ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА»
ФИЛИАЛ МГУ В Г. ДУБНЕ

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
КАФЕДРА ФИЗИКИ ПРИКЛАДНЫХ ЯДЕРНЫХ ВЗАИМОДЕЙСТВИЙ

ОРГАНИЗАЦИЯ ХРАНЕНИЯ И ОБРАБОТКИ ФИЗИЧЕСКИХ ДАННЫХ ЭКСПЕРИМЕНТА SPD

Выполнила
студентка группы 202
Будтуева Замира Алановна

Научный руководитель
академик РАН, д. ф.-м. н.
Трубников Григорий Владимирович

Научный консультант
с.н.с. ЛЯП ОИЯИ, к.ф.-м.н.
Прокошин Фёдор Валерьевич

Дубна
2026

Актуальность исследования

Современные эксперименты физики высоких энергий направлены на изучение фундаментальных свойств материи, в частности спиновой структуры нуклонов, что остаётся одной из ключевых нерешённых задач физики.

Эксперимент SPD на ускорительном комплексе NICA предназначен для исследования этих процессов с использованием высокоточных детекторных систем и бестриггерной системы сбора данных, обеспечивающей регистрацию большого количества физических событий.

Такой подход приводит к формированию значительных объёмов информации, требующих разработки эффективных методов хранения, обработки и быстрого доступа к данным.

Цель и задачи работы

Цель

Создание программного обеспечения для индексирования файлов данных эксперимента SPD в рамках разработки Event Index — каталога физических событий, как моделированных, так и полученных с экспериментальной установки.

Задачи:

1. Ознакомиться с целями эксперимента SPD.
2. Рассмотреть системы хранения и обработки данных SPD
3. Оценить требования к хранению и поиску событий для использования в SPD Event Index.
4. Разработать прототип программных модулей для формирования Event Header, содержащего идентификатор события и служебные метаданные.
5. Отработать методики записи и чтения файлов с добавленной идентификационной информацией события.

Эксперимент SPD

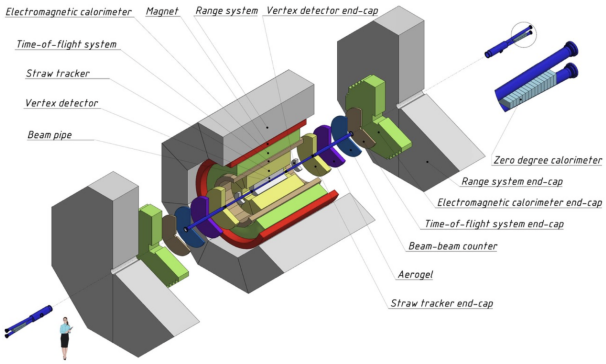


Рисунок 1: Детектор SPD

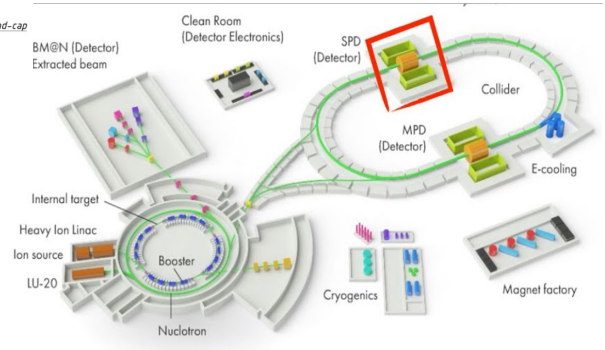


Рисунок 2: Ускорительный комплекс NICA

Сбор данных: Система DAQ

В эксперименте SPD используется бестриггерная система сбора данных (triggerless DAQ)

– Частота взаимодействий достигает $3{\times}10^6$ $с^{-1}$, число каналов считывания — до 600 000, поток данных — до 20 ГБ/с;
– Использование классической триггерной системы затруднено из-за высокой частоты событий и необходимости реконструкции импульса и вершины для их отбора;
– В triggerless DAQ все сигналы непрерывно считываются, снабжаются временными метками и группируются во временные интервалы (slice 10–100 мкс, frame 0.1–10 с) для последующей обработки и хранения.

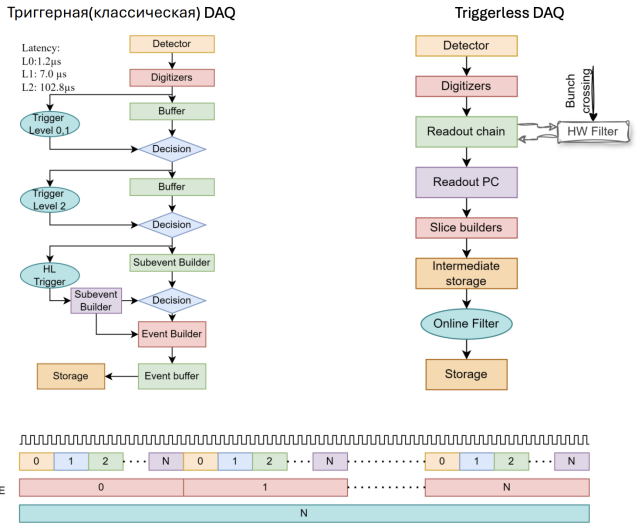


Рисунок 3: Схема бестриггерной DAQ (Сессия ОЯФ РАН – А.Бойков)

Сбор данных: Система SPD Online Filter

Система Online Filter является первым этапом обработки данных в эксперименте SPD после системы сбора данных и предназначена для первичной обработки и отбора событий в условиях бестриггерного режима работы.

Главная цель онлайн-фильтра — быстрая реконструкция событий SPD и подавление фоновых событий как минимум в 20 раз. Он выполняет подготовку информации для последующего хранения и офлайн-анализа.

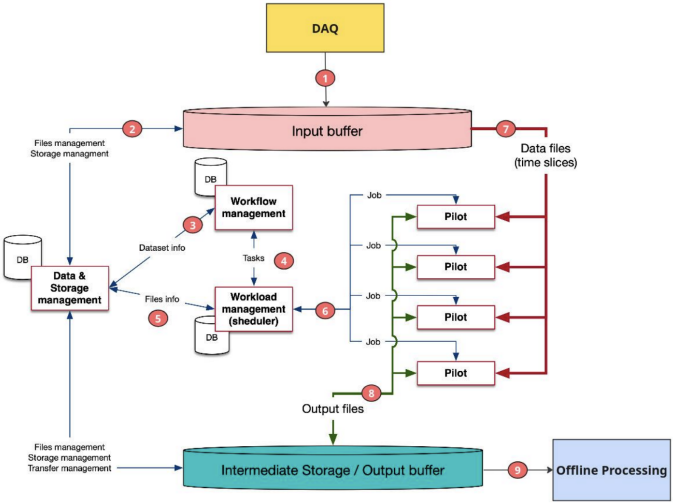


Рисунок 4: Основные компоненты SPD Online Filter (X SPD collaboration meeting – N. Greben)

Оценка объёмов данных

- Скорость потока данных: 20 ГБ/с, что соответствует 10 ПБ/год необработанных (RAW) данных
- Годовой объём: около 30 млрд событий
- Объём данных и частота событий сопоставимы с крупнейшими экспериментами, такими как ATLAS на LHC. Также ожидается сравнимый объём данных моделирования (MC)
- В результате реконструкции и обработки для физического анализа будет создано несколько экземпляров каждого события в различных форматах
- Данные событий в различных форматах хранятся в файлах, объединённых в датасеты, размещённые в распределённой системе хранения.

Системы хранения и обработки данных SPD

Rucio — система управления распределёнными данными:

- каталог файлов и датасетов;
- размер, контрольные суммы, число событий;
- расположение датасетов и их реплик.

PanDA — система управления распределённой обработкой:

- задачи обработки;
- входные и выходные датасеты;
- статусы заданий и вычислительных ресурсов.

Event Index позволяет найти экземпляр события, относящийся к определённой стадии обработки, хранящийся в файле из соответствующего датасета.

Event Index

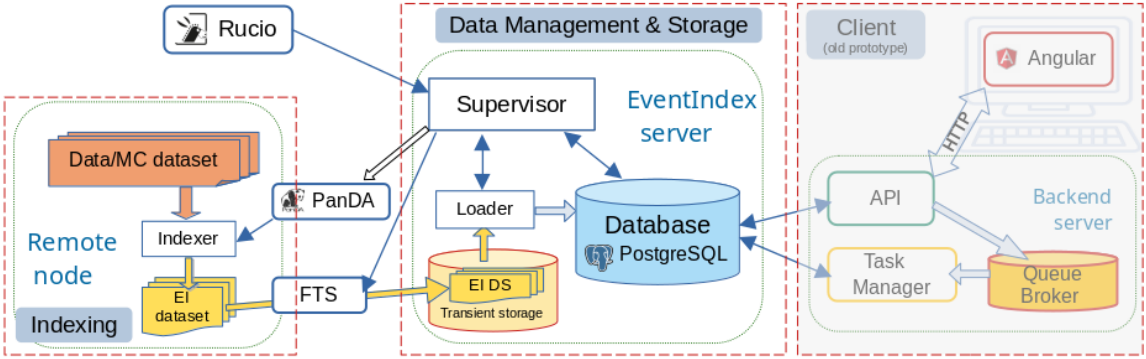


Рисунок 5: Схема Event Index

EventIndex разрабатываетя как каталог физических событий эксперимента SPD. Его задача — не хранить сами данные событий, а хранить сведения, по которым можно быстро найти событие и связанные с ним файлы/датасеты на разных стадиях обработки. На ранней стадии система будет работать с моделированными данными, но архитектура должна быть рассчитана на последующий переход к реальным данным с установки.

Проблема индексации событий

На текущем этапе EventIndex рассматривается для работы со смоделированными событиями. Такие события проходят несколько стадий обработки — от генерации до реконструкции и анализа. На каждой стадии данные могут сохраняться в новых файлах и датасетах, поэтому необходимо сохранять устойчивую связь между исходным событием и его производными представлениями с помощью уникального идентификатора события.

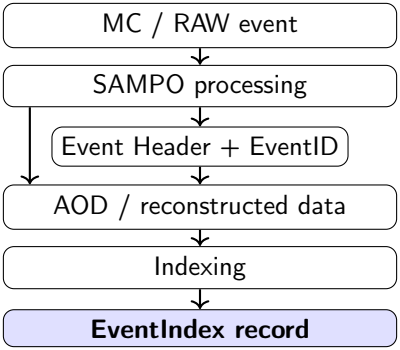


Рисунок 6: Схема индексации

Event Header

Event Header представляет собой служебный блок данных, добавляемый к событию. Он предназначен для хранения идентификатора события и метаданных, необходимых для последующего индексирования, поиска и сопоставления данных на разных стадиях обработки. В рамках работы был реализован прототип Event Header, который может быть добавлен в поток обработки SAMPO и использоваться как основа для формирования записей в SPD Event Index.

```cpp
void HepMCHeaderWriterAlg::addHeaderAttributes(HepMC3::GenEvent& event, long seed) const {
    const long runNumber = m_runNumber.value();
    const long eventNumber = resolveEventNumber(event);
    const std::string fuid = buildFUID(runNumber, eventNumber);
    const std::string metadata = buildMetadata(event);

    event.set_event_number(static_cast<int>(eventNumber));
    event.add_attribute("sampo_run_number", makeStringAttribute(runNumber));
    event.add_attribute("sampo_event_number", makeStringAttribute(eventNumber));
    event.add_attribute("sampo_seed", makeStringAttribute(seed));
    event.add_attribute("sampo_fuid", makeStringAttribute(fuid));
    event.add_attribute("sampo_metadata", makeStringAttribute(metadata));
}
```

Рисунок 7: Часть кода прототипа Event Header

Event Header

Event Header может служить источником информации для формирования записи в EventIndex. При индексировании из заголовка события могут быть считаны идентификатор события и служебные метаданные, а также добавлены ссылки на файл, датасет и производные данные, полученные на последующих стадиях обработки. Показана предварительная структура Event Header, набор полей будет уточнятся в ходе разработки.

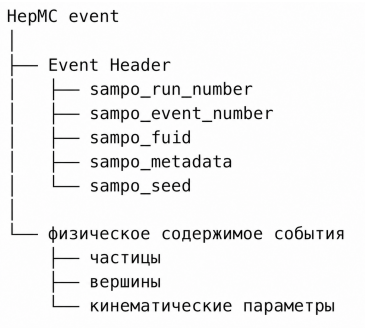


Рисунок 8: Предварительная структура события в HepMC-файле

```
HepMC::Version 3.03.00
HepMC::Asciiv3-START_EVENT_LISTING

E 0 45 88
U GEV MM
W 1.0000000000000000000000e+00

A 0 sampo_event_number 0
A 0 sampo_fuid 7_0
A 0 sampo_metadata n_particles=88;n_vertices=45
A 0 sampo_run_number 7
A 0 sampo_seed 0

P 1 0 2212 0.0000000000000000e+00 0.0000000000000000e+00 13.467...
P 2 1 2    1.7170853581764609e-01 7.5734128639446618e-01 2.069...
P 3 1 2    1.3917700884995554e-01 -1.7884628160800831e-01 4.947...
```

Рисунок 9: Выходной файл с Event Header

Заключение

В результате работы была рассмотрена задача индексации событий в эксперименте SPD. Поскольку одно событие может проходить несколько стадий обработки и сохраняться в разных файлах, для его последующего поиска необходим устойчивый идентификатор. В рамках практической части был реализован прототип Event Header, который добавляет к событию номер запуска, номер события, уникальный идентификатор и базовые метаданные. Этот заголовок был встроен в пайплайн SAMPO, а результатом стал выходной файл с добавленной служебной информацией. Таким образом, была подготовлена основа для дальнейшей связи обработанных событий с системой SPD Event Index.

Спасибо за внимание!

Back up

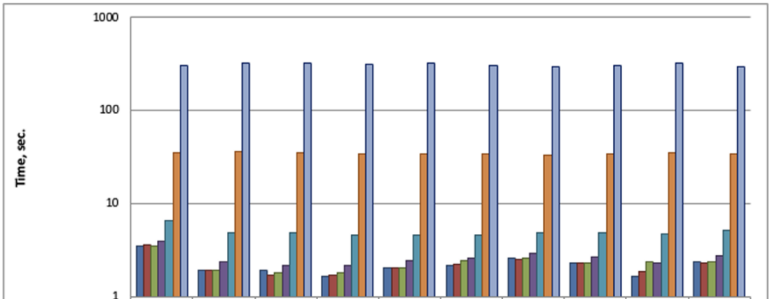
SAMPO

SAMPO — унифицированная среда многоэтапной обработки данных эксперимента SPD. Она использует фреймворк Gaudi, хорошо зарекомендовавший себя в крупных ускорительных экспериментах, таких как ATLAS на БАК. Ключевые особенности:

- Модульность
- Многопоточность
- Гибкая конфигурация
- Интеграция библиотек

Таким образом, SAMPO выступает как каркас для всех стадий: от генерации и моделирования до реконструкции и анализа.

Результаты оптимизации



	asyncpg	pg8000	psycopg2	pygresql	sql_alchemy(psycopg2)	sql_alchemy(psycopg3)	sql_alchemy(async)	sql_alchemy(pg8000)	psycopg3	psycopg3(async)
10 events	3,59	1,93	1,93	1,66	2,10	2,21	2,63	2,35	1,66	2,41
100 events	3,64	1,94	1,72	1,72	2,06	2,27	2,55	2,30	1,87	2,33
1 000 events	3,58	1,97	1,82	1,82	2,05	2,49	2,64	2,32	2,42	2,42
10 000 events	3,96	2,38	2,22	2,18	2,51	2,66	2,98	2,69	2,34	2,78
100 000 events	6,69	4,90	4,92	4,65	4,70	4,67	5,01	4,95	4,75	5,24
1 000 000 events	36,04	36,99	35,38	34,55	34,22	34,31	33,14	34,30	35,27	34,29
10 000 000 events	311,29	322,49	320,81	316,66	322,78	307,54	299,82	310,25	324,60	301,26

Рисунок 10: Результаты тестирования модулей с чтением данных из ROOT-файлов (по горизонтали: тестируемые модели; по вертикали: логарифмическая шкала времени в секундах)

Сбор данных: Система SPD Online Filter

Основная логика работы:
1.DAQ — система сбора данных генерирует поток событий.
2. Данные поступают во входной буфер (Input buffer), где временно накапливаются.
3. Data Storage management управляет хранением файлов и формирует наборы данных (datasets).
4. Workflow management определяет, какие задачи нужно выполнить над данными.
5. Workload management (scheduler) распределяет задачи по вычислительным узлам.
6. Задачи отправляются на несколько рабочих процессов (Pilot), которые выполняют фильтрацию/обработку данных параллельно.
7. Исходные данные передаются в эти процессы порциями (time slices).
8. Результаты обработки сохраняются как выходные файлы.
9. Эти файлы помещаются в Intermediate Storage / Output buffer, откуда затем используются для офлайн-обработки.

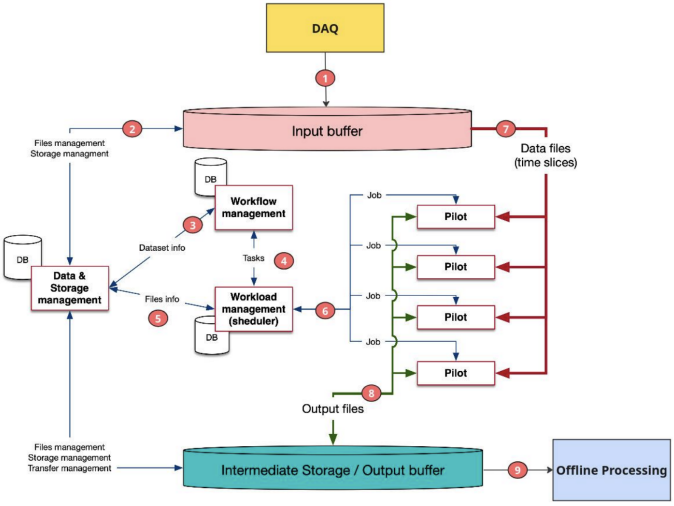


Рисунок 4: Основные компоненты SPD Online Filter (X SPD collaboration meeting – N. Greben)

Дальнешие планы:

Дальнейшая работа связана с:

- развитием как самого прототипа Event Header, так и всей цепочки индексирования.
- оптимизацией схемы БД для уменьшения размера и повышения производительности;
- оптимизацией загрузки данных в рамках полного цикла и на высокопроизводительном оборудовании;
- модернизацией и оптимизацией клиента и клиент-серверного взаимодействия.