

*На правах рукописи*



Ремнев Михаил Анатольевич

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ДЛЯ СИСТЕМЫ СБОРА ДАННЫХ  
ЭЛЕКТРОМАГНИТНОГО КАЛОРИМЕТРА  
ДЕТЕКТОРА BELLE II**

1.3.2. Приборы и методы экспериментальной физики

**АВТОРЕФЕРАТ**  
диссертации на соискание учёной степени  
кандидата технических наук

Новосибирск — 2023

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте ядерной физики им. Г.И. Будкера Сибирского отделения Российской академии наук.

**НАУЧНЫЙ РУКОВОДИТЕЛЬ** доктор физико-математических наук  
**Кузьмин Александр Степанович**

#### ОФИЦИАЛЬНЫЕ ОППОНЕНТЫ

**Куденко Юрий Григорьевич** доктор физико-математических наук, профессор, Федеральное государственное бюджетное учреждение науки Институт ядерных исследований Российской академии наук, главный научный сотрудник

**Прокошин Федор Валерьевич** кандидат физико-математических наук, Международная межправительственная организация Объединенный институт ядерных исследований, старший научный сотрудник

**ВЕДУЩАЯ ОРГАНИЗАЦИЯ** Федеральное государственное бюджетное учреждение науки Физический институт им. П.Н. Лебедева Российской академии наук

Защита диссертации состоится «19» декабря 2023 года в «14:00» часов на заседании диссертационного совета 24.1.162.01 на базе Федерального государственного бюджетного учреждения науки Института ядерной физики им. Г.И. Будкера Сибирского отделения Российской академии наук по адресу: 630090, г. Новосибирск, проспект Академика Лаврентьева, 11.

С диссертацией можно ознакомиться в библиотеке Федерального государственного бюджетного учреждения науки Института ядерной физики им. Г.И. Будкера Сибирского отделения Российской академии наук и на сайте <https://inp.nsk.su/obrazovanie/dissertatsionnye-sovety#24-1-162-01>.

Автореферат разослан «08» ноября 2023 года.

Ученый секретарь  
диссертационного совета 24.1.162.01  
доктор физико-математических наук

  
Шехтман Лев Исаевич

# ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

## **Актуальность темы исследования**

Эксперимент Belle II начался в 2018 году. Цель эксперимента — набрать  $50 \text{ абн}^{-1}$  данных для исследования физики  $B$ - и  $D$ -мезонов,  $\tau$ -лептонов, нарушения  $CP$ -симметрии, определения параметров СКМ-матрицы и поиска Новой физики. В 2019 году эксперимент перешёл к третьей фазе (Phase III) — регистрации событий в столкновении  $e^+e^-$  пучка с использованием всех подсистем детектора. В этой фазе важно иметь готовую систему сбора данных (DAQ, Data Acquisition), которая позволит с минимальными задержками проводить чтение данных со всех систем детектора с частотой до 30 кГц, осуществлять запуск заходов, проводить мониторинг и калибровку систем детектора.

Электромагнитный калориметр (ECL, Electromagnetic Calorimeter), на котором фокусируется данная работа, является важной системой детектора Belle II, предназначенной для измерения энергии зарегистрированных частиц. Считывающая электроника ECL была усовершенствована для работы в условиях увеличенного уровня фона, поэтому возникла необходимость разработки нового программного обеспечения, которое будет осуществлять инициализацию и управление электроникой ECL DAQ. Кроме того, поскольку эксперимент Belle II перешёл к другому фреймворку медленного контроля и отказался от большинства программных средств, использовавшихся в эксперименте Belle, в пользу новых решений, возникла необходимость разработать заново или существенно переписать многое высокоуровневое программное обеспечение (ПО), используемое в ECL DAQ. Разработанное ПО для ECL DAQ позволяет обеспечивать стабильную работу калориметра в ходе эксперимента, уменьшая время, необходимое дежурному на обнаружение и устранение потенциальных проблем, влияющих на качество записываемых данных.

## **Степень разработанности темы**

Можно выделить три класса задач, которые решает программное обеспечение для системы сбора данных. Технические задачи относятся к проблемам быстродействия и масштабируемости системы на всех уровнях её работы. Задачи управления включают в себя управление конфигурациями системы, осуществление мониторинга качества данных и оперативное обнаружение сбоев DAQ. Интеграционные задачи подразумевают определение протоколов, использующихся для объединения отдельных узлов системы.

Одной из важных технических задач является разработка ПО для триггера высокого уровня. По сравнению с системой Belle DAQ, частота триггера первого уровня в системе Belle II DAQ выросла от 0,5 кГц

до 30 кГц. По этой причине было решено использовать программный триггер высокого уровня для реконструкции и дополнительного отбора событий. Один из первых прототипов триггера высокого уровня был разработан в 2003 году для детектора ATLAS. На данный момент существует несколько реализаций триггера высокого уровня, однако только в Belle II используется единый фреймворк для обработки данных, который применяется как в триггере высокого уровня, так и при последующей детальной обработке сохранённых данных. Это позволяет лучше организовать процесс разработки, но приводит к дополнительным техническим требованиям на быстрдействие программного обеспечения и более широкую оптимизацию под набор различных сценариев использования.

Для решения задач управления в Belle II в большинстве случаев используется фреймворк Control System Studio (CSS). С 2013 года CSS активно применяется в физике высоких энергий, представляя единый набор инструментов для мониторинга и управления детектором. Однако, как и для любых других фреймворков, ориентированных на построение графических интерфейсов пользователя, для CSS затруднена реализация процедур автоматического тестирования и интеграции с веб-интерфейсами, что является важной задачей в эксперименте Belle II.

В рамках задач интеграции, распространённым решением для объединения приложений в сети сбора данных является система медленного контроля Experimental Physics and Industrial Control System (EPICS), широко используемая в физических экспериментах и постоянно развивающаяся с 1994 года. Модули, разработанные при помощи EPICS, способны обмениваться информацией, существуют готовые решения для архивирования данных мониторинга и для реализации систем оповещения о сбоях в системе сбора данных. Тем не менее в EPICS невозможно получить полный список запущенных модулей, что может затруднять администрирование системы. Поэтому для эксперимента Belle II используется новый фреймворк Network Shared Memory 2, который, в рамках данной работы, было необходимо расширить для решения существующих интеграционных задач.

### **Цели и задачи диссертационной работы**

Целью данной работы является разработка программного обеспечения для системы сбора данных электромагнитного калориметра, в частности систем медленного контроля и мониторинга.

Задачи данной работы:

1. Разработка ПО для взаимодействия с электроникой калориметра, автоматизирующего следующие процессы:
  - Управление конфигурациями электроники.

- Инициализация электроники и её диагностика.
2. Разработка модулей системы медленного контроля.
  3. Создание системы мониторинга качества данных ECL в режиме реального времени.
  4. Разработка системы DAQ для монитора светимости, являющегося отдельным модулем ECL.
  5. Разработка инструментов для более детальной диагностики качества данных ECL по сохранённой информации.

### **Научная новизна**

Детектор Belle II является уникальной установкой, нацеленной на проведение передовых исследований в области физики высоких энергий. Автором создано уникальное программное обеспечение, позволяющее инициализировать считывающую электронику, эффективно организовывать процесс сбора данных с электромагнитного калориметра, автоматически обнаруживать и исправлять возможные сбои на всех уровнях работы системы, анализировать долговременную стабильность электроники сбора данных.

Ряд новых программных решений, которые было решено использовать в эксперименте, требует пересмотра всего существующего стека технологий, использующихся в DAQ. Система медленного контроля эксперимента Belle II использует новый фреймворк Network Shared Memory 2 (NSM2), разработанный в КЕК. Кроме того, в эксперименте используются две базы данных (БД), БД калибровочных констант и БД конфигураций электроники. Они имеют существенно различающуюся внутреннюю структуру, поэтому их синхронизация, необходимая при разработке программного обеспечения для калориметра, является нетривиальной задачей. Для учёта этих требований были разработаны программные решения и модули, которые лучше сочетаются с новыми парадигмами построения ПО сбора данных. Данная работа рассматривает современные методики программирования в применении к задачам автоматизации физического эксперимента.

Впервые был рассмотрен процесс разработки систем медленного контроля и мониторинга в контексте методов инкрементного прототипирования. В частности, разработан модуль для экспорта библиотеки C++ в специально разработанный интерпретатор, что позволяет легко интегрироваться с другими запущенными процессами системы сбора данных.

В рамках данной работы автором применены методики быстрого прототипирования приложений в контексте разработки для фреймворка NSM2, реализованного на языке C. Чтобы ускорить процесс разработки, были рассмотрены различные методы реализации интерфейса Python к

NSM2, представлены преимущества и недостатки возможных решений. Разработана библиотека ruNSM2, использующая программный модуль `stures` для быстрого доступа к функциям и структурам языка C. Разработаны инструменты, поддерживающие реализацию ruNSM2 в полном соответствии с основной реализацией NSM2.

Впервые разработана система сборки, позволяющая автоматически интегрировать независимо разрабатываемые на разных языках программирования библиотеки NSM2 и ruNSM2.

### **Теоретическая и практическая значимость**

Разработанное программное обеспечение внедрено в систему сбора данных эксперимента Belle II и обеспечивает стабильный сбор и контроль данных с электроники электромагнитного калориметра и монитора светимости. Таким образом, созданное ПО позволяет считывать критически необходимые в физических исследованиях эксперимента Belle II данные, а также осуществлять мониторинг работы коллайдера SuperKEKB. Созданное ПО является одним из необходимых компонентов для проведения эксперимента Belle II. Суммарный объём конечного кода составляет около 45000 строк.

Использование данной работы в эксперименте Belle II подтверждено актом о внедрении.

Разработанные библиотеки могут использоваться в других системах сбора данных, позволяя с минимальными временными затратами реализовывать системы мониторинга и медленного контроля.

### **Методология и методы исследования**

В работе использованы методологии Rapid Application Development и Test-Driven Development. Программное обеспечение разрабатывалось на языках программирования C, C++, Python, Jython, PHP, JavaScript. Для обработки данных использовался фреймворк CERN ROOT и Belle II Analysis Software Framework. Для интеграции процессов медленного контроля, относящихся к ECL, с глобальной системой сбора данных использовалась разработанная для эксперимента Belle II программная библиотека `daq_slc`. В ПО системы сбора данных широко использовались программные пакеты EPICS, NSM2, Control System Studio, Qt 5, система управления базами данных SQLite.

### **Основные положения, выносимые на защиту**

1. Разработана система инициализации и управления конфигурациями электроники калориметра.
2. Новый программный модуль Python предоставляет доступ к функциям системы медленного контроля NSM2 и широко используется

- в эксперименте Belle II, существенно ускоряя разработку приложений для системы сбора данных.
3. Создан веб-сервер ECL DAQ, который обеспечивает стабильную работу считывающей электроники калориметра, предоставляя централизованный доступ ко всем реализованным инструментам мониторинга качества данных.
  4. Разработана независимая система сбора данных с онлайн-мониторинга светимости, сопутствующие модули мониторинга и калибровки.

### **Степень достоверности и апробация результатов**

Разработанное программное обеспечение используется в эксперименте Belle II и позволяет обеспечивать стабильный сбор данных с электроники электромагнитного калориметра. Результаты данной работы были представлены на внутренних семинарах Belle II и конкурсах молодых учёных ИЯФ СО РАН, а также на следующих международных конференциях: 55-ая международная студенческая конференция (16–20 апреля 2017 г., Новосибирск, Россия), Instrumentation for Colliding Beam Physics (INSTR-20, 24–28 February 2020, BINP, Novosibirsk, Russia). Разработанное программное обеспечение используется в эксперименте Belle II и позволяет обеспечивать стабильный сбор данных с электроники электромагнитного калориметра.

### **Публикации**

Основные результаты диссертации представлены в пяти публикациях, из них пять в научных изданиях, рекомендуемых ВАК при Минобрнауки России [1–5].

### **Личный вклад соискателя**

Изложенные в работе результаты получены автором лично либо при его определяющем вкладе. Автор принимал активное участие в проведении экспериментальных исследований и организации работы дежурных экспертов по ECL.

Автором было разработано программное обеспечение для работы с электроникой калориметра, синхронизирующее параметры электроники с базами данных, использующимися в эксперименте Belle II. Это позволило, с одной стороны, использовать актуальные параметры электроники в моделировании и, с другой стороны, быстро обновлять амплитудные пороги, используя новые калибровочные коэффициенты. В рамках данной работы автор также расширил систему мониторинга качества данных, разработал программное обеспечение для инициализации электроники и управления системой сбора данных. Автором был реализован ряд модулей для быстрой интеграции с системой медленного контроля, которые используются в глобальной системе сбора данных эксперимента

Belle II. Также была разработана независимая система сбора данных с онлайн-монитора светимости.

Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы. Подготовка полученных результатов к публикации была проведена совместно с соавторами. Все представленные в диссертации результаты получены лично автором, либо при его участии.

### **Структура и объём диссертации**

Данная работа разделена на введение, 8 глав и заключение. Полный объём диссертации составляет **105** страниц, включая **42** рисунка и **1** таблицу. Список литературы содержит **117** наименований.

## **СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** описана актуальность темы, определены цель и задачи исследования, приведена новизна и значимость работы, а также перечислены положения, выносимые на защиту.

**В первой главе** приводится общая информация об эксперименте Belle II и детальная информация об электромагнитном калориметре и системе сбора данных, использующейся в эксперименте.

*В первом разделе* кратко описывается коллайдер SuperKEKB. Повышение светимости в SuperKEKB достигается за счёт использования схемы нанопучков и, поскольку рост светимости ведёт к росту фона, возникает необходимость модификации всех подсистем детектора Belle II.

*Во втором разделе* кратко описывается детектор Belle II. Модификация детектора повлекла за собой необходимость модификации системы сбора данных для всех его подсистем.

*В третьем разделе* описывается электромагнитный калориметр (ECL) детектора Belle II, считывающая электроника которого была модифицирована с учётом повышения фона. ECL служит для измерения энергии и координат фотонов и электронов в диапазоне энергий от 20 МэВ до 4 ГэВ. Как показано на рисунке 1, ECL состоит из 8736 кристаллов CsI(Tl), имеющих форму усечённой пирамиды, вершина которой направлена к окрестности точки взаимодействия. Калориметр покрывает телесный угол  $0,91 \cdot 4\pi$ .

Система сбора данных (DAQ, Data Acquisition) ECL изображена на рисунке 2. Сцинтилляционный свет с кристаллов CsI(Tl) регистрируется двумя PIN фотодиодами Hamamatsu S2744-08. Формирование, оцифровка и подгонка сигналов со счётчиков проводится 576 модулями ShaperDSP и 52 модулями ECLCollector. В считывающей электронике задаются амплитудные пороги, параметры подгонки, параметры

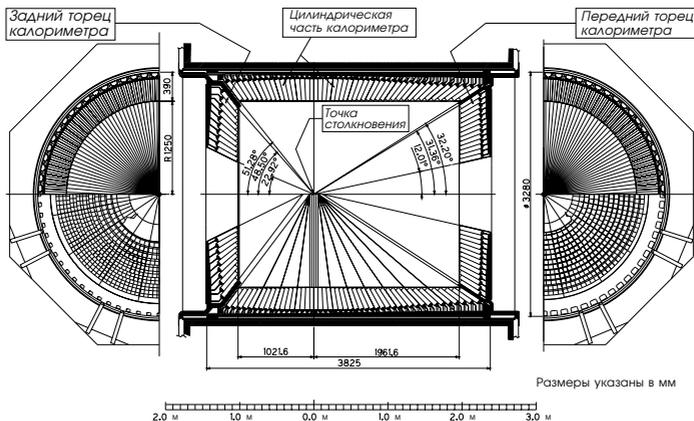


Рисунок 1 — Схема электромагнитного калориметра Belle II

сохранения данных, калибровочные коэффициенты для триггерного канала.

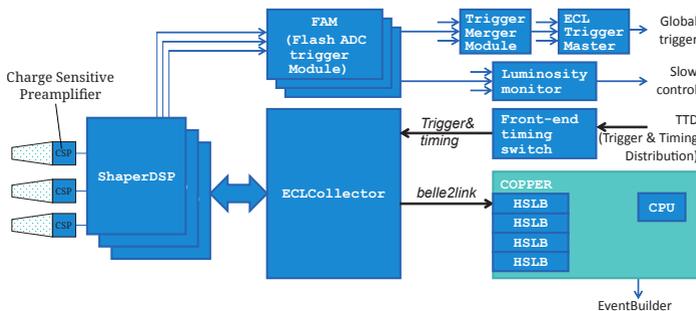


Рисунок 2 — Схема системы сбора данных с калориметра [5]

В четвёртом разделе описывается онлайн-монитор светимости (LOM, Luminosity Online Monitor) на базе электромагнитного калориметра. LOM использует быстрые сигналы с модулей ShaperDSP, относящихся к переднему и заднему торцам ECL. Встроенный микропроцессор модуля LOM может получать и обрабатывать внешние запросы через интерфейс Ethernet по протоколу TCP.

В пятом разделе описываются стандартные калибровочные процедуры, регулярно проводящиеся для эффективного функционирования калориметра и монитора светимости. Выделяется четыре типа калибровки. Ежедневно проводится калибровка с тестовым импульсом, раз в две недели калибруется монитор светимости. Раз в месяц проводится

калибровка нелинейности, при которой набираются данные с тестовым импульсом на разных амплитудах. В случае замены плат ShaperDSP, проводится ряд заходов для калибровки формы сигнала.

*В шестом разделе* описывается общая система DAQ эксперимента Belle II, от процесса сборки событий до сохранения их на диск и последующей обработки. По сравнению с детекторами LHC, эксперимент Belle II работает с меньшими объёмами данных, но поток данных 18 ГБ/с приводит к необходимости использовать триггер высокого уровня (High Level Trigger, HLT) для фильтрации событий.

*В седьмом разделе* описывается программное обеспечение (ПО) системы сбора данных, в частности используемые базы данных (БД), ПО медленного контроля и управления заходами, а также инструменты для координации деятельности экспертов. Двумя основными БД являются ConditionsDB и DAQ DB. ConditionsDB предназначена преимущественно для хранения калибровочной информации, а DAQ DB предназначена для хранения различных конфигураций, относящихся к системе сбора данных. Обе БД используют систему управления БД PostgreSQL.

С этими базами данных взаимодействует модульный фреймворк Belle Analysis Software Framework 2 (BASF2) и системы медленного контроля, причём в эксперименте используется довольно широкое определение системы медленного контроля — это сеть взаимодействующих демонов, которые осуществляют мониторинг оборудования, предоставляют интерфейс чтения/записи к регистрам электроники детектора и управляют общей системой сбора данных. Для медленного контроля используется два фреймворка: NSM2 и EPICS. Для графического интерфейса пользователя используется фреймворк Control System Studio.

Важной частью системы DAQ является монитор качества данных (DQM, Data Quality Monitor) — набор программных модулей BASF2, которые в реальном времени обрабатывают экспериментальные данные и предоставляют гистограммы ROOT, позволяющие судить о текущем качестве данных.

Также для обеспечения стабильности сбора данных важны инструменты по координации работы экспертов. График смен дежурств устанавливается в менеджере расписаний ShiftTool, долговременная информация хранится в Confluence, для быстрой коммуникации используется RocketChat.

**Во второй главе** анализируются требования к разработке программного обеспечения в сети сбора данных. Описывается специфика разработки ПО в рамках эксперимента Belle II. Рассматривается процесс выбора программных средств, наиболее подходящих для поставленных задач.

*В первом разделе* описываются основные задачи, которые должны осуществляться системами мониторинга и медленного контроля.

1. Обновление конфигурации электроники.
2. Обновление конфигурации систем мониторинга.
3. Инициализация электроники.
4. Считывание данных.
5. Онлайн-мониторинг качества данных.
6. Подготовка новой конфигурации.

Поскольку для чтения данных используется универсальное программное обеспечение, одинаковое для всех систем детектора, данная работа фокусируется на всех задачах, кроме задачи 4.

Кроме того, поскольку монитор светимости должен предоставлять информацию непрерывно, вне зависимости от текущего статуса захода, для него необходимо разработать отдельную система сбора данных. Она должна решать все те же задачи, что и основная система DAQ — управление конфигурациями, инициализацию, мониторинг, запись и анализ данных, при этом интегрируясь с основной системой медленного контроля.

*Во втором разделе* рассматривается специфика разработки программного обеспечения для распределённых систем с быстро эволюционирующими требованиями, рассматриваются архитектурные и организационные решения для наиболее оптимального процесса разработки проекта.

Одним из важных требований является быстрая диагностика и обработка исключительных ситуаций — возможных сбоев в работе как электроники, так и программного обеспечения. Для этого важно проработать пользовательский интерфейс и систему автоматического оповещения об ошибках.

Поскольку ECL DAQ является распределённой системой, разработанные программные модули должны предоставлять Application Programming Interface (API) для обмена информацией друг с другом. Причём они должны удовлетворять двум сценариям использования — основной режим, интеграция со всеми остальными процессами системы сбора данных и изолированный режим, предназначенный для локальных тестов во время технических работ, когда центральные БД и сервера системы медленного контроля могут быть недоступны.

Исходя из этих требований было решено использовать микросервисную архитектуру при разработке программного обеспечения для данных задач.

*В третьем разделе* приводятся требования к пользовательскому интерфейсу, рассматриваются способы реализации, потенциальные трудности и процесс дальнейшего сопровождения.

В работе используется 4 типа пользовательских интерфейсов, упорядоченные по возрастанию сложности разработки, от наиболее простого к наиболее сложному:

1. Интерфейс командной строки (CLI).
2. Текстовый интерфейс пользователя (TUI).
3. Графический интерфейс пользователя (GUI).
4. Веб-интерфейс пользователя (WUI).

Важно также иметь возможность автоматизировать тестовые запросы. В CLI и TUI разработка автоматизированных тестов осуществляется довольно просто, в то время как для GUI и WUI нужно либо реализовывать специальные API для доступа ко всем основным функциям, либо использовать специальные библиотеки наподобие Selenium.

Поскольку Belle II является международным экспериментом, необходимо, чтобы все разрабатываемые интерфейсы были доступны удалённо. Для доступа к CUI и TUI используется протокол Secure Shell, для GUI — протокол Virtual Network Computing.

*В четвёртом разделе* кратко формулируются требования к программному интерфейсу. Устанавливается, что разрабатываемые программные интерфейсы должны соответствовать стандарту REpresentational State Transfer (REST) API, если они просто предоставляют доступ к данным или стандарту Remote Procedure Call (RPC), если они обеспечивают удалённый вызов процедур.

*В пятом разделе* приводятся используемые программные средства, рассматривается использование прототипирования, а также построение процесса разработки ПО.

Используемые программные средства были преимущественно определены существующими фреймворками, использующимися в эксперименте Belle II, для обработки больших объёмов данных использовался язык программирования C++, для остальных задач применялся язык Python, за исключением веб-разработки, где также применялись языки PHP и Javascript.

Отмечается, что стандартная реализация Python (CPython) имеет плохую производительность в многопоточных приложениях, поэтому

такие приложения либо реализовывались на C++, либо разбивались на несколько независимых процессов.

Разработка выполнялась итеративно, основываясь на модели Rapid Application Development. Фокус шёл на разработку прототипов, где существующие прототипы использовались, чтобы обеспечивать обратную связь и последовательную корректировку проекта на основе быстро меняющихся требований эксперимента.

**В третьей главе** описывается программное обеспечение, управляющее конфигурациями электроники калориметра. Указывается, как реализованные программы далее увязываются с задачами инициализации и мониторинга калориметра.

Конфигурация ECL содержит три типа информации:

1. Статические параметры, то есть параметры, которые не зависят от калибровочных коэффициентов. Эти параметры обычно имеют одинаковое значение для всех модулей и модифицируются только для особых тестов. К примеру, битовая маска используемых модулей ShaperDSP является статическим параметром.
2. Энергетические пороги и коэффициенты аттенюатора — параметры, которые, как правило, могут быть определены как арифметическое выражение. К примеру, порог величиной 50 МэВ определяется как  $50 \cdot \text{energy\_conversion\_coef}$ . Коэффициенты для преобразования из единиц АЦП в энергию определяются из процедуры энергетической калибровки.
3. Коэффициенты DSP, которые определяются индивидуально для каждого канала, записываются во флэш-память ECLCollector и затем загружаются в SDRAM модулей ShaperDSP. Из-за большого числа этих коэффициентов, их требуется сжимать перед загрузкой в базу данных.

Конфигурация ECL включает в себя 80256 параметров, причём используются разные наборы этих конфигураций для физических заходов и 14 типов калибровочных заходов.

*В первом разделе* более детально рассматривается специфика работы двух основных баз данных в эксперименте, описывается синхронизация информации между ними.

По причине сильного различия внутренней структуры ConditionsDB и DAQ DB, синхронизировать информацию между ними довольно затруднительно. Поэтому были разработаны две утилиты на языке программирования Python, осуществляющие доступ к ConditionsDB и DAQ DB. Диаграмма передачи информации между двумя БД показана на рисунке 3.



Рисунок 3 — Передача данных между ConditionsDB и DAQ DB

Во втором разделе представлена схема хранимых данных, позволяющая гибко настраивать конфигурацию отдельных модулей или секторов калориметра. Была переработана существующая схема хранения данных в DAQ DB, удалось уменьшить дублирование информации. Для детальных настроек разработан парсер, обрабатывающий сложные конфигурации, внутри которых можно задавать циклы и логические условия.

В третьем разделе описан процесс синхронизации энергетических порогов и карты каналов между двумя базами данных. Чтобы моделирование соответствовало эксперименту, в Belle II доступен режим «позаходного моделирования», в котором параметры моделирования основываются на конфигурации детектора во время соответствующего периода времени.

Чтобы такой режим моделирования работал, а также, чтобы обновлять конфигурацию электроники на основе новых калибровочных коэффициентов, необходимо иметь надёжную процедуру для синхронизации данных между ConditionsDB и DAQ DB.

Реализована процедура, при которой энергетические пороги и некоторые другие параметры электроники автоматически генерируются и загружаются в считывающую электронику перед запуском набора данных. В момент загрузки, новая версия параметров загружается в ConditionsDB, для использования в мониторе качества данных и в моделировании.

Также для упрощения синхронизации в ConditionsDB была добавлена карта каналов калориметра, задающая соответствие между кристаллом и каналом в электронике. Таким образом, утилиты, синхронизирующие две БД всегда могут использовать актуальное соотношение канала и кристалла.

В четвёртом разделе описан процесс синхронизации DSP-коэффициентов, используемых для подгонки сигналов в модулях ShaperDSP. Во время сбора данных ионизирующее излучение может привести к сбоям в логике работы ПЛИС, поэтому необходимо в реальном времени проверять корректность подгонки сигнала в каждой плате. Для этого в

мониторе качества данных для небольшого числа событий выполняется программа, повторно подгоняющая сигнал. Результаты подгонки затем сверяются с данными из ПЛИС.

Чтобы такая проверка работала, необходимо синхронизировать использующиеся в подгонке DSP-коэффициенты между DAQ DB и ConditionsDB. Была разработана процедура преобразования DSP-коэффициентов из бинарного файла в файл библиотеки ROOT. Также был реализован алгоритм упаковки данных, позволяющий уменьшить размер массива DSP-коэффициентов в 3 раза вдобавок к стандартным алгоритмам сжатия. Улучшение эффективности сжатия показано на рисунке 4.

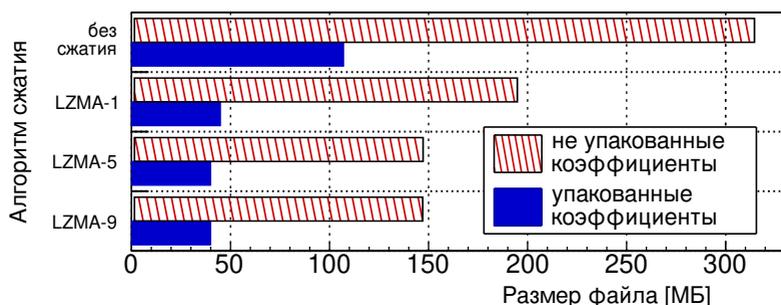


Рисунок 4 — Сравнение размеров выходного файла на разных уровнях сжатия для не упакованных и упакованных данных

**В четвёртой главе** описан процесс инициализации электроники калориметра. Указаны все шаги процесса инициализации, сопутствующие сложности и выбранные пути их решения.

*В первом разделе* описана процедура инициализации. Модули ECLCollector и ShaperDSP инициализируются через последовательно проводящиеся шаги, выполняющиеся по интерфейсу тестирования электроники JTAG, а также протоколам User Datagram Protocol (UDP) и Belle2link, с серверов, показанных на рисунке 5.

По JTAG инициализируется прошивка модуля ECLCollector, по UDP обновляются значения во флэш памяти. Наконец, через Belle2link устанавливается около 60 000 параметров, зависящих от калибровки (энергетические пороги, аттенуаторные коэффициенты и др.) и 20 000 статических параметров.

Процедура инициализации должна надёжно работать в нескольких сценариях использования, поддерживать резервные опции, корректно работать по нестабильному соединению.

*Во втором разделе* описаны дополнительные требования к про-

граммным средствам для инициализации calorimetра. В частности, должны поддерживаться два сценария: стандартный режим работы, когда требуется интеграция с системой медленного контроля и изолированный режим, который не зависит от глобальной системы DAQ.

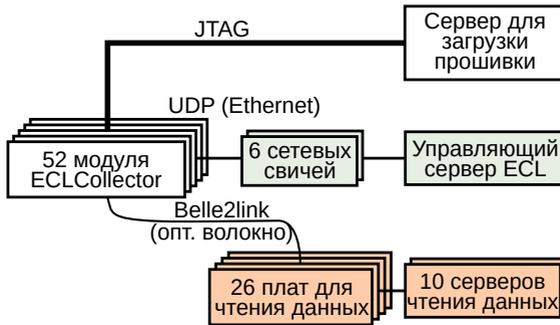


Рисунок 5 — Сервера, используемые в процедуре инициализации ECL

В третьем разделе описана структура фреймворка для инициализации модулей ECLCollector и ShaperDSP. Показано, как возможности фреймворка удовлетворяют ранее поставленным требованиям. Операции ввода-вывода вынесены в абстрактный класс, поэтому интерфейс работы с ECLCollector легко модифицируется. Также отдельная система классов реализована для работы с конфигурацией электроники, можно использовать как информацию из DAQ DB, так и локальное хранилище, что позволяет при необходимости запускать инициализацию в изолированном режиме, без доступа к базе данных. Кратко описывается система сборки — используется GNU Make с нерекурсивным Makefile.

Приведены некоторые дополнительные детали реализации фреймворка — для обработки ошибок вместо исключений C++ используется система с отправкой информации об ошибке, использующая класс, реализованный наподобие `std::variant`, что позволяет улучшить быстродействие на ~10%.

Кроме того, описаны средства автоматизированного тестирования, которые позволяют реализовать фреймворк и улучшения в быстродействии, связанные с внутренней реорганизацией запросов к электронике.

В четвертом разделе описаны высокоуровневые утилиты инициализации. Поскольку инициализация должна запускать и отслеживать корректность выполнения 52 параллельных процессов на 26 удалённых серверах, представлены различные способы предоставления удобного пользовательского интерфейса, который позволяет эксперту получать доступ к детальной информации о процессе инициализации и коррект-

но обнаруживать ошибки на любом шаге процесса. Итоговое решение использует фреймворк Falcon, позволяющее добавлять задачи инициализации по REST API. Отображение статуса задач реализовано с помощью библиотеки urwid. Также описаны различные способы указать набор модулей ECLCollector, которые требуется инициализировать.

**В пятой главе** описывается система медленного контроля и управления заходами для ЭМ калориметра детектора Belle II. Также описывается процесс организации деятельности дежурных экспертов по калориметру и средства автоматизации, которые позволяют упростить работу дежурных экспертов.

*В первом разделе* описан программный модуль, использующий программную библиотеку Selenium, чтобы автоматизированно управлять большим числом сетевых коммутаторов HP1810. Это позволяет на сетевом уровне управлять доступом к модулям ECLCollector, позволяя надёжно предотвращать ошибочный доступ к ним во время набора данных.

*Во втором разделе* описаны процессы медленного контроля, управляющие модулями ECLCollector. Разработанные процессы устанавливают конфигурацию электроники, а также подстраивают параметры конфигурации в зависимости от текущего типа захода и типа триггера. Также реализован программный интерфейс, позволяющий управлять модулями ECLCollector с любого сервера, входящего в сеть медленного контроля.

*В третьем разделе* описана библиотека pyNSM2, которая оборачивает фреймворк NSM2 в модуль Python с помощью библиотеки ctypes. Также реализован отдельный шаг сборки, в котором модуль rusparser автоматически обновляет код pyNSM2, если версия NSM2 обновилась. Рассматриваются реализованные средства для дополнительного мониторинга системы. Чтобы упростить использование и поддержку pyNSM2, разработан набор вспомогательных библиотек и автоматизированные тесты.

*В четвёртом разделе* описаны вспомогательные программные средства для координации деятельности дежурных. Поскольку информация о текущих дежурных хранится в нескольких местах, была разработана утилита, периодически синхронизирующая информацию между тремя разными источниками.

*В пятом разделе* показан графический интерфейс для управления локальными заходами с электромагнитным калориметром. Этот интерфейс используется для набора калибровочных данных, использующихся в калибровочных процедурах, перечисленных в первой главе, четвёртом разделе. Указаны добавления к графическому интерфейсу, в том числе API, логирование и база данных. Также была добавлена интеграция

с DAQ DB, что позволило отображать логи системы сбора данных в графическом интерфейсе.

*В шестом разделе* приведены шаги, выполненные для автоматизации процедуры калибровки по тестовому сигналу. Набранные данные автоматически обрабатываются, заносятся в базу данных и загружаются на веб-сервер. Реализовано отображение результатов калибровки в библиотеке Dugraphs.

*В седьмом разделе* описан веб-сервер экспертов по ECL. Это Apache сервер, предоставляющий единый доступ к большинству описанных выше функций. Для быстрого доступа к функциям инициализации и конфигурации используется встроенный в браузер эмулятор терминала на основе библиотеки Shell In A Box, доступ к графическим приложениям осуществляется через протокол Virtual Network Computing (VNC) и веб-интерфейс по VNC. Внутри ИЯФ также организован промежуточный веб-сервер, предоставляющий централизованный доступ ко всем утилитам через osproху, сервис, поддерживающий соединение по Virtual Private Network с удалённым сервером.

Поскольку в ходе дежурства используется довольно большое количество веб-приложений, на языке JavaScript, используя библиотеку Simone был реализован оконный менеджер, позволяющий на одной странице отображать в настраиваемом виде информацию с нескольких различных веб-приложений.

*В восьмом разделе* описывается структура документации для экспертов по калориметру. Она разделена на четыре основные секции: средства коммуникации, мониторинг качества данных, диагностика проблем, исправление проблем с калориметром.

**В шестой главе** описываются средства мониторинга данных с ECL, средства оповещения об ошибках, интеграция с центральным репозиторием Belle II для более детального мониторинга и оперативного информирования о качестве данных.

*В первом разделе* перечисляются гистограммы, которые отображаются в мониторе качества данных. Гистограммы разделены на три группы: низкоуровневая информация для контроля целостности данных; калибруемые величины (энергетическое и временное распределение); данные о корректности работы ПЛИС.

*Во втором разделе* описываются небольшие добавления к веб-интерфейсу DQM. В веб-интерфейс интегрирована информация о карте каналов калориметра, добавлено отображение всплывающих подсказок о том, какие модули соответствуют указанному каналу. Также реализован адаптивный стиль расстановки гистограмм, ориентирующийся на размер окна браузера.

В третьем разделе описывается мониторинг фона инъекции. Было обнаружено, что высокий фон инъекции приводит к систематическому занижению реконструированной амплитуды сигнала. Чтобы отслеживать влияние фона инъекции, был реализован набор гистограмм, который позволяет отдельно рассматривать энергетическое распределение фона в разные моменты времени после инъекции.

В четвёртом разделе показан монитор качества данных для интервала заходов, позволяющих отслеживать долговременную динамику мониторируемых параметров калориметра. При остановке захода, интегральные характеристики качества данных отправляются на внешний веб-сервер, где эксперты могут строить графики этих характеристик от номера захода. Пример такого графика показан на рисунке 6.

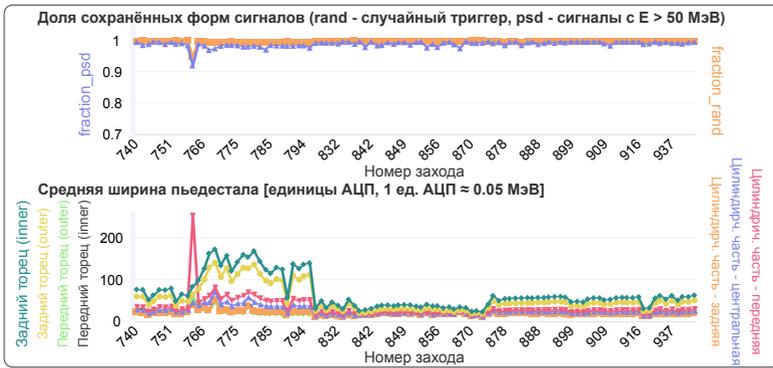


Рисунок 6 — Зависимость доли сохранённых форм сигнала и ширины пьедестала от номера захода

В пятом разделе описаны средства интеграции монитора качества данных с системой медленного контроля EPICS. Гистограммы в формате JSON ROOT автоматически скачиваются и проверяются. Было реализовано несколько процессов медленного контроля, оповещающих дежурных в случае проблем в качестве данных. Также интегральная информация с гистограмм в реальном времени обрабатывается и отправляется в EPICS, в частности информация о текущем уровне фона.

В шестом разделе описаны средства архивации данных мониторинга. Для этого в эксперименте Belle II используется EPICS Archiver. Архивируются статические параметры со всех модулей ECL; параметры, основанные на калибровке (энергетически пороги и аттенюаторные коэффициенты); версия DSP-коэффициентов.

В седьмом разделе описаны средства для автоматического информирования дежурных о возможных проблемах с качеством данных.

Реализован набор утилит для автоматического оповещения по email и в RocketChat. Сообщения содержат детальную информацию об исправлении возникшей проблемы, система оповещений конфигурируется, создана небольшая база данных, хранящая историю недавно отправленных сообщений.

**В седьмой главе** описано программное обеспечение для считывания данных с монитора светимости. Поскольку измерение светимости должно определяться независимо от состояния системы триггера детектора, была разработана отдельная система сбора данных для мониторинга светимости.

*В первом разделе* описаны используемые программные средства и сетевая диаграмма процессов, отвечающих за чтение данных с монитора светимости. Доступ к монитору светимости осуществляется единообразно через прокси. На уровне прокси реализованы кэширование и построение очереди запросов.

*Во втором разделе* описаны инструменты мониторинга качества данных с монитора светимости на основе фреймворка Qt 5. Было реализовано чтение данных с монитора светимости, а также доступ к графическому интерфейсу через веб-сервис noVNC.

*В третьем разделе* указаны средства для экспорта данных в системы медленного контроля и дополнительные требования, повлиявшие на формирование архитектуры ПО. Для экспорта в EPICS используется библиотека PythonIOC. Чтобы избежать явной зависимости от NSM2, в эту систему медленного контроля экспорт выполняется отдельной программой.

*В четвёртом разделе* описана оболочка командной строки для детального управления монитором светимости. Детальное управление модулем монитора осуществляется через командную строку, реализованную с помощью библиотеки `prompt_toolkit`. Поддерживаются как низкоуровневые команды наподобие установления регистров, так и многоступенчатые процедуры для тестирования и инициализации модуля.

*В пятом разделе* описана процедура энергетической калибровки. Был реализован автоматический перенос данных калибровки на сервер вычислительного центра, а также функции, позволяющие параллельно конфигурировать электронику калориметра и монитор светимости.

*В шестом разделе* описаны средства автоматизированного тестирования. Разработаны процедуры тестирования как новых версий прошивки модуля, так и новых версий программного обеспечения.

*В седьмом разделе* описаны средства автоматизации развёртки программного обеспечения монитора светимости. Чтобы упростить управление внешними зависимостями, был создан отдельный репозиторий, содержащий набор скриптов, определяющих используемый дистри-

бутив Linux и архитектуру процессора, скачивающий и компилирующий необходимые библиотеки.

**В восьмой главе** описано программное обеспечение для обработки сохранённых данных с калориметра. Приводится описание ПО, призванного упростить обработку записанных файлов для детальной проверки качества данных.

*В первом разделе* описана временная калибровка калориметра. В рамках работы были реализованы начальные версии двух процедур: временная калибровка по космическим событиям и по событиям Бабы.

*Во втором разделе* описаны процедуры по анализу кода распаковщика данных с калориметра и его последующая оптимизация. Распаковка событий занимает существенную долю процессорного времени, поэтому оптимизация этой части кода являлась приоритетной задачей. Для этого было проведено профилирование кода с использованием callgrind. Определив узкие места в коде, удалось получить улучшение быстродействия на ~20%.

*В третьем разделе* описана конфигурация распаковщика, позволяющая по-особому обрабатывать данные за указанные временные периоды. Реализованный код позволяет исключить или попытаться восстановить данные с неисправных модулей электроники калориметра. Также такая конфигурация может повторно использоваться в моделировании, чтобы минимизировать различия с экспериментом.

*В четвёртом разделе* описана утилита, призванная ускорить процесс калибровки при обработке большого объёма данных. Для указанного набора файлов на кластере запускается соответствующее число задач и выставляется порядок запуска, чтобы постепенно объединять все результаты в конечный файл.

*В пятом разделе* описано программное обеспечение, разработанное для калибровки нелинейности. Реализованная программа автоматически определяет входные файлы на основе базы данных калибровочных заходов.

*В шестом разделе* описаны вспомогательные библиотеки, используемые для разработки управляющих файлов на Python. Реализован модуль TTreeWriter, упрощающий запись данных в деревья ROOT, а также модули для агрегирования информации из различных баз данных и систем медленного контроля.

**В заключении** перечислены основные результаты работы и даны рекомендации дальнейшей разработки текста.

## ЗАКЛЮЧЕНИЕ

- Подготовлены программные модули для синхронизации и управления конфигурацией электроники сбора данных в двух базах данных эксперимента Belle II. Конфигурация может корректироваться для разных типов заходов, история конфигурации архивируется и может быть в дальнейшем использована для моделирования калориметра с заданными настройками. Разработаны решения для задания более детальной тестовой конфигурации. Чтобы эффективнее использовать ресурсы БД, создан алгоритм упаковки, подстроенный под использующуюся структуру конфигурационных данных. Его использование позволило увеличить эффективность сжатия в три раза, не теряя при этом в быстродействии.
- Разработан фреймворк, объединяющий все функции инициализации электроники калориметра. Используемая архитектура фреймворка позволяет абстрагироваться от используемого протокола, предоставляет возможности для автоматизированного тестирования программного обеспечения и может адаптироваться ко всем требуемым сценариям использования. Также были подготовлены высокоуровневые утилиты инициализации, которые позволяют довольно просто выполнять стандартные операции и поддерживают большое число дополнительных параметров работы. Эти утилиты регулярно используются дежурными экспертами для диагностики и исправления проблем с калориметром, а также при тестировании новых версий прошивки. Процедура инициализации может работать независимо от основной системы медленного контроля, развёртывая свою сеть взаимодействующих процессов.
- Большинство разработанных приложений интегрированы с фреймворком Network Shared Memory 2. Для ускорения процесса разработки была реализована библиотека ruNSM2, позволяющая взаимодействовать с системой медленного контроля с использованием языка Python. Подготовлено несколько системных демонов, управляющих электроникой калориметра, расширен графический интерфейс для запуска калибровок, создана отдельная база данных с информацией по набранным калибровочным заходам.
- Все основные процедуры диагностики и устранения проблем с электроникой калориметра были внесены в документацию дежурных. Для экспертов разработан веб-сервер, объединяющий все необходимые утилиты для быстрого доступа. Разработано несколько приложений для мониторинга качества данных с кало-

риметра, реализована система автоматических оповещений.

- Было подготовлено программное обеспечение для считывания данных с монитора светимости и передачи информации о светимости в систему медленного контроля. Детальная информация о мониторе светимости также доступна на веб-сервере дежурного.
- В ходе работы были разработаны и дополнены некоторые модули для обработки данных с калориметра. В частности, были разработаны исходные версии модулей временной калибровки, оптимизирована распаковка данных и реализованы процедуры для первичной обработки данных в калибровке нелинейности. Для детального исследования качества данных реализован набор управляющих скриптов Python, с отдельной библиотекой для структурированной записи выходных данных.

### **Публикации автора по теме диссертации**

1. CsI(Tl) pulse shape discrimination with the Belle II electromagnetic calorimeter as a novel method to improve particle identification at electron–positron colliders / S. Longo, J.M. Roney, C. Cecchi, S. Cunliffe [et al.]. — Текст : электронный // Nuclear Instruments and Methods in Physics Research Section A. — 2020. — Vol. 982. — P. 164562. — DOI 10.1016/j.nima.2020.164562. — Дата публикации: 01.12.2020.
2. Data acquisition system for Belle II electromagnetic calorimeter / A. Kuzmin, M. Remnev, D. Matvienko, Y. Usov [et al.]. — Текст : электронный // Journal of Instrumentation. — 2020. — Vol. 15, nr. 7. — P. C07020. — DOI 10.1088/1748-0221/15/07/C07020. — Дата публикации: 13.07.2020.
3. Data acquisition system for the calorimeter of the Belle II detector / A. Kuzmin, M. Remnev, D. Matvienko, Y. Usov [et al.]. — Текст : электронный // Physics of Atomic Nuclei. — 2021. — Vol. 84, nr. 1. — P. 42–44. — DOI 10.1134/S1063778821010257. — Дата публикации: 13.04.2021.
4. Trigger slow control system of the Belle II experiment / С.-Н. Kim, Y. Unno, Н.Е. Cho, В.Г. Cheon [et al.]. — Текст : электронный // Nuclear Instruments and Methods in Physics Research Section A. — 2021. — Vol. 1014. — P. 165748. — DOI 10.1016/j.nima.2021.165748. — Дата публикации: 17.08.2021.
5. Development of data acquisition system for Belle II electromagnetic calorimeter / V. Aulchenko, A. Bobrov, В.Г. Cheon, A. Kuzmin [et al.]. — Текст : электронный // Nuclear Instruments and Methods in Physics Research Section A. — 2022. — Vol. 1030. — P. 166468. — DOI 10.1016/j.nima.2022.166468. — Дата публикации: 15.02.2022.

*Ремнев Михаил Анатольевич*

Разработка программного обеспечения для системы сбора данных  
электромагнитного калориметра детектора Belle II

Автореф. дис. на соискание ученой степени кандидата технических наук

Подписано в печать « 13 » \_\_\_\_\_ октября \_\_\_\_\_ 2023 г.

Заказ № \_\_\_\_\_ 14 \_\_\_\_\_

Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.

Типография \_\_\_\_\_ ИЯФ СО РАН \_\_\_\_\_