

Wstęp:

Rozwiązania zadania znajdują się w folderze lab4. Zmodyfikowany plik źródłowy do zadania 5 to plik cw4_openwtr_pkg/cwicz4mak/cw4b.c. Plik źródłowy do zadania 6 to cw4_openwtr_pkg/cwicz4mak/cw4a.c

Zad 1:

Pobrano pakiet openWRT z przygotowanym do zajęć programem spod linku:
https://moodle.usos.pw.edu.pl/pluginfile.php/241562/mod_folder/content/0/skps_lab4_student.tar.xz?forcedownload=1
Zbudowano program na komputerze host i zobaczono, że działa.

Zad 2:

Następnie skompilowano pakiet i zainstalowano go na OpenWRT.

Zad 3:

Zmodyfikowano liczbę dostępnych rdzeni i sprawdzono poprawność konfiguracji poleceniem:
stress-ng -m 0 -t 30s & htop

Metodą inżynierską oszacowano graniczne wartości.

Wariant 1: 200000

Wariant 2: 350000

Wariant 3: 700000

Wariant 4: 800000

By uzyskać pełne obciążenie wykonano w tle komendę:

stress-ng --matrix 0 -t 1m

Działający jeden rdzeń

```
stress-ng: info: [1597] successful run completed in 60.02s (1 min, 0.02 secs)
CPU[|||||||||||||||||||||||||||||100.0%] Tasks: 20, 0 thr; 1 running
Mem[||| 26.0M/3.69G] Load average: 1.81 0.66 0.24
Swp[ 0K/0K] Uptime: 00:03:06
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1611	root	20	0	15412	1156	960	R	43.8	0.0	0:02.23	stress-ng-matrix
1608	root	20	0	15412	1200	1004	R	43.1	0.0	0:20.10	stress-ng-matrix
1610	root	20	0	1656	1408	1040	R	0.7	0.0	0:00.07	htop
1	root	20	0	1608	1012	824	S	0.0	0.0	0:00.27	/sbin/procd
120	ubus	20	0	1196	648	600	S	0.0	0.0	0:00.03	/sbin/ubusd
121	root	20	0	1228	1016	960	S	0.0	0.0	0:00.02	/bin/ash --login
122	root	20	0	816	468	432	S	0.0	0.0	0:00.00	/sbin/askfirst /u
300	logd	20	0	1180	580	536	S	0.0	0.0	0:00.01	/sbin/logd -S 64
352	root	20	0	2124	1204	1012	S	0.0	0.0	0:00.00	/sbin/rpcd -s /va
491	dnsmasq	20	0	1356	976	872	S	0.0	0.0	0:00.01	/usr/sbin/dnsmasq
553	root	20	0	1012	544	504	S	0.0	0.0	0:00.00	/usr/sbin/dropbea
651	root	20	0	4316	728	672	S	0.0	0.0	0:00.00	/usr/sbin/hostapd
652	root	20	0	4316	784	728	S	0.0	0.0	0:00.00	/usr/sbin/wpa_sup
712	root	20	0	1796	1200	992	S	0.0	0.0	0:00.02	/sbin/netifd
769	root	20	0	1428	728	676	S	0.0	0.0	0:00.00	/usr/sbin/odhcpd
886	root	20	0	3936	1888	1744	S	0.0	0.0	0:00.00	/usr/sbin/uhttpd
1033	root	20	0	1220	536	500	S	0.0	0.0	0:00.00	udhcpd -p /var/ru

Pełne obciążenie na 1 rdzeń

```
root@OpenWrt:~/skps22_sztanga_stelmakh/lab4# stress-ng --matrix 0 -t 1m & cw4a 3 100 10000 200000
stress-ng: info: [1594] successful run completed in 30.00s

Client: 2, nsmp=100, del=200000
Client: 1, nsmp=100, del=200000
Client: 0, nsmp=100, del=200000
Sample 0, client 1, delivery time: 438
Sample 0, client 2, delivery time: 569
Sample 0, client 0, delivery time: 667
stress-ng: info: [1603] dispatching hogs: 1 matrix
Sample 1, client 2, delivery time: 1020
Sample 1, client 1, delivery time: 1163
Sample 1, client 0, delivery time: 1240
Sample 2, client 2, delivery time: 6597
Sample 2, client 1, delivery time: 6889
Sample 2, client 0, delivery time: 7781
Sample 3, client 1, delivery time: 4128
Sample 3, client 2, delivery time: 6117
Sample 4, client 1, delivery time: 3409
Sample 3, client 0, delivery time: 14869
Sample 4, client 2, delivery time: 15303
Sample 4, client 0, delivery time: 18986
Sample 5, client 2, delivery time: 11733
Sample 5, client 1, delivery time: 13060
```

Uruchomienie bez obciążenia dla 2 rdzeni

```
root@OpenWrt:~/skps22_sztanga_stelmakh/lab4# cw4a 3 100 10000 700000
Client: 0, nsmp=100, del=700000
Client: 1, nsmp=100, del=700000
Client: 2, nsmp=100, del=700000
Sample 0, client 0, delivery time: 31
Sample 0, client 1, delivery time: 129
Sample 0, client 2, delivery time: 240
Sample 1, client 1, delivery time: 65
Sample 1, client 0, delivery time: 397
Sample 1, client 2, delivery time: 8934
Sample 2, client 1, delivery time: 15
Sample 2, client 2, delivery time: 63
Sample 2, client 0, delivery time: 4499
Sample 3, client 1, delivery time: 196
Sample 3, client 0, delivery time: 10807
Sample 4, client 1, delivery time: 59
Sample 3, client 2, delivery time: 14126
Sample 4, client 0, delivery time: 12846
Sample 4, client 2, delivery time: 20616
Sample 5, client 1, delivery time: 9161
Sample 5, client 0, delivery time: 12155
Sample 6, client 1, delivery time: 8334
Sample 6, client 0, delivery time: 11783
Sample 5, client 2, delivery time: 29748
```

Działające tylko 2 z 4 rdzeni

0

[|||||||||||||||||||||100.0%]

1

[|||||||||||||||||||||100.0%]

Mem

[|||36.4M/3.69G]

Swp

[0K/0K]

Tasks: 31, 0 thr; 2 running

Load average: 3.16 0.78 0.26

Uptime: 00:01:37

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1601	root	20	0	15416	1128	932	R	50.0	0.0	0:15.53	stress-ng-matrix
1602	root	20	0	15416	1128	932	R	50.0	0.0	0:15.70	stress-ng-matrix
1609	root	20	0	15416	1132	932	R	50.0	0.0	0:09.18	stress-ng-matrix
1610	root	20	0	15416	1132	932	R	50.0	0.0	0:09.26	stress-ng-matrix
1616	root	20	0	15416	1168	972	R	50.0	0.0	0:05.99	stress-ng-matrix
1617	root	20	0	15416	1168	972	R	50.0	0.0	0:06.10	stress-ng-matrix
1624	root	20	0	15416	1092	888	R	50.0	0.0	0:03.85	stress-ng-matrix
1627	root	20	0	1672	1428	1036	R	50.0	0.0	0:00.04	htop
1628	root	20	0	15416	1092	892	R	50.0	0.0	0:00.05	stress-ng-matrix
1629	root	20	0	15416	1092	892	R	50.0	0.0	0:00.04	stress-ng-matrix
1	root	20	0	1608	1016	828	S	0.0	0.0	0:00.27	/sbin/procd
126	ubus	20	0	1196	656	608	S	0.0	0.0	0:00.03	/sbin/ubusd
127	root	20	0	1228	1020	960	S	0.0	0.0	0:00.01	/bin/ash --login
128	root	20	0	816	476	440	S	0.0	0.0	0:00.00	/sbin/askfirst /u
307	logd	20	0	1180	576	532	S	0.0	0.0	0:00.01	/sbin/logd -S 64
359	root	20	0	2124	1204	1012	S	0.0	0.0	0:00.00	/sbin/rpcd -s /va

działające 4 rdzenie

0[|||||||||||||||||100.0%]

1[|||||||||||||||||100.0%]

2[|||||||||||||||||100.0%]

3[|||||||||||||||||100.0%]

Mem[|||28.4M/3.69G]

Swp[0K/0K]

Tasks: 21, 0 thr; 4 running

Load average: 0.38 0.09 0.03

Uptime: 00:01:36

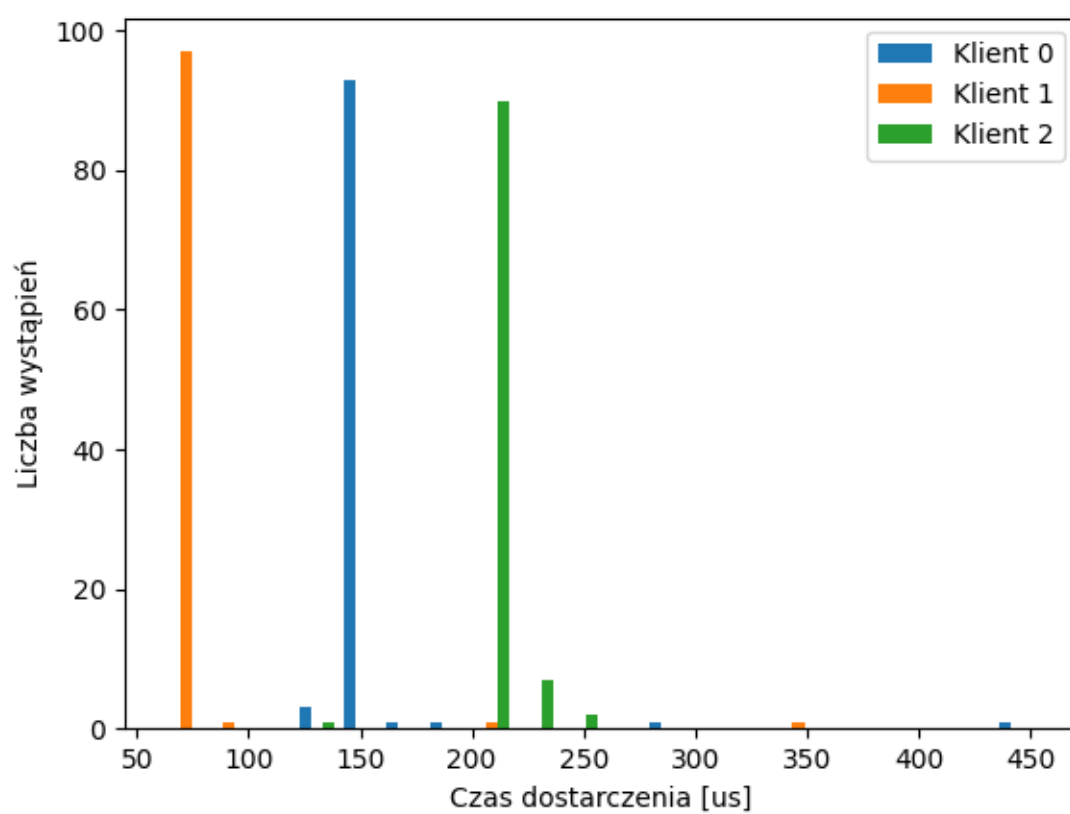
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1623	root	20	0	15420	1148	948	R	99.8	0.0	0:06.15	stress-ng-matrix
1624	root	20	0	15420	1148	948	R	99.8	0.0	0:06.15	stress-ng-matrix
1625	root	20	0	15420	1148	948	R	99.8	0.0	0:06.15	stress-ng-matrix
1626	root	20	0	15420	1148	948	R	99.2	0.0	0:06.09	stress-ng-matrix
1622	root	20	0	1672	1428	1036	R	0.7	0.0	0:00.08	htop
1	root	20	0	1608	1020	828	S	0.0	0.0	0:00.27	/sbin/procd
140	ubus	20	0	1196	652	604	S	0.0	0.0	0:00.02	/sbin/ubusd
141	root	20	0	1228	1020	960	S	0.0	0.0	0:00.01	/bin/ash --login
142	root	20	0	816	472	436	S	0.0	0.0	0:00.00	/sbin/askfirst /u
322	logd	20	0	1180	580	536	S	0.0	0.0	0:00.00	/sbin/logd -S 64
374	root	20	0	2124	1208	1012	S	0.0	0.0	0:00.01	/sbin/rpcd -s /va
514	dnsmasq	20	0	1356	976	872	S	0.0	0.0	0:00.01	/usr/sbin/dnsmasq
576	root	20	0	1012	536	496	S	0.0	0.0	0:00.00	/usr/sbin/dropbea
674	root	20	0	4316	708	652	S	0.0	0.0	0:00.00	/usr/sbin/hostapd

Uruchomienie bez obciążenia dla 4 rdzeni

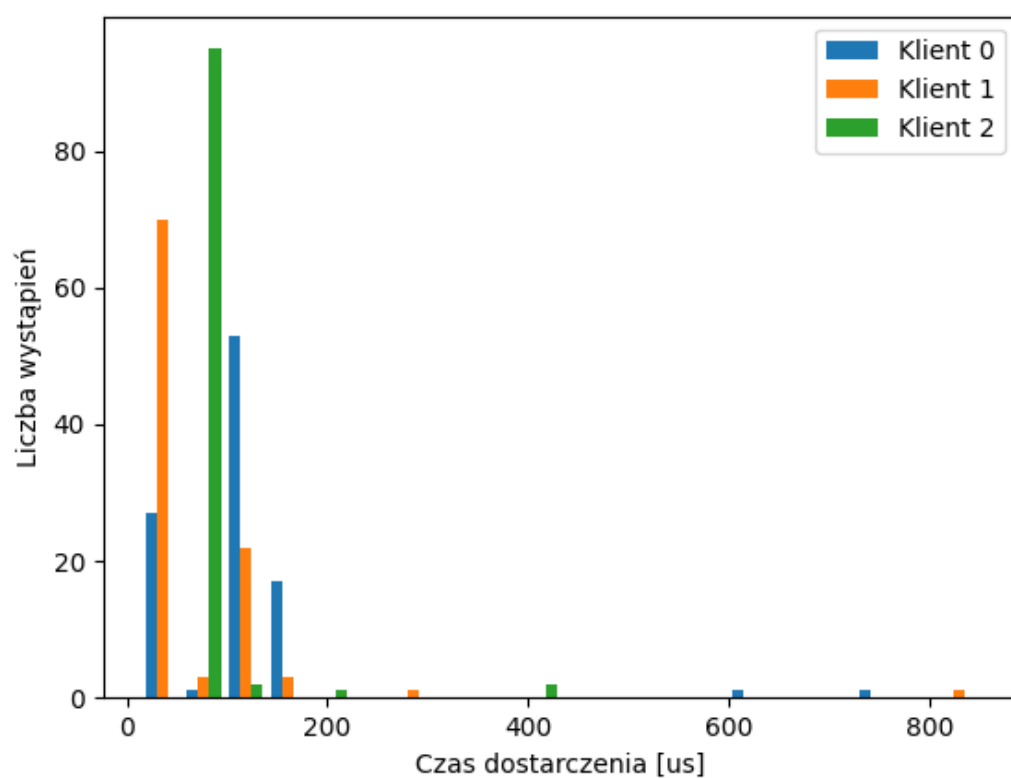
```
root@OpenWrt:~/skps22_sztanga_stelmakh/lab4# cw4a 1 100 10000 800000
Client: 0, nsmp=100, del=800000
Sample 0, client 0, delivery time: 27
Sample 1, client 0, delivery time: 656
Sample 2, client 0, delivery time: 1974
Sample 3, client 0, delivery time: 3642
Sample 4, client 0, delivery time: 4264
Sample 5, client 0, delivery time: 5932
Sample 6, client 0, delivery time: 7595
Sample 7, client 0, delivery time: 9044
Sample 8, client 0, delivery time: 9954
Sample 9, client 0, delivery time: 11395
Sample 10, client 0, delivery time: 13178
Sample 11, client 0, delivery time: 13059
Sample 12, client 0, delivery time: 13709
Sample 13, client 0, delivery time: 14312
Sample 14, client 0, delivery time: 15481
Sample 15, client 0, delivery time: 16030
Sample 16, client 0, delivery time: 17698
Sample 17, client 0, delivery time: 18372
Sample 18, client 0, delivery time: 19110
Sample 19, client 0, delivery time: 20455
```

Zadanie 4:

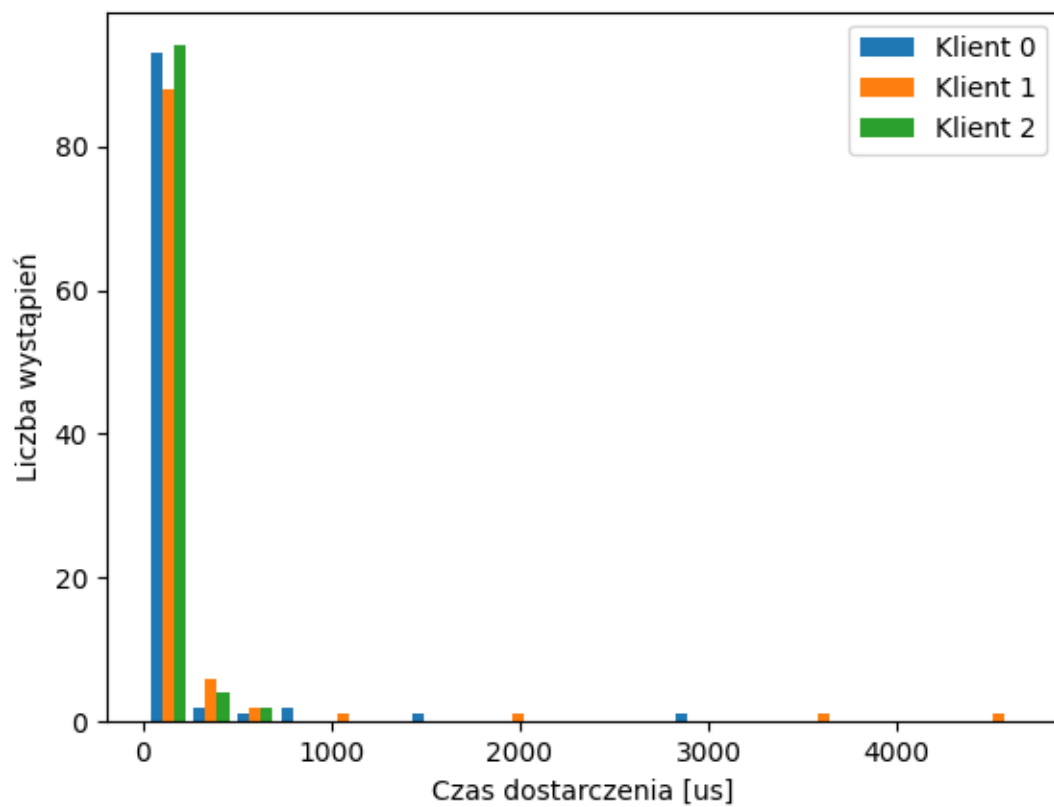
Wykonano testy dla wszystkich wariantów zadania 3 używając połowy wyznaczonej wartości granicznej czasu przetwarzania. Wyniki umieszczono w histogramie. Do wygenerowania histogramów użyto pythona z biblioteką matplotlib. Przedział osiągniętych wartości został podzielony na 20 części. Słupki dla wariantów 1, 2 i 3 są 3 razy węższe niż dla wariantu 4, by klienci się nawzajem nie zasłaniaли.



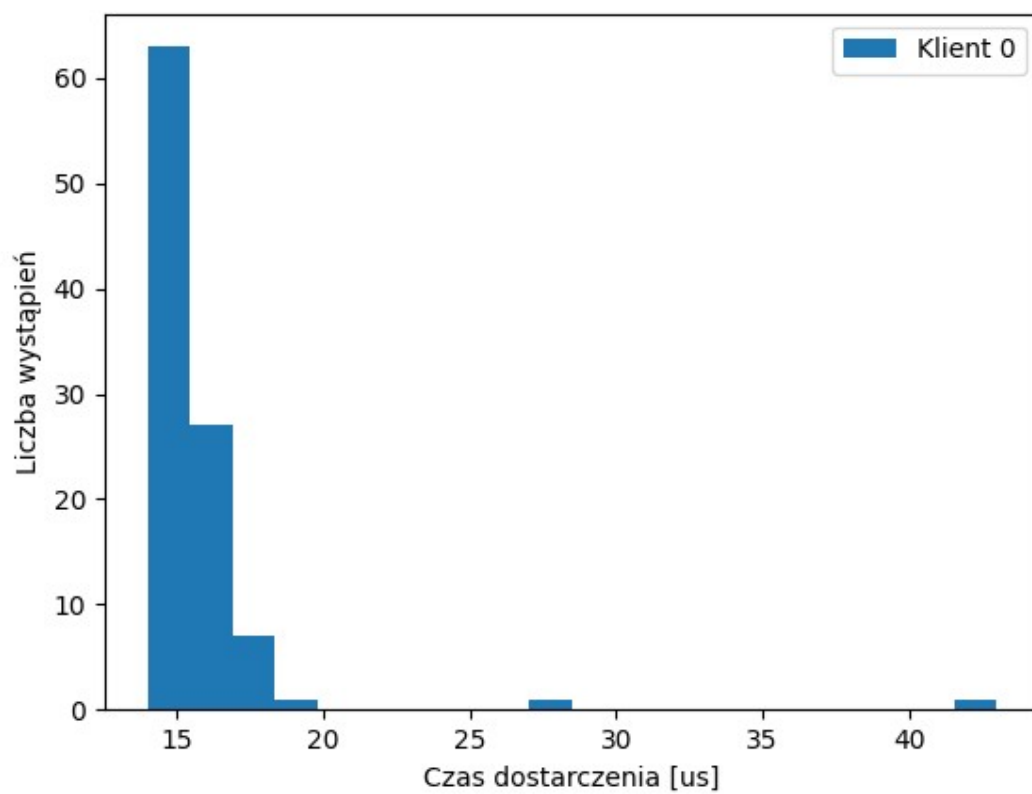
Wariant 1



Wariant 2



Wariant 3

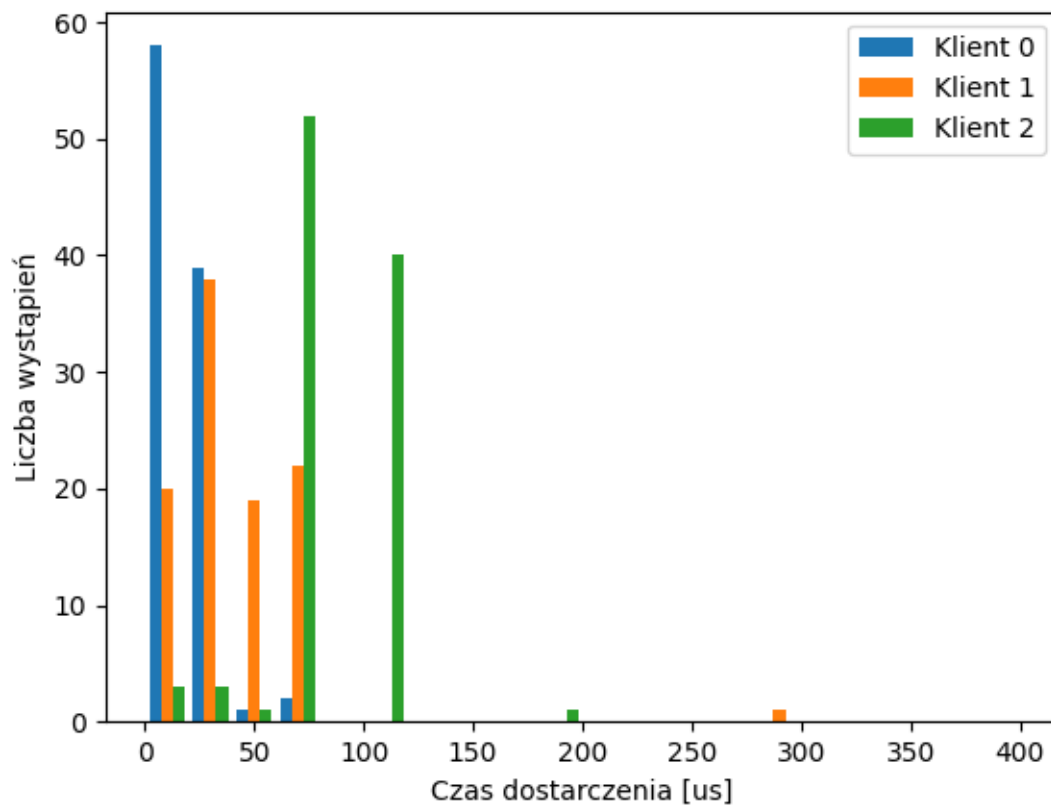


Wariant 4

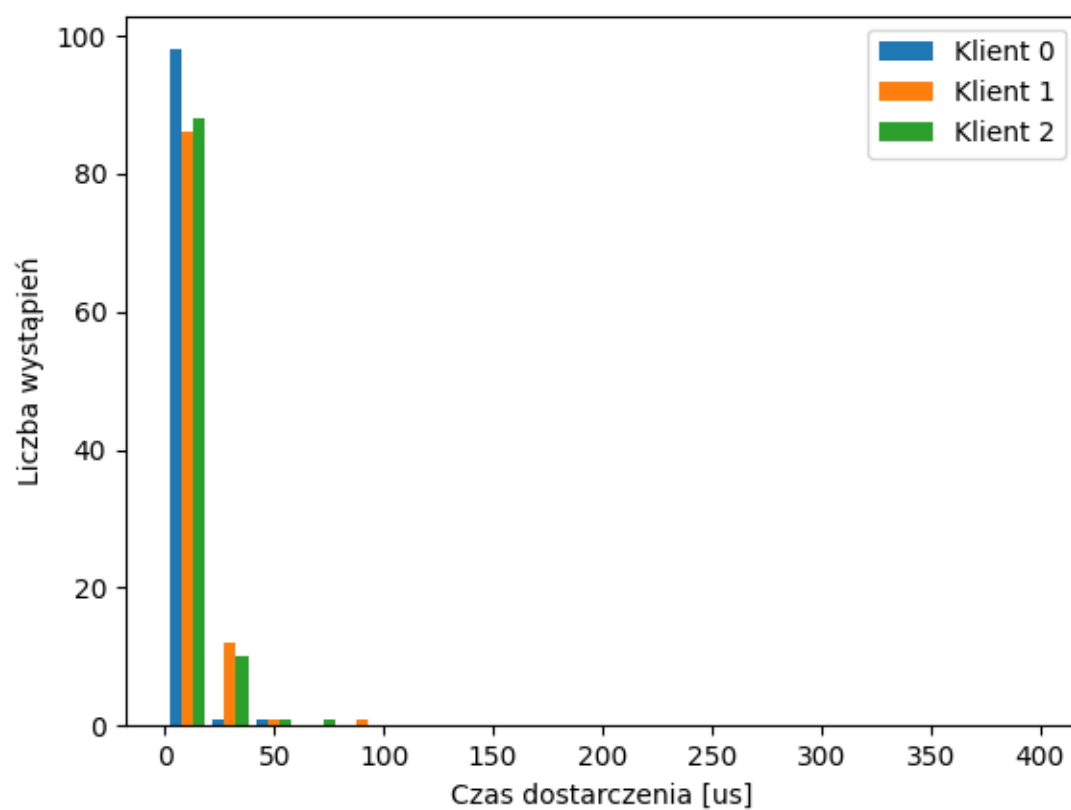
Jak widać histogramy są najgęstsze z lewej strony. Te niewielkie słupki po prawej oznaczają pojedyncze duże czasy oczekiwania.

Zad 5:

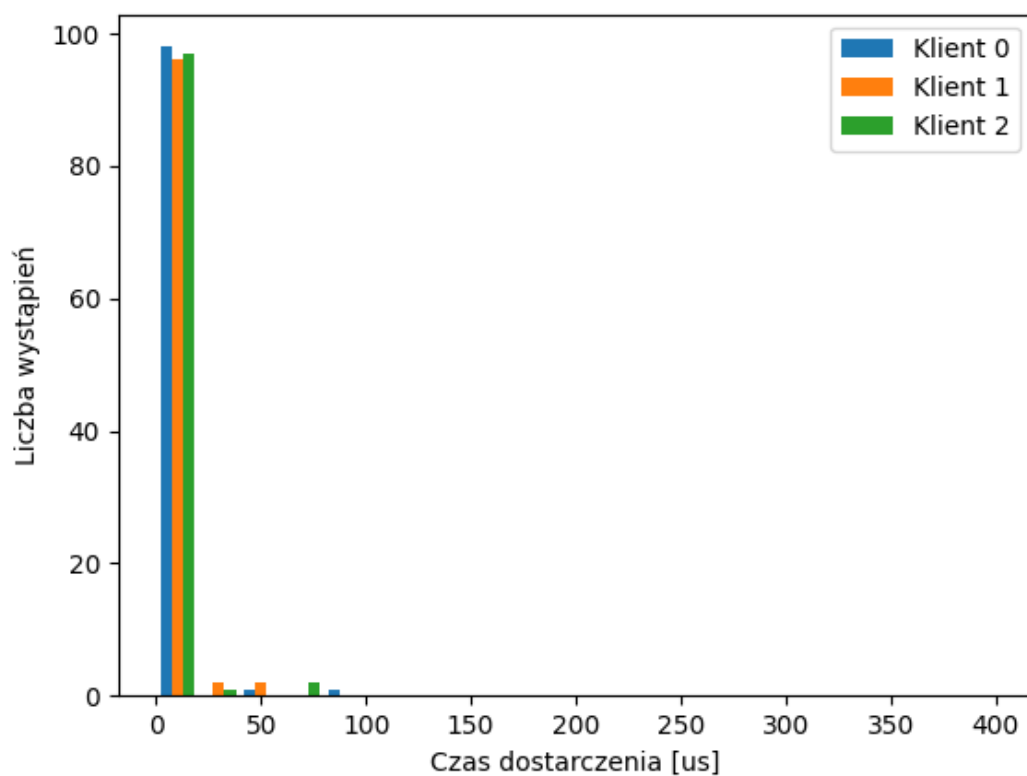
Przeprowadzono nowe doświadczenie mające sprawdzić wpływ zmiany usypiania procesu na aktywne oczekiwanie. Przeprowadzono doświadczenie dla 3 klientów, okresu próbkowania 20000 i 4 rdzeni bez dodatkowego obciążenia. Wygenerowano histogramy używając tego samego skryptu w pythonie.



Usypianie procesu



Aktywnie oczekujący klient 0 i usypiani klienci 1 i 2.



Aktywnie oczekujący wszyscy klienci

Aktywne oczekiwanie sprawiło, że klienci krócej oczekują na odpowiedź. Dzieje się tak gdyż czas nie jest tracony na przełączaniu procesów.

```
// Check for active wait env variables
char active_wait = 0;
if (getenv("ACTIVE_WAIT_ALL")){
    active_wait = 1;
    printf("ACTIVE_WAIT_ALL = %d\n", active_wait);
}

if (ncli == 0 && getenv("ACTIVE_WAIT_CLI_0")) {
    active_wait = 1;
    printf("ACTIVE_WAIT_CLI_0 = %d\n", active_wait);
}

} else {
    if (active_wait)
        while (rbuf->head != rbuf->tail[ncli]);
    else
        pthread_cond_wait(&rbuf->cvar, &rbuf->cvar_lock);
    pthread_mutex_unlock(&rbuf->cvar_lock);
}
```

Zmiany w kodzie

Zadanie 6:

Ta implementacja nie jest poprawna. Poprawna implementacja opierałaby się na timerach.

```
timer = timerfd_create(CLOCK_MONOTONIC, 0);
struct itimerspec timerdata;
timerdata.it_interval.tv_sec = 0;
timerdata.it_interval.tv_nsec = 1000* udelsmp;
timerdata.it_value.tv_sec = 20;
timerdata.it_value.tv_nsec = 0;

timerfd_settime(timer, 0, &timerdata, NULL);
```

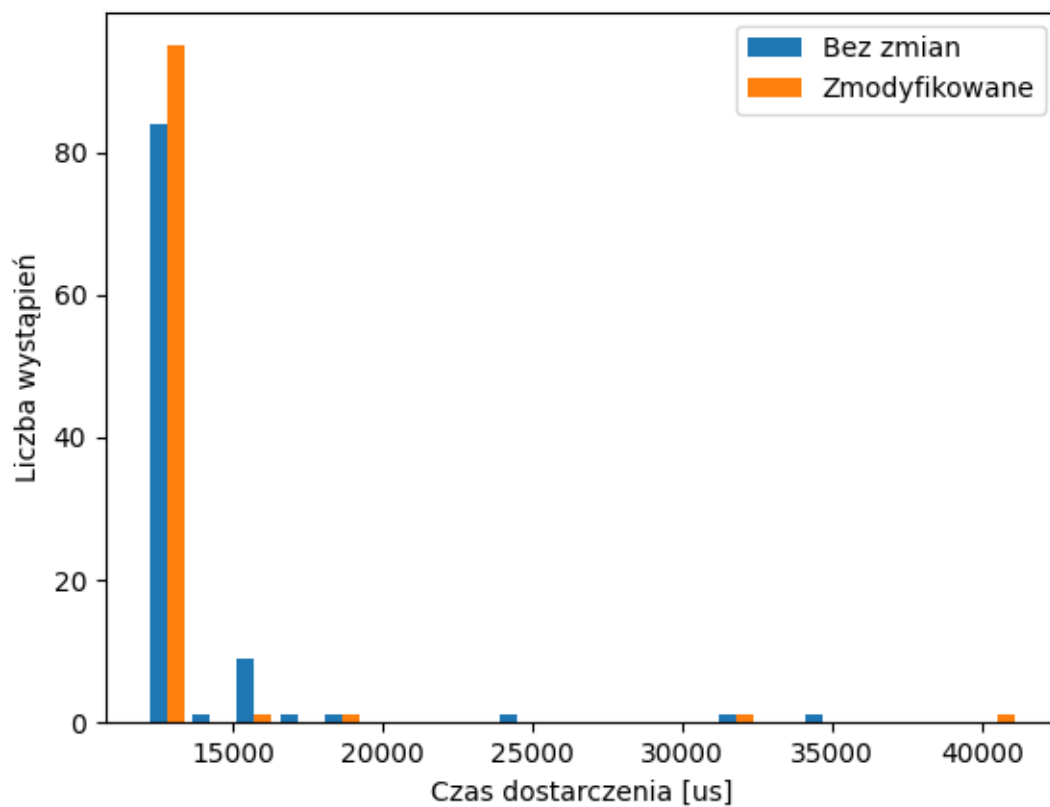
```

timerfd_gettime(&timer, 0, &prev_smp_time, NULL);

//Now we can start generating data and delivering data
unsigned long prev_smp_time = 0;
for(i=0; i<nsmp; i++) {
    int j;
    unsigned long smp_time;
    //usleep(udelsmp);
    uint64_t exp = 0;
    read(timer, &exp, sizeof(uint64_t));
    //Prepare data to be inserted

```

Zmiany w kodzie



Histogram porównujący timery