CS 418, Spring 2004
How to use *Rachel* at PSC

To work on the first assignment, you may *optionally* run your jobs and collect performance data on an HP AlphaServer system, named "*Rachel*" at Pittsburgh Supercomputing Center (PSC). This is not a requirement. *Be advised that your jobs may experience unpleasantly long queuing time on Rachel.*

Because OpenMP is supported on both NSCA *Coppper* and PSC *Rachel*, you do not need to modify your source code to run on either platform. (A proof of the strength of the OpenMP standard). However, you do need to use different compilers to compile your programs and write different job scripts to run your programs in batch mode. (This shall not surprise you at all since the two systems are from two competing vendors.)

1 Accessing Rachel

You need to use ssh to connect to the frontend of *Rachel* (rachel.psc.edu). In particular, type:

```
%ssh userID@rachel.psc.edu
```

2 Compiling your programs

To compile OpenMP programs on *Rachel*, use `cc` and specify `-omp` to the compiler. Because the frontend node of *Rachel* have different processors (EV6.7) than the compute node (EV7). You need to tell the compiler to optimize for the compute processors by specifying `-O -fast -tune ev7 -arch ev7`. For example, to compile a program `hello.c` with optimization, type:

```
%cc -omp -O -fast -tune ev7 -arch7 hello.c -o hello
```

3 Running your programs

You can run your program on the frontend node of *Rachel* by typing the program name at command line directly. (Same as the interactive mode on *Copper*).

However, *Rachel* uses Portable Batch Scheduler (PBS) to control access to compute processors for batch jobs. In contrast, *Copper* uses IBM LoadLeveler and Maui as its batch control system. Therefore you need to compose different job scripts for *Rachel*. Although syntactially different, PBS directives serve the same purpose as the LoadLeveler directives, that is, to inform the system how much resources (number of processors, memory size, running time etc.) a job needs. Here is a simple example:

```
#!/bin/csh
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:10:00
#PBS -j oe
#PBS -m abe

./primes 8 50000
```

You may have noticed that there is no explicit memory size specification in this job script. This is because *Rachel* always associates 3.7 GB memory with each processor. Therefore, when the job script requests 8 processors, 3.7GB * 8 = 29.6GB memory is allocated for the job implicitly.

4 Further information

Information on *Rachel* and how to use it can be found at the following URL:

- `http://http://www.psc.edu/machines/marvel/rachel.html`

However, the section of "Running a job" at the above link is rather sketchy. You may find more information about PBS by type:

```
% man pbs
```