Comparison of MasPar MP-1 and MP-2 Communication Operations

Lutz Prechelt (prechelt@ira.uka.de)
Institut für Programmstrukturen und Datenorganisation
Fakultät für Informatik
Universität Karlsruhe, Postfach 6980
D–7500 Karlsruhe, Germany
++49/721/608-4068, Fax: ++49/721/694092

April 23, 1993

Technical Report 16/93

Abstract

Report 01/93 [Pre93] describes the findings of a series of communication measurements performed on a MasPar MP-1 series MP-1216A machine. The current report covers the same measurements performed on a MP-2 series MP-2216 machine. It compares the results and outlines and discusses the main differences. In these measurements, raw router communication was sometimes faster, sometimes slower on the MP-2 than on the MP-1, depending on the parameters of the communication requested. The relative performance of the MP-2 varied between 93% and 120%. Xnet communication was faster in all cases (performance 100% to 175%). Complex functions from the communication library were also always faster (performance 100% to 180%). Some of these results contradict technical specifications for the MP-1 and MP-2 published by MasPar.

Contents

List of Figures

1 Introduction

1.1 The scope of this report

A few weeks after my technical report [Pre93] had been published, Jan Berger Henriksen from the University of Bergen, Norway, contacted me. He asked whether he could run my measurement program on Bergen's new MasPar MP-2 machine. I agreed and he made the results available to me. I compiled the results into this report. The machine used was a MP-2216 (the 16384 processor version of the MP-2), which corresponds directly to the MP-1216A (the 16384 processor version of the MP-1) that was used for the measurements in the old report. This history has the following consequences:

- This report covers exactly the same measurements as [Pre93], which I will now call the *old report*.
- For easier overview, it also contains exactly the same set of diagrams. All diagrams use the same scales and nomenclature as in the old report. Only the actual data points are different.
- Since this model of comparison tends to introduce a lot of redundancy, I have stripped most of the textual contents of the old report from this one. Thus, it is not useful to read this report alone, you need the old report as a reference to understand what the measurements mean. In particular, often-used phrases such as *"as we can see in figure x"* usually meann *"as we can see by comparing figure x from this report with figure x from the old report"*.
- The discussion of the MP-2 measurement results does not cover the absolute effects that can be observed in MP-2 communication behavior but only the relative effects in comparison to the MP-1.

The report is structured as follows: After this introduction follows a comparison of the technical specifications of the two machines and a very short discussion of the effects that can be *expected* from their differences.

The other three sections (2 to 4) follow the same structure as in the old report: They discuss, in order, the raw router communication, the xnet communication, and the higher communication library functions. A short summary of the observations follows in section 5.

To ease reading, the sequence of the diagrams from the old report was preserved, as was the substructuring of the individual sections. This decision resulted in some figures not being referenced at all and some subsections being almost completely empty. All comparisons are expressed in *percent relative performance of the MP-2 compared to the MP-1*, i.e. the performance of the MP-1 is 100% for each operation measured.

1.2 Technical specification comparison of MP-1 and MP-2

The main difference between the MP-1 and MP-2 series of MasPar machines is that for the MP-2 the CPU chip has been completely redesigned. This greatly enhances its computational performance. The communication network, on the other hand, was kept almost without (externally visible) change.

Table 1 shows some technical specifications taken from MasPar marketing material. These specifications give an idea of the differences between the machines that we can expect to observe in communication measurements. The most important lines to look at are those that specify the router and xnet bandwith and the memory bandwith.

	Unit	MP-1	MP-2	relative
Raw computation				
32 bit Integer	MIPS	26000	68000	261%
32 bit Floating Point	MFLOPS	1200	6300	525%
64 bit Floating Point	MFLOPS	550	2400	436%
Memory bandwith				
direct addressing	MB/s	11000	20000	182%
indirect addressing	MB/s	4000	7800	195%
Communication				
xnet bandwith	MB/s	23000	20000	87%
router bandwith	MB/s	1300	1300	100%

Table 1: MasPar MP-1216 and MP-2216 technical specifications

They tell us that we can expect to find memory-to-memory router communication slightly faster on the MP-2 than on the MP-1, in particular when the packet size is high (due to higher memory bandwith) and in particular when indirect addressing is used. Xnet communication may be somewhat faster on MP-2 than on MP-1 for the same reasons, but may as well sometimes be a bit slower, because of the reduced bandwith of the xnet on the MP-2. For more complex library operations that use communication we can expect to find increased performance on the MP-2, because the high raw processing speed of its CPU will accelerate the computations embedded within these operations by factors of 2 or 3.

As we will see, these expectations were not always met by the actual measurements: Router communication happened to be slower on the MP-2 in some cases. We can only speculate about the reasons for this lack of speed: one possibility is that the specifications MasPar published are simply not quite correct, another possibility would be that the MP-2 machine had some hardware problems during the measurements [1], a third explanation would be that synchronization of CPU and communication network is more difficult on the MP-2 and eats up a lot of time. Whatever the actual reason was, in the following I assume that the differences are real, i.e., not produced by hardware problems. This assumption is supported by figure 1. It shows that over 100 repetitions of the same communication operation (a permutation with changing communication pattern), the MP-2 averages about 4900 ticks while the MP-1 averages about 4800 ticks and the variance of both samples is very similar. From this I conclude that the MP-2 is indeed about 2% slower on this operation than the MP-1.

2 Raw router communication

In this section we will examine the behavior of the router statement and the rsend and rfetch library functions.

2.1 The router statement

From figure 2 we find that router send (using a permutation as the communication pattern and moving data from memory and to memory) is slightly slower on the MP-2. The relative performance for full

[1] Since the MasPar machines can transparently perform automatic retries when communication operations fail, a machine can get a little slower (instead of failing) when some component has intermittant errors.

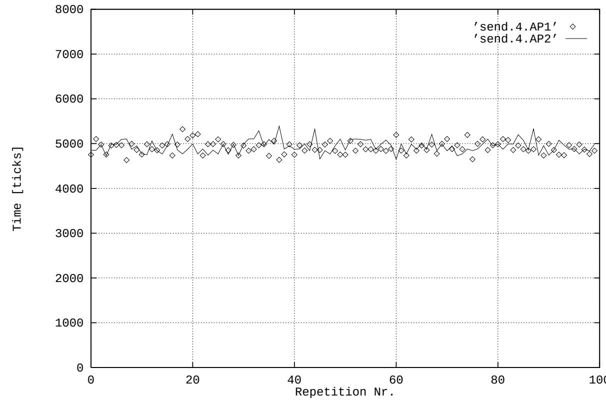
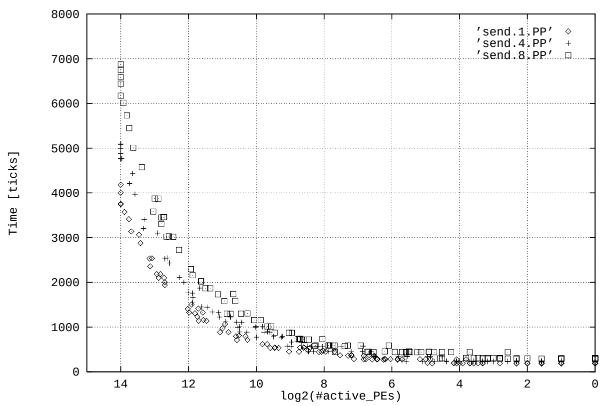


Figure 1: router permutation sensitivity



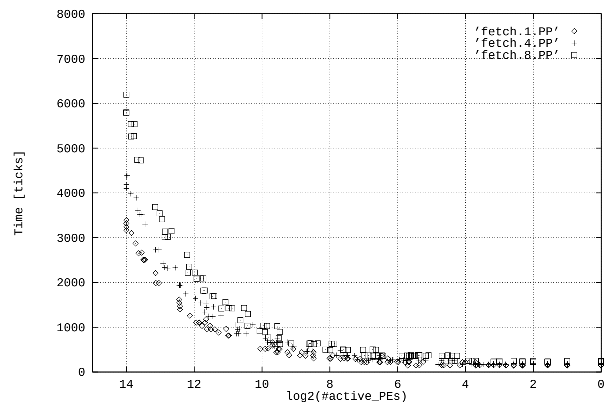


Figure 2: router send

Figure 3: router fetch

PE activity are on the order of 6300/6700 (=94%) for 8 byte packets, 4800/4900 (=98%) for 4 byte packets, and 3800/3950 (=96%) for 1 byte packets.

Note that these values are derived from averaging of small samples of about 4 measurement values which differ by up to 11%. These variations mean that the results can only give an idea of the overall behavior but are not precise in the sense of statistical significance. The absolute values given in the text are also rounded somewhat to ease their perception.[2]

For the router fetch corresponding to the above send, on the other hand, we find that the MP-2 is slightly faster, as figure 3 shows. The relative performance is 103% for 8 byte packets, 105% for 4 byte packets, and 100% for 1 byte packets.

That send performance is below 100% while fetch performance is above 100% is no accident, as we can see from figure 6: For a constant number of router steps, that is, independent of the actual permutation used, the same differences show up: For instance for 35 router steps, the send took about 3900 ticks on the MP-1 but almost 4000 ticks on the MP-2, a relative performance of about 98%. For 39 fetch router steps, on the other hand, the MP-1 took almost 4000 ticks while the MP-2 needed only 3890 — a relative performance of 103%.

As expected, figure 5 indicates that the same number of router steps are required for a given problem

[2] This comment applies to most other comparisons as well. Sometimes, however, the number of relevant data points is much larger.

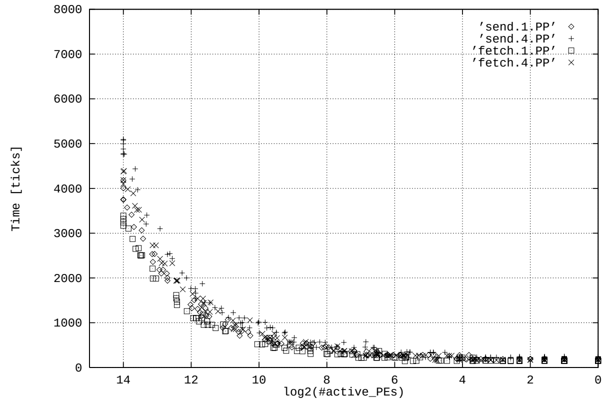
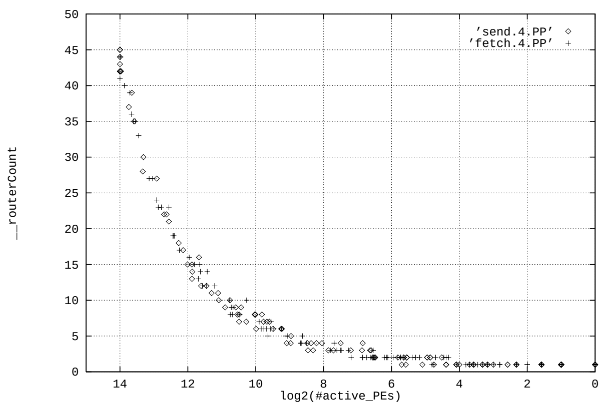


Figure 4: router send vs. fetch



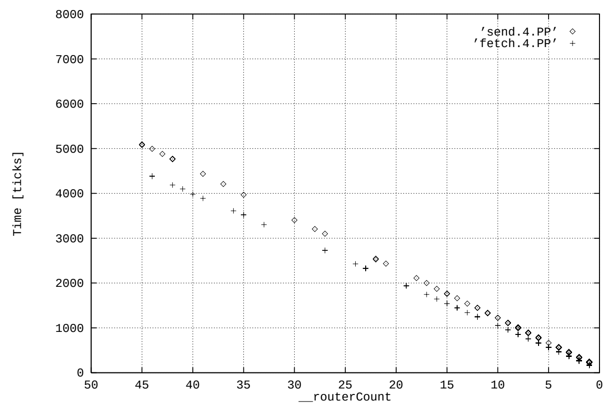


Figure 5: send/fetch routerCount by PE activity

Figure 6: router send/fetch by routerCount

on both machines. This was to be expected since the communication network functions algorithmically identical on both machines.

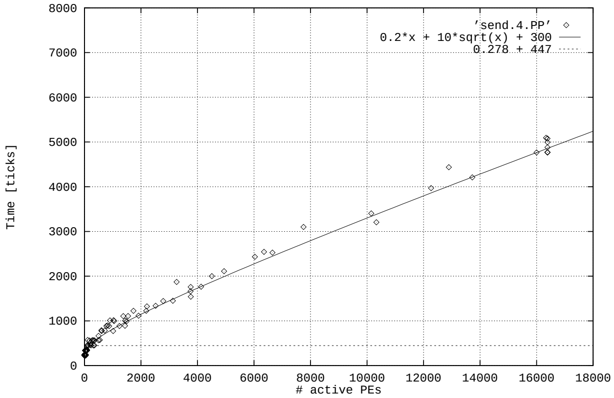
Figures 7 and 8 depict the analytic approximation describing the dependency between the number of active PEs and the time used for a 4-byte permutation send on the MP-1. We see that for the MP-2 this approximation is a bit too small for high PE activity (right part of figure 7) and a bit too large for low PE activity (left part of figure 8).

2.2 Using other activation patterns

Using regular activation patterns instead of stochastic ones does not reveal any interesting new effects (figure 9).

2.3 Using other communication patterns

Using regular communication patterns instead of random permutations also does not reveal any interesting new effects (figure 10).



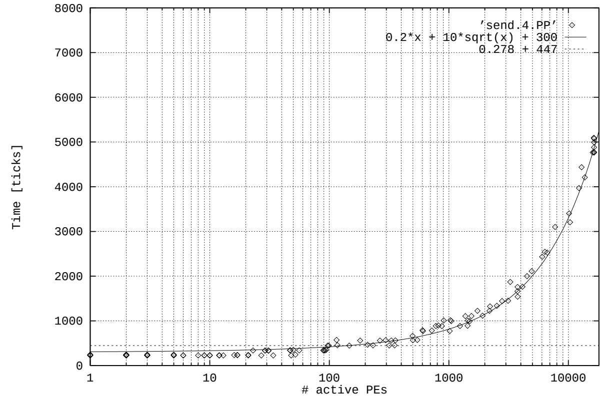
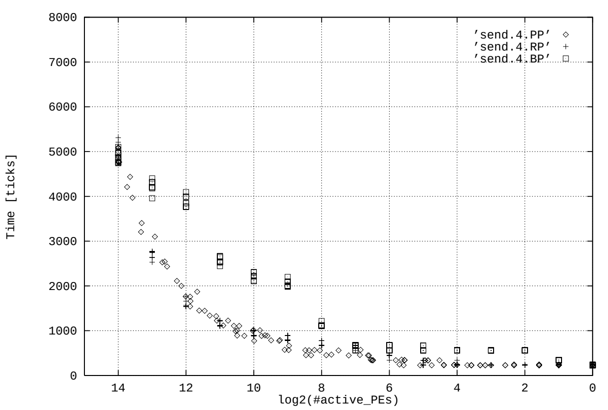


Figure 7: router send analytic approximation (linear scale)

Figure 8: router send analytic approximation (logarithmic scale)



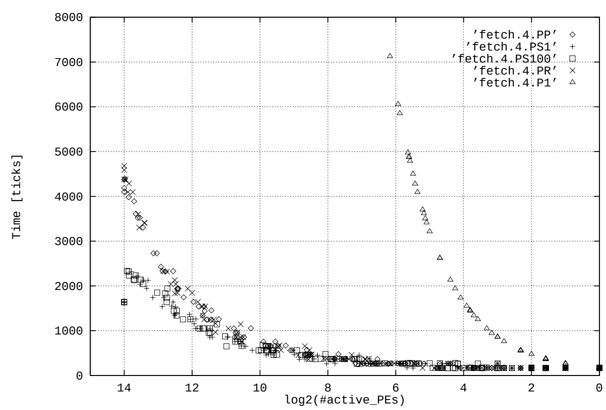
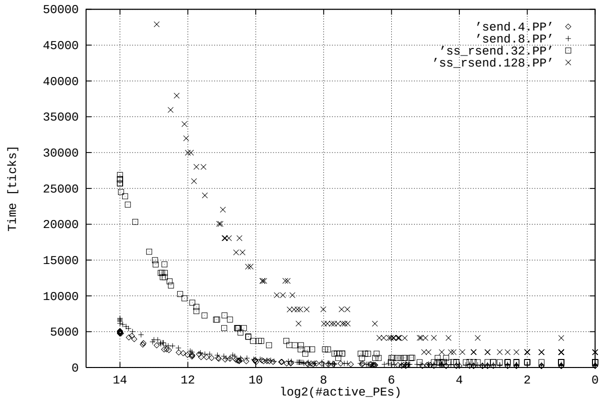


Figure 9: router send with probabilistic vs. regular activity

Figure 10: router fetch with different communication patterns



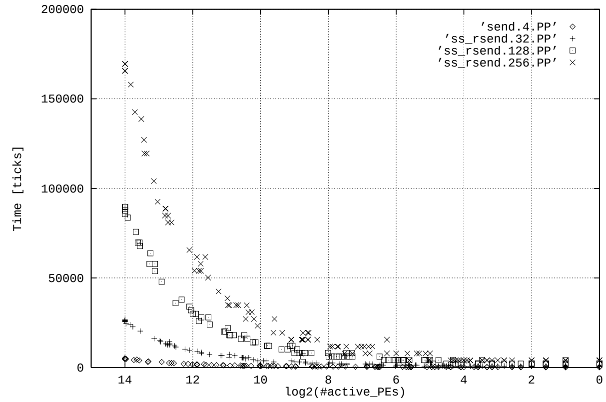
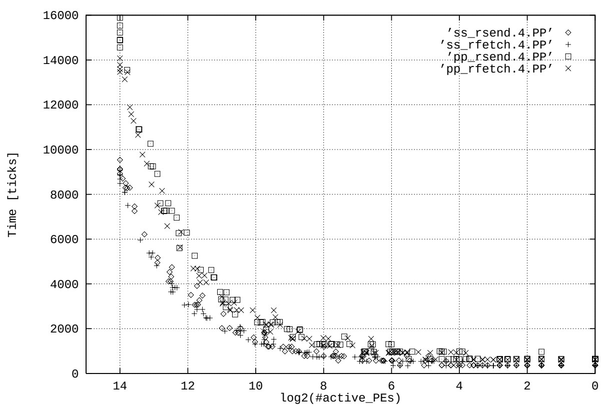


Figure 11: router send large packets

Figure 12: router send very large packets



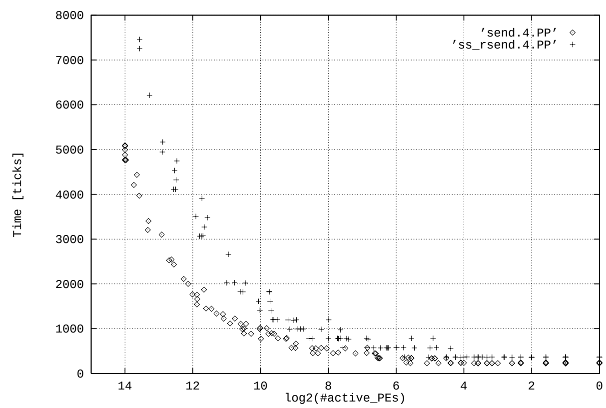


Figure 13: rsend vs. rfetch

Figure 14: router send vs. ss_rsend

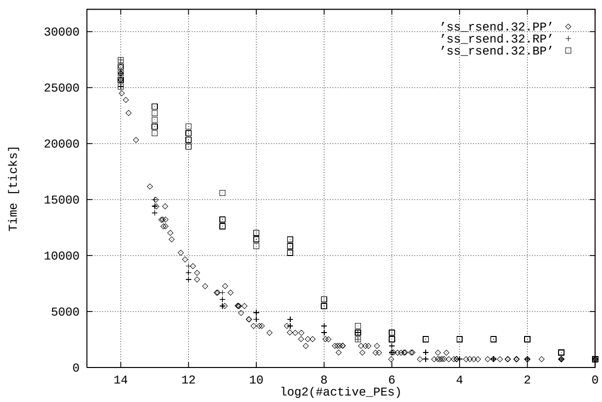
2.4 rsend and rfetch

2.4.1 sending large packets

The timings for sending larger packets of data are given in figures 11 and 12. They indicate that large packets may result in significant increase in the relative performance of the MP-2. This is due to the higher memory bandwith. For instance, a 256-byte send with 2^{13} PEs active uses only about 100000 ticks instead of 120000 on the MP-1 (120% performance). 128-byte send with 2^{12} PEs active uses 30000 instead of 36000 (120% performance). 32-byte sends perform at about 115%.

2.4.2 rsend vs. rfetch

Figure 13 indicates that ss_rsend performs at 100% for 4-byte packets and ss_rfetch at about 103%. For the indirect-addressing versions pp_rsend and pp_rfetch, the values are similar.



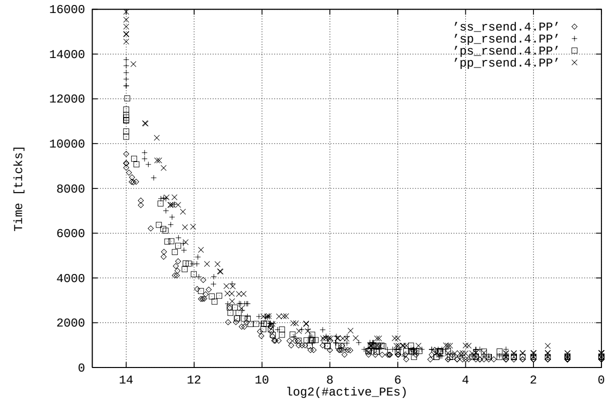
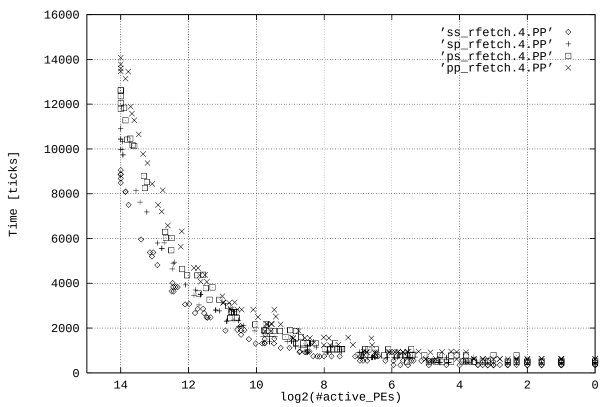


Figure 15: rsend with probabilistic vs. regular activity

Figure 16: rsend with singular and plural pointers (4 bytes)



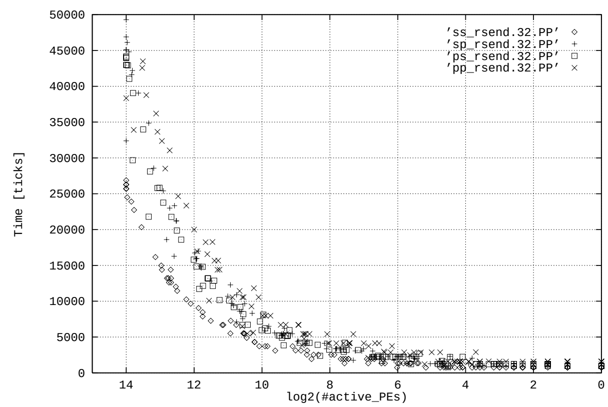


Figure 17: rfetch with singular and plural pointers (4 bytes)

Figure 18: rsend with singular and plural pointers (32 bytes)

2.4.3 rsend vs. router statement

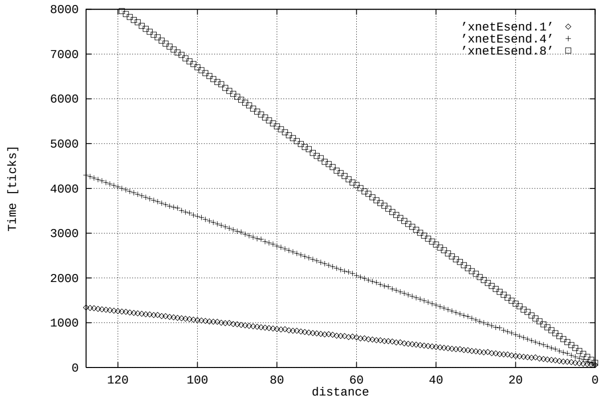
Given the previous results, the comparison of rsend versus router statements (figure 14) yields no additional information.

The results using different activation patterns on rsend (figure 15) are consistent with the results from figure 9 and 11.

2.4.4 singular vs. plural pointers

In figures 17 and 17 we find all four variants of the xx_rsend as well as the xx_rfetch command to perform at roughly 100% for 4-byte packets.

When sending larger packets of 32 bytes, very surprising observations can be made (figure 18): While ss_rsend performs at about 115%, as seen before, the other variants are *not* above 100% on the MP-2! For instance at 2^{12} PEs active, sp_rsend performs at about 94%, ps_rsend at 100%, and pp_rsend at 93%. For more than 2^{12} PEs active, the measurements deviate very much, so that it is not possible to give a good estimation of performance, but it seems that the performance tends to be below 100%. I am not able to explain this effect.



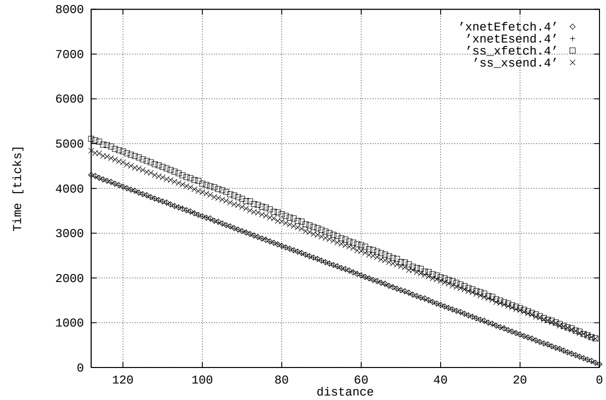


Figure 19: xnet send

Figure 20: xnet send vs. fetch

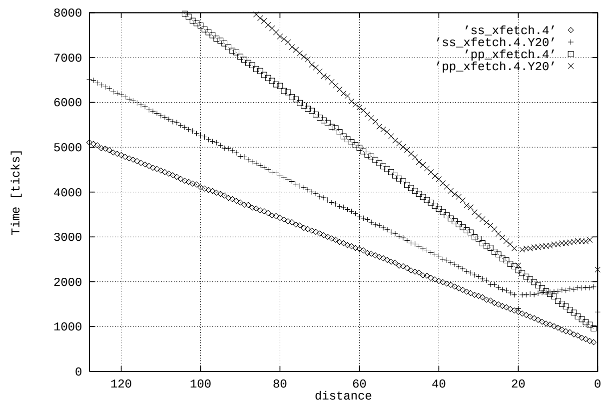


Figure 21: xfetch in nonstandard direction

3 xnet communication

For a simple xnet send operation we find in figure 19 that for 1-byte packets the MP2 performs at 100%, for 4-byte packets at 105%, and for 8-byte packtets at about 101%

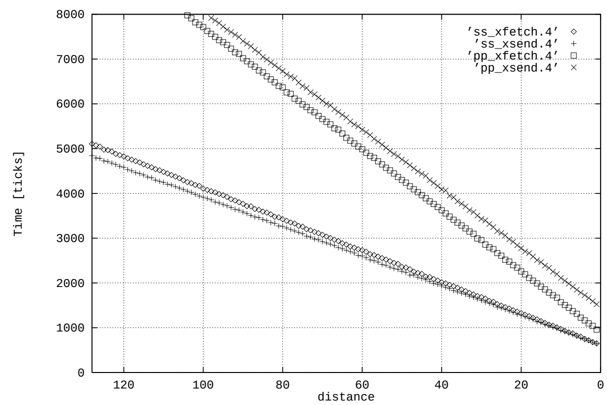
Figure 20 shows that both, ss_xsend and ss_xfetch perform at about 104% for 4-byte packets. A bit surprising (although only when judged in the light of the previous results) is the high performance for pp_xfetch (figure 21) of 116%.

For xfetch in non-straight directions, figure 21 indicates a constant acceleration of about 100 ticks for the ss_xfetch variant and 550 ticks pp_xfetch, resulting in a performance of 109% for distance 60.

For standard directions, we find 100% performance for the ss_xsend and ss_xfetch operations (figure 22). The pp_xsend/pp_xfetch operations show a performance increase that consists of two parts: a constant acceleration of 1200 and 650 ticks, respectively, plus a slightly faster communication per distance step. This results in relative performances of 175% and 168% for distance 0 or 126% and 116% for distance 60 (also figure 22).

A similar although much less drastic effect occurs for the sp_xxx and ps_xxx variants of xsend and xfetch. The relative performance for these cases varies between 133% and 108% (figure 23).

For larger 32-byte packets, we get a performance of 114% from pp_xfetch and 120% from pp_xsend for



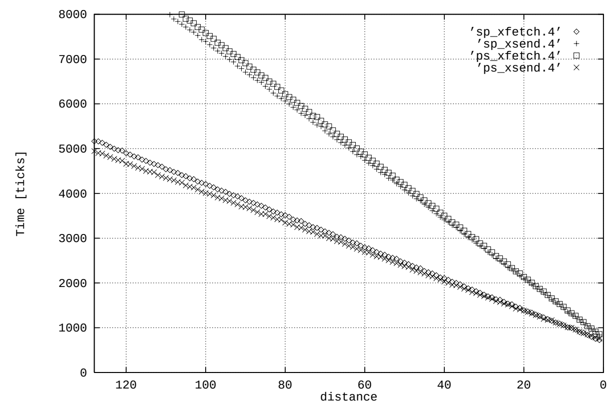


Figure 22: ss_xfetch/send vs. pp_xfetch/send

Figure 23: sp_xfetch/send vs. ps_xfetch/send

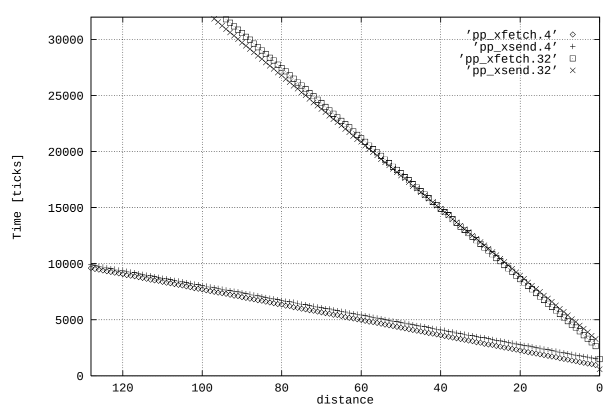


Figure 24: pp_xfetch/send for small and large packets

distance 60 (figure 24).

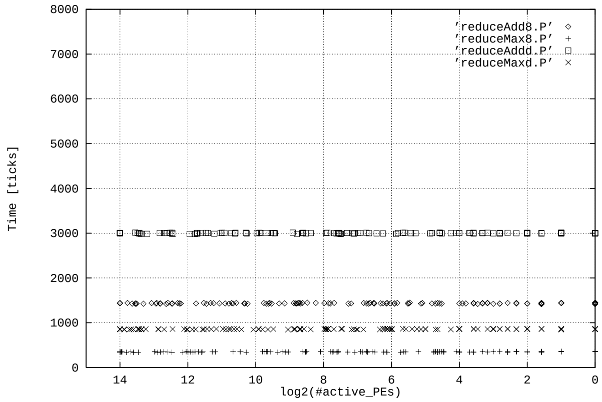
4 Communication library functions

4.1 reduce

The performance of the various reduce operations that were tested (figure 25) were 114% for reduceMax8, 133% for reduceMaxd, 107% for reduceAdd8, and 180% for reduceAddd.

4.2 scan

Scan operations (figure 26) are not faster on the MP-2 for 8 bit integers. For 64-bit floating point values there is a small acceleration, provided there are only a few segments, i.e. the computation part of the call is dominant. In this case (2^2 segments) the performance is 107% for scanMaxd and 112% for scanAddd. For floating point scans on a large number of small segments (2^{12}), performance drops to about 100% again.



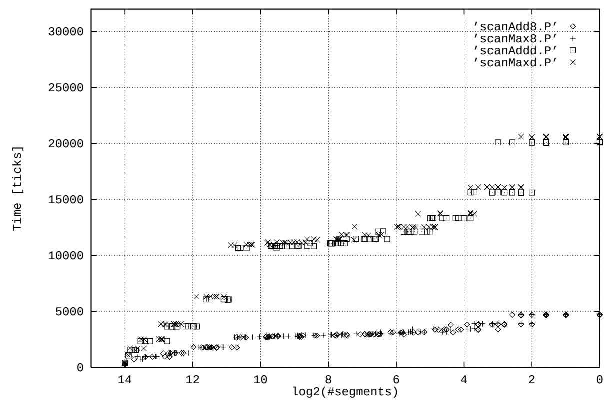
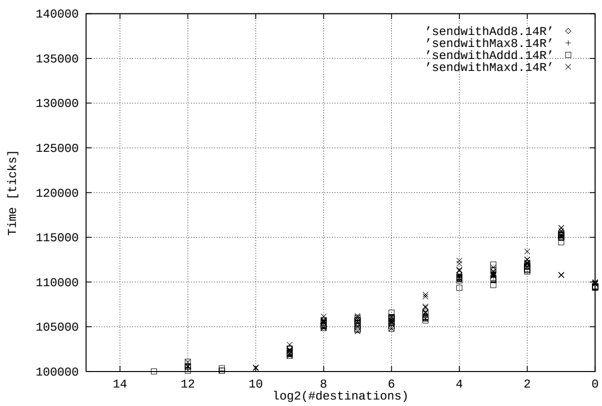


Figure 25: reduce

Figure 26: scan



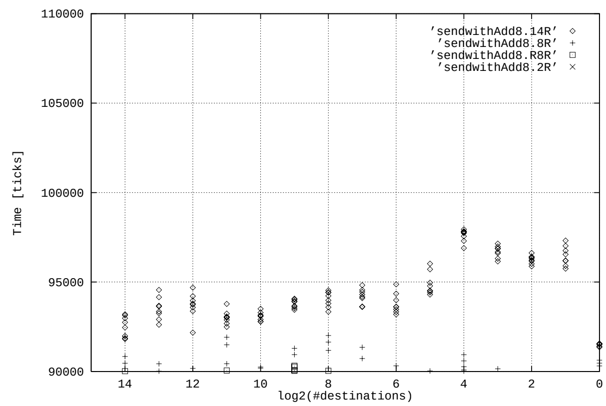


Figure 27: sendwithAdd vs. sendwithMax

Figure 28: sendwithAdd with different PE activity

4.3 sendwith

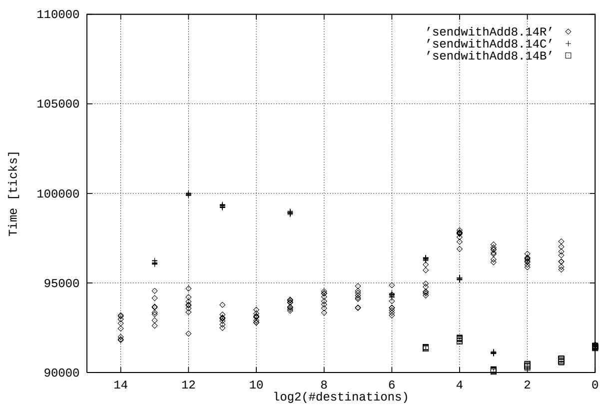
For the variants of the sendwith operation, the performance differences are not as dramatic as figures 27 and 28 may suggest, because these diagrams do not use 0 as the lower end of the vertical axis. Nevertheless, the MP-2 performs at about 106% to 109%, regardless of the kind of sendwith operation performed and regardless of the number of destinations.

4.4 enumerate, selectOne, selectFirst

The performance for enumerate is about 103%, for selectOne and for selectFirst 113%. Since the actual diagrams are still boring, they are still not shown here.

4.5 psort, rank

For the rank and psort operations, as shown in figure 30, we find roughly the following performance scores: rankd 109%, psortd 102%, rank8 132%, and rankd 105%.



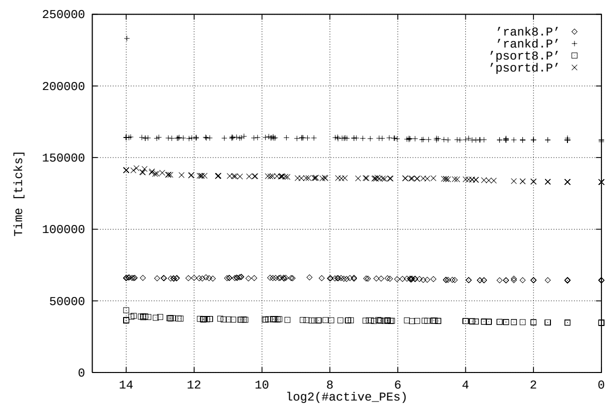


Figure 29: sendwithAdd with different destination patterns

Figure 30: rank and psort

5 Summary of Results

Similar to the measurements made on the MP-1, the comparison of the same measurements for MP-1 and MP-2 yielded some results that were surprising. The most significant of these is the fact that raw router communication can be slower on the MP-2 than it is on the MP-1.

Looking at the big picture, the message that can be derived from the results in this report is the following: For realistic programs, one can usually expect a relative performance of between 100% and 120% from the communication operations, provided they are fed from and deliver to memory. This value is better than one would expect at first when looking at the technical specifications alone and stems mostly from the almost-twice-as-high memory bandwith of the MP-2.

Other observations to be made from the measurements are:

1. Router fetch is slightly cheaper than send on the MP-1. This difference is noticeably bigger on the MP-2.
2. Router send is slower on the MP-2 than on the MP-1 for small packets (98% relative performance for 4-byte packets), but faster for large packets (115% relative performance for 32-byte packets). Even for 32-byte packets, relative performance mysteriously drops below 100% when indirect addressing (via a plural pointer) is used.
3. Xnet communication was not slower on the MP-2 than on the MP-1 for any of the measurements used in this report, despite the fact that the technical specifications published by MasPar announced reduced bandwith of the xnet on the MP-2.
4. The constant cost of an xfetch or xsend operation is significantly lower on an MP-2. This can result in a performance above 150% for small distances.
5. All complex communication functions from the MPL library are faster on the MP-2 than on the MP-1. For some cases this advantage is only in the per mille range (e.g. scan with very many segments), for others it is almost factor 2 (e.g. reduceAddd: 180%).

References

[Pre93] Lutz Prechelt. Measurements of MasPar MP-1216A communication operations. Technical Report 1/93, Institut für Programmstrukturen und Datenorganisation, Universität Karlsruhe, D-7500 Karlsruhe, Germany, January 1993.