Journal of
**Applied
Crystallography**

ISSN 0021-8898

Editor: **Gernot Kostorz**

Qs v.1.3: a parallel version of *Queen of Spades*

Nicholas M. Glykos

Journal of
Applied
Crystallography

ISSN 0021-8898

computer program abstracts

This category provides a rapid means of communicating up-to-date information concerning both new programs or systems and significant updates to existing ones. Submissions should follow the standard format given in *J. Appl. Cryst.* (1985). **18**, 189–190, also available from **Crystallography Journals Online** at http://journals.iucr.org/j/services/authorservices.html.

Qs v.1.3: a parallel version of *Queen of Spades*

Nicholas M. Glykos

Department of Molecular Biology and Genetics, Democritus University of Thrace, Dimitras 19, 68100 Alexandroupolis, Greece. Correspondence e-mail: glykos@mbg.duth.gr

Received 27 February 2005
Accepted 8 April 2005

Keywords: molecular replacement; stochastic molecular replacement; message passing interface; computer program parallelization; computer clusters; Beowulf

1. The crystallographic problem

The program *Queen of Spades* (Glykos & Kokkinidis, 2000; Glykos & Kokkinidis, 2001) encodes an algorithm for a stochastic multidimensional approach to molecular replacement. The program has been shown to be capable of successfully locating solutions even in cases as complex as a 23-dimensional 4-body problem (Glykos & Kokkinidis, 2003). Recently, we extended our approach to tackle the full molecular-replacement problem by allowing the possibility of using many different search models simultaneously, and showed that we could successfully locate solutions in the case of a 17-dimensional problem involving one DNA and two (different) protein search models (Glykos & Kokkinidis, 2004). This multimodel multidimensional approach does have its cost: with a few thousand unique reflections in a high-symmetry space group and with more than two search models, a typical *Qs* run would take well over two to three weeks of CPU time on the fastest personal workstations. The way forward for such computationally intensive calculations is of course parallelization. Here, I report the availability of a parallel version of *Qs* which is based on the Message Passing Interface (MPI) paradigm.

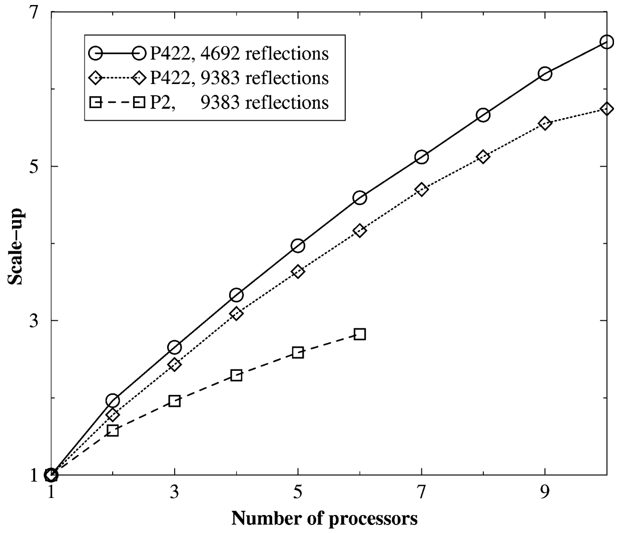

Figure 1
Parallel scale-up of *Qs* for three different problems.

2. Method of solution

Profiling the *Qs* executable on a stand-alone GNU/Linux machine clearly showed what was already anticipated: well over 90% of the CPU time is spent in the structure-factor calculation routine (in reality, it should be referred to as a ‘structure-factor interpolation routine’, since the program is based on the presence of pre-calculated molecular transforms to avoid continuous FFTs). Clearly, an implementation that would successfully parallelize this function would suffice. The current version of the *Qs* source code runs to approximately half a megabyte, covering over 8000 lines of C code. It came as a surprise that a fine-grained parallelization of the program could be implemented by adding less than 10 lines of code (containing a handful of calls to the MPI functions). The principal idea is trivial: the master node broadcasts [function MPI_Bcast()] the position and orientation of the molecule for which structure factors are being calculated and then each and every node (including the master node) calculates structure factors for only ($1/n$) of the total number of reflections, where n is the total number of nodes. In the final step, the master node collects the results [function MPI_Gather()] and proceeds with the Metropolis algorithm (Metropolis *et al.*, 1953). Because for each unique reflection the program has to interpolate values of molecular transform corresponding to each and every of the crystallographically related molecules, the total CPU time spent on each node is proportional to the product (number of reflections times the number of crystallographic symmetry operators). The important point is that the lengths of the messages sent to the master node depend only on the number of unique reflections, not the number of crystallographic symmetry operators. The implication is that for high-symmetry space groups, the CPU requirements per node increase, but the network load (required communication bandwidth) decreases. The expectation that the program will scale better as the number of unique reflections decreases and the symmetry increases was confirmed *via* a series of tests performed on two Beowulf clusters. Representative results from these tests are shown in Fig. 1.

3. Software environment

Programming language and operating systems. The program is written in the C programming language and has been shown to compile on a wide variety of operating systems, including but not limited to various flavours of Unix (Linux, Irix, SunOS, OSF, HPUX), VMS and Windows. The parallel version should compile on any system supporting MPI program execution.

Overlay structure. None.

Subroutine libraries accessed. The minimum requirement for successful compilation and linking of *Qs* is the availability of the free open-source FFTW library (obtainable from http://www.fftw.org/), plus the MPI-specific libraries (which are system and vendor dependent).

doi:10.1107/S0021889805011052

4. Hardware environment

Computers and installation. The parallel version of *Qs* has been extensively tested on two GNU/Linux-based Beowulf clusters, but it should compile on any system supporting an MPI environment. The stand-alone executable can be located in any suitable directory.

Number of bits per byte. 8.

Minimum size of physical memory required. The program requires physical memory sufficient to store the molecular transforms of the search models plus a fixed-size array containing partial structure factors for each reflection. The memory requirements do not scale down in the program's parallel version (each and every node keeps a complete copy of the molecular transforms and reflections).

5. Program specification

Restrictions on the complexity of the calculation. The calculation is always performed in space group *P*1, and consequently there are no space-group-specific restrictions. The program supports multi-dimensional molecular-replacement searches for up to (and including) 36-dimensional problems involving up to six different search models.

Typical run times. For the parallel version of *Qs*, the scale-up (which measures how many times faster the program executes when parallelized compared with the uniprocessor version of the program) depends on (i) the number of processors, (ii) the number of unique reflections, and (iii) the number of crystallographic symmetry operators of the target structure. Fig. 1 shows a graph of observed scale-up *versus* number of processors for three different problems.

Number of lines. 8291 for the C source code, 3134 for the raw L^AT_EX documentation.

Test status. The parallel version of *Qs* has been tested on two Beowulf clusters, one based on a Gigabit ethernet interconnect, the other on a 10/100 Mbps switched ethernet. The tests included usage of two different MPI implementations (MPICH and LAM-MPI) with different underlying compilers and processor architectures.

6. Documentation and availability

Extensive documentation is available with the distribution in the form of a PDF file, html pages, and a Unix manual page.

Qs is free open-source software. The program's distribution includes source code, documentation, example scripts, and stand-alone executables (for the uniprocessor version) and is immediately available for download from http://www.mbg.duth.gr/~glykos/ or from the various mirrors generously provided by the Collaborative Computational Project, Number 14.

References

Glykos, N. M. & Kokkinidis, M. (2000). *Acta Cryst.* D**56**, 169–174.

Glykos, N. M. & Kokkinidis, M. (2001). *Acta Cryst.* D**57**, 1462–1473.

Glykos, N. M. & Kokkinidis, M. (2003). *Acta Cryst.* D**59**, 709–718.

Glykos, N. M. & Kokkinidis, M. (2004). *J. Appl. Cryst.* **37**, 159–161.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). *J. Chem. Phys.* **21**, 1087–1092.