Improvement of the Bandwidth of Cross-Site MPI Communication Using Optical Fiber

Kiril Dichev[1], Alexey Lastovetsky[1], Vladimir Rychkov[1]

[1] UCD School of Computer Science and Informatics,
University College Dublin,
Belfield, Dublin 4, Ireland
Kiril.Dichev@ucdconnect.ie, {Alexey.Lastovetsky, Vladimir.Rychkov}@ucd.ie

Abstract. We perform a set of communication experiments spanning multiple sites on the heterogeneous Grid'5000 infrastructure in France. The backbone widely employs high-bandwidth optical fiber. Experiments with point-to-point MPI communications across sites show much lower bandwidth than expected for the optical fiber connections. This work proposes and tests an alternative implementation of cross-site point-to-point communication, exploiting the observation that a higher bandwidth can be reached when transferring TCP messages in parallel. It spawns additional MPI processes for point-to-point communication and significantly improves the bandwidth for large messages. The approach comes closer to the maximum bandwidth measured without using MPI.

Keywords: Heterogeneous Communications, Optical fiber, Point-to-point, MPI

1 Introduction and Related Work

Grid infrastructures such as Grid'5000 can have very complex communication networks involving local area networks as well as optical fiber connections between sites.

We observed that for this infrastructure several TCP connections across sites can be used in parallel with significant increase of bandwidth. We tested the bandwidth of the cross site optical fiber connection without MPI by varying the number of parallel TCP connections. We observed that the bandwidth is better when using 4 or 8 parallel connections and comes close to 1 Gbps. This observation was important for the proposed modification in point-to-point communication.

For MPI point-to-point benchmarks spanning two sites we use NetPIPE with MPI. Since MPI is used for connecting different sites, the underlying communication uses the TCP protocol. The results show a peak bandwidth of around 70 Mbps, which is a much lower bandwidth than any of the TCP benchmarks.

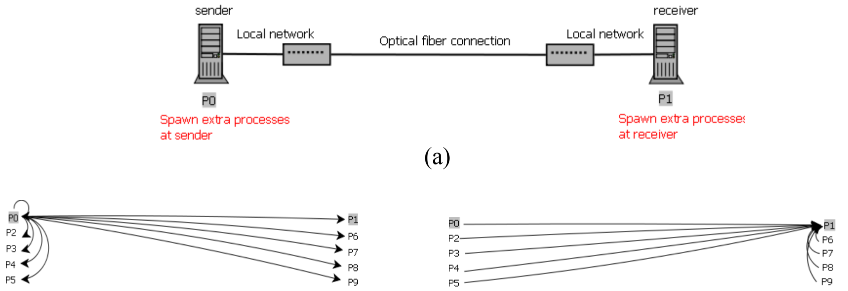
To improve the low bandwidth, we propose a modified MPI point-to-point communication with parallel transfer of different fragments of the same message from the sender to the receiver. Research in this direction has been done in distributed computing. In [1], a similar approach is followed by GridFTP to transfer large data volumes in parallel over the internet by using a number of TCP data streams. The idea is less popular in the high-performance computing domain. Multi-railing is one such example, but it is mostly used for a different setting - when a number of network interfaces are available for the communicating processes at each node [2].

2 Modified MPI Point-to-point Algorithm

We experiment with two methods of parallel transfer of different message fragments. Both cases are implemented on top of the MPI library without internal modifications.

In the first proposed implementation, we use a number of different OpenMP threads, each of which is responsible for submitting a different fragment of a message through MPI point-to-point calls. We used 2 or 4 threads per node and compared this with the original point-to-point calls. The results show no advantage of the multi-threaded implementation. We believe this is due to the internal serialization of point-to-point calls in the MPI library, which prevents true parallelization of the different communicating threads.

We then implement the same idea with MPI processes instead. At the start, a fixed number of extra MPI processes are spawned on each node (Fig. 1a). Any point-to-point communication between processes e.g. P0 and P1 is then divided into two phases – a scatter phase and a gather phase (Fig. 1b and 1c). Each phase is implemented as a linear sequence of point-to-point calls for the different message chunks of the original message. To exploit the parallelism of point-to-point calls, the scatter/gather implementation is a linear sequence of non-blocking sends and receives in MPI.



(a)

(b)

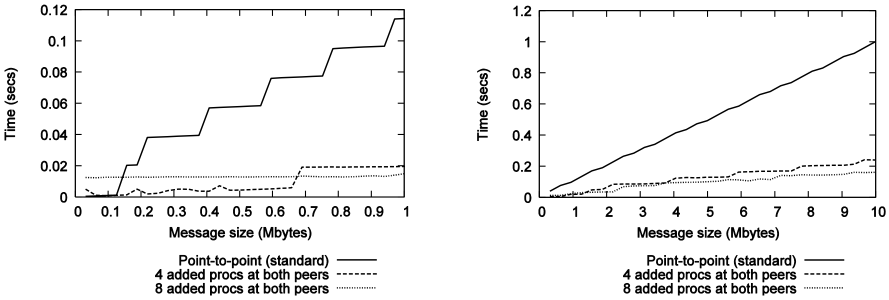
(c)

Fig. 1. Diagram of spawning extra processes at the initialization (a), and transferring a message through two-phases of point-to-point calls on the message chunks – a linear scatter (b) and a linear gather (c)

We present results of experiments with the proposed implementation for message sizes range from 100 KB to 1 MB (Fig. 2a) and from 1 MB to 10 MB (Fig. 2b). In the experiments with the proposed algorithm, we spawn 4 / 8 additional MPI processes per node and involve all of them in the modified point-to-point communication.

The runs with additional processes demonstrate increased bandwidth compared to the original runtime for all message sizes larger than 200 KB, and the improvement is relatively constant for this range. For example, for messages of 10 MB, the MPI point-to-point implementation only reaches around 80 Mbps, while the two-phase version using 16 additional processes (8 at receiver/sender) achieves 498 Mbps, which is more than 6 times increase in bandwidth. This bandwidth is still far from the peak bandwidth we could achieve with a TCP connection (nearly 1 Gbps), but is much closer to it.

The improvements are related to the optical fiber cross-site connection since for a local site run the modified algorithm does not improve the point-to-point bandwidth.



(a) (b)

Fig. 2. Comparison of MPI point-to-point communication and the modified version for messages in the range 100 Kbytes-1 MB (a) and 1 MB-10MB (b)

Acknowledgments. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/IN.1/I2054.

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

References

1. Allcock, B., Bester, J., Bresnahan. J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D., Tuecke, S.: Data Management and Transfer in High-Performance Computational Grid Environments. In: Parallel Computing, vol. 28:p. 749--771 (2002)
2. Moreaud, S., Goglin, B., Namyst, R.: Adaptive MPI Multirail Tuning for Non-Uniform Input/Output Access. In: Proceedings of the 2010 17th EuroMPI conference. LNCS, vol. 6305, p. 239--248 (2010)