Benchmarking MPI One-sided Communication with SKaMPI-5

Werner Augustin\textsuperscript{1}, Marc-Oliver Straub\textsuperscript{2}, and Thomas Worsch\textsuperscript{3}

\textsuperscript{1} IZBS, Universität Karlsruhe, Germany, \texttt{augustin@ira.uka.de}
\textsuperscript{2} IAKS, Universität Karlsruhe, Germany, \texttt{straub@sb-software.de}
\textsuperscript{3} IAKS, Universität Karlsruhe, Germany, \texttt{worsch@ira.uka.de}

**Summary.** SKaMPI is now an established benchmark for MPI implementations. Two important goals of the development of version 5 of SKaMPI were the extension of the benchmark to cover more functionality of MPI, and a redesign of the benchmark allowing it to be extended more easily (thus matching requests from SKaMPI users). In the present paper we give an overview of the extension of SKaMPI-5 for the evaluation of one-sided communication and present a few selected results of benchmark runs, giving an impression of the breadth and depth of SKaMPI-5.

A look at the source code, which is available under the GPL, reveals that it was extremely easily to extend the SKaMPI with benchmarks for one-sided communication using the infrastructure of version 5.

1 Introduction

“The MPI standard defines a set of powerful collective operations useful for coordination and communication between many processes. Knowing the quality of the implementations of collective operations is of great interest for application programmers. In particular, one has to decide, whether to use predefined collective operations, [...]” This text [Worsch et al.(2003)] describes one of the major motivations for the development of SKaMPI, the Special Karlsruher MPI-Benchmark [Reussner et al.(2002)].

SKaMPI (http://liinwww.ira.uka.de/~skampi/) measures the performance of an MPI implementation on a specific underlying hardware. By providing not simply one number, but detailed data about the performance of each MPI operation, a software developer can judge the consequences of design decisions regarding the performance of the system to be built.

The text also indicates, and benchmark results from a wide variety of machines indeed show, that more often than it should be there is for example a collective operation which is implemented in a suboptimal way, asking for self-made replacements. The development of new implementations for collective operations covered by MPI and for more complex collective operations not
covered by MPI is thus a natural next step. The same holds for the possibility to explore the influence of virtual topologies.

The rest of this paper is organized as follows: In Section 2 we give a short overview over SKaMPI-5 and related work. The main part are Sections 3 to 5. Section 3 discusses some measurements for determining synchronization costs. Section 4 is devoted the standard latency and bandwidth measurements using the pingpong pattern. Less trivial communication patterns are the topic of Section 5. This part is based on the diploma thesis of the second author [Straub(2004)]. We conclude this paper in Section 6.

This paper is an extended version of a paper accepted for EuroPVM/MPI 2005.

2 Overview

SKaMPI-5 offers more than 60 different functions for investigating different performance aspect of one-sided communication routines in an MPI-2 library. Several groups of functions can be identified; many of them accept an MPI_Info and/or an assert parameter which can be used to investigate their influence.

1. Functions for measuring the amount of time needed for calls to synchronization functions like MPI_Win_fence. This includes more complicated cases where for example MPI_Win_wait is delayed by a specified amount of time after another process has called MPI_Win_complete.

2. Functions for measuring the amount of time needed for calls to communication functions like MPI_Win_Put. Latency and bandwidth of for example simple pingpong communications implemented with one-sided communication are also easily determined.

3. Functions for measuring the amount of time needed for more complex communication patterns, e.g. the shift and exchange patterns mentioned above, one-sided implementations of broadcast, alltoall and (all)reduce, exchange of “ghost cells” etc.

Of course, it is impossible to discuss all facets of SKaMPI-5’s one-sided measurements in this paper. We have chosen a few aspects with an emphasis on those not covered by other benchmarks, e.g. those mentioned next.

2.1 Related other MPI benchmarks

There are some benchmarks which allow to do some (in particular latency and bandwidth) measurements. The following ones are available on the WWW and similar in the sense, that they produce data with running times for message exchange operations. Benchmarks like PPerfMark [Mohror and Karavenic(2004)] take a different approach.