

# § 2 ПОКАЗАТЕЛИ КАЧЕСТВА И ПОВЫШЕНИЕ НАДЕЖНОСТИ ПРОГРАММНЫХ СИСТЕМ

Афондеркин С.Ю., Гаязов С.Е., Игнатов Ф.В.

## ИСПОЛЬЗОВАНИЕ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДЛЯ КОНТРОЛЯ КАЧЕСТВА ДАННЫХ ДЕТЕКТОРА КМД-3

**Аннотация:** КМД-3 – детектор элементарных частиц, работающий на электрон-позитронном коллайдере ВЭПП-2000 в Институте Ядерной Физики им. Г.И.Будкера. Для обеспечения конечного контроля качества записанных данных и диагностики сбоев в подсистемах детектора, выполняется полная реконструкция событий с помощью программного обеспечения, разработанного коллаборацией КМД-3. В зависимости от светимости коллайдера, на установке может регистрироваться до 1500 событий в секунду, что в 30 раз превышает среднюю скорость их обработки одним процессом. В целях повышения эффективности работы системы сбора данных детектора и ускорительного комплекса в целом, с помощью распределения потока событий по нескольким процессам обработки, скорость реконструкции была увеличена до значений, возникающих при пиковой светимости. В статье приведен общий обзор систем для сбора и обработки данных детектора, а также описана архитектура распределенной версии программного обеспечения реконструкции событий, зарегистрированных детектором. Для поиска оптимального архитектурного решения были использованы методы системного анализа и структурной декомпозиции компонентов систем сбора и обработки данных, в результате в качестве решения был предложен и реализован паттерн “трубы и фильтры”. При реализации была применена методология императивного параллельного программирования. Приложение для обработки данных было дополнено модулями для передачи необходимых данных между обрабатываемыми процессами. Реализованная система может обрабатывать события со скоростью их записи, что позволяет получать информацию о интегральных характеристиках работы комплекса и реагировать на сбои в работе электроники и ПО детектора в реальном времени.

**Ключевые слова:** быстрый анализ, ускорительный комплекс, анализ данных, распределенная система, КМД-3, повышение скорости обработки, распределенная обработка, обработка данных, контроль качества, реконструкция событий детектора

**Abstract:** *CMD-3 – a particle detector, running on electron-positron collider VEPP-2000 in the Budker Institute of Nuclear Physics. In order to ensure the final quality control of recorded data and diagnosis of faults in the subsystems of a detector a full reconstruction of events is performed by software developed by CMD-3 collaboration. Depending on the luminosity of the collider up to 1500 events per second can be detected, which is 30 times higher than the average speed of its processing by a single process. In order to improve the efficiency of data collection and detector system as a whole, the speed of reconstruction was increased using multiple distribution processing processes to keep up with the values arising at peak luminosity. The article provides an overview of systems for data collection and processing of the detector and also describes the architecture of a distributed version of the software reconstruction of the events recorded by the detector. To find the best architectural solution methods of system analysis and structural decomposition components of data collection and processing systems were used, resulting in a proposed solution that implements a pattern of “pipes and filters”. A methodology of imperative parallel programming has been applied. Modules for the transfer of necessary data processing between processes were added to the application. The implemented system can handle events with the speed of recording, which provides information on the integral characteristics of the complex, and allows responding to malfunctions in the electronics and software of the detector in real time.*

**Keywords:** *reconstruction of the events of the detector, rapid analysis, accelerating complex, data analysis, distributed system, increasing the speed of data processing, CMD-3, distributed processing, data processing, quality control*

## Введение

Одним из важных направлений исследований в Институте Ядерной Физики им. Г.И.Будкера [1] является измерение сечений процессов электрон-позитронной аннигиляции в адроны в области энергий, соответствующим массам легких векторных мезонов. Подобные измерения являются основным источником информации о свойствах легких векторных мезонов и их возбужденных состояниях. Прецизионное измерение сечений аннигиляции с образованием адронов требуется для теоретического расчета адронного вклада в аномальный магнитный момент мюона, значение которого служит тестом для проверки Стандартной Модели и поиска новой физики [2].

Для решения этих и других задач в Институте Ядерной Физики был создан коллайдер ВЭПП-2000 [3; 4], работающий с «круглыми» встречными  $e+e-$  пучками с энергией до 2 ГэВ в системе центра масс. Для проведения экспериментов на коллайдере ВЭПП-2000 используются два детектора: Сферический Нейтральный Детектор (СНД) [5] и Криогенный Магнитный Детектор третьего поколения (КМД-3) [6].

КМД-3 является универсальным детектором, позволяющим с высокой точностью определять характеристики как заряженных, так и нейтральных частиц. Электроника сбора данных детектора КМД-3 - это совокупность блоков приема/передачи данных, оцифровывающих, триггерных и других видов блоков. Все физические величины сигналов (заряды, амплитуды, времена) преобразуются в напряжения, которые оцифровываются

с помощью АЦП. Данные с плат АЦП формируются в пакеты и отправляются на принимающий компьютер по сети.

На принимающем компьютере специальное программное обеспечение регистрирует данные и объединяет данные, полученные из различных источников. Оцифрованные сигналы, полученные с датчиков детектора в один момент времени, формируются в событие. События объединяются в заход и хранятся в виде сжатых файлов, которые после записи отправляются в систему долговременного хранения.

### Сбор данных детектора КМД-3

Все программное обеспечение детектора можно разделить на две основные части: систему сбора данных и систему обработки данных [7]. В основе системы сбора данных детектора КМД-3 лежит программный пакет Maximum Integrated Data Acquisition System (MIDAS) [8]. Программный пакет был разработан в PSI (Швейцария) и используется в Институте ядерной физики и других институтах в качестве системы сбора данных в экспериментах в области физики высоких энергий. MIDAS состоит из набора сервисов, библиотеки для работы с ними, а так же набора вспомогательных программ. MIDAS позволяет построить расширяемую, гибкую программную систему автоматизации эксперимента благодаря двум основным сервисам:

- Онлайн базы данных (ODB) – используется как средство хранения конфигурации и контроля над экспериментом.
- Буфер(Buffer) – используется как средство передачи данных (событий) между процессами.

Источниками данных в системе являются специализированные программы – фронтенды, которые получают данные с оцифровывающей аппаратуры эксперимента, производят первичную обработку данных и переформатирование. Процесс EventBuilder выполняет сборку события из фрагментов, полученных от фронтендов. Затем сформированные события передаются процессу Logger, который осуществляет запись события в файл захода. Процесс сбора данных схематично изображен на рисунке 1.

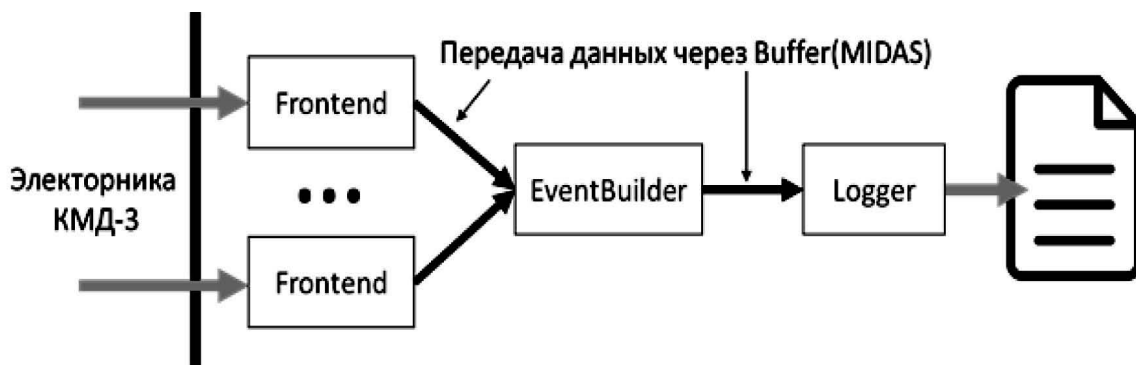


Рисунок 1 – Процесс сбора данных детектора КМД-3

В зависимости от светимости установки формирование файла захода может длиться от нескольких минут до часа, причем меньшей светимости соответствует большее время. Данные одного события, зафиксированного детектором, имеют размер в среднем 10 Кб. За одну секунду на комплексе может регистрироваться до 1500 событий, при этом поток данных с детектора может достигать до 15 Мб в секунду. Полученные данные объединяются в файлы захода примерно по 200 тысяч событий каждый и до сжатия имеют размер около 2 Гб. После сжатия файл захода помещается в хранилище, а его копия по сети отправляется на другой компьютер для стандартной обработки и построения гистограмм со статистикой. Набранные гистограммы используются для оперативного контроля набираемой статистики. В дальнейшем, в процессе анализа данных, файлы заходов неоднократно переобрабатываются с использованием версий алгоритмов реконструкции, адаптированных для анализа конкретного типа физических событий.

Файлы заходов пишутся непрерывно, так как установка работает круглые сутки, семь дней в неделю, с небольшими перерывами на техническое обслуживание, поэтому к используемому ПО предъявляются требования высокой надежности и корректной работы в случае ошибок в работе оборудования детектора и других внештатных ситуациях.

### Обработка данных детектора КМД-3

На базе программного пакета Gaudi [9] было разработано приложение Cmd3fwk [10] для обработки данных детектора. Пакет Gaudi предоставляет систему управления модулями и выполняет следующие функции:

- загрузка, удаление и конфигурация каждого модуля;
- организация последовательности выполнения модулей;
- обеспечение передачи данных между модулями;
- управление и синхронизация работы модулей.

Единицей управления в системе является модуль. Модуль – это логически завершенная взаимосвязанная совокупность функциональных частей, оформленных в виде библиотеки, содержащей статическую реализацию абстрактного класса CmdModule и реализующей методы для собственной инициализации, выполнения обработки события и завершения обработки захода. Каждый модуль Cmd3fwk реализует определенный этап обработки, одни выполняют чтение события из источника данных, а другие их реконструкцию, часть модулей накапливают статистику в виде гистограмм, так же реализованы модули, которые сохраняют данные в файл.

Связь между модулями осуществляется посредством общего хранилища данных – DataProxu. DataProxu поддерживает работу с данными любого типа. Модуль может регистрировать данные в DataProxu и запрашивать их оттуда. Доступ к данным в пределах DataProxu осуществляется при помощи строкового имени. Количество

данных, которые модуль может зарегистрировать или запросить в/из DataProxy, не ограничено.

Модули Cmd3fwk компилируются в динамически загружаемые библиотеки. С помощью определяемого набора модулей пользователь может создать собственную программу по обработке данных с нужной ему логикой работы. Модули, которые будут загружены для обработки события, и порядок их выполнения указывается в конфигурационном файле, путь к которому передается в аргументах запуска Cmd3fwk. Процесс обработки данных детектора с помощью Cmd3fwk схематично изображен на рисунке 2.

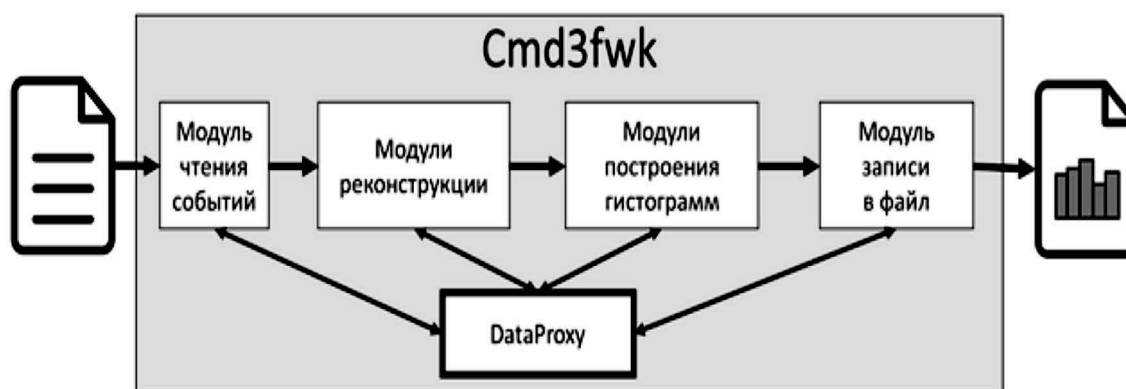


Рисунок 2 – Процесс обработки данных детектора с помощью Cmd3fwk

При обработке данных детектора КМД-3 используется два типа событий – сырые и обработанные, и два формата их хранения. Сырое событие генерируется системой сбора данных и представляет собой набор банков MIDAS, в которых хранится информация о сработавших элементах. Сырое событие сохраняется в виде бинарной записи в файле, в формате, используемом системой MIDAS. Обработанное событие содержит реконструированную информацию высокого уровня – измеренные значения физических величин, параметры реконструированных частиц и т.п.

В системе сбора данных доступ к событиям осуществляется с помощью средств MIDAS, либо с помощью разработанных на КМД-3 C++ интерфейсов. Для хранения обработанных данных на детекторе КМД-3 используется контейнер, именуемый CmdEvent, в котором данные хранятся в коллекциях, доступ к которым может быть осуществлен по их имени. Коллекции могут содержать объекты различных типов. Сохранение коллекций контейнера CmdEvent осуществляется в отдельную ветвь дерева ROOT [11].

Примеры коллекций в контейнере CmdEvent: “DCFittedTracks” – список реконструированных треков в дрейфовой камере, “CslRawCollection” – список сработавших элементов в Csl калориметре, и т.д.



Один файл захода содержащий 200 тысяч сырых событий обрабатывается процессом Cmd3fwk со скоростью ~50 событий в секунду, что обычно занимает около часа. Результатом обработки является ROOT файл, который в зависимости от использованных при обработке модулей, содержит либо коллекцию обработанных событий, либо набор гистограмм. При обработке большого количества заходов в распределенной компьютерной системе установки запускается множество, до нескольких сотен, процессов Cmd3fwk, что позволяет производить обработку с эффективной скоростью, на порядок и более превышающей скорость набора данных [14]. Однако такая скорость достигается только при обработке большого числа заходов, скорость обработки одного захода значительно уступает скорости набора данных.

### Оперативная обработка данных детектора КМД-3

Детектор КМД-3 представляет собой сложный аппаратно-программный комплекс. Ошибки в работе электроники и ПО снижают эффективность детектора и могут приводить к тому, что часть данных станет непригодной для дальнейшего использования. Детальная информация о качестве набранных данных становится доступна после полноценной обработки захода с помощью Cmd3fwk, что может занимать до двух часов от начала набора захода.

Чтобы улучшить обратную связь при диагностике и отладке детектора, необходимо сократить временную задержку между сбором данных и получением информации о их качестве. Для этого необходимо производить реконструкцию всего потока регистрируемых событий параллельно процессу сбора.

Текущее программное обеспечение позволяет выполнять реконструкцию до 50 событий в секунду, что недостаточно для обработки всего потока событий, генерируемых детектором, в режиме реального времени. Добиться необходимой скорости обработки потока данных можно путем увеличения объема обрабатываемых событий в секунду, распределяя этот поток между несколькими обрабатывающими процессами.

Цикл обработки сырого события разбивается на выполнение отдельных этапов: чтение, реконструкция, сбор статистики и запись гистограмм. Основная часть вычислительных ресурсов расходуется именно на реконструкцию событий, чтение/запись и сбор статистики происходит быстро. Обработка одного события не зависит от другого, поэтому каждое событие можно обрабатывать в отдельном процессе, а после обработки производить объединение статистики.

Таким образом, для увеличения максимального потока обрабатываемых событий до значений, получаемых при максимальной светимости комплекса, коллаборацией КМД-3 было принято решение о разработке распределённой системы обработки событий на базе приложения Cmd3fwk, что позволит получать доступ к статистике захода уже во время сбора данных.

### Распределенная обработка событий детектора КМД-3

С целью обработки всего потока данных детектора со скоростью регистрации, была реализована система распределённой обработки событий. Данная система позволяет выполнять обработку нескольких событий одновременно, и состоит из трех типов процессов Cmd3fwk: читающий, обрабатывающие и записывающий. Читающий процесс получает сырые события с помощью модуля чтения и распределяет их между обрабатывающими процессами. Обрабатывающие процессы выполняют обработку получаемых событий с помощью модулей реконструкции и отправляют обработанное событие записывающему процессу. Записывающий процесс, получив обработанное событие, выполняет набор статистики, строит гистограммы и сохраняет результаты обработки. Распределенная система обработки событий схематично изображена на рисунке 3.

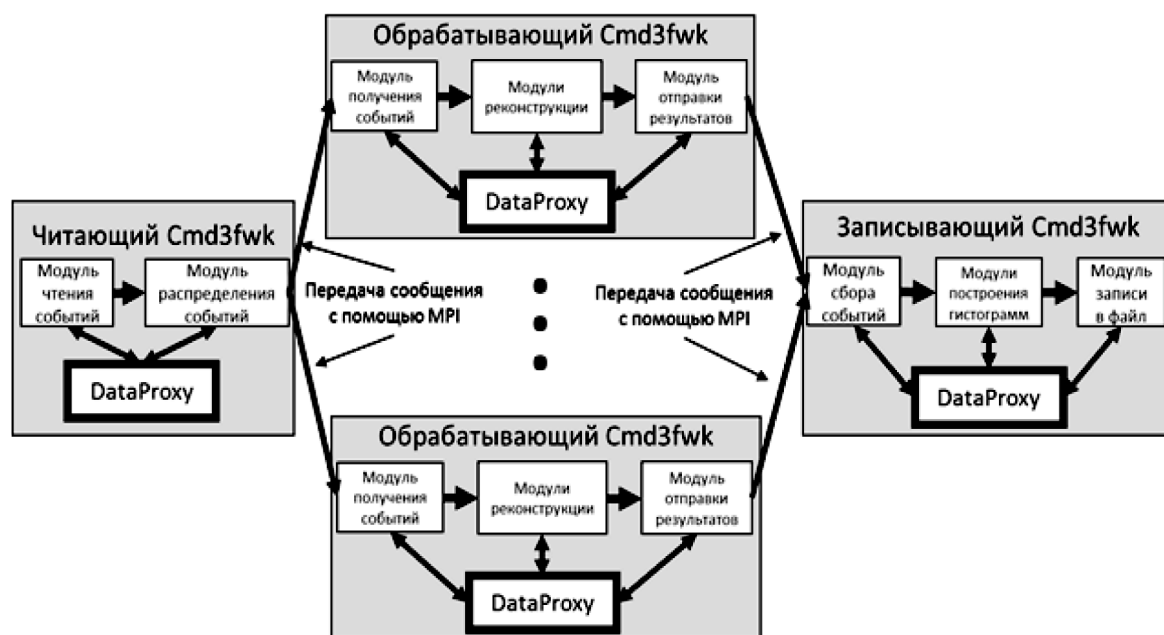


Рисунок 3 - Распределенная система обработки данных детектора КМД-3

Три типа процессов используют разные файлы конфигурации, содержащие список используемых модулей и порядок их выполнения. Читающему процессу достаточно использовать модуль чтения сырого события и модуль распределения событий. Обрабатывающий процесс включает в себя модули, выполняющие реконструкцию событий, модуль приема сырых событий и модуль отправки обработанных. Записывающий процесс использует модули для приема обработанных событий, набора гистограмм и записи в файл.

В качестве средства передачи сообщений между процессами, была использована открытая реализация программного интерфейса OpenMPI [12]. Стандарт MPI [13] позволяет

посылать синхронные и асинхронные сообщения, которые буферизуются в коммутаторе. Отправленные сообщения можно так же синхронно и асинхронно считывать и осуществлять фильтрацию их по тегу.

Для передачи событий между процессами Cmd3fwk, были разработаны дополнительные модули:

- *Модуль распределения событий* получает сырое событие из DataProxu, которое туда поместил модуль чтения, и затем сериализует данное событие в массив байтов. Модуль накапливает сериализованные события в буфере, пока не получит запрос на получение новых событий от одного из обрабатывающих процессов. В ответ на данный запрос модуль отправляет сообщение, в теле которого передается буфер с сериализованными событиями.
- *Модуль получения событий* после инициализации отправляет запрос для получения сырых событий читающему процессу. Ответом на этот запрос является сообщение, содержащее несколько сериализованных событий. После десериализации, событие помещается в DataProxu и обработка передается другим модулям. После обработки всех событий, переданных в сообщении, читающему процессу отправляется новый запрос на получение сырых событий, и так продолжается пока обработка не будет завершена.
- *Модуль отправки результатов* работает последним в цепочке обработки события, к моменту его вызова в DataProxu уже находится обработанное событие. Модуль получает обработанное событие из DataProxu и сериализует его в массив байтов. Чтобы снизить нагрузку на сеть, сообщение с обработанным событием сжимается перед отправкой пишущему процессу.
- *Модуль сбора событий* после инициализации ждет сообщений от обрабатывающих процессов. Получив сообщение, модуль восстанавливает обработанное событие и помещает его в DataProxu.

Чтение сырых событий из системы сбора данных, можно осуществить при помощи дополнительного модуля:

- *Модуль чтения событий* из MIDAS получает сырые события из MIDAS. При инициализации модуль подключается к службе MIDAS и подписывается на получение событий из буфера MIDAS. Далее модуль помещает в DataProxu полученные события.

Для запуска распределённой обработки получаемых событий с детектора, ресурсов одного компьютера недостаточно, поэтому для тестирования работы системы был использован вычислительный кластер, позволяющий использовать 90 реальных ядер и 90 виртуальных, 512Гб оперативной памяти и общее дисковое пространство в 10 Тб. Запуск и управление процессами участвующими в распределенной обработке на кластере, осуществляется с помощью `mpirun` [14].

Обработка поступающих данных с детектора, с помощью распределенной системы обработки, схематична изображена на Рисунке 4.



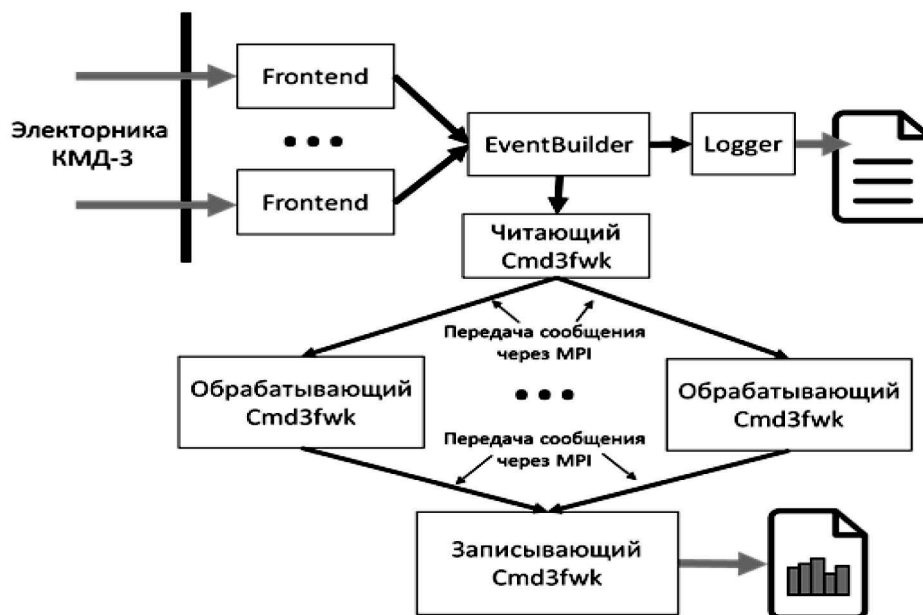


Рисунок 4 – Обработка данных детектора КМД-3 с помощью распределенной системы обработки событий

### Результаты

Реализованная система позволяет наращивать производительность за счет дополнительных процессов обработки событий. Скорость её работы была измерена на вычислительном кластере детектора. На рисунке 5, представлен график роста потока обрабатываемых событий в зависимости от количества обрабатывающих процессов.



Рисунок 5 – График роста потока обрабатываемых событий в зависимости от количества обрабатывающих процессов

Для обработки требуемого потока событий в 1,5кГц достаточно использовать 30-35 обрабатывающих процессов. В зависимости от типа событий, максимальная скорость обработки составляет  $1600 \pm 200$  событий в секунду. Узким местом стал модуль сбора событий, реализованная в нем десериализация данных позволяет восстанавливать до 1800 событий в секунду. Достигнутой скорости достаточно для использования системы на сегодняшний день. Планируется модернизировать алгоритмы сериализации и десериализации данных для дальнейшего повышения максимальной скорости обработки.

### Заключение

Не меняя общей архитектуры системы обработки данных детектора КМД-3, удалось экстенсивно нарастить ее производительность и организовать параллельную обработку отдельных событий. Увеличение скорости обработки потока событий позволяет избежать появления задержки в 1-2 часа между набором данных и получением детальной информации о данных. Разработанная система позволит организовать оперативный контроль качества собираемых детектором КМД-3 данных.

### Библиография :

1. Институт Ядерной Физики им. Г.И.Будкера. URL: <http://www.inp.nsk.su>.
2. Bennett G.W., Bousquet B., Brown H., Bunce G., Carey R., Cushman P., Danby G., Debevec P., Deile M., Deng H. Final report of the E821 muon anomalous magnetic moment measurement at BNL // Physical Review D. 2006. Т. 73. № 7. — С. 072003.
3. Berkaev D., Kirpotin A., Koop I., Lysenko A., Nesterenko I., Otboyev A., Perevedentsev E., Rogovsky Y., Romanov A., Shatunov P. VEPP-2000 operation with round beams in the energy range from 1 to 2 GeV // Nuclear Physics B-Proceedings Supplements. 2012. Т. 225. — С. 303-308.
4. Shatunov Y.M., Evstigneev A., Ganyushin D., Ivanov P., Koop I., Kuzminykh V., Lysenko A., Mezentsev N., Mityanina N., Nesterenko I. Project of a new electron-positron collider VEPP-2000. 2000. — 439-441.
5. Аульченко В., Аксенов В., Бесчастнов П. СНД–Сферический Нейтральный Детектор для ВЭПП–2М // Препринт ИЯФ. — С. 87-36.
6. Khazin B. Physics and Detectors for VEPP-2000 // Nuclear Physics B-Proceedings Supplements. 2008. Т. 181. — С. 376-380.
7. Gaiazov S., Banzarov V., Ignatov F., Logashenko I., Pirogov S., Sukharev A., Zaytsev A. Distributed data analysis system for CMD-3 detector // Journal of Instrumentation. 2014. Т. 9. № 08. — С. C08011.
8. Ritt S., Arnaudruz P., Olchanski K. MIDAS (Maximum Integration Data Acquisition System) // URL <https://midas.psi.ch>. 2001.
9. Hayes S., Cabrero K. Generalised and Advanced Urban Debiting Innovations: THE GAUDI PROJECT. 1. OVERVIEW // Traffic engineering & control. 1995. Т. 36. № 1.

10. Anisenkov A., Ignatov F., Pirogov S., Sibidanov A., Viduk S., Zaytsev A. CMD-3 detector offline software development. : IOP Publishing, 2010. — 032027.
11. Brun R., Rademakers F. ROOT—an object oriented data analysis framework // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 1997. T. 389. № 1. — С. 81-86.
12. Gabriel E., Fagg G.E., Bosilca G., Angskun T., Dongarra J.J., Squyres J.M., Sahay V., Kambadur P., Barrett B., Lumsdaine A. Open MPI: Goals, concept, and design of a next generation MPI implementation // Recent Advances in Parallel Virtual Machine and Message Passing Interface : Springer, 2004. — С. 97-104.
13. Gropp W., Lusk E., Doss N., Skjellum A. A high-performance, portable implementation of the MPI message passing interface standard // Parallel computing. 1996. T. 22. № 6. — С. 789-828.
14. Fineberg S. MPIRUN: A loader for multidisciplinary and multizonal MPI applications // NAS News. 1994. T. 2. № 6. — С. 102.

### References:

1. Institut Yadernoi Fiziki im. G.I.Budkera. URL: <http://www.inp.nsk.su>.
2. Bennett G.W., Bousquet B., Brown H., Bunce G., Carey R., Cushman P., Danby G., Debevec P., Deile M., Deng H. Final report of the E821 muon anomalous magnetic moment measurement at BNL // Physical Review D. 2006. T. 73. № 7. — С. 072003.
3. Berkaev D., Kirpotin A., Koop I., Lysenko A., Nesterenko I., Otboev A., Perevedentsev E., Rogovsky Y., Romanov A., Shatunov P. VEPP-2000 operation with round beams in the energy range from 1 to 2 GeV // Nuclear Physics B-Proceedings Supplements. 2012. T. 225. — С. 303-308.
4. Shatunov Y.M., Evstigneev A., Ganyushin D., Ivanov P., Koop I., Kuzminykh V., Lysenko A., Mezentsev N., Mityanina N., Nesterenko I. Project of a new electron-positron collider VEPP-2000. 2000. — 439-441.
5. Aul'chenko V., Aksenov V., Beschastnov P. SND—Sfericheskii Neitral'nyi Detektor dlya VEPP-2M // Preprint IYaF. — С. 87-36.
6. Khazin B. Physics and Detectors for VEPP-2000 // Nuclear Physics B-Proceedings Supplements. 2008. T. 181. — С. 376-380.
7. Gaiazov S., Banzarov V., Ignatov F., Logashenko I., Pirogov S., Sukharev A., Zaytsev A. Distributed data analysis system for CMD-3 detector // Journal of Instrumentation. 2014. T. 9. № 08. — С. C08011.
8. Ritt S., Arnaudruz P., Olchanski K. MIDAS (Maximum Integration Data Acquisition System) // URL <https://midas.psi.ch>. 2001.
9. Hayes S., Cabrero K. Generalised and Advanced Urban Debiting Innovations: THE GAUDI PROJECT. 1. OVERVIEW // Traffic engineering & control. 1995. T. 36. № 1.
10. Anisenkov A., Ignatov F., Pirogov S., Sibidanov A., Viduk S., Zaytsev A. CMD-3 detector offline software development. : IOP Publishing, 2010. — 032027.
11. Brun R., Rademakers F. ROOT—an object oriented data analysis framework // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 1997. T. 389. № 1. — С. 81-86.

12. Gabriel E., Fagg G.E., Bosilca G., Angskun T., Dongarra J.J., Squyres J.M., Sahay V., Kambadur P., Barrett B., Lumsdaine A. Open MPI: Goals, concept, and design of a next generation MPI implementation // Recent Advances in Parallel Virtual Machine and Message Passing Interface : Springer, 2004. — С. 97-104.
13. Gropp W., Lusk E., Doss N., Skjellum A. A high-performance, portable implementation of the MPI message passing interface standard // Parallel computing. 1996. Т. 22. № 6. — С. 789-828.
14. Fineberg S. MPIRUN: A loader for multidisciplinary and multizonal MPI applications // NAS News. 1994. Т. 2. № 6. — С. 102.