Parallel GAP

Gene Cooperman
gene@ccs.neu.edu

GAP/MPI is soon to be a share package for GAP. It is currently available at *ftp://ftp.ccs.neu.edu/pub/people/gene/gapmpi/*. It provides a simple, yet powerful programmer's model of parallelism, which will be shown in various examples as part of the demonstration. For example, the GAP/MPI function ParList was written in 20 lines of code, and is called as follow:

```
ParList( [1,2,3,4], x->x^2 ) => [1,4,9,16]
```

By design, GAP/MPI runs well on a cheap network of UNIX/Linux workstations or PC's. Parallelizing an interpreted language such as GAP has some unique issues for any production-quality implementation. These include keyboard-generated interrupts, handling of external signals, exceptions and "break" loops on a slave process, escaping from infinite loops in a slave process, flushing pending messages, and resetting to a known state after an error.

The same programmer's model is also available in C as TOP-C (Task-Oriented Parallel C) (*ftp://ftp.ccs.neu.edu/pub/people/gene/top-c/*), thus allowing the applications programmer to use GAP's foreign function interface and a uniform model of parallelism to parallelize packages both in GAP's own language and in C, while linking to GAP. Both GAP/MPI and TOP-C includes an MPI subset implementation based on top of Berkeley sockets, thus providing a self-contained package for most UNIX installations.

GAP/MPI provides the ability to program GAP in parallel at three distinct levels:

(1) using a low-level GAP binding to the MPI (Message Passing Interface) standard;

(2) using slave GAP processes on remote machines as GAP "servers" or GAP "listeners", on which to spawn "what-if" computations in background, without slowing down the main GAP session; and

(3) treating all attached processors as a single parallel machine to develop and run large computations in background.

Some of the natural features of the last and highest level are latency tolerance, natural load balancing, robustness in the presence of processors crashing and checkpointing. An experimental version for SMP (shared memory hardware) is also available upon request to the author.