FLYSCAN: A FAST AND MULTI-TECHNIQUE DATA ACQUISITION PLATFORM FOR THE SOLEIL BEAMLINES

N. Leclercq, J. Berthault, F. Langlois, S. Le, S. Poirier, Control and Data Acquisition Group, Synchrotron SOLEIL, France
J. Bisou, F. Blache, Electronics Group, Synchrotron SOLEIL, France
K. Medjoubi, C. Mocuta, Scientific Division, Synchrotron SOLEIL, France

Abstract

Synchrotron SOLEIL [1] is continuously optimizing its 29 beamlines in order to propose state of the art synchrotron radiation based experimental techniques to its users. Among the topics addressed by the related transversal projects, the enhancement of the computing tools is identified as a high priority task. In this area, the aim is to optimize the beam time usage providing the users with a fast, simultaneous and multi-technique scanning platform. The concrete implementation of this general concept allows the users not only to acquire more data in the same amount of beam time, but also to address specific phenomena otherwise difficult to tackle (like processes involving rapid transformation of the sample).

This paper gives the reader a detailed overview of the so called 'Flyscan' platform. It notably details the solution retained to generalize a prototype previously developed as a proof of concept [2][3]. Some application examples are also reported.

MOTIVATION AND SPECIFICATIONS

The main motivation for launching the Flyscan project was beam time optimization. Acquiring data faster obviously means obtaining more exploitable data at constant beam time. But it also means that one will be able to quickly discover that experimental conditions will not provide usable data – a benefit which can potentially save hours of expensive and highly sought out beam time.

From a scientific point of view, fast data acquisition also gives access to dynamics of processes involving rapid transformation of the sample (e.g. thermal annealing, continuous mechanical deformation, etc.).

The technical translation of the requirement above led to the specification of an N-dimensional scanning platform whose first dimension is "continuous". Here continuous is opposed to the step-by-step approach – *i.e.* the classical scanning technique known to generate a huge amount of dead time (acceleration / deceleration times of the motors, steady state return information, etc.). In order to reinforce the concept of beam time optimization, the ambition is also to combine the continuous acquisition approach with a multi-technique solution on the sensors (*i.e.* detectors) side. Clearly speaking, the idea is to simultaneously collect – spatially or temporally – correlated data from different sensors along the continuous trajectory of one (or more) actuator(s).

DESIGN OVERVIEW

From a technical point of view, the Flyscan backbone relies on 3 buses: the timing bus, the communication bus and the data bus.

The timing bus is in charge of the synchronization signals (*i.e.* trigger) distribution. Timing events are polymorphic and can be dispatched in the form of hardware (*e.g.* TTL) or software (*e.g.* UDP) notifications. The communication bus routes the control oriented exchanges between the scan actors. Since the latter are implemented as Tango devices [4], these exchanges are mainly composed of Tango queries and events emitted over an Ethernet network. Finally, the data bus transports the experimental data from their respective source to a central point where they are merged into a common repository. The choice has been made to use HDF5 [5] files for both data transport and storage. Figure 1 illustrates the Flyscan architecture.

It results that the Flyscan system can be seen as an aggregation of individual components sharing a common timebase to guarantee temporally – and consequently, spatial - correlation of data produced by heterogeneous sensors distributed over a network.

IMPLEMENTATION DETAILS

Timing

In the Flyscan scheme, everything starts with a software or hardware master clock. The master clock source selection is guided by application requirements. For continuous motion driven experiments – such as tomography or X-ray microscopy – a hardware solution, named *Spietbox*, has been developed to generate a spatially periodic pattern (*e.g.* one clock pulse every tenth of a degree of a rotation) [6]. The *Spietbox* can be coupled with a counting board in order to sample and record the exact axis positions at which data is acquired on each data source while the axis is moving. Solutions also exist to generate slave clocks.

ISBN 978-3-95450-148-9

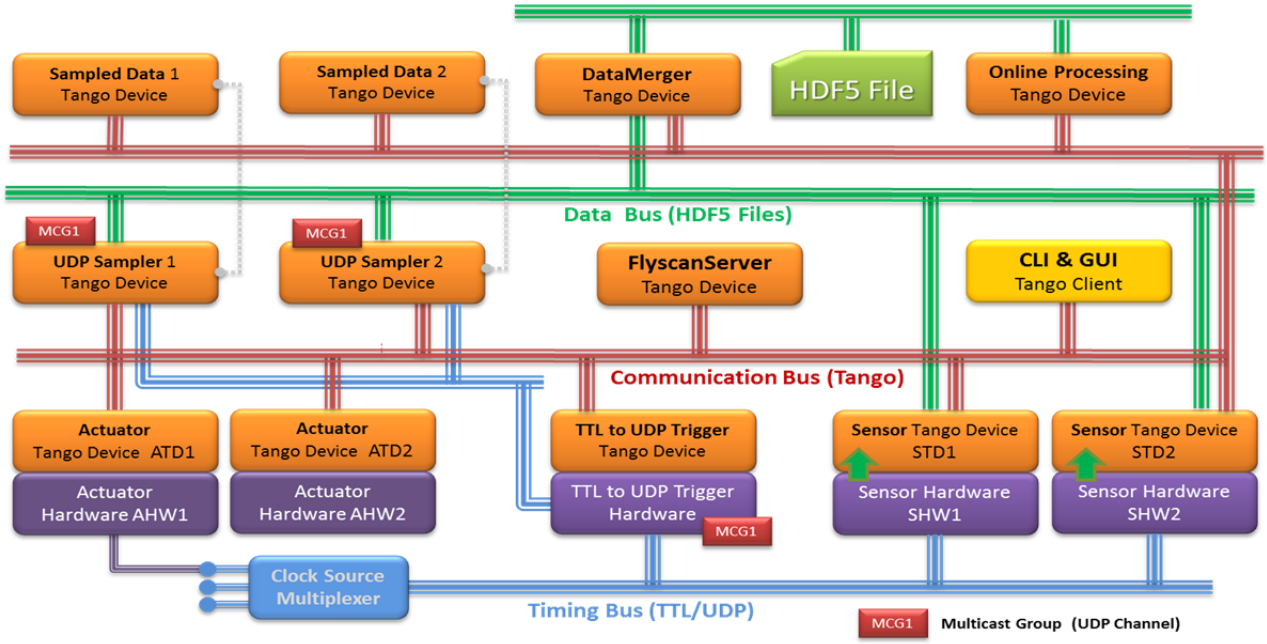

Figure 1: The Flyscan Architecture

Data Production and Transport

The timing signal is then distributed to any data source (or sensor) involved in the scan process. In most cases, a TTL signal in simply injected into the trigger input of the associated hardware. The latter is itself bound to a Tango device whose one of the main tasks is to save the acquired data into a temporary HDF5 file. The produced file is then – directly or indirectly – placed in one of repositories monitored by the data merging process. The Flyscan deployment scheme is mainly governed by technical and/or performance constraints. It is usually chosen so that the "biggest data" – such as 2D detector images – is not moved over the network. In cases where data displacement is inevitable, high speed network links can be setup to deploy our optimized FTP service.

It could seem technically archaic to use files as a data transport service. In fact, this approach offers numerous advantages. First of all, files act as data buffers whose size might not be reached by a RAM based implementation. Moreover, by their nature, files provide a way to asynchronously process gigabytes of data by relaxing the performance constraint on the consumer side. Once stored into a file, the data can safely wait for being processed, consumed or moved. Finally, technologies such as 10 Gb/s Ethernet, SSD, tmpfs… offer OTS solutions to implement file based exchanges with the expected performances.

However, the Flyscan has been globally designed as a general purpose solution and implemented using OTS technologies. Consequently, it could reach its limits when confronted with use cases producing very high data throughputs. Today, such applications – like ultrafast tomography – are specifically addressed by optimized and specially designed hardware and software infrastructures [7]. The Flyscan is not designed to compete with these kinds of applications. See the PISCHE beamline use case example for some Flyscan performance figures related to file transfers.

Online Processing and Visualization

Among the Flyscan specifications, online processing and visualization was one of the most challenging demands. In some applications – *e.g.* long scanning process, such as 2D raster scans – providing the user with a live feedback is mandatory. Here, the idea is to propose a generic mechanism for visualizing both raw and preprocessed data in order to validate the experimental context while the data acquisition is in progress.

Since the whole data scan is merged into a single or multiple files it is directly exploitable by any process with proper read access to the file(s). The "one writer, several readers" problem has been easily solved by pre-allocating the data into the HDF5 file(s) produced by the merging process. Our experience proves that space pre-allocation is a safe and reliable cunning to process the data while the acquisition is in progress (*i.e.* while data is continuously merged and written to the file).

This online processing and visualization requirement has been implemented in the form of a Python [8] Tango device supporting a plugin mechanism. It then becomes straightforward to process the raw data and to make the resulting information available on the network as Tango dynamic attributes. Any Tango aware client can then display this information in order to provide the expected feedback.

ISBN 978-3-95450-148-9

Scan Configuration, Sequencing and Monitoring

The core component of the Flyscan implementation is a Tango device whose aim is to orchestrate and monitor the scan process: the FlyscanServer (FSS). From a client point of view, the FlyscanServer is the unique entry point of the system. The main FSS activity is to delegate the actions constituting the scan sequence to abstract entities called "actors". The FlyscanServer currently support the following actors: continuous-actuators, actuators, sensors, timebases, hooks and monitors.

Continuous-actuators are associated with the physical parameters whose continuous variation represents the first dimension of the scan. Actuators incarnate the same concept for a step-by-step variation in any other dimension of the scan. Sensors are the experimental data sources producing the temporary files consumed by merging process. As its name might suggest, a timebase represent a timing signal generator. A hook is a user defined action that can be performed at several "key moments" of the scanning sequence. Finally, a monitor represents a required experimental condition for the scan to be performed.

The system allows attaching an unlimited number of actors to each scan action. For instance, two or more continuous-actuators can be attached to the continuous dimension of the scan; similarly several timebases can be managed in the same scanning sequence. In order to fulfil the initial specifications – *i.e.* the multi-technique constraint – the number of sensors is obviously unlimited.

For the FlyscanServer, an actor – whatever its type – is an abstract entity encapsulating the implementation details of the subsystem its represents. The FSS simply delegates the execution of a specific action to a specific plugin. The latter can encapsulate a very simple – or, on the contrary – quite complex logic. In the average case, the plugin acts as a proxy to a Tango device representing the actual actor (motion axis, counting board, 2D detector …). In order to offer the highest level of flexibility, plugins can be written in either C++ or Python.

A FlyscanServer plugin comes with all the details about the configuration parameters of its associated subsystem. Information such as parameter unit, data type, data format, default value, documentation or specific constraints is provided. This feature allows implementing smart Flyscan clients using an introspection of the plugins parameters description. Thus, the Flyscan platform gives the user access to any actor specific configuration parameters in a generic manner and allows these actors to be finely tuned and controlled.

Despite the flexibility of the approach, handling actors' configuration at this level of detail can lead to complexity. Beyond the fact of being obliged to valuate each parameter – which is error-prone – it also requires each user to have a deep knowledge of each actor's behaviour. In order to reach the expected user-friendliness, a so called "easy-configuration" mechanism has been implemented in order to propose a way to configure the scan with a small set of "master parameters". The latter are defined by an application expert and are supposed to allow computation of the whole set of scan parameters. The easy-configuration mechanism is implemented in the form of a python script executed by the FlyscanServer as the first step of the scan sequence. This approach encapsulates (*i.e.* hides) the technical details and the specificities of a given experimental setup into the expert's python code. It is important to note that the easy-configuration feature also provides the end-user with an "application oriented" view of our generic scanning platform. Thus, the Flyscan deployment can be fully customized to hide all the details in order to fit with the users' experimental habits.

The Flyscan Client Interfaces

The Flyscan comes with a Python command line and a Java GUI. The former is an *ipython* profile offering access to the Flyscan python client API. Beyond its scan configuration capabilities, the GUI also provides a visual feedback on actors' activity.

APPLICATION EXAMPLES

Tomography on PSICHE

One of the PSICHE beamline [9] experimental platforms is dedicated to tomography. The current setup integrates a 2D detector generating 8MB/image at up to 90 Hz. A local technical constraint requires the images to be transferred from the detector host to the one hosting the merging process. Despite this constraint, an average data throughput of 5.4 Gb/s (with 6.1 Gb/s peak) has been achieved on a dedicated 10 Gb/s network link using the Flyscan FTP service (which is itself implemented by 2 Tango devices implementing a subset of the FTP protocol). The 4000 images acquired during the 360° sample stage rotation are finally obtained in an average time of 50s.

X-ray Microscopy on NANOSCOPIUM

The NANOSCOPIUM [9] beamline, currently under commissioning, is aimed to high-resolution fast scanning multi-modal imaging. It will provide high sensitivity elemental, and structural mapping with down to 30 nm spatial resolutions by fast scanning XRF (X-ray fluorescence) imaging combined with absorption, differential phase contrast and dark field imaging. The Flyscan architecture, including the FSS, has been deployed on a microprobe prototype station. A 2D preliminary fast scanning test experiment has been performed. A nanostructure calibration chart – which includes a 250 µm x 75 µm SOLEIL logo – was used as a test sample. A scan of 500 x 1000 pixels with 4ms of dwell time took 35 minutes to complete and produced a

ISBN 978-3-95450-148-9

HF5 file of approximately 100 Go of raw. Thanks to the Flyscan online data processing mechanism, live reconstructions have been performed to produce some information-rich feedback to the users. Such an experiment could not have been achieved without a continuous scanning approach.

Raster Scans on DIFFABS

On the DIFFABS [9] beamline a performance gain of nearly 60 was measured using the Flyscan. A 121 x 121 raster scan with a 10ms integration time per point has been obtained in 4h28' in step-by-step mode. In the same conditions, a 121 x 225 raster scan completed in 8'40'' when managed with the Flyscan. Large area (20 x 20 mm) fluorescence raster maps were also performed with a 200 µm resolution (~ 10^6 points) and 10ms dwell time in slightly less than 3 hours.

Figure 2 presents results based on real performance measurements and linear extrapolation.

FUTURE AND PERSPECTIVES

Deployments

We already identified and fully specified 30 potential Flyscan applications on 14 of the 29 SOLEIL beamlines. As of this paper, the Flyscan platform is currently in production on 3 beamlines. The deployment schedule is under discussion – taking into account the scientific priorities and resources availabilities.

Developments

On the hardware side, we just started a collaboration on the so called "*PandA*" project [10] with the DIAMOND Light Source [11]. The main objective of *PandA* is the development of a powerful and feature rich hardware solution dedicated to continuous motion oriented applications. It means that *PandA* will replace the *Spietbox* in the near future.

On software side, the refactoring and enhancement of the merging process will constitute the major part of the activities in the next months. Compressed data support is one of the main topics that will be addressed by this work.

In parallel, we are also working on advanced features – such as continuous scan in the "reciprocal space" (aka hkl scans) for X-ray diffraction measurements. A solution already exists at SOLEIL and we are now on our way to make it available to any beamline with an experimental setup dedicated to crystallography.

CONCLUSION

The Flyscan project led to a highly scalable and tunable solution for continuous and multi-technique scans. The obtained performances and the provided features make the solution fulfill the initial requirements and specifications.

The Flyscan already saved tens of hours of beam time and allowed to analyze a huge quantity of samples. The platform extensions currently under development will offer even more experimental capabilities at medium term.

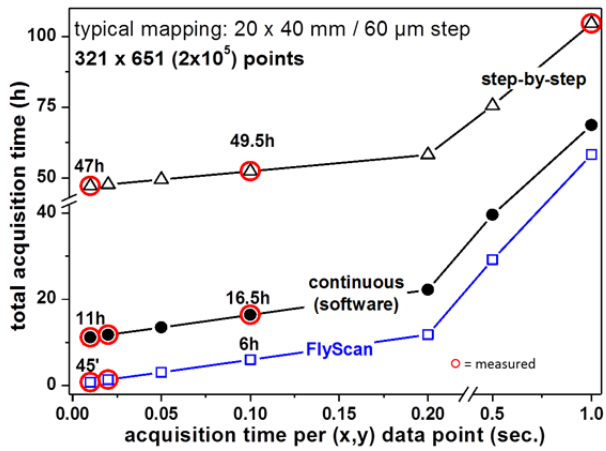

Figure 2: Performance comparison between several scanning techniques: step-by-step, user's implementation and the Flyscan. Thanks to the optimization work done on some actors, the results are now even better than the ones reported here (a supplementary gain of approx. 15% has been obtained).

ACKNOWLEDGEMENTS

The authors are grateful to the SOLEIL management team for is constant support.

As the project leader, I, Nicolas Leclercq, express my deep gratitude to my SOLEIL colleagues involved in this transversal project. I particularly thank the coauthors from their respective contributions.

REFERENCES

[1] Synchrotron SOLEIL: http://www.synchrotron-soleil.fr
[2] Medjoubi et al., Journal of Synchrotron Radiation
[3] Langlois et al., ICALEPCS 2011, Grenoble, France, WEPKN003
[4] Tango Controls: http://www.tango-controls.org
[5] HDF5: http://www.hdfgroup.org
[6] Y.M. Abiven et al., ICALEPCS 2011, Grenoble, France, WEMMU004
[7] Swiss Light Source giga-Frost project: https://www.psi.ch/sls/tomcat/detectors#Detectors_and_Optics
[8] Python: https://www.python.org
[9] http://www.synchrotron/soleil.fr/Recherche/LignesLumiere
[10] I. Uzun et al. - PandA Motion Project - Proc. of ICALEPCS'2015 - Melbourne – Australia
[11] The DIAMOND Light Source: www.diamond.ac.uk

ISBN 978-3-95450-148-9