

Sprawozdanie laboratorium 4

Marcin Jarczewski 330234

Bartosz Jasiński 318777

Cel laboratorium:

- wykorzystanie komunikacji międzyprocesowej w czasie rzeczywistym na przykładzie OpenWRT i Raspberry Pi, uwzględniając opóźnienia wynikające między współpracującymi procesami podczas komunikacji

Zadanie 1:

- Skompilowaliśmy program pakiet udostępniony do realizacji laboratorium poleceniem `make`.
- Zmodyfikowaliśmy zmienną path poprzez dodanie ścieżki do folderu z programami, następnie uruchomiliśmy program.

```
export PATH="/home/user/Pobrane/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src:$PATH"
```

Na zamieszczonych zrzutach ekranu widać, że program zadziałał i pliki wynikowe wygenerowały się poprawnie.

```

user@lab-39:~/Pobrane/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ ./cw4a 3 5 10000 3
Client: 0, nsmp=5, del=3
Client: 1, nsmp=5, del=3
Client: 2, nsmp=5, del=3
Sample 0, client 1, delivery time: 47
Sample 0, client 2, delivery time: 72
Sample 0, client 0, delivery time: 92
Sample 1, client 0, delivery time: 38
Sample 1, client 2, delivery time: 46
Sample 1, client 1, delivery time: 47
Sample 2, client 0, delivery time: 38
Sample 2, client 1, delivery time: 48
Sample 2, client 2, delivery time: 48
Sample 3, client 2, delivery time: 31
Sample 3, client 1, delivery time: 38
Sample 3, client 0, delivery time: 37
Sample 4, client 0, delivery time: 39
Sample 4, client 2, delivery time: 48
Sample 4, client 1, delivery time: 48
waiting for childrenuser@lab-39:~/Pobrane/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ cat server.txt
0, 1683816677002852, 1683816677002852
1, 1683816677141839, 138987
2, 1683816677216372, 74533
3, 1683816677266369, 49997
4, 1683816677316447, 50078
user@lab-39:~/Pobrane/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ cat cli_0.txt
0, 1683816677002852, 1683816677002944, 92
1, 1683816677141839, 1683816677141877, 38
2, 1683816677216372, 1683816677216410, 38
3, 1683816677266369, 1683816677266406, 37
4, 1683816677316447, 1683816677316486, 39

```

Zadanie 2:

- Pobraliśmy sdk, zmodyfikowaliśmy plik `feeds.conf.default` i zastosowaliśmy polecenia z poprzedniego laboratorium aby zbudować pakiet.
- Przed zbudowaniem w OpenWRT usunęliśmy pliki wynikowe poprzedniego budowania.
- Przesłaliśmy pakiet przy pomocy polecenia wget na OpenWRT.
- Zainstowaliśmy pakiet i dodatkowo popraliśmy pakiety htop i stress-ng poleceniem:

```

- opkg install
- opkg update

```

Zadanie 3:

Miedzy pomiarami modyfikowaliśmy plik `cmdline.txt` zmieniając parametr maxcpus opsiujacy liczbę dostępnych rdzeni. Natomiast pomiary z obciążeniem uruchomiliśmy poleceniem(gdzie N było wartością graniczną) :

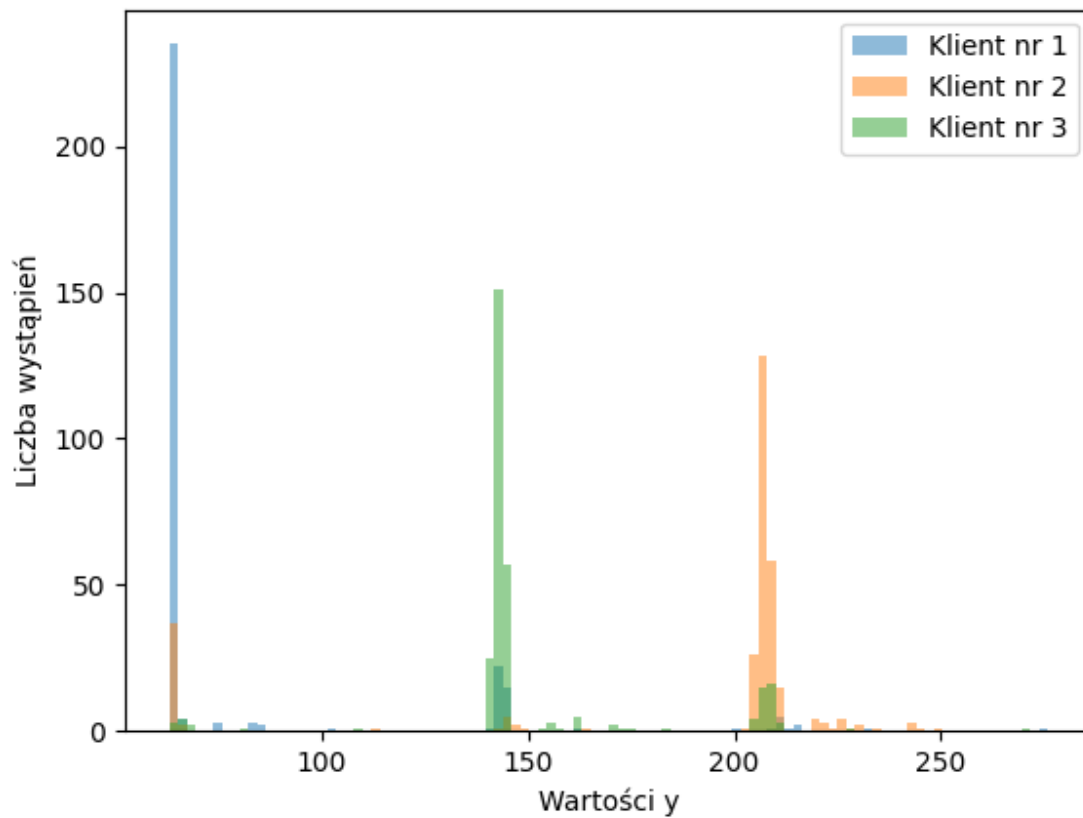
```
nano /boot/user/cmdline.txt  
stress-ng --matrix 0 -t 10s & cw4a 3 300 10000 N
```

	Wariant 1	Wariant 2	Wariant 3	Wariant 4
Liczba rdzeni	1	2	2	4
Liczba klientów	3	3	3	1
Obciążenie	pełne	pełne	brak	brak
Graniczna wartość	300000	500000	630000	790000

Zadanie 4

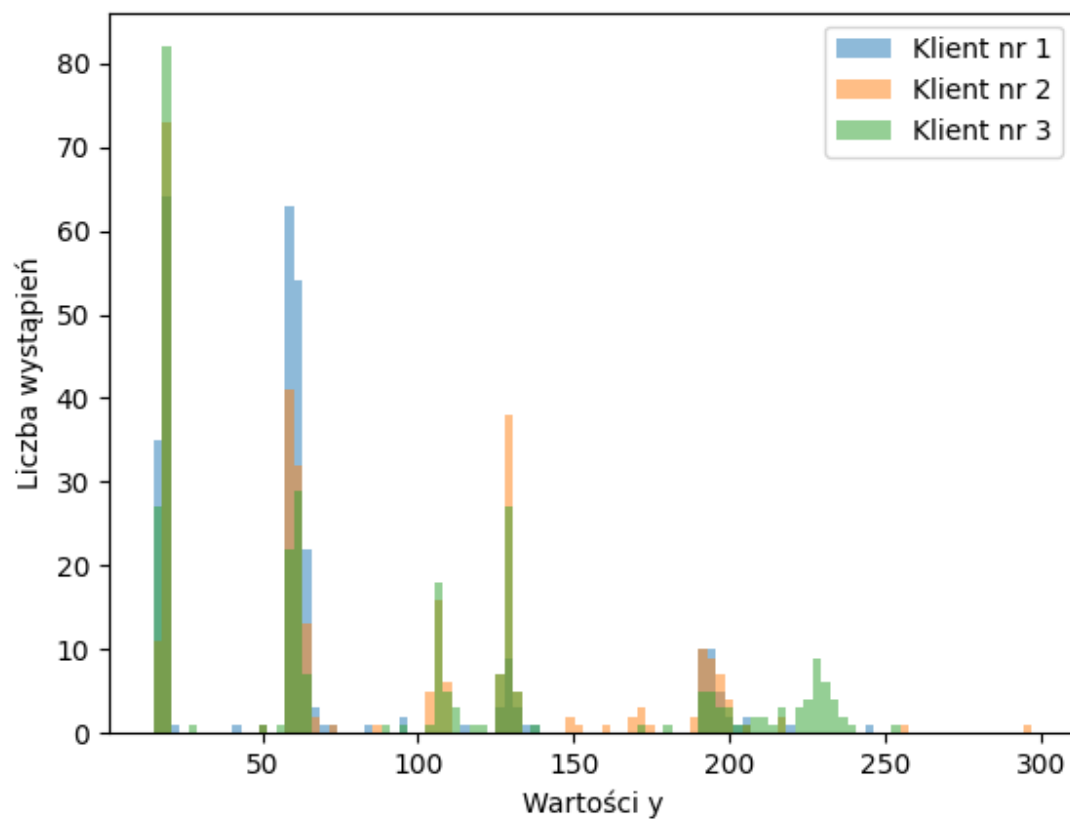
Wariant 1: 3 klientów, 1 rdzeń, pełne obciążenie

Ustalona wartość graniczna: 300000



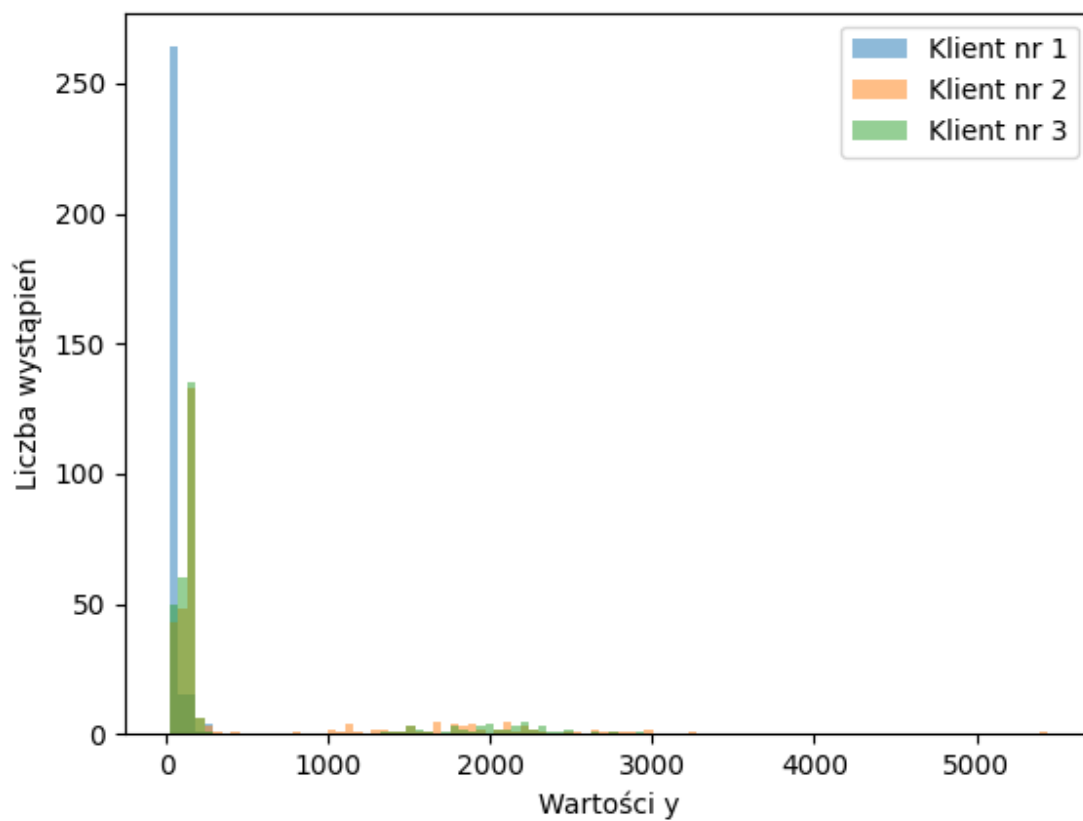
Wariant 2: 3 klientów, 2 rdzenie, pełne obciążenie

Ustalona wartość graniczna: 500000



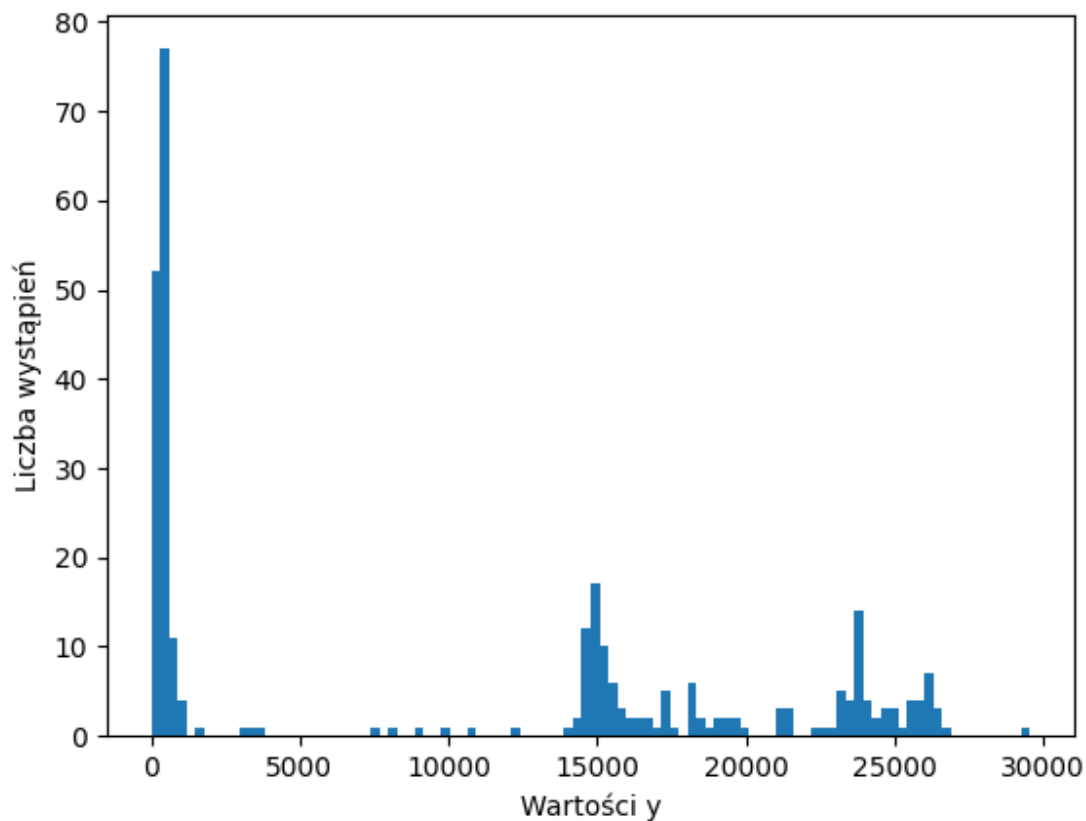
Wariant 3: 3 klientów, 2 rdzenie, bez obciążenia

Ustalona wartość graniczna: 630000



Wariant 4: 1 klient, 4 rdzenie, bez obciążenia

Ustalona wartość graniczna: 790000



Podsumowanie

- Udało się w trakcie laboratorium wykonać zadania 1, 2, 3 i 4. Wykonane zadania i histogramy zostały zatwierdzone przez prowadzącego.
- Główne cele laboratorium zostały przez nas osiągnięte
- Poszerzyliśmy swoją wiedzę i przećwiczyliśmy zagadnienia związane z komunikacją międzyprocesową, wydajnością platformy sprzętowej w zależności od liczby dostępnych rdzeni
- Nie zdążyliśmy wykonać modyfikacji w ramach polecenia 5 i 6