

# STXM analysis: Preparing to go live @ 750 Hz

Cite as: AIP Conference Proceedings **2054**, 060075 (2019); <https://doi.org/10.1063/1.5084706>  
Published Online: 16 January 2019

Markus Osterhoff, Jan Goeman, Tim Salditt, and Sarah Köster



View Online



Export Citation

## ARTICLES YOU MAY BE INTERESTED IN

[Preface: 13th International Conference on Synchrotron Radiation Instrumentation \(SRI2018\)](#)  
AIP Conference Proceedings **2054**, 010001 (2019); <https://doi.org/10.1063/1.5084557>

[Applications of single-crystal CVD diamond XBPM detectors with nanometre x-ray beams](#)  
AIP Conference Proceedings **2054**, 060052 (2019); <https://doi.org/10.1063/1.5084683>

[Detectors for present and future light sources at Elettra](#)  
AIP Conference Proceedings **2054**, 060071 (2019); <https://doi.org/10.1063/1.5084702>

## Lock-in Amplifiers up to 600 MHz



Zurich  
Instruments



# STXM Analysis: Preparing To Go Live @ 750 Hz

Markus Osterhoff<sup>1,a)</sup>, Jan Goeman<sup>1</sup>, Tim Salditt<sup>1</sup>, Sarah Köster<sup>1</sup>

<sup>1</sup>*Institut für Röntgenphysik, University of Göttingen;  
Friedrich-Hund-Platz 1, D-37077 Göttingen*

<sup>a)</sup> Corresponding author: mosterh1@gwdg.de

**Abstract.** Recent advances in detector technology, combined with continuous 2D Piezo scanning, are posing a significant challenge on data analysis. With the EigerX 4M detector by Dectris, for example, four million single-photon counting pixels can be read out at 750 Hz frame rate. Although compressed, this yields a data stream of roughly 100 MB per second. For analysis, data need to be uncompressed, yielding a data rate of more than 6 GB per second. To deal with the data obtained at such high rates in scanning transmission x-ray microscopy (STXM) measurements, we have developed a dedicated STXM analysis cluster, the “Heinzelmännchen system”. With a special software, 24 computing nodes can process up to 3 500 EigerX frames per second, i.e. about four times as fast as the data is acquired. This allows for new algorithms to be developed, when datasets need to be processed multiple times in iterative schemes.

## INTRODUCTION

Classical Small-Angle X-ray Scattering (SAXS) is a well established technique both in laboratory as well as in synchrotron radiation facilities to quantitatively characterise biological macromolecules in solution, or more generally, colloids in suspension [1–2]. From one-dimensional SAXS curves that show the intensity as a function of scattering vector length,  $I(q)$ , information about the size distribution and shape of the particles can be extracted. Typically, the particles have sizes in the range of 1 to 100 nm, and mesoscopic volumes are probed at once. With pixelated 2D detectors, resolution in reciprocal space has increased; highly sensitive photon counting detectors and 3<sup>rd</sup> generation synchrotron beams allow for short illumination times on the sub-second time scale.

With the high brilliance and focusing capabilities of 3<sup>rd</sup> generation synchrotrons, it is a natural step to use small X-ray beams. Thus, the high reciprocal space information of SAXS is combined with high real-space resolution [3]. The latter is basically given by beam-size or step-size, which ever is larger. Today, beam-sizes of sub-100 nm can easily be achieved using Kirkpatrick-Baez (KB) focusing mirrors [4] or Fresnel Zone Plate (FZP) lenses [5]. For some experiments, even micron sized Compound Refractive Lens (CRL) beams are used very efficiently [6]. Combined with piezo scanning sample towers and continuous 2D trajectories, large sample areas can be raster-scanned in short times [7]. Recently, a scanning micro-SAXS measurement of cardiac tissue slice in a  $2\ \mu\text{m} \times 3\ \mu\text{m}$  CRL beam at a step size of  $5\ \mu\text{m}$  was performed [8]. With 1 430 by 1 501 scan points, a total area of more than  $40\ \text{mm}^2$  was measured in about twelve hours. At each of the 2.1 million real-space positions, a full EigerX 4M detector image was acquired in 10 ms, yielding a (compressed) data set of more than one Tebi Byte and more than nine trillion pixel values.

With the brilliance of X-ray sources outperforming Moore’s law since decades, researchers are facing problems of data handling and data analysis. For a long time, it has been possible to analyse STXM measurements on-line, quasi in real-time during the experiment. Fast feedback of the measurements is crucial both to align the sample, and to decide how to proceed. Analysis has to become parallel, and run on clusters of computers.

Data storage and archival at the facilities has – for the time being – been solved by large financial investments. At PETRA III, a write-cache (beamline filesystem) of about 100 TiB is being upgraded from mixed hard drive-flash to flash-based drives only. In the background, data is then transferred to the main storage area in the several PiB range, and after some time archived to tape. Also, large-scale machines for parallel analysis are available to treat the data [9].

On a smaller scale, we here present the current efforts of data handling and analysis at the Göttingen Institute for X-Ray Physics. In the following, we will introduce the Heinzelmännchen cluster and the *dada* (data daemon) software, which is designed for parallel analysis of massive STXM data sets.

## HARDWARE: THE HEINZELMÄNNCHEN CLUSTER

The Heinzelmännchen cluster consists of (currently) one control unit and 24 analysis nodes. The control computer acts as a load balancer and data cache; it is connected via a  $2 \times 10$  Gb/s Ethernet bond to a central switch, and then with 1 Gb/s to each node. The connection to the main storage servers is also carried out via redundant  $2 \times 10$  Gb/s connections; raw data is mounted via NFS 3 on the control computer, where the *dada* filesystem proxy daemon (*dadafsp*) uses 4.4 TiB SSD (expandable) as a cache. The SSDs are connected via an LSI SAS 9300-8i adaptor.

The control unit provides DNS resolution, IP management (DHCP), and further network services for the internal network; nodes boot via network and obtain their root filesystem via NFS. In the following, we will give more technical details on the hardware. The software is presented in the next section.

The Heinzelmännchen are named after fairy-tale creatures from Cologne; these gnomes are said to have done all the work of the citizens during the night, so that the inhabitants of Cologne could be very lazy.

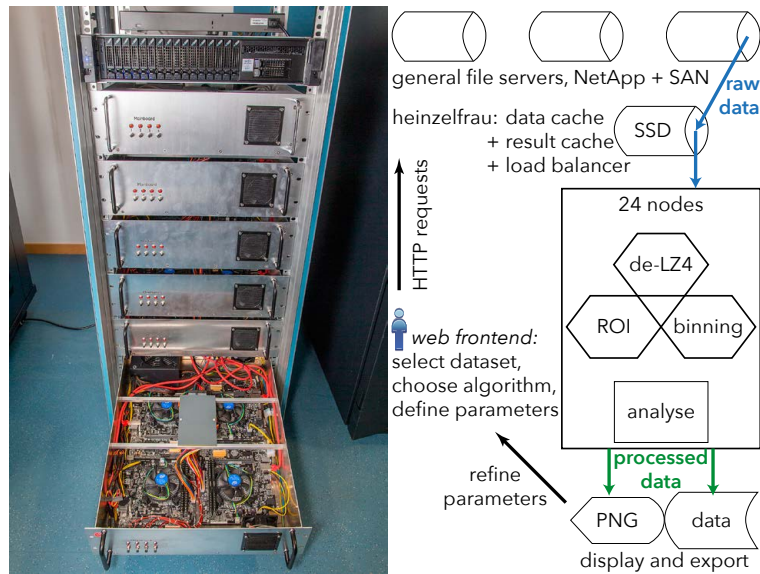


FIGURE 1. A photo of the “Heinzelmännchen”, and a data flow chart.

### Control Computer

As a control unit, an Intel Core i7-8700 (six cores + hyperthreading, 3.2 GHz base frequency) processor with 64 GiB of RAM is used; as mainboard, an MSI Z370 was chosen. The operating system is Gentoo Linux, running a kernel 4.14.12 in customised and minimal configuration, installed on a Samsung SSD 850 Pro with 238 GiB of storage; most of the storage is configured to be used as cache for analysis results.

In addition, five SanDisk Ultra II SSDs with 894 GiB each are used as a cache of the raw data. The raw data at the Institute for X-ray Physics is currently being migrated to a new server (Dell PowerEdge R730), which is fibre-connected to a NetApp E-Series E2800 with 150 TiB (to be upgraded every year).

Network connectivity is provided by two dual-port (in total: four ports) 10 Gb/s Ethernet cards (Intel X540-AT2), with bonded  $2 \times 10$  Gb/s into the internal network and to the institute’s file servers. For the internal network, a Netgear S3300-52X switch with 48 client 1 Gb/s-ports, two server 10 Gb/s-ports, and two 10 Gb/s fibre-ports for future upgrades is used. Users interact via the HTTP API (web frontend, rendered image files, and data access) via a dedicated 1 Gb/s link into the office network. The control computer “heinzelfrau” was built by Sysgen.

### Analysis Nodes

For the actual data analysis, we have chosen a minimal hardware configuration per node:

- Intel core i7-7700 (four cores + hyperthreading, 2.6 GHz),
- 8 GiB RAM,
- Asus H110M-A;
- no storage (booting via TFTP, root-fs via NFS).

Four nodes share a common power supply (650 Watt), and are installed into custom-built 19 inch sliding drawers. The thermal design of the CPUs allows for a load of about 360 W per drawer, but we did not observe a power load of more than 250 W during testing. We have measured a peak power consumption of less than 1.5 kW for the full rack with 24 nodes in six drawers, and one control PC.

We have measured that each node can retrieve data with a bandwidth of about 750 Mb/s single-threadedly, corresponding to a total throughput of about 17 Gb/s for the full system. With typical LZ4 compressed EigerX SAXS datasets at short exposure times and low background noise, one four megapixel frame can be compressed down to 100 kiB to 1 MiB; so the current Heinzelmännchen bandwidth allows to transport at least ten thousands of frame per second to the analysis nodes. Then, the LZ4 data needs to be uncompressed to about 17 MiB of data per frame (saved as signed 32 bit integer).

Benchmarks have shown that a single node can decompress and analyse at a rate of about 2 GiB/s (raw data) or about 100 MiB/s (LZ4). With our dada software (see next section), analysis rates of 14 gigapixels per second can be reached in a simple STXM analysis.

## SOFTWARE: DADA, THE DATA DAEMON

One of the original ideas of the `data daemon` (or `dada`, for short) was to have a single interface to detector images, which are stored in various file formats and have differing (and changing!) file name conventions. Especially for new Bachelor students joining a university group for only few weeks, it can pose a strong barrier to open detector files, and read them in correctly. Examples of the problems are the orientation, handling of bad pixels, different integer formats and compression settings.

During the experiment, fast feedback is necessary. The `dada` web frontend gives easy access to detector images as PNG images, allowing also for simple pre-processing and analysis. In the following sections, we describe the HTTP interface including the web frontend and the HTTP API, and show the parallelisation during STXM analysis using the Heinzelmännchen cluster. We further present the included caching mechanisms needed to achieve the data rates, and summarise the currently included analysis methods for STXM scans.

### Interfaces

The `dada` can be accessed via its HTTP API, by simply submitting a request URL to its TCP socket. This URL fully describes the analysis, and can be generated by use of the web frontend. But it is human readable and can be used as a generic interface for 3<sup>rd</sup> party programs. URLs can be built graphically via the web frontend.

#### *Web Frontend*

The `dada` web frontend is a jQuery based GUI to select single images, and perform basic pre-processing on the data. First, the user chooses an instrument and an experiment; then for each used detector, the list of sample names is generated from the filesystem layout, and individual images can be rendered with different colour palettes, and colour map settings (lin and log scale, user-defined min and max values, etc).

As basic pre-processing, a rectangular region of interest (ROI) can be chosen, and integer-valued binning (in horizontal and vertical direction individually) can be applied. To accomplish an “empty beam correction”, an image can be divided pixel-wise by another image; similarly, the pixel-wise difference of two images can be calculated. If an image series as been recorded, the stack operations (total sum, average, minimum and maximum projection) can be applied.

The same operations can be applied on a two dimensional “grid of images”. This is especially useful if detector frames have been recorded with the sample being raster-scanned as is common in STXM setups. This 2D view of 2D data helps for visual inspection, and to find e.g. regions in real-space with significant scattering. The “multi” mode could also be used to import a large number of consecutive images – including the aforementioned pre-processing – into own analysis code.

The STXM module applies the selected pre-processing on the images, and then performs different algorithms on a raster-scan of images. Each detector image is reduced to a single scalar number and colour coded on a corresponding two dimensional grid. See further below for currently implemented and planned analysis methods.

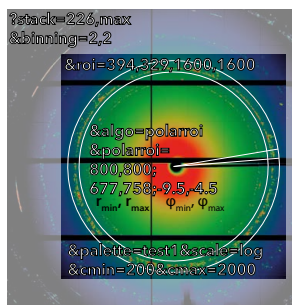
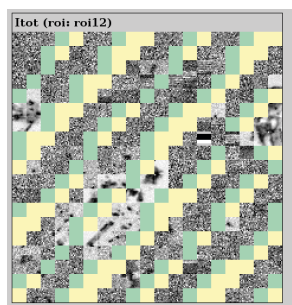
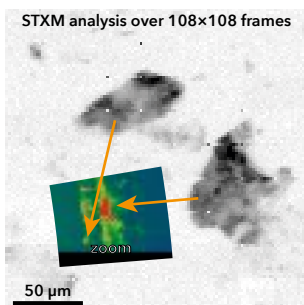


FIGURE 2. A rectangular master ROI has been cut out of the full detector image; then, a polar ROI around a Bragg peak has been defined, here highlighted in white. A STXM analysis shows two grains and their respective Bragg signals.



super grid:  $19 \times 19$ ; STXM:  $12 \times 12$   
 pixel size:  $2 \mu\text{m}$ ; field of view:  $456 \mu\text{m}^2$   
 time per pixel: 20 ms  
 effective time: **17 minutes**  
 total time: 66 minutes (large overhead)  
 parallelised analysis over super grid,  
 361 jobs on 24 nodes  $\times$  4 cores  
 time to analyse: **2.5 minutes**  
 green: currently analysed,  
 yellow: queued job  
[http://dada/stxm/GINIX/run74/eiger/L\\_trypCuCl\\_06/{1 ... 361}/1?horz=12&vert=12](http://dada/stxm/GINIX/run74/eiger/L_trypCuCl_06/{1 ... 361}/1?horz=12&vert=12)

FIGURE 3. Parallel analysis on a super-grid of STXM scans. The 361 individual scans are put into 96 worker queues, which are then distributed to the 24 computing nodes. Green squares are currently being processed, yellow are still queued.

### 3<sup>rd</sup> Party Interface

While the web frontend basically shows the results as PNG images to the user, other software can interact with the data server by a simple HTTP request, as well. Apart from a graphical rendering, also the raw values of the processed data can be accessed in different formats. The most important `&format=` specifier is `tiff_raw`, which produces a 32 bit floating-point encoded TIFF file of two-dimensional data. In addition, text representations in ASCII or JSON format are available.

#### HTTP API

The data daemon can be accessed via its HTTP API remotely. Here we only give a few examples of request URLs. Assume that all URLs begin with something like `http://dada/`.

- `show/GINIX/run67/sCMOS/preKB_01/38?roi=320,500,1400,940&binning=2,2&cmin=4e3&cmax=120e3`  
render a ROI with binning of image number 38 using a linear colour scale, with given min/max values.
- `show/GINIX/run74/eiger/spiCuCl_03/2/1?stack=361,max&empty=362&format=tiff_raw`  
obtain a (pixel-wise) maximum projection of 361 images, starting at image 1; divide (pixel-wise) the result by image 362; output as 32 Bit TIFF file.
- `multi/GINIX/run74/eiger/D_tryp_04/30/1?&horz=12&vert=12&roi=450,1100,32,32`  
Show a small ROI of the detector, for 12 by 12 images.
- `stxm/GINIX/run74/p300/spiCuCl_05e/432?&signal=Itot&horz=12&vert=121&palette=wb`  
Show the total intensity scattered onto the Pilatus detector, for 12 by 12 images.
- `stxm/GINIX/run74/eiger/D_tryp_04/30/1?&algo=polarroi&polarroi=800,800;677,758;-180,-178&horz=12&vert=12&roi=394,329,1600,1600`  
Show the total intensity scattered into a polar ROI on the detector, for 12 by 12 images.

#### Analysis methods for STXM module

In this subsection, we briefly describe the different analysis algorithms currently implemented for the STXM module. They can be chosen via the `&algo=xxx` modifier, and can have different data channels accesible with `&signal=yyy`.

The most simple algorithm is `&algo=brightfield` (default), which integrates over all pixels (after ROI and binning). It offers the signals `&signal=Itot` (total intensity in the selected ROI, default), `&signal=Imax` (maximum intensity over all pixels in the selected ROI), `&signal=comh` (horizontal centre-of-mass) and `&signal=comv` (vertical centre-of-mass).

Especially for WAXS datasets, analysis limited to a curvilinear ROI in polar coordinates can be useful. This is possible with `&algo=polarroi`. Before the polar ROI is applied, the (rectangular) ROI and binning operations are applied. Then, the modifier `&polarroi=cenx,ceny;minq,maxq;minφ,maxφ` is used to define the centre of

the polar grid (usually the beam centre), and the minimum and maximum distance ( $q$ ) and angle ( $\phi$ ) to be considered. Currently, only the (default) signal  $I_{tot}$  is calculated. The origin – given by  $cenx, ceny$  – can be placed outside the current field of view. The distance  $q$  is measured in pixels, the angle  $\phi$  in degree.

Another algorithm working in polar coordinates can be selected with **&algo=radial**; the detector data is integrated azimuthally, and signals for different  $q$ -ranges can be extracted in parallel. The parameters are passed as **&radial=cenx,ceny;minq,maxq;numq**. Again, the analysis is limited by the rectangular ROI and binning modifiers. In the general case, a three-dimensional dataset will be returned, providing the function  $I(q)$  for all STXM positions simultaneously.

Under development is the **&algo=gridroi**, which will calculate  $I_{tot}$ ,  $comh$ , and  $comv$  signals for a grid of rectangular ROIs. This will be used for multi-order imaging analysis using zone plate optics without on order sorting aperture. More algorithms can be implemented as plug-ins.

### Parallelisation and Caching

Within the STXM module of the web frontend, jobs are split line-wise and sent for parallel analysis to the load balancer, to be run on all 24 nodes of the cluster. The measurement shown in Fig. 3 has been carried out on a “stitched STXM” layout, i.e. the whole 2D scan was split into small STXM scans of  $24\mu m$  edge size, and then repeated on a super-grid. For the analysis, the STXM scans are analysed individually and in parallel. The different scan regions are placed into a working queue, and shown in yellow; scans currently being analysed are shown in green, and the  $I_{tot}$  signal of finished regions is shown with a white–black colour scale.

Results of the STXM analysis, also partial results in a line-wise setup, are cached for future re-use. Currently, the Aerospike database is being used, but in future a filesystem based caching module might be implemented. It is also planned to cache results of the `show` module, if the ratio of computational time and storage space is above some threshold; this is important for repeated working on large images stacks.

### *dadafs, a caching filesystem*

To achieve high network bandwidth – without disturbing the regular file server –, we have designed the caching filesystem *dadafs*. On the client side (Heinzelmännchen nodes `h001...h024`), it provides a FUSE based filesystem. The client connects to a proxy server (*dadafsp*) running on *heinzelfrau*, which is connected to the final server daemons (*dadafsd*) on running *homer4a* and *homer4b*.

When the *dada* software (or any other program) accesses data from the FUSE directroy tree at `/mnt/dadafs`, the client first tries its very limited RAM cache. If the request can be answered, it is forwarded to the proxy server, which makes a hash table lookup in RAM, and eventually serves the data

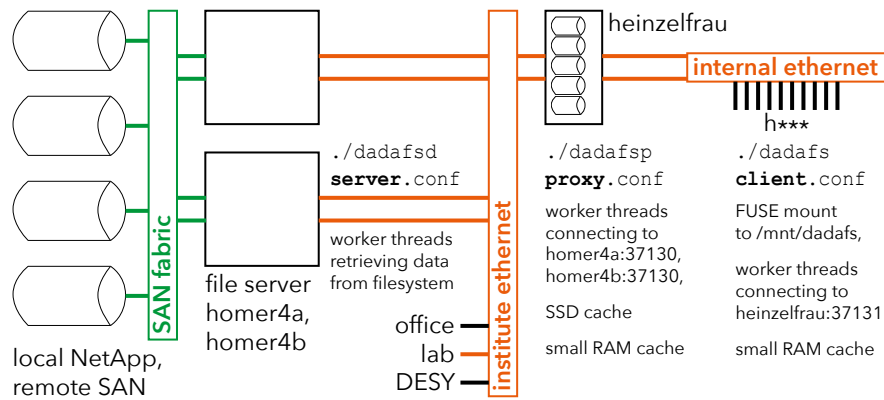


FIGURE 4. asd

from SSD. If the data is not cached, the request is forwarded to the file servers, which look into the nominal file system. The design of the *dadafs* allows for very flexible configurations, with multiple layers of caching and multiple paths to data as failover.

With the first prototype, a total network bandwidth of up to 16Gb/s is achieved for SSD-cached data. In the release version, the *dadafs*-clients will be multi-threaded and the cache lookup on *dadafsp* will be improved. We envisage a bandwidth of 19Gb/s or better on the  $2 \times 10$ Gb/s bond link. Since the client is FUSE based, the *dadafs* infrastructure will be used by other programs on other Linux machines, apart from the Heinzelmännchen system.

## Performance

During development of the `dada` STXM module and the `dadafs`, benchmarks on several datasets have been carried to quantify the current speed and improvements. First, detector raw data was accessed via plain NFS from the central file server, providing only very bad performance. With the introduction of the SSD caching `dadafs` proxy on the control computer, bandwidth could be increased tremendously. Currently, about 16 to 17 Gb/s are achieved, limited by the single-threaded implementation of the `dadafs` `client`. We found that from a hot cache, more than 3 500 EigerX 4M frames can be (i) sent to, (ii) uncompressed on, and (iii) analysed by the Heinzelmännchen computing nodes per second. Here, only the very basic “total intensity per frame” was calculated, which is limited by network capacity. More complex analysis methods involving polar ROIs and azimuthal integrations become CPU-bound.

## SUMMARY

We have designed, build, and programmed a dedicated cluster for parallel analysis of STXM datasets. The `dada` software serves as a centralised entry point for experimental data obtained with different detectors at different experiments. Via its HTTP interface, raw data and pre-processed data can be accessed. The web frontend allows the user to easily browse through the data, and to generate an URL to access the data from 3<sup>rd</sup> party software. With the STXM module, large scans can be analysed using different algorithms.

By introducing an SSD-cached network filesystem, data rates of more than 16 Gb/s can be reached, which in case of highly compressed nano-SAXS measurements corresponds to more than 3 500 EigerX 4M frames per second.

## OUTLOOK

The `dadafs` is still under development, but a first version with a `client-proxy-server` architecture will be released soon; we expect to increase the bandwidth to 19 Gb/s by that. The `dada` core code stills lacks documentation, but will also be released to the community. More analysis algorithms will be implemented.

## REFERENCES

- [1] P. Fratzl et al.: Position-Resolved Small-Angle X-ray Scattering of Complex Biological Materials, *J. Appl. Cryst.* 30, 1997.
- [2] M. Brennich et al.: Online data analysis at the ESRF bioSAXS beamline, BM29; *J. Appl. Cryst.* 49, 2016.
- [3] B. Weinhausen et al.: X-ray nano-diffraction on cytoskeletal networks, *New J. Phys.* 14, 2012.
- [4] M. Priebe et al.: Scanning X-Ray Nanodiffraction on *Dictyostelium discoideum*, *Biophys. J.* 107, 2014.
- [5] R. Wilke et al.: Hard X-ray imaging of bacterial cells: nano-diffraction and ptychographic reconstruction, *Optics Express* 20, 2012.
- [6] C. Riekkel et al.: Progress in micro- and nano-diffraction the the ESRF ID13 beamline, IOP Conf. S. 14, 2010.
- [7] M. Holler et al.: OMNY – A tOMography Nano crYo stage, *Review of Scientific Instruments* 89, 2018.
- [8] J.D. Nicolas et al.: Scanning X-ray diffraction on cardiac tissue: automatized data analysis and processing, *J. Synchrotron Rad.* 24, 2017.
- [9] A. Rothkirch (private communication).