

## ИНЖЕНЕРНОЕ ПРОЕКТИРОВАНИЕ ЯДЕРНО-ФИЗИЧЕСКОЙ АППАРАТУРЫ

УДК 530.1

### МОДЕРНИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ КОНТРОЛЯ ДЕТЕКТОРА ATLAS

© 2025 г. И. А. Кочергин\*

Московский государственный университет имени М.В. Ломоносова, Москва, 119991 Россия

\*E-mail: Ilya.kochergin@cern.ch

Поступила в редакцию 15.10.2023 г.

После доработки 30.10.2023 г.

Принята к публикации 06.11.2023 г.

В экспериментах Большого адронного коллайдера, помимо получения данных о физических величинах, крайне важно получать всеобъемлющую информацию о техническом состоянии детекторов и систем для запуска, выключения и контроля всех экспериментов коллайдера. Эти задачи выполняются системами контроля детекторов. В данной статье описана система контроля детектора ATLAS, ее общая структура и взаимосвязь элементов. Представлено направление модернизации программного обеспечения системы. Описана модернизация одного из основных узлов передачи информации между базами данных.

*Ключевые слова:* детектор ATLAS, система контроля детектора, система визуализации и контроля процессов, SQL, Crest, COOL, CORAL

DOI: 10.56304/S2079562924050233

#### ВВЕДЕНИЕ

Эксперимент ATLAS Большого адронного коллайдера (БАК, LHC) является одним из самых крупных и технологичных в мире [1, 2]. Одноименный детектор соединяет множество элементов самого уровня и вида, от отдельных сенсоров пиксельных детекторов до сверхпроводящих магнитов и компьютеров системы обработки и анализа данных, объединенных в одну распределенную вычислительную сеть WLCG (Worldwide LHC Computing Grid) [3]. В детекторе более 100 млн. каналов считывания [4], и он состоит из 12 детекторных систем. Всеми системами необходимо управлять и контролировать их состояние, внося изменения в режим их работы. С этой целью разработана и используется система контроля детектора (detector control system, DCS) – сложный многоступенчатый объект.

Цель данной статьи – рассмотреть системы контроля детектора ATLAS, их общую схему и внутреннее устройство, их цель и способ работы. В первом разделе описываются вкратце подсистемы детектора ATLAS и рассматривается устройство используемой системы контроля детектора. Второй раздел посвящен программному обеспечению системы контроля, его структуре и внутренним связям. Последний третий раздел содержит описание модернизации программного обеспечения системы контроля детектора ATLAS.

#### 1. ДЕТЕКТОР ATLAS И СИСТЕМА КОНТРОЛЯ

Детектор ATLAS можно разделить на три глобальные структуры. Внутренний детектор (ID) состоит из кремниевых пиксельных (Pixel) и микростриповых координатных кремниевых детекторов (SCT) и трекового детектора переходного излучения (TRT) из тонких дрейфовых трубок (straw) и перекрывает область псевдобыстрот  $|\eta| < 2.5$ . Калориметрическая система включает электромагнитный жидкоаргоновый калориметр (LAr) высокой степени гранулярности и адронный тайл-калориметр из пластин скинтиллятора, перемежающихся со стальными пластинами (TILE), и перекрывает область  $|\eta| < 4.9$ . Мюонный спектрометр окружает калориметры и включает три больших воздушно-зазорных сверхпроводящих тороидальных магнита, мониторируемые дрейфовые камеры (MDT) и катодные стриповые камеры (CSC) для прецизионных измерений импульсов мюонов и быстродействующие триггерные камеры (RPC и TGC) для запуска системы регистрации событий.

Эксперимент ATLAS управляется двумя сотрудничающими параллельными системами. Система запуска и сбора данных (Trigger and Data Acquisition, TDAQ) отвечает за сбор, обработку и хранение данных эксперимента. Система управления детектором ATLAS была разработана и

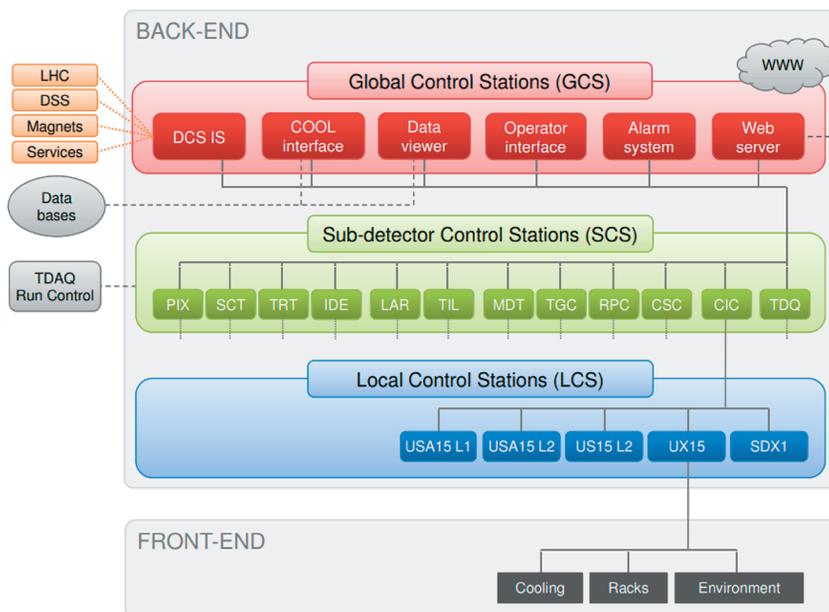


Рис. 1. Архитектура системы контроля детектора ATLAS [4].

внедрена при сотрудничестве соответствующей группы ЦЕРН (CERN) и групп контроля детекторов экспериментов LHC в рамках проекта Joint Controls Project (JCOP) [5]. Система контроля детектора DCS должна переводить детектор в любое требуемое рабочее состояние, постоянно контролировать и архивировать рабочие параметры, сигнализировать об отклонении параметров. Кроме того, ATLAS DCS должна служить единым интерфейсом для всех подсистем и технической инфраструктуры эксперимента. Она должна поддерживать связь с другими системами, которые управляются независимо, такими как ускоритель LHC, технические службы ЦЕРН, магниты ATLAS и система безопасности детекторов (DSS).

Архитектура систем контроля приведена на схеме рис. 1. Она имеет два основных слоя – внешний интерфейс (Front-end, FE) и серверная часть (Back-end, BE). Взаимодействие между слоями и между их элементами происходит с помощью шины CAN (Controller Area Network), в том числе с использованием интернета. Система FE включает в себя аппаратное обеспечение системы контроля, то есть датчики окружения, системы охлаждения и источники питания.

Слой BE включает в себя серверную часть и программное обеспечение, используемое для интеграции внешних элементов управления. В нем можно выделить три уровня: локальные станции управления (LCS), размещенные в разных центрах сбора данных ATLAS, расположение показано на рис. 2; станции управления подсистемами детекторов (SCS) для их высокоуровневого автономного управления; глобальные станции управ-

ления (GCS) с интерфейсом пользователя в пункте управления ATLAS (ATLAS Control Room).

В качестве универсального устройства ввода-вывода и получения информации от датчиков была разработана плата мониторинга (ELMB, Embedded Local Monitor Board). Она может быть встроена в пользовательские конструкции и имеет модульную, дистанционно расширяемую прошивку. В эксперименте ATLAS используется более 5 тысяч таких устройств.

На рис. 3 показана цепочка считывания на основе ELMB. Платы собирают данные с датчиков во всем объеме детектора, их можно настроить для передачи калиброванных данных датчика либо через регулярные промежутки времени, либо при изменении считываемых параметров, благодаря чему достигается первый уровень сокращения данных. Все ELMB связаны с соответствующей локальной станцией сбора данных через шину CAN, обеспечивающей до 125 кбит/с. В общей сложности 63 ELMB могут быть последовательно подключены к одной шине CAN и получать энергию от специальных источников питания, позволяющих обнаружить эффекты старения ELMB из-за излучения [6].

Для управления источниками данных, подключенных к шине CAN, разработан сервер CANopen. Данные предоставляются системе визуализации и контроля процессов PVSS (Prozess Visualisierung und Steuerungs System) [7] с использованием интерфейса OPC (Open Platform Communications). Управление системами контроля детектора осуществляется через промышленные компьютерные станции [8].

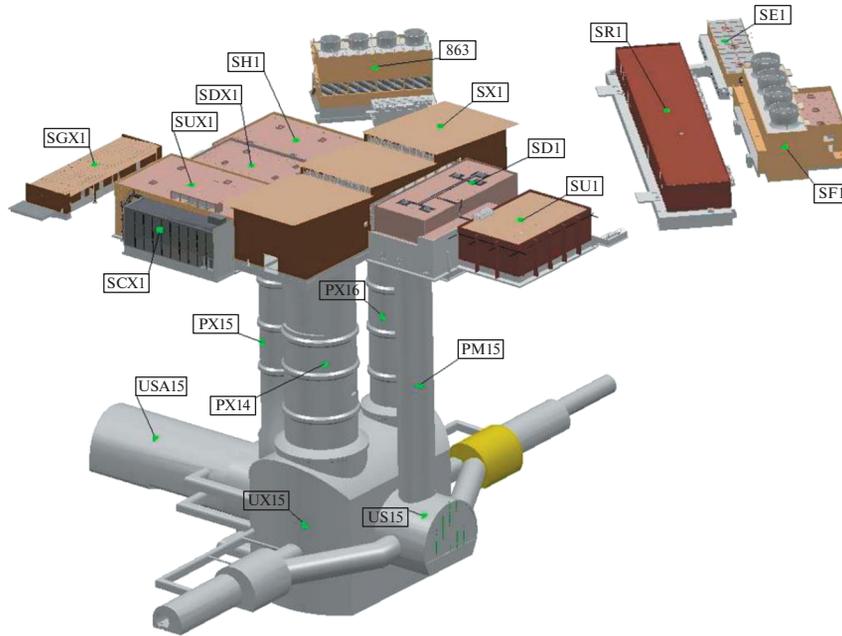


Рис. 2. Пространственная схема установки ATLAS вместе с помещениями сбора данных и пунктов управления [1].

## 2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ КОНТРОЛЯ

Для эффективного управления всеми частями системы контроля DCS используются как коммерческие программные продукты, так и специфический софт, разработанный под конкретные задачи. Так, для хранения информации используются базы данных SQL, для создания и редактирования которых используется стандартный Oracle интерфейс.

Для выборки информации и перезаписи данных из одной базы в другую может использоваться уникальный программный продукт, такой как PVSS2COOL. Эта программа извлекает запрашиваемую информацию о контрольных параметрах выбранной системы детектора за определенный временной период и записывает ее в соответствующую папку базы данных системы COOL. Система COOL содержит набор 32 схем хранения данных, API интерфейс для взаимодействия с этими данными, более десяти тысяч папок с соответствующими им базовыми реляционными таблицами [9]. Преимуществом всей системы является удобство ее использования пользователем: клиент может управлять своими собственными структурами данных, а также данными, которые эти структуры содержат.

Система визуализации и контроля процессов PVSS представляет из себя коммерческий пакет SCADA (Supervisory Controls And Data Acquisition – промышленная система диспетчерского управления и сбора данных) и является платформой для приложений BE ATLAS (рис. 4). Поскольку си-

стемам контроля необходимо считывать и обрабатывать данные с более чем 200 тысяч датчиков, число которых растет со временем, было принято решение с достаточной масштабируемостью по отношению к вычислительным мощностям.

Работа PVSS осуществляется по следующему принципу: станция управления запускает проект

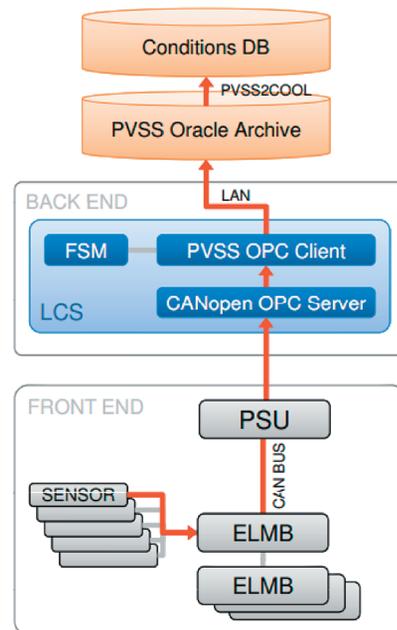


Рис. 3. Цепочка считывания данных на основе ELMB [8].

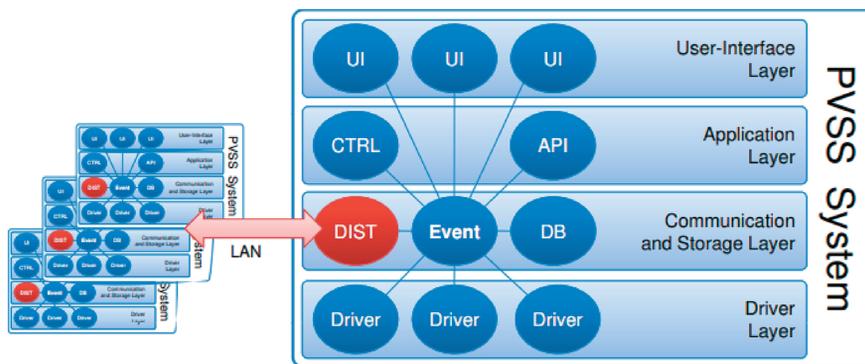


Рис. 4. Схема PVSS [8].

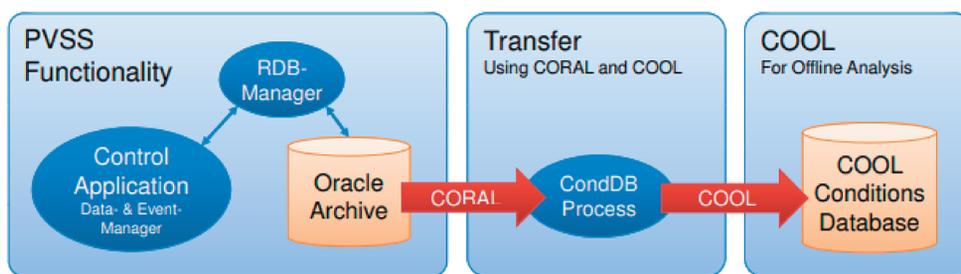


Рис. 5. Схема цепочки потока данных до CondDB [8].

“Project”, который содержит набор процессов “Managers”, использование и тип которых может различаться. Каждый проект использует центральную базу данных для всех текущих значений, хранящихся в объектах “Datapoints”, которые содержат данные в реальном времени от устройств FE. Обработка данных выполняется в событийно-ориентированном подходе с использованием многопоточных процедур обратного вызова при изменении значения параметров. Различные проекты могут быть связаны через локальную сеть для формирования распределенной системы, дающей удаленный доступ к базам данных и событиям всех подключенных проектов. Это обеспечивает масштабируемость до полного размера DCSATLAS в 150 станций управления. Общий интерфейс прикладного программирования (API) позволяет расширить функциональность управляющих приложений с помощью дополнительных программных компонентов, в том числе с использованием языка программирования C++ и соответствующих библиотек.

Базы данных SQL можно разделить на две категории. Конфигурационные базы данных (ConfDB) – спроектированы для управления настройкой детектора и параметров DCS. Базы данных состояний (CondDB) – хранят данные о состояниях детектора ATLAS с использованием реляционной базы данных для возможности

эффективного доступа к ним любых приложений через интерфейс COOL API.

На рис. 5 представлена схема потока данных базы состояний из системы PVSS в систему COOL. Для извлечения данных из Oracle архива используется интерфейс CORAL (COmmon Relational Abstraction Layer) [10]. Он предоставляет функции доступа к данным в реляционных базах с использованием API без C++ и SQL, защищая пользователя от специфичных API-интерфейсов.

Общую схему архитектуры для данных состояний, которая использовалась в периоды работы коллайдера Run 1 можно видеть на рис. 6а.

Минимальный набор таблиц, используемый в качестве основы для архитектуры системы в этапе Run 2, показан на рис. 6б. Данные полезной нагрузки хранятся в одной таблице с одним столбцом полезной нагрузки. Другие таблицы содержат информацию метаданных для тегов, IOV (interval of validity, срок действия) и глобальных тегов [11].

Визуализация параметров детекторов, и в целом данных систем контроля, осуществляется с помощью PVSS, также возможно использовать программный пакет ROOT [12]. Более подробно требования к программному обеспечению, причины его выбора, особенности программ и их взаимосвязь, возможности внешнего доступа и контроля описаны в [8].

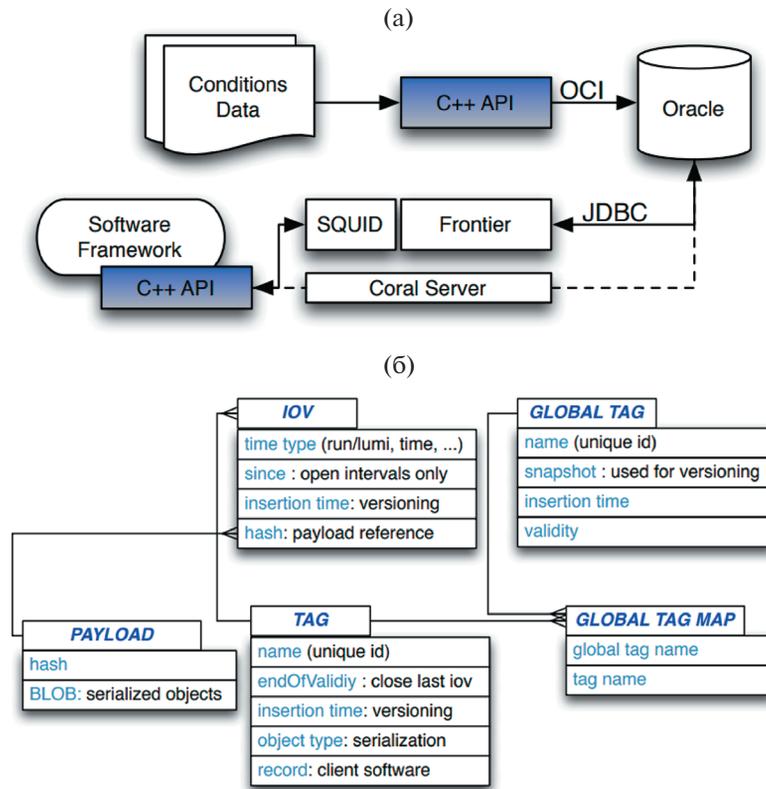


Рис. 6. Устройство архитектуры для данных состояний в Run 1 (а) и таблица структур данных условий в Run 2 (б) [13].

### 3. МОДЕРНИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Большая часть компонент программного обеспечения была создана и приспособлена для применения еще до начала этапа Run 1, то есть до 2008 г. Реализованы программы с использованием в основном языка С и С++, которые дают возможность использовать большое количество дополнительных библиотек и создавать при необходимости новые, а также с использованием Python и, в значительно меньшей степени, иных для особых задач. Впоследствии в процессе использования они улучшались, модифицировались и оптимизировались алгоритмы и методы отбора. В настоящее время идет этап принципиальной модернизации части программного обеспечения с переходом на новые технологии и возможности. Одна из таких модернизаций — использование вместо COOL интерфейса распределенной системы Crest, использующей хранение на серверах. Мотивацией для замены используемого API являются следующие требования [12]:

1. Возможность управления тегами и глобальными тегами: некоторые функции не были разработаны в COOL API и добавляются поверх с помощью Python.

2. Упрощение кэширования: загрузка данных условий при стандартной обработке данных требует особого внимания к кэшируемости запросов. В COOL API полезные данные загружаются одновременно с IOV, что усложняет процесс кэширования.

3. Усовершенствование структуры баз данных: возможность создавать таблицы непосредственно из COOL API оказывает большое влияние на количество таблиц, хранящихся в схемах Oracle. Каждый раз, когда системе приходится изменять свой формат, создается новый набор таблиц.

4. Долгосрочное обслуживание и развитие: COOL API (а также CORAL) в будущем не будет поддерживаться ИТ-специалистами.

В основе Crest системы лежит архитектурный стиль взаимодействия компонентов распределенного приложения в сети REST — Representational State Transfer. Архитектурным стилем называется набор согласованных ограничений и принципов проектирования, позволяющий добиться определенных свойств системы. Стиль REST основан на шести принципах:

1. Клиент-серверная архитектура: разделение зон ответственности. Система разделяется так, что клиент реализует только функциональное взаимодействие с сервером, а сервер уже реализует в себе

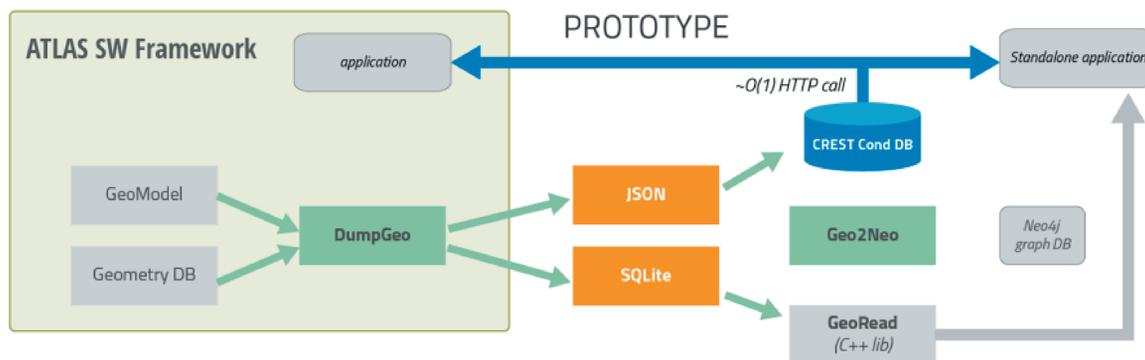


Рис. 7. Новая архитектура данных [14].

логику хранения данных, сложные взаимодействия со смежными системами и т.д. Основные плюсы — позволяет добиться масштабируемости и обеспечивает простоту поддержки.

2. Уменьшение информации: сервер не должен хранить у себя информацию о сессии с клиентом, а должен в каждом запросе получать всю информацию для обработки. Основные плюсы — уменьшение времени обработки запроса, упрощение поддержки, возможность кэширования, и также масштабируемость.

3. Кэширование: сохранение некоторых данных локально для более быстрого доступа при повторных запросах. Позволяет уменьшить количество сетевых взаимодействий и снизить нагрузки на всю систему.

4. Единообразие интерфейса: сервер возвращает не только ресурс, но и его связи с другими ресурсами. То есть в ответе содержится не только информация об объекте, но и все возможные действия с ним.

5. Слоистая архитектура: ни клиент, ни сервер не должны знать о том, как происходит цепочка вызовов дальше своих прямых соседей. Это позволяет изменять общую архитектуру без доработок на стороне клиента или сервера.

6. Код по требованию: идея передачи некоторого исполняемого кода от сервера клиенту.

Для реализации REST архитектуры используется HTTP RESTfull API. Вид реализуемой архитектуры данных представлен на рис. 7.

Как видно из рис. 7, теперь будет возможно выгрузить полное описание детектора из памяти в новые структуры данных для любого заданного тега геометрии. Сохранение полного описания геометрии для каждого тега приводит к избыточной информации, но это дает возможность иметь постоянную копию версии геометрии, которую

можно сохранять, совместно использовать, обслуживать, исследовать и отлаживать без доступа к платформе эксперимента. Это особенно полезно для заданий по обработке данных, которым необходимо выполнять только операции чтения.

Модель данных в Crest состоит из пяти таблиц, содержащих метаданные и данные полезной нагрузки, первоначально модель вдохновлена базой данных условий эксперимента CMS. Данные условий хранятся в таблице PAYLOAD. Значения используются в виде набора, метаданные условий разделены на три части. Параметр IOV в Crest содержит информацию о периоде времени для одного действительного значения характеристики, по умолчанию действует до следующего ввода времени. Метка TAG в Crest используется для идентификации определенного набора IOV. Метка GLOBAL TAG используется для идентификации согласованного набора меток TAG, участвующих в данном потоке данных.

Параметр IOV извлекаются отдельно от PAYLOADS, что отличается от COOL, который загружает все (IOV + PAYLOAD) одновременно. Это дает несколько преимуществ: во-первых, доступ к PAYLOADS можно кэшировать, чтобы он стал быстрее, но также можно очень быстро проверить IOV, поскольку они де-факто являются метаданными PAYLOADS.

Таким образом, заметно упрощается и ускоряется работа с данными по сравнению с использованием системы и интерфейса COOL. В рамках перехода на Crest должны быть модернизированы или заменены ряд программных инструментов по работе с данными PVSS и интерфейса CORAL. Среди таких инструментов — PVSS2COOL, который собирает информацию из баз данных PVSS о состоянии выбранной системы детектора за указанный период и записывает эти данные в COOL, создавая новую таблицу. Новый же инструмент

PVSS2CREST должен аналогично записывать выбранные данные за указанный период времени в json-файл и передавать его на соответствующий сервер системы Crest. За счет модульности программ и языка C++, возможно использовать существующие библиотеки для создания файлов с записанной информацией в формате JSON, а также для взаимодействия с CORAL и Crest. Интерфейс для последней все еще находится в стадии разработки, улучшений и согласований, но уже позволяет выполнять основные действия [15]. Формат JSON [16] – текстовый формат обмена данными, основанный на JavaScript, но независим от него – является широко используемым и стандартным для REST API, немного весит и довольно легко читается человеком. Однако возникают сложности при модернизации PVSS2COOL, и других инструментов, из-за необходимости согласования записываемых и передаваемых данных, форматов, а также необходимости абсолютной уверенности в считывании и передаче точно и исключительно указанных данных за выбранный период в соответствии с установленным IOV.

В результате перехода на использование Crest будет достигнуто значительное упрощение использования данных систем контроля детектора в физических анализах и методических работах, увеличение возможностей и объема для хранения информации и оптимизация ее организации в памяти, что в сумме сильно увеличит потенциальное количество хранимой и обрабатываемой информации и ее доступность.

## ЗАКЛЮЧЕНИЕ

Система контроля эксперимента ATLAS обеспечивает успешное функционирование детектора и системы анализа данных в течение 15 лет. За этот время существенно изменились условия работы БАК: энергия соударения протонов практически достигла плановой и составляет 13.6 ТэВ, светимость вдвое превышает плановую  $10^{34} \text{ см}^2 \text{ с}^{-1}$  и еще возрастет до величины, в 5–7 раз больше плановой. Изменился детектор ATLAS: добавлен слой пиксельных детекторов (B-layer) [17], система новых мюонных модулей (New Small Wheels, NSW), трехуровневая структура триггера преобразована в двухуровневую и др. Все это время система DCS ATLAS обеспечивала управление детектором и важнейшими процессами его работы, связанными с безопасностью эксплуатации установки ATLAS.

Система контроля эксперимента ATLAS имеет сложную многоуровневую структуру, объединяющую подсистемы детекторов установки, систему триггера и сбора физической информации с детекторов и обеспечивающую синхронизацию с

системами БАК, другими экспериментами БАК и всего ЦЕРН и распределенной вычислительной системой WLCG.

С развитием экспериментальной установки ATLAS, программных и вычислительных ресурсов, в том числе коммерческого производства, развивается и модернизируется система контроля DCS ATLAS. Важным направлением модернизации является обновление систем работы с базами данных с целью ускорения и облегчения работы с интерфейсами системы. В работе описана одна из имплементируемых систем для решения этих задач.

Показано, что для взаимодействия с большими базами данных, с которыми работает эксперимент ATLAS, актуально применение системы Crest. Ее реализация обеспечивает упрощение обработки данных систем контроля детектора, увеличение объема хранимой информации и ее доступность.

## СПИСОК ЛИТЕРАТУРЫ/REFERENCES

1. *ATLAS Collab.* // J. Instrum. 2008. V. 3. P. S08003.
2. *ATLAS Collab.* // ATL-SOFT-PUB-2021-001. 2021.
3. *ATLAS Collab.* // ATLAS-TDR-17, CERN-LHCC-2005-022. 2025. CERN.
4. *Lantzsch K., Arfaoui S., Franz S., et al.* // J. Phys.: Conf. Ser. 2012. V. 396. P. 012028.
5. *Holme O. et al.* // Proc. 10th Int. Conf. Accelerator and Large Experimental Physics Control Systems ICALEPCS-2005 (2005). C051010:WE2.1-60.
6. *Hallgren B. et al.* // Proc. Conf. Electronics for LHC Experiments, Stockholm. 2001. P. 325–330.
7. <https://www.winccoa.com/company.html>.
8. *Barriuso Poy A. et al.* // J. Instrum. 2008. V. 3. P. P05006.
9. *Hawkings R., Viegas F., Dimitrov G.* // J. Phys.: Conf. Ser. 2008. V. 119. P. 042032.
10. <https://twiki.cern.ch/twiki/bin/view/Persistency/Coral>.
11. *Gallas E.J., Albrand S., Borodin M., Formica A.* // J. Phys.: Conf. Ser. 2014. V. 513. P. 042020. <http://iop-science.iop.org/1742-6596/513/4/042020>.
12. *Brun R., Rademakers F.* // Nucl. Instrum. Methods Phys. Res., Sect. A. 1997. V. 389 (1–2). P. 81–86.
13. *Barberis D., Formica A., Gallas E.J., Govi G., Lehman Mitto G., Pfeiffer A. (ATLAS Collab.)* // J. Phys.: Conf. Ser. 2015. V. 664. P. 042015.
14. *Formica A., Bianchi R.M., De Salvo A. (ATLAS Collab.)* // ATL-SOFT-PROC-2018-056. 2018.
15. *Alexandrov E., Formica A., Mineev M., Roe S.* // Proc. 9th Int. Conf. Distributed Computing and Grid Technologies in Science and Education GRID'2021. July 5–9, 2021. Dubna, Russia. 2021.
16. <https://json.nlohmann.me/>.
17. *ATLAS Collab.* // CERN-LHCC-2010-013, ATLAS-TDR-19.

## Upgrading the ATLAS Detector Control System Software

I. A. Kochergin\*

*Lomonosov Moscow State University, Moscow, 119991 Russia*

*\*e-mail: Ilya.kochergin@cern.ch*

Received October 15, 2023; revised October 30, 2023; accepted November 6, 2023

**Abstract**—In Large Hadron Collider experiments, in addition to obtaining data on physical quantities, it is extremely important to obtain comprehensive information about the technical condition of detectors and systems for starting, shutting down and monitoring all collider experiments. These tasks are performed by detector control systems. This article describes the ATLAS detector control system, its general structure and the interconnection of elements. The direction of modernization of the system software is presented. The modernization of one of the main nodes for transferring information between databases is described.

*Keywords:* detector ATLAS, detector control systems, PVSS, SQL, Crest, COOL, CORAL