

P10-2023-14

А. С. Кирилов, И. А. Морковников, С. М. Мурашкевич,
Т. Б. Петухова, Л. А. Трунтова

РЕОРГАНИЗАЦИЯ ПРОГРАММНОГО
КОМПЛЕКСА **SONIX+** ДЛЯ РАБОТЫ
С ДАННЫМИ В ВИДЕ СПИСКА СОБЫТИЙ

Направлено в журнал «Письма в ЭЧАЯ»

Реорганизация программного комплекса Sonix+ для работы с данными в виде списка событий

Применение на измерительных установках реактора ИБР-2 новых DAQ-контроллеров, в которых отсутствует аппаратное гистограммирование данных, в ближайшие годы примет все более массовый характер. При этом происходит замена первичной формы представления данных в виде гистограмм в пользу накопления в виде списка отдельных событий, что соответствует общемировой тенденции. Это повлечет за собой определенную перестройку комплекса Sonix+. При этом, с одной стороны, гистограммы следует сохранить для оперативной визуализации и для программ обработки, которые специфичны для каждой установки. С другой стороны, снимается аппаратное ограничение по дискретизации при построении этих гистограмм, что, как ожидается, на практике приведет к существенному увеличению размеров гистограмм, в том числе превышающему лимит доступного виртуального адресного пространства для 32-разрядных приложений.

В работе предложен подход к разрешению этой проблемы.

Работа выполнена в Лаборатории нейтронной физики им. И. М. Франка ОИЯИ.

Reorganization of the Sonix+ Software Package to Work with Data in the Event List Form

In the coming years the application of new DAQ controllers in which there is no hardware data histogramming will become more and more widespread at the IBR-2 instruments. At the same time, the primary data presentation in the histogram form is replaced in favor of accumulation in the form of a list of individual events, which corresponds to the global trend. This will entail a certain restructuring of the Sonix+ complex. Thus, on the one hand, histograms should be kept for operative visualization, as well as for data reduction programs, which are specific for each instrument. On the other hand, the hardware sampling limitation in the construction of these histograms is removed, which is expected to lead to a significant increase in histogram size, including exceeding the limit of available virtual address space for 32-bit applications in practice.

In this work we propose an approach to solve this problem.

The investigation has been performed at the Frank Laboratory of Neutron Physics, JINR.

ВВЕДЕНИЕ

В последние годы в нейтронных центрах растет интерес к использованию DAQ-контроллеров, сохраняющих информацию в виде списка событий (list-mode, event-mode).

Так, на последнем рабочем совещании NOBUGS22 (октябрь 2022 г.), которое специализируется на вопросах создания программного обеспечения для нейтронных, синхротронных и мюонных центров, эта тема была затронута в докладах участников из PSI (Виллиген, Швейцария), ANL (Аргонн, США), ESS (Лунд, Швеция), IHEP (Пекин) и ряда других центров [1].

Ранее традиционной формой предоставления результатов измерения пользователям были массивы со спектральным распределением интенсивностей (гистограммы), причем сам процесс гистограммирования в основном выполнялся аппаратно с заранее заданными параметрами по временному и пространственному разрешениям. При этом происходило определенное статистическое усреднение исходных данных, но результат был предельно компактен. Напротив, исходные данные в формате event-mode содержат максимально детализированную информацию об измерении, которую можно сортировать и гистограммировать с произвольной выборкой и разрешением.

Однако сравнительно с гистограммами эти данные существенно больше по объему, программное гистограммирование требует больших ресурсов: оперативной памяти, процессорного времени компьютера и дискового пространства. Как показывает практика, обработка массивов большого объема может потребовать перевода 32-разрядных приложений (которые далее для краткости и наглядности будем обозначать $\times 32$) в 64-разрядные (далее — $\times 64$), что в общем случае не является тривиальной задачей. Кроме того, передача больших объемов данных требует значительного увеличения пропускной способности локальной сети. Таким образом, замена первичной формы представления измеренных данных «гистограммной» на файлы со списком событий приводит к существенной перестройке всего конвейера данных измерений от собственно систем управления установками до программ обработки данных, а также всей сетевой инфраструктуры, включая средства передачи и хранения информации.

В системах управления установками реактора ИБР-2 успешно используется программный инструментальный комплекс Sopix+ [2]. Комплекс постоянно развивается в соответствии с запросами пользователей, учетом накопленного опыта и мировыми тенденциями в этой области.

Рассмотрим необходимые изменения в комплексе, которые позволяют организовать работу с данными в форме списков событий.

АНАЛИЗ ТЕКУЩЕЙ СИТУАЦИИ

В ЛНФ ОИЯИ переход на новые DAQ-контроллеры [3–6] на установках ИБР-2 уже начался и в ближайшие годы примет все более массовый характер. На ряде установок (ФДВР, ФСД, РТД) запись данных в форме списка событий применялась достаточно давно. Однако поддержка со стороны Sopix+ ограничивалась только записью данных. И хотя для контроллера DeLiDAQ-2 гистограммирование приходилось выполнять программно, тем не менее общий подход сохранялся. При этом максимальный размер запрашиваемых гистограмм был много меньше лимита виртуального адресного пространства $\times 32$ (см. таблицу).

Ситуация изменилась при попытках использования диджитайзера N6730 фирмы CAEN [6, 7]. Получив возможность задавать произвольное разрешение, пользователи столкнулись с ошибками при попытках визуализации гистограмм из-за превышения лимита виртуального адресного пространства программ.

Рассмотрим два графика (рис. 1, 2), построенных с разным разрешением программой SpectraViewer на основе данных измерения прямого пучка, выполненного на 2D ПЧД-детекторе на установке «Грейнс» в сентябре 2021 г. На них изображена центральная часть карты плотности (X , ToF), полученная при суммировании данных по оси Y (X, Y — простран-

Размерность гистограмм для данных с двухкоординатных позиционно-чувствительных детекторов (2D ПЧД), использованных на установках реактора ИБР-2 в мае 2021 г.

Установка	Тип DAQ-контроллера	Ширина канала, мкс	ToF	X	Y	Длина гистограммы, Мбайт
Грейнс	DeLiDAQ-2	128	1024	128	256	128
ФДВР	DeLiDAQ-2	64	1024	128	128	64
Рефлекс	DeLiDAQ-1	192	668	160	80	32,6
Ремур	DeLiDAQ-2	256	512	256	128	64
РТД	DeLiDAQ-2	64	204	32	64	16
	DeLiDAQ-1	64	2048	16	32	4
ЮМО	—	128	512	256	256	128

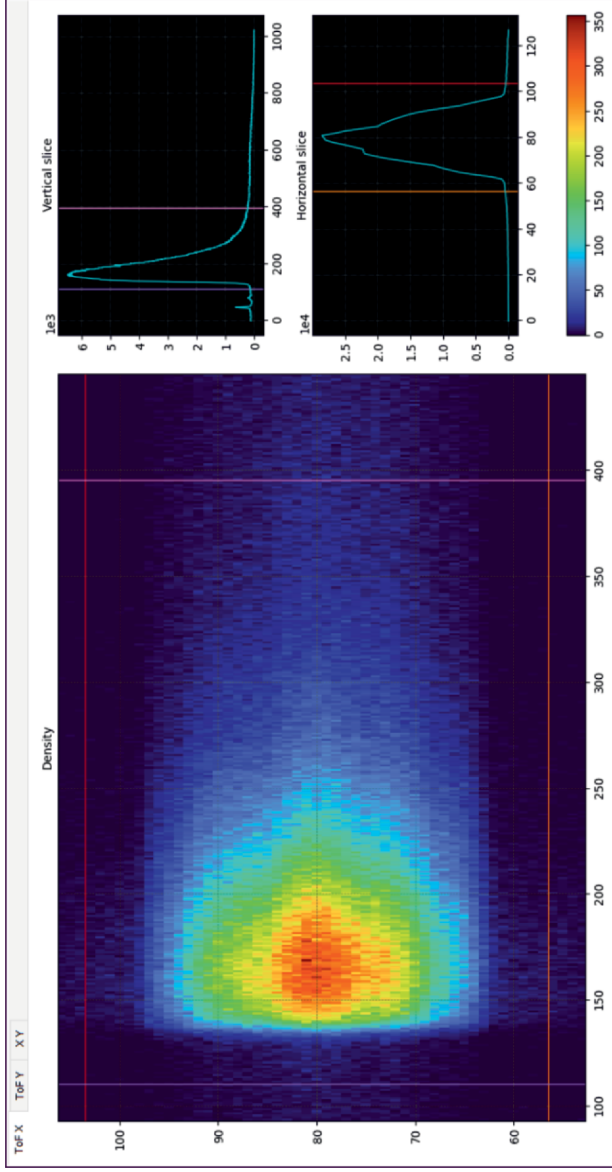


Рис. 1. Графики прямого пучка установки «Грейнс» (10-й канал), построенные с разрешением $ToF = 1024$, $X = 128$, $Y = 256$. Размер гистограммы — 128 Мбайт. Все поверхности и кривые выглядят гладкими

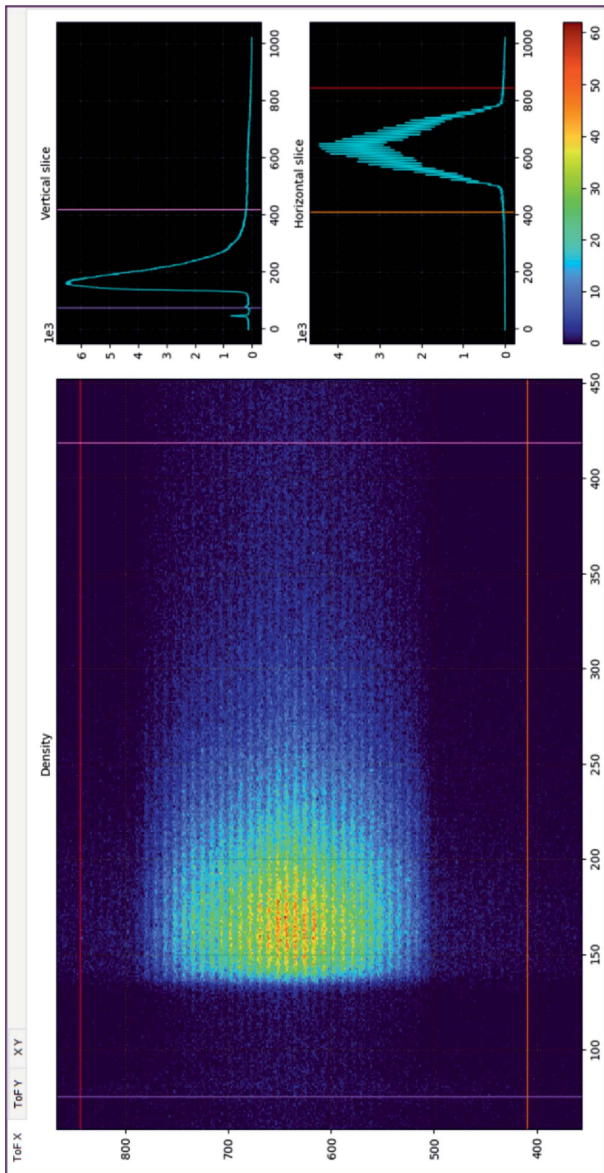


Рис. 2. Графики для того же измерения, построенные с разрешением $\text{ToF} = 1024$, $X = 1024$, $Y = 1000$. Размер гистограммы — 4000 Мбайт. Явственно видна структура регистрационных элементов детектора

ственные каналы регистрации; ToF — временные каналы регистрации, заполняемые по методу времени пролета). Одномерные графики представляют результат суммирования данных по осям ToF и Y (справа сверху), и аналогично — результат суммирования по осям ToF и X (справа внизу). Первая гистограмма размером 128 Мбайт построена с помощью 32-разрядной версии Python (далее Python32), вторая — размером 4000 Мбайт с помощью 64-разрядной версии Python (далее Python64). На первом графике все кривые и поверхности гладкие, на втором явно видна структура регистрационных элементов детектора.

Консультанты с пользователями подтвердили интерес к гистограммам с большим разрешением, тем более что при их обработке последнее можно уменьшить суммированием значений в соседних каналах.

Несмотря на переход к данным в форме списка событий, полный отказ от гистограмм в комплексе Sonix+ невозможен. С одной стороны, гистограммы следует сохранить для оперативной визуализации, с другой — для существующих программ обработки, ориентированных на входные данные в этой форме.

СТРУКТУРА DAQ Sonix+ И «УЗКИЕ МЕСТА» В НЕЙ

Замена первичной информации в форме гистограмм на форму списка событий повлечет за собой определенную перестройку комплекса. Sonix+ реализован как набор модулей $\times 32$, и, следовательно, все его компоненты располагают виртуальным адресным пространством 2 ГБ для программных кодов и данных.

Последовательность обработки данных в Sonix+ приведена на рис. 3. Данные от контроллеров запрашиваются драйверами и далее передаются на следующий уровень. При необходимости драйверы формируют файлы со списком событий.

В конце экспозиции сервер S_expo запрашивает данные в виде гистограмм от всех драйверов и формирует общий файл в формате Sonix+,

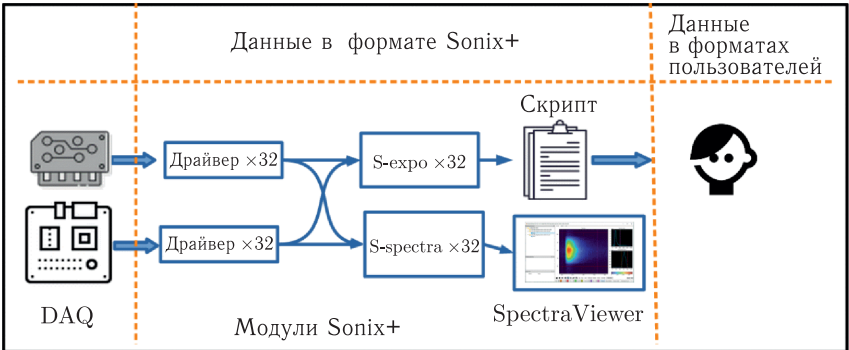


Рис. 3. Структура конвейера обработки данных в Sonix+

который далее в скрипте трансформируется в набор файлов конкретного спектрометра.

Сервер `S_spectra` выполняет аналогичную работу по запросу программы оперативной визуализации `SpectraViewer`.

Все модули нижнего уровня (драйверы и серверы) написаны на языке C++ и оформлены как приложения $\times 32$.

Программа визуализации данных `SpectraViewer` и остальной пользовательский интерфейс (GUI) написаны на языке Python и запускаются через его 32-разрядный интерпретатор.

В комплексе можно отметить три «узких» места при работе с «большими» массивами:

- ограниченность размера внутренних буферов, прежде всего, в серверах `S_expro` и `S_spectra`;
- аналогичная ситуация с программой `SpectraViewer`;
- недостаточность выбранной размерности длины гистограммы в 4 байта в формате `Sonix+`.

ВОЗМОЖНЫЕ ПОДХОДЫ И ОСНОВНАЯ ИДЕЯ РЕШЕНИЯ

В связи с тем, что на управляющих компьютерах установлена 64-разрядная система, а поддержка подсистемы `WoW64` (`Windows-on-Windows 64-bit`), позволяющей запускать 32-разрядные приложения, не лимитирована, ограничение размеров массивов данных является фактически единственным, которое необходимо устранить.

Лобовое решение проблемы по полному переводу комплекса в $\times 64$ сопряжено с трудностями перевода ядра комплекса на C++ и не выглядит рациональным решением [8]. Тем более, что значительная часть компонентов комплекса, написанная на языке Python (GUI, визуализация, системные скрипты `Sonix+`), может работать под `Python64` и такого перевода не требует.

Вместе с тем очевидно, что, поскольку при работе с файлами событий гистограммирование выполняется программно после измерения, можно решить данную проблему, выделив перевод данных в формат гистограмм, а также их визуализацию в $\times 64$. При этом управляющая часть комплекса на C++ с небольшой модификацией сохраняется в $\times 32$.

Основной сложностью в подобном подходе является разработка нового средства взаимодействия между модулями, написанными на Python, и модулями, написанными на C++, поскольку коммуникационная библиотека динамической компоновки, которая обеспечивала эту связь, работает только в $\times 32$ и не импортируется в `Python64`.

Реорганизация комплекса выглядит следующим образом. Детекторы (`DAQ-контроллеры`) в режиме `event-mode` выделяются в отдельный класс. Данные от этих детекторов записываются в особом формате (рис. 4).

Здесь и далее серым выделены поля, которых не было ранее. Список параметров для гистограммирования зависит от типа контроллера и включает все необходимые параметры помимо разрешения.

Тип контроллера	Имя детектора	Список параметров для гистограммирования	Данные
-----------------	---------------	--	--------

Рис. 4. Формат записи данных в event-mode

Для гистограммирования данных каждого контроллера разрабатывается отдельная программа-конвертер как приложение $\times 64$. Размерность гистограмм и имена входного и выходного файлов задаются через строку параметров конвертера. Конвертеры могут вызываться как автоматически из скрипта в процессе проведения эксперимента, так и вручную из специализированного GUI по запросу пользователя.

Каждая из конвертированных гистограмм сохраняется в отдельном файле по формату на рис. 5.

Признак формата	Имя детектора	Число детекторов	Размерность (ToF, X.Y)	Длина (8 байтов)	Гистограмма
-----------------	---------------	------------------	------------------------	------------------	-------------

Рис. 5. Формат записи конвертированной гистограммы

Необходимые изменения для работы с новым классом детекторов вносятся в программы драйверов, серверов и SpectraViewer. При этом обработка данных с остальных детекторов (контроллеров) сохраняется неизменной.

Для связи ядра Sonix+ и модулей на языке Python предложено воспользоваться программой PyChannel [9].

ОПЕРАТИВНАЯ ВИЗУАЛИЗАЦИЯ

Схема оперативной визуализации не претерпела существенных изменений. Пользователь программы SpectraViewer [10] посылает запрос модулю S+spectra с именем интересующего детектора. По этому запросу S_spectra запрашивает данные от драйвера и с помощью соответствующего конвертера формирует файл данных, который передается для визуализации. Единственным существенным изменением стало использование программы PyChannel для передачи сообщений от SpectraViewer модулю S+spectra и обратно.

GUI ДЛЯ РУЧНОГО КОНВЕРТИРОВАНИЯ

Для ручного конвертирования на языке Python разработана программа EventModeToHistogram, интерфейс которой приведен на рис. 6.

Программа позволяет обрабатывать сразу несколько файлов. Размерность гистограммы задается пользователем, а значения специфических параметров извлекаются из файла данных. После запуска программа проверяет корректность заданных параметров и в случае успеха выполняет

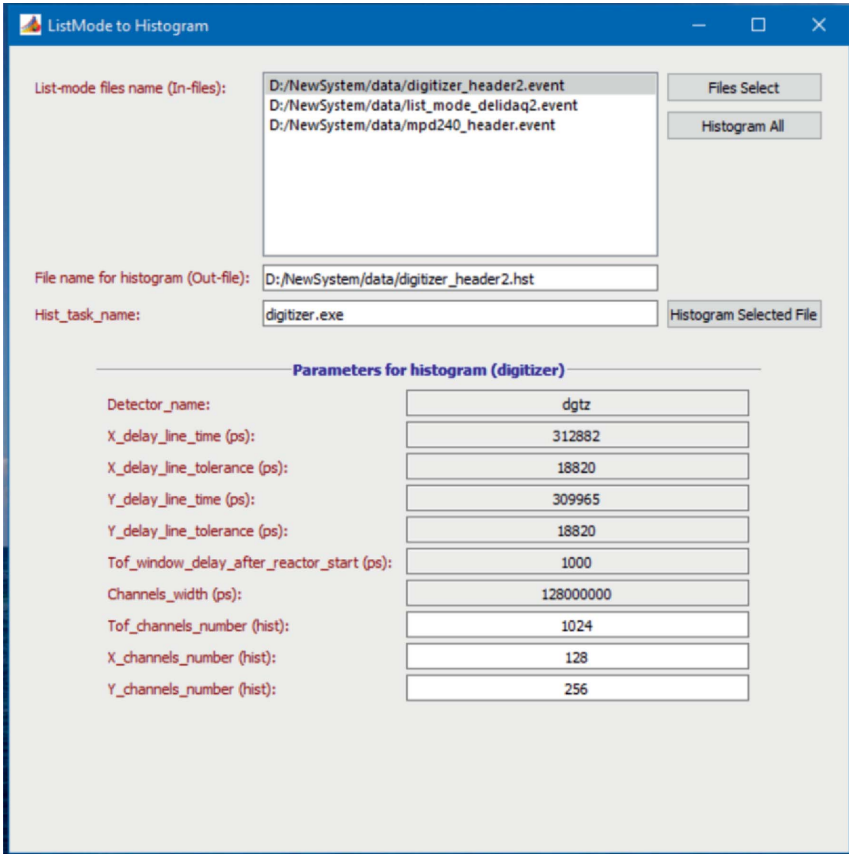


Рис. 6. Интерфейс конвертера файлов

гистограммирование. Проверка нужна, так как существуют ограничения на выбор размерности гистограмм.

Список конвертеров вместе с необходимым для каждого составом параметров вынесен в конфигурационный файл задачи, чем обеспечивается ее универсальность.

ЗАКЛЮЧЕНИЕ

Перевод модулей, задействованных для работы с массивами измеренных данных, в 64-разрядную архитектуру и сохранение остальных модулей в 32-разрядной архитектуре позволили оптимальным способом адаптировать программный комплекс Sonix+ к работе с данными в форме списка событий. Самым сложным моментом в этом решении была разработка программы RuChannel для связи приложений в архитектурах разных разряд-

ностей. Применение языка Python для реализации GUI и ряда системных скриптов существенно упростило решение общей задачи.

На момент написания этой статьи основные работы были выполнены и опробованы на реальных экспериментальных данных.

Следует отметить, что для установок реактора ИБР-2 перевод в формат первичных данных в форме списка событий предполагается производить постепенно по мере замены контроллеров старого типа, подключения новых детекторов и создания новых установок.

Авторы выражают признательность В. И. Боднарчуку за поддержку работы и ценные замечания при подготовке данной рукописи.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (соглашение № 075-10-2021-115 от 13 октября 2021 г. (внутренний номер 15.СИН.21.0021)).

ЛИТЕРАТУРА

1. <https://indico.psi.ch/event/12738/book-of-abstracts.pdf>
2. <https://sonix.jinr.ru/wiki/doku.php?id=ru:index>
3. *Levchanovskiy F. V., Murashkevich S. M.* // Proc. of the XXIV Intern. Symp. on Nuclear Electronics and Computing (NEC'2013), Varna, Bulgaria, Sept. 9–16, 2013, P. 176–179; Dubna: JINR, 2013.
4. *Levchanovskiy F. V., Murashkevich S. M.* // Phys. Part. Nucl. Lett. 2016. V. 13, No. 5. P. 591–594.
5. *Кирилов А. С., Мурашкевич С. М.* Адаптация программного комплекса Sonix+ для работы с DAQ контроллерами DeLiDAQ-2 и диджитайзером N6730 фирмы CAEN. Препринт ОИЯИ P10-2022-5. Дубна, 2022.
6. *Литвиненко Е. И. и др.* Сравнительный анализ характеристик систем сбора данных с позиционно-чувствительных детекторов нейтронов // ПТЭ. 2020. № 3. С. 56–64.
7. <https://www.caen.it/products/n6730/>
8. <https://pvs-studio.com/ru/blog/posts/cpp/a0042/>
9. https://sonix.jinr.ru/wiki/doku.php?id=ru:websonix:py_channel_article
10. *Кирилов А. С.* Новые версии программ юстировки и визуализации спектров для рефлектометров ИБР-2 // Письма в ЭЧАЯ. 2016. Т. 13, № 1(199). С. 208–221.

Получено 27 апреля 2023 г.

Редактор *М. И. Зарубина*

Подписано в печать 19.05.2023.

Формат 60 × 90/16. Бумага офсетная. Печать цифровая.

Усл. печ. л. 1,00. Уч.-изд. л. 0,67. Тираж 125 экз. Заказ № 60661.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: publish@jinr.ru

www.jinr.ru/publish/