Sprawozdanie: Testy systemu transputerowego

Szymon Francuzik Stanisław Jankowski

24 listopada 2008

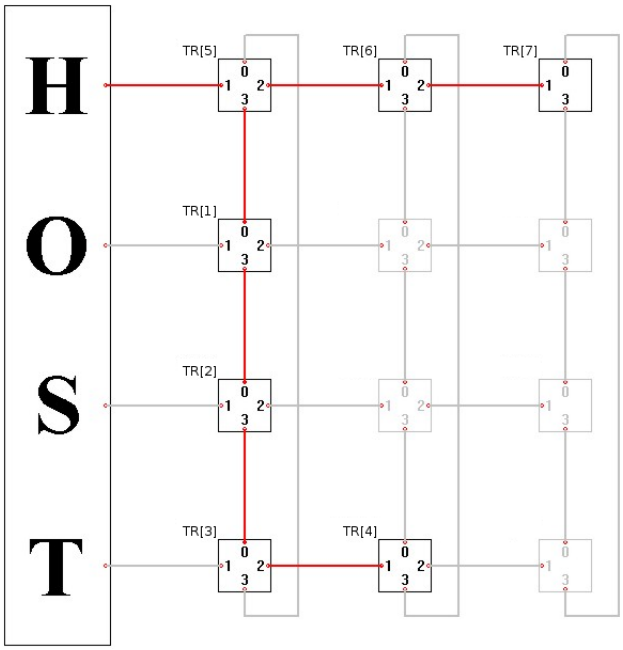
1 Opis zadania

Problem polega na porównywaniu różnych sposobów przetwarzania zadań jednorodnych w systemie transputerowym. Zadaną architekturą systemu jest łańcuch 7 transputerów z centralnie położonym źródlem danych. Zgodnie z treścią zadania założono, iż w miarę oddalania się od źródła danych, prędkość każdego kolejnego transputera maleje dwu-, cztero- i ośmiokrotnie w stosunku do jego rzeczywistej szybkości. Rozważono następujące sposoby realizacji przetwarzania:

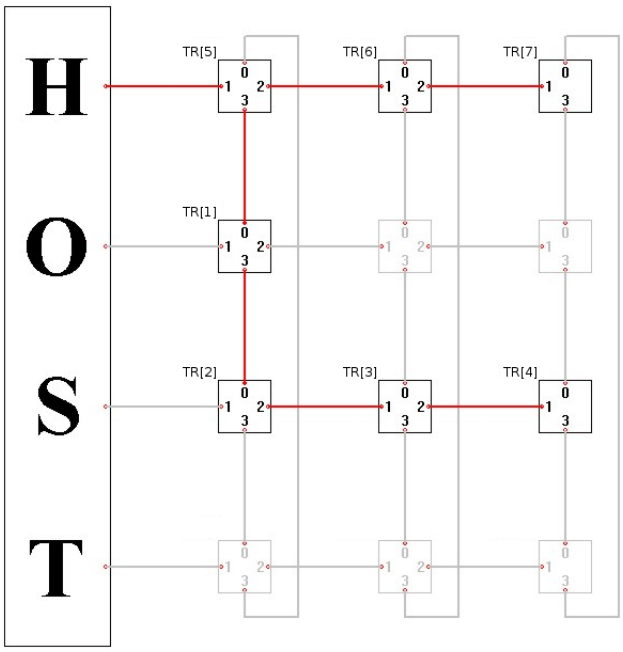
- przetwarzanie szeregowe
- przetwarzanie równoległe z jedną fazą obliczeń i jednym przesłaniem danych - 1f1p
- przetwarzanie równoległe z jedną fazą obliczeń i dwoma przeslaniami danych - 1f2p
- przetwarzanie równoległe z dwoma fazami obliczeń i jednym przesłaniem danych na fazę obliczeń - 2f1p

Ponadto dla każdego z podanych wariantów rozpatrzono dwa sposoby wyboru transputerów, na których realizowane są obliczenia:

- wybór maksymalizujący liczbę szybkich transputerów



Rysunek 1: Model maksymalizujący liczbę szybkich transputerów.



Rysunek 2: Model maksymalizujący liczbę szybkich łączy komunikacyjnych.

- wybór maksymalizujący liczbę szybkich łączy komunikacyjnych

Wszystkie powyższe warianty rozpatrzono teoretycznie, konstruując odpowiednie problemy programowania liniowego, które pozwoliły dobrać opytmalną liczbę zadań do przetworzenia dla każdego procesora. Kryterium optymalności stanowił jak najkrótszy czas przetwarzania całej puli zadań. Aby ułożyć problemy programowania liniowego, przeprowadzono pomiary czasów komunikacji, oraz przetwarzania dla odpowiednich transputerów.

Następnie zaimplementowano warianty z przetwarzaniem 1f1p oraz 1f2p dla maksymalizacji liczby szybkich łączy oraz dla maksymalizacji liczby szybkich procesorów. Dla każdego transputera przydzielono liczbę zadań wyliczoną z modelu teoretycznego. Wyniki uzyskane z testów powyższych implementacji porównano następnie z modelami teoretycznymi.

2 Wyznaczanie parametrow systemu

Należało wyznaczyć czas inicjalizacji połączenia, prędkość komunikacji oraz prędkość przetwarzania pojedynczego zadania. Uwzględniono fakt, iż czas inicjalizacji połączenia oraz prędkość przetwarzania pojedynczego zadania, zależy od prędkości transputera - transputery w pierwszej kolumnie od lewej są szybsze niż pozostałe. Natomiast czas komunikacji zależny jest od rodzaju łącza - łącza poziome są szybsze niż pionowe.

2.1 Pomiar czasów inicjalizacji połączenia, oraz prędkości komunikacji

Napisano program mierzący czasy inicjalizacji połączeń, oraz prędkości przesyłania danych. Pomiary dokonano pomiędzy:

- dwoma szybkimi transputerami połączonymi wolnym łączem
- dwoma wolnymi transputerami połączonymi szybkim łączem
- dwoma wolnymi transputerami połączonymi wolnym łączem

Dokonano odpowiedniej synchronizacji procesów, aby upewnić się iż wszystkie one są zainicjowane i gotowe do przeprowadzenia pomiarów. Następnie uruchamiano funkcję ProcTime(), aby określić moment początku komunikacji i wysyłano pakiet danych do drugiego z transputerów, ktory natychmiast odsyłał otrzymane dane do nadawcy. Nadawca uruchamiał po raz drugi funkcję ProcTime(), aby określić moment zakończenia komunikacji. Zarowno proces nadawcy jak i odbiorcy uruchamiany był z wysokim priorytetem, co gwarantowało wysoką dokładność pomiarów.

Procedurę powtarzano dla różnych wielkości pakietów z danymi. Następnie dla otrzymanych wyników obliczono regresję liniową, dzięki której otrzymano wartości czasu inicjalizacji komunikacji, oraz prędkości komunikacji. Powyższą procedurę powtarzano dla każdej z wymienionych trzech par transputerów. Oto otrzymane wyniki czasów inicjalizacji połączeń oraz przesyłania jednego pakietu danych:

Typ Transputera	SetupTime[μs]
szybki	**10,4**
wolny	**10,94**

Tabela 1: Wyniki pomiarów czasów inicjalizacji połączeń

Typ Łącza	Czas[μs]
pionowe	**4,3**
poziome	**2,25**

Tabela 2: Wyniki pomiarów czasów przesyłania danych

2.2 Pomiar prędkości przetwarzania

Napisano program mierzący prędkość przetwarzania pojedynczego zadania na transputerze szybkim oraz wolnym. Pojedyncze zadanie polegalo na obliczeniu metodą Hornera wartości następującego wielomianu: $W(x) = (((4 * x + 12) * x + 3) * x + 3) * x + 7$.

W tym celu na każdym z dwóch tranasputerów, których dotyczyły pomiary, utworzono po dwa procesy. Proces z priorytetem niskim, na którym odbywały się obliczenia, oraz proces z priorytetem wysokim, na którym odbywały się pomiary czasu. Wybór takiego sposobu pomiarów wynika z próby jak najdokładniejszego odwzorowania architektury zastosowanej przy implementacji modeli. Po odpowiedniej synchronizacji procesów, która gwarantuje ich gotowość do przeprowadzenia pomiarów, proces obliczeniowy wysyłał sygnał - liczbę całkowitą do procesu pomiarowego i zaczynał obliczenia. Proces pomiarowy po otrzymaniu pierwszego sygnału uruchamiał funkcję ProcTime() do uzyskania czasu rozpoczęcia obliczeń. Proces obliczeniowy po zakończeniu obliczeń wysyłał do procesu mierzącego czas kolejny sygnał. Proces mierzący, po odebraniu drugiego sygnalu, uruchamiał funkcję ProcTime() aby uzyskać czas zakończenia obliczeń. Różnica obu czasów dała czas trwania obliczeń powiększony o czas przesyłania komunikatu między procesami (ok. 18 μs).

Procedurę powtarzano dla różnej liczby zadań do przetworzenia przez procesor, przy czym dla każdej liczby zadań pomiar przeprowadzano dwudziestokrotnie. Następnie dla otrzymanych wyników obliczono regresję liniową, dzięki ktorej otrzymano wartości prędkości obliczeń dla transputera szybkiego oraz wolnego. Oto otrzymane wyniki prędkości przetwarzania pojedynczego zadania:

Typ Transputera	Czas[μs]
szybki	**6,6**
wolny	**6,83**

Tabela 3: Wyniki pomiarów prędkości przetwarzania pojedynczego zadania

3 Opis modeli systemów

Dla każdego wariantu realizacji obliczeń opisanego w punkcjie pierwszym, utworzono model matemetyczny. Jest nim zadanie programowania liniowego, które następnie rozwiązywano za pomocą programu lp_solve. Pozwoliło to uzyskać optymalne rozkłady puli zadań na poszczególnych procesorach, minimalizując tym samym czasy zakończenia obliczeń.

- **v** - całkowita liczba zadań do realizacji przez system
- **T** - minimalizowany całkowity czas przetwarzania v zadań
- **li** - liczba zadań dla itego procesora, i=1..7
- **ai** - czas przetwarzania pojedynczego zadania na itym transputerze, i=1..7

- **s1, s2** - czas inicjalizacji komunikacji, odpowiednio dla szybkiego i wolnego transputera
- **c1, c2** - czas przesłu jednego pakietu danych, odpowiednio wolnym i szybkim łączem

Uwaga!

- **Poniżej przedstawiono tylko równania i/lub ograniczenia dla wybranych modeli, bez inicjalizowania zmiennych i stałych problemu. Kompletne kody dla każdego problemu umieszczone zostały na dołączonej płycie CD.**
- **Przyjęte oznaczenia (indeksy według rysunków z rozdziału Opis zadania)**

3.1 Model szeregowy

W modelu tym założono iż komunikacja między transputerami przebiega szeregowo, to znaczy transputer - źródło przesyła dane najpierw do jednego końca łańcucha a następnie do drugiego. Dopiero po zakończeniu przesyłania danych, transputer zaczyna obliczenia. Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.

3.1.1 Maksymalizacja liczby szybkich procesorów

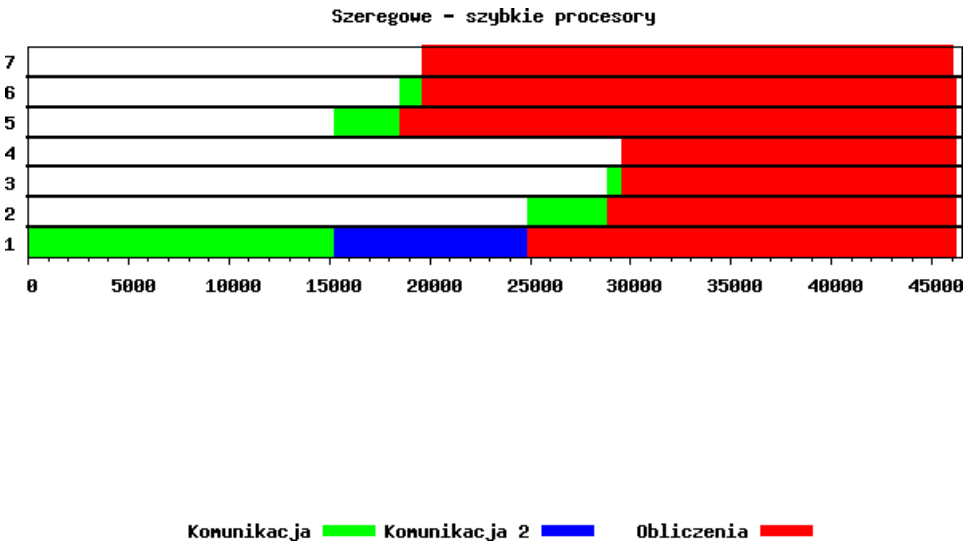
minimize obj: +T;

R1: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + l1*a1;
R2: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c1 + l2*a2;
R3: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c1 + s1+(l4)*c2 + l3*a3;
R4: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c1 + s1+(l4)*c2 + l4*a4;
R5: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + l5*a5;
R6: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + s2+(l7)*c2 + l6*a6;
R7: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + s2+(l7)*c2 + l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

Rozwiązanie problemu programowania liniowego dla tego wariantu:

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	15245,3	46129,1
1	0	24870,5	
2	24870,5	28875,6	46128
3	28875,6	29567,75	46094,15
4	29567,75	29567,75	46123,67
5	15245,3	18522,7	46123,9
6	18522,7	19618,14	46118,54
7	19618,14	19618,14	45954,62

Tabela 4: Wyniki obliczeń dla modelu szeregowego z maksymalizacją szybkich procesorów



Rysunek 3: Diagram Gantta dla modelu szeregowego z maksymalizacją szybkich procesorów

3.1.2 Maksymalizacja liczby szybkich łączy

minimize obj: +T;

R1: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + l1*a1;
R2: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + l2*a2;
R3: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + s2+(l4)*c2 + l3*a3;
R4: T >= s1+(l5+l6+l7)*c1 + s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + s2+(l4)*c2 + l4*a4;
R5: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + l5*a5;
R6: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + s2+(l7)*c2 + l6*a6;
R7: T >= s1+(l5+l6+l7)*c1 + s1+(l6+l7)*c2 + s2+(l7)*c2 + l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

Rozwiązanie problemu programowania liniowego dla tego wariantu:

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	15167,9	45887,6
1	0	25322	
2	25322	27508,15	45882,55
3	27508,15	28243,59	45864,99
4	28243,59	28243,59	45837,67
5	15167,9	18429,55	45885,55
6	18429,55	19524,99	45834,15
7	19524,99	19524,99	45861,47

Tabela 5: Wyniki obliczeń dla modelu szeregowego z maksymalizacją szybkich łączy

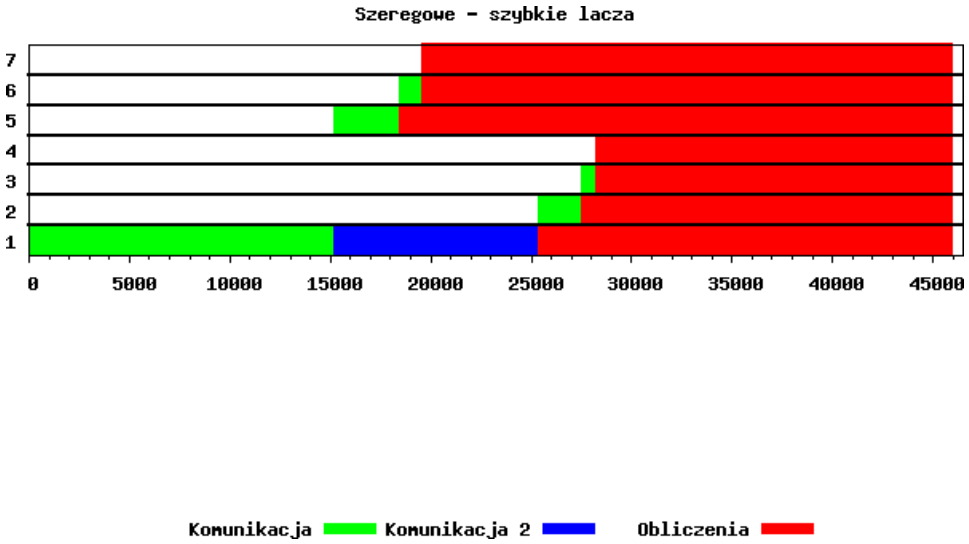
3.2 Model 1f1p

W modelu tym założono, że komunikacja na każdym transputerze przebiega równolegle z obliczeniami. Ponadto dla transputera-źródła równolegle odbywa się przesyłanie w obie strony. Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.(rysunek!!!)

3.2.1 Maksymalizacja liczby szybkich procesorów

minimize obj: +T;

R1: T >= l1*a1;
R2: T >= s1+(l2+l3+l4)*c1 +l2*a2;



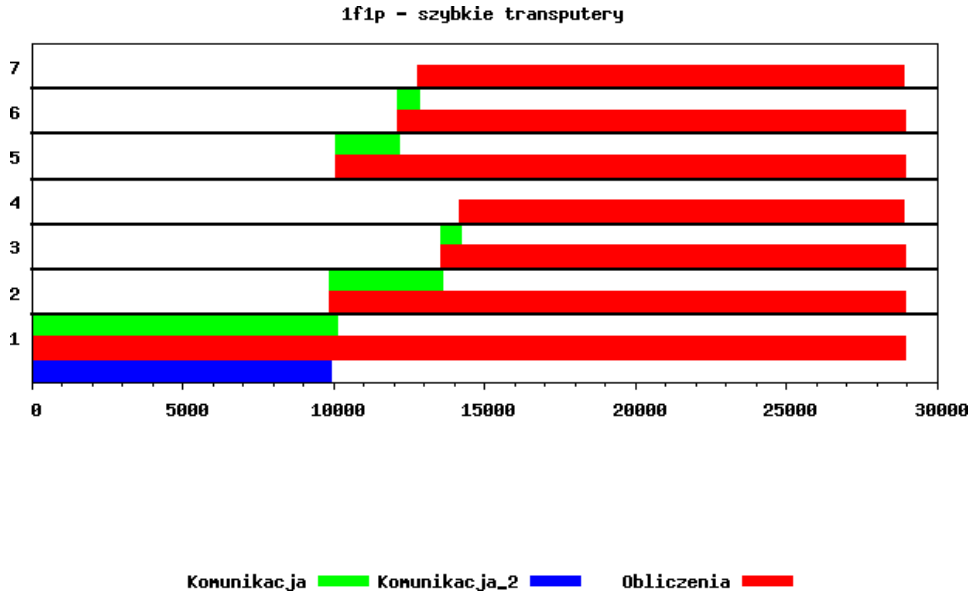
Rysunek 4: Diagram Gantta dla modelu szeregowego z maksymalizacją szybkich łączy

R3: T >= s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c1 + l3*a3;
R4: T >= s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c1 + s1+(l4)*c2 +l4*a4;
R5: T >= s1+(l5+l6+l7)*c1 + l5*a5;
R6: T >= s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + l6*a6;
R7: T >= s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + s2+(l6)*c2 + l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

Rozwiązanie problemu programowania liniowego dla tego wariantu:

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	10046,6	28875
1	0	9861,7	
2	9861,7	13527,1	28882,9
3	13527,1	14142,75	28865,5
4	14142,75	14142,75	28840,91
5	10046,6	12100	28869,8
6	12100	12772,44	28874,48
7	12772,44	12772,44	28836,6

Tabela 6: Wyniki obliczeń dla modelu 1f1p z maksymalizacją szybkich procesorów



Rysunek 5: Diagram Gantta dla modelu 1f1p z maksymalizacją szybkich procesorów

3.2.2 Maksymalizacja liczby szybkich łączy

minimize obj: +T;

R1: T >= l1*a1;
R2: T >= s1+(l2+l3+l4)*c1 + l2*a2;
R3: T >= s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + l3*a3;
R4: T >= s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + s2+(l4)*c2 +l4*a4;
R5: T >= s1+(l5+l6+l7)*c1 + l5*a5;
R6: T >= s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + l6*a6;
R7: T >= s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + s2+(l6)*c2 + l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

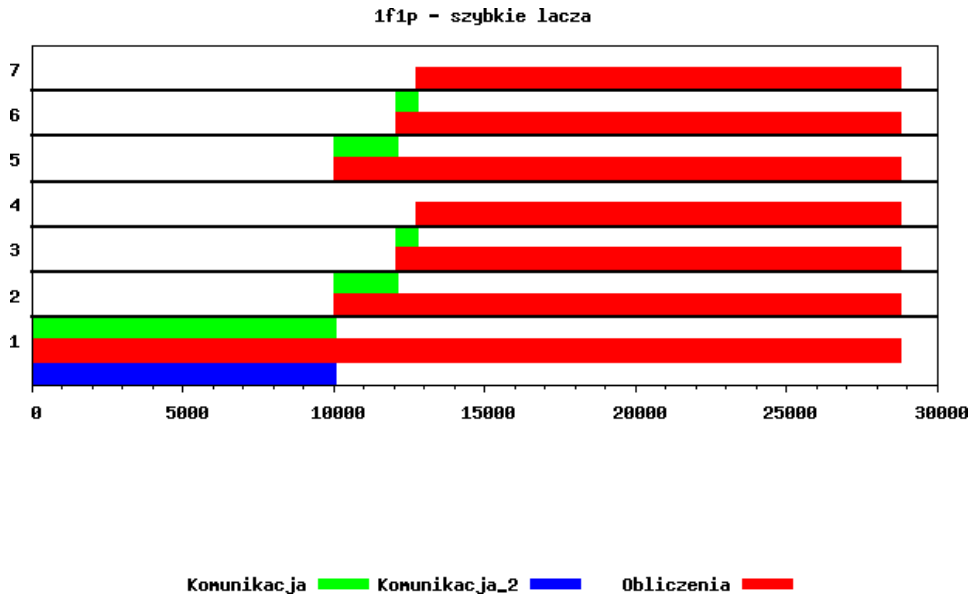
Rozwiązanie problemu programowania liniowego dla tego wariantu:

3.3 Model 1f2p

W modelu tym założono, że komunikacja odbywa się w dwóch etapach. W pierwszym etapie komunikacji, transputer odbiorca otrzymuje dane potrzebne mu do całkowitej realizacji swojej części obliczeń. W drugim etapie komunikacji otrzymuje natomiast dane do przesłania kolejnym transputerom w łańcuchu. Proces obliczeń i komunikacji został zrównoleglony w maksymalnym możliwym stopniu. Oto problemy liniowe dla omawianego modelu,

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	9999,3	28736,4
1	0	9999,3	
2	9999,3	12043,7	28730,1
3	12043,7	12713,89	28736,22
4	12713,89	12713,89	28723,41
5	9999,3	12043,7	28730,1
6	12043,7	12713,89	28736,22
7	12713,89	12713,89	28723,41

Tabela 7: Wyniki obliczeń dla modelu 1f1p z maksymalizacją szybkich łączy



Rysunek 6: Diagram Gantta dla modelu 1f1p z maksymalizacją szybkich łączy

przy przyjętych oznaczeniach.

3.3.1 Maksymalizacja liczby szybkich procesorów

minimize obj: +T;

R1: T >= l1*a1;
R2: T >= s1+(l2)*c1 + l2*a2;
R3: T >= s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c1 + l3*a3;
R4: T >= s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c1 + s1+(l4)*c1 +
s1+(l4)*c2 + l4*a4;
R5: T >= s1+(l5)*c1 + l5*a5;
R6: T >= s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + l6*a6;
R7: T >= s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + s1*(l7)*c2 +
s2*(l7)*c2 +l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

Rozwiązanie problemu programowania liniowego dla tego wariantu:

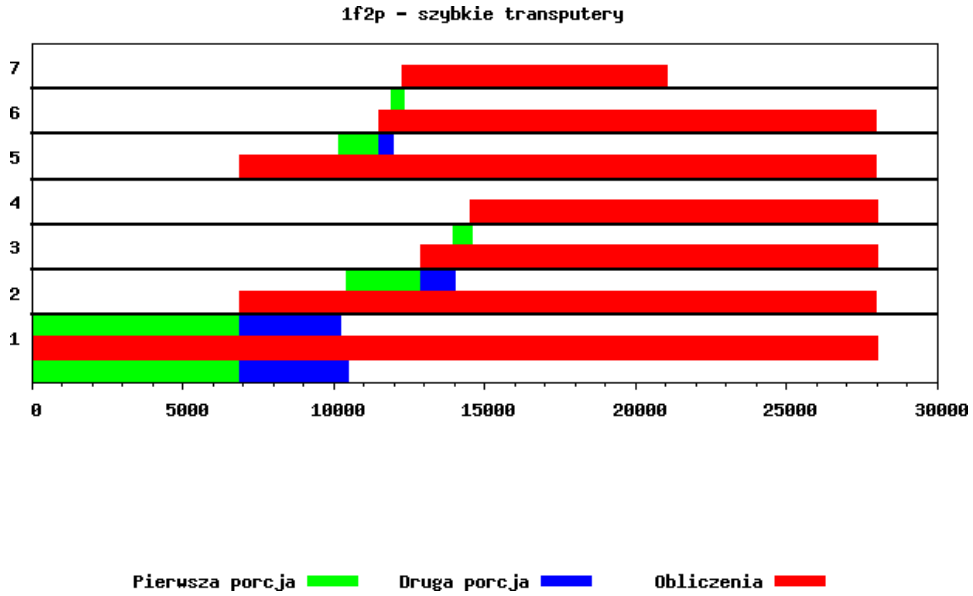
Transputer	Odebranie 1. paczki	Odbieranie 2. paczki	Wysyłanie 1. paczki	Wysyłanie 2. paczki	Koniec obliczeń
1			6868,9	10147,3	27937,8
1			6868,9	10392,4	
2	6868,9	10392,4	12858,1	13926,3	27922,9
3	12858,1	13926,3	14490,2		27932,5
4	14490,2				27931,64
5	6868,9	10147,3	11507,7	11878,1	27922,9
6	11507,7	11878,1	12249,04		27899,7
7	12249,04				20991,44

Tabela 8: Wyniki pomiarów dla modelu 1f2p z maksymalizacją szybkich procesorów

3.3.2 Maksymalizacja liczby szybkich łączy

minimize obj: +T;

R1: T >= 4 l1*a1;
R2: T >= s1+(l2)*c1 + l2*a2;
R3: T >= s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c2 + l3*a3;
R4: T >= s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c2 + s1+(l4)*c2 +
s2+(l4)*c2 + l4*a4;
R5: T >= s1+(l5)*c1 + l5*a5;



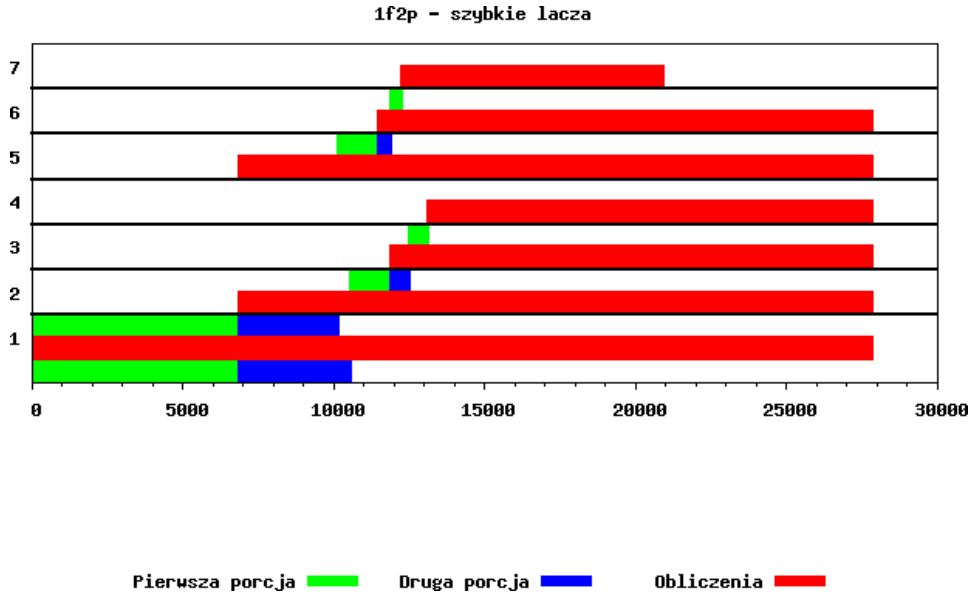
Rysunek 7: Diagram Gantta dla modelu 1f2p z maksymalizacją szybkich procesorów

R6: T >= s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + l6*a6;
R7: T >= s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + s1*(l7)*c2 + s2*(l7)*c2 + l7*a7;
R8: v = l1+l2+l3+l4+l5+l6+l7;

Rozwiązanie problemu programowania liniowego dla tego wariantu:

Transputer	Odebranie 1. paczki	Odbieranie 2. paczki	Wysyłanie 1. paczki	Wysyłanie 2. paczki	Koniec obliczeń
1			6838,8	10104,3	27812,4
1			6838,8	10517,1	
2	6838,8	10517,1	11841,5	12457,15	27800,4
3	11841,5	12457,15	13073,34		27796,38
4	13073,34				27771,5
5	6838,8	10104,3	11460,2	11828,35	27800,4
6	11460,2	11828,35	12197,04		27797,56
7	12197,04				20884,8

Tabela 9: Wyniki pomiarów dla modelu 1f2p z maksymalizacją szybkich łączy



Rysunek 8: Diagram Gantta dla modelu 1f2p z maksymalizacją szybkich łączy

3.4 Model 2f1p

W modelu tym komunikację między transputerami rozbito na dwie fazy. W pierwszej, każdy z transputerów otrzymuje tylko część potrzebnych mu do obliczeń danych i może wykonać pierwszę fazę obliczeń. Natomiast w drugiej fazie komunikacji każdy z transputerów otrzymiuje drugą część potrzebnych mu do obliczeń danych. Przesyłanie i obliczenia realizowane sę w maksymalnie zrównoleglony sposob. Do opisu powyższego modelu niezbędne bylo wprowadzenie nowych parametrów opisujących system:

- **ob**ij - Początek obliczeń na itym transputerze w jotej fazie, i=1..7, j=1..2, wyjątek dla pierwszego transputera - ob1
- **oe**ij - Koniec obliczeń na itym transputerze w jotej fazie, i=1..7, j=1..2, wyjątek dla pierwszego transputera - oe1
- **kb**ij - Początek komunikacji na itym transputerze w jotej fazie, i=1..6, j=1..2, wyjątek dla pierwszego transputera - kbijl, gdzie l oznacza: g - komunikację w stronę węzła 7, d - w stronę węzła 4
- **ke**ij - Koniec komunikacji na itym transputerze w jotej fazie, i=1..6, j=1..2, wyjątek dla pierwszego transputera - keijl, gdzie l oznacza: g - komunikację w stronę węzła 7, d - w stronę węzła 4

Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.

3.4.1 Maksymalizacja liczby szybkich procesorów

minimize obj: T;

r1: T >= oe1;
r2: T >= oe22;
r3: T >= oe32;
r4: T >= oe42;
r5: T >= oe52;
r6: T >= oe62;
r7: T >= oe72;

r8: kb21 >= ke11d;
r9: kb22 >= ke12d;
r10: kb31 >= ke21;
r11: kb32 >= ke22;
r12: kb41 >= ke31;
r13: kb42 >= ke32;
r14: kb51 >= ke11g;
r15: kb52 >= ke12g;
r16: kb61 >= ke51;
r17: kb62 >= ke52;

r18: v = l1 + l21 + l22 + l31 + l32 + l41 + l42 + l51 + l52 + l61 + l62 + l71 + l72;

r19: ke11g = kb11g + s1+(l51+l61+l71)*c1;
r20: ke12g = kb12g + s1+(l52+l62+l72)*c1;
r21: ke11d = kb11d + s1+(l21+l31+l41)*c1;
r22: ke12d = kb12d + s1+(l22+l32+l42)*c1;
r23: ke21 = kb21 + s1+(l31+l41)*c1;
r24: ke22 = kb22 + s1+(l32+l42)*c1;
r25: ke31 = kb31 + s1+(l41)*c2;
r26: ke32 = kb32 + s1+(l42)*c2;
r27: ke51 = kb51 + s1+(l61+l71)*c2;
r28: ke52 = kb52 + s1+(l62+l72)*c2;
r29: ke61 = kb61 + s2+(l71)*c2;
r30: ke62 = kb62 + s2+(l72)*c2;

r31: kb12g >= ke11g;
r32: kb12d >= ke11d;
r33: kb22 >= ke21;
r34: kb32 >= ke31;
r35: kb42 >= ke41;

r36: kb52 >= ke51;
r37: kb62 >= ke61;

r38: ob22 >= oe21;
r39: ob32 >= oe31;
r40: ob42 >= oe41;
r41: ob52 >= oe51;
r42: ob62 >= oe61;
r43: ob72 >= oe71;

r44: ob21 >= ke11d;
r45: ob22 >= ke12d;
r46: ob31 >= ke21;
r47: ob32 >= ke22;
r48: ob41 >= ke31;
r49: ob42 >= ke32;
r50: ob51 >= ke11g;
r51: ob52 >= ke12g;
r52: ob61 >= ke51;
r53: ob62 >= ke52;
r54: ob71 >= ke61;
r55: ob72 >= ke62;

r56: oe1 = ob1 + a1*l1;
r57: oe21 = ob21 + a2*l21;
r58: oe22 = ob22 + a2*l22;
r59: oe31 = ob31 + a3*l31;
r60: oe32 = ob32 + a3*l32;
r61: oe41 = ob41 + a4*l41;
r62: oe42 = ob42 + a4*l42;
r63: oe51 = ob51 + a5*l51;
r64: oe52 = ob52 + a5*l52;
r65: oe61 = ob61 + a6*l61;
r66: oe62 = ob62 + a6*l62;
r67: oe71 = ob71 + a7*l71;
r68: oe72 = ob72 + a7*l72;

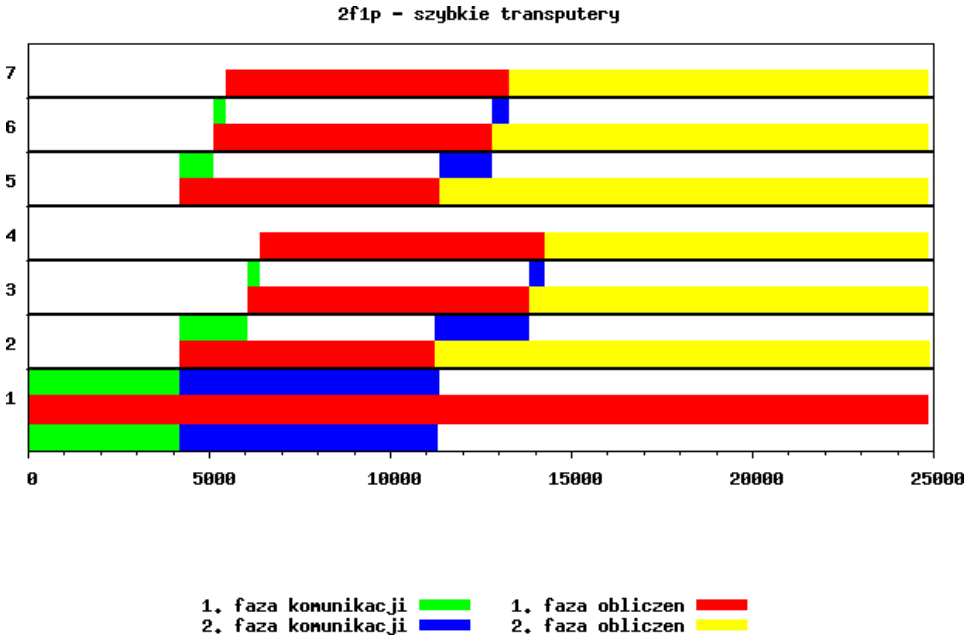
Rozwiązanie problemu programowania liniowego dla tego wariantu:

3.4.2 Maksymalizacja liczby szybkich łączy

minimize obj: T;

Trans.	start 1. obl.	start 1. kom.	koniec 1. kom.	koniec 1. obl.	start 2. obl.	start 2. kom.	koniec 2. kom.	Koniec obl.
1	0	0	4172.8	24802.8		4172.8	11351.3	24802.8
1	-1	0	4190			4190	11230.9	
2	4190	4190	6083.8	11238.8	11238.8	11230.9	13851.4	24808.4
3	6083.8	6083.8	6418.2	13845.4	13851.4	13851.4	14293.8	24807.4
4	6418.2			14286.4	14293.8			24784.7
5	4172.8	4172.8	5137.2	11353.6	11353.6	11351.3	12819.7	24804.4
6	5137.2	5137.2	5469.89	12814.1	12819.7	12819.7	13303.1	24785.9
7	5469.89			13283.4	13303.1			24777.5

Tabela 10: Wyniki pomiarów dla modelu 2f1p z maksymalizacją szybkich procesorów



Rysunek 9: Diagram Gantta dla modelu 2f1p z maksymalizacją szybkich procesorów

r1: T >= oe1;
r2: T >= oe22;
r3: T >= oe32;
r4: T >= oe42;
r5: T >= oe52;
r6: T >= oe62;
r7: T >= oe72;

r8: kb21 >= ke11d;
r9: kb22 >= ke12d;
r10: kb31 >= ke21;
r11: kb32 >= ke22;
r12: kb41 >= ke31;
r13: kb42 >= ke32;
r14: kb51 >= ke11g;
r15: kb52 >= ke12g;
r16: kb61 >= ke51;
r17: kb62 >= ke52;

r18: v = l1 + l21 + l22 + l31 + l32 + l41 + l42 + l51 + l52 + l61 + l62 + l71 + l72;

r19: ke11g = kb11g + s1+(l51+l61+l71)*c1;
r20: ke12g = kb12g + s1+(l52+l62+l72)*c1;
r21: ke11d = kb11d + s1+(l21+l31+l41)*c1;
r22: ke12d = kb12d + s1+(l22+l32+l42)*c1;
r23: ke21 = kb21 + s1+(l31+l41)*c2;
r24: ke22 = kb22 + s1+(l32+l42)*c2;
r25: ke31 = kb31 + s2+(l41)*c2;
r26: ke32 = kb32 + s2+(l42)*c2;
r27: ke51 = kb51 + s1+(l61+l71)*c2;
r28: ke52 = kb52 + s1+(l62+l72)*c2;
r29: ke61 = kb61 + s2+(l71)*c2;
r30: ke62 = kb62 + s2+(l72)*c2;

r31: kb12g >= ke11g;
r32: kb12d >= ke11d;
r33: kb22 >= ke21;
r34: kb32 >= ke31;
r35: kb42 >= ke41;
r36: kb52 >= ke51;
r37: kb62 >= ke61;

r38: ob22 >= oe21;

r39: ob32 >= oe31;
r40: ob42 >= oe41;
r41: ob52 >= oe51;
r42: ob62 >= oe61;
r43: ob72 >= oe71;

r44: ob21 >= ke11d;
r45: ob22 >= ke12d;
r46: ob31 >= ke21;
r47: ob32 >= ke22;
r48: ob41 >= ke31;
r49: ob42 >= ke32;
r50: ob51 >= ke11g;
r51: ob52 >= ke12g;
r52: ob61 >= ke51;
r53: ob62 >= ke52;
r54: ob71 >= ke61;
r55: ob72 >= ke62;

r56: oe1 = ob1 + a1*l1;
r57: oe21 = ob21 + a2*l21;
r58: oe22 = ob22 + a2*l22;
r59: oe31 = ob31 + a3*l31;
r60: oe32 = ob32 + a3*l32;
r61: oe41 = ob41 + a4*l41;
r62: oe42 = ob42 + a4*l42;
r63: oe51 = ob51 + a5*l51;
r64: oe52 = ob52 + a5*l52;
r65: oe61 = ob61 + a6*l61;
r66: oe62 = ob62 + a6*l62;
r67: oe71 = ob71 + a7*l71;
r68: oe72 = ob72 + a7*l72;

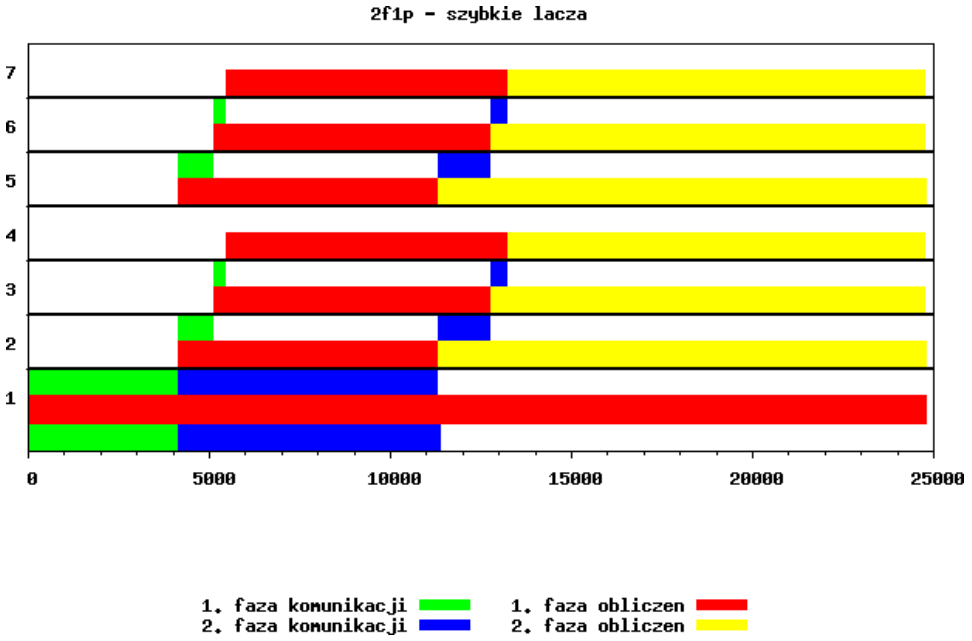
Rozwiązanie problemu programowania liniowego dla tego wariantu:

4 Opis implementacji modeli

Zaimplementowano programy realizujące modele 1f1p oraz 1f2p w wersjach uwzględniających architekturę zarówno dla maksymalizacji liczby szybkich procesorów jak i maksymalizacji liczby szybkich łączy - patrz opis zadania. Dla każdego transputera użytego w danej implementacji stworzono dwa procesy: obliczeniowy i pomiarowo-komunikacyjny. Proces obliczeniowy uruchamiano z priorytetem niskim. Natomiast proces pomiarowo-komunikacyjny z

Trans.	start 1. obl.	start 1. kom.	koniec 1. kom.	koniec 1. obl.	start 2. obl.	start 2. kom.	koniec 2. kom.	Koniec obl.
1	0	0	4159.9	24723.6		4159.9	11316.9	24723.6
1		0	4159.9			4159.9	11316.9	
2	4159.9	4159.9	5122.05	11314.3	11316.9	11316.9	12780.8	24728.1
3	5122.05	5122.05	5454.74	12771.7	12780.8	12780.8	13262	24719.6
4	5454.74			13268.3	13268.3			24688
5	4159.9	4159.9	5122.05	11314.3	11316.9	11316.9	12780.8	24728.1
6	5122.05	5122.05	5454.74	12771.6	12780.8	12780.8	13262	24719.6
7	5454.74			13268.3	13268.3			24688

Tabela 11: Wyniki pomiarów dla modelu 2f1p z maksymalizacją szybkich łączy



Rysunek 10: Diagram Gantta dla modelu 2f1p z maksymalizacją szybkich łączy

priorytetem wysokim.

Pierwszą fazą działania programów była synchronizacja porcesów. Po dokonaniu synchronizacji na każdym z transputerów, począwszy od najbardziej zewnętrznych, odczytywano aktualną wartość zegara, następnie przesyłano komunikat w kierunku środka łańcucha. Każdy następny transputer odbierając ten komunikat pobierał aktualną wartość TimeProc(). Pomierzone w ten sposób czasy uznano za początek przetwarzania. Jako, że najpierw mierzono czas na najbardziej zewnętrznych transputerach, należy mieć na uwadze, że uzyskane na nich względne punkty pomiarowe (liczone od rozpoczęcia przetwarzania) są nieco większe od faktycznych. Różnica ta jest niewielka w stosunku do długości operacji na danych.

4.1 Model 1f1p

W modelu tym mierzono trzy momenty czasowe dla każdego z transputerów: bezpośrednio przed i po wyslaniu danych do kolejnego transputera oraz po otrzymaniu informacji o zakończeniu obliczeń. Do rozsyłania danych z centralnego transputera użyto dwóch procesów o wysokim priorytecie realizujących zadanie równocześnie. Obliczenia wykonywano na procesach o niskim priorytecie, co gwarantowało, że będą one uruchamiane w czasie wysyłania danych, nie blokując jednocześnie inicjalizacji tego wysyłania. Gdy proces komunikacyjny otrzymał dane, wysyłał informację synchronizującą do procesu obliczeniowego, informując o możliwości rozpoczęcia obliczeń. Następnie proces o wysokim priorytecie inicjował przesyłanie danych na kolejny transputer. Po zakończeniu przetwarzania proces o niskim priorytecie wysyłał komunikat do procesu o wysokim priorytecie, na którym mierzony był czas zakończenia obliczeń. Dopiero po ustaleniu wszystkich potrzebnych czasów dane przesyłane były do procesu wejścia-wyjścia.

4.2 Model 2f1p

W tym przypadku wyznaczano czasy:

- zakończenia odbierania pierwszej porcji danych (początku obliczeń)
- zakończenia odbierania drugiej porcji danych (początku przesyłania danych na kolejny transputer)
- zakończenia przesyłania pierwszej porcji danych
- zakończenia przesyłania drugiej porcji danych
- zakończenia obliczeń na transputerze

Struktura i zadania procesów pozostały niezmienione w stosunku do implementacj modelu 1f1p. Po otrzymaniu pierwszej porcji danych proces komunikacyjny wysyłał komunikat do procesu obliczeniowego o możliwości rozpoczęcia obliczeń, po czym odbierał drugą porcję danych, przekazując ją w

dwóch porcjach do następnego transputera. Pomiar czasu zakończenia obliczeń odbywał się w sposób opisany dla modelu 1f1p.

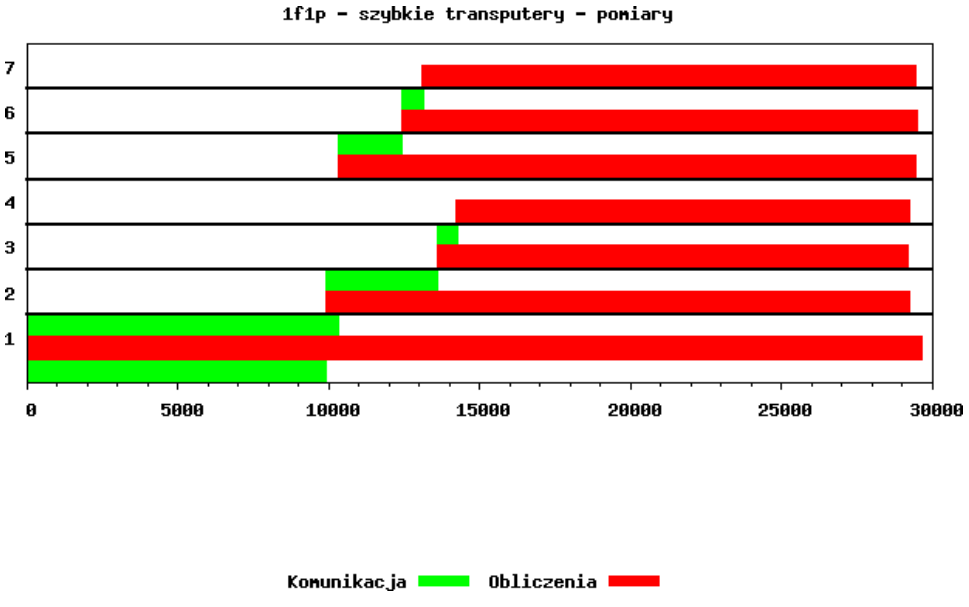
5 Wyniki testów implementacji dla wybranych modeli

5.1 Model 1f1p

5.1.1 Maksymalizacja liczby szybkich procesorów

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	10282	29568
1	0	9865	
2	9887	13558	29189
3	13582	14202	29103
4	14224		29157
5	10314	12370	29384
6	12392	13069	29448
7	13091		29410

Tabela 12: Wyniki pomiarów dla modelu 1f1p z maksymalizacją szybkich transputerów

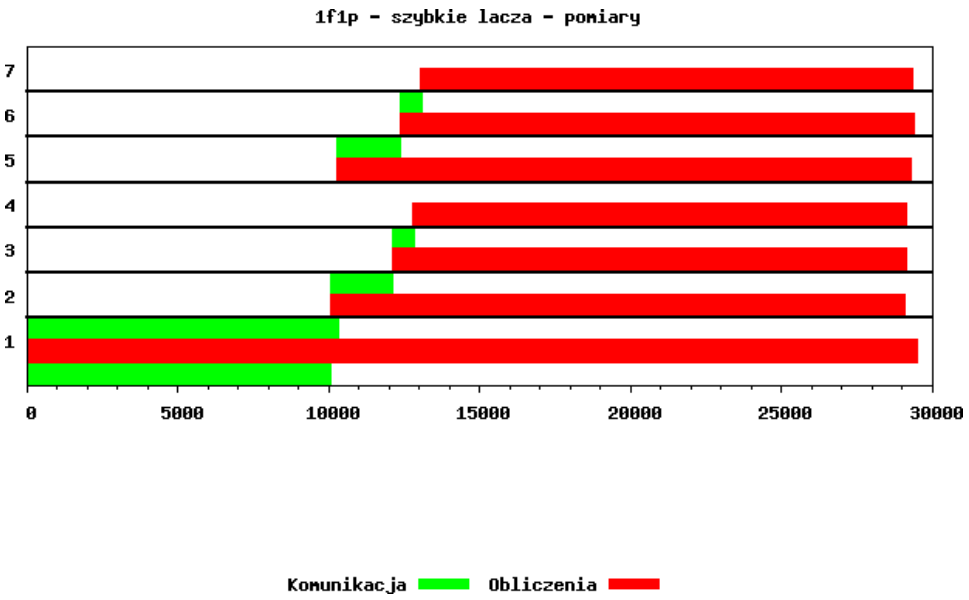


Rysunek 11: Diagram Gantta dla pomiarów 1f1p z maksymalizacją szybkich transputerów

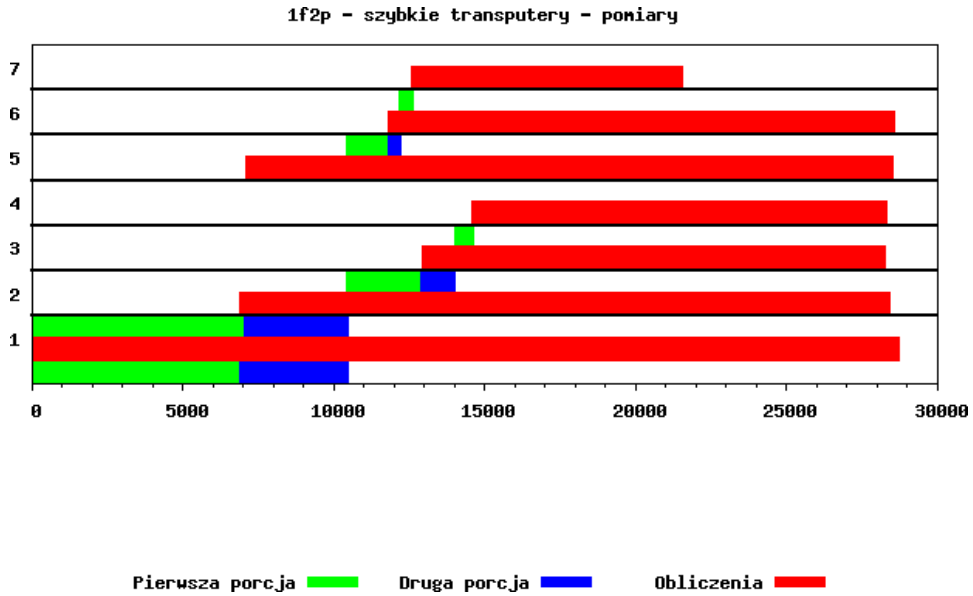
5.1.2 Maksymalizacja liczby szybkich łączy

transputer	poczatek przesyłania(obliczeń)	koniec przesylania	koniec obliczeń
1	0	10234	29434
1	0	10003	
2	10026	12075	29002
3	12098	12773	29071
4	12794		29056
5	10265	12312	29243
6	12334	13010	29308
7	13031		29294

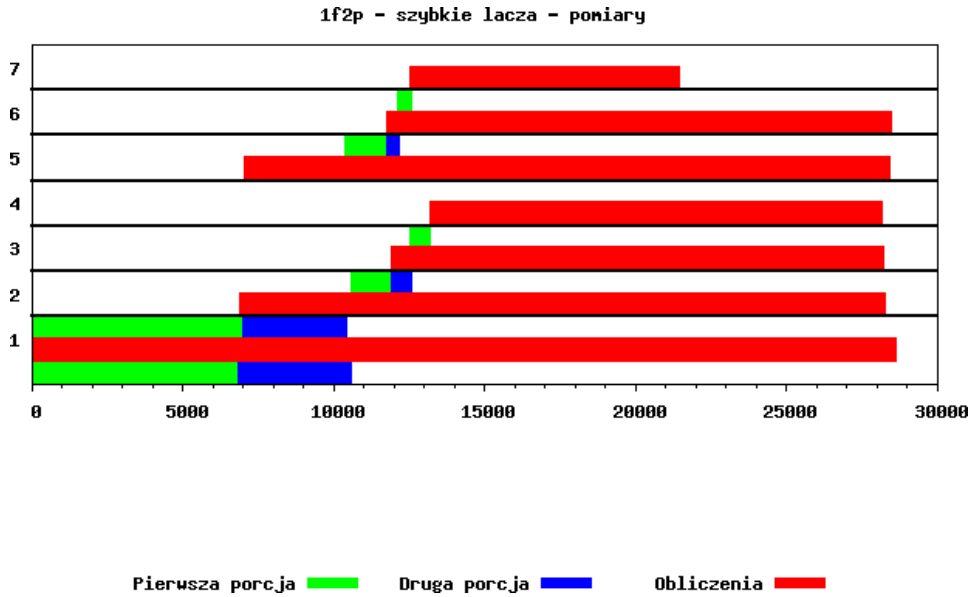
Tabela 13: Wyniki pomiarów dla modelu 1f1p z maksymalizacją szybkich łączy



Rysunek 12: Diagram Gantta dla pomiarów 1f1p z maksymalizacją szybkich łączy



Rysunek 13: Diagram Gantta dla pomiarów 1f2p z maksymalizacją szybkich transputerów



Rysunek 14: Diagram Gantta dla pomiarów 1f2p z maksymalizacją szybkich łączy

Transputer	Odebranie 1. paczki	Odbieranie 2. paczki	Wysyłanie 1. paczki	Wysyłanie 2. paczki	Koniec obliczeń
1	0	0	7031	10395	28673
1	0	0	6873	10405	10405
2	6896	10421	12890	13969	28350
3	12913	13985	14553	14553	28207
4	14575	14575	14575	14575	28236
5	7061	10418	11781	12163	28468
6	11804	12178	12553	12553	28490
7	12576	12576	12576	12576	21478

Tabela 14: Wyniki pomiarów dla modelu 1f2p z maksymalizacją szybkich transputerów

5.2 Model 1f2p

5.2.1 Maksymalizacja liczby szybkich procesorów

5.2.2 Maksymalizacja liczby szybkich łączy

6 Omówienie wyników i wnioski

Porównano ze sobą wyniki odpowiadających sobie problemów liniowych z wynikami przeprowadzonych testów na systemie transputerowym. Okazało się, że różnice całkowitego czasu przetwarzania między modelem a wynikami testów nie przekraczały 3%. Różnice te mogą być spowodowane nieuwzględnieniem w modelach teoretycznych czasów przełączania kontekstu pomiędzy wątkami na transputerze. Wpływ na rozbieżności mogł również mieć fakt oparcia się na modelu liniowym, który korzysta z parametrów systemu wyznaczonych przy pomocy regresji liniowej - a więc w sposób przybliżony.

Zbadano, który z rozważonych modeli dał najkrotszy czas realizacji zadania. Okazało się, iż zdecydowanie najgorszym modelem jest model szeregowy, co wydaje się być oczywiste, ponieważ ani komunikacja ani obliczenia nie są w nim zrównoleglone. Następny w kolejności model to 1f1p, który wykonał zadanie w czasie zdecydowanie krótszym niż model szeregowy. Dalszą, lecz niewielką, poprawę czasu wykonania zadania przyniosło zastosowanie modelu 1f2p. Jednak niewielki zysk, okupiony większym skomplikowaniem pociągającym za sobą trudniejszą implementację systemu, wydaje się być nieuzasadniony. Najlepszy model to 2f1p. Tutaj zysk w porównaniu z modelem 1f1p jest znaczny i rekompensuje on większą złożonośc systemu. Wyniki uzyskane z modeli liniowych zostały potwierdzone przez wyniki testów wybranych modeli na systemie transputerowym. W tym przypadku rownież model 1f2p był nieco lepszy od modelu 1f1p.

Kolejne porównanie wyników dotyczyło podejść realizujących taki sam tryb przetwarzania - ale z architekturą dobraną pod kątem maksymalizacji

liczby szybkich łącz lub maksymalizacji szybkich procesorów. We wszystkich przypadkach, zarówno dla modeli teoretycznych jak i dla testów na systemie transputerowym, nieco szybciej obliczenia wykonywane były dla architektury wykorzystującej maksymalną liczbę szybkich łącz. Różnice te jednak były minimalne, ponieważ obie architektury różniły się w niewielkim stopniu. Architektura maksymalizująca liczbę szybkich procesorow zawiera 4 szybkie procesory i 3 szybkie łącza, natomiast architektura maksymalizująca liczbę szybkich łącz zawiera 3 szybkie procesory i 4 szybkie łącza. Ponadto różnice występują w jednym z końców łańcucha, a więc w strefie systemu, do której dociera stosunkowo mało danych do obliczeń. Mimo to można wnioskować iż priorytetem w konstrukcji architektury systemu powinno być zapewnienie maksymalnej liczby szybkich łączy.

Transputer	Odebranie 1. paczki	Odbieranie 2. paczki	Wysyłanie 1. paczki	Wysyłanie 2. paczki	Koniec obliczeń
1	0	0	7000	10351	28545
1	0	0	6842	10529	10529
2	6866	10546	11873	12499	28202
3	11895	12514	13134	13134	28153
4	13158	13158	13158	13158	28091
5	7031	10375	11733	12113	28348
6	11755	12127	12500	12500	28388
7	12523	12523	12523	12523	21371

Tabela 15: Wyniki pomiarów dla modelu 1f2p z maksymalizacją szybkich łączy