ATL-SOFT-PROC-2023-032
19 September 2023

The ATLAS Event Picking Service and its evolution

E.I. Alexandrov[a,1], *I.N. Alexandrov*[a], *D. Barberis*[b], *L. Canali*[c], *E. Cherepanova*[d], *E.J. Gallas*[e], *S. Gonzalez de la Hoz*[f], *F.V. Prokoshin*[a], *G. Rybkin*[g], *J. Salt Cairols*[f], *J. Sanchez*[f], *M. Villaplana Perez*[f], *A.V. Yakovlev*[a]

[a] Joint Institute for Nuclear Research, Dubna, Russia

[b] Università di Genova and INFN Genova, Italy

[c] CERN, Geneva, Switzerland

[d] University of Amsterdam and NIKHEF, Amsterdam, Netherlands

[e] University of Oxford, Oxford, UK

[f] Instituto de Física Corpuscular IFIC, Valencia, Spain

[g] Université Paris-Saclay, CNRS/IN2P3, IJCLab, Orsay, France

The 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; its implementation has been substantially revised in advance of LHC Run 3 to be able to scale to the higher production rates. The Event Picking Server 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. The convenient graphical interface of the Event Picking Server is integrated with ATLAS SSO. The monitoring system controls the performance of all parts of the service.

EventIndex — это полный каталог всех реальных и смоделированных событий ATLAS, включающий ссылки на все постоянные файлы, содержащие данное событие на любом этапе обработки. Его реализация была существенно пересмотрена перед Run 3 LHC, для того, чтобы сохранять высокую производительность работы при увеличении потоков данных. Сервер сбора событий автоматизирует процедуру поиска набора событий, их извлечения и сбора в отдельные файлы. Он поддерживает различные форматы событий и имеет гибкий рабочий процесс для различных входных данных. Удобный графический интерфейс сервера выбора событий интегрирован с ATLAS SSO. Система мониторинга контролирует работу всех частей сервиса.

INTRODUCTION

Every year the ATLAS experiment [1] produces several billion event records in raw and other formats. The data are spread among hundreds of computing Grid sites around the world. The EventIndex system [2] catalogues all ATLAS events and provides a set of tools to search and retrieve information about single events or on event groups, following user selections. The main goal of EventIndex is to enable ATLAS members to search for and retrieve one or more individual events from the tens of millions of data files, in order to perform detailed checks, or more refined analyses. The first prototype of EventIndex was deployed in 2015. The core data storage system was reimplemented during 2021 and deployed in 2022 for the start of LHC Run 3.

[1] aleksand@jinr.ru

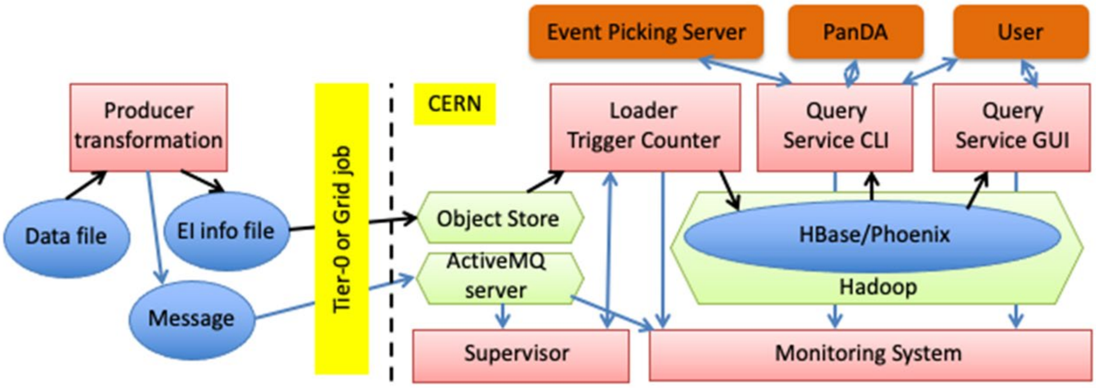

Figure 1. Architecture of the EventIndex system as implemented for LHC Run 3.

Figure 1 shows a schema of the architecture and its components, as implemented for LHC Run 3. The system must be able to scale to eventually store trillions of event records, stand ingestion rates in excess of 10 kHz and react to queries in times that are independent of the volume of stored data. Details of core parts are described in [3].

Sometimes physics analyses require massive event picking to select a set of interesting events from the wealth of ATLAS data and reprocess them with enhanced algorithms or save additional variables that can help downstream analysis steps. The EventIndex can help with it, but it requires a lot of time in manual mode. The main goal of the Event Picking Service is to perform all these actions automatically.

ARCHITECTURE OF EVENT PICKING SERVICE

The architecture of the Event Picking Service is shown in Figure 2. The service consists of two parts: a web server and a daemon, which are independent and interact using the database. The client generates a request to the server using a form embedded in an HTML page. The client receives an identifier for its request, which can be used to obtain information about the request. The web server keeps a new request in the database. The daemon monitors the database for a new request. When it encounters a new one, it starts executing the tasks for that request according to the workflow taken from the database. The input and output data, logs and results of all tasks are placed in the database. All this data can be viewed by the user or administrator using the web server. After all tasks corresponding to the workflow are completed, or critical errors are received during their execution, the daemon flags the request as finished in the database. The web server monitors the database for these flags and sends a message to the user (or administrator) about the results of the request. More details of Web Server or Daemon can be found in [4]. The new version of the Event Picking Service has a Monitoring component.

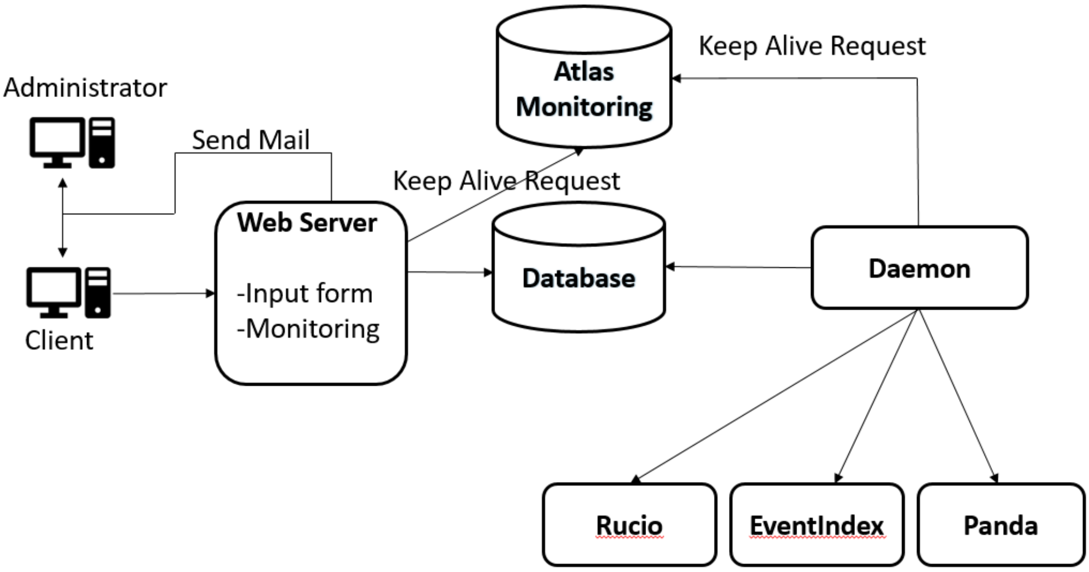

Figure 2. Architecture of the Event Picking Service.

MONITORING

The monitoring of the Event Picking Service is done according to the same principle as the monitoring of other Event Index components [5]. Both main elements of the Event Picking Service (Web Server and Daemon) send monitoring data to the database every five minutes. The scheduler uses a cron utility for periodically running jobs (a Python script) at fixed times. The Python script reads data from the database, checks the time of this data and inserts it into the InfluxDB database. The viewer part uses Grafana facilities for visualization (see Figure 3).

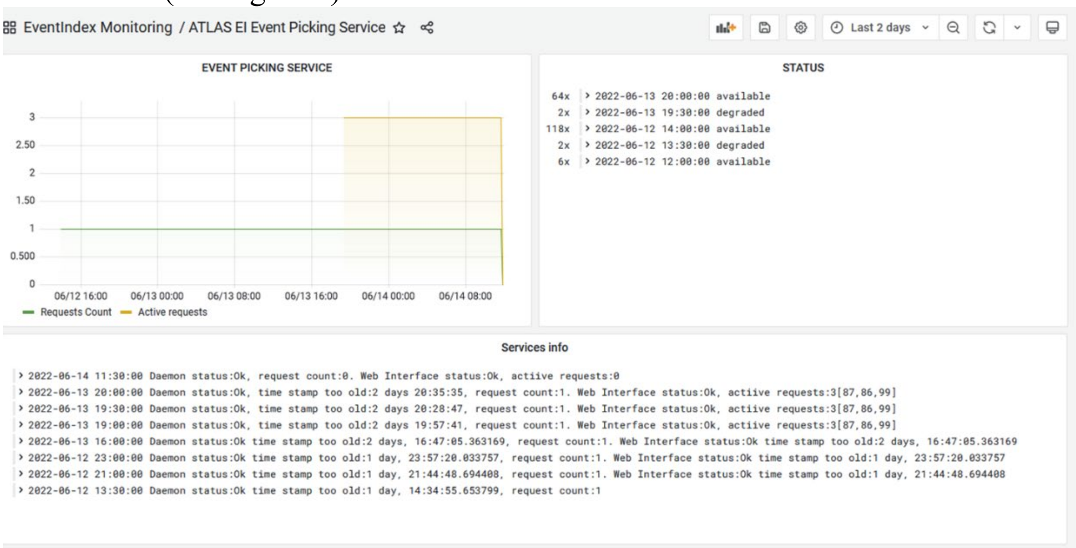

Figure 3. Monitoring of the Event Picking Service.

WORKFLOW

The Event Picking Service is not intended for solving one specific task, but for a whole type of event picking. To solve a specific problem, chains of tasks are created.

Chains can be different for different types of tasks. They are defined in the database, so that you can update or add any chain without changing the common part of the code. Figure 4 presents the current workflow of the Event Picking Service. It has two types of chain. One is the common way. The first step is to sort and split the events by run. Then it starts in parallel mode the following task: get GUIDs, get Dataset name, run main task in PanDA, check results and set metadata in Rucio. Another useful workflow is the restart request, where an earlier event picking job has returned errors for a few run/event pairs, and the goal is to complete the original job by running the original workflow, but only on the error-inducing events. The need for this type of workflow became apparent during operations, where sometimes only a small part of a task temporarily gave errors.

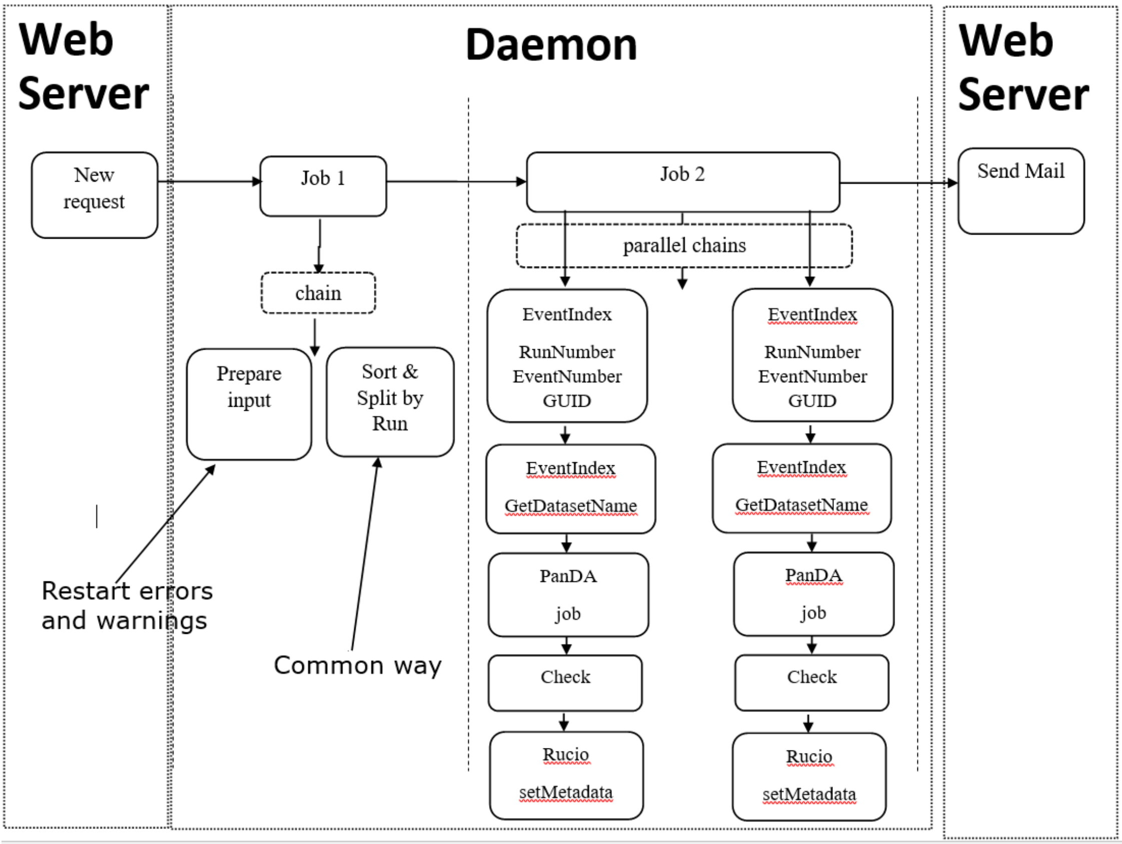

Figure 4. Workflow of the Event Picking Service.

RESULTS

Table 1. Time of work of different requests

Request	Number of events	Version	Time
$\gamma\gamma$ -> WW	50K	manual	3 months
		1.0.0	2 weeks
$\gamma\gamma$ -> WW	136K	Beta version	3 months
B_c^* -> B_c	16K	1.2.37	84h
Z -> TauTau	11K	1.2.37	40h

At present 3 different types of analyses use the Event Picking Service. The first type of analyses ($\gamma\gamma$ -> WW) was done few years ago and used the old versions of service. It required weeks to finish all requests (see table 1). The second and third type of analyses were started several months ago. They used the updated version and the completion time was only hours.

CONCLUSIONS

A production version of the Event Picking Service has been developed. The monitoring part is implemented and working. The service has separate workflows to restart on problematic run/event pairs. The production version substantially reduces the time required for work completion.

REFERENCES

1. *ATLAS Collaboration.* The ATLAS Experiment at the CERN Large Hadron Collider *// JINST — 2008 — 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. *Barberis D. et al.,* The ATLAS Eventindex Using the HBase/Phoenix Storage Solution *// CEUR Workshop Proceedings — 2021 —* V. *3041 — P. 17-25.*
4. *Alexandrov E., Alexandrov I., Barberis D., Prokoshin F., Yakovlev A., Development of the ATLAS Event Picking Server // CEUR Workshop Proceedings — 2021 —* V. *3041 — P. 223-228.*
5. *Alexandrov E., Kazymov A., Prokoshin F., BigData tools for the monitoring of the ATLAS EventIndex // CEUR Workshop Proceedings — 2018 — V.2267 — P. 91-94.*