Program typu ping-pong

Rafał Selewońko

7 grudnia 2013

1 Zadanie

Aplikacja MPI składa się z dwóch procesów: P0 oraz P1. Obydwa procesy wymieniają się komunikatami stosując wzorzec wymiany jak w grze w ping-ponga. P0 wysyła komunikat do P1 następnie P1 odsyła komunikat do P0. Ta sekwencja wymiany jest powtarzana odpowiednią (bardzo dużą) liczbę iteracji, tak aby program wykonywał się przez kilka sekund. Czas pracy programu jest obliczany przy pomocy funkcji MPI_Walltime. Dzieląc ten czas przez liczbę iteracji pomnożoną przez dwa (ponieważ komunikat podróżuje w obie strony) otrzymujemy czas transmisji komunikatu w jedną stronę. Program powinien wykonywać szereg takich pomiarów dla rosnącej długości komunikatów, tak abyś mógł w stanie sporządzić wykres czas komunikatu=f(długość). Z kolei dzieląc długość komunikatu przez czas jego transmisji otrzymujemy przepustowość w bajtach/s dla danej długości komunikatu. Należy sporządzić również wykres przepustowość=f(długość). Program powinien implementować test ping-pong w dwóch wersjach: przy pomocy blokujących operacji MPI_Send/MPI_Recv oraz przy pomocy nieblokujących MPI_Isend/MPI_Recv wraz z MPI_Wait. Wykonaj pomiar na klastrze i sporządź raport z wykonania zadania zawierający wydruk programu + obydwa wykresy. Porównaj otrzymane wyniki z wynikiem polecenia ping oraz z parametrami sieci Gigabit Ethernet. Porównaj wynik na czas transmisji komunikatu z modelem przedstawionym na wykładzie. Czy przy pomocy MPI możliwe jest osiągnięcie maksymalnej teoretycznej przepustowości sieci? Jeżeli nie to wymień przyczyny, które Twoim zdaniem to uniemożliwiają.

2 Rozwiązanie

main.cpp

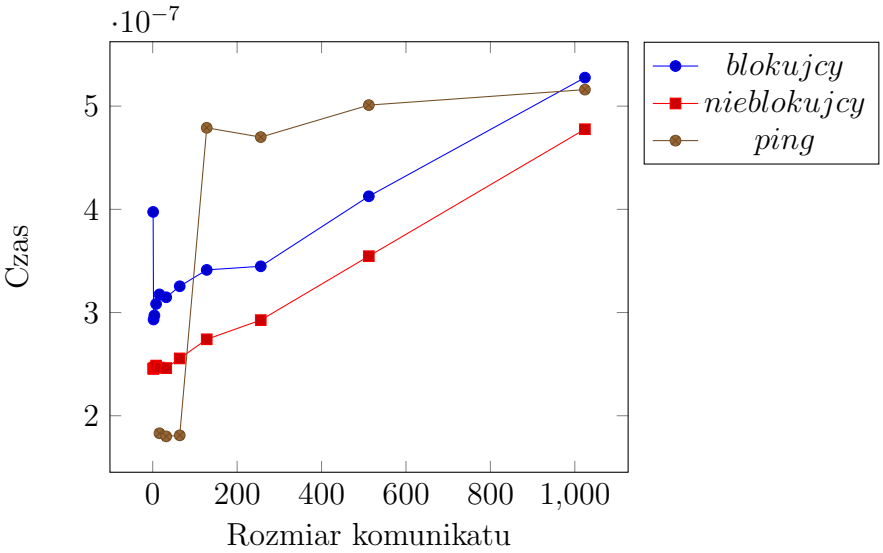
```
1  // Copyright (c) 2013 Rafal Selewońko <rafal@selewonko.com>.
2
3  #include <mpi.h>
4
5  #define ILE 65536
6  #define BUFFER_SIZE 1024
7  #define WRITE_TO_FILE 1
8
9
10 int main(int argc, char *argv[]) {
11     int npes;
12     int myrank;
13     int otherrank;
14
15     char buffer1[BUFFER_SIZE];
16     char buffer2[BUFFER_SIZE];
17
18     double t1;
19     double t2;
20
21     short int v;
22     int i;
23     unsigned int j;
24
25
26
27     // Inicjalizacja podsystemu MPI
28     MPI_Init(&argc, &argv);
29
30     // Pobierz rozmiar globalnego komunikatora
31     MPI_Comm_size(MPI_COMM_WORLD, &npes);
32
33     // Pobierz numer procesu w globalnym komunikatorze
34     MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
35
36     otherrank = myrank ? 0 : 1;
37
38
39     FILE *file;
40
41     if (WRITE_TO_FILE) {
42
43         if (myrank == 0) {
44             file = fopen("wyniki_proces_1.txt", "w+");
45         } else if (myrank == 1) {
```

```
46              file = fopen("wyniki_proces_2.txt", "w+");
47          }
48      } else {
49          file = stdout;
50      }
51      fprintf(file, "id procesu, czy nieblokujacy, dlugosc, czas,
            przepustowosc\n");
52
53      for (v = 0; v < 2; v += 1) {
54          for (i = 1; i <= BUFFER_SIZE; i <<= 1) {
55              t1 = MPI_Wtime();
56
57              for (j = 1; j <= ILE; j += 1) {
58                  MPI_Status status[2];
59                  if (v == 0) {
60                      MPI_Request request[2];
61                      MPI_Isend(&buffer1, i, MPI_BYTE, otherrank,
                            13, MPI_COMM_WORLD, &request[0]);
62                      MPI_Irecv(&buffer2, i, MPI_BYTE, otherrank,
                            13, MPI_COMM_WORLD, &request[1]);
63                      MPI_Waitall(2, request, status);
64                  } else if (v == 1) {
65                      MPI_Send(&buffer1, i, MPI_BYTE, otherrank,
                            13, MPI_COMM_WORLD);
66                      MPI_Recv(&buffer2, i, MPI_BYTE, otherrank,
                            13, MPI_COMM_WORLD, status);
67                  }
68              }
69
70              t2 = MPI_Wtime();
71
72              fprintf(file, "%d, %d, %d, %e, %f\n", myrank, v, i,
                    (t2 - t1) / (ILE * 2), i / (t2 - t1));
73          }
74      }
75
76      if (WRITE_TO_FILE) {
77          fclose(file);
78      }
79
80      MPI_Finalize();
81      return 0;
82  }
```

3 Wyniki badań

wielkosc	blokujacy	nieblokujacy	ping
1	3.975074e-07	2.455981e-07	
2	2.932502e-07	2.454362e-07	
4	2.971865e-07	2.465749e-07	
8	3.083114e-07	2.486267e-07	
16	3.175810e-07	2.466659e-07	1.83e-07
32	3.147961e-07	2.462602e-07	1.80e-07
64	3.254700e-07	2.555389e-07	1.81e-07
128	3.413152e-07	2.740780e-07	4.79e-07
256	3.448095e-07	2.926863e-07	4.70e-07
512	4.126596e-07	3.546302e-07	5.01e-07
1024	5.276343e-07	4.776539e-07	5.16e-07

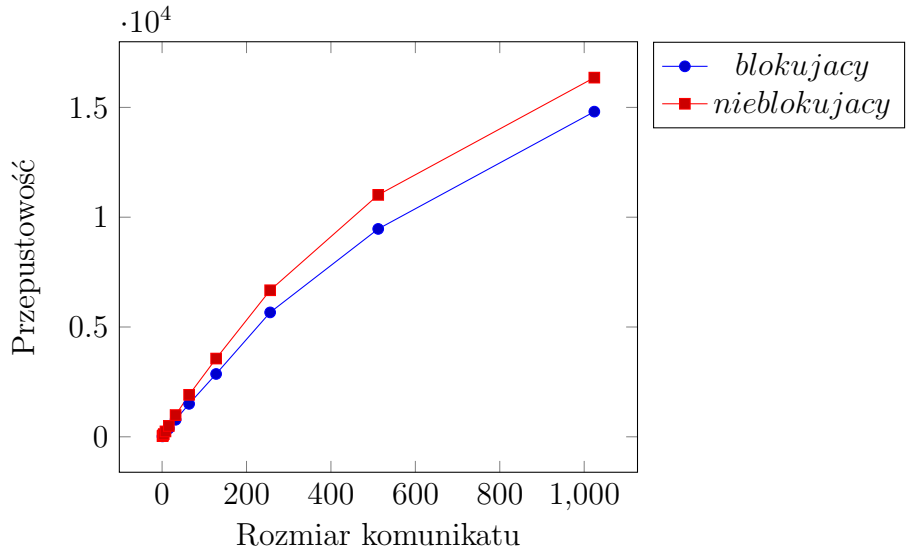
Tabela 1: Czas w zależności od rozmiaru komunikatu.



Rysunek 1: Wykres czasu w zależności od rozmiaru komunikatu.

wielkosc	blokujacy	nieblokujacy
1	19.193088	31.064546
2	52.033347	62.170073
4	102.688309	123.765942
8	197.965922	245.489099
16	384.375367	494.881229
32	775.551557	991.392775
64	1500.234484	1910.790239
128	2861.175187	3563.081791
256	5664.360412	6673.099971
512	9466.035070	11014.996143
1024	14806.658034	16355.985483

Tabela 2: Czas w zależności od rozmiaru komunikatu.



Rysunek 2: Wykres czasu w zależności od rozmiaru komunikatu.