IRIS-HEP Fellowship Project Proposal

Intelligent Caching for the HSF Conditions Database: Investigate patterns in conditions database accesses

Candidate: Ernest Sorokun
Mentor: Lino Gerlach
Project Duration: June 12th to September 1st, 2023

1 Introduction

Conditions data (voltage, current, temperature, detector calibration, etc.) is an important part of all HEP experiments and is required to process event data. Access to conditions data is critical to producing the best physics results from HEP experiments. The challenges for conditions data access are many, notably the requirement to provide simultaneous read access to conditions data for distributed computing resources at kHz rates. This is a highly non-trivial problem that has brought difficulties, especially to the biggest experiments with the largest distributed computing resources.

Today, many HEP experiments raise the question of optimal interaction with these databases. To solve this and many other problems, the HEP Software Foundation (HSF) platform was created, where representatives of different projects can share experiences, create and improve work on further experiments.

One of the areas that this community is working on is the HSF Conditions Database. They have published a set of best practices that are required to develop a system to handle conditions data. The implementation of these principles is already being used for some of the HEP experiments, and other projects are already interested in deploying this solution.

One way to improve the performance of a Conditions Database is to use a cache, but the main problem is that it can only store a limited number of queries, so we need to make a smart choice of the records we want to store, namely those that are more likely to be queried many times.

The first experiment that deployed the HSF Conditions Database is the Pioneering High Energy Nuclear Interaction eXperiment (sPHENIX), a detector designed to investigate high-energy collisions of heavy ions and protons at Brookhaven National Laboratory (BNL). It will collect data very soon and we will have access to the conditions data access patterns for our studies.

The goal is to identify patterns in database queries that will help develop an intelligent caching solution.

2 Software Deliverables

The first deliverable will be a stand-alone python project that identifies reoccurring patterns in past database queries. If time allows, the pattern recognition will be incorporated in a containerized intelligent cache as part of the conditions database application.

3 Preliminary Time Line

Week 1-2

Understand basic principles of Databases, caches, webservers and REST APIs, read literature (HSF conditions data white paper), look at source code of the implementation at hand.

Week 2-3

Understand the data format of the log files that document the DB access and extract time-series data from it for further analysis. Make first plots.

Week 4-5

Understand different access patterns qualitatively and decide on an optimized caching strategy.

Week 6-8

Develop an algorithm that automatically and in real time identifies the current access pattern and suggests an optimized caching strategy.

Week 8-10

Write documentation, summarize findings on slides. Present results in IRIS-HEP Fellowship meeting.
Fine-tune and deploy automated.