
COMPUTING WITH PARALLEL VIRTUAL MACHINES

C.E.Dean^{\$}, R.C.Denny^{\$#}, H. F. van Garderen^{*} and E.Pantos^{\$}

^{\$} SERC, Daresbury Laboratory, Warrington WA4 A4D, Great Britain.

[#]The Blackett Laboratory, Imperial College, London SW7 2BZ.

^{*} Schuit Institute of Catalysis, Department of Chemistry, University of Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

A large number of scientific applications have now been implemented on parallel supercomputers of the multiple processor, distributed memory, shared bus kind. Access to such resources is limited to relatively a few sites. For many research groups the principal computing resource is a small number of workstations accessed by a small user population. Modern workstations can be quite powerful machines in their own right. Highly suitable as they are in providing high peak performance and graphics visualisation capability they are not always fully utilised. Consequently, "spare" capacity goes begging during idle periods.

Recent developments in harnessing the power of independent computers have resulted in the emergence of software tools which enable the parallel utilisation of workstation clusters of different configuration, type, or even manufacture. Such a software harness is the PVM package [1] which utilises a protocol for data and message passing between the workstations over the network. The individual workstations constitute the elements of the *parallel virtual machine*.

We have employed the PVM software to implement four programs of relevance to SAXS and fiber diffraction work. The first uses the Debye formula to compute the structure factor of a structure modelled by a number of spheres [2]. The CPU intensive part of the code computes the histogram of pair distances from which the structure factor as well as the pair correlation function, CF, and the radial distribution function, RDF, can be calculated as defined in [3]. Fig. 1 shows the computed SAXS, CF and RDF for a model of a gel network produced by program GRASP [4].

The second program calculates the mass projection of a model structure for a series of projection angles. It has been used in the past to produce mass projections of muscle fibre models [5]. It is now being modified to read coordinates of any structure produced by any model building program. The third computes the Fourier Transform of the mass projection series to produce the simulated X-ray diffraction pattern [5]. The fourth program is a new application under development for background subtracting experimental X-ray diffraction patterns and computing the area under diffraction peaks.

The first two programs have already been implemented on the Daresbury 64-node Intel iPSC/860 hypercube [5]. The principal restriction arose out of the limited memory on each processor (16MB), a significant handicap for processing large 2D images or for large sphere models. Workstations, on the other hand, are usually equipped with larger memory. This makes implementation significantly easier as there is no need to utilise temporary disk storage.

In all four cases we take the coarse-grain SIMD approach where identical copies of the compute-intensive code are executed on each workstation but operate on different data. In the first two cases the same dataset is read in by all the processing nodes but each processor operates on different subsets of the coordinates. In the last two cases each workstation processes different datasets. Communication overheads are small compared to the total processing time. Processing of the different chunks of data is independent of each other and no synchronisation is necessary at intermediate stages of the calculation.

The small communication overheads result in almost ideal performance, i.e., processing time is inversely proportional to the number of processors used. The slower processor determines the overall rate. We can run the programs in a mixed environment of SUN, SG and HP workstations. Even just a few nearly equally powerful machines can make a significant impact on turnaround time. PVM also allows execution of the same code on multiprocessor, shared-memory machines where parallelisation can be achieved by explicit compiler directives and, in cases where the problem is sufficiently small, even on a single workstation.

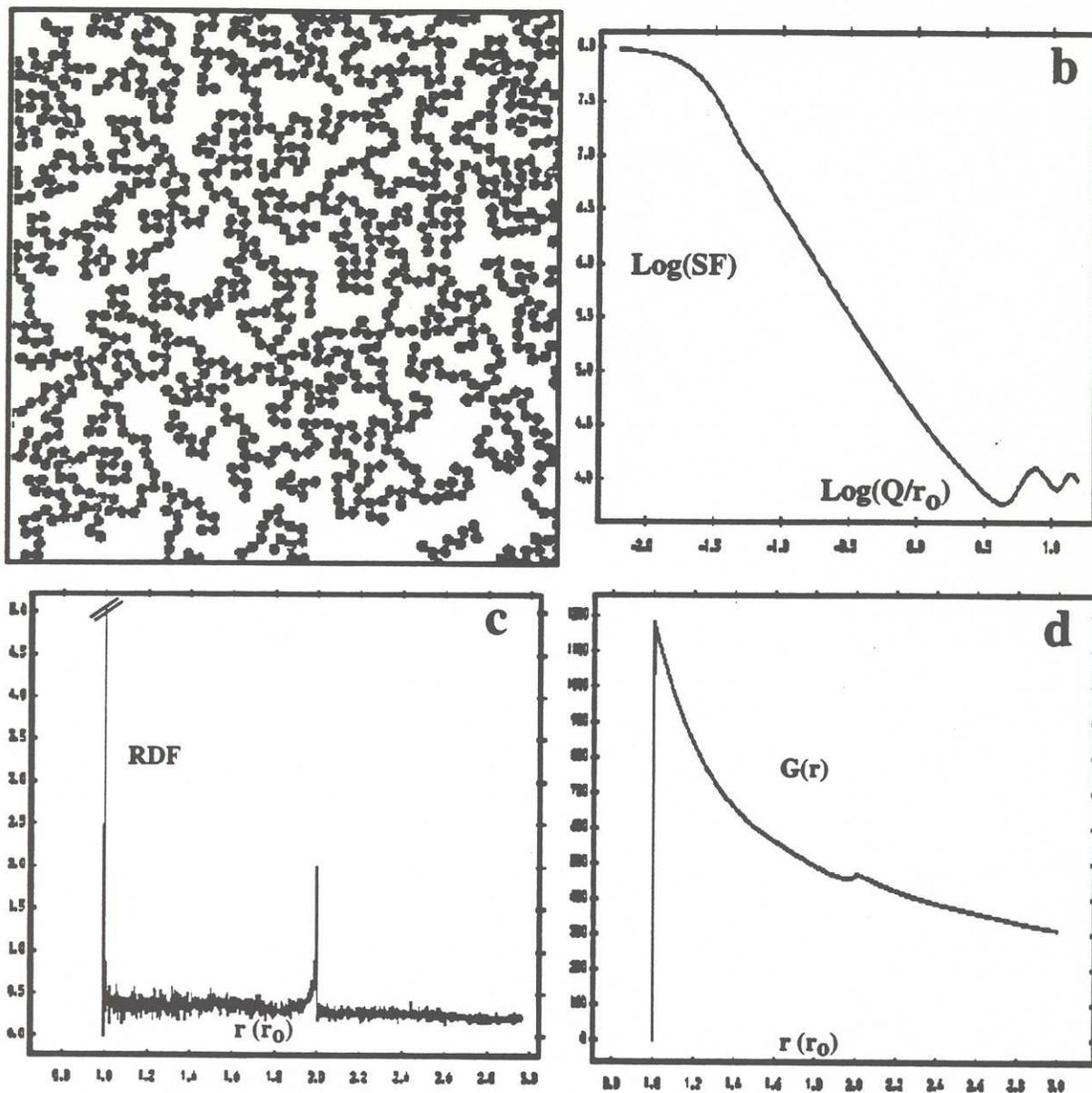


Figure 1. a) A section of a 2D model of a percolating gel network. The simulation conditions promote the growth of linear polymers with occasional branching. b) The SAXS structure factor in log-log plot for a system of 10000 particles in 2D at 40% vol/vol concentration. The slope of the linear section measures the fractal dimension of the system. c) The radial distribution function (peak at $r=1$ truncated) and d) the correlation function up to 3 particle diameters.

REFERENCES

1. J.Dongarra, G.A.Geist, R.Manckek and V.S.Sunderam, *Comp. in Phys.*, 7,166-174, 1993.
2. E.Pantos, C.E.Dean, P.C.Stephenson, G.J.Milne and H. F. van Garderen, Daresbury Preprint DL/CSE/P27E November 1993.
3. M.P.Allen and D.J.Tildesley, *Computer Simulation of Liquids*, McGraw-Hill, London, 1981.
4. H.F van Garderen, E.Pantos, W.H.Dokter, T.P.M.Beelen and R.A. van Santen, Daresbury Preprint DL/CSE/P22, May 1993, and *Model.Simul.Mater.Sci.Eng.*, in press.
5. F.G.Diaz, H.K.F.Yeung, E.Pantos and J.Bordas, in *Parallel Computing and Transputer Applications*, M.Valero, E.Onate, M.Jane and B.Suarez (Eds), pp. 1070-1079, IOS Press/CIMNE, Barcelona 1992.