Demo & Profiling

April 27, 2021

Profiling

- Source instrumentation
- Compile-time instrumentation
- Run-time instrumentation

Source Code Instrumentation

```
start = MPI_Wtime()
…
…
…
end = MPI_Wtime()
```

Source Code Instrumentation

```
/* Start counting events */
if (PAPI_start_counters (Events, num_hwcntrs) != PAPI_OK)
        handle_error(1);
```

NAS Parallel Benchmarks

https://www.nas.nasa.gov/publications/npb.html

- Five kernels
  - IS - Integer Sort, random memory access
  - EP - Embarrassingly Parallel
  - CG - Conjugate Gradient, irregular memory access and communication
  - MG - Multi-Grid on a sequence of meshes, long- and short-distance communication, memory intensive
  - FT - discrete 3D fast Fourier Transform, all-to-all communication
- Three pseudo applications
  - BT - Block Tri-diagonal solver
  - SP - Scalar Penta-diagonal solver
  - LU - Lower-Upper Gauss-Seidel solver

Profiling NPB using IPM

- cd NPB3.3.1/NPB3.3-MPI
- make cg NPROCS=8 CLASS=A
  - executable in bin

- Integrated Performance Monitoring (IPM) profiler
- http://ipm-hpc.sourceforge.net/userguide.html
- ls ipm/bin ipm/lib

Compile-time Instrumentation

- Recompile
  - mpicc … -L<path-to-lib> -l<libname> ..

Using IPM

- export IPM_REPORT=full
- export IPM_LOG=full
- export IPM_REPORT_MEM=yes
- export IPM_HOME=<your ipm directory>

- cd ../NPB3.3.1/NPB3.3-MPI/bin
- mpirun –np 4 ./cg.A.4
- $IPM_HOME/bin/ipm_parse –full <xml file name>
- $IPM_HOME/bin/ipm_parse –html <xml file name>
- firefox <directory>/index.html