Profiling and Debugging Labs

These labs notes are also available in the main slide pack.

Lab 1 - Profiling

- Included in the lab section is linpack.c which is a traditional tool for measuring the performance of High Performance Computing systems, but is also useful for looking at any computing system.
- The goal of this lab is to generate gprof output for this function.
- You will need to
  - Compile this function with the appropriate options (standard build below)

    $ gcc –DDP –DROLL –lm linpack.c –o linpac
  - Run the program to get gmon.out output
  - Run gprof to get the text-ified results
  - Examine the output and identify the function that should most likely be optimized to make linpack faster.
- **Advanced users can fix take a look at timing substring.c and figuring out how to make it less awful!**

Lab 2 – Profiling (Optional)

- p2p_perf.c uses MPI_Wtime to get a accurate timer available in jobs.
- The object of p2p_perf is to provide is to provide information about latency and bandwidth characteristics of an interconnect. You can learn something about latency by sending very small messages and you can learn something about bandwidth by sending very large messages.

1) Examine p2p_perf.c to see the use of MPI_Wtime use.
2) Compile p2p_perf using your preferred compiler
3) Write a batch script that will run p2p_perf to perform a ping-pong between two different nodes. From the results, estimate the maximum bandwidth that you are able to achieve on the Ranger interconnect.

Lab 3 - Debugging

- Example3.c is a buggy print mangler, that you will likely be able to fix.

1) Compile example3.c
2) Execute example3 – it will segfault on you
3) Compile example3.c with debugging turned on
4) Start example3.c in the debugger so you can examine the backtrace information.
5) Set a breakpoint at line 6, and step through the code until you can find what happens.
6) Once you understand the problem, fix the offending line(s) and recompile

If you get hungup, look in example3.fixed for some solutions.

Lab 4 - DDT Lab

- The DDT Lab is a free-form opportunity to get DDT running.
- Open an SSH session with an X-tunnel to ranger.tacc.utexas.edu and get the example code:

```
login3$ cp ~train00/ddt_debug/debug_code.f .
```

- Compile

```
login3% mpif90 -g -O0 debug_code.f -o ddt_app
```

- Load the DDT Module and run ddt

```
login3% module load ddt
login3% ddt ddt_app
```