HPC and Grid Applications in High Throughput Protein Crystallography.

R. Keegan, D. Meredith, G. Winter and M. Winn,

Daresbury Laboratory, Warrington, Cheshire, WA4 4AD, U.K.

Abstract

We present details of work being carried out to aid the development of high throughput protein crystallography through the use of high performance computing and Grid technologies. We focus upon the speedup of data collection processes and structure solution, and upon the remote access to high-performance compute resources via the Grid. Rapid feedback regarding the quality of data collected on a synchrotron beam-line is essential for the efficient operation of high-throughput systems. To achieve this goal, we have developed parallel versions of data processing codes for execution on cheap “Beowulf” type clusters. Access to these high-performance resources has been achieved through the use of web-services and Grid-enabled web-portal technologies. This facilitates the remote submission and monitoring of computationally intensive jobs in a secure, platform independent environment.

1. Introduction

Protein Crystallography is one of the most widely used methods for determining the molecular structure of proteins. It can be carried out using a small, lab-based, apparatus but for best results X-rays generated from a synchrotron source need to be used. Facilities for generating this kind of radiation are currently only provided by large-scale developments such as the SRS at Daresbury Laboratory.

There are several steps involved in determining a protein structure. Figure (2) gives an illustrative view of the process. The first step is to determine the target protein, the protein is then produced and purified before it is crystallised. The crystals are then shipped to a synchrotron beam-line and exposed to X-rays. A set of images containing diffraction patterns created by the scattering of the X-rays by the atoms in the crystallised protein is generated. These images are then processed using a system of computational tools to derive the 3-dimensional structure. Once the structure has been determined it can then be deposited in the Protein Data Bank (PDB).

The e-HTPX project is involved in the development of a Grid-based e–science environment to allow structural biologists remote, integrated access to web and grid technologies associated with protein crystallography. The project encompasses all stages of the protein structure determination process, from protein production through to deposition of the final refined model. The emphasis is on the unification of all the procedures involved in protein structure determination. This will be achieved by the development of an easy-to-use, all-in-one interface from which users can initiate, plan, direct and document their experiment either locally or remotely from a desktop computer. The project is spread across several sites, including Daresbury Laboratory, York University, the OPPF in Oxford, the EBI in Cambridge and BM14 at the ESRF.

Here, we will focus on the use of high performance computing (HPC) and Grid technologies within the e-HTPX project.

2. HPC and Grid technologies in e-HTPX

The data collection and structure determination stages of the protein crystallography pipeline involve both heavy use of computational resources and the generation of large quantities of data. With the development of high-throughput methods and the every growing complexity of the structures being solved, use of HPC and Grid technologies is a necessity.

High-speed CCD imaging cameras are used on the beam-lines to collect the diffraction images. In a high-throughput automated system these cameras can produce up to 250 Megabytes of data per minute. The development of sophisticated tools using the latest software and hardware developments are required for the processing, communication and storing of this data.

Taking a set of images containing a large set of diffraction spots, and interpreting this data so as to produce a 3-dimensional structure for the protein, involves a complex system of computational processing with a lot of user input. A popular suite of codes designed to do this is the CCP4 [1] distribution. To aid the development of a high-throughput, automated, system we have developed parallelised versions of some of the key codes and utilised job-farming tools deployed on dedicated compute clusters. These compute clusters will be Grid accessible to allow users to submit and monitor jobs via the e-HTPX web/grid portal.

3. Software Development

Figure (2) illustrates one possible “roadmap” for the steps involved in the data processing stage of resolving a protein structure.

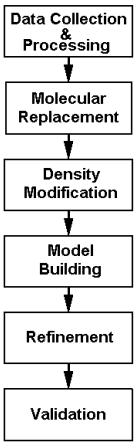

Figure (1) – Data processing roadmap

The procedure may be divided into two separate sections. Firstly, the initial data collection at the beam-line, where the images are collected, stored and processed to reduce the diffraction spot data into a manageable form. The second stage is the processing of the reduced data set to resolve the protein structure. For each of these steps we have developed methods to improve the speed at which they can be carried out.

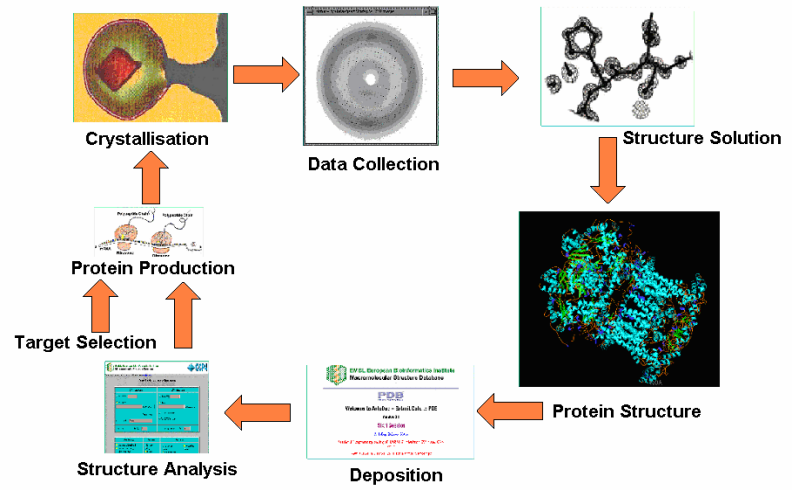

Figure (2) – Protein Crystallography procedure for solving a protein structure.

Initial data Collection:

For the data collection on the beam line we have investigated the use of small, high specification cluster machines in order to speed up the analysis of the diffraction images. Crystallising a protein is a hit-and-miss procedure and quite often it is necessary to produce several crystals in order to find a good one. The quality of a crystal only becomes apparent at the data collection stage. As a result it is important to have rapid feedback on the crystal quality from the data collection programs to a user or an automated system. We have produced parallel versions of two data collection programs, Mosflm [2], designed to integrate the diffraction spots on all of the images, and Scala [3], a code for scaling and merging the spot data to account for changes that may occur in the experimental conditions through the course of the data collection procedure.

Mosflm

Mosflm takes the diffraction images as its input. Each image can be up to 20 Megabytes in size and with the images being produced at a rate of 12-15 per minute. As a result Mosflm is I/O bound and requires a lot of memory bandwidth. We have used a dedicated high performance machine directly on the beam-line to speedup it's processing. Since the images being collected can be treated independently, the parallelisation can be performed at a higher level – running multiple instances of the program on different data. Using this mechanism the throughput of the data processing can be substantially increased. Figure (3) shows how each processor will take a batch of images, as they are produced by the collection procedure, and process them to give the integrated results.

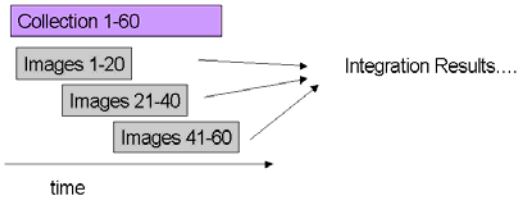

Figure (3) – Image processing using parallel Mosflm.60 images are collected and processed in batches as they are produced.

Scala

Scala takes the output file from Mosflm, containing the integrated spot intensities, and scales and merges the data. The most time-consuming part of the code is the scaling and is the most suitable for parallelisation. The input file can be split among a few processors and each one scales its subset of the data. The scaling involves the global summing of a matrix of parameters so communication between the processors is necessary.

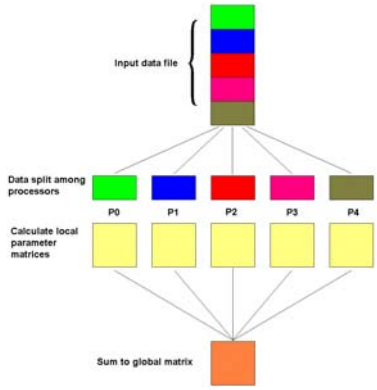

Figure (4) – Parallelisation strategy for Scala.

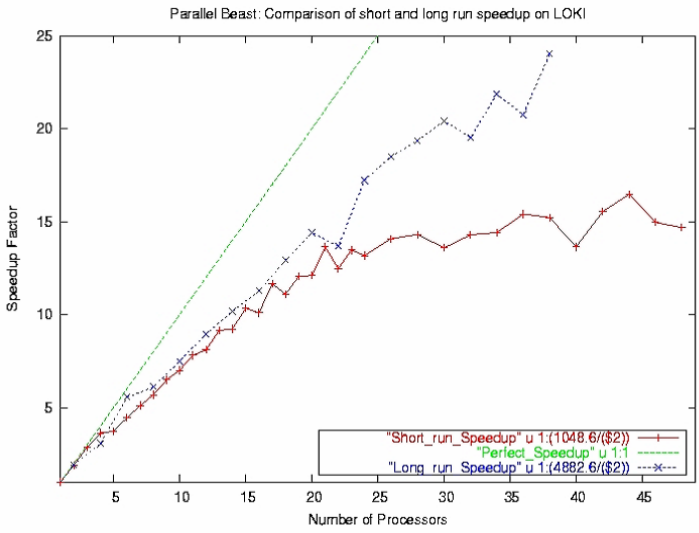

Figure (5) – Speedup of parallel MR code. The red line shows the results for a search over a small set of rotations and translations. When the search range was broadened (blue line) the performance scaled a lot better as the communication overhead was less significant.

We have used the MPI message passing libraries to achieve this. The parallelisation strategy for Scala is outlined in figure (4).

Structure solution:

Once the data from the diffraction images has been initially processed and accepted as being of a sufficient quality, the processing moves on to the structure solving stage. One of the most time consuming stages of the structure solving is molecular replacement (MR). Here, the unidentified structure data is compared with known related structures. Depending on the quality of the initial data and the level of homology of any related structures, this process can involve a lot of trial and error.

A popular code for doing MR is Beast [4]. Beast involves a 6-dimensional search (3 rotation directions and 3 translation directions). We have developed a parallel version of this code for running on a "Beowulf" type cluster using MPI to handle inter-processor communications. Figure (5) shows a plot of the speedup gained when we ran this code on a 48-processor cluster.

4. Computational resources

A 9 node dual Xeon cluster has been procured for the purpose of the HPC aspects of the project. In addition a single, high specification, dual Xeon processor machine, including high speed SCSI hard disks has been assigned to running the parallel Mosflm code on the synchrotron beam-line along with a 4-processor Xeon mini-cluster for running the parallel Scala code.

The main cluster is assigned to the running of the structure solution codes. Apart from the parallel MR code, Beast, this machine is configured as a task-farming machine using OpenMosix and the Condor batch queuing system to schedule and manage the submitted jobs.

The e-HTPX web/grid portal allows for remote access to use the cluster facilities based at Daresbury via the Grid using the Globus GRAM job submission manager. Remote users at university or company laboratories, along with users at the UK beam-line at the ESRF in Grenoble, can transfer data, submit remote processing jobs and monitor their experiments via this hub. Figure (6) shows the infrastructure set-up for the project.

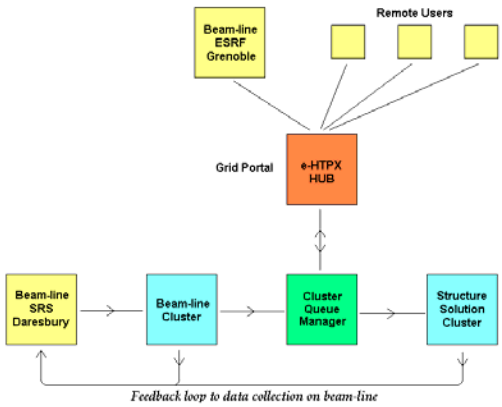

Figure (6) HPC and Grid applications in e-HTPX

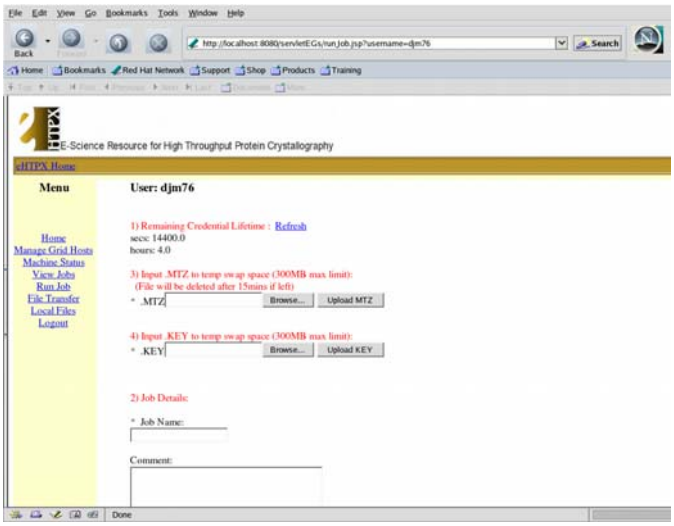

Figure (7) - Remote job submission from the e-HTPX portal.

5. Grid

A grid-enabled web-portal interface has been designed from which the user can remotely plan, submit and monitor the high-performance compute jobs described above, and to securely transfer necessary data. The ability to continuously monitor job-status and obtain feedback without significant delay aids rapid decision-making required for high-throughput techniques.

The key technologies applied in the project include the Globus toolkit 2.4 and Java Cog-Kit API, which have been used to access and deploy the grid-services. Security is provided by the GSI (Grid Security Infrastructure) certificate model, which facilitates single sign-on and access to multiple-resources accessed by the portal. Figure (7) shows a screenshot of the job submission page of the e-HTPX portal.

6. Conclusions

We have developed a set of HPC and Grid tools to aid work being done to create a system for high-throughput protein crystallography. The parallelisation of key codes has allowed for a user or an automated system to receive real-time feedback on the quality of their data during the data collection procedure on a synchrotron beam-line. Developments have also been made in HPC and Grid tools to improve the speed of post data collection structure solving. Future work will target the development of HPC methods for automated molecular replacement. At present the PDB contains more than 21,000 solved structures. For many unsolved structures, it is desirable to do MR against as many of the known proteins, with similar structures, as possible. This can only be done with a computational resource such as that provided by the Grid and the utilization of clustering tools such as Condor and OpenMosix.

References:

[1] CCP4 collaborative computational project http://www.ccp4.ac.uk
[2] Mosflm http://www.mrc-lmb.cam.ac.uk/harry/mosflm/
[3] Scala http://www.ccp4.ac.uk/dist/html/scala.html
[4] Beast http://www.ccp4.ac.uk/dist/html/beast.html