

A performance model for the IMB multi-mode PingPong benchmark

G. Hager G. Wellein

November 7, 2007

1 Introduction

The well-known PingPong benchmark from the IMB (Intel MPI Benchmarks) suite shows a peculiar performance characteristic when run with more than one process per node. If all processes on a node are sending and all processes on the other node are receiving, measured bandwidth is larger than could be expected from purely unidirectional communication, at least in a certain range of message sizes (Fig. 2). When the message size is increased even further, this effect vanishes and bandwidth goes down to the unidirectional level.

2 Model

This behaviour can be explained by assuming that an `MPI_Recv()` on one processor of a node can be overlapped by an `MPI_Send()` by another processor on the same node, implying bidirectional communication at least during a certain phase of the whole process (Fig. 1a). If, however, the message size x gets large, this means that effective latency seen by processes that post their send operation slightly later than others becomes large as well. If, however, messages sent by different processes on a node are *interleaved* using a certain chunk size C , effective latency is reduced significantly. In this case, overlapping send and receive can take place only for the transfer time of a single chunk. Asymptotically, bidirectional transfer is eliminated and measured bandwidth goes down to the unidirectional case (Fig. 1b). Qualitatively this explains the general form of the measured curves in Fig. 2.

Whether above assumptions describe the real situation accurately can be checked by establishing a simple model for data transfer in the PingPong benchmark for the case of 2 processes per node. We define $T_p(x)$ to be the (unidirectional) transfer time for a single message of size x without latency. If B_r is the raw unidirectional bandwidth, then

$$T_p(x) = \frac{x}{B_r} . \tag{1}$$

The number of chunks that a single sent or received message of size x gets divided into is

$$\beta(x) = \begin{cases} 1 & ; x < C \\ x/C & ; x \geq C \end{cases} \tag{2}$$

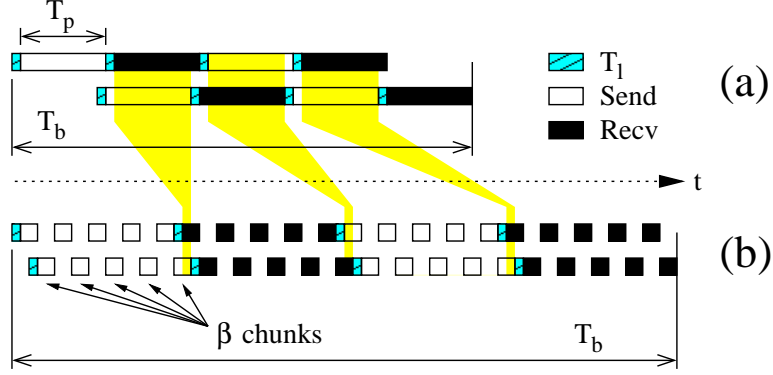


Figure 1: Model for PingPong send/receive overlap. Timing for two processes on one node is shown for the case of $n(x) = 2$. (a) Ideal overlap between send and receive operations. (b) If messages from different processes on a node are interleaved, overlap is limited to one chunk size C per ping-pong.

According to the documented IMB source code (`IMB_settings.h`), the number of times each ping-pong sequence is repeated is

$$n(x) = \min(\text{MSGSPERSAMPLE}, \max(1, \text{OVERALL_VOL}/x)) , \quad (3)$$

where `MSGSPERSAMPLE` is the maximum repetition count and `OVERALL_VOL` is the maximum number of transferred bytes if $x < \text{OVERALL_VOL}$. In the standard benchmark settings, `OVERALL_VOL`=40MBytes and `MSGSPERSAMPLE`=1000. Let T_b be the transfer time for $n(x)$ successive ping-pong communications. Then,

$$T_b(x) = 2n(x)(2T_p(x) + T_1) - (2n(x) - 1) \left[\frac{T_p(x)}{\beta(x)} - T_1 \right] \quad (4)$$

$$= T_p \left[4n(x) - \frac{2n(x) - 1}{\beta(x)} \right] + T_1 (4n(x) - 1) \quad (5)$$

by Fig. 1b. Eq. (4) makes it clear that the overlap effect is annihilated if the transfer time for a single chunk, T_p/β , becomes comparable to T_1 . Finally, the aggregated bandwidth (this is actually twice the bandwidth reported by the benchmark) is

$$b(x) = \frac{4xn(x)}{T_b(x)} . \quad (6)$$

This function of x is parametrized by C , B_r and T_1 . The solid curve in Fig. 2 shows a fit to the measured data for 2 processes per node. The quality of the parameter fit is striking for this case (running one process on each core of a single CPU chip). Although the model can be easily generalized to 4 processes per node, the corresponding data (triangles in Fig. 2) shows a more complex structure and a refinement of the model seems in order. In particular, there are now two message sizes ($C_1 \approx 2^{19}$ and $C_2 \approx 2^{21}$) indicating fundamental changes in transfer characteristics. This could be due to partial interleaving of two pairs of processes on the node.

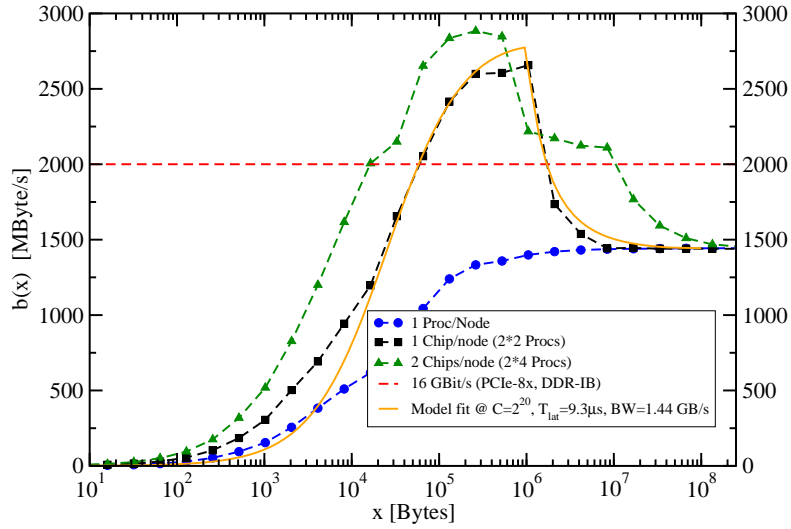


Figure 2: Multi-mode PingPong performance and fit of the model (6) to measured PingPong data for 2 cores per node (solid line). The fit starts at $x > 32768$ because of a different transfer mode for small messages on Intel MPI. The measurements were done on two HP DL140G3 nodes with Xeon 5160 processors and DDR-IB. The `taskset` command was used for pinning processes to cores.

3 Conclusion

A model was derived that could explain the peculiar performance characteristics of the multi-mode PingPong benchmark when using 2 processes per node. The model will have to be refined for the 4ppn case.