

МЕТОДЫ АВТОМАТИЗАЦИИ ОБРАБОТКИ ОШИБОК В EVENT PICKING SERVICE ЭКСПЕРИМЕНТА ATLAS

*Е. И. Александров^{1,2}, И. Н. Александров^{1,2},
Д. Барберис^{3,4}, А. В. Яковлев^{1,*}*

¹ Объединенный институт ядерных исследований, Дубна, Россия

² Государственный университет «Дубна», Дубна, Россия

³ Калифорнийский университет, Беркли, США

⁴ Национальный институт ядерной физики, Генуя, Италия

ATLAS EventIndex — это полный каталог всех реальных и смоделированных событий ATLAS, включающий ссылки на все постоянные файлы, содержащие данное событие на любом этапе обработки. Event Picking Service (EPS) автоматизирует процесс поиска местоположений большого количества событий, их извлечения и сбора в отдельные файлы. Он поддерживает различные форматы событий и обладает гибким рабочим процессом для входных данных. Описываются типы ошибок, возникающих при выполнении рабочих процессов и потоков, а также методы автоматизации обработки таких ошибок для обеспечения стабильности и бесперебойности работы сервиса.

The ATLAS EventIndex is the complete catalogue of all ATLAS real and simulated events, keeping the references to all permanent files that contain a given event in any processing stage. The Event Picking Service (EPS) automates the procedure of finding the locations of large numbers of events, extracting and collecting them into separate files. It supports different formats of events and has an elastic workflow for different input data. This article describes the different types of errors that occur while performing different workflows and threads. It also describes methods for automating the processing of such errors to ensure the stability and uninterrupted operation of the service.

PACS: 89.20.Ff

ВВЕДЕНИЕ

Эксперимент ATLAS — один из четырех экспериментов на ускорителе LHC в ЦЕРН [1]. Ежегодно в эксперименте ATLAS производится несколько миллиардов записей событий в виде «сырых» данных и в других форматах. Проект EventIndex [2] каталогизирует все события ATLAS и предоставляет набор инструментов для поиска и извлечения

* E-mail: yakovleva@jinr.ru

информации об отдельных событиях или группах событий в соответствии с выбором пользователя. Основное назначение EventIndex — предоставить участникам ATLAS возможность искать и извлекать одно или несколько отдельных событий из десятков миллионов файлов данных для проведения детальных проверок или более точного анализа. Для каждого события, каждого формата данных и каждой версии обработки он содержит: идентификаторы событий (номер сеанса (run) и события), местоположение (GUID файла, содержащего событие) и происхождение; триггер и другие полезные метаданные.

Для некоторых физических анализов требуется извлечь множество событий, чтобы обработать их с помощью усовершенствованных алгоритмов.

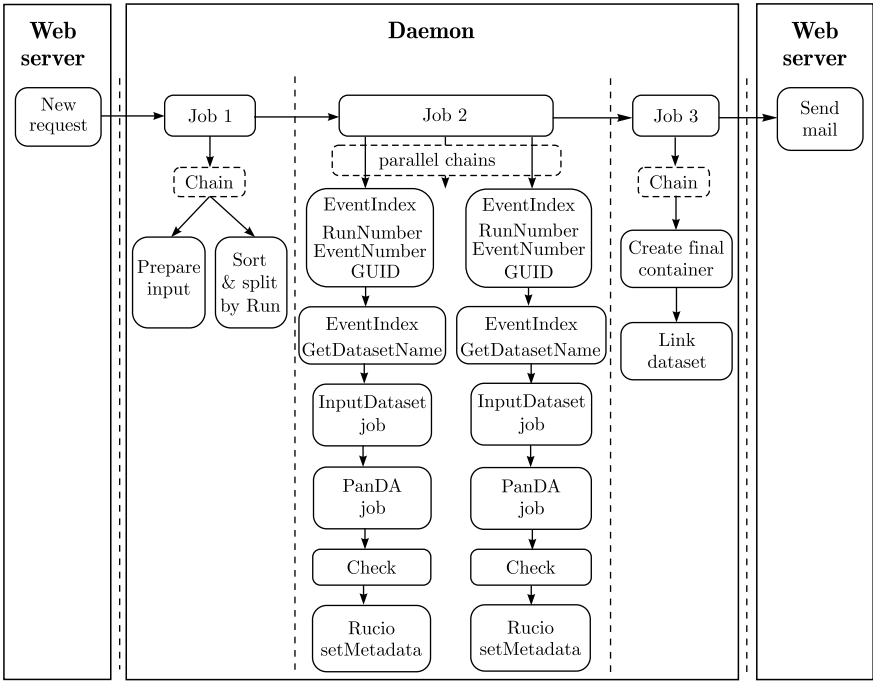
EventIndex может помочь в этом, но для этого требуется много времени работы в ручном режиме, поскольку приходится выполнять такие шаги, как поиск событий в EventIndex, отправка заданий на выбор событий PanDA [3], их мониторинг и повторные попытки при истечении времени ожидания, а также выполнять другие шаги. Для автоматического выполнения всех этих действий был разработан Event Picking Service (EPS).

АРХИТЕКТУРА EVENT PICKING SERVICE

Сервис состоит из двух частей: веб-сервера и демона, которые независимы и взаимодействуют через базу данных. Только веб-сервер взаимодействует с клиентами, а демон взаимодействует с защищенными системами (такими как PanDA и Rucio [4]). Клиент формирует запрос к серверу, используя форму, встроенную в HTML-страницу. Веб-сервер сохраняет новый запрос в базе данных. Демон отслеживает базу данных на предмет новых запросов. При обнаружении нового запроса он начинает выполнять задачи для этого запроса в соответствии с рабочим процессом (workflow), взятым из базы данных. После того как все задачи, соответствующие рабочему процессу, завершены или во время их выполнения получены критические ошибки, демон помечает запрос как завершенный в базе данных. Веб-сервер отслеживает в базе данных появление завершенных запросов и отправляет сообщение пользователю (или администратору) о результатах запроса. Более подробную информацию об архитектуре EPS можно найти в [5].

WORKFLOWS

EPS предназначен не для решения одной конкретной задачи, а для целого типа выбора событий. Для решения конкретной проблемы создаются цепочки задач. Цепочки могут быть разными для разных типов задач.



Рабочий процесс (workflow) в Event Picking Service

На рисунке представлен текущий общий рабочий процесс EPS. Первым шагом является сортировка и разделение событий по сеансам. Затем в параллельном режиме запускаются задачи для получения необходимых данных из EventIndex, PanDA, Rucio и проверки результатов.

Другой полезный рабочий процесс — перезапуск запроса, когда предыдущее задание по выбору событий вернуло ошибки для нескольких пар сеанс/событие, а цель состоит в том, чтобы завершить исходное задание, запустив исходный рабочий процесс, но только для событий, вызвавших ошибку. Одной из целей EPS была его гибкость, поэтому рабочие процессы гибкие и допускают различные изменения: добавление новых задач в цепочки, изменение порядка и т. д. Они определены в базе данных, поэтому можно обновить или добавить любую цепочку, не изменяя общую часть кода.

ОБРАБОТКА ОШИБОК В EVENT PICKING SERVICE

Для корректной работы EPS должен в определенной степени выполнять функции оркестратора. EPS должен создавать параллельные цепочки задач, корректно формируя входные данные. В рамках каждой

цепочки EPS должен корректно вызывать каждый внешний сервис, передавать ему входные данные, корректно обрабатывать полученный ответ и затем передавать его следующей задаче в цепочке.

Время от времени при выполнении различных рабочих процессов и потоков возникают ошибки. Для корректной работы EPS должен уметь выявлять различные ошибки и по возможности корректно их обрабатывать. Эти ошибки можно сгруппировать по «серьезности» их влияния на конечный результат и по методам обработки. Было реализовано несколько типовых методов обработки ошибок.

Автоматический перезапуск задач. Внешние сервисы могут не отвечать, или соединение с внешним сервисом может быть потеряно во время работы. Такие ошибки могут иметь фатальное влияние на результат. Однако их можно устранить автоматически или вручную. EPS имеет механизм автоматического перезапуска задач. Если внешний сервис не отвечает, то EPS ожидает некоторое время и пытается снова вызвать этот сервис. Если внешний сервис отвечает, то цепочка продолжает работать в обычном режиме. Ошибка устраняется автоматически без вмешательства администратора. В системе установлено заданное количество попыток автоматического перезапуска определенной задачи. При превышении этого количества EPS фиксирует эту ошибку и выполнение цепочки останавливается. Для устранения этой ошибки требуется вмешательство пользователя или администратора.

Перезапуск запроса вручную. Перезапуск запроса в EPS означает создание нового запроса с использованием входных данных родительского запроса. В веб-интерфейсе EPS реализовано несколько типов перезапуска запросов:

- «полный перезапуск»: запрос начинается с начала, результаты родительского запроса не используются в новом запросе;
- «перезапуск цепочек с предупреждениями и ошибками»: перезапускаются только цепочки со статусом «ошибка». Новый запрос будет содержать результаты обработки только для этих цепочек.

Специальный перезапуск для ошибок проверки данных (check errors). Данные, полученные от сервиса PanDA, следует проверить на наличие пропущенных или дублированных событий. Возможны случаи, когда файлы, возвращаемые PanDA, в настоящее время недоступны. В этом случае возникает ошибка и пользователю возвращается список пропущенных событий. Ошибка не является фатальной, данные уже получены, необходимо просто повторить этап проверки еще раз. Для этого случая реализован специальный тип перезапуска — «перезапуск только задач проверки». Запускаются только те цепочки, в которых обнаружены ошибки проверки, начиная только с задачи проверки. В этом случае данные PanDA берутся из родительского запроса.

Дополнительная доработка алгоритма. Внешний сервис возвращает устаревшие данные. В этом случае следующий по порядку в рабочем

процессе сервис не сможет использовать эту информацию. Для решения этой проблемы необходимо усовершенствовать алгоритм, например, внедрив дополнительные проверки релевантности полученных данных. Для исправления подобных ошибок разработчикам потребуется много времени и усилий, но в результате добавление новых методов позволит автоматически избегать их в будущем.

Рассылка сообщений об ошибках. При возникновении некоторых типов ошибок во внешних сервисах EPS отправляет электронное письмо об ошибке либо на официальный адрес такого сервиса (при наличии), либо непосредственно экспертам этих сервисов. Список экспертов формируется вручную администраторами EPS. При возникновении ошибки система записывает информацию о ней в специальную таблицу в базе данных. EPS периодически отслеживает эту таблицу. При возникновении ошибки веб-интерфейс отправляет сообщение списку экспертов в соответствии с типом ошибки.

ЗАКЛЮЧЕНИЕ

Представлены обоснование создания и общая архитектура сервиса EPS. Описаны рабочие процессы EPS. Показана важность автоматического восстановления после ошибок без вмешательства администратора. Также представлены усовершенствования системы автоматического восстановления после ошибок, удобный интерфейс для ручного перезапуска и доработка алгоритмов для повышения эффективности и предотвращения потенциальных ошибок. Гибкая архитектура и предлагаемые методы разрешения ошибок позволяют EPS успешно реализовывать свою основную задачу — массовый отбор событий для выбора интересующих событий из множества данных ATLAS.

СПИСОК ЛИТЕРАТУРЫ

1. *ATLAS Collab.* The ATLAS Experiment at the CERN Large Hadron Collider // J. Instrum. 2008. V. 3. S08003.
2. *Barberis D. et al.* The ATLAS EventIndex: A BigData Catalogue for All ATLAS Experiment Events // Comput. Softw. Big Sci. 2023. V. 7.
3. *Barreiro Megino F. et al.* ATLAS WORLD-Cloud and Networking in PanDA // J. Phys.: Conf. Ser. 2017. V. 898. 052011.
4. *Barisits M. et al.* Rucio: Scientific Data Management // Comput. Softw. Big Sci. 2019. V. 3, No. 11.
5. *Alexandrov E., Alexandrov I., Barberis D., Prokoshin F., Yakovlev A.* Development of the ATLAS Event Picking Server // CEUR Workshop Proc. 2021. V. 3041. P. 223–228.