Cluster Computing 1 (2000) 1–33

Using Computational Grid Capabilities to Enhance the Capability of an X-Ray Source for Structural Biology

Gregor von Laszewski [a,*] Mary L. Westbrook [b] Craig Barnes [d] Ian Foster [a]
Edwin M. Westbrook [c]

[a] *Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL*
E-mail: gregor@mcs.anl.gov
[b] *Electronics and Computing Technologies, Argonne National Laboratory, Argonne, IL*
[c] *Biosciences Division, Argonne National Laboratory, Argonne, IL*
[d] *University of Illinois at Chicago, Chicago, IL*

The Advanced Photon Source at Argonne National Laboratory enables structural biologists to perform state-of-the-art crystallography diffraction experiments with high-intensity X-rays. The data gathered during such experiments is used to determine the molecular structure of macromolecules to enhance, for example, the capabilities of modern drug design for basic and applied research.

The steps involved in obtaining a complete structure are computationally intensive and require the proper adjustment of a considerable number of parameters that are not known a priori. Thus, it is advantageous to develop a computational infrastructure for solving the numerically complex problems quickly, in order to enable quasi-real-time information discovery and computational steering. Specifically, we propose that the time-consuming calculations be performed in a "computational grid" accessing a large number of state-of-the-art computational facilities. Furthermore, we envision that experiments could be conducted by researchers at their home institution via remote steering while a beamline technician performs the actual experiment; such an approach would be cost-efficient for the user.

We conducted a case study involving multiple tasks of a structural biologist, including data acquisition, data reduction, solution of the phase problem, and calculation of the final result—an electron density map, which is subsequently used for modeling of the molecular structure.

We developed a parallel program for the data reduction phase that reduces the turnaround time significantly. We also distributed the solution of the phase problem in order to obtain the resulting electron density map more quickly. We used the GUSTO testbed provided by the Globus metacomputing project as the source of the

* Corresponding author.

necessary state-of-the-art computational resources, including workstation clusters.

Keywords: Metacomputing; Grid-enabled application; Structural biology; X-ray diffraction.

1. Introduction

High-brilliance X-ray sources promise to revolutionize the discipline of structural biology by providing imaging capabilities with unprecedented spatial and temporal resolution. However, the effective use of these capabilities requires the ability to collect, archive, analyze, and visualize orders of magnitude more data than is currently possible. This paper describes work conducted in a Department of Energy Grand Challenge project that seeks to produce innovations in methods, algorithms, and software that will allow this data to be utilized fully by scientists.

The focus of this paper is a case study in which advanced computational infrastructure components are applied to complete the set of tasks faced by a structural biologist, ranging from acquiring the data, to solving the phase problem, to calculating the final result—an electron density map. As part of this study, we developed a parallel program that reduces the turnaround time for the data acquisition phase dramatically. We also distributed the task of solving the phase problem in order to obtain the resulting electron density map more quickly.

The results presented in this paper provide useful data about the utility of clusters and computational grids in general for such applications.

The paper is organized as follows We first discuss the tasks performed in a typical structural biology experiment. We then analyze the experiment to derive requirements for a *grid-enabled* structural biology environment. We outline the tasks needed to develop and improve the infrastructure to fulfill these requirements. Performance data for the various steps in the processing pipeline are obtained by combining two unique state-of-the-art infrastructure components: the Advanced Photon Source (APS) and the GUSTO testbed provided by the Globus project [18]. We demonstrate that the turnaround time for conducting APS experiments can be improved significantly by using grid resources. Finally, we summarize our results and conclude the paper by pointing out opportunities for further research.

2. Structural Biology and X-Ray Diffraction

Knowledge of accurate molecular structures is a prerequisite for basic research in molecular or cell biology and for applied research, such as modern drug design and structure-based functional studies to aid the development of effective therapeutic agents. Brilliant synchrotron X-ray beamlines are powerful tools to determine the structures of large macromolecules, such as proteins. X-ray crystallography is a technique that exploits the X-rays diffraction from a crystal. Based on the diffraction pattern obtained from the periodic assembly of molecules in the crystal being examined, the electron density is reconstructed. The last step of the structure determination then is to progressively refine a molecular model that corresponds to the experimental electron density. In summary, we distinguish four main steps in a typical X-ray diffraction experiment:

Step 0: Experiment planning and preparation
Step 1: Data collection
Step 2: Data reduction
Step 3: Data analysis and model construction

The result of this multistep process is a quite accurate molecular structure. X-ray crystallography can reliably provide the answer to many structure-related questions, from global folds to atomic details of bonding. In contrast to other methods, no size limitation exists for the molecule or complex to be studied [19].

In the following sections, we analyze each of the steps listed above and determine requirements in order to derive guidelines for creating a computational infrastructure that will support the experimentation facilities for structural biology.

The case study we have chosen focuses on the 3-oxo-Δ^5-steroid isomerase, also called Δ^5-3-ketosteroid isomerase, or KSI. KSI has been the focus of extensive biochemical and chemical study for over forty years, with the objective of understanding the catalytic mechanism and the basis for its extraordinary efficiency. Determination of the crystallographic structure of KSI remained elusive, however, until recently with the possibility of Multiple-energy Anomalous Dispersion (MAD) phasing at an Advanced Photon Source beamline. The result of the analysis process is shown in Figure 1. It is far beyond the scope of this paper to give a more comprehensive description of X-ray crystallography and the details of the case study. For more information we refer to [16,2]. In Table 1 we list the execution times and data sizes as they appear while using common

Table 1
Sequential execution times and data sizes as they appeared during the KSI case study

	Time on RS/6000	Data
Step 0:	weeks/month	-
Step 1:	4 × 675 seconds	18 Gbytes
Step 2:	4 × 973 seconds	<1 Gbyte
Step 3:	4 × 45 minutes	< 50 Mbytes
Step 1 + 2 + 3:	appr. 5 hours 20 minutes	

sequential programs and methods during the KSI case study.

Besides the case study we performed additional experiments with other datasets. We also tested the algorithm with tetragonal chicken eggwhite lysozyme and triclinic cytochrome-c. These datasets are representative of the wide range of experiments that are conducted at macromolecular crystallography beamlines. This includes "small", "medium", and "large" problems.

3. Infrastructure Design and Requirements Analysis

The ideal infrastructure that allows a structural biologist to perform the steps in a crystallographic experiment is shown in Figure 2. This environment is depicted from the point of view of the scientist, with as many details as possible hidden.

While analyzing our case study of an X-ray experiment, we determined requirements that are reflected in the infrastructure design. First, the structural biologists must have easy access to, and use of, the computational infrastructure. Second, the environment should provide a great deal of flexibility to encompass the different experiment configurations of the large and diverse user community. Third, the computer analysis time should be minimal, in order to achieve the highest possible utilization by the users for the current running experiment at the beamline, as well as postprocessing of the experiment data with different parameter settings.

The increased technical complexity of today's detectors and compute resources motivates the development of easy-to-use interfaces for interacting with the compute environment. This environment also includes programs that provide a simple interface for controlling and monitoring the experiment. The interface to

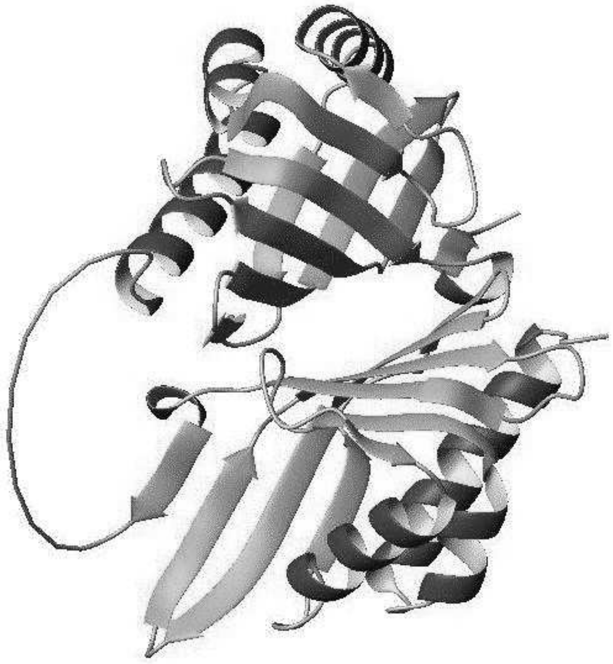

Figure 1. *The KSI structure diagram, a result of the case study.*

state-of-the-art software will ultimately determine the success of the equipment. The goal is to enable the scientist to concentrate on the scientific problem at hand, rather than spending time learning about often-complex compute environments.

3.1. Experiment Planning and Preparation

Step 0. Before the actual experiment is conducted, plans are made, literature searches are conducted, and supporting data is retrieved from online databases. A large amount of interaction takes place between members of the experiment team, resulting in a precise plan to conduct a diffraction experiment and to generate appropriate crystals for the diffraction experiments.

The need for a collaborative environment is obvious. The researchers are often at different geographical locations. The literature search is done at local libraries; but increasingly, with the popularity of the Web, more and more articles are being obtained online. Certainly, online databases located throughout the

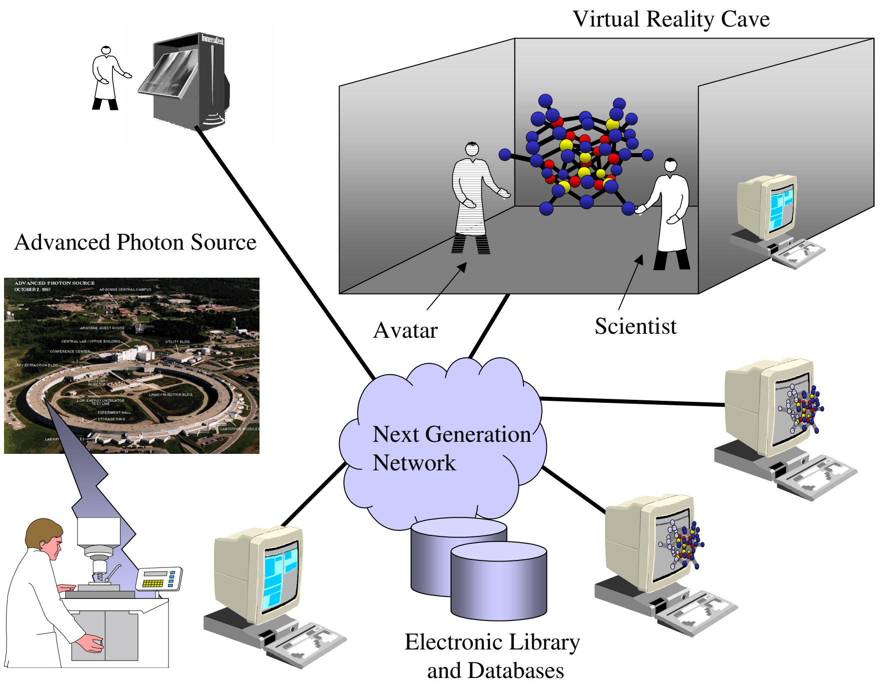

Figure 2. *The ideal infrastructure design for an ideal computational environment from the point of view of the scientist. The design hides the technical details of the hardware components, enabling the structural biologist to concentrate on the science.*

world provide access to data useful for the initial research. Properly used, collaborative tools can significantly improve the scientific idea exchange and reduce the necessity for travel to make scientific information exchange possible.

3.2. Data Collection

Step 1. The Structural Biology Center (SBC) at the APS serves as a national user facility for macromolecular crystallography [15]. The two major design goals have been to

1. provide extremely brilliant beamlines for studies that cannot be performed elsewhere, and
2. maintain a high user throughput in order to serve the largest number of crystallographers possible.

Therefore, the SBC emphasizes both speed and brightness. To this end, all SBC components—optics, detector, controls/data acquisition, and data analysis—were designed explicitly to work together as an integrated system. Ideally, data analysis would proceed *on the fly*, at the same rate as data acquisition, for the highest level of scientific productivity.

The experiment setup at the SBC beamline is as follows. An X-ray beam of a characteristic wavelength is used to perform the diffraction experiment. First, intense X-ray radiation produced by a synchrotron is monochromated and focused. Then it passes through the sample crystal mounted on a pin on a goniometer, which permits the crystal to be positioned in different orientations in the beam. The diffracted X-rays are recorded with a CCD detector.

The APS CCD X-ray detector consists of nine CCD elements with a total active area of 210 by 210 mm^2. Two readout modes are supported. In the full-resolution readout mode, an image of 18 Mbytes (3072 $\times$ 3072 pixels) can be read out in 1.8 seconds, while in binned mode, an image of size 4.6 Mbytes (1536 $\times$ 1536 pixels) can be read in 0.45 seconds [13,17,14]. Each pixel is represented by a 16-bit integer value. A typical experiment will gather between 250 and 1000 images, called a *scan*. Thus, the maximum size to store the images for one scan is currently 18 Gbyte. The minimal *acquisition time* with any electronic detector equals the exposure time added to the detector readout time. Currently, the difference between the actual acquisition time of 4 seconds and the expected minimal acquisition time of 2.8 seconds (for a 0.1 degree image width and full-resolution images) can be attributed to the motor motions needed to position and scan the crystal. As these motor motions are tuned, the minimum acquisition times are gradually being approached. Figure 3 shows an example image taken during a scan of KSI. The spots shown in Figure 3 are created by constructive inference described by Bragg's law [1,5]. These spots are often referred to as *Bragg spots.*

For the KSI case study it was necessary to collect four datasets, each with a different wavelength. Each of the four datasets contained 450 images and was recorded in 1680 seconds.

With future upgrades to the electronics, we expect the readout time to drop by a factor of four. Thus, each of the complete KSI datasetes could than be collected in 1176 seconds. Because of the rapid improvement in the CCD market, we believe that the readout time will be further reduced.

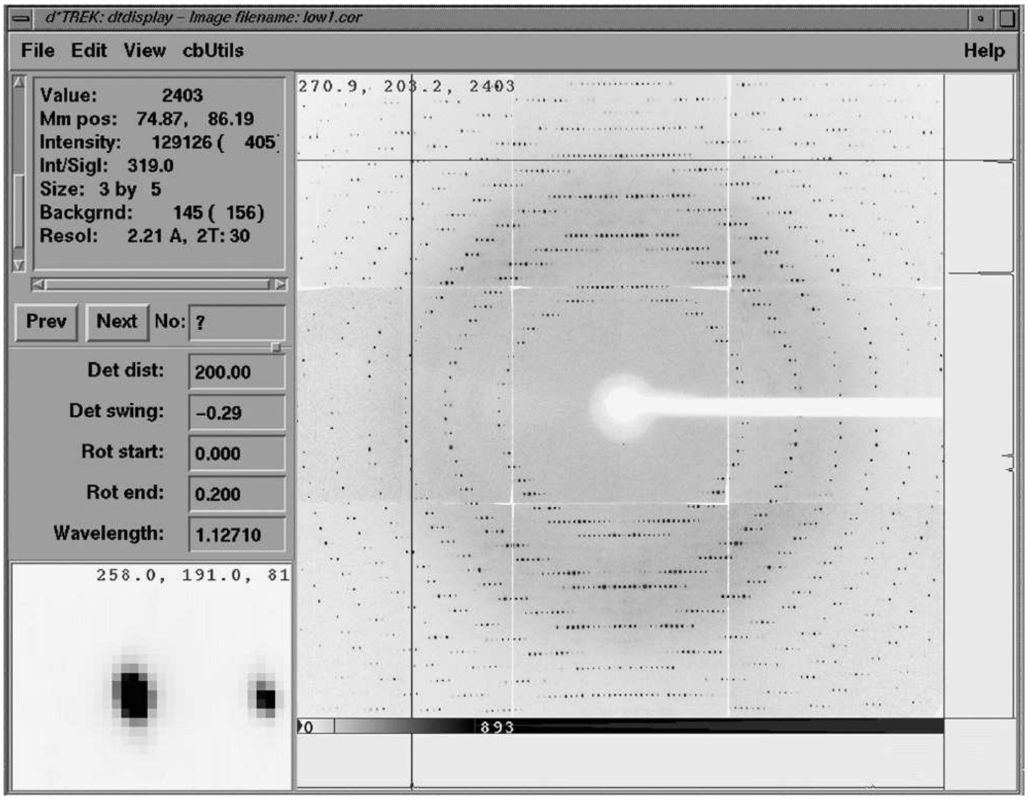

Figure 3. *A sample diffraction image of KSI taken with the SBC 9-element CCD camera. The image has the dimensions of 3072 × 3072 pixels and is 18 Mbytes large.*

High-performance networked computing systems support the aggregate tasks of beamline control, data acquisition, data analysis, data archiving, and data visualization. SBC systems are multihomed, with both ATM OC-3 (155 Mbps) and 10 Mbps Ethernet connectivity to support "slow" control functions. Image data is transferred from the APS detector to memory on the beamline symmetric multiprocessor, an SGI Challenge L Server, currently equipped with four 250 MHz MIPS R4400 processors, via the high-performance parallel interface (HIPPI) network protocol. Once the image data is in SGI memory, the data is transferred to a RAID disk array and may be analyzed, archived, and visualized locally or from distributed Unix-based workstations on the beamlines via ATM OC-3 connections [14]

3.3. Data Reduction

Step 2. To allow the determination of the positions of the atoms in a crystal, the precise position of the Bragg spots have to be identified, and all Bragg peaks must be accurately determined in space and time. For this purpose, a 3D shape-recognition algorithm is applied to each spot, and the data is reduced from the images to the position of all Bragg spots. Clearly, crystallographic data reduction is a computationally intensive process, because of the size of the images and the enormous number of Bragg diffraction peaks (on the order of 10^6). The entire process can be tedious and slow, and may need to be iterated with varied parametric settings before it is completed properly.

Two modes of operations are important for the structural biologist: (1) a mode that enables one to deliver the data reduction shortly after the data collection is completed, and (2) a mode that enables one to redo the reduction step with modified parameter settings. The latter mode is important in case a mistake during the parameter setup prevents the detection of all necessary Bragg spots on the images. In Section 4.2 we present a parallel data reduction algorithm that dramatically reduces the time needed for this step.

Besides the obvious need for a powerful compute infrastructure, another requirement arises from the physical experiment setup. Because of the hazardous and often unpleasant environment, remote operation is desirable. With remote operation, the facility can maintain a small but well-trained team of beamline staff experimentalists. This approach offers several benefits. It reduces the operational and user-specific cost and minimizes travel cost to the facility. It allows novice user groups to gain access to a unique facility such as the APS. Furthermore, it increases the access time to the beamline while minimizing the effort to set up experiments by the trained experts.

3.4. Data Analysis and Model Construction

Step 3. In this last step of an X-ray experiment, a model for the electron density map is constructed by using the output of the previous steps. Once the exact positions of the Bragg peaks are known, a best fit to the available diffraction data is found that does not violate the physical reality. In practice, X-ray structure determination is not an absolute technique. In many cases one must rely on other information besides the diffraction data. One way to obtain additional information is to use multiple wavelength anomalous dispersion (MAD). In this technique, diffraction changes are induced in the atomic scattering factor of a heavy atom bound to the protein by measuring diffraction data at a number of different X-ray energies where the anomalous scattering factors of the heavy atom are significantly different from one another.

We used a well-known phase-solving algorithm (SOLVE) [11], which is applied repeatedly to find the correct parameter setting. SOLVE is designed to automate the steps of macromolecular structure determination. It scales data, solves Patterson functions, calculates difference Fouriers, looks at a native Fourier to see whether there are distinct solvent and protein regions, and can score partial MAD [7] solutions to build up a complete solution automatically.

Because of the complex geometrical structure of the solutions, the need arises to display and analyze the structure with other colleagues. It is not sufficient just to provide a collaborative tool with a white board or video camera. Instead multiple user require simultaneous access to specialized datasets displayed in three dimensions.

3.5. Additional Requirements

The requirements posed by the structural biologists result in implicit requirements for the computational environment.

Real-Time Data Processing. Because of the uniqueness of the APS facility, access time at the beamlines is limited. Thus, developing a quasi-real-time data processing framework is needed. The faster scientists can obtain information about the state of the experiment and the analysis of the experiment data, the faster they can react to erroneous experiment conditions. Real-time data processing also provides the ability to do a rapid comparison with data gathered during previous experiments prepared with different parameter values.

Quality of Service. An additional consequence of the unique experimental facility and its limited access is that the computational environment should be, to some extent, fault tolerant. Real-time processing depends on at least a minimal degree of quality of service.

Reservation. To ensure timely execution of the program, it is necessary to perform the reservation of compute and network resources as part of a quality-of-service request.

Local Specialized Services. Because of the financial cost involved with licensing software on a particular computer, it must be possible to reserve a compute resource on which the software is installed. It also must be possible to specify mapping constraints to the programs to be executed. These additional requirements motivated us to use the GUSTO testbed (part of the Globus project), since it addresses these issues [4].

4. Parallel System and Algorithm Design

In this section we outline the programs we developed in order to support the framework shown in Figure 2. We start with the remote system control and the collaborative environment. Then we describe the parallel data reduction algorithm and outline how we used a compute cluster to execute the structure determination program in parallel.

4.1. Remote System Control and Collaborative Environment

As pointed out earlier, a major goal is to design the interface to the control and analysis software to be simple for the computational scientist. Online control of the experiment should be possible from various points and through various devices, including workstations on the experimentation floor as well as at collaborating sites. Given the nature of the resulting structures, stereo viewing capabilities are essential.

To this end, we have designed two specialized stereo graphic tools, depicted in Figures 4 (a) and (b). The first tool can animate the photon impact on the CCD detector during the experiment. This tool was especially useful in developing the parallel data reduction algorithm and in determining appropriate parameter settings for the data reduction step. The second tool can display the crystal structure in a ball-and-stick diagram for large molecules.

4.2. Parallel Data Reduction

Typically, X-ray crystallography data reduction executes the following steps:

1. *Find* and identify intense diffraction peaks (Bragg spots).

2. *Index* the peaks to discover the crystal unit cell dimensions, crystal orientation, and other important structural parameters.

3. *Refine* the crystal, detector, goniometer, and x-ray source parameters.

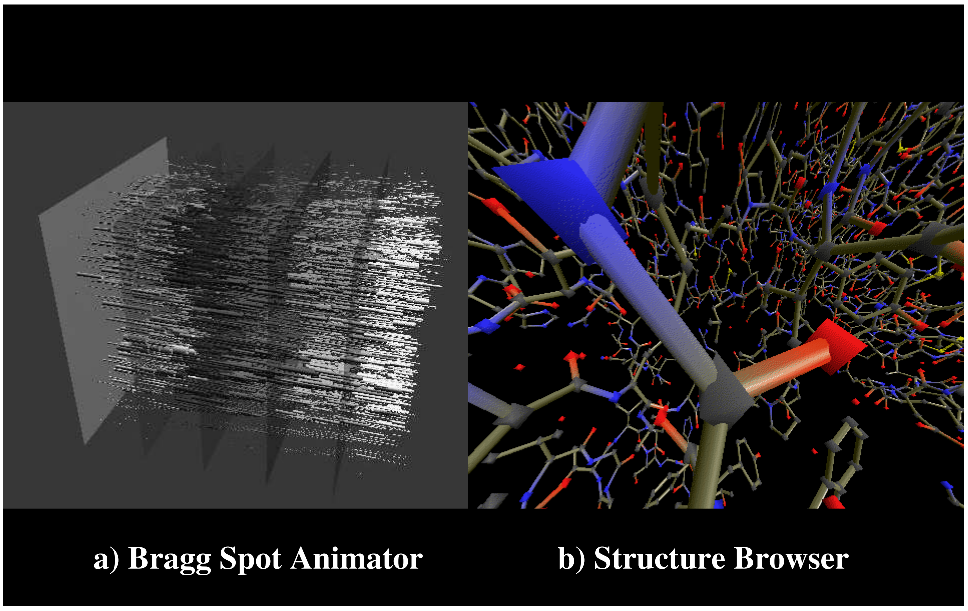

Figure 4. *Screen dumps from the CAVE applications developed to assist the structural biologist during data analysis. Figure (a) shows the Bragg spots as they develop during time in space. The colors indicate on which parallel processor the Bragg spot was analyzed (using the parallel algorithm described later). Figure (b) shows a "walk" through a molecule to better see the structure.*

4. *Integrate* over the intensities of the reflections that appear on the images in the data set obtaining a list of reflections. The result of this step is a reduced list of reflections.

5. *Merge and filter* optionally the reflection lists to perform, for example, resolution cutoffs.

6. Calculate and apply *scale* factors to different batches of reflections, average symmetry equivalent reflections, calculate *merging* and completeness statistics, and create a reflection list of unique reflections.

Steps 1–3 are performed with data from a single image, step 4 is performed on the whole dataset; and steps 5 and 6 are performed on the list of reflections found by the integration step. We have analyzed these steps and determined that the integration step is the most time consuming in the current operational use. Thus, our efforts to derive a parallel algorithm concentrated on the development of a parallel integration algorithm.

We derived a parallel algorithm from a sequential code, called d*trek, available at the SBC (see Section 7). The parallel extensions we made have been integrated into the production version of the sequential code.

Sequential Algorithm As mentioned earlier, the diffraction data appear at varying intensity on multiple images as they are rotated into, through, and out of the diffracting condition. In the language of the crystallographer, the complete set of images gathered during an experiment is called a *scan*. The sequential integration algorithm defines a *shoebox* as a 3D volume containing the intensity information of a single spot over its diffracting range, which contains a subset of images of the scan. The algorithm predicts the range of images on which the spot appears.

After the image is detected on which the spot disappears, the program defines a complete 3D "profile" of the spot for a subsequent profile analysis. If the spot extends across too many images (predefined to include 50 images for the sequential algorithm), it can be rejected. This situation usually occurs because the spot was too close to the projected rotation axis and therefore may contain errors. The parameter defining the cutoff value is called the *shoeboxlimit*. Greater efficiency of the sequential and parallel algorithm is achieved by decreasing the shoeboxlimit. If such a spot occurs over a large number of images, it is likely to be eliminated in the steps following the integration. We varied the shoeboxlimit parameter while observing dataset completeness, which is defined as the percentage of reflections found without introducing the shoeboxlmit. Since the sequential algorthim defines also a predefined "cutoff" radius, only 96% of all detected Bragg spots are actually used for the analysis.

Figure 5 shows the variation of the dataset completeness while varying the shoeboxlimit for the KSI high-energy dataset. We found that 25 is a conservative and well-suited shoeboxlimit, resulting in a detection of over 95% of the reflections and providing high accuracy in a single scan, which is sufficient for the structural biologist to solve the structure accurately.

Based on the sequential algorithm, two distinct parallel algorithms can be developed. The first algorithm divides the scan (sequences of images) among a

Figure 5. *Variation of the shoeboxlimit for the KSI "high" energy dataset versus the dataset completeness, which is the percentage of all possible reflections that can be detected.*

group of processors. The second algorithm divides the images in small subregions, each of which is assigned to one processor.

Parallel Algorithm Based on Distributing a Sequence of Images To improve the performance exhibited by the sequential integration program, the parallel algorithm divides each dataset into subscans and processes each subscan on a different processor, thereby achieving an accelerated processing rate. Data are subsequently merged and scaled with a fast merging algorithm [9]. Figure 6 depicts the distribution of subscans onto multiple processors.

In the parallel algorithm, a boundary condition arises when a scan is divided into subscans. We start out by dividing the number of images evenly among the processors. We call each of the sets the *base image set of a processor*, or the *base image set*.

At the end of a base image set of a processor, some shoeboxes are still *ac-*

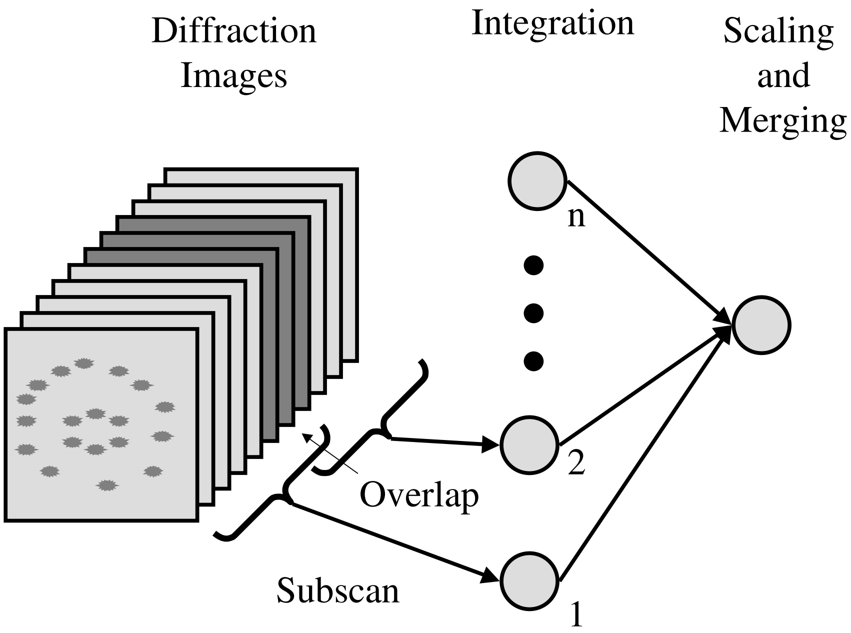

Figure 6. *Distribution of subscans onto multiple processors. Overlap regions have to be considered in order to achieve scientifically sound results.*

tive; that is, the spot is still visible on a number of following images. In order to guarantee numerical stability of the method, a dynamic boundary between consecutive subscans placed on different processors is used to prevent early shoebox termination. The dynamic boundary of each subscan is extended to complete all active shoeboxes or to continue until the shoeboxlimit has been reached.

To calculate the memory overhead, let N_{base} define the number of images in a base image set of a processor, and let N_{limit} denote the shoeboxlimit. Then the maximum number of images stored per processor is $N_{base}+N_{limit}$. Let N_p denote the numbers of processors used in the parallel algorithm, and let N_I denote the total number of images in the scan.

Then, the memory overhead for the storage of the parallel algorithm on MIMD machines is at most

$$N_p \frac{N_{limit}}{N_I}.$$

Thus, it is desirable to keep the shoeboxlimit as small as possible while guaranteeing dataset completeness.

One consequence of this parallel algorithm is that it performs best when the data collection is nearly completed and the majority of the images are available before the integration begins. Hence, it supports very well a quick analysis of the data with a varying set of parameters.

Parallel Algorithm Based on the Subdivision of the Image Area We also considered parallelizing the sequential algorithm while partitioning the images into a number of different *image areas*, or *subimages*, with each subimage being processed on a different processor, as depicted in Figure 7. We decided not to develop this algorithm for the following reasons.

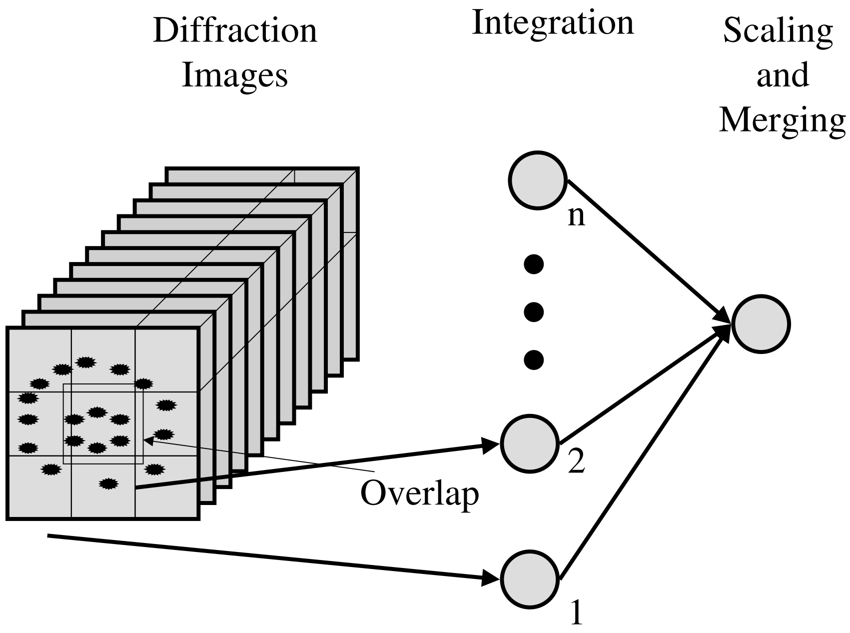

Figure 7. *Distribution of the images in subimages onto multiple processors.*

First, the algorithm requires extensive modification to the original sequential code, resulting in a significant divergence from the original algorithm. Such diver-

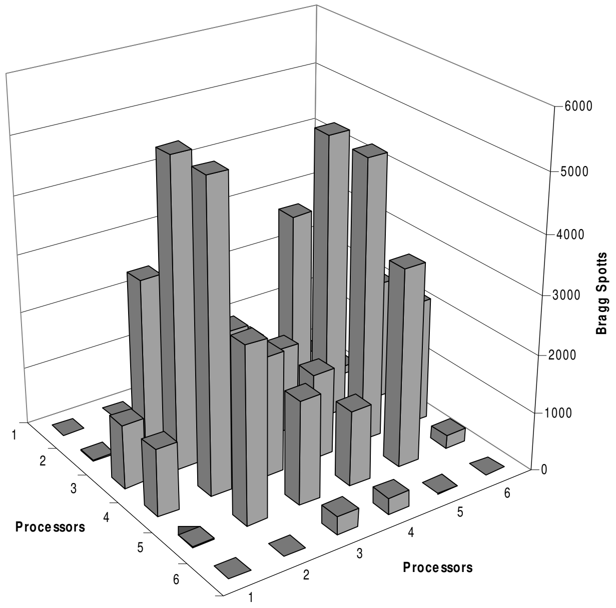

Figure 8. *Distribution of the Bragg spots as they appear on the detector plate.*

gence would have prevented us from upgrading to new versions of the sequential algorithms as they became available, and maintaining the overall program would have involved an immense overhead.

Second, one of the biggest problems with this algorithm is that the crystal, detector, goniometer, and source parameters are re-refined periodically (typically every two to five images), to accommodate small inaccuracies in definition of the experiment or small physical errors in apparatus calibration. This refinement is corrected with the help of observed reflections from all areas of the detector. Hence, considerable overhead for a global communication step between each processor occurs every two to five images. A further problem related to the partitioning of the image is that the area of a reflections is approximately 10×10 pixels wide. When a reflection is in a subimage boundary, it is calculated in the neighboring processors at least twice. Special care has to be taken in order to remove these double-calculated points. Because of the shift in x-y direction of a reflection during the image-taking process, the boundary should be chosen larger, resulting in lower speedup. Third, because of the irregular distribution of the Bragg spots, which are directly proportional to the calculation time, a considerable load imbalance is introduced. Figure 8 shows the number of Bragg spots occurring during the experiment at a particular region of the detector. Clearly, in order to achieve good efficiency, a load balancing algorithm should be developed. This is not an easy task because we do not know a priory the distribution of the Bragg spots.

We tested the speedup possible with this algorithm by decomposing the 18 Mbyte 3072×3072 square pixel images into nine 1024×1024 square pixel images using a large dataset. We distributed each subimage on different processors and observed only a factor of 4.5 speedup on nine processors. The speedup will decrease as the number of processors increases. Furthermore, our results confirmed the need for refinement with reflections from all areas of the detector to produce scientifically sound results. Hence, we have not parallelized the code in this manner.

Changes in the Control Flow Having identified a suitable parallel algorithm for the data reduction step, we now list the necessary steps to start the parallel algorithm.

1. Determine parameters for the integration
2. Decide how many processors should be used
3. Decide on which machines the processors are located
4. Determine the set of images needed for the processor
5. Make the images known to the processor
6. Do the integration on each processor in parallel
7. Collect the result from each processor
8. Display the result

Combining the sequential algorithm and the parallel algorithm to steer the computation, as depicted in Figure 9, allows for a better utilization of the resources. The sequential algorithm is used to quickly decide whether the experiment has been started successfully and the data produced are sufficient for the analysis performing in consecutive steps. While moving the data reduction to a remote computer, more powerful graphical analysis tools can be used. In addition, a gateway to remote collaboration is provided.

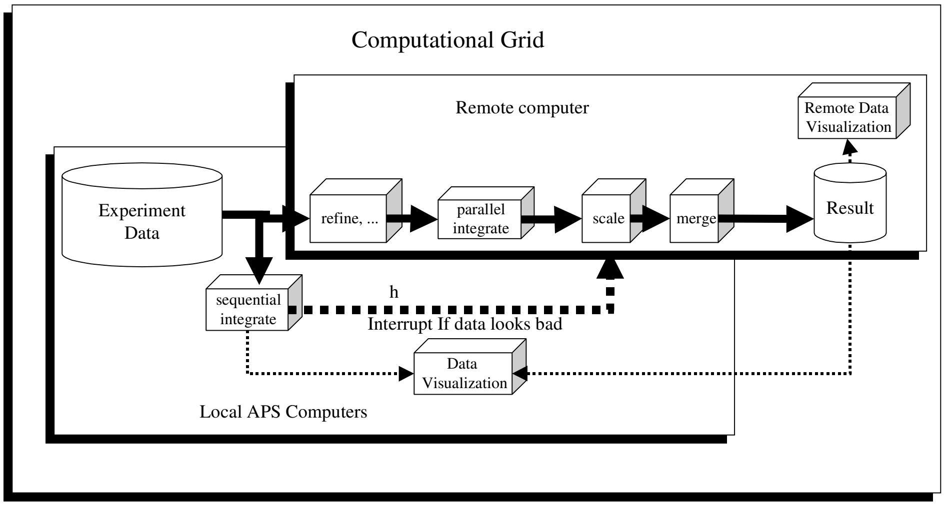

Figure 9. *The control of the experiment and the data reduction can be performed while combining the sequential and the parallel algorithm.*

4.3. Parallel Data Analysis

After the data reduction step is completed, the structural biologist must create a model. We used the program SOLVE in our case study and were able to run it in parallel on different input parameters. For the KSI structure determination we needed four instantiations of SOLVE to find the appropriate answer. Each instantiation required 45 minutes on a single-processor SGI Origin2000 workstation. Hence, we reduced the calculation time fourfold. Other algorithms, such as Shake-n-Bake (SnB), used for solving the structure are described elsewhere [3]. The choice of the algorithm depends on the structure to be analyzed. We intend to integrate the selection of appropriate solve routine as part of a graphical user interface.

5. Performance Evaluation

The programs based on the algorithms described above are written in such a way that they can be executed on diverse compute resources. This includes workstation clusters, supercomputers, and metacomputing environments.

As part of this project, we explored technologies to enable direct access from the Advanced Photon Source to Argonne's Mathematics and Computer Science Division (MCS) that houses a variety of compute resources. Specifically, we have pioneered access to groups of workstations and supercomputers as part of the GUSTO testbed, which is supported by the Globus project [18].

5.1. Hardware Infrastructure

We focused in the performance evaluation on the use of compute nodes that are part of an IBM SP-2 and a small Origin2000. Each compute node on the SP-2 is based on an RS/6000 processor and has a main memory of 512 Mbytes. The operating system is AIX 4.2. We conducted experiments with up to 64 processors on this machine. The maximum number of processors on the Origin is sixteen. The Origin is based on the MIPS R10000 chipset running IRIX6.4. The total amount of memory for this machine is 4 Gbyte. Each of the systems has sufficiently sized local disk space to store all of the images obtained during an experiment. The disks are able to hold multiple datasets, each of which can be at least 18 Gbytes. The large datasets can be archived on a 20-Tbyte storage robot attached to both systems. While moving the data-reduction and analysis process away from the beamline computer systems, congestion on the experiment hardware is alleviated. This kind of "computeservice" is especially important for future scientific projects that lack the budgets to purchase and maintain a large system to facilitate the work locally. Other projects such as the SnB and CMT (Computed Microtomography) projects [12], have also benefited from this pioneering work [20].

5.2. Network Infrastructure and File Transfer

To perform the data reduction on a remote compute server, the image data must be transferred to the remote machine. The transfer of up to 1000 of the image files, 18 Mbytes in size, in quasi-real time is possible only over high-performance networks (100 Mbps+) and between systems equipped with high-performance disks. The transfer of images can be conducted in parallel with the data acquisition. As soon as an image is written to disk, it is transferred to the remote system. As pointed out before, the limiting factor is the wallclock time needed to acquire an image from the detector. Thus, the transfer time should not exceed 2.8 seconds in order to keep up with the image acquisition.

Figure 10 shows the connectivity for the SBC computing systems to the ANL labwide ATM network infrastructure, including Argonnes supercomputers, and external networks, such as the ESnet and the vBNS. We have measured 6.9 Mbytes per second FTP image data transfer rates from ATM OC-3 capable SBC-CAT beamline systems to ATM OC-3 connected systems in MCS. This corresponds to 2.6 seconds per 18 Mbyte image, fulfilling the request to keep up with the acquisition rate [14].

Table 2 depicts elapsed times to transfer data from Sector 19 ATM OC-3 connected systems using FTP, over the ANL labwide ATM network, to ATM OC-3 connected systems in Argonne's Mathematics and Computerscienece Division (MCS). For the KSI datasets the acquisition could proceeded *on-the-fly* with 1 second image acquisition time achieved with the next generation of detectors. The the current image acquisition is 2.8 seconds and will be improved shortly. Thus, the program developed is well suited to support the next generation detector systems.

5.3. Parallel Data-Reduction Performance

Table 3, Table 4, Figure 11, and Figure 12 show the performance of the parallel data reduction algorithm applied to the KSI case study on the Origin2000 and the RS/6000-based compute nodes. The speedup on the SGI Origin2000 and the RS/6000 is similar, although the overall processing speed is faster on the Origin. Because of overlap between subscans on different processes the efficiency levels off while using 8 to 20 processors. This decrease of efficiency is less of a problem for datasets with many images, because the overlap regions are a

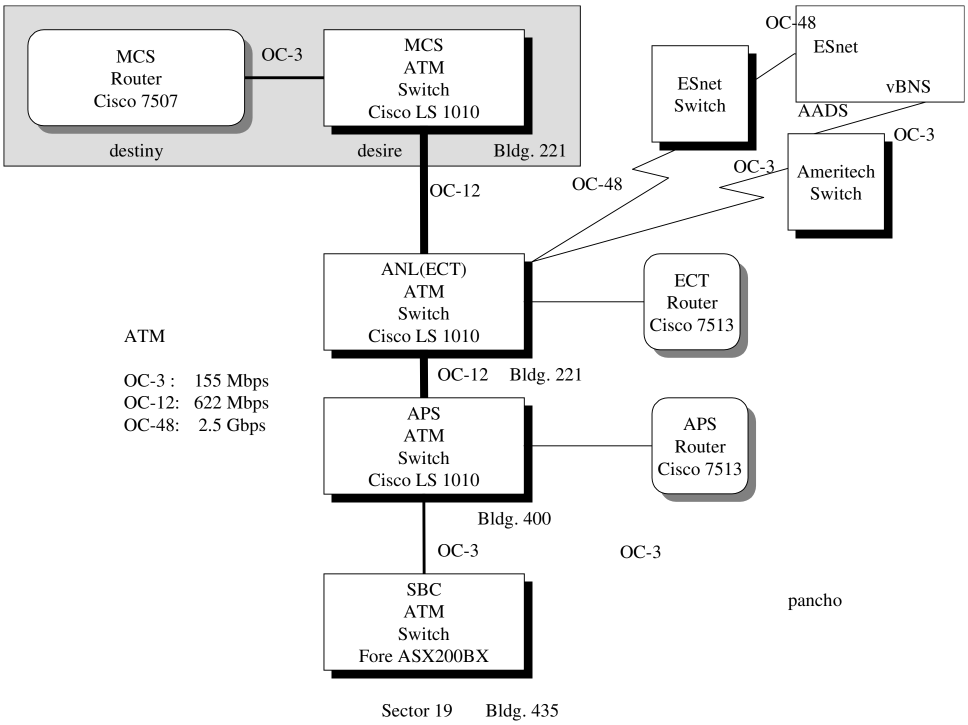

Figure 10. *The ANL labwide ATM network infrastructure with connection to the Internet.*

Table 2
File transfer performance between the APS Sector 19 and the MCS supercomputers for different datasets.

Data Set	N_I	Image Size in Mbyte	FTP Time in s per Image	FTP Time in s per Data Set
lys023s1a	384	18.0	2.60	999
cytochrome c	720	18.0	2.60	1872
ksi-high	450	4.5	0.65	294
ksi-infl	450	4.5	0.65	294
ksi-low	450	4.5	0.65	294
ksi-peak	450	4.5	0.65	294

smaller fraction of the entire data set. Thus, we conclude that a small powerful workstation cluster is well suited for many experiments.

A significant increase in data-processing speed is achieved by using multiple processors for analysis. For the cytochrome c data, as one example, the CPU time to process the complete dataset of 720 (18 Mbyte) images, using the sequential integration program, on a single RS/6000 processor was 3,860 seconds. This represents an integration rate of 5.4 s/image for the sequential algorithm. Distribution of integration across eight RS/6000 workstations requires an elapsed time of 614 seconds, or 10 minutes, which represents an effective integration rate of 0.86 s/image and a speedup better than six-fold. More important, reduction of this process from more than 1 hour to just 10 minutes is a meaningful improvement in the actual use of a synchrotron beamline experiment.

Larger numbers of processors are useful when multiple datasets for a MAD analysis (see Section 3.4) are analyzed in parallel, as it can be done for the KSI structure. This dataset is referred to in the tables and figures as *KSI-all.* It contains four datasets taken at different phases. These sets are referred to as KSI *low, infl, high* and *peak.*

With the parallel integration program, all four datasets are processed simultaneously across multiple processors, resulting in a significant acceleration over the sequential processing of the data. For example, it requires 1590 seconds to process a single KSI data set using the sequential integration program, but only 501 seconds (less than 9 minutes) to process all four data sets in parallel distributed over 16 RS/6000 workstations. This represents a three-fold acceleration over a single KSI data set done sequentially, and a twelve-fold acceleration in processing speed when compared with the time to analyze the four data sets sequentially. As more and more processors are applied to the task, the overall analysis time continues to decrease. With 64 RS/6000 processors, the four KSI MAD data sets are completed in 163 seconds: less than 3 minutes, as compared with the 6,290 seconds (1 hour and 45 minutes) required to analyze the four data sets sequentially. Thus, the data reduction can be performed quickly with a varied set of parameters.

As pointed out earlier, the limiting factor for the efficiency is the potentially large overlap between different subscans. As a rule of thumb, we recommend the use of eight processors for each scan collected during the experiment to achieve a high rate of efficiency.

Memory Utilization The maximum memory used at any one time in a single integration process is defined by:

$$\begin{array}{rcll} M_{total} & = & 1 * \text{diffraction image} & (1 * 18Mbyte) \\ & + & 4 * \text{spatial distortion files} & (4 * 2.4Mbyte) \\ & + & 1 * \text{nonunformity image} & (1 * 18Mbyte) \\ & + & 1 * \text{dark image} & (1 * 18Mbyte) \\ & + & 1 * M_{as} & \\ & = & M_{as} + 63.6Mbyte & \end{array}$$

M_{as} specifies the memory used to maintain the active shoeboxes. Let r be the number of reflections in the list of reflections, $s_x(r)$ be the predicted shoeboxsize in x direction, $s_y(r)$ the predicted shoeboxsize in y direction, $\rho_w(r)$ the predicted spot width in degrees of the shoebox, and I_w the image width in degrees. Here *pad* is a factor used for increasing the analyzed area in order not to overlook information. Then, M_{as} is defined as

$$M_{as} = \sum_r s_x(r) * s_y(r) * (2pad + \rho_s(r)/I_w) \tag{1}$$

The integration algorithm spends the majority of its time managing shoeboxes rather than performing floating point or integer calculations. For the KSI data set this number is smaller than 2×18 Mbytes.

5.4. Scaling and Merging

We recognized early the need to increase the performance of the integration algorithm. In addition, we found that the original sequential scaling program ran extremely slow for low-symmetry crystals, specifically the triclinic cytochrome c (see Table 5). Thus, the authors of the sequential program replaced the original scale-merge by a program called *REQAB*, which performs significantly faster [6]. For the cytochrome c data set, 10 cycles of the original sale-merge algorithm resulting in over 7000 seconds are required to properly scale the data and to generate output summaries. When scaling with REQAB, 1 cycle of the original scale-merge algorithm is needed for the generation of the output data in addition to performing REQAB. Thus, a total of 398 seconds to scale and summarize the cytochrome c data is needed. This is a 17-fold acceleration compared with the original scale-merge program.

6. Summary and Conclusions

This paper analyzed the requirements for improving the computational facilities at a unique research facility, the APS. Two very important requirements are the ease of use and the quasi-real-time-processing of the data to increase usability and throughput.

To fulfill these requirements, we developed a unifying framework that enables data acquisition, data reduction, data analysis, structure solving, and data visualization. We utilized a real metacomputing environment by using the GUSTO testbed, which is part of the Globus project. Moreover, we pioneered network parameters and infrastructure enabling fast access to the GUSTO testbed from the APS that is reused by other projects.

We successfully ported the sequential integration program to a parallel environment. We integrated our changes to the code into the production code, distributed by the original author of the program. We also initiated improvements in the performance of the production code.

We have demonstrated that the parallel version of the data reduction code renders results identical to those of the serial version, but in a fraction of the time, depending on the number of processors applied to the task. We have solved a crystal structure using this program. Thus, we have completed the first stages of validation. As with all crystallographic software, more thorough validation is accomplished through use with a wide variety of crystals by the user community. The parallel integration program is ready to proceed to this final stage of validation. Future work involves fine-tuning of the parallel algorithm.

Using the parallel integration program, users can achieve significant acceleration of processing speeds. With the trend in the detector marketplace toward faster readout speeds, the ability to analyze the data in parallel becomes increasingly advantageous. A planned upgrade to the APS 1 Detector readout electronics would decrease the readout speed by a factor of 4. In these cases, using the parallel integration program, data analysis would not become the overall "rate-limit" at a synchrotron beamline.

Using the parallel integration program, users can distribute the data reduction task over multiple processors within a symmetric multiprocessor system, heterogeneous "off-the-shelf" compute cluster, or over multiple nodes of a supercomputer which may be accessed through the Internet via a high-performance link. This has been demonstrated multiple times during the past year. In addition, we have shown a simple way to incorporate a structure-solving algorithm utilizing a small number of workstations. Currently, research is underway to include other state-of-the-art solving algorithms.

For experimentation at synchrotron sources like the APS at Argonne that have a high-performance link to supercomputing resources onsite and to the ESnet and vBNS networks and as more high-speed backbones and uplinks are installed, this approach to real-time crystallographic data analysis becomes an achievable reality. MOreover, collaboration tools supporting multiple output devices from CAVE to graphic workstations are already available today.

7. Code Availability

The program d*TREK is compatible with viturally all modern operating system platforms (AIX, IRIX, HP-UX, Linux, Solaris, and Windows NT) and can process data from most, detectors currently in service at synchrotron beamlines that perform macromolecular crystallography.

We have tested the d*TREK parallel dtintegrate implementation on AIX, IRIX, and Solaris using both GNU and platform-specific ANSI-C++. The parallel d*TREK code changes have been introduced into the distributed code and may be compiled and run by anyone licensed to use d*TREK.

The d*TREK suite and toolkit for analysis of single crystal experimentation with 2-D position-sensitive detectors, was written by Dr. James W. Pflugrath of Molecular Structure Corporation [10]. d*TREK can be used freely by users of APS and is available for a minimal handling fee to all researchers funded by DOE. Currently, d*TREK has been licensed to over 30 institutions in the U.S.A., Canada, Japan, the UK and France.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38. Globus research and development is supported by DARPA, DOE, and NSF.

We thank Dr. James Pflugrath, the developer of the sequential d*TREK program [8], provided valuable comments during the duration of the project. The APS 1 Detector was designed and built for the Structural Biology Center at ANL by the ANL-ECT Instrument Design Group, led by Istvan Naday. We thank Dr. Ruslan Sanishvili (ANL-CMB) for the cytochrome c data.

We are grateful to Joe Insley for the implementation of the molecule viewer for the ImmersaDesk and CAVE.

References

[1] Bragg, L. X-ray Crystallography. *Scientific American 219*, 1 (July 1968), 58–70.

[2] Cantor, C. R., and Schimmel, P. R. *Biophysical Chemistry, Part II: Techniques for the study of biological structure and function.* W. H. Freeman, San Francisco, 1980.

[3] Chang, C.-S., DeTitta, G., Miller, R., and Weeks, C. On the Application of Parallel Genetic Algorithms in X-Ray Crystallography. In *Proceedings of the Scalable High-Performance Computing Conference, May 23–25, 1994, Knoxville, Tennessee* (1994), IEEE Computer Society Press, pp. 796–802.

[4] Foster, I., and Kesselman, C. The Globus Project: A Status Report. In *Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop* (1998), pp. 4–18.

[5] Howells, M. R., Kirz, J., and Sayre, D. X-ray Microscopes. *Scientific American 264*, 2 (Feb. 1991), 88–94 (int. ed. 42–48).

[6] Jacobson, B. REQAB Scaling and Merging Algorithm. Ames Laboratory, 1998.

[7] Karle, J. Some Developments in Anomalous Dispersion for the Structural Investigation of Macromolecular Systems in Biology. *International Journal of Quantum Chemistry Quantum Biology Symposium 7* (1980), 357–367.

[8] Messerschmidt, A., and Pflugrath, J. W. Crystal Orientation and X-Ray Pattern Prediction Routines for Area-Detector Diffractometer Dystems in Macromolecular Crystallography. *Journal of Applied Crystalography 20* (1987), pp. 306–315.

[9] Pflugrath, J. W. Diffraction-Data Processing for Electronic Detectors: Theory and Practice. *Methods in Enzymology 276* (1997), pp. 286–306.

[10] Pflugrath, J. W. d*TREK. Molecular Structure Corporation, The Woodlands, TX, http://www.msc.com, 1998.

[11] Terwilliger, T. C. Multiwavelength Anomalous Diffraction Phasing of Macromolecular Structures: Analysis of MAD Data as Single Isomorphous Replacement with Anomalous Scattering Data using the MADMRG program. In *Methods in Enzymology*, J. (Ed.) C.W. Carter and R. Sweet, Eds., vol. 276, Part A. of *Macromolecular Crystallography.* San Diego, Academic Press., 1997, pp. pp. 530–537.

[12] von Laszewski, G., Su, M.-H., Insley, J. A., Foster, I., Bresnahan, J., Kesselman, C., Thiebaux, M., Rivers, M. L., Wang, S., Tieman, B., and McNulty, I. Real-Time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Sources. In *Ninth SIAM Conference on Parallel Processing for Scientific Computing* (March 1999).

[13] Westbrook, E., and Naday, I. Charge-Coupled Device-Based Area Detectors. *Methods in Enzymology 276* (1997), 244–268.

[14] Westbrook, M. L., Coleman, T. A., and Daly, R. T. SBC-CAT Control System Commissioning Report. Tech. Rep. SBC01-0308, Argonne National Laboratory, Feb. 1997.

[15] Westbrook, M. L., Coleman, T. A., Daly, R. T., and Pflugrath, J. W. Data Acquisition and Analysis at the Structural Biology Center. In *Proceedings IUCr Macromolecular Crystallography Computing School* (1996).

[16] Woolfson, M. M. *An introduction to X-ray crystallography*, 2nd ed. Cambridge University Press, 1997.

[17] APS1 Detector Specification, 3x3 CCD-Array Imaging X-RAY Detector. http://www.ee.anl.gov/engineer/aps1_s.htm.

[18] The Globus Project Web Page. http://www.globus.org/.

[19] Macromolecular Structure Determination at the Lawrence Livermore National Laboratory. http://www-structure.llnl.gov/Xray/101index.html.

[20] The X-ray Grand Chalange Project Web Page. http://www.mcs.anl.gov/xray.

Gregor von Laszewski is a research scientist at Argonne National Laboratory and a fellow of the Computation Institute at the University of Illinois at Chicago. His research interests include the utilization of commodity technologies in distributed and parallel computing and the development of novel uses of computing in scientific applications. He received a PhD in computer science from Syracuse University. Contact him at gregor@mcs.anl.gov.

Mary Westbrook received a B.S. in both physics and computer science, from Lewis University. She has been at Argonne National Laboratory for over 20 years, and has designed and implemented controls systems and data acquisition systems for numerous scientific facilities. Together with Edwin Westbrook, she optimizes the performance of protein crystal diffraction facilities at synchrotron sources. Contact her at westbroo@mcs.anl.gov.

Craig Barnes is a PhD candidate in Computer Science at the University of Illinois at Chicago. His research interests include virtual reality, intelligent agents, and 3D interfaces. He received his BS from the University of Massachusetts and his MS from University of Illinois at Chicago He is a member of ACM SigGraph and SigCHI.

Ian Foster is a senior scientist and associate director in the Mathematics and Computer Science Division at Argonne National Laboratory and an associate professor of computer science at the University of Chicago. His research interests include high-performance computing, distributed computing, and computational science. He received a PhD in computer science from Imperial College, London. He is a member of the ACM and the American Association for Advancement of Science. Contact him at foster@mcs.anl.gov.

Edwin Westbrook received an A.B. in Physics at the University of California at Berkeley, an M.S. in Applied Physics at Stanford, and a Ph.D. in Biophysics and an M.D. in medicine, both from the University of Chicago. He has been at Argonne National Laboratory for 17 years. He designs and builds position-sensitive X-ray detectors and X-ray beamlines at synchrotron sources, that are used for protein crystallography. He is a practicing protein crystallographer, currently studying the structural molecular biology of proteins that carry and transmit information within bacteria, allowing them to respond to varying chemical stimuli in their environment.Contact him at westbrook@anl.gov.

Table 3
Performance of the parallel integration program on an SP2 and an Origin2000 aplied to the 720 18Mbyte cytochrome c images

N_p	RS/6000 $t_{integrate}$ in s	RS/6000 Speedup	Origin2000 $t_{integrate}$ in s	Origin2000 Speedup
1	3860	1.0	2160	1.0
2	2000	1.9	1130	1.9
4	1100	3.5	607	3.6
8	614	6.3	339	6.4
16	368	10.5	208	10.4
20	315	12.0	-	-

Table 4
Performance of the parallel integration program on an SP2 and an Origin2000 applied to the 4 KSI all datasets with 4×450 images, each 4.5Mbyte large.

N_p	RS/6000 $t_{integrate}$ in s	RS/6000 Speedup	Origin2000 $t_{integrate}$ in s	Origin2000 Speedup
1	6290	1	3378	1
2	3030	2.1	1761	1.92
4	1520	4.2	1040	3.2
8	917	6.9	575	5.9
16	501	12.6	312	10.83
20	388	16.2	-	-
32	273	23.0	-	-
48	230	27.4	-	-
64	162	38.8	-	-

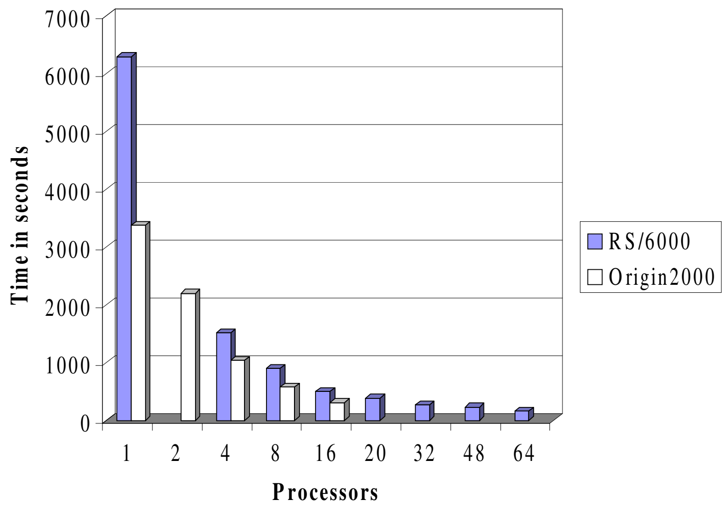

Figure 11. *Runtime performance of the parallel integration program on an SP2 and an Origin2000 applied to the KSI all dataset with 4×450 images.*

Table 5
Merging and scaling performance on an Origin2000.

Dataset	S_I in MByte	N_I	N_{rl}	t_{rmerge} in s	10 cycles t_{smerge} in s	t_{reqab} in s	1 cycle t_{smerge} in s	t_integrate in s
Lysosyme	18	384	9	6.9	428.7	55.0	7.1	-
KSI-low	4.5	450	9	25.1	1055.5	167.2	35.2	-
KSI-infl	4.5	450	9	26.8	1070.7	157.9	38.4	-
KSI-peak	4.5	450	9	26.6	1123.2	129.7	37.9	-
KSI-high	4.5	450	9	25.0	1041.2	131.3	101.7	1040
Cytochrome c	18	720	10	62.8	7010.9	325.2	72.4	2160

Note: S_I=Imag Size, N_I=Number of images, N_{rl}=Number of reflection list, t_{rmerge}=time to conduct the reflection merge, t_{smerge}=time to conduct the scalemerge, t_{reqab}=time for reqab scaling and merging.

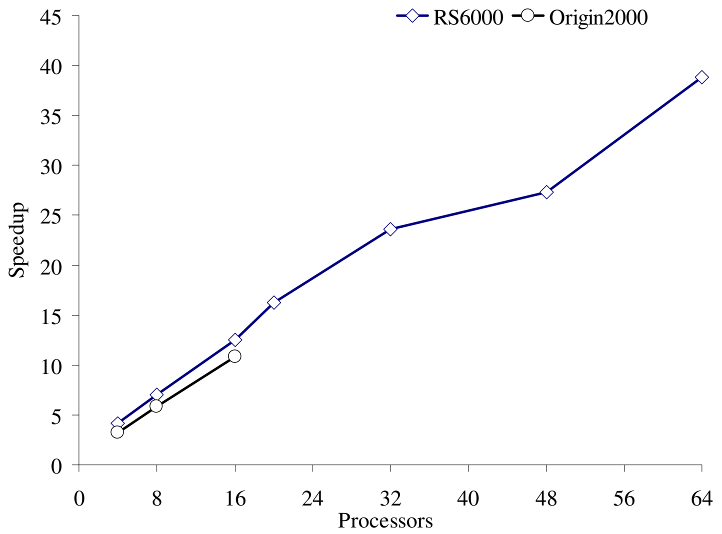

Figure 12. *Speedup performance of the parallel integration program on an RS/6000 and an Origin2000 applied to the KSI all dataset with 4×450 images.*

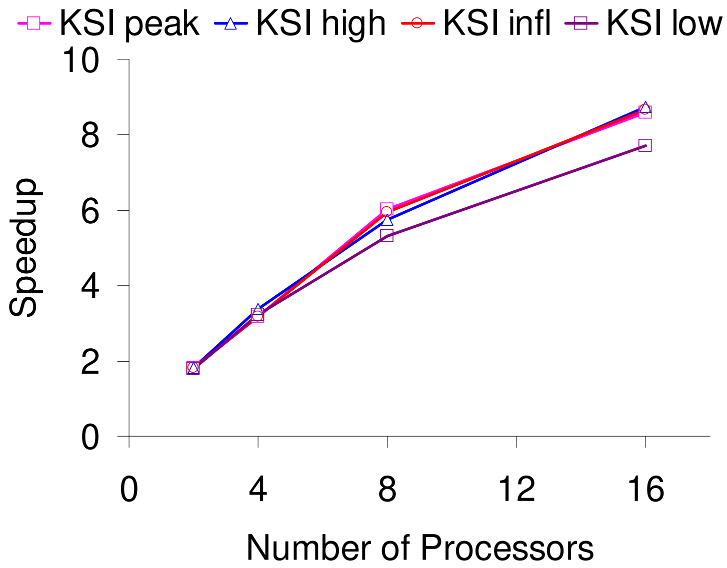

Figure 13. *Speedup of the parallel integration program on the Origin2000 applied to the KSI all dataset.*

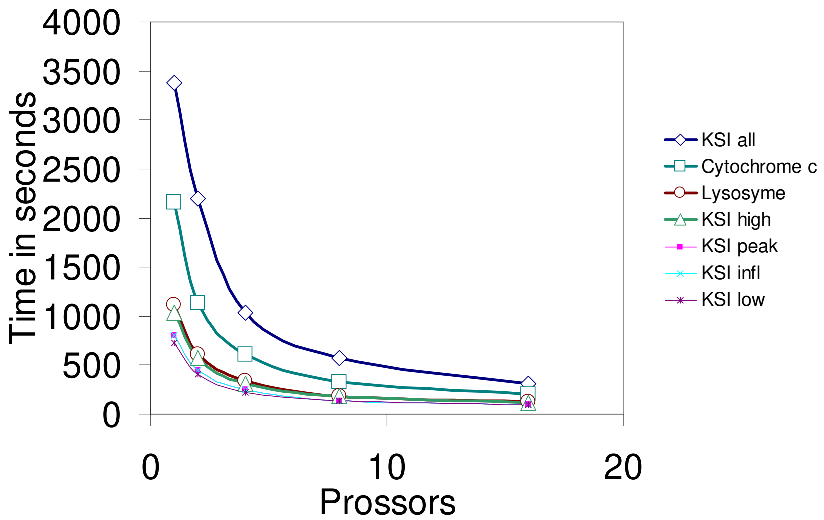

Figure 14. *Execution time of the parallel integration program on the Origin2000 applied to the KSI all dataset.*

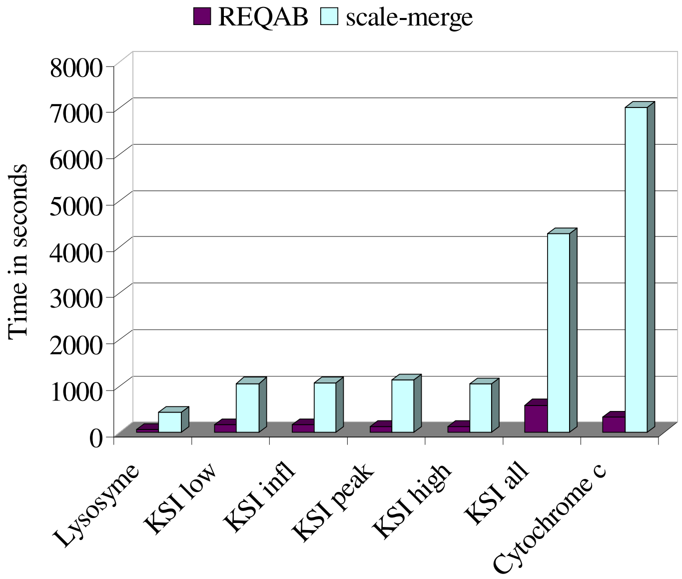

Figure 15. *Merging and scaling performance on an Origin2000.*