Contribution ID: **39** Type: **Talk**

Advancing Research Data Management with Python-Flask Applications

Wednesday 11 October 2023 10:50 (20 minutes)

Data acquisition (DAQ) systems continue to advance in power, but manual data input will remain required as experiments necessarily check for the unforeseen. Researchers often use electronic lab books or fallback solutions like Excel or Google Docs to record actions and events, highlighting the need for an intuitive interface that enables live- and post-processing while remaining linkable to DAQ systems. A major challenge lies in the dynamic nature of the incoming information, rendering fixed-structure databases like SQL impractical. To tackle this issue, we have developed intuitive Python-Flask applications that harness the inherent flexibility of document-based databases, particularly MongoDB, for storing curated and query-ready data. Although these applications were initially tailored for the laser-particle acceleration group at HZDR as part of the DAHPNE4NFDI project, the intention is to generalize their utility.

Our three interrelated apps are currently as follows:

1. This app facilitates manual data entry during experiments like in traditional table schemas but via a database form. The form can be easily –and on-the-fly –changed according to the needs and allows also to store the DAQ configuration per entry. Choices can be pre-configured or pulled from other sources like a MediaWiki lab documentation system. Entries are directly written to a MongoDB.
2. ZeroMQ Relayer: This extracts metadata from the experiment's drive laser (via zeroMQ) and forwards this in real-time to the experiments. This enables harmonized metadata like ID's and timestamps, either appended to data as well as logged for post-hoc reconstruction.
3. KafkaWatcher: Functioning as an intermediary, KafkaWatcher receives data from the Relayer and various experiment software agents and publishes it to MongoDB. This app is hosted on a gateway machine and allows consumption of incoming Kafka messages and subsequent storage in MongoDB. Flask-SocketIO is used for real-time reception of Kafka messages.

From a technical perspective, the Python-Flask web development microframework was chosen due to its extensive support and wide range of extensions. Python, being a widely adopted programming language among physicists, was a natural choice. Form validation is accomplished using WTForms, while Jinja2 handles template management. Dynamic form modifications are achieved primarily using JavaScript, while Bootstrap ensures a consistent form layout. MongoDB serves as the backend database for storing form fields and the various collections. Finally, Waitress is employed to serve the applications, offering compatibility with both Linux and Windows environments.

Looking ahead, enhancements like online analysis tools, data path logging, and advanced visualizations are on the horizon. While Shotsheet and KafkaWatcher offer native data search and visualization, integration of MongoDB with Grafana amplifies these capabilities. A harmonization with DAQ and SciCat databases is anticipated, facilitating sophisticated analyses.

For simulations, backing up the experiments, scripts have been created for metadata extraction from SMILEI, WarpX, and PIConGPU codes, tailored for SciCat. The convergence of electronic notebook-databases with simulation data and DAQ inputs holds promise for expanded opportunities in experimental analyses, aligning with the principles of Findable, Accessible, Interoperable, and Reusable (FAIR) data.

Please assign your contribution to one of the following topics

Metadata annotation and management close to the research process

Please specify "other" (stakeholder)

In addition please add keywords.

MongoDB, Mediawiki, Python, Flask, WebApp

Please assign yourself (presenting author) to one of the stakeholders.

Scientists and technicians who maintain and operate research infrastructure for data generation

Primary authors: SCHLENVOIGT, Hans-Peter (HZDR); TIPPEY, Kristin Elizabeth (HZDR)

Presenter: TIPPEY, Kristin Elizabeth (HZDR)

Session Classification: Parallel Track 2

Track Classification: Facilitating connectivity of research data: Metadata annotation and management during and close to the research process