

НОВЫЙ ДИЗАЙН ИНСТРУМЕНТОВ ДЛЯ ДОСТУПА К БАЗЕ ДАННЫХ УСЛОВИЙ ATLAS CREST В Athena

Е. И. Александров^{1,2,}, М. А. Минеев¹*

¹ Объединенный институт ядерных исследований, Дубна, Россия

² Государственный университет «Дубна», Дубна, Россия

В ходе разработки прототипа CREST в Athena был интегрирован новый инструмент для взаимодействия с базой данных условий. Изначально этот инструмент был основан на существующей реализации COOL, что позволяло быстро тестировать новую базу данных в производственных рабочих процессах. Однако из-за сложностей обслуживания и ограниченного использования инструментом специфических для CREST функций было принято решение о его переработке. Описывается новый дизайн для доступа к данным CREST в Athena. Обновленный инструментарий упрощает обслуживание, объединяет многочисленные методы обработки метаданных в один класс и добавляет класс для сериализации и десериализации данных CREST. Такой подход обеспечивает гибкую обработку различных форматов хранения данных в CREST.

During the development of the CREST prototype, a new tool for interacting with the conditions database was integrated into Athena. Initially, this tool was based on the existing COOL implementation, enabling rapid testing of the new database in production workflows. However, due to maintenance challenges and the tool's limited accommodation of CREST-specific features, a decision was made to redesign it. The new design for accessing CREST data from Athena is described. The redesigned toolkit simplifies maintenance, consolidates numerous metadata handling methods into a single class, and introduces a class for serializing and deserializing CREST data. This approach supports flexible handling of various data storage formats in CREST.

PACS: 89.20.Ff

ВВЕДЕНИЕ

ATLAS [1] — один из крупнейших экспериментов на Большом адронном коллайдере (БАК) в ЦЕРН, и обработка данных событий требует извлечения большого количества вспомогательных данных, не связанных напрямую с событиями, например, юстировка и калибровка детектора. Эти данные изменяются с течением времени и называются данными об условиях. Обычно они хранятся в базе данных, называемой базой

* E-mail: aleksand@jinr.ru

данных условий. Для экспериментов БАК в ЦЕРН был разработан специализированный комплекс интерфейсов (COOL, POOL, CORAL) [2] для упрощения работы с существующими базами данных, который стал использоваться также и в других экспериментах на БАК. POOL — это универсальное гибридное хранилище C++ объектов, каталогов метаданных и коллекций, использующее потоковые и реляционные технологии. CORAL — это универсальный уровень абстракции с API без SQL для доступа к реляционным базам данных. COOL — это специальное программное обеспечение для чтения и записи данных условий. На рис. 1 представлена схема взаимодействия физических программ и баз данных с использованием промежуточного программного обеспечения.

Начиная с Run 3, ATLAS был единственным экспериментом, в котором использовался COOL. Со временем ряд ограничений, связанных с возможностями кэширования API COOL, стал вызывать проблемы. В результате коллаборация ATLAS решила перейти на новую базу данных условий, используя модель данных, аналогичную той, что была разработана в эксперименте CMS [3] для Run 2. Для этой цели в рамках проекта CREST [4, 5] была начата разработка новой системы. Прототип CREST успешно прошел испытания в 2024 г., и начались исследования кэширования данных. При создании прототипа задача разработки унифицированного подхода к интеграции CREST в другие приложения была отложена. Поэтому в большинстве случаев для интеграции использовался подход, схожий с COOL. Это позволило ускорить создание полноценного прототипа. Данная работа посвящена описанию нового дизайна для интеграции приложений с базой данных CREST.

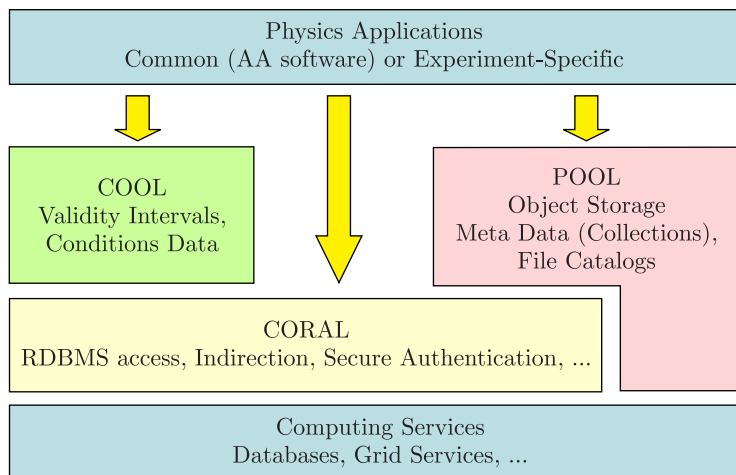


Рис. 1. Схема взаимодействия физических программ с базами данных с использованием промежуточного программного обеспечения

ИСПОЛЬЗОВАНИЕ ДАННЫХ CREST В ЭКСПЕРИМЕНТЕ ATLAS

В эксперименте ATLAS используется программная платформа Athena [6] для управления практически всеми рабочими процессами обработки физических данных. Для работы с данными об условиях в Athena используется CORAL, но непосредственно за соединение с базой данных состояний отвечает COOL API. Доступ к базам данных условий реализован в пакете IOVDbSvc.

Для полноценного тестирования, кроме самой интерпретации Athena с CREST, нужно сконвертировать необходимые данные из COOL в CREST. Для этого были разработаны соответствующие приложения: Cool2Crest — для копирования данных из COOL в CREST и Pvss2Crest — для копирования Detector Control System (DCS) данных в CREST. Особенности данных DCS будут описаны в следующем разделе.

Для базы данных CREST реализован компонент CrestApi, который на языках C++ и Python предоставляет методы для отправки и получения данных от клиента к серверу. На рис. 2 представлена первоначальная схема взаимодействия с данными CREST в клиентах. У данной схемы есть несколько слабых мест. Нет единой сериализации и десериализации, что приводит к дублированию части кода и возможным ошибкам, когда изменения вносятся несинхронно. В Athena разбор данных и различных метаданных от базы данных условий реализован в разных классах. Это усложняет понимание кода и дублирует хранение некоторых данных. Данная реализация не предполагает различные форматы данных, что не позволяет полноценно интегрировать поддержку DCS-формата.

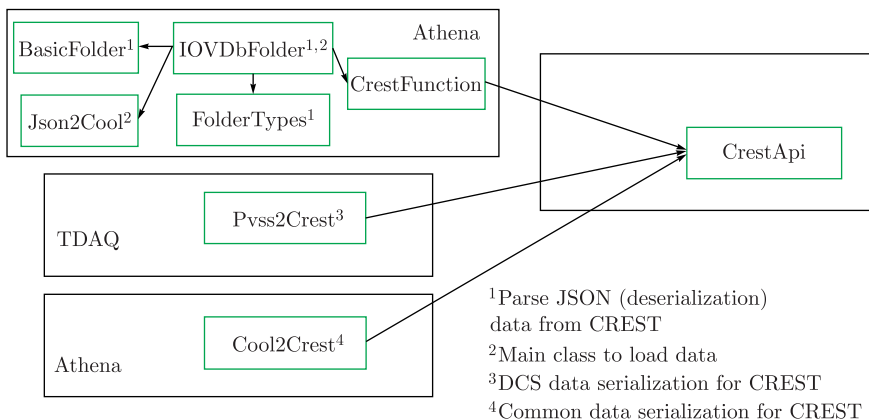


Рис. 2. Схема взаимодействия с данными CREST в клиентах

ФОРМАТ DCS

В CREST данные разбиты по тэгам (в COOL это папки), а в самих тэгах — по каналам. COOL же хранит данные только по каналам, и эти данные независимы. В CREST данные хранятся так, как они используются пользователем, т. е. данные из всех каналов хранятся в одном месте для каждого момента времени. Особенности DCS-данных — это большое количество каналов и часто меняющиеся данные. Причем в один момент меняются данные одного, редко двух каналов. Это приводит к огромному количеству временных точек. Поэтому использование обычного формата хранения данных CREST нерационально. Для хранения таких данных был разработан специальный формат DCS. Он заключается в том, что в каждой временной точке хранятся данные за определенный период времени, например, за 10 мин. Это дает возможность загружать за один запрос данные за определенный период. На сегодня все данные со всех каналов за выбранный период инкапсулируются в одну временную метку, но планируется модернизация формата с уменьшением объема данных.

НОВЫЙ ДИЗАЙН ИНСТРУМЕНТОВ ДЛЯ ДОСТУПА К БАЗЕ ДАННЫХ УСЛОВИЙ ATLAS CREST

На рис.3 представлена новая схема взаимодействия с данными CREST в клиентах. Структура была упрощена и унифицирована. Сериализация и десериализация данных теперь осуществляется в одном классе с именем *CrestContainer*. Он хранит данные в формате JSON, которые может как преобразовать в/из заданный CREST-формат, так и конвертировать данные в Coral. JSON позволяет хранить все существующие

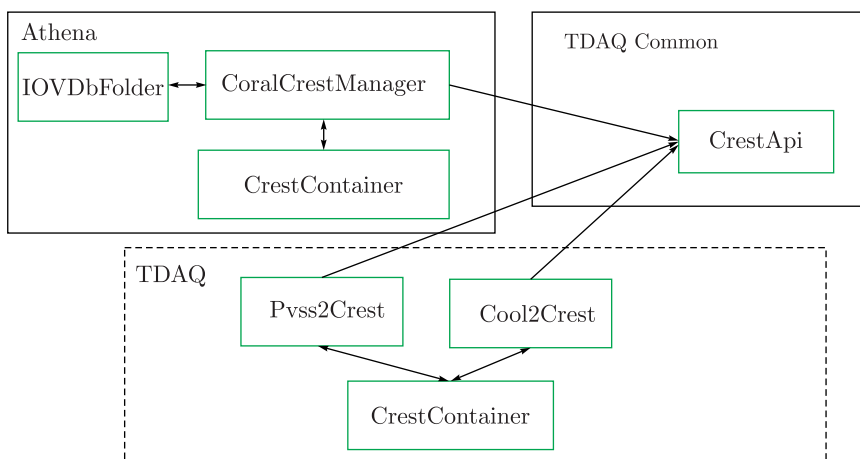


Рис. 3. Новая схема взаимодействия с данными CREST в клиентах

форматы данных COOL. Для копирования данных используется тот же самый класс. В будущем планируется перенос данного класса в пакет CrestApi, который в конце 2024 г. вошел в состав TDAQ Common. Разбор всех метаданных был перенесен в отдельный класс CoralCrestManager.

Были проведены следующие тесты: задачи Athena для обработки HLT (High Level Trigger) в P1, сервера CREST и связанной с ним инфраструктуры кэширования (с использованием Varnish), задачи Athena, использующей данные формата DCS. Реализовано приложение для сравнения папок по выбранной временной метке. Все тесты показали корректные результаты. В середине 2025 г. реализация новой архитектуры была включена в состав Athena.

ЗАКЛЮЧЕНИЕ

Первоначально основные усилия были направлены на разработку сервера новой базы данных условий CREST и реализацию методов передачи данных на него. Из-за этого интеграция в клиентские приложения была выполнена неоптимально. После создания прототипа CREST была разработана схема взаимодействия с данными CREST для приложений. Реализация данной схемы позволила оптимизировать код клиентских приложений, в частности модуль в Athena (IOVDbSvc), отвечающий за связь с базой данных условий. Были произведены тесты, которые показали работоспособность CREST в новой архитектуре взаимодействия с данными.

СПИСОК ЛИТЕРАТУРЫ

1. *ATLAS Collab.* The ATLAS Experiment at the CERN Large Hadron Collider // J. Instrum. 2008. V. 3. S08003.
2. *Trentadue R. et al.* LCG Persistency Framework (CORAL, COOL, POOL): Status and Outlook in 2012 // J. Phys.: Conf. Ser. 2012. V. 396. 053067.
3. *CMS Collab.* The CMS Experiment at the CERN LHC // J. Instrum. 2008. V. 3.
4. *Alexandrov E., Formica A., Mineev M. et al.* Evolution of the ATLAS CREST Conditions DB Project // Phys. Part. Nucl. 2024. V. 55. P. 441–443.
5. *Alexandrov E., Formica A., Mineev M.* Towards a New Conditions Data Infrastructure in ATLAS // Eur. Phys. J. Web Conf. 2024. V. 295. 01013.
6. Athena. <https://atlas-software.docs.cern.ch/athena/basics/intro/> (accessed 10.09.2025).