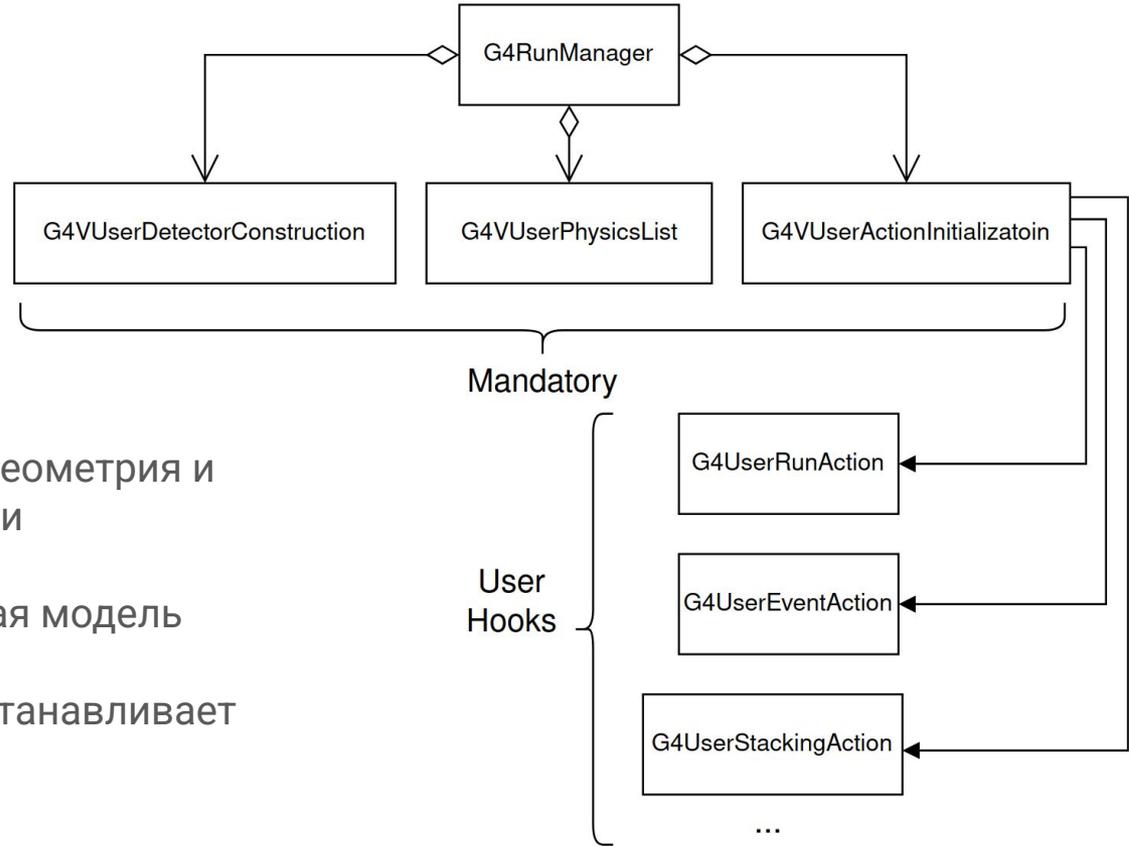
Introducing concurrency in the Gaudi data processing framework / M. Clemencic

Интеграция GeoModel

Концепция: GeoModelSvc имеет доступ к БД с геометрией через GMDBManager. ReadGeoModel используется для построения геометрии. IDetGeoTools являются клиентами сервиса, инициализируются и содержат информацию о подсистемах детектора.



Geant4: Пользовательские классы



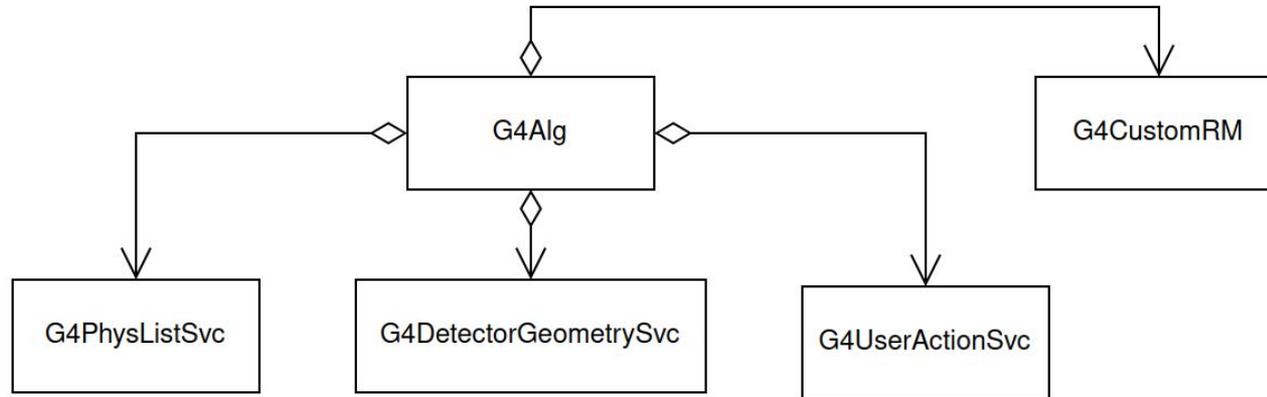
G4VUserDetectorConstruction -> геометрия и чувствительные части установки

G4VUserPhysicsList -> физическая модель

G4VUserActionInitialization -> устанавливает пользовательские хуки

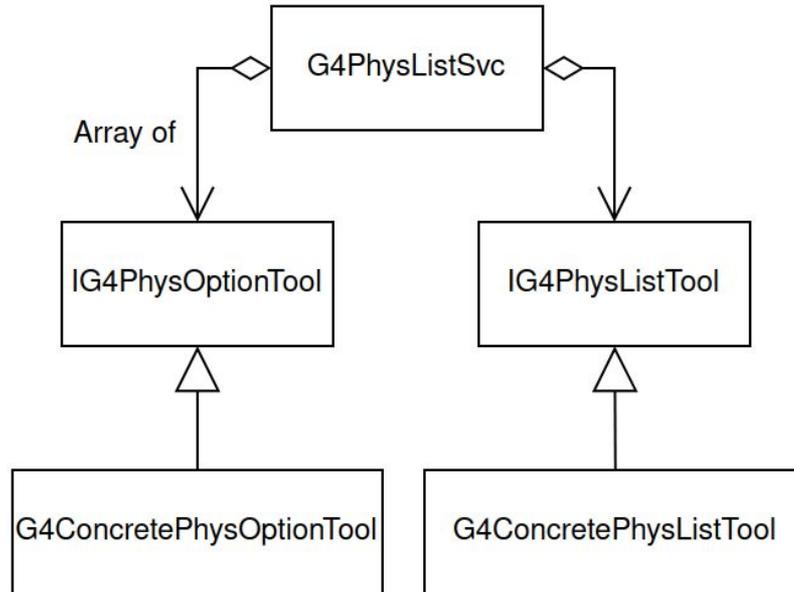
Интеграция Geant4

Концепция: Initialization-классы Geant4 представлены конфигурируемыми сервисами. Генерация событий осуществляется в алгоритме G4Alg.



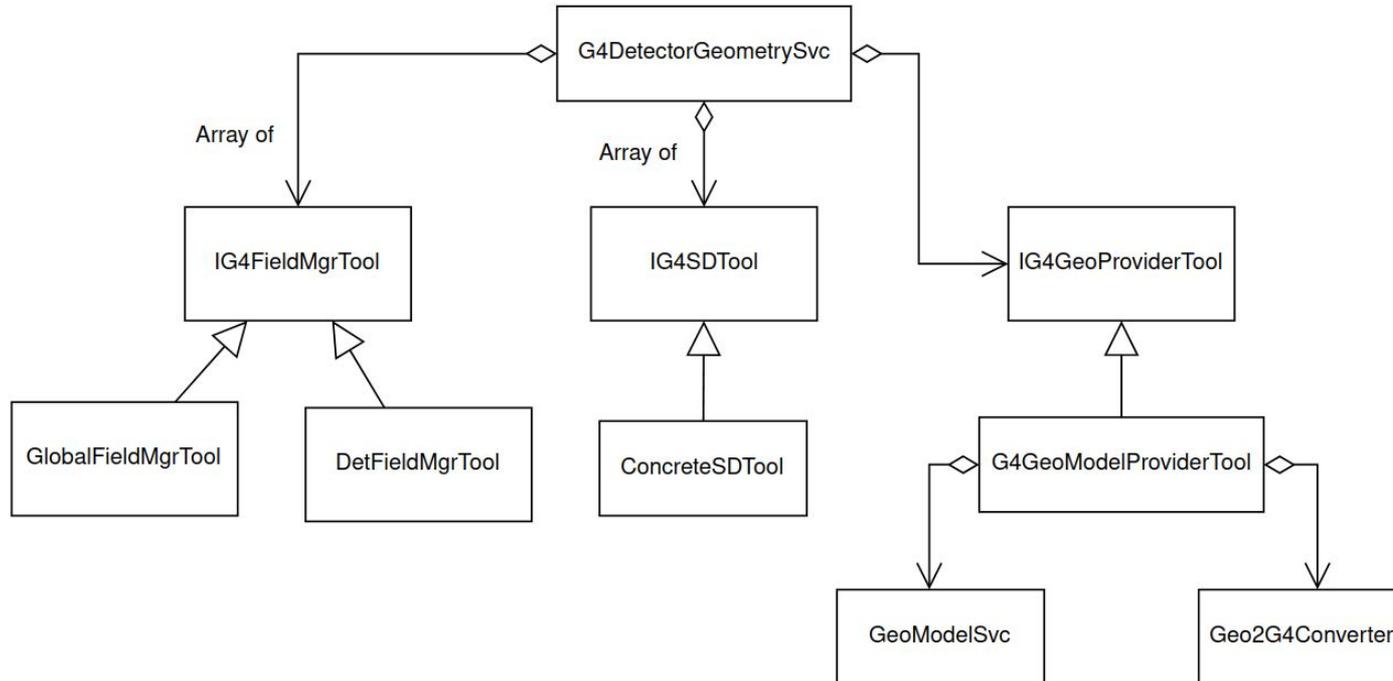
G4PhysListSvc

Концепция: G4PhysListSvc создает объекты класса G4VUserPhysicsList.
G4ConcretePhysListTool создает непосредственно PhysList, опции создаются с помощью G4ConcretePhysOptionTools.



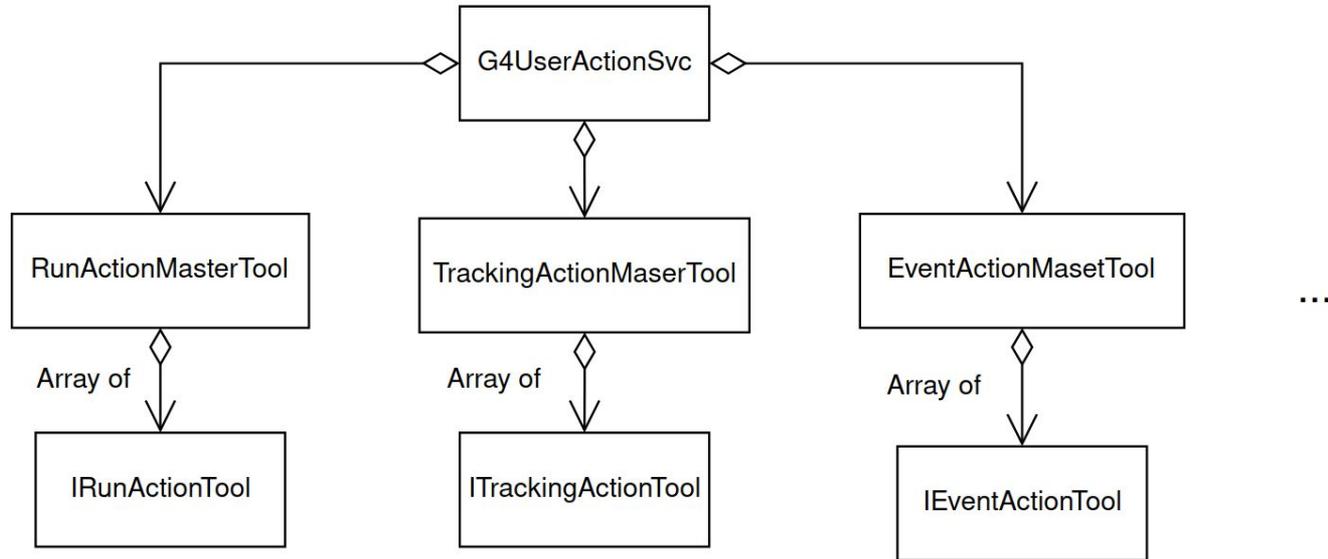
G4DetectorGeometrySvc

Концепция: Geant4 геометрия создается инструментом с интерфейсом IG4GeoProviderTool. ConcreteSDTools создают чувствительные детекторы и назначают их указанным логическим объемам. Инструменты с интерфейсом IG4FieldMgrTool управляют полями.



G4UserActionSvc

Концепция: G4UserActionSvc держит MasterTools, ответственные за создание Action-классов (хуков в Geant4). Каждый хук представляет собой композицию ActionTools.



Заключение и планы

- > Разобрана многопоточная архитектура GaudiHive
- > Завершена интеграция Pythia8
- > Завершена интеграция GeoModel
- > Разработан дизайн интеграции Geant4
- > Написан tutorial
- > Контейнеризация с AlmaLinux 9

Предстоит:

- > Завершить интеграцию Geant4
- > Наладить взаимодействие с другими компонентами ПО SPD



Ковка Сампо, Akseli Gallen-Kallela, 1893