SKPS laboratorium nr 4

Kinga Świderek (318 734) i Jakub Kowalczyk (318 676)

0. Przygotowanie stanowiska pracy i SDK

Przed rozpoczęciem pracy pobraliśmy SDK do folderu openwrt-sdk oraz pobraliśmy i zainstalowaliśmy pakiet cwicz4mak analogiczne jak na poprzednich laboratoriach.

1. Przetestowanie działania programów na “gospodarzu”.

Udało się poprawnie zbudować pakiet poleceniem make:

```
user@lab-18:~/skps_lab4_student/cw4_owrt_pkg/cwicz4mak$ cd src/
user@lab-18:~/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ ls
Makefile  cw4a.c  cw4a.h  cw4b.c
user@lab-18:~/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ make
cc -c  cw4a.c -o cw4a.o
cw4a.c: In function 'main':
cw4a.c:73:28: warning: 'sprintf' writing a terminating nul past the end of the d
estination [-Wformat-overflow=]
             sprintf(nr,"%5d",i);
                            ^
cw4a.c:73:13: note: 'sprintf' output between 6 and 12 bytes into a destination o
f size 5
             sprintf(nr,"%5d",i);
             ^~~~~~~~~~~~~~~~~~~
cc -o cw4a cw4a.o  -pthread -lrt
cc -c  cw4b.c -o cw4b.o
cc -o cw4b cw4b.o -pthread -lrt
```

Oraz uruchomić go na gospodarzu z uwzględnieniem ścieżki w zmiennej PATH:

```
user@lab-18:~/skps_lab4_student/cw4_owrt_pkg/cwicz4mak/src$ PATH="$PWD:$PATH" cw
4a 3 5 10000 5
Client: 0, nsmp=5, del=5
Client: 1, nsmp=5, del=5
Client: 2, nsmp=5, del=5
Sample 0, client 0, delivery time: 8471
Sample 0, client 1, delivery time: 8488
```

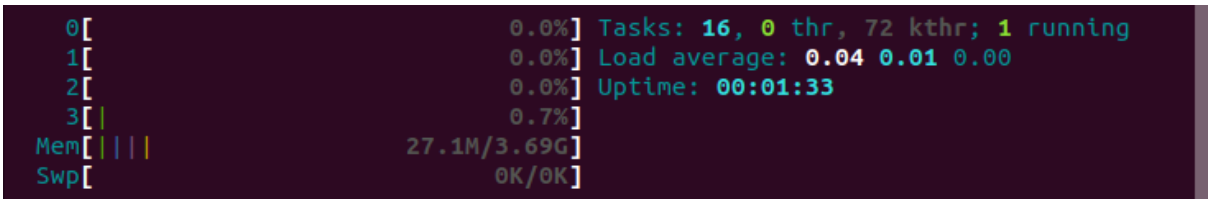
2. Zbudowanie pakietu dla OpenWRT

Tak jak na poprzednich laboratorium, przesłaliśmy pakiet za pomocą komendy wget i zainstalowaliśmy przez opkg.

```
root@OpenWrt:/# wget http://192.168.9.104:8000/cwicz4mak_1_aarch64_cortex-a72.ip
k
Downloading 'http://192.168.9.104:8000/cwicz4mak_1_aarch64_cortex-a72.ipk'
Connecting to 192.168.9.104:8000
Writing to 'cwicz4mak_1_aarch64_cortex-a72.ipk'
cwicz4mak_1_aarch64_ 100% |*******************************|  4903   0:00:00 ETA
Download completed (4903 bytes)
root@OpenWrt:/# opkg install cwicz4mak_1_aarch64_cortex-a72.ipk
Installing cwicz4mak (1) to root...
Configuring cwicz4mak.
```

3. Ustalenie granicznej wartości czasu przetwarzania

Na początku zainstalowaliśmy programy (opkg install) stress-ng oraz htop. W celu zmiany liczby wykorzystywanych rdzeni edytowaliśmy plik */boot/user/cmdline.txt* dodając parametr *maxcpus = (1/2/4)*. Po każdej zmianie liczby rdzeni przeprowadzaliśmy *reboot*. Poprawność liczby pracujących rdzeni sprawdzaliśmy w programie htop, obserwując ich obciążenie.



Pełne obciążenie uzyskaliśmy komendą: *stress-ng --matrix 0 -t 1m & cw4a 3 100 10000 X*

Wariant 1. 3 klientów, 1 rdzeń, pełne obciążenie.
Wynik: 440 000

```
root@OpenWrt:/# stress-ng --matrix 0 -t 10s & cw4a 3 100 10000 440000
Client: 2, nsmp=100, del=440000
Client: 1, nsmp=100, del=440000
Client: 0, nsmp=100, del=440000
stress-ng: info:  [1632] dispatching hogs: 1 matrix
Sample 0, client 2, delivery time: 251
Sample 0, client 1, delivery time: 365
Sample 0, client 0, delivery time: 465
Sample 1, client 1, delivery time: 9888
Sample 1, client 0, delivery time: 12251
Sample 1, client 2, delivery time: 12693
Sample 2, client 0, delivery time: 21344
Sample 2, client 1, delivery time: 23946
Sample 2, client 2, delivery time: 30437
Sample 3, client 0, delivery time: 21224
Sample 3, client 1, delivery time: 31261
Sample 3, client 2, delivery time: 32180
Sample 4, client 1, delivery time: 40161
Sample 4, client 2, delivery time: 46916
Sample 4, client 0, delivery time: 47429
Sample 5, client 0, delivery time: 57398
Sample 5, client 1, delivery time: 62586
Sample 5, client 2, delivery time: 62871
Sample 6, client 1, delivery time: 53326
Sample 6, client 0, delivery time: 56886
Sample 6, client 2, delivery time: 60168
Sample 7, client 1, delivery time: 65708
Sample 7, client 2, delivery time: 69265
Sample 7, client 0, delivery time: 72622
Sample 8, client 0, delivery time: 63631
```

Wariant 2. 3 klientów, 2 rdzenie, pełne obciążenie.
Wynik: 750 000

```
root@OpenWrt:/# stress-ng --matrix 0 -t 10s & cw4a 3 100 10000 750000
Client: 2, nsmp=100, del=750000
Client: 0, nsmp=100, del=750000Client: 1, nsmp=100, del=750000

stress-ng: info:  [1657] dispatching hogs: 2 matrix
Sample 0, client 0, delivery time: 57
Sample 0, client 2, delivery time: 161
Sample 0, client 1, delivery time: 268
Sample 1, client 0, delivery time: 4052
Sample 1, client 1, delivery time: 5917
Sample 1, client 2, delivery time: 13419
Sample 2, client 0, delivery time: 16500
Sample 2, client 1, delivery time: 29715
Sample 2, client 2, delivery time: 30273
Sample 3, client 0, delivery time: 18228
Sample 3, client 2, delivery time: 40539
Sample 4, client 0, delivery time: 31702
Sample 3, client 1, delivery time: 50827
Sample 4, client 1, delivery time: 55839
Sample 5, client 0, delivery time: 47933
Sample 4, client 2, delivery time: 67817
Sample 5, client 1, delivery time: 65211
Sample 6, client 1, delivery time: 61852
Sample 6, client 0, delivery time: 62725
Sample 5, client 2, delivery time: 92222
Sample 7, client 1, delivery time: 69932
Sample 7, client 0, delivery time: 74053
Sample 8, client 1, delivery time: 72599
Sample 6, client 2, delivery time: 104880
```

Wariant 3. 3 klientów, 2 rdzenie, bez obciążenia.
Wynik: 1 000 000

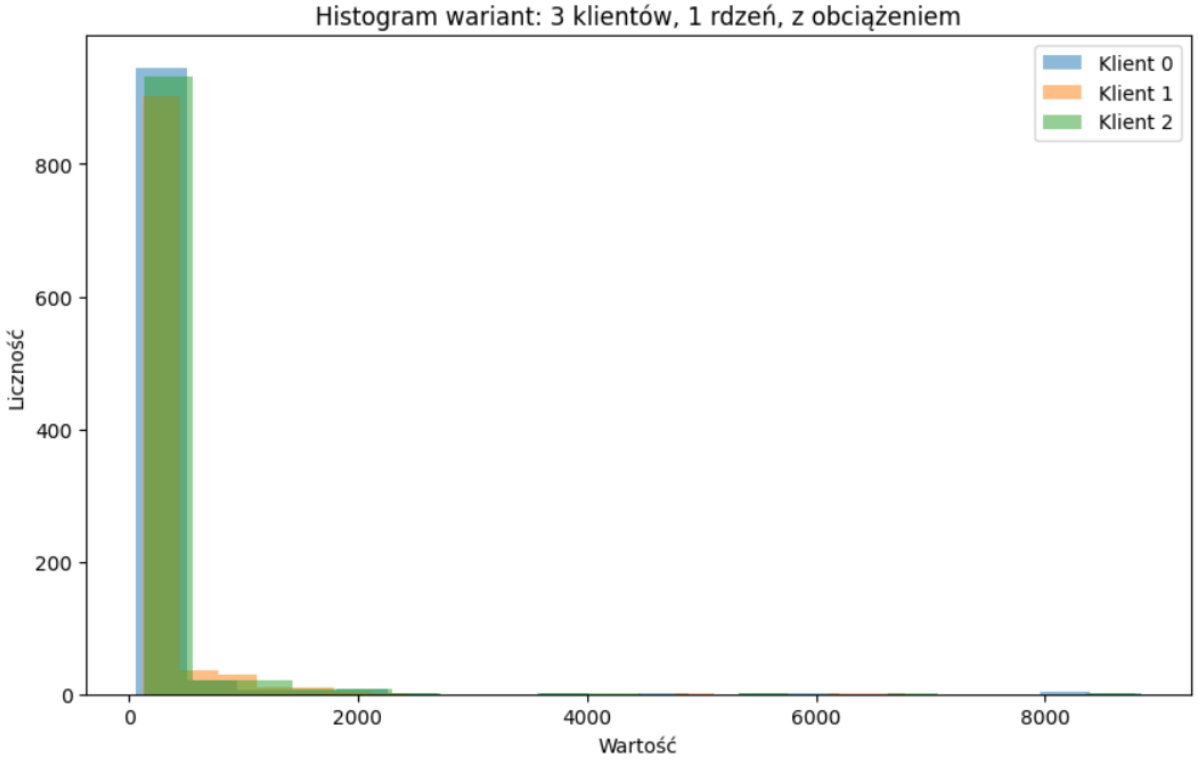
```
root@OpenWrt:/# cw4a 3 100 10000 1000000
Client: 0, nsmp=100, del=1000000
Client: 1, nsmp=100, del=1000000
Client: 2, nsmp=100, del=1000000
Sample 0, client 0, delivery time: 30
Sample 0, client 2, delivery time: 148
Sample 0, client 1, delivery time: 256
Sample 1, client 0, delivery time: 5196
Sample 1, client 1, delivery time: 14443
Sample 1, client 2, delivery time: 25710
Sample 2, client 1, delivery time: 17823
Sample 2, client 0, delivery time: 25029
Sample 3, client 1, delivery time: 20855
Sample 2, client 2, delivery time: 35206
Sample 4, client 1, delivery time: 24009
Sample 3, client 0, delivery time: 37830
Sample 3, client 2, delivery time: 43438
Sample 5, client 1, delivery time: 30896
Sample 4, client 2, delivery time: 54163
Sample 4, client 0, delivery time: 56655
```

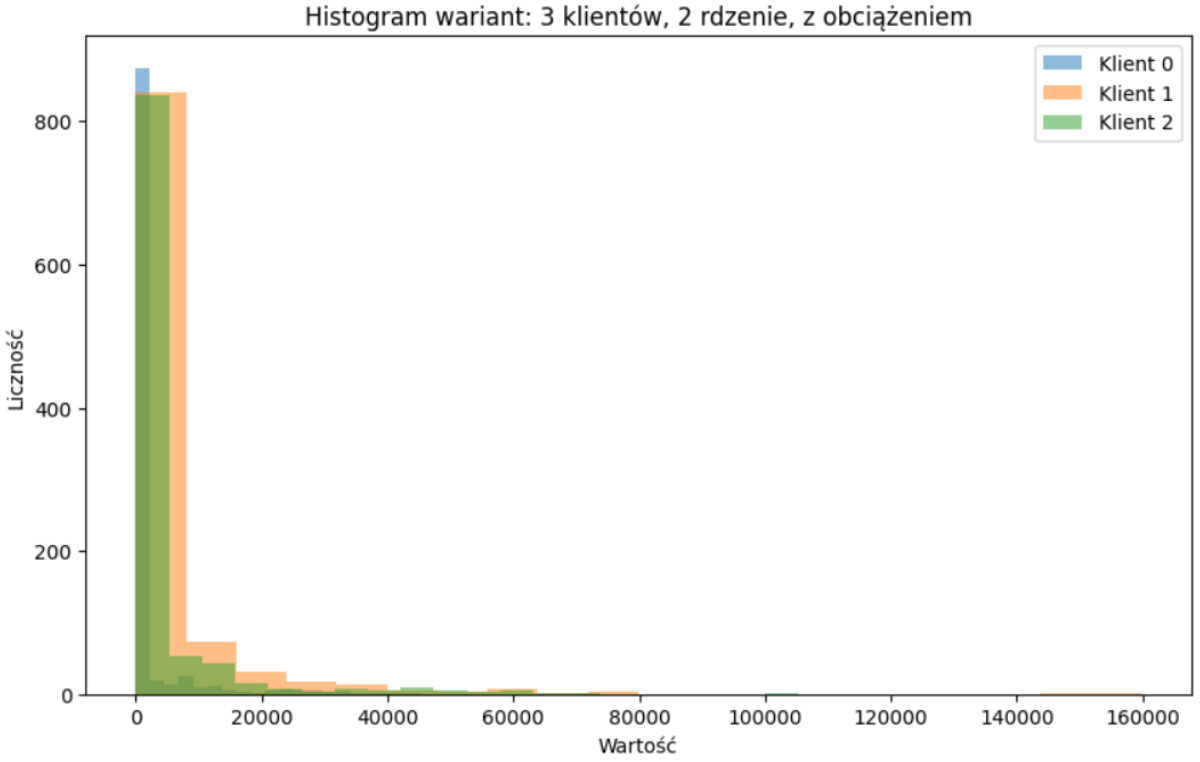
Wariant 4. 1 klient, 4 rdzenie, bez obciążenia.
Wynik: 840 000

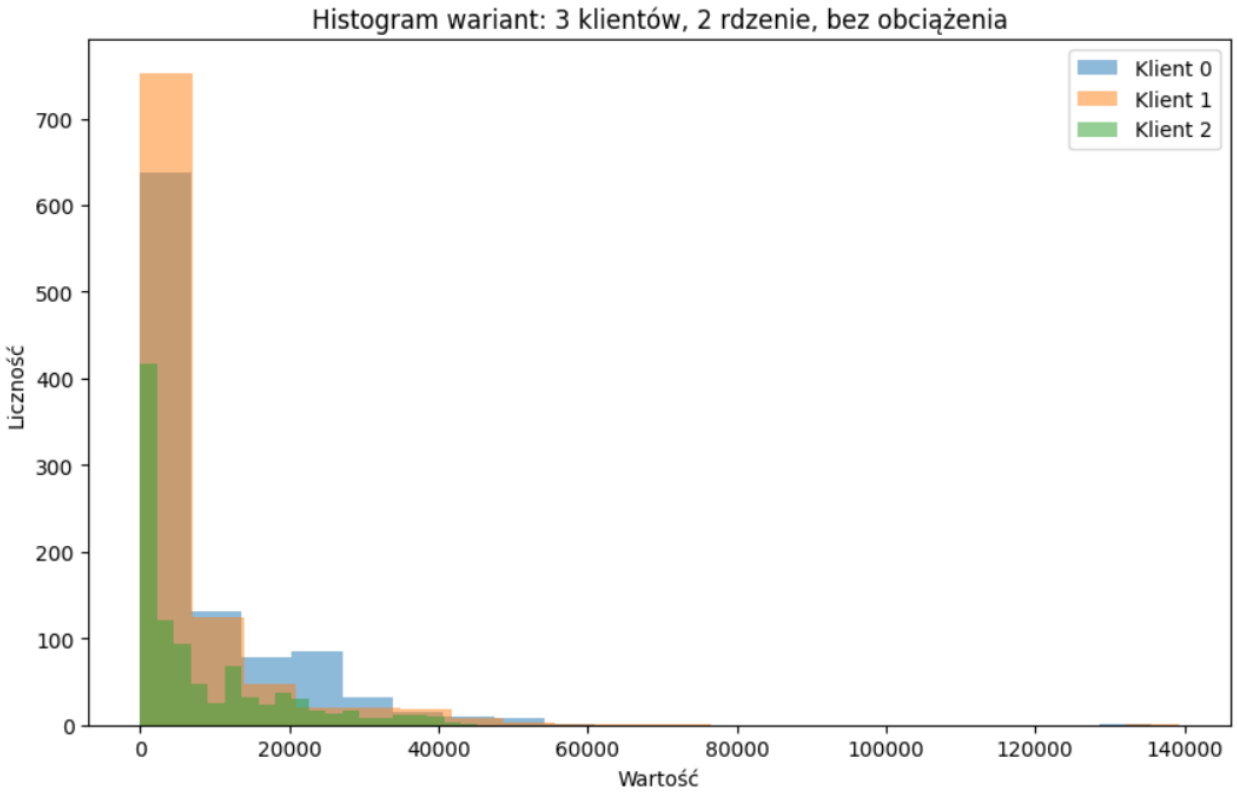
```
root@OpenWrt:/# cw4a 1 100 10000 840000
Client: 0, nsmp=100, del=840000
Sample 0, client 0, delivery time: 26
Sample 1, client 0, delivery time: 2230
Sample 2, client 0, delivery time: 3975
Sample 3, client 0, delivery time: 5616
Sample 4, client 0, delivery time: 7777
Sample 5, client 0, delivery time: 8523
Sample 6, client 0, delivery time: 10055
Sample 7, client 0, delivery time: 10468
Sample 8, client 0, delivery time: 15936
Sample 9, client 0, delivery time: 16506
Sample 10, client 0, delivery time: 15687
Sample 11, client 0, delivery time: 16526
Sample 12, client 0, delivery time: 17039
Sample 13, client 0, delivery time: 18444
Sample 14, client 0, delivery time: 19025
Sample 15, client 0, delivery time: 19414
Sample 16, client 0, delivery time: 20875
Sample 17, client 0, delivery time: 22913
```

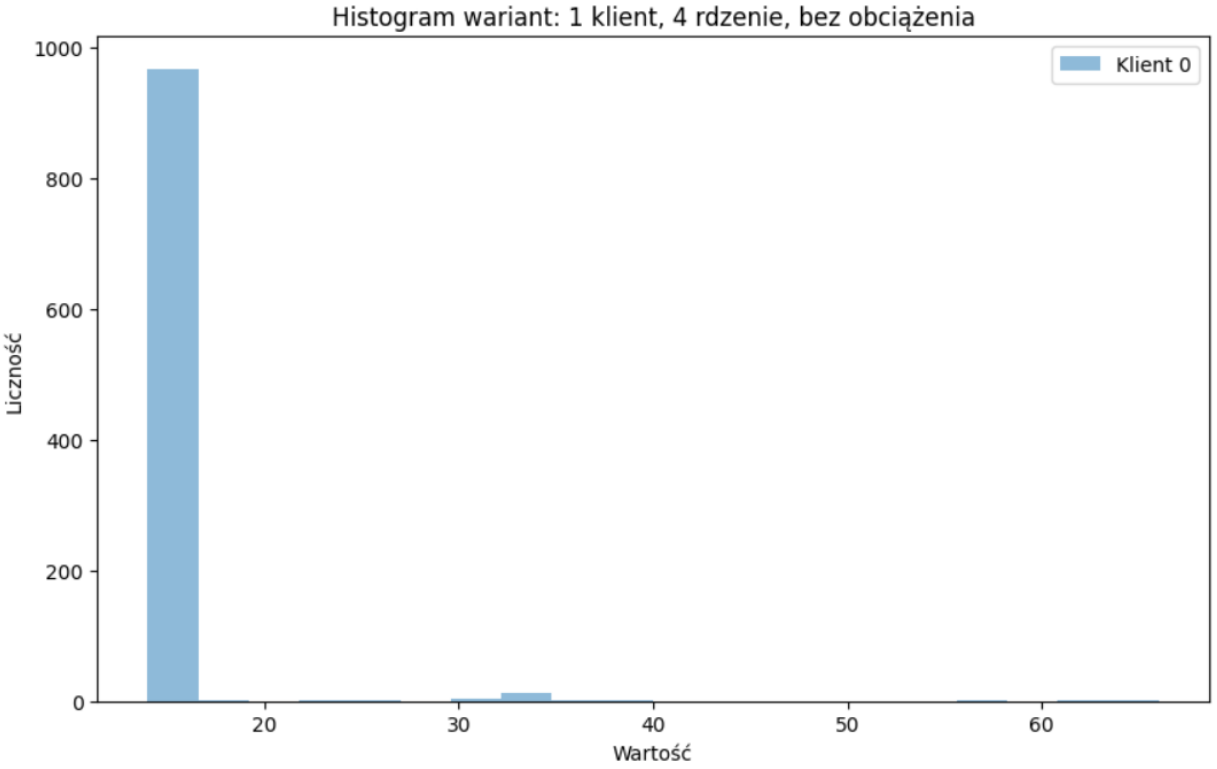
4. Rozkład czasu dostarczenia danych

Zbadaliśmy rozkład wartości czasu dostarczenia danych do klienta na 1000 próbek. Wykresy utworzyliśmy za pomocą pyplota.






Histogram wariant: 3 klientów, 2 rdzenie, bez obciążenia
Klient 0
Klient 1
Klient 2
Liczność
Wartość
0
100
200
300
400
500
600
700
0
20000
40000
60000
80000
100000
120000
140000


Histogram wariant: 1 klient, 4 rdzenie, bez obciążenia
Klient 0
Liczność
Wartość
0
200
400
600
800
1000
20
30
40
50
60

5. Aktywne oczekiwanie

Porównaliśmy wyniki zmodyfikowanego programu w Wariancie 3. - 3 klientów, 2 rdzenie, brak obciążenia. Do programu dodaliśmy piąty parametr, który wybierał wersję programu (0 - tylko klient 0 aktywnie oczekiwał, 1 - wszyscy klienci aktywnie oczekiwali.