ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA045

DATA ACQUISITION STRATEGY AND DEVELOPMENTS AT MAX IV

M. Eguiraun*, A. Amjad, P. J. Bell, A. Dupre, D. A. Erb, N. A. Håkansson,
V. H. Hardion, A. Milan-Otero, J. F. J. Murari, E. Rosendahl
MAX IV Laboratory, Lund, Sweden

Abstract

The experimental capabilities at the MAX IV synchrotron consists of 17 beamlines at full capacity. Each beamline puts different requirements on the control system in terms of data acquisition, high performance, data volume, pre-processing needs, and fast experiment feedback and online visualization. Therefore, high demands are put on the data management systems, and the reliability and performance of these systems has a big impact on the overall success of the facility. At MAX IV we have started the DataStaMP (Data Storage and Management Project) with the aim of providing a unified and reliable solution for all data sources in our facility. This work presents the control system aspects of the project. It is initially aimed at providing data management solution for a selected number of detectors and beamlines. It is developed in a modular and scalable architecture and combines several programming languages and frameworks. All the software runs in a dedicated cluster and communicates with the experimental stations through high performance networks, using gRPC to talk to the control system and ZMQ for retrieving the data stream.

DATASTAMP PROJECT

MAX IV has been funded to improve the data management services offered to users, notably storage, without which it would be much more difficult to provide such services. In order to create these services, a number of areas need be given more resources than they are today – where currently there is only an ambition and no available time or money. The overall mission is to improve the value of the data generated at MAX IV in terms of the benefit to research and according to the vision of the European Open Science Commons. For these reasons MAX IV decided to unify all the related development efforts in a single project: DataStaMP [1].

Currently the control system drives all the equipment at MAX IV to run the accelerator and beamline devices. Essentially, highly complex instruments are made operable by combining information from thousands of sensors and actuators into a comprehensible scheme that users and operators can operate. In order to achieve this, a high level of automation is required to reduce the large numbers of manual steps into procedures that are automatically driven by the software. The MAX IV IT team is able to deliver a functional, integrated Accelerator and Experimental Operation to a level where today's experiments can be performed.

The increased data rates and volume coming from modern detectors impose a need for new data acquisition strategies.

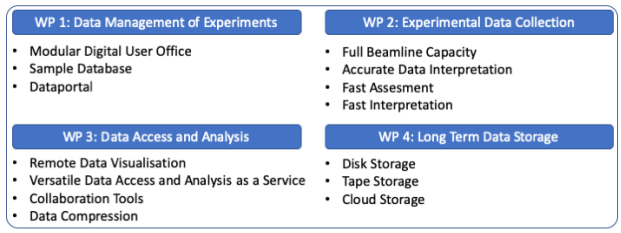

Figure 1: DataStaMP work packages, covering controls, IT infrastructure and scientific computing.

Moreover, the scale and responsibilities of the MAX IV control team suggest a unified approach for all the detectors in the facility. Therefore, this project is about a distributed (clustered) data acquisition platform for MAX IV, where all data from detectors is gathered by a multi-agent streamer ingestor. It also includes data analysis, (pre)processing and file saving.

Several implementation prototypes are being made for faster data acquisition streamed to online computing infrastructure. However, this is a moving target and plans are being made for generating data at ever higher rates in the future. The control system will need to be enhanced, particularly to improve the data acquisition but also to extract meaningful metadata and performance metrics. The researchers will analyse and interpret those in order to determine the success of the acquisition and be able to continue with the experiment. An essential factor in this is to be able to record metadata automatically while data gathering is happening. Another challenge is the variety of sensors, i.e. detectors, whose level of integration differs. This means that currently, data from all the beamlines and accelerator are not saved in the preferred standard format.

Figure 1 shows the main work packages into which the DataStaMP project has been split. Four main areas have been identified, each one covering one particular subject. *Data Management* covers all aspects related to user centred operations (user office, various databases such as sample and metadata). *Experimental Data Collection* includes acquisition, transportation and basic processing. The data analysis, scientific computing tools and services is covered by the *Data Access and Analysis*. Finally, the *Data Storage* is in charge of all the disk and network infrastructure.

The present work presents the current developments regarding data acquisition for several types of detectors. Needless to say, all mentioned topics must work together in order to provide the user first with a successful data acquisition strategy, and second, with valuable experimental data.

* mikel.eguiraun@maxiv.lu.se

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA045

EXPERIMENTAL DATA ACQUISITION

The most capable and powerful detectors push the limits of the existing hardware and network infrastructure at MAX IV. Therefore, there is already a strong commitment to provide solutions for these new demanding and challenging requests. However, this is not a new problem in scientific facilities, since these high data rate detectors have been on the market for a number of years already. Moreover, in-house developments can also demand similar control requirements.

For example, the solution provided at DESY [2] is based on a message and event notification platform. Events are related to the availability of new data. Once the availability of new data is triggered, this data is retrieved and then forwarded to multiple clients. It makes extensive use of ZeroMQ and it supports all the detectors in the facility.

On the other hand, the ODIN framework at DLS [3] splits the job into several services, and thus separates control from data acquisition or data transfer processes. Although there are not many detectors supported, it is detector agnostic and in continuous development. The communication is based on ZMQ [4] and shared memory for data transfer. In addition to those, [5] solves the data transfer issue for a single type of detector, therefore it can not be extended or generalised for multiple types of detectors.

As can be seen in the above examples (only a few from this particular ecosystem), the particular needs of a facility with many facets (e.g. data source types, budget and planning) conditions the resulting solution. Tailoring to one experimental solution can result in high performance but also make the solution non-extendable. On the other hand, making a solution that fits all experiments can lead to longer development times but also improved maintenance and development optimisation.

The solution that is under development at MAX IV tries to get the best of the available technologies and tries to learn from other solutions, knowing that every solution is very much tied to the facility. The main decisions that were made are the following:

- All data operations must be run in a dedicated cluster (see Table 1) (data transfer, disk IO, preprocessing, etc.), thus allowing low performance control tasks to run in standard control infrastructure
- Non-controls services run as docker containers
- Single framework for all data sources. Data format adapters may be required. This also requires multiple configuration parameters
- Data streaming wherever possible to increase performance
- Real time live viewers must be provided with fast feedback capabilities
- Basic online data processing (filtering, downsampling, etc.), also linked to the live viewing applications
- It should be possible to mix data from different sources
- There will be a central registry for configuration parameters

Table 1: Main Specifications of the DAQ Cluster at MAX IV

Nodes	5
Node CPU	2 x Intel Xeon Gold 5118 2.30GHz, 2x12 cores
Node Memory	64GB
Network	2 x 40Gbit/s, control and data

There are currently two main detector integration developments underway in MAX IV. They are developed with the above-mentioned ideas in mind and these are the next detectors that will arrive to the facility. Hence the need for providing a solution for these beamlines in time, but at the same time it should boost the long term developments of DataStaMP project.

A simple data protocol intended for all data transfers inside DataSTaMP is being developed, it is a unidirectional packet streaming protocol. The API is very simple and it does not impose any particular structure on the data. The data sources provide data at a given rate and the data receivers connect and consume the stream as fast as it comes. It currently supports ZMQ and websockets [6].

In addition to this, part of the project also involves the development of a pure control interface. It provides a unified API for servers to listen and clients to send commands and get feedback. This allows beamline control elements to communicate with the data transfer API. It currently supports gRPC [7] but it can be easily extended to provide ZMQ and HTTP interfaces.

The next sections describes the integrations of two detectors. The first one is for several EIGER detectors arriving this year (some already on site) for the beamlines Balder, NanoMAX and COSAXS. The second approach is part of the ICE endstation project, where the detector consists of a RoentDek spectrometer.

EIGER DETECTOR

EIGER detectors are based on Hybrid Photon Counting (HPC) technology [8]. They appeared on the market in recent years and their usage is continuously increasing. They have extremely low noise and high data acquisition frequencies. They come with the Dectris Control Unit (DCU), to which all the communications are made. An HTTP API for control and data retrieval is provided, and they also stream data frames via ZMQ sockets.

Table 2 shows the different EIGER detectors at MAX IV. The one in BioMAX is running in production since MAX IV inauguration, but in this case all the control and data acquisition software runs in a dedicated machine physically located in the beamline. As part of the new developments presented here, it is planned to move this software to the cluster. However as it is already a production ready implementation, where the user operation cannot be compromised, this transition is planned on a mid-term time scale.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA045

For the regular control of the detector we developed a python tango device [9, 10] that implements the HTTP API and exposes the attributes and commands of the detector to the beamline controls. Originally the data transfer was also part of this device, but it was removed in order to keep the logical separation between control operations and data operations. This tango device runs in standard control machines (in fact virtual machines) in each beamline network, alongside the rest of the beamline controls.

Two different data acquisition approaches have been implemented as services running in the cluster. One is a filewriter application that saves to disk data files stored in the DCU. And the second one listens to ZMQ frames coming from the detector and, after a basic data processing, stores the data in disk in HDF format.

Table 2: Different EIGER Detectors at MAX IV

BioMAX	Eiger 16M
NanoMAX	Eiger 2 4M
Cosaxs	Eiger 2 4M
Balder	3x Eiger 1M

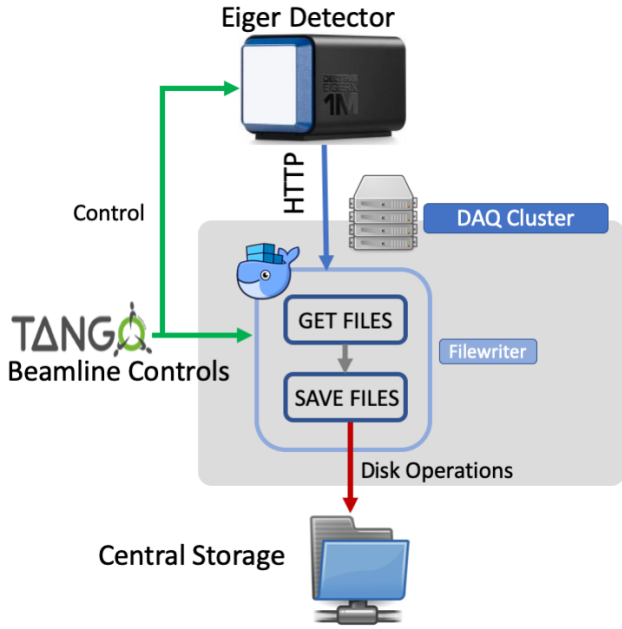

Figure 2: Filewriter application for Eiger detectors.

EIGER Filewriter

A filewriter for Eiger detectors, it queries for HDF files on the detector control unit (DCU) every few seconds and once new files are available it downloads them into locally mounted storage. It is written in *Go* and runs as a container in the DAQ cluster, therefore it can be deployed for any detector very easily. The main configuration parameters are the hostname of the DCU and the beamline name. See Figure 2 for a schematic diagram of the application.

The tango device running in the beamline can communicate with this service to retrieve its status, but also for starting or stopping it. It does that by gRPC calls [7], which easily allows connection between services in different languages and locations. The communication service is defined using Protocol Buffers, which then client and server stubs are automatically generated in the language of choice. These Protocol Buffers definitions are distributed among all the elements (filewriter service and tango device) so that any update is easy to deploy.

The central disk storage is mounted on the cluster, and consequently it is also available inside the docker container. From the beamline side the filenames are set as full paths in the tango device for simplicity, although this will change in the future with a base path together with the filename. Any file written in the storage is quickly available in the beamline workstations.

EIGER ZMQ Consumer

A ZMQ consumer for Eiger detectors is deployed when faster performance is required or when some preprocessing is needed on each frame. In this particular case, the beamline asked a downsampled image for each frame and realtime visualization. It is written in python to take advantage of existing python data manipulation libraries, and the communication between the tango device and the ZMQ consumer service uses the same protocol buffer definitions as for the previous solution.

Since the availability of a detector for testing against is limited, an EIGER ZMQ data simulator has been developed. This code mimics the frame by frame structure of a real detector, the initial metadata right after detector arming, as well as every raw data frame. The basic arming and triggering commands are also implemented. It must be noted that this simulator does not implement the HTTP API for control and configuration, the development effort of such a feature was not considered justified.

In this application, the service running in the cluster opens a ZMQ.PULL socket to the detector and waits for data. Every ZMQ frame is then saved to disk in HDF format, with a configurable number of frames per container parameter. In addition, every frame is downsampled and saved as image to an additional file on disk. The downsampling parameters are configurable at startup of the docker image. See Figure 3 for a schematic diagram of the implemented solution.

Furthermore, one of the requests in this application was to be able to see the downsampled image at run time, so that the user can stop the experiment if the resulting data fails an initial quality check. For that purpose, every Nth downsampled image is published via zmq.PUB to subscribed live viewer clients. Currently two basic image viewers have been developed, one Qt desktop application and one web application. The later transforms the data coming from the ZMQ socket to Websockets so that a Javascript client can work with it.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA045

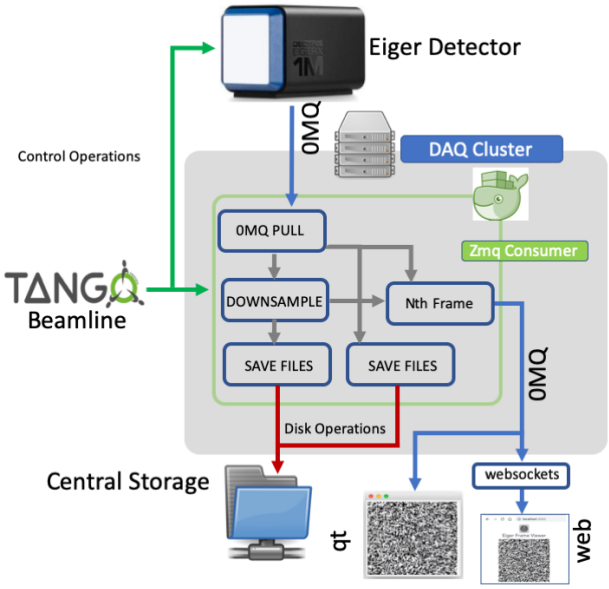

Figure 3: ZMQ consumer application for Eiger detectors.

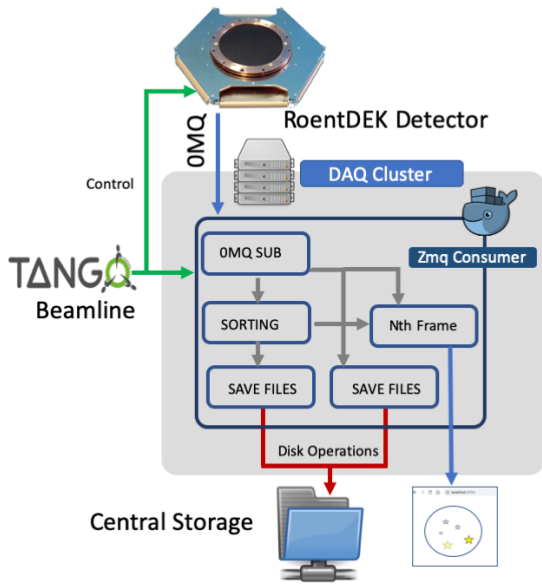

Figure 4: ZMQ consumer application for Roentdek spectrometer.

ROENTDEK SPECTROMETER

RoentDEK HexAnode Delay Line anodes [11] are 2D-imaging and timing devices that enable single particle/photon counting and the determination of the position and time of impact of a particle with high precision. A sorting algorithm distinguishes the particle traces and serializes electron and ion hits on the detector.

At MAX IV there is one detector arrangement to detect electrons and one detector arrangement to detect ions. In order to have a complete understanding of the reaction under investigation, the position and timing information for both the electrons and the ions that were emitted in the reaction are required. Realtime display of time-of-flight information (position) histograms is important for instrument adjustment and configuration.

The hardware vendor provides capture drivers on a Windows 10 platform. They also have a basic GUI control interface. Although most of the code is proprietary to RoentDEK, they do provide a user-customizable library. This library is compiled into a DLL imported by the main GUI at runtime.

The implementation required modifications on this DLL to hook it into the data acquisition pipeline. The server provides one initial start packet which contains the current hardware configuration. Thereafter, each packet consists of the raw acquisition traces from the detector which are published as ZMQ.PUB sockets.

As in the case of the Eiger above, during software development process the detector is rarely available for testing. Therefore, a simulator has been developed. It is configurable to produce a known set of packets at any desired data rate. It produces an output stream identical to the real vendor machine as well as control and status command requests.

Figure 4 shows the architecture of the application. A ZMQ subscriber (ZMQ.SUB) retrieves the raw data from the in-house modified detector software. It then applies the sorting algorithm and saves to the central storage both original raw data and processed particle positions. The users will have the option to reprocess the raw data with different experimental configurations. Varying the configuration provides an important tool for gaining different views of the underlying physics. Furthermore, for every Nth message, it is published so that a live view application displays the particle hits in realtime.

NEXT STEPS

The present work has tried to summarize the first steps made at MAX IV towards a unified data acquisition platform as part of the DataStaMP project. It is initially aimed to two types of detectors but it will cover all the data sources in the future. It is running in a dedicated data acquisition cluster connected to all beamlines and uses gRPC, ZMQ and HTTP as main communication and data transfer protocols.

In the near future the team devoted to the project will grow since the funding was finally secured. The next steps will be to get ready for the new Eiger detectors arriving in second half of 2019 so that we can fully support the initial commissioning. The basic capabilities of the system are in place so most of the development will be focused on code refactoring and modularization, mostly aimed at improving reusability and performance. In addition, the docker container deployment and maintenance strategy must be defined since new beamlines will start making use of the cluster for their new detectors.

This increased usage will require deep analysis of the performance of the whole data flow in order to detect bot-

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA045

tlenecks early in the process as well as to learn where the limits of the system are.

In the mid term, as the DataStaMP project progresses, there is a need to improve the performance of the data acquisition to to maximise the scientific exploitation of the beamlines and in addition to increase the metadata gathering and to improve live viewing applications.

ACKNOWLEDGEMENTS

The authors are very grateful to the rest of the MAX IV KITS team for their help and support during design, development and testing stages.

The DataStaMP project has been funded by The Knut and Alice Wallenberg foundation.

REFERENCES

[1] DataStaMP – Data Storage and Management Project home page, https://www.maxiv.lu.se/accelerators-beamlines/technology/kits-projects/datastamp

[2] HiDRA: High Data Rate Access, DESY https://github.com/hidra-org/hidra

[3] G. D. Yendell, U. K. Pedersen, N. Tartoni, S. Williams, A. Greer, and T. C. Nicholls, "Odin - a Control and Data Acquisition Framework for Excalibur 1M and 3M Detectors", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 966–969. doi:10.18429/JACoW-ICALEPCS2017-TUPHA212

[4] ZeroMQ, An open-source universal messaging library, https://zeromq.org/

[5] M. Hilgart, StreamWriter: A CBF Image Writer for the Eiger. NOBUGS 2018

[6] I. Fette, A. Melnikov, The Websocket Protocol, IETF Internet draft, Dec. 2011. https://tools.ietf.org/html/draft-hixie-thewebsocketprotocol-76

[7] gRPC, A high performance, open-source universal RPC framework. https://grpc.io/

[8] EIGER detectors home page, https://www.dectris.com/products/eiger/overview

[9] J-M. Chaize, A. Götz, W-D. Klotz, J. Meyer, M. Perez, E.Taurel, "Tango - an object oriented control system based on CORBA", in *Proc. ICALEPCS'99*, Trieste, Italy, pp. 475–479, 1999.

[10] PyTango: Python bindings for TANGO, http://pytango.readthedocs.io/

[11] RoentDEK HexAnode Delay, http://www.roentdek.de