

A Faster Software Pipeline for Crystal Centering



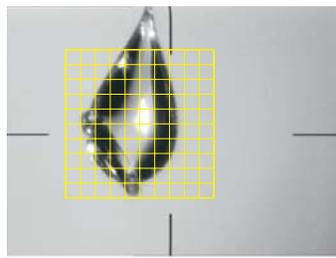
Nicholas Sauter & Ralf Grosse-Kunstleve

Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Overview

Among the benefits of using microfocused beams is the ability to obtain higher-signal, lower-mosaicity datasets from small crystal samples. With small crystals on the order of 1 to 10 μm there is a concomitant need to examine numerous specimens in order to assemble a complete dataset, making it highly desirable to automate the process of centering each sample in the microbeam. Various approaches have been proposed; here we focus on the use of low-dose X-rays to probe the diffracting power at several sample positions. A recent paper from SSRL (Song *et al.*, 2007) develops X-ray based auto-centering, but notes that the process may take up to 10 minutes for the smallest crystals, which presumably require very fine rastering to locate the sample. This outcome could potentially be improved by the use of a photon-counting pixel array detector such as the Pilatus-6M that supports a framing rate of 10 Hz, a tenfold improvement over the standard CCD-based detectors that have been the workhorses at all synchrotron beamlines. An accompanying challenge is to write software that quantifies the measured signal at a reasonably high turnaround rate, so that the diffraction analysis can keep pace with the rapid data acquisition pace needed for fine-grained coverage of the crystal. This poster presents a prototype software system affording a reasonable turnaround time of about 350 ms.

Low-Dose Rastering



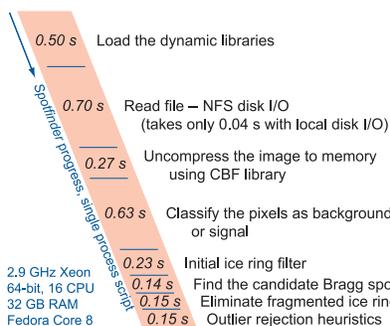
It has been possible to automatically center the sample-holding loop using videomicrographs, but it is difficult to visually identify small crystal samples within the loop. The alternative approach taken here is to scan the sample by translating the loop with respect to the microbeam, so as to collect a diffraction image at each translational position. A 10×10 grid is illustrated here. Presence of protein is scored based on the strength of the resulting Bragg diffraction.

Spotfinder as an Application

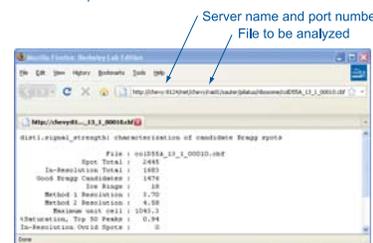
As shown by the SSRL team (Song *et al.*, 2007) it is too computationally intensive to quantify the diffraction data by integration when the purpose is simply to position the crystal optimally in the beam. It is sufficient to do a simple analysis of the potential Bragg spots with a program like *Spotfinder* (Zhang *et al.*, 2006), which returns either the total number of spots or the average intensity of the spots found within a given resolution range. Special heuristics are used to make sure that the spots being considered are likely Bragg spots, rather than ice-ring artifacts or other outliers.

Even this minimal calculation can take several seconds, leading us to ask how the overall process can be accelerated. A first step is to move toward a client/server architecture. A dedicated server saves the run-time overhead of importing the dynamic libraries, and can return the results over a convenient internet protocol such as http.

...and as a Server



2.9 GHz Xeon
64-bit, 16 CPU
32 GB RAM
Fedora Core 8



SINGLE-PROCESS SERVER: Implementation as a dedicated server completely circumvents the time-consuming import of the dynamic libraries that would otherwise take 0.5 s for each image analyzed.

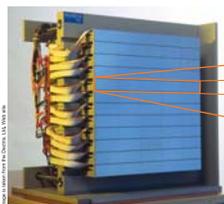
Results

Here we test the performance time to analyze the Bragg diffraction signal (if any) on 100 Pilatus-6M images. Images are contained in CBF-format disk files; therefore the tests are performed twice: once to evaluate the complete cycle time including disk I/O, and the second time to see performance when the operating system (or server) has already cached the files in memory.

TOTAL TURNAROUND FOR 100 IMAGES

Implementation	1st Run	2nd Run
Single process script No server NFS disk I/O	284 s	223 s (images cached by operating system)
Single process script No server Local disk I/O	227 s	222 s
Single process server Local disk I/O	173 s	172 s
Single process server Image server accessing local disk	186 s	155 s
Multiprocess server Image server accessing local disk	62 s	35 s
Multiprocess server Image server accessing local disk Results not marshalled for return to client	52 s	21 s

Modular Detectors and Multiprocessing



The Pilatus-6M pixel array detector produces a final image of 2527 vertical \times 2463 horizontal pixels. This arises from an array of 60 independent pixel array modules as shown, with small inter-module spacings. It is natural to break the whole image up along module boundaries for data processing.



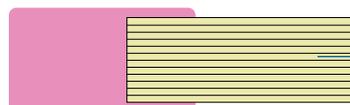
Spotfinder analysis of an image slice: pink—intermodule spacings; red—inactive pixels; green—brightest Bragg spots

- data objects and algorithms reimplemented to handle a single horizontal image slice
- 12 child processes are forked to simultaneously analyze all the slices
- results are marshalled afterward by the parent process
- OpenMP is not as effective a multiprocessing strategy as forking separate processes (for this application)
- slices are individually addressed by a special resource syntax: `file://data/dataset/colD55A_13_1_00010.cbf?slice=4`
- extension to hypertext protocol: `http://dataserver:8080/data/dataset/colD55A_13_1_00010.cbf?slice=4`

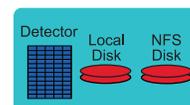
Overall Data Processing Architecture



http



http



MULTIPROCESS SERVER: Forks 12 child processes to analyze image slices. Waits for the results, packages them up, and returns them to the client using an HTTP-based protocol. Asynchronous image analysis takes about 210 ms, but an additional 120 ms wait is typically required in order to coordinate the activities of all child process, so that a single result can be sent back.

DATA PROCESSING CLIENT: Contacts a server with a request to measure the diffraction strength on an individual image that has just been acquired. Generally, the client is same the process that controls data collection and displays the crystal centering results to the beamline user.

IMAGE SERVER: If the data are organized as files, the server is responsible for disk I/O, image decompression, and caching the data in memory. Requests for individual data slices are fulfilled based on cached data. Caching 100 images takes 2.5 GB memory, but this is easily accommodated on today's machines. Trials to date used a same-host server.

Future Directions

- remote-host image server
- images served directly from Pilatus host—no file I/O
- test using shared memory instead of sockets

More Information

Pothiniemi SB, Strutz T & Lamzin VS. Automated detection and centering of cryocooled protein crystals. *Acta Cryst. D62*, 1358-1368 (2006).

Song J, Mathew D, Jacob SA, Corbett L, Moorhead P & Soltis SM. Diffraction-based automated crystal centering. *J. Synchrotron Rad.* 14, 191-195 (2007).

Zhang Z, Sauter NK, van den Bedem H, Snel G & Deacon AM. Automated diffraction analysis and spot searching for high-throughput crystal screening. *J. Appl. Cryst.* 39, 112-119 (2006).

Acknowledgements:

Work was supported by the US Department of Energy under Contract No. DE-AC02-05CH11231. NS is funded through NIH/NIAMS grant GM77071. RGK is funded through NIH/NIAMS grant GM063210.