Многопоточный режим моделирования событий в пакете BmnRoot

Multithreaded mode of event simulation in the BmnRoot framework

Margaita Stepanova[1], Andrei Druck[1], Sergei Mertz[2], Konstantin Gertsenberger[2]
[1]Saint Petersburg State University, [2]Joint Institute for Nuclear Research
e-mail: mstep@mms.nw.ru

Dubna, GRID-2021, 6.06.2021

Основные задачи

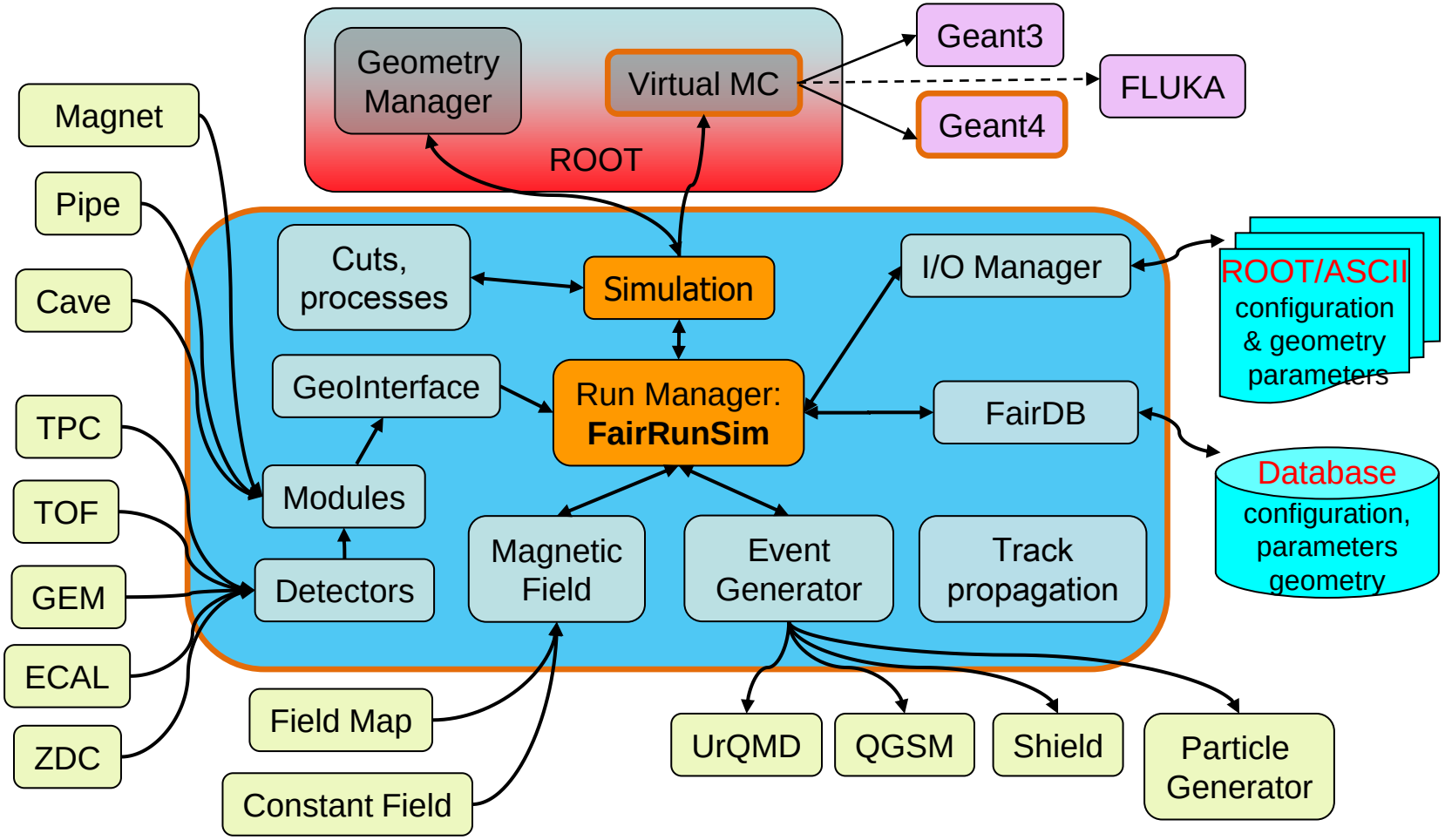
Цель работы – добавление поддержки многопоточности в пакет BmnRoot для ускорение режима моделирования в экспериментах NICA

Задачи

- оценка работоспособности MT-версий базовых пакетов;
- определение критичных классов BmnRoot и их модификация;
- отладка и тестирование

Работа поддержана грантом Российского фонда фундаментальных исследований 18-02-40104 мега

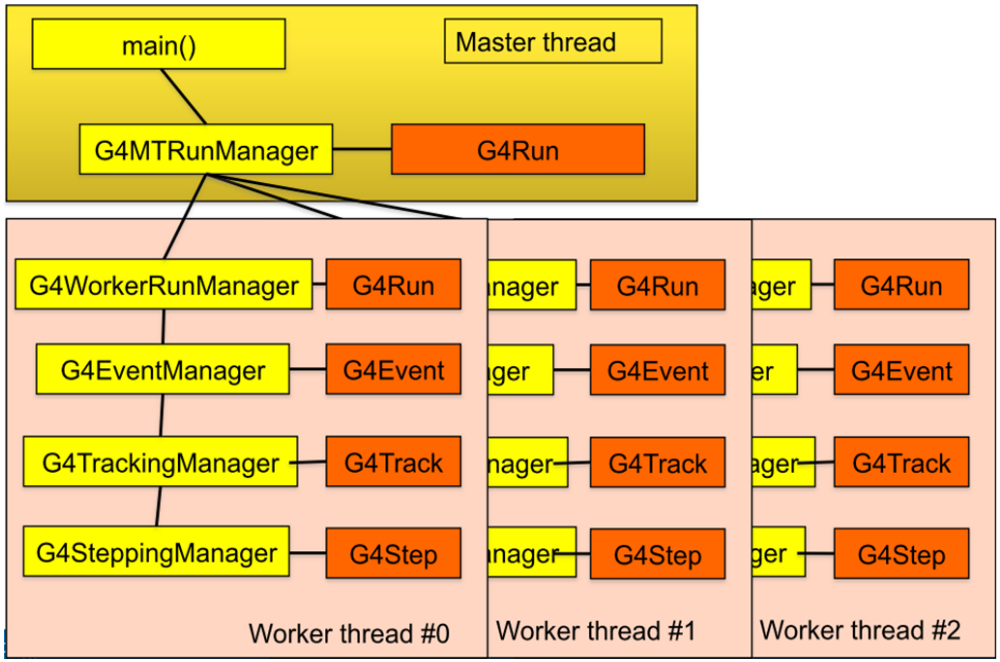
Моделирование в BmnRoot



Переход к многопоточному режиму касается пакетов на разных уровнях иерархии: Geant4, Geant4 VMC, VMC в ROOT, FairRoot и BmnRoot

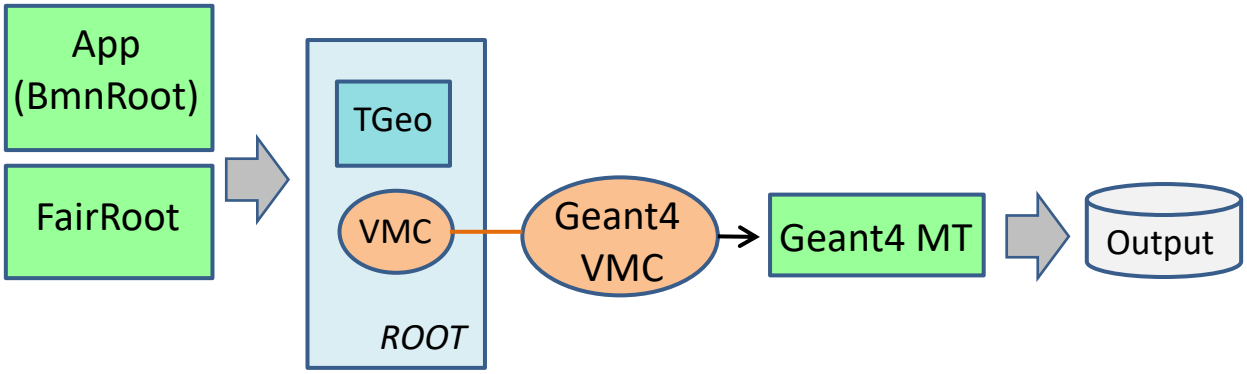
Geant4 MT

- Geant4 MT (>10.03, 2013 г.)
- Параллелизм на уровне событий, модель master-workers
- **G4MTRunManager** управляет рабочими потоками.
- Экземпляры класса **G4WorkerRunManager** – моделирование событий в потоках
- Распараллеливание GEANT4 использует стандарт POSIX.
- Безопасность потоков достигается за счет локального хранилища потоков (Thread-local storage, TLS)
- Возможен гетерогенный параллелизм
  - MPI (интерфейс передачи сообщений) работает вместе с MT.
  - Intel Thread Building Block (TBB): платформа параллелизма на основе задач.



Geant4_VMC MT

- Виртуальный Монте-Карло (VMC) предоставляет единый интерфейс для различных транспортных средств моделирования детекторов.
- Пакет Geant4 VMC - реализация Virtual Monte Carlo для Geant4.
- Первая версия Geant4 VMC MT (3.0) появилась в 2014 году. Адаптирована для многопоточности с использованием того же подхода, что и в Geant4 MT.



Многопоточность в FairRoot

Основные изменения для поддержки MT:

1) Класс **`FairMCApplication`**:

   добавлены функции **`CloneForWorker()`**, **`InitOnWorker()`**

2) Класс **`FairModule`**: добавлена новая виртуальная функция

   ```
   FairModule* CloneModule() const;
   ```

   ее переопределение в режиме MT обязательно для клонирования объектов на воркерах!

3) Класс **`FairRootManager`** расширен: (управление I/O - сделано создание и запись выходных файлов ROOT для каждого потока)

4) Модифицированы **`FairRunSim, FairStack, FairRootManager и FairLogger`**

Для миграции приложений необходимо реализовать новые функции **`TVirtualMCApplication`**, которые затем используются для клонирования приложения и его объектов на воркерах.

Software

	Min (MT+)	Используется	Last**
GEANT4	10.03.p01	10.05.p01	10.7.1
Geant4VMC	3.0	4.0.p1	5-3
ROOT	6/09/04	6.16/00	6.22.08
Fairroot	18.2.0	18.2.0	v18.6.3

Узел ncx-кластера (LHEP)

2x Intel Xeon E5-2697a, DDR4-2400, 2.6GHz, 32(64) cores, 512G RAM, CentOS7, EOS Storage

Сборка ПО с поддержкой MT

Все просто:

- Если ставится FairSoft, то при начальном запуске **`./configure`** надо поменять дефолтное значение

  `# 2) Yes (not compile Geant4 in multihreaded mode)`

  Это ⇔ **`cmake -DGEANT4_BUILD_MULTITHREDED_ON`**

- FairRoot и BmnRoot собираются как раньше.

Установка числа потоков для запуска:

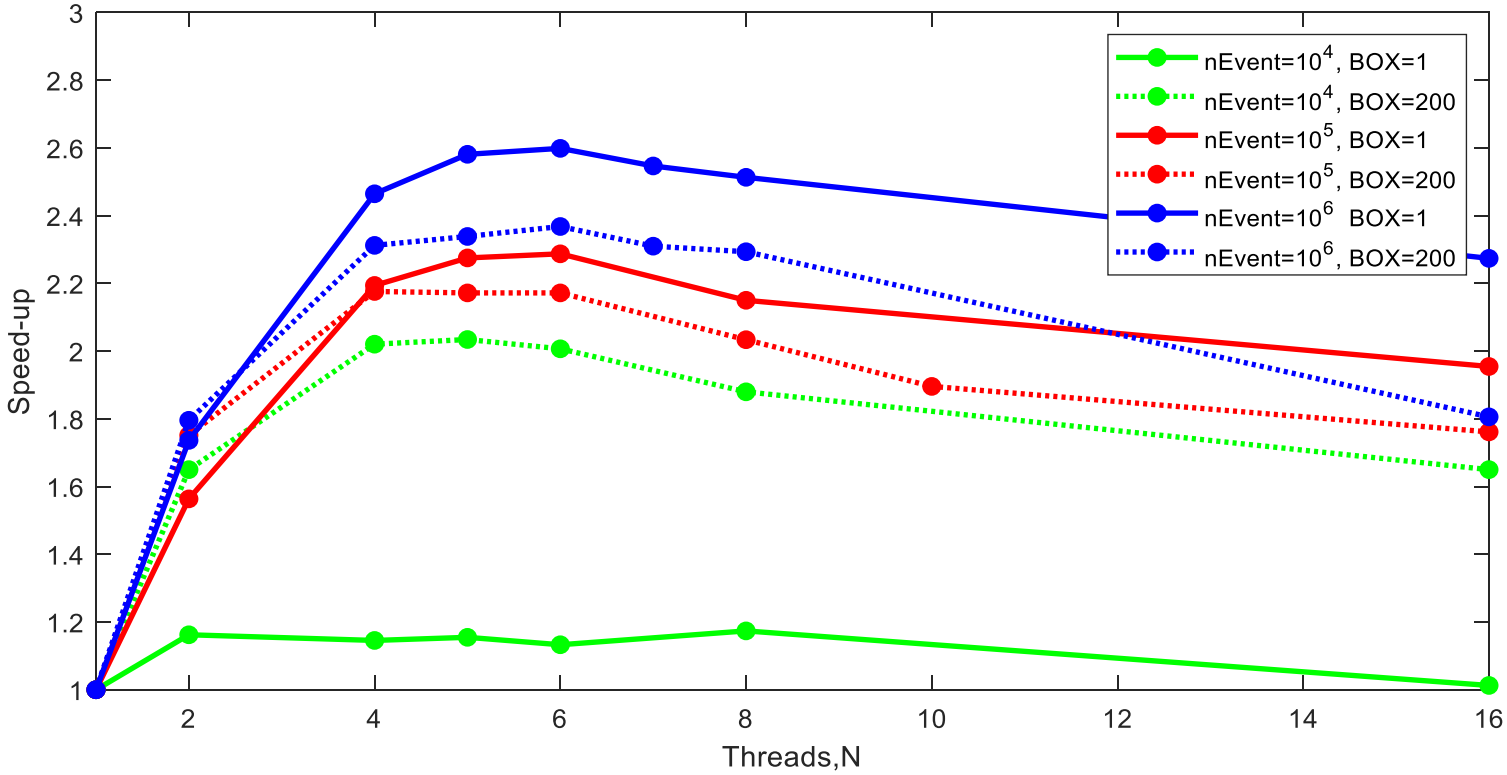
- По умолчанию : 2
- Можно изменить
  - C++ :

    `runManager-> SetNumberOfThreads(8)`
  - через переменную окружения:

    `export G4FORCENUMBEROFTHREADS=8`

Fairroot example “rutherford”

Простое моделирование эксперимента Резерфорда
Ion+BOX Generator
CAVE,Target,RutherfordDetector



Графики зависимости ускорения работы программы от числа потоков при разных параметрах (nEvens, BOX)

Поддержка MT в BmnRoot

Основные изменения на текущий момент:

1) Все детекторы

(BmnFD, BmnBd, BmnCSC, BmDch, BmnEcal, BmnSilicon

BmnMWPC, CbmSts, BmnTOF, BmnTOF1, BmnZdc, BmnPsd,BmnRecoil):

добавлены функции `FairModule* CloneModule() const;`

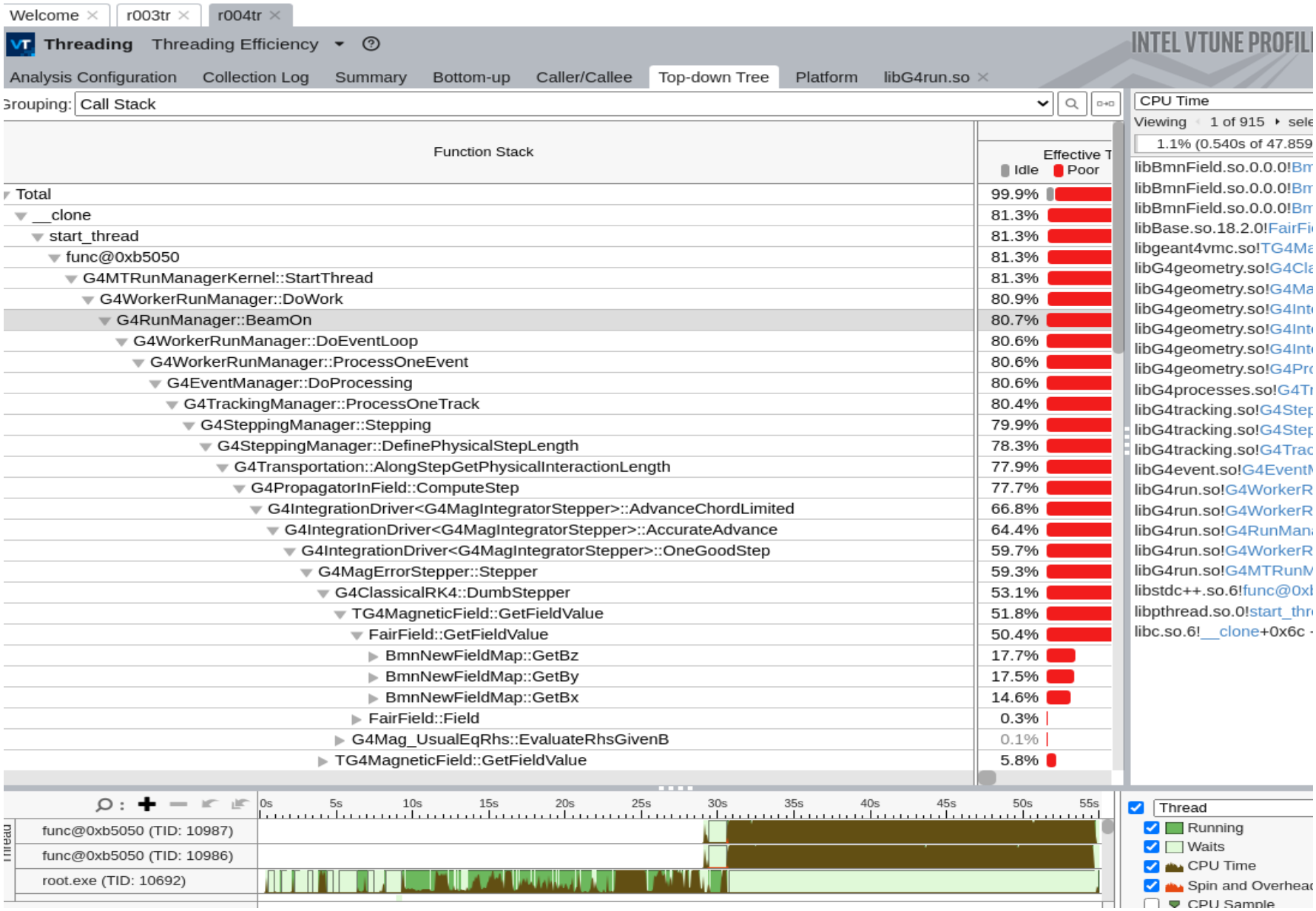
и копирующие конструкторы;

2) Класс CbmSts: добавлены методы для регистрации частиц в потоках;

3) Класс CbmStack: реализовано клонирование объектов в потоках.

Моделирование MT-режиме стало запускаться, но … с катастрофическим падением производительности

Локализация проблемы



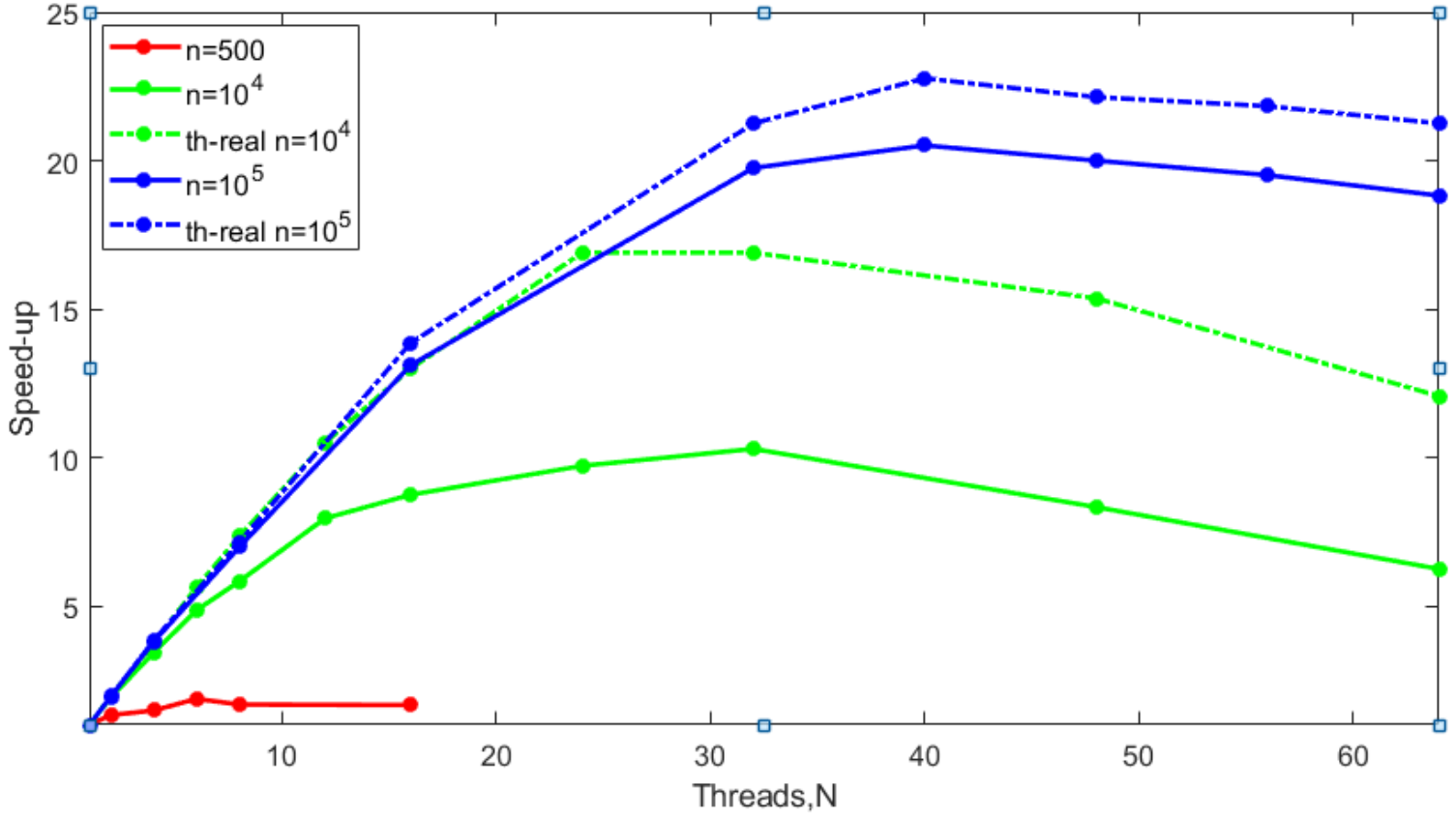
Отключение BmnFieldMap

Постоянное магнитное поле: BmnFieldfConst
Реальная карта поля: BmnFieldMap (чтение из файла)

```
/*
  BmnFieldMap* magField = new    BmnNewFieldMap("field_sp41v5_ascii_Extrap.root");
  Double_t fieldScale = 1200. / 900.;
  magField->SetScale(fieldScale);
  fRun->SetField(magField);
*/
        BmnFieldConst* magField = new BmnFieldConst();
        magField->SetFieldRegion(-300., 300., -300., 300., -300., 300);
        magField->SetField(0., -9. * 0.44, 0.);
        fRun->SetField(magField);
```

Ускорение при постоянном магнитном поле

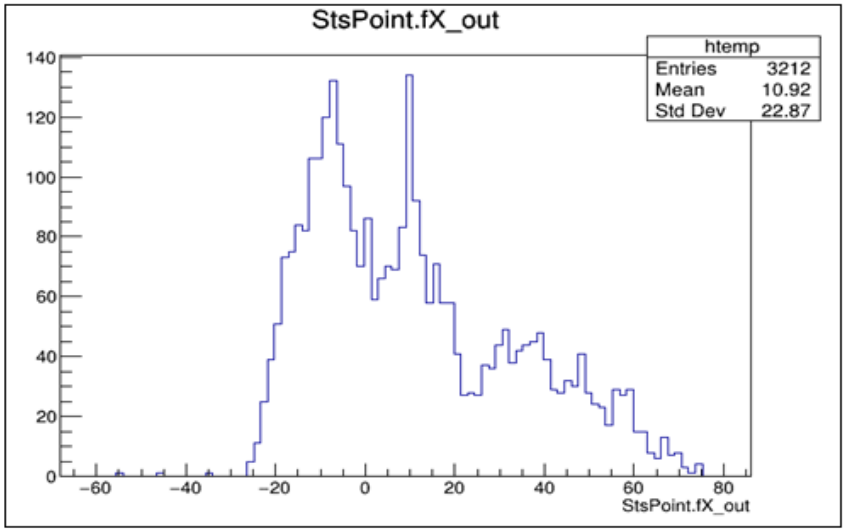
Geometry : CAVE MAGNET STS ECAL ZDC
BOX=(2110,10)



Ускорение зависит от числа событий (nEvent)
Пунктир – время выполнения в тредах(усредненное), сплошная – полное время работы.

О корректности по данным

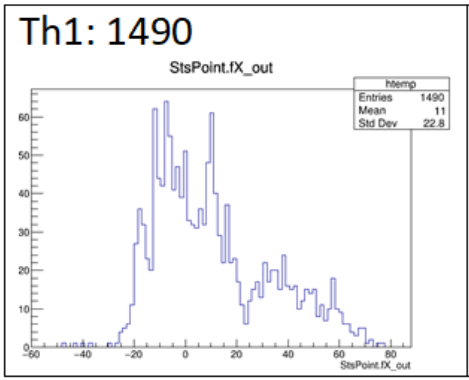
1 поток: 3212



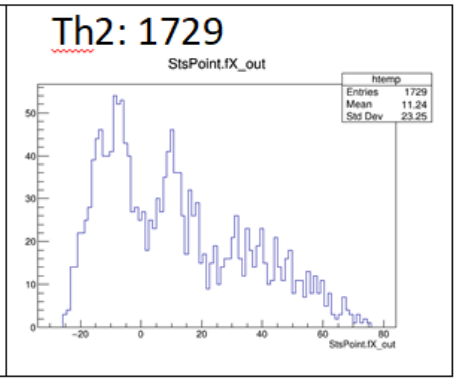
gRandom->SetSeed(1);
FairBoxGenerator* boxGen = new FairBoxGenerator(2212, 1);

2 потока: 3219

Th1: 1490

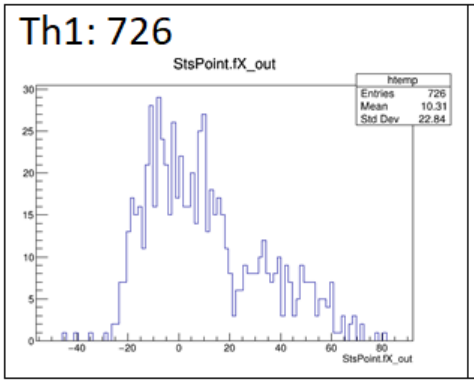


Th2: 1729

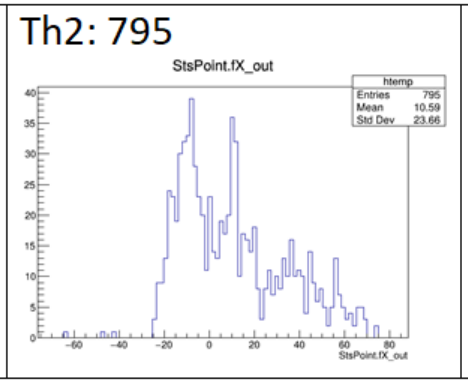


4 потока : 3214

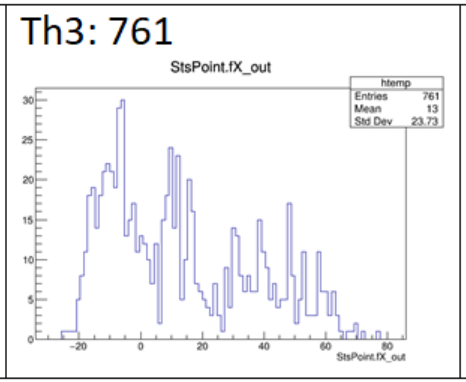
Th1: 726



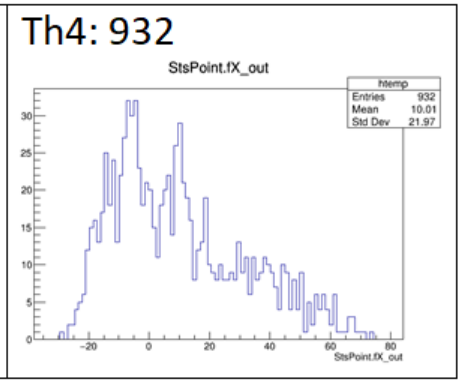
Th2: 795



Th3: 761



Th4: 932



Выводы и планы

Показана возможность реализации многопоточного режима.
Требуется еще значительная доработка кода.

Ближайшие планы

- Модификация детекторов для регистрации событий.
- Устранение проблем с безопасностью потоков.
- Модификация генератора MpdLAQGSMGenerator.
- Поддержка чтения реального магнитного поля из файла.
- Поддержка чтения первичных частиц из файла.
- Корректное слияние выходных файлов в один.

И отдаленные

- Интеграция системой параллельного моделирования на основе FairMQ

Спасибо за внимание!