Many SVU users may not be aware that they can now run parallel C or FORTRAN programs on our latest Linux cluster. Our Linux cluster with 16 compute nodes, each with two 3.06GHz Intel Xeon processors with Hyper Threading is built for such programs. In addition, high-speed Myrinet interconnects increases the speed of programs up to 10 times that of ordinary Gigabit Ethernet connections. Highly efficient compilers from Intel and Portland Group complete the setup to enable highly efficient parallel programs to run in this cluster. Most users will use MPI (message passing interface) protocol to program their parallel jobs, a popular choice among programmers. MPI libraries are available for both C and FORTRAN programs.

Since the release of the cluster, SVU has helped a few users to parallelise and run their programs using the above tools and libraries. We present two case studies below.

The first user was running his MIMD program which had the following characteristics: integer and double arrays of about 1200 elements each, and parallelised to run on up to 20 CPUs. Figure A, provided by the user, the relationship between the speed of the program with the number of processors and the problem size. Figure B shows that the parallel efficiency for a particular job decreases if the number of processors used is increased beyond the optimum number. Parallel programmers should take note of this fact and try to find their optimum number of processors for their own jobs.

A second user ran his ab-initio quantum-mechanical molecular dynamics program on both the linux (serial run) and linux_4p (4-CPU parallel run) queues. As expected, the parallel run was faster, in fact, 3.7 times the speed of the serial run. This is very close to the theoretical maximum of 4 times speedup.

The advent of the Linux cluster and its parallel tools has made parallel programming for easily achievable by almost everyone. And since the underlying Myrinet communications are built into the MPI libraries, programmers need only knowledge of MPI to make use of this highly efficient parallel Linux cluster. To access the cluster, telnet or ssh to atlas0.nus.edu.sg with your SVU account, or click on the atlas0 link on our e-SVU portal (http://e-svu.nus.edu.sg/e-svu). For more information, please contact me at enghee@nus.edu.sg.
Figure A: Speed-up vs. No. of problem size for different combination of processes

Figure B: Parallel efficiency vs. No. of different combination of processes