

# Programowanie Równoległe: Zadanie hI

3 kwietnia 2006 roku

## 1 Treść zadania

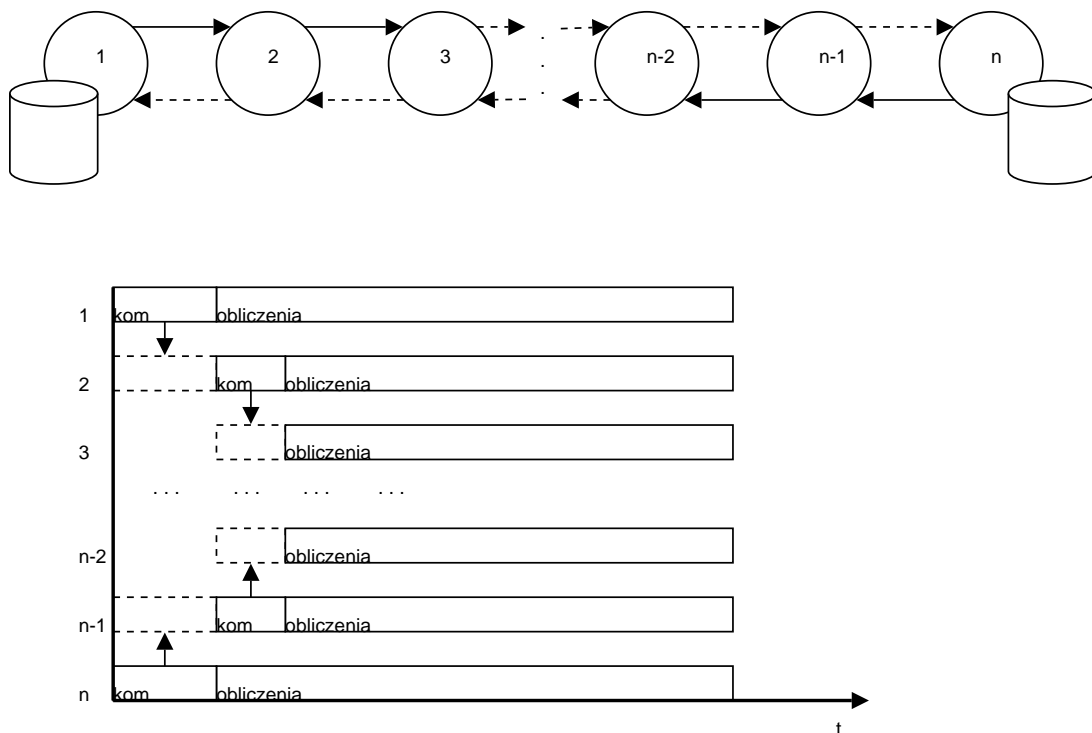
**h** Łańcuch w dwóch kierunkach, dane w momencie rozpoczęcia obliczeń znajdują się w pierwszym oraz ostatnim węźle  
**I** Sekwencyjnie przetwarzanie i komunikacja (komunikacja jednorazowa)

## 2 Proponowane rozwiązania

Na postawione wyżej zadanie zespół znalazł dwa rozwiązania. Dokładniej mówiąc jedno rozwiązanie jest szczególnym przypadkiem drugiego, jednak jest dosyć intuicyjne i powstało pierwsze.

### 2.1 Podział łańcucha

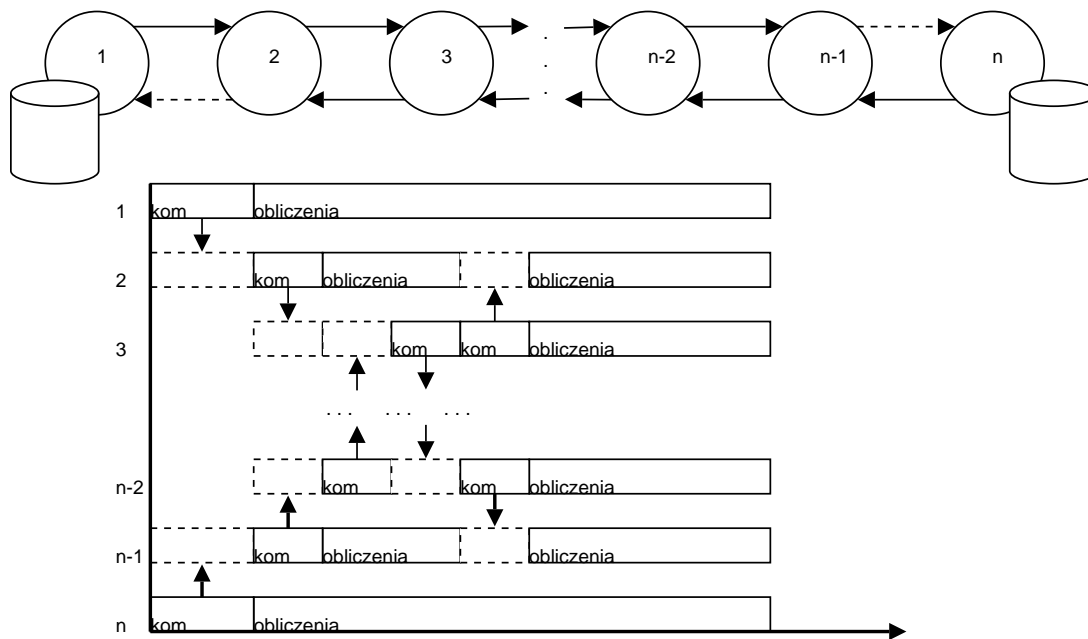
Pomysł polega na podzieleniu łańcucha. Dane są propagowane ze skrajnych procesorów równoległe do kolejnych jednostek aż do miejsca podziału. Miejsce podziału, a więc punkt w łańcuchu gdzie nie wystąpi komunikacja zależy od szybkości łączy i jednostek obliczeniowych. W przykładzie poniżej założono, że sieć jest symetryczna (tj. szybkości łączy i łączy są stałe), więc podział wypada na środku łańcucha.



Rysunek 1: Podział łańcucha

## 2.2 Współdzielenie łańcucha

Analizując powyższy graf, uznaliśmy że tak symetryczny podział może prowadzić do nie pełnego wykorzystanie zasobów, stąd propozycja bardziej złożona. Procesory od drugiego do przedostatniego są współdzielone, tzn. mogą otrzymać dane od każdej ze skrajnych jednostek. Lewy skrajny procesor wysyła dane do dowolnej liczby procesorów w prawo (poza ostatnim, czyli prawym skrajnym procesorem). Prawy postępuje analogicznie w drugą stronę. Dalej rozwijany będzie właśnie ten pomysł.

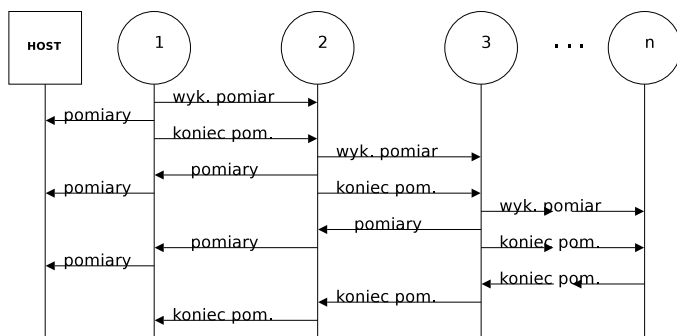


Rysunek 2: Współdzielenie łańcucha

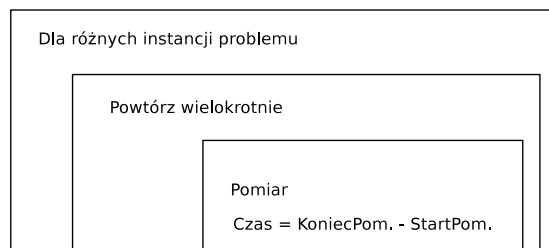
## 3 Pomiary

### 3.1 Opis metody pomiarowej

Pomiary dotyczyły dwóch parametrów: prędkości przesyłania danych między transputerami, oraz prędkości przetwarzania. Aby pobrać wyniki pomiarów z transputerów nie połączonych bezpośrednio do hosta zastosowano połączenie łańcuchowe. W pomiarach prędkości połączeń, w kolejnych ogniwach łańcucha transputer wysyła do następnego urządzenia z góry określoną liczbę danych i mierzy ile czasu mu to zajęło. Następnie odsyła poprzedniemu wyniki pomiarów. Przykładowy scenariusz komunikacji przedstawia rys. 3. Podsumowując, nasza metoda nie wymaga synchronizacji zegarów (dostępnych poprzez `ProcTime()`) na transputerach, a jedynie by były one taktowane z tą samą częstotliwością.



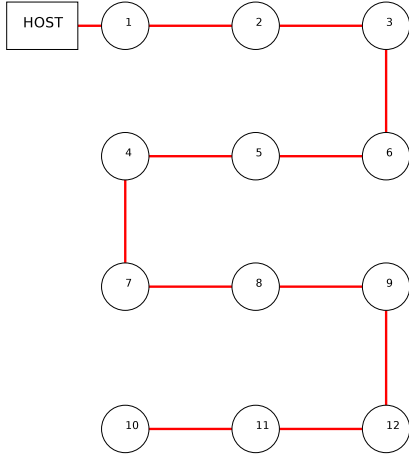
Rysunek 3: Scenariusz komunikacji podczas pomiarów



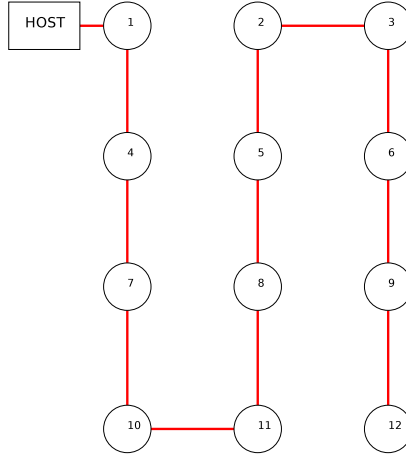
Rysunek 4: Przebieg pomiaru

Badanie prędkości polegało na mierzeniu czasu wykonania funkcji `ChanOut()` dla zbiorów danych o wielkościach 256, 512, ..., 4096 bajtów.

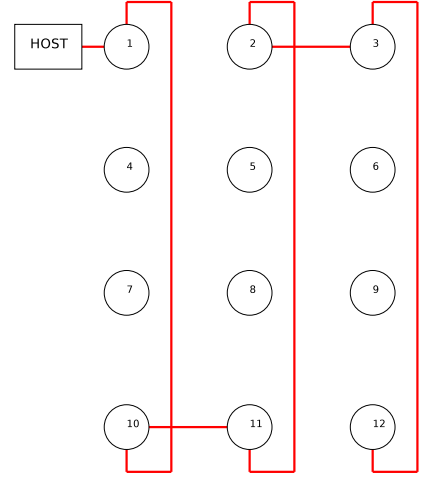
Aby zmierzyć prędkości przesyłu na wszystkich łączach, pomiary wykonywano na trzech różnie ułożonych na hiper-kostce łańcuchach przedstawionych na rys. 5, 6 oraz 7.



Rysunek 5: Test 1



Rysunek 6: Test 2



Rysunek 7: Test 3

Pomiary szybkości przetwarzania poszczególnych transputerów zostały przeprowadzone w podobny sposób - również wykorzystując mechanizm łańcucha. Ponieważ na etapie pomiarów nie było jeszcze dokładnie pewne, co będzie przetwarzać łańcuch, mierzone były prędkości obliczania sumy wielu czynników, oraz w drugim kroku wielomianu o 256, 512, ..., 4096 współczynników..

W obu przypadkach (badanie łącz i transputerów), pomiary były wykonywane wielokrotnie na danej jednostce, by wykluczyć przypadek, że urządzenie dopiero po pewnym czasie nabiera większej wydajności. Ponadto cały zestaw testów został przeprowadzony kilkakrotnie. Schemat testu przedstawia rysunek 4.

### 3.2 Wyniki pomiarów

Na podstawie pomiarów, dla każdego połączenia i dla każdego transputera, za pomocą regresji liniowej, wyznaczono równania badanych parametrów. Szukane parametry to startup time  $S_{i,j}$ , prędkość transmisji  $C_{i,j}$  na łączu między  $i$  a  $j$  (zakładamy jednakowość łącz  $i - j$  i  $j - i$ ), tak by można rozwiązać równanie:

$$t_{i,j} = S_{i,j} + \alpha_{i,j} C_{i,j} \quad (1)$$

gdzie  $t_{i,j}$  to całkowity czas przesłania porcji  $\alpha_{i,j}$  danych. Natomiast współczynniki wyznaczane dla prędkości przetwarzania  $A_i$  i-tego transputera to  $a_i$ , oraz  $b_i$ , tak by spełniały:

$$A_i = \alpha_i a_i + b_i \quad (2)$$

Tabela 1 zawiera wyznaczone z pomiarów współczynniki. Lepszy pogląd na różnice prędkości połączeń między transputerami ilustruje graf na rys. 8. Natomiast wyniki pomiarów prędkości samych transputerów wskazują, że wszystkie jednostki mają mniej więcej taką samą prędkość przetwarzania.

## 4 Model matematyczny

Model matematyczny został opracowany na podstawie pomysłu przedstawionego na rysunku 2 (*współdzielenie łańcucha*). Jednak by wyprowadzić wzory potrzebne były dodatkowe oznaczenia. Wprowadzono więc następujące zmiany i założenia:

1. Dla transputerów od 3 do  $n$  wyznaczany jest czas oczekiwania na dane z transputera 1, a dla transputerów od 1 do  $n-2$ , wyznaczany jest czas oczekiwania na dane z transputera  $n$ . Transputery w tym czasie są bezczynne.
2. Pierwsza porcja danych, jaką otrzyma transputer jest dzielona na dwie części: dane które zostaną obliczone do nadejścia drugiego połączenia, oraz dane które zostaną obliczone później.

$i, j$	$C_{i,j}$	$S_{i,j}$	$i, j$	$C_{i,j}$	$S_{i,j}$	$i$	$a_i$	$b_i$
01-02	0.008866	0.174200	06-09	0.017255	0.179462	01	0.038244	0.757526
01-04	0.016862	0.153659	07-08	0.008853	0.185529	02	0.039907	0.938374
01-10	0.016887	0.185913	07-10	0.017242	0.169452	03	0.039895	1.023457
02-03	0.008848	0.186130	08-09	0.008881	0.184632	04	0.038241	0.747327
02-05	0.017462	0.175640	08-11	0.017456	0.188101	05	0.039891	0.889240
02-11	0.016656	0.170649	09-12	0.017265	0.166424	06	0.039890	1.045919
03-06	0.017455	0.174845	10-11	0.008852	0.179192	07	0.038222	0.723403
03-12	0.016657	0.176856	11-12	0.008873	0.156769	08	0.039891	0.953436
04-05	0.009061	0.172019				09	0.039892	0.998836
04-07	0.017259	0.170635				10	0.038231	0.794916
05-06	0.008836	0.149225				11	0.039895	0.968761
05-08	0.017429	0.177466				12	0.039882	0.966867

Tabela 1: Współczynniki regresji liniowej wyznaczone z pomiarów szybkości łącz i szybkości transputerów

Wprowadzone oznaczenia:

- Zbiór transputerów liczy  $n = 12$  urządzeń:  $t \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c\}$ , czasy pracy transputerów:  $T_i, i \in t$ ,
- $V_{i1}$  - Dane dostępne na transputerze  $i \in t$ , pochodzące z transputera 1,
- $V_{i2}$  - dane dostępne na transputerze  $i \in t$ , pochodzące z ostatniego transputera,
- Zatem początkowy zbiór danych to  $V_{i1} + V_{c2}$ ,
- Czas oczekiwania, analogicznie jak dane,  $w_{i1}$  i  $w_{i2}$ , dla  $i \in t$ ,
- $k_{ij}$  - Czas komunikacji z transputera  $i$  do  $j$ ,
- $o_{i1}, o_{i2}$  - Czas obliczeń na transputerze  $i$ , gdy dane pochodzą odp. z transputera 1 i  $c$ ,
- $a_{ij}$  - Rozmiar danych do wykonania obliczeń  $o_{ij}$ .

Rozwiązaniem modelu matematycznego jest zadanie programowania liniowego postaci:

$$\min. \quad T \quad (3)$$

$$\text{ogr.} \quad T_i < T \quad i \in [1..n] \quad (4)$$

$$w_{i1} + k_{i-1,i} + k_{i,i+j} + o_{i1a} + w_{i2} + k_{i,i-1} + o_{i,i-1} + o_{i1b} = T_i \quad i \in t \quad (5)$$

$$o_{i1a} = k_{i+1,i+2} + o_{i+1,1a} + k_{i+2,i+1} \quad i \in [1.. \frac{n}{2} - 1] \quad (6)$$

$$o_{\frac{n}{2},1a} = k_{n,\frac{n}{2}+1} - k_{1,\frac{n}{2}} \quad (7)$$

$$o_{i2a} = k_{i-1,i-2} + o_{i-1,2a} + k_{i-2,i-1} \quad i \in [n.. \frac{n}{2} + 2] \quad (8)$$

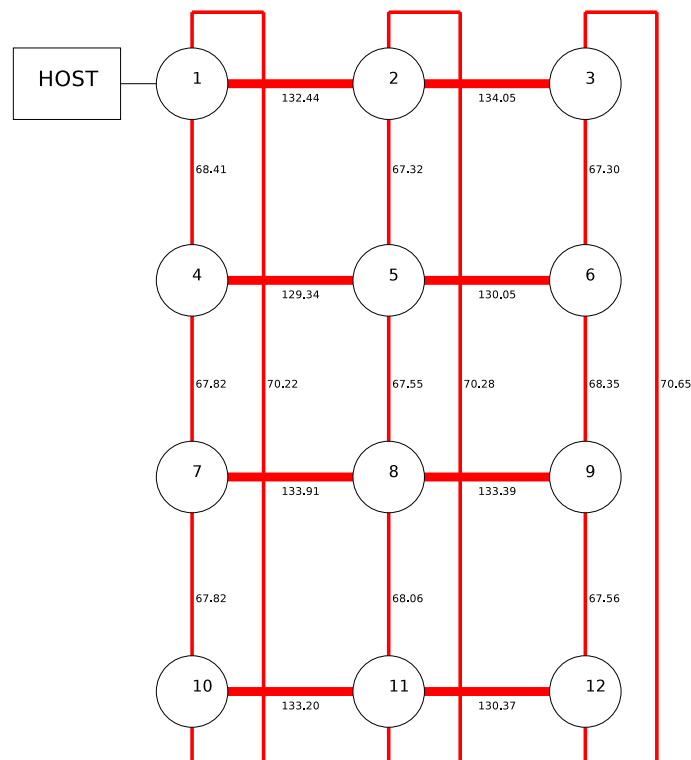
$$o_{\frac{n}{2}+1,2a} = k_{\frac{n}{2},\frac{n}{2}-1} \quad (9)$$

$$w_{i2} = k_{n,i+1} \quad i \in [\frac{n}{2} + 1, n - 2] \quad (10)$$

$$w_{i1} = k_{1,i-1} \quad i \in [3, \frac{n}{2}] \quad (11)$$

Tak więc minimalizacji podlega czas wykonywania pracy i jest on równy czasowi pracy tego transputera, który pracował najdłużej (Ograniczenia o numerach 4 i 5). Ograniczenia opisane kolejnymi czterema równaniami odzwierciedlają fakt, że transputer po otrzymaniu pierwszych danych może od razu przejść do obliczeń i poświęcić na nie tylko tyle czasu, by być gotowym, gdy przyjdzie druga porcja danych. Równania 6 i 7 określają ten warunek dla komunikacji w kierunku od pierwszego transputera, natomiast równania 8 i 9 - w kierunku od końca. Żaden transputer nie może rozpocząć przetwarzania, dopóki ich nie otrzyma, co opisują równania 10 i 11. Wyjątkiem są dwa pierwsze i dwa ostatnie transputery, które bez oczekiwania od razu rozpoczynają komunikację.

By nie komplikować powyższego zapisu, pominięto równania oczywiste, ilustrujące powiązanie rozmiaru danych z czasem obliczeń i czasem komunikacji, oraz podział danych.



Rysunek 8: Prędkości przesyłania wyrażone w liczbie bajtów na jednostkę czasu wyznaczonego przez zegar transputera.

#### 4.1 Wnioski dotyczące ograniczeń modelu

Nasz model, w postaci powyżej, nie jest doskonały, ani nie wyczerpuje wszystkich przypadków. Zakłada on, dla pierwszej połowy transputerów, że dane napłynął w pierwszej kolejności z jednostki nr. 1, a dla pozostałych z ostatniego. Urządzeniem, na którym następuje “spotkanie” strumieni danych jest transputer 6. Jednak model zakładający odwrotną kolej komunikacji, lub “spotkanie” na innym transputerze można wyprowadzić na podstawie naszego. W dalszej części pracy skupiliśmy się więc na analizie tylko tego modelu.

#### 4.2 Wybór wersji problemu

W analizie modelu przyjęto całkowitoliczbową wersję obliczeń. Niestety napotkaliśmy następujące problemy:

1. Opracowany model nie pozwala w prosty sposób znaleźć optymalnego rozwiązania całkowitoliczbowego.
2. Solver zwracał co najwyżej rozwiązania suboptymalne. Bliskie optymalnych rozwiązań ciągłych.

Mając na uwadze fakt, że testowanie rozwiązań ciągłych w praktyce, wymagałoby przebudowy sprzętu transputerowego, do jakiego mamy dostęp (wysyłanie ułamkowych części bajtów), doszliśmy do następującego kompromisu. Rozwiązania całkowitoliczbowe będą uzupełniane wnioskami z rozwiązań ciągłych, szczególnie gdy te drugie będą znacząco lepsze.

#### 4.3 Uzyskane wyniki

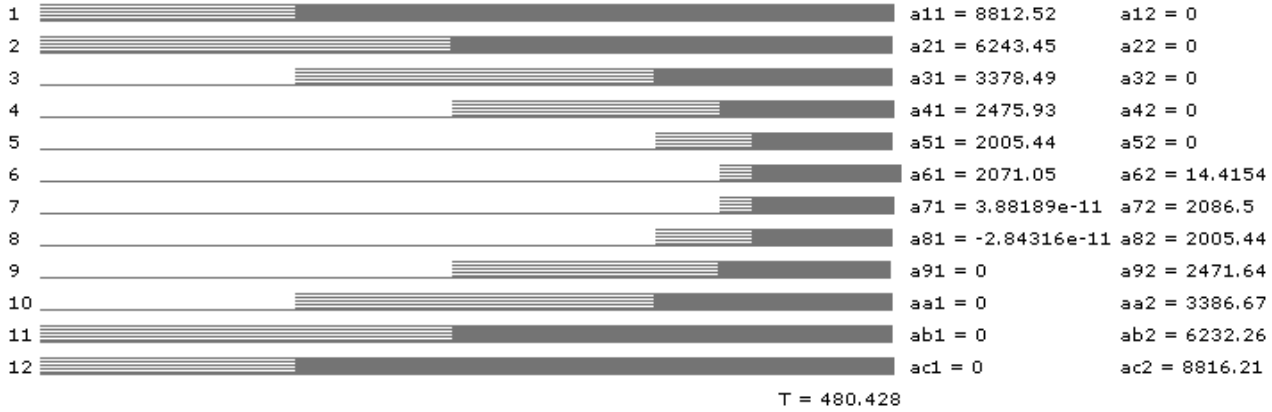
Model przebadano na czas przetworzenia 40000 danych, w konfiguracjach:

1. Łańcuch 12 transputerów o przeważającej liczbie szybkich łącz, symetryczny,
2. Łańcuch 12 transputerów o przeważającej liczbie wolnych łącz, symetryczny,
3. Łańcuch 12 transputerów, nie symetryczny,
4. Łańcuch 12 transputerów o przeważającej liczbie szybkich łącz, symetryczny z opóźnieniami komunikacyjnymi
5. Łańcuch 12, 10, 8 i 6 transputerów

#### 4.3.1 Łańcuch 12 transputerów o przeważającej liczbie szybkich łącz, symetryczny

Jest to taki sam model, jak na rysunku 5 na stronie 3. Aż 8 z 11 łącz łańcucha jest szybkich. Co więcej połączenia skrajne, czyli te na których przesyłane będzie najwięcej danych, są szybkie. Łańcuch jest symetryczny z punktu widzenia prędkości łącz na kolejnych ogniwach. Gdyby ułożyć go odwrotnie prędkości te byłyby takie same.

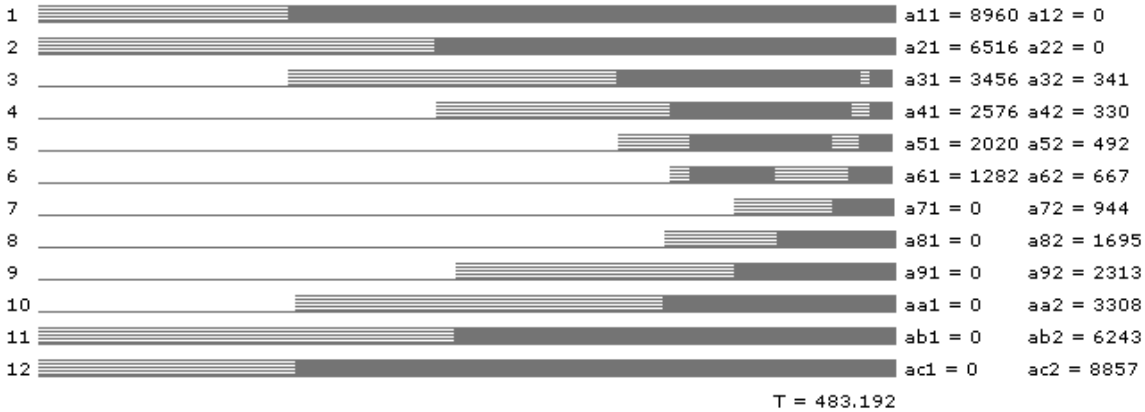
Poniższy wykres Gantt'a ilustruje znalezione ciągle rozwiązanie optymalne. Łączny czas wykonania to 480.428. Podkreśleniem oznaczony jest czas oczekiwania, na białą szaro oznaczona jest komunikacja, a kolorem jednolitym czas poświęcony na obliczenia.



Rysunek 9: Łańcuch [01,02,03,06,05,04,07,08,09,12,11,10], szybkie łącza, symetryczny, wersja ciągła, optymalne

Z boku widać rozmiary danych, jakie były przetwarzane w kolejnych ogniwach. W pierwszej kolumnie dane pochodzące z pierwszego transputera, w drugiej z ostatniego.

Natomiast poniższy wykres przedstawia rozwiązanie tego samego problemu, w wersji całkowitoliczbowej. Jest to rozwiązanie suboptymalne, uzyskane w 588 iteracji (lp\_solve z timeoutem 20 sekund).

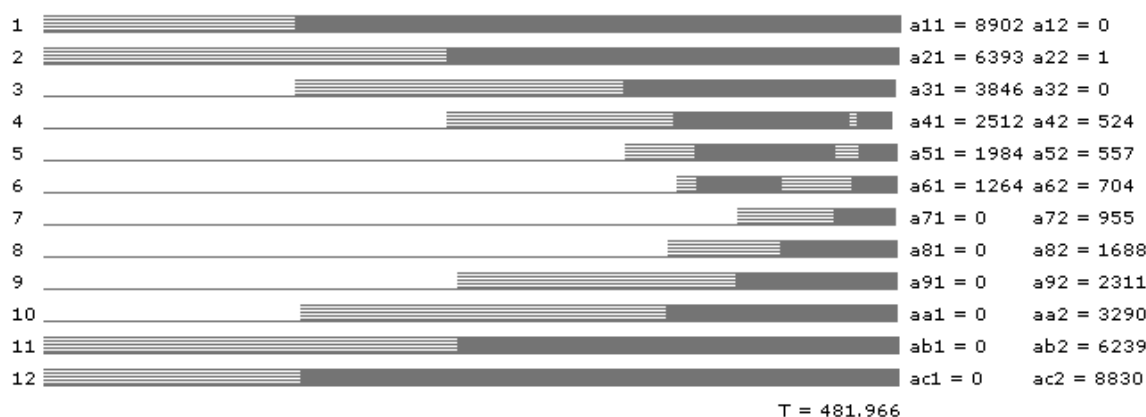


Rysunek 10: Łańcuch [01,02,03,06,05,04,07,08,09,12,11,10], szybkie łącza, symetryczny, Suboptymalne rozwiązanie całkowitoliczbowe

Zweryfikowano także założenie o symetryczności, rozwiązując układ równań dla transputerów ułożonych odwrotnie. Różnica  $\Delta T = 1.226$  między symetrycznymi rozwiązaniami, to skutek nie idealnie równych szybkości transputerów, oraz łącz. W danych pomiarowych prędkości transputerów różnią się od trzeciego miejsca po przecinku, a w prędkościach komunikacji, od czwartego. Jednak ogólne proporcje prędkości są zachowane. Założenie jest więc prawdziwe. Symetryczność ta jest także dobrze widoczna po podziale danych między pierwszy i ostatni transputer, dokonany solidarnie.

Podobnie różnica w czasie wykonania, między rozwiązaniem całkowitoliczbowym a ciągłym jest znikoma ( $\Delta T = 2.764$ ). Natomiast zastosowane schematy komunikacji są zupełnie odmienne.

Założenie pełnej podzielności zadań, skutkuje w równomiernym rozłożeniu obciążeń, uwidacznia tym samym charakter zadań podzielnych (*divisible loads*). Natomiast założenie całkowitoliczbowości, wymusza bardziej rzeczywiste

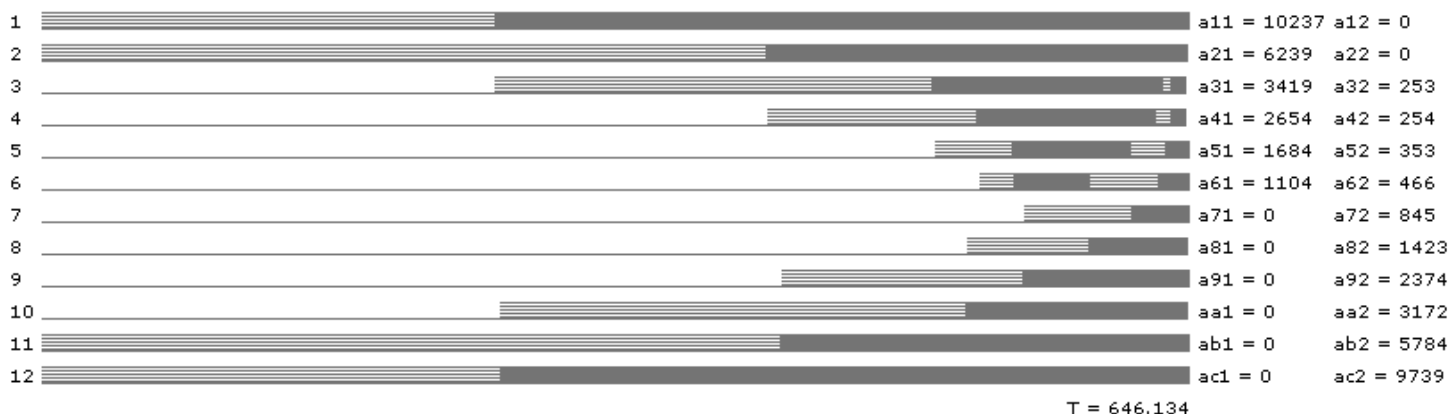


Rysunek 11: Łańcuch [10,11,12,09,08,07,04,05,06,03,02,01], szybkie łącza, Suboptymalne rozwiązanie całkowitoliczbowe

podejście. W nim okazuje się że skomplikowana komunikacja na transputerze 6 się nie opłaca. Mówiąc inaczej, zysk z wprowadzenia skomplikowanych modernizacji (komunikacja ułamkami bajtów) w laboratorium transputerowym byłby znikomy.

#### 4.3.2 Łańcuch 12 transputerów o przeważającej liczbie wolnych łącz, symetryczny

Kolejne rozwiązanie także ilustruje jeden z modeli komunikacji zastosowanych podczas pomiarów. Tym razem jest to model z rysunku 6, na stronie 3. W modelu tym, największa część komunikacji odbywa się na wolnych (pionowych) połączeniach w sieci transputerowej.



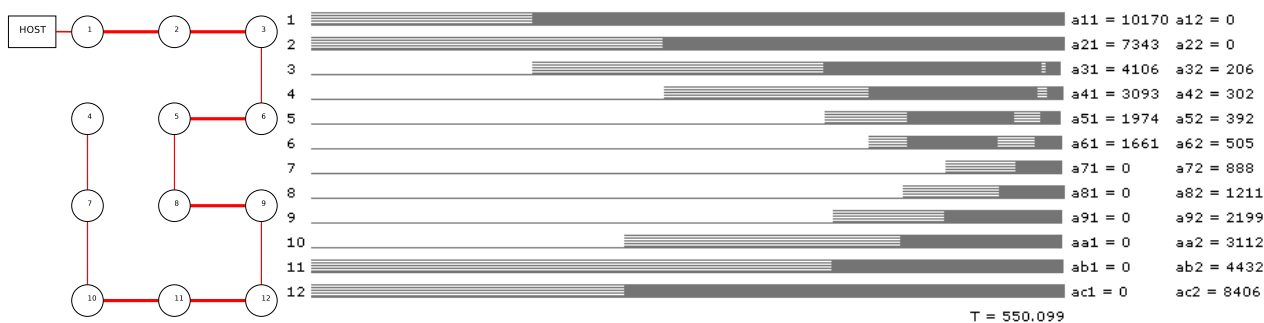
Rysunek 12: Łańcuch [01,04,07,10,11,08,05,02,03,06,09,12], wolne łącza, Suboptymalne rozwiązanie całkowitoliczbowe

Rozwiązanie jest dużo gorsze od poprzednich. Widać zatem, jak duże znaczenie ma jakość łącz. Mimo, że więcej danych jest przetwarzanych na skrajnych transputerach, największe straty czasu są związane z komunikacją w połączeniach 1 – 2 i 12 – 11, czyli tych gdzie wysyłane jest najwięcej danych.

#### 4.3.3 Łańcuch 12 transputerów, nie symetryczny

W tym przykładzie, postaramy się wykluczyć, że nasz model ma skłonność do równoważenia danych po równo między strumieniami. Do tego celu, łańcuch ułożono na sieci transputerowej tak, że z jednej strony jest przewaga szybkich łącz, a z drugiej wolnych.

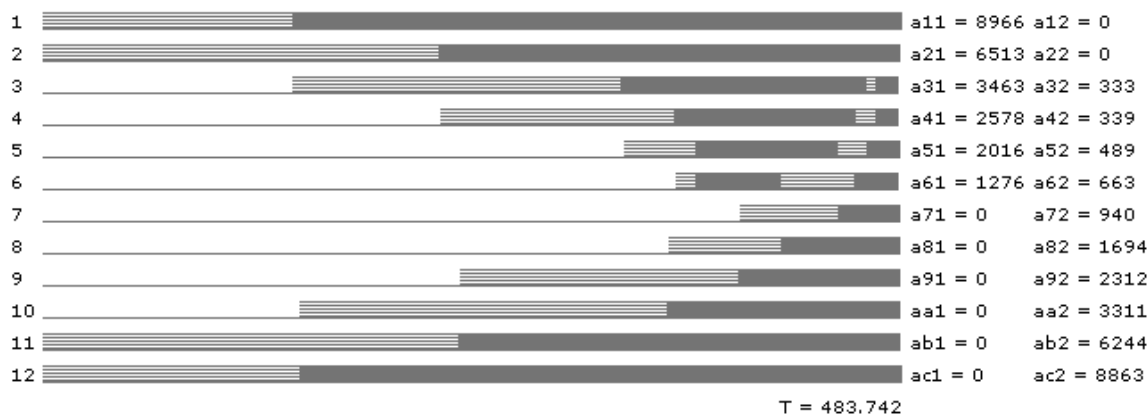
Widoczna jest istotna różnica proporcji ilości danych w strumieniu z transputera pierwszego, a ostatniego ( $V_{11} = 28371$ ,  $V_{c2} = 21629$ ). W optymalnym rozwiązaniu tego problemu w wersji ciągłej, wyniki są podobne:  $V_{11} = 28459.5$ ,  $V_{c2} = 21540.5$ . Dla porównania, wyniki dla rozwiązania z rysunku 10 (symetrycznego) to  $V_{11} = 24810$ ,  $V_{c2} = 25190$ .



Rysunek 13: Komunikacja na łańcuchu o niesymetrycznych prędkościach połączeń.

#### 4.3.4 Łańcuch 12 transputerów o przeważającej liczbie szybkich łącz, symetryczny z opóźnieniami komunikacyjnymi

Opóźnienia komunikacyjne, tj. *startup time*, w pomiarach oznaczone jako *S*, do tej pory nie były brane pod uwagę. Należy się jednak upewnić, czy i jak duży mają wpływ na całkowity czas przetwarzania.



Rysunek 14: Łańcuch [01,02,03,06,05,04,07,08,09,12,11,10], uwzględniając opóźnienia komunikacyjne

Nie ma więc prawie żadnej różnicy w porównaniu z rozwiązaniem nie uwzględniającym opóźnień (rys. 10). Opóźnienia w komunikacji są pomijalnie małe przy badanym rozmiarze problemu.

#### 4.3.5 Łańcuch 12, 10, 8 i 6 transputerów

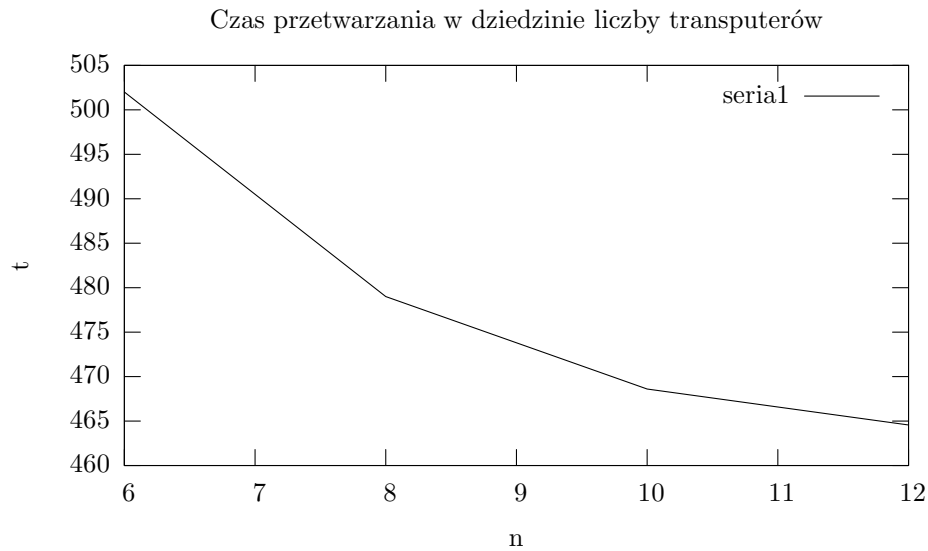
W tym punkcie oceniamy skalowalność naszego rozwiązania, tj. jak liczba jednostek obliczeniowych wpływa na zwiększenie szybkości przetwarzania. Wyjątkowo założyliśmy tu, że wszystkie połączenia i transputery są jednakowe, by różnice między nimi nie wpłynęły na wynik. Poniżej przedstawiono wykres Gantt'a jedynie dla ostatniego badanego łańcucha, o długości 6, bo dla łańcuchów o innej długości mechanizm komunikacji właściwie się nie zmienia.



Rysunek 15: Połączenie 6 transputerów

Bardziej interesujący jest wykres na rysunku 16 obrazujący zmianę czasu przetwarzania w dziedzinie liczby transputerów. Płyne z niego wniosek, że zwiększanie liczby jednostek obliczeniowych daje coraz mniejsze zyski. Do wyników tych powrócimy jeszcze w analizie efektywności przetwarzania równoległego.





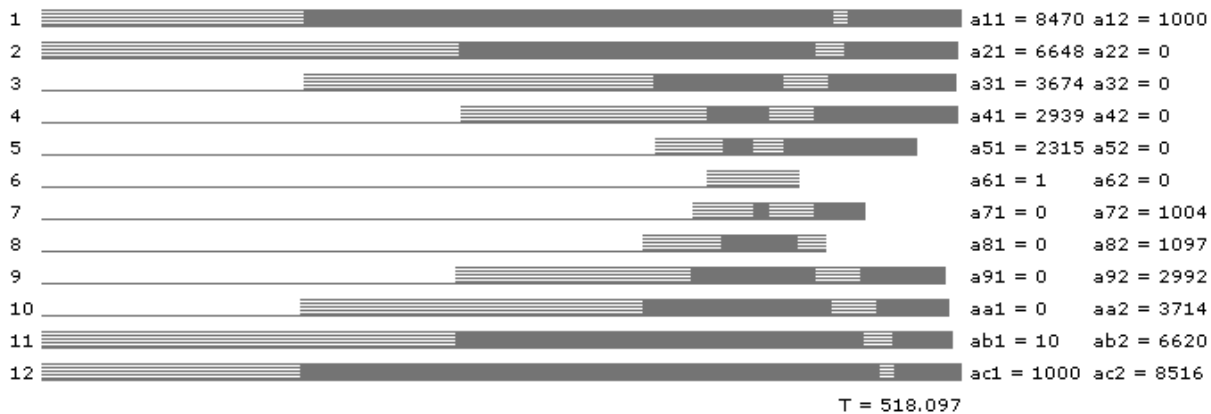
Rysunek 16: Wykres zależności czasu przetwarzania od liczby transputerów

n	t(n)
6	502.006
8	479.002
10	468.606
12	464.559

Tabela 2: Dane do wykresu zależności czasu przetwarzania od liczby transputerów

#### 4.3.6 Sens współdzielenia łącza

Analizując komunikację na modelowym łańcuchu i kolejno otrzymując prawie identyczne wykresy Gantt'a, zadaliśmy sobie pytanie, czy zaproponowany model komunikacji, z przecinaniem się strumieni danych ma w ogóle sens i kiedykolwiek zajdzie. Po uwzględnieniu dodatkowego wymagania, że do pierwszego transputera muszą dotrzeć dane z ostatniego, oraz w drugą stronę także, otrzymaliśmy rozwiązanie:



Rysunek 17: Komunikacja z wymuszonym przepływem danych

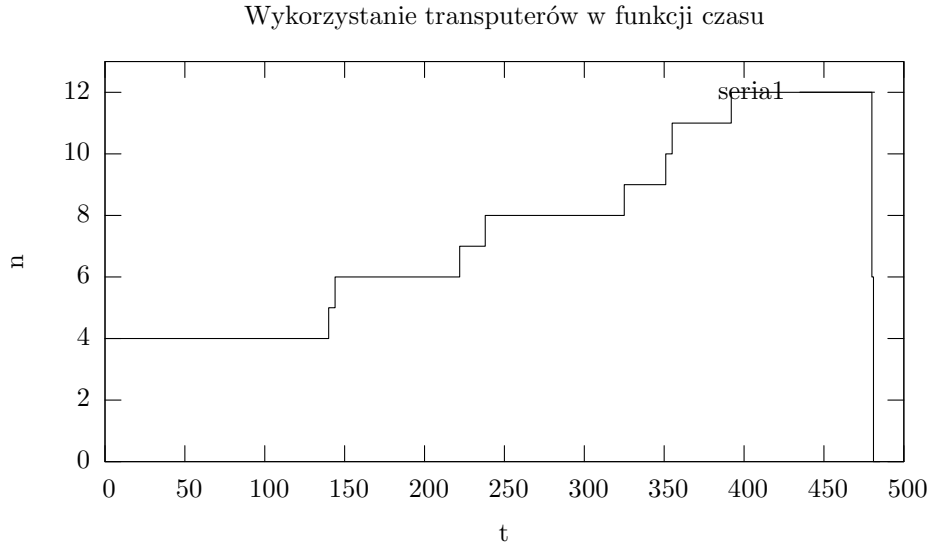
Niestety w architekturze łańcucha, jaką mamy, nie da się przesłać dostatecznie szybko wystarczająco dużo danych, dlatego środkowe jednostki odgrywają jedynie rolę komunikacyjną a później w większości pozostają bezczynne. Niestety całkowity czas takiego rozwiązania faktycznie nie jest zadowalający. Wykres ten można porównywać z innymi, bo model opiera się na danych testowych, takich jak w punkcie 4.3.1.

## 4.4 Efektywność obliczeń równoległych

Mając już pewne pojęcie o zachowaniu modelu badanego problemu, możemy wyznaczyć efektywność naszego rozwiązania, korzystając z kilku standardowych metod.

### 4.4.1 Wykorzystanie transputerów w trakcie przetwarzania

Nie wszystkie maszyny są cały czas wykorzystywane podczas realizacji zadania. Wynika to w głównej mierze z charakteru łańcucha. Drugim kluczowym czynnikiem jest stosunek prędkości komunikacji do szybkości obliczeń.



Rysunek 18: Wykres wykorzystania transputerów w trakcie przetwarzania

### 4.4.2 Przyspieszenie

Przyspieszenie na  $n$  maszynach, czyli stosunek czasu najlepszego algorytmu sekwencyjnego do czasu na  $n$  maszynach, z punktu widzenia naszego modelu to:

$$S(n) = \frac{t_{sekw}}{t(n)} = \frac{a * V}{t(12)} = \frac{0.039 * 50000}{483.192} = \frac{1950}{483.192} = 4.036 \quad (12)$$

Najlepszy czas algorytmu sekwencyjnego  $t_{sekw}$ , to iloczyn rozmiaru danych  $V$  i szybkości jednostki obliczeniowej. Rozmiar danych jest taki, jak w pozostałych modelach, szybkość to średnia szybkość transputera, z tabeli 1. Jako czas rozwiązania równoległego przyjęto wynik najlepszego rozwiązania wersji całkowitoliczbowej.

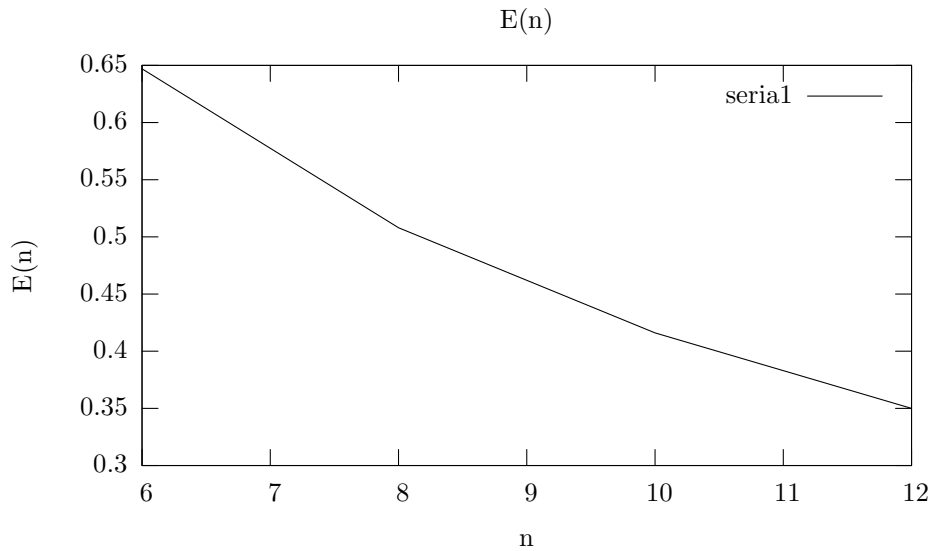
n	6	8	10	12
S(n)	3.885	4.071	4.161	4.203

Tabela 3: Zmiana przyspieszenia w funkcji liczby procesorów.

### 4.4.3 Efektywność

Zależność efektywności od liczby jednostek w łańcuchu przedstawia wykres na rysunku 19. Powraca więc wniosek, z wcześniejszego punktu, że wydłużanie łańcucha daje coraz mniejsze korzyści. Efektywność została wyznaczona, korzystając z wyników wcześniejszych wyliczeń, oraz wzoru:

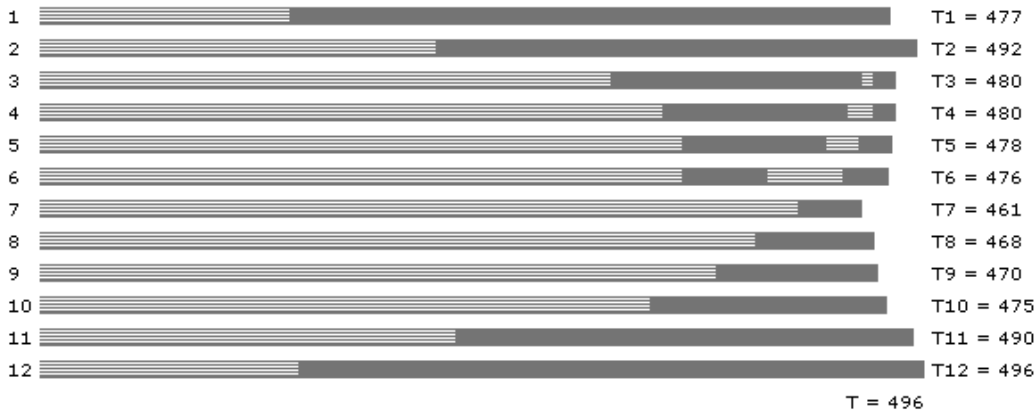
$$E(n) = \frac{S(n)}{n} \quad (13)$$



Rysunek 19: Wykres zależności efektywności od liczby jednostek w łańcuchu

## 5 Rozwiązanie rzeczywiste

Poniższy diagram przedstawia rozwiązanie rzeczywiste, jakie udało się osiągnąć dla zbioru danych wyliczonego w punkcie 4.3.1. Jest to więc łańcuch jak na rysunku 5. Nie był tu już możliwy pomiar czasu bezczynności transputerów. Jednak urządzenia pozostawały bezczynne oczekując na komunikację, dlatego czasy komunikacji pokryły czasy bezczynności z modelu matematycznego. Różnice nie są bardzo znaczące. Większość urządzeń faktycznie wykonała



Rysunek 20: Rozwiązanie rzeczywiste

zadanie w ciągu ok 480 jednostek czasu. Natomiast środkowe szybciej uporały się z zadaniem, a skrajnym zajęło to więcej czasu. Przyczyny różnic między rozwiązaniem teoretycznym a rzeczywistym, to wg. nas:

1. Niedokładność pomiarów w środowisku rzeczywistym. Zdarzało się, że z niektórych transputerów przychodziły złe wyniki pomiarów. Np. czas każdego połączenia był mierzony przez obie strony, choć później wyniki były różne. Dopiero po wykonaniu próbnej komunikacji, ten problem zniknął,
2. Niedokładność współczynników wyliczonych w fazie pomiarów,
3. Wpływ czynników zewnętrznych. Z wykonania, na wykonanie, wyniki różniły czasem o kilka jednostek.
4. Charakter różnic pozwala przypuszczać, że w rzeczywistości komunikacja była realizowana szybciej, natomiast obliczenia wolniej niż w modelu. Skrajne transputery więcej czasu poświęciły na obliczenia, a środkowe, które się głównie komunikują, zrobiły to szybciej.