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

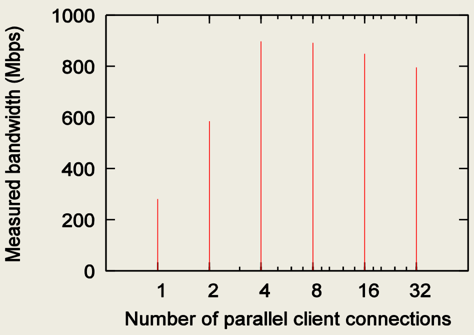
Kiril Dichev, Alexey Lastovetsky, Vladimir Rychkov
Heterogeneous Computing Laboratory
UCD School of Computer Science and Informatics

Introduction

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 demonstrate that this approach improves cross-site point-to-point communication.

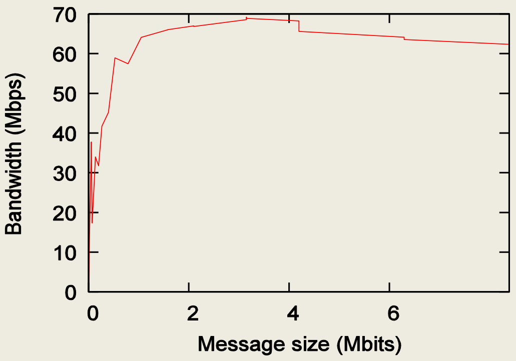
TCP Benchmarks

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.



MPI Benchmarks

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.



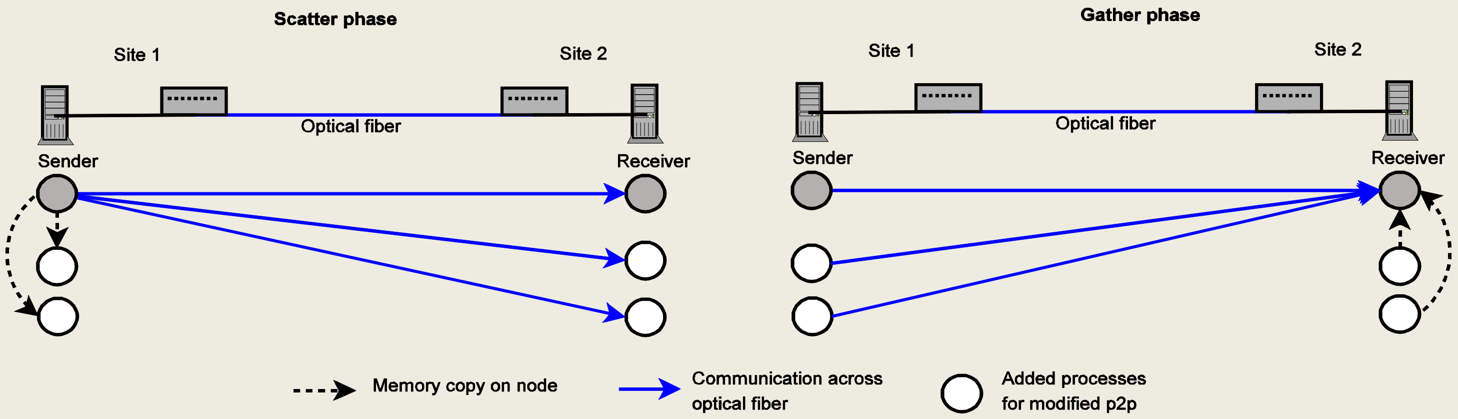
Modified MPI Point-to-point Algorithm

We experiment with two methods of parallel transfer of different message fragments. Both cases are implemented on top of the MPI library.

Thread-Based Implementation

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.

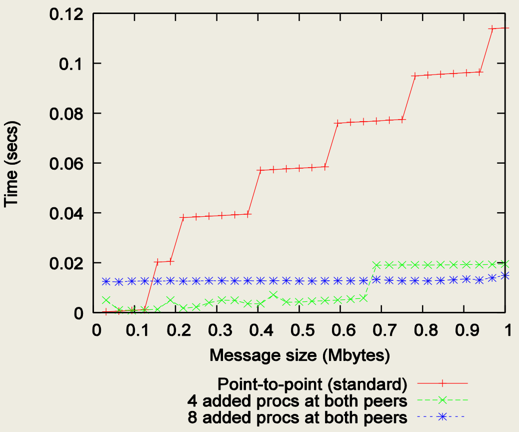
Spawning Additional MPI Processes

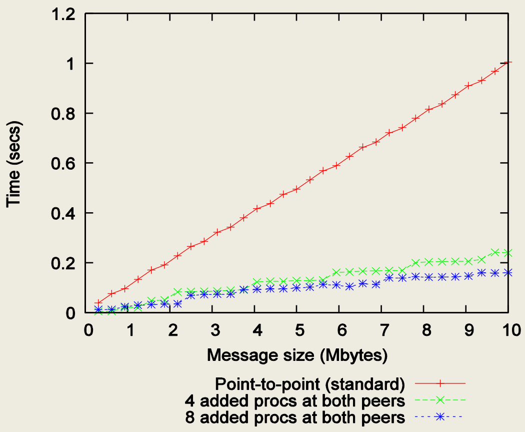


At initialization, a fixed number of extra MPI processes are spawned on sender node and receiver node. Any point-to-point communication between sender and receiver process is then divided into two phases – a scatter phase and a gather phase. Each phase is implemented as a sequence of point-to-point calls 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.

Results

We present results of experiments with the proposed implementation for message sizes range from 100 KB to 1 MB and from 1 MB to 10 MB. 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 non-MPI TCP connection (nearly 1 Gbps), but is much closer to it.





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).