Multicore/Multinode hands-on exercises

Nandhana Sakthivel

Exercise 5

Aim

The aim of this exercise is to get an understanding of benchmarking tools like **IMPI Pingpong, STREAM** and **nodeperf**. It contains the following exercises: to measure the latency among MPI processes, measuring bandwidth and running a program using nodeperf.

Introduction

Benchmarking is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it.

IMPI Pingpong Benchmark

The Intel MPI Benchmarks perform a set of MPI performance measurements for point-to-point and global communication operations for a range of message sizes. The generated benchmark data fully characterizes:

- Performance of a cluster system, including node performance, network latency, and throughput
- Efficiency of the MPI implementation used

Latency Measure

Ping Pong is a benchmark created by Intel to measuring latency and throughput of a single message sent between two processes. Latency is the time between initiating a request for a byte or a word in memory until it is retrieves by a processor.

Here, the latency is measured for 1000 iterations for 3 different runs for 0 to 2 bytes. The average latency is found to be **0.65** μs

Bytes	**Run 1**	**Run 2**	**Run 3**	**Min**
0	**0.68**	**0.62**	**0.65**	**0.65**
1	**0.75**	**0.65**	**0.67**	**0.69**
2	**0.75**	**0.66**	**0.67**	**0.69**

Bandwidth Measure

Bandwidth is the maximum rate of data transfer across a given path. To measure bandwidth we will look at the largest messages when bandwidth tends to be flat.

Here, the bandwidth is measured for 1000 iterations for 3 different runs for 1 to 4 bytes. The average bandwidth is found to be **1.43 Mb/s**

Bytes	**Run 1**	**Run 2**	**Run 3**	**Min**
1	1.41	1.39	1.51	1.43
2	2.82	2.82	3.06	2.9
4	5.60	5.64	6.06	5.76

Intranode/Internode Communication Performance

Internode

Here we are estimating the latency between two cores which are on the same socket.

```
#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000         0.76         0.00
            1         1000         0.76         1.26
            2         1000         0.75         2.53
            4         1000         0.76         5.01
            8         1000         0.75        10.11
           16         1000         0.75        20.28
           32         1000         0.78        39.00
           64         1000         0.80        75.87
          128         1000         0.79       153.92
          256         1000         0.83       295.40
          512         1000         1.05       465.24
         1024         1000         1.27       767.11
         2048         1000         1.57      1245.27
         4096         1000         2.41      1621.86
         8192         1000         3.65      2142.82
        16384         1000         7.13      2191.73
        32768         1000        12.65      2469.68
        65536          640        15.48      4036.72
       131072          320        27.98      4466.78
       262144          160        50.06      4994.11
       524288           80       106.34      4701.74
      1048576           40       219.07      4564.67
      2097152           20       442.25      4522.34
      4194304           10       908.65      4402.14
```

Figure 1: Latency and Bandwidth of Internode

Intranode

Here we are estimating the latency between two cores which are on different sockets.

```
#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000        11.50         0.00
            1         1000        11.94         0.08
            2         1000        11.21         0.17
            4         1000        11.29         0.34
            8         1000        12.19         0.63
           16         1000        12.22         1.25
           32         1000        11.69         2.61
           64         1000        12.11         5.04
          128         1000        11.89        10.26
          256         1000        11.84        20.62
          512         1000        11.38        42.91
         1024         1000        11.97        81.61
         2048         1000        12.69       153.92
         4096         1000        12.70       307.61
         8192         1000        14.14       552.53
        16384         1000        15.76       991.34
        32768         1000        20.19      1548.02
        65536          640      1663.98        37.56
       131072          320      1653.12        75.61
       262144          160       829.78       301.28
       524288           80      2524.36       198.07
      1048576           40      5912.70       169.13
      2097152           20     12700.22       157.48
      4194304           10     26546.41       150.68
```

Figure 2: Latency and Bandwidth of Intranode

Comparison

From the result, we can see that for 4194304 bytes the internode has the latency of **988.65** microseconds and bandwidth of **4.4 Gb/s** where as the intranode has the latency of **26546.41** microseconds and bandwidth of

0.15 Gb/s. So, we can see that the communication between cores on the same socket is much better than the cores on different sockets.

Stream Benchmark

The **STREAM** Benchmark is a simple synthetic benchmark program that measures sustainable bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels. Here, we are using **numactl** to bind the threads according to our need. The following graph gives the comparison of memory bind within same socket and different sockets.

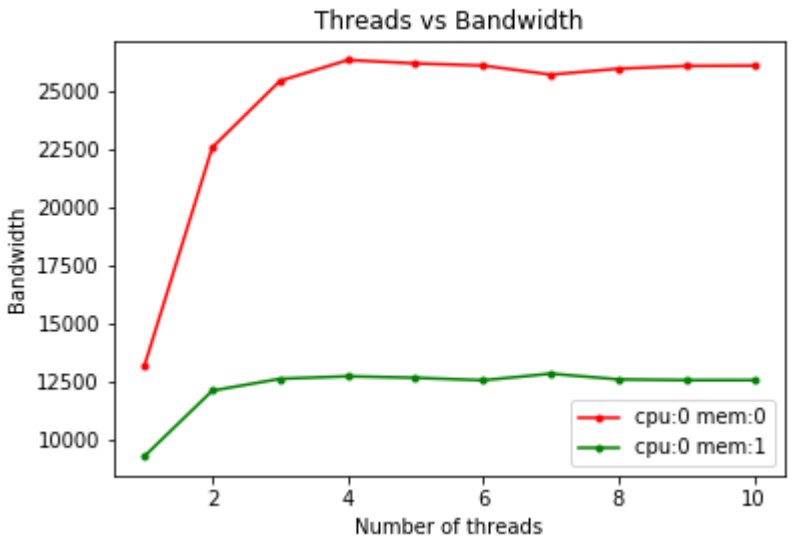

Figure 3: Comparison between close and distant memory

From the graph we can see that the bandwidth of the memory accessed on the same socket (**26 Gb/s**) which is higher than the bandwidth of the memory accessed on differnet socket (**12 Gb/s**)

Nodeperf

nodeperf.c is a small program to test the computational node's performance. This is a simple MPI program that runs a highly optimized version of a double precision general matrix multiply (DGEMM) library routine from the MKL. This routine is also the core of the HPL test.

When the program is compiled using **Intel Compiler** we obtained a performance of **458.219 Gflops** which is **102%** of the peak performance of Ulysses. When the program is compiled using **GNU compiler** we obtained a performance of **27.835 Gflops** which is **6.2%** of the peak performance of Ulysses.

This shows that the Intel compiler provides better performance compared to GNU compiler. Hence, we should opt for the compilers according to our need.