

УДК 004.42:539.1.074.22

Программная реализация метода быстрой кластеризации данных время-проекционной камеры эксперимента MPD/NICA

А.В. Крылов¹✉, В.А. Крылов¹, А.С. Будников²

¹Объединенный институт ядерных исследований, Дубна, Россия

²Воронежский государственный университет, Воронеж, Россия

В статье рассматривается метод и программная реализация быстрой кластеризации данных время-проекционной камеры эксперимента MPD/NICA. Метод предназначен для предварительного преобразования цифровых амплитудно-временных отсчетов детектора в компактный набор хитов, пригодный для визуализации событий и контроля качества данных. Обработка выполняется на уровне секторов плоскости считывания данных, что позволяет использовать естественную декомпозицию события и многопоточную реализацию алгоритма. В работе описаны модель входных данных, основные этапы кластеризации, формирование выходного представления и результаты экспериментальной оценки.

Ключевые слова: MPD, NICA, TPC, онлайн-кластеризация, обработка событий, многопоточная обработка, контроль качества данных.

Software Implementation of a Fast Clustering Method for Time Projection Chamber Data of the MPD/NICA Experiment

A.V. Krylov¹✉, V.A. Krylov¹, A.S. Budnikov²

¹Joint Institute for Nuclear Research, Dubna, Russia

²Voronezh State University, Voronezh, Russia

The paper presents a software implementation of a fast clustering method for time projection chamber data of the MPD/NICA experiment. The method is intended for preliminary conversion of digital amplitude-time detector samples into a compact set of spatial hits suitable for event visualization and data quality control. Processing is performed at the level of sector fragments, which makes it possible to use the natural decomposition of an event and a multithreaded implementation of the algorithm. The paper describes the input data model, the main clustering stages, the output representation and the experimental evaluation procedure.

Keywords: MPD, NICA, TPC, online clustering, event processing, multithreaded processing, data quality control.

Введение

В эксперименте MPD на коллайдере NICA основная трековая информация поступает от время-проекционной камеры (TPC) [1]. Этот детектор регистрирует ионизационные электроны, возникающие при прохождении заряженных частиц через рабочий объем камеры. Под действием электрического поля электроны дрейфуют к детектирующим плоскостям, где их сигнал считывается системой катодных площадок (падов). Зная положение площадки и время дрейфа, можно восстановить пространственное положение локального отклика детектора.

В рамках разрабатываемой системы онлайн-обработки данные TPC рассматриваются не как окончательный набор реконструированных физических

объектов, а как поток цифровых отсчетов амплитуды сигнала. Для задач оперативной визуализации и контроля качества требуется преобразовать этот поток в компактное представление. Полная реконструкция треков и идентификация частиц на данном этапе не выполняются, поскольку они требуют более сложных вычислительных процедур. Достаточно получить устойчивую оценку положения локальных откликов и их интегральных характеристик.

Цель работы состоит в описании метода кластеризации ТРС-сигналов, его программной реализации на языке C++ и экспериментальной оценке вычислительных характеристик алгоритма. Отдельно рассматривается пригодность метода к параллельной обработке и использованию в составе онлайн-систем визуализации и мониторинга.

1. Исходные данные и постановка задачи онлайн-кластеризации ТРС-сигналов

1.1. Модель входных данных

Считывание информации ТРС выполняется по секторам. В используемой программной модели один сектор содержит набор рядов катодных площадок. Для каждого сработавшего канала фиксируется последовательность амплитудно-временных отсчетов, соответствующих дискретным временным бакетам (интервалам). Один отсчет описывается номером сектора, номером ряда, номером площадки, номером временного бакета и значением амплитуды аналого-цифрового преобразования.

Для данных моделирования дополнительно может присутствовать служебная информация, например идентификатор изначального трека, с которым связан данный сигнал. Эта информация используется при проверке алгоритма и построении контрольных распределений, однако не является обязательной для выполнения самой кластеризации.

Входной поток алгоритма можно представить как последовательность элементов:

$$d_i = \{s_i, r_i, p_i, t_i, a_i\}, \quad (1)$$

где s_i обозначает номер сектора ТРС, r_i – номер ряда, p_i – номер сенсорной площадки, t_i – номер временного бакета, а a_i – амплитуду сигнала. Пространственная координата отклика в локальной системе сектора восстанавливается на основании геометрии пад-плоскости и положения временного бакета. После этого локальные координаты могут быть преобразованы в глобальную систему координат детектора.

Данные разных секторов не пересекаются на этапе кластеризации, поэтому секторные фрагменты события могут обрабатываться независимо. В рассматриваемой реализации ряды внутри каждого сектора обрабатываются последовательно.

1.2. Формат данных детектора ТРС

Система считывания детектора находится на торцах ТРС и представлена многопроводными пропорциональными камерами с катодными площадками для съема данных. Согласно техническому проекту время-проекционной камеры эксперимента MPD/NICA, торцевые плоскости разделены на 24 считывающих сектора, а данные с падовой плоскости передаются через front-end электронику [6]. Электроника считывает сигнал, характеризующийся быстрым временем нарастания и длительным спадом вследствие движения положительных ионов.

Для съема данных с одного сектора используются фронтенд-карты с микросхемами SAMPA. Схема взаимодействия фронтенд-карт, FPGA и последующих компонентов считывания показана на рисунке 1 [6]. Одна микросхема SAMPA содержит 32 канала считывания, каждый из которых включает аналоговый тракт, 10-битный аналого-цифровой преобразователь и цифровую обработку сигнала [5]. Каждая карта SAMPA может собирать информацию с 32 каналов, а фронтенд-карта обеспечивает съем с 64 каналов.

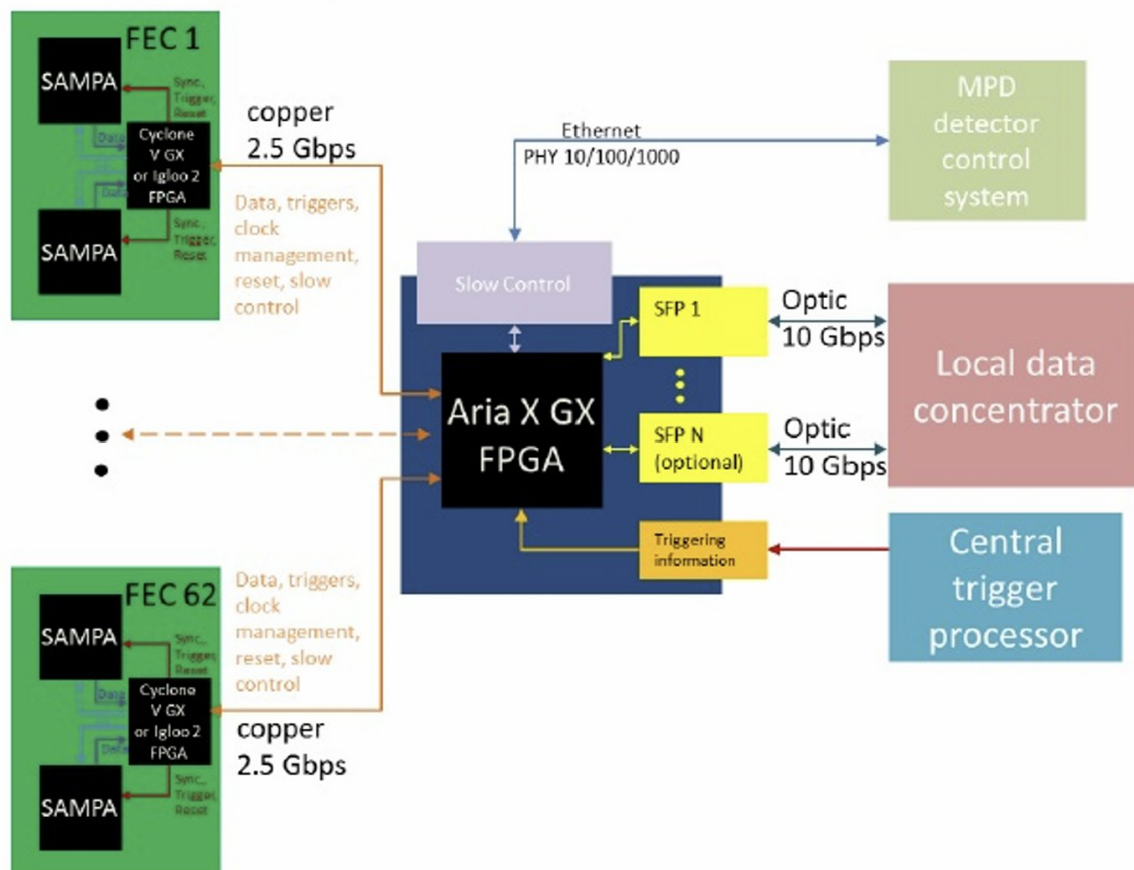


Рисунок 1. Схема электронной карты съема данных с сектора детектора ТРС

Пакет данных каждого канала карты SAMPA состоит из заголовка и полезной нагрузки. Заголовок содержит тип пакета, код канала и чипа, размер данных и код коррекции ошибок по Хэммингу. После заголовка записываются данные, представленные в виде 311 10-битных слов, соответствующих временным бакедам. На рассматриваемом этапе сигналы передаются в необработанном виде, то есть без подавления нулей и сжатия.

Основные параметры микросхемы SAMPA приведены в таблице 1.

Таблица 1

Параметры карты SAMPA

Параметр	Значение
Количество каналов	32
Разрешение АЦП	10 бит
Чувствительность	20 мВ/фКл или 30 мВ/фКл

2. Метод и программная реализация кластеризации ТРС-сигналов

Алгоритм обрабатывает АЦП-отсчеты (ADC digits) последовательно, сохраняя естественную структуру ТРС-данных. Сначала выделяются локальные временные отклики на отдельных пад-каналах. Затем они объединяются в кластеры внутри ряда, после чего для каждого кластера формируется хит в локальной системе координат детектора (сектор, ряд, пад, временной бакет).

Алгоритм реализован в виде C++ библиотеки, предназначенной для использования в системе онлайн-обработки. Основными объектами библиотеки являются структуры, описывающие сектор, ряд, локальный пад-кластер, объединенный ТРС-кластер и выходной хит. Общая последовательность обработки показана на рисунке 2.

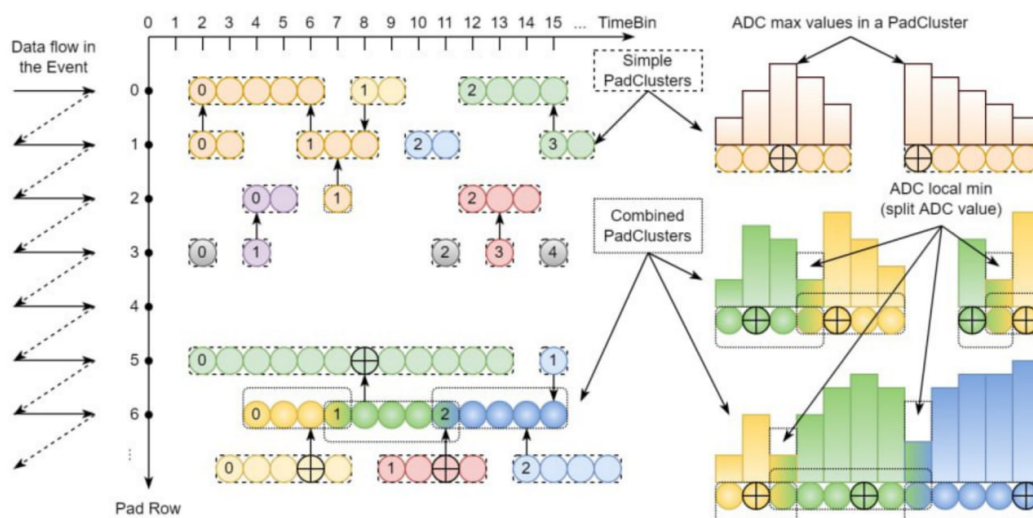


Рисунок 2. Последовательность обработки ТРС-сигналов в библиотеке кластеризации

Первым этапом обработки является анализ сигнала на одном пад-канале. Последовательность ненулевых или превышающих заданный порог АЦП-отсчетов рассматривается как локальный временной отклик. Если соседние временные бакеты принадлежат одному непрерывному сигналу, они объединяются в один пад-кластер. Для такого кластера вычисляются минимальный и максимальный временной бакет, суммарная амплитуда, положение максимума и предварительная оценка центра сигнала.

На одном пад-канале может наблюдаться несколько близких откликов. В этом случае локальный минимум между двумя максимумами используется как признак разделения. Если минимум выражен слабо, кластер может быть помечен как неопределенный и отброшен или разделен с использованием пропорционального распределения амплитуды между соседними откликами.

2.1. Объединение кластеров в пространстве пад-время и разделение кластеров в условии наложения

После выделения локальных откликов на отдельных пад-каналах выполняется объединение пад-кластеров внутри одного ряда ТРС. Анализируются кластеры, расположенные на соседних площадках. Если их временные интервалы пересекаются, такие объекты считаются принадлежащими одному физическому отклику и объединяются в общий кластер.

Формально два локальных пад-кластера C_i и C_j могут быть объединены, если они принадлежат соседним пад-каналам и их временные интервалы имеют пересечение:

$$|p_i - p_j| = 1, \quad (2)$$

$$[t_i^{\min}, t_i^{\max}] \cap [t_j^{\min}, t_j^{\max}] \neq \emptyset. \quad (3)$$

В результате получается набор завершенных кластеров в пределах текущего ряда. Объединение кластеров выполняется независимо для каждого ряда.

В условиях высокой загрузки детектора некоторые кластеры могут перекрываться другими. При рассмотрении подобного кластера как единое целое, это приводит к некорректной оценке центра кластера и существенному смещению позиции хита.

Для уменьшения подобного эффекта алгоритм производит разделение составного кластера на простые посредством рассмотрения проекций на ось пад-каналов и ось временных бакетов. Если внутри объединенного кластера обнаруживается несколько локальных максимумов, разделенных локальным минимумом, кластер может быть разбит на два или более объекта. Такой подход позволяет приблизительно разделить наложенные сигналы без применения глобальной оптимизации. Сигнал на границе подобных кластеров делится пропорционально амплитудам локальных максимумов соседних кластеров.

2.2. Оценка центра кластера

После формирования кластера необходимо определить его положение. Простое вычисление центра тяжести по амплитудам является быстрым, но может давать смещение при асимметричной форме сигнала, шуме или частичном наложении откликов. Поэтому в библиотеке используется параметризация формы сигнала функцией Гаусса:

$$y(x) = Ae^{\frac{-(x-\mu)^2}{2\sigma^2}}. \quad (4)$$

После логарифмического преобразования выражение приводится к квадратичной форме:

$$\ln(y) = a + bx + cx^2. \quad (5)$$

Параметры исходной функции восстанавливаются по формулам:

$$\mu = -\frac{b}{2c}, \sigma = \sqrt{-\frac{1}{2c}}, A = e^{\left(a - \frac{b^2}{4c}\right)}. \quad (6)$$

Основным преимуществом подобного подхода является алгоритмическая скорость. Оценка центра кластера сводится к решению системы линейных уравнений и не требует итеративного подхода. Для онлайн-обработки данных подход особенно важен, так как число локальных кластеров в одном событии может быть большим, а основным критерием является скорость обработки.

2.3. Декомпозиция события и формирование выходного представления

В алгоритме используется параллелизация обработки по независимым секторам. Данные события разделяются на 24 секторных фрагмента, что соответствует

естественному геометрическому делению детектора. Обработка секторов не требует обмена промежуточными данными между потоками, поскольку объединение АЦП-отсчетов и формирование кластеров выполняются только в пределах одного сектора. После обработки каждого независимого сектора данные объединяются в единый выходной поток.

Входные данные каждого сектора имеют одинаковую структуру и размер, по причине отсутствия алгоритма подавления нулей (zero suppression) на стороне алгоритмов сбора данных детектора. Это приводит к равномерной скорости обработки секторов события по всем потокам.

Ряды катодных площадок каждого сектора обрабатываются последовательно. Несмотря на независимость данных рядов, декомпозиция не производилась, поскольку чрезмерное дробление события на отдельные ряды увеличило бы накладные расходы на маршрутизацию и сериализацию.

Дальнейшее повышение производительности требует либо параллельной обработки нескольких событий, либо декомпозиции вычислений внутри сектора, что в данной работе не исследовалось.

Выходными данными алгоритма является набор объектов, описывающих восстановленные хиты детектора. Каждый объект содержит локальные координаты кластера – номер ряда, номер пада и номер временного бакета, а также дополнительную информацию, такую как идентификатор кластера, амплитуду сигнала всего кластера (интеграл), а также максимальную амплитуду (пиковое значение). Также учитывается поправка на задержку формирования сигнала в электронике считывания, заранее известная по документации.

Итоговое представление данных существенно компактнее исходного набора АЦП-отсчетов и может быть использовано для визуализации посредством Дисплея событий (Event Display) или для мониторинга гистограмм качества набора данных (Quality Assurance histograms).

3. Экспериментальная оценка C++ алгоритма кластеризации

Для проверки производительности и качества работы разработанной библиотеки была проведена экспериментальная оценка C++ алгоритма кластеризации на моделированных данных детектора ТРС. Здесь рассматривается только вычислительное ядро алгоритма: преобразование входных АЦП-отсчетов в набор восстановленных кластеров и хитов.

Время чтения и передачи данных, а также время записи результатов в измерения не включались. Поэтому результаты ниже следует интерпретировать как оценку вычислительной стоимости самого алгоритма, а не всей цепочки подготовки и доставки данных.

3.1. Методика измерений и тестовые данные

Тестирование проводилось на наборе моделированных событий, полученных с использованием генератора UrQMD [2, 3]. В качестве входного набора использовались события с множественностью minimum bias. Общее число событий в тестовом наборе составило 3000. Моделированные данные хранились в файле формата ROOT [4].

Для каждого события фиксировались число АЦП-отсчетов, число восстановленных хитов и время выполнения алгоритма. Измерения выполнялись на рабочей станции, конфигурация которой приведена в таблице 2.

Таблица 2

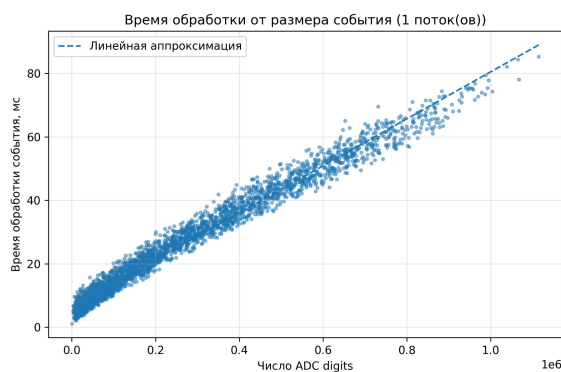
Конфигурация вычислительного стенда для тестирования алгоритма

Параметр	Значение
Генератор событий	UrQMD
Тип событий	minimum-bias
Количество событий	3000
Процессор	Intel Xeon Platinum 8380
Оперативная память	128 GB DDR4-3200
Накопитель	SSD 1 TB
Измеряемая часть	C++ алгоритм кластеризации

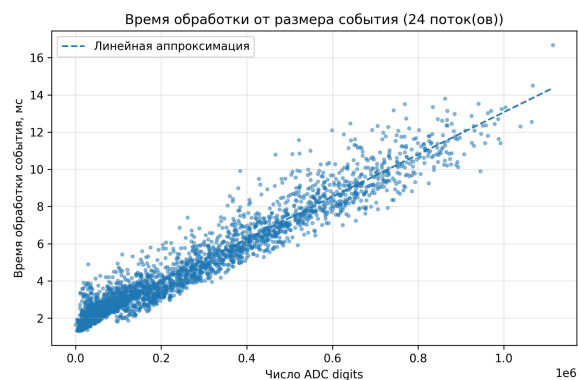
Для оценки масштабируемости алгоритм запускался с различным числом потоков. В качестве базового режима использовалась последовательная обработка секторных данных (1 поток). Многопоточные режимы соответствовали обработке независимых секторных фрагментов события. Поскольку детектор ТРС состоит из 24 секторов, режим с 24 потоками является очевидной конфигурацией для параллельной обработки одного события.

3.2. Зависимость времени обработки от размера события

Одной из основных характеристик алгоритма является зависимость времени обработки от числа входных данных. На рисунке 3 показана зависимость времени обработки события от числа ADC-отсчетов для последовательного режима и для режима с 24 потоками. В обоих случаях наблюдается близкий к линейному рост времени обработки при увеличении размера события.



а) 1 поток



б) 24 потока

Рисунок 3. Зависимость времени обработки от числа АЦП-отсчетов:

а) последовательная обработка; б) обработка в 24 потока

3.3. Распределение времени обработки события

Для анализа устойчивости времени выполнения были построены распределения времени обработки событий. На рисунке 4 показано распределение для однопоточного режима и при распараллеливании на 24 независимых потока. Однопоточный режим имеет выраженный правый «хвост», связанный с событиями большего размера. При использовании 24 потоков распределение смещается в область меньших времен.

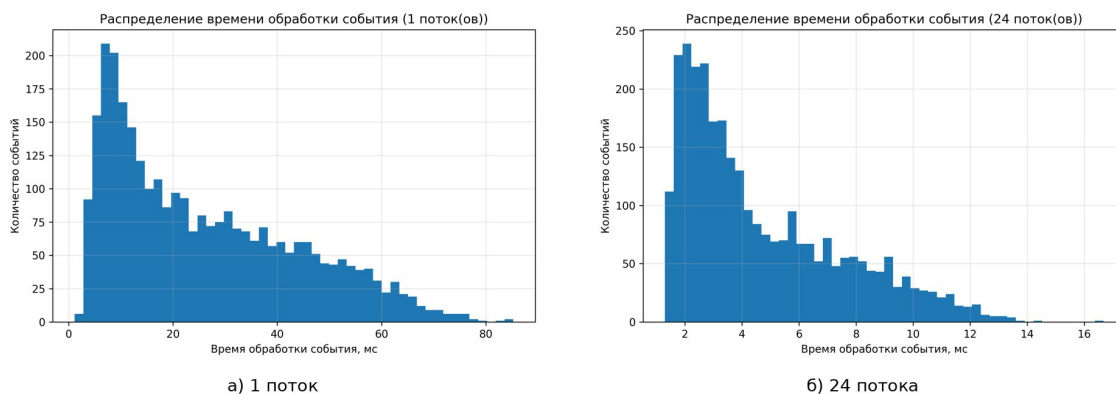


Рисунок 4. Распределение времени обработки события: а) 1 поток; б) 24 потока

Для количественной оценки были рассчитаны процентиля времени обработки. Результаты приведены в таблице 3. В качестве характеристики типичного события использовался 50-й процентиль, а для оценки редких тяжелых событий 99-й процентиль.

Таблица 3

Процентили времени обработки C++ алгоритма кластеризации

Число потоков	Среднее, мс	p50, мс	p99, мс
1	26,05	24,76	70,56
8	5,36	4,83	13,16
24	4,71	4,11	12,27
32	4,58	3,96	12,60

Для конфигурации с 24 потоками значение p50 составило 4,11 мс, а p99 – 12,27 мс.

3.4. Масштабирование при многопоточной обработке

Для оценки эффективности параллелизации было рассчитано ускорение:

$$S_N = \frac{T_1}{T_N}, \quad (7)$$

где T_1 – среднее время обработки в один поток, а T_N – среднее время обработки при использовании N потоков. Результат показан на рисунке 5.

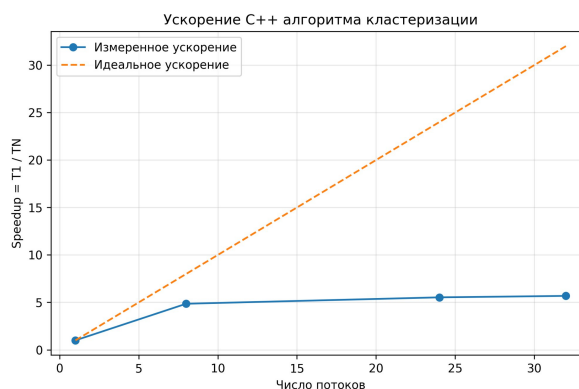


Рисунок 5. Ускорение C++ алгоритма кластеризации при увеличении числа потоков

При переходе от одного потока к 8 потокам ускорение достигает примерно 4,86. При использовании 24 потоков ускорение составляет около 5,53, а при 32 потоках – около 5,68. После 24 потоков наблюдается насыщение, поскольку основная единица параллелизации в данном тесте соответствует сектору ТРС, а полное событие содержит 24 секторных фрагмента.

3.5. Уменьшение объема данных

Одной из задач алгоритма является преобразование большого набора входных ADC-отсчетов в компактное представление события в виде восстановленных хитов. Для оценки этого эффекта был рассчитан коэффициент уменьшения объема данных как отношение размера входного представления к размеру выходного представления:

$$R = \frac{V_{input}}{V_{output}}, \quad (8)$$

где V_{input} – размер входного набора АЦП-отсчетов, а V_{output} – размер массива восстановленных хитов. Распределение коэффициента уменьшения объема данных и зависимость размера выходного представления от размера входного показаны на рисунке 6.

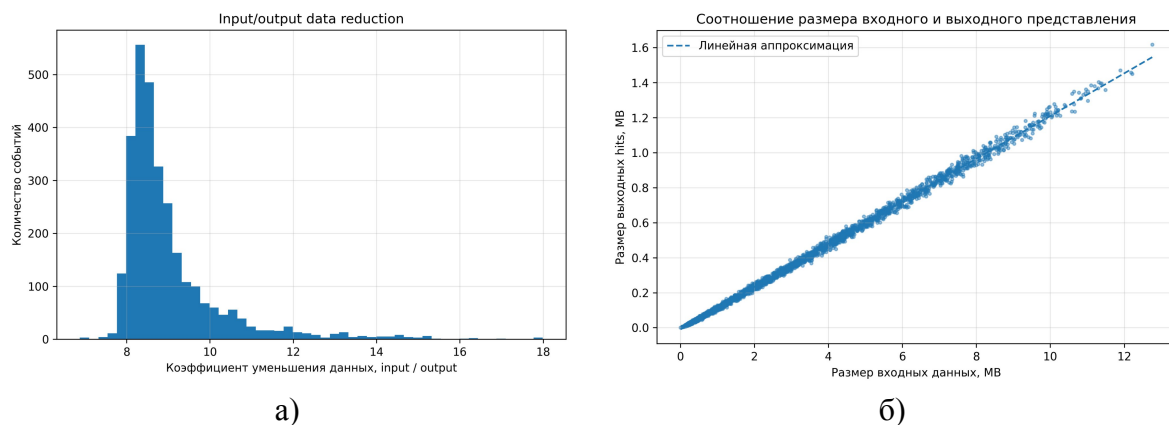


Рисунок 6. Уменьшение объема выходных данных:
а) коэффициент уменьшения; б) размер входного и выходного представления

Основная часть событий имеет коэффициент уменьшения порядка 8–10.

3.6. Оценка пространственного положения хитов относительно моделированных треков

Для проверки физической осмысленности восстановленных хитов было построено распределение расстояния от восстановленного хита до соответствующего МС-трека (моделированный методом Monte Carlo трек). Для сопоставления с МС-траекториями локальные координаты хита преобразовывались в глобальную систему координат ТРС с использованием геометрической модели детектора. После кластеризации для восстановленного хита выбирался связанный с ним МС-трек, а отклонение рассчитывалось как евклидово расстояние от точки хита в глобальных пространственных координатах до ближайшего отрезка траектории МС-трека в глобальной системе координат ТРС.

Для глобального хита H и отрезка МС-трека, заданного соседними точками P_i и P_{i+1} , вычислялось расстояние:

$$d(H, P_i, P_{i+1}) = |H - \text{proj}_{P_i P_{i+1}}(H)|. \quad (9)$$

Для каждого хита в глобальных координатах детектора выбиралось минимальное расстояние по всем отрезкам соответствующего МС-трека:

$$d_{\min} = \min_i d(H, P_i P_{i+1}). \quad (10)$$

Полное распределение расстояний показано на рисунке 9(а). Логарифмический масштаб по оси числа хитов позволяет одновременно отобразить центральную часть распределения и редкий дальний хвост. Красной пунктирной линией отмечено значение 95-го перцентиля, равное 1,607 см. На рисунке 7(б) показан тот же результат в линейном масштабе для области 0–10 см.

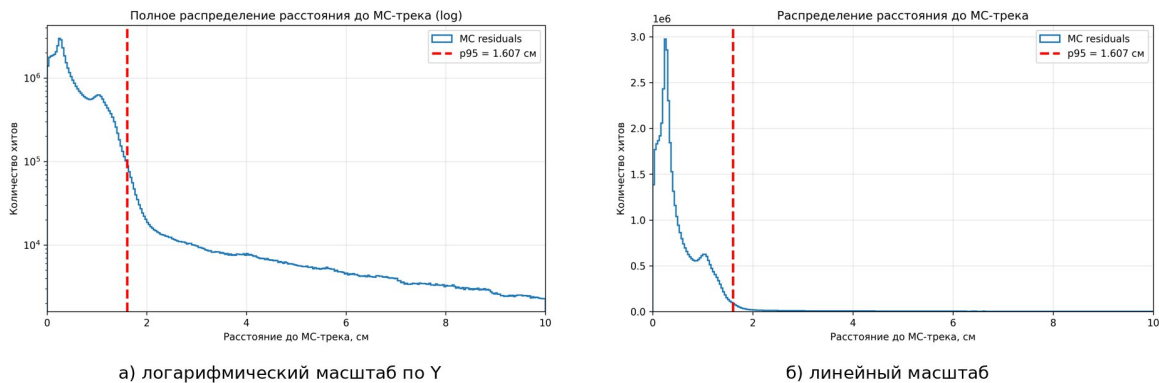


Рисунок 7. Распределение расстояния от восстановленных хитов до МС-треков:
а) логарифмический масштаб по оси Y; б) линейный масштаб

По распределению видно, что большая часть восстановленных хитов попадает в окрестность соответствующих МС-траекторий. Дальний хвост распределения может быть связан с неверными идентификаторами МС-треков вследствие их вторичной природы (шумовые частицы во время события), а также неоднозначным выбором идентификатора трека для составных кластеров. Детальное исследование причин появления хвоста распределения в данной работе не проводилось.

Заключение

В работе рассмотрен метод онлайн-кластеризации сигналов время-проекционной камеры, ориентированный на предварительную обработку данных в задачах визуализации событий и контроля качества. Метод преобразует набор АЦП-отсчетов в компактное представление в виде хитов в локальных координатах (сектор, пад, ряд, временной бакет), не выполняя полную реконструкцию траекторий и идентификацию частиц. За счет этого алгоритм может использоваться как быстрый этап подготовки данных для онлайн-сценариев.

Программная реализация выполнена на языке программирования C++ и учитывает естественную структуру данных ТРС: разделение события по секторам, рядам, пад-каналам и временным бакетам. Такая организация обеспечивает локальность вычислений и позволяет эффективно использовать параллельную обработку секторных фрагментов.

Экспериментальная оценка была проведена на 3000 моделированных событиях UrQMD класса minimum bias. В исследованном диапазоне время выполнения возрастало близко к линейному закону с увеличением числа входных АЦП-отсчетов.

При использовании 24 потоков среднее время обработки события составило 4,71 мс, значение p_{99} – 12,27 мс, а ускорение относительно однопоточного режима достигло 5,53. Дальнейшее увеличение числа потоков до 32 давало незначительный прирост, поскольку число независимых секторных задач для одного события ограничено числом секторов TPC.

После кластеризации размер выходного набора хитов для основной части событий был в 8–10 раз меньше размера входного представления. При сопоставлении восстановленных хитов с соответствующими MC-треками значение 95-го перцентиля расстояния составило 1,607 см.

Полученные результаты показывают, что предложенный подход можно использовать для дальнейшей интеграции в систему онлайн-обработки и мониторинга MPD/NICA. Проведенные измерения характеризуют только C++-алгоритм кластеризации и не включают рассмотрение времени на чтение исходных данных, сетевую передачу, сериализацию и доставку результатов конечному приложению. Оценка полной задержки онлайн-цепочки должна выполняться отдельно после интеграции алгоритма с остальными компонентами системы.

СПИСОК ИСТОЧНИКОВ

1. Kisiel A. The NICA Complex and the MPD experiment at JINR: status and physics potential / A. Kisiel // EPJ Web of Conferences. – 2022. – Vol. 259. – URL: <https://doi.org/10.1051/epjconf/202225909002> (дата обращения: 05.05.2026).
2. Microscopic models for ultrarelativistic heavy ion collisions / S.A. Bass, M. Belkacem, M. Bleicher [et al.] // Progress in Particle and Nuclear Physics. – 1998. – Vol. 41. – P. 255–369.
3. Relativistic hadron-hadron collisions in the ultra-relativistic quantum molecular dynamics model / M. Bleicher, E. Zabrodin, C. Spieles [et al.] // Journal of Physics G: Nuclear and Particle Physics. – 1999. – Vol. 25. – P. 1859–1896.
4. Brun R. ROOT – An object oriented data analysis framework / R. Brun, F. Rademakers // Nuclear Instruments and Methods in Physics Research. Section A. – 1997. – Vol. 389, No. 1–2. – P. 81–86.
5. SAMPA chip: a new ASIC for the ALICE TPC and MCH upgrades / S.H.I. Barboza, M. Bregant, V. Chambert [et al.] // Journal of Instrumentation. – 2016. – Vol. 11. – URL: <https://doi.org/10.1088/1748-0221/11/02/C02088> (дата обращения: 05.05.2026).
6. Time Projection Chamber for Multi-Purpose Detector at NICA: Technical Design Report / A. Averyanov, A. Bazhazhin, V.F. Chepurinov [et al.] // MPD [Электронный ресурс]. – URL: <https://mpd.jinr.ru/wp-content/uploads/2023/11/TpcTdr-v07.pdf> (дата обращения: 05.05.2026).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Крылов Александр Викторович, младший научный сотрудник, Объединенный институт ядерных исследований, Дубна, Россия.

e-mail: avkrylov@jinr.ru

Крылов Виктор Александрович, ведущий инженер, Объединенный институт ядерных исследований, Дубна, Россия.

e-mail: kryman@jinr.ru

Будников Андрей Сергеевич, магистр, Воронежский государственный университет, Воронеж, Россия.

e-mail: budnikov@jinr.ru