Mixing MPI and OpenMP

Please make sure you read the SLAC specific instructions about MPI and OpenMP before you continue here.

Mixed "Hello World"

Name this little "Hello World" program `hello.c`:

```
#include <stdio.h>
#include "mpi.h"
#include <omp.h>

int main(int argc, char *argv[]) {
  int numprocs, rank, namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int iam = 0, np = 1;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(processor_name, &namelen);

  #pragma omp parallel default(shared) private(iam, np)
  {
    np = omp_get_num_threads();
    iam = omp_get_thread_num();
    printf("Hello from thread %d out of %d from process %d out of %d on %s\n",
           iam, np, rank, numprocs, processor_name);
  }

  MPI_Finalize();
}
```

Compiling and Linking Mixed MPI and OpenMP Programs

Once you have your example program, you can compile and link it with

- Linux:
  `/afs/slac.stanford.edu/package/OpenMPI/bin/mpicc -openmp hello.c -o hello`
- Solaris: N/A

Running Mixed Programs

- Interactively

```
alfw@morab> export OMP_NUM_THREADS=4
alfw@morab> /afs/slac.stanford.edu/package/OpenMPI/bin/mpirun --mca pls_rsh_agent ssh \
-np 2 -machinefile machinefile.morab -x OMP_NUM_THREADS ./hello
Hello from thread 0 out of 4 from process 0 out of 2 on morab006
Hello from thread 1 out of 4 from process 0 out of 2 on morab006
Hello from thread 2 out of 4 from process 0 out of 2 on morab006
Hello from thread 3 out of 4 from process 0 out of 2 on morab006
Hello from thread 0 out of 4 from process 1 out of 2 on morab001
Hello from thread 3 out of 4 from process 1 out of 2 on morab001
Hello from thread 1 out of 4 from process 1 out of 2 on morab001
Hello from thread 2 out of 4 from process 1 out of 2 on morab001
```

Note that you have to tell OpenMPI to set the `OMP_NUM_THREADS` environment variable for OpenMP for each process it starts using the `-x OMP_NUM_THREADS` command line argument.

- Executing via LSF Batch System

LSF's `bsub` command will pick up the `OMP_NUM_THREADS` and other environment variables and set them before starting your job on the different hosts.

```
alfw@morab> export OMP_NUM_THREADS=4
alfw@morab> bsub -a openmpi -q mpiq  -n 4 ./hello
```

References

- SLAC specific MPI tutorial
- SLAC specific OpenMP tutorial
- Parallel Computing at SLAC

Alf Wachsmann. Last Modified: Mar 22, 2006.