In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

In [48]:
```python
data = pd.read_csv("results.csv")
data
```

Out[48]:

	Bufsize	Prod_or_cons	Size	PC_config	Is_fair	Randomization	Accesses
0	10000	producer	1	100+100	False	irregular	44
1	10000	producer	2	100+100	False	irregular	31
2	10000	producer	3	100+100	False	irregular	30
3	10000	producer	4	100+100	False	irregular	29
4	10000	producer	5	100+100	False	irregular	23
...	...	...	...	...	...	...	...
880026	100000	consumer	49996	1000+1000	True	uniform	40
880027	100000	consumer	49997	1000+1000	True	uniform	32
880028	100000	consumer	49998	1000+1000	True	uniform	41
880029	100000	consumer	49999	1000+1000	True	uniform	34
880030	100000	consumer	50000	1000+1000	True	uniform	0

880031 rows × 7 columns

In [3]:
```python
data.groupby(["Bufsize", "PC_config", "Is_fair", "Randomization", "Prod_or_cons"]).count()
```

Out[3]:

Bufsize	PC_config	Is_fair	Randomization	Prod_or_cons	Size	Accesses
10000	**100+100**	**False**	**irregular**	**consumer**	5001	5001
				producer	5000	5000
			uniform	**consumer**	5001	5001
				producer	5001	5001
		True	**irregular**	**consumer**	5001	5001
				producer	5001	5001
			uniform	**consumer**	5001	5001
				producer	5001	5001
	1000+1000	**False**	**irregular**	**consumer**	5001	5001
				producer	5001	5001
			uniform	**consumer**	5001	5001
				producer	5001	5001
		True	**irregular**	**consumer**	5001	5001
				producer	5001	5001
			uniform	**consumer**	5001	5001
				producer	5001	5001
100000	**100+100**	**False**	**irregular**	**consumer**	50001	50001
				producer	50001	50001
			uniform	**consumer**	50001	50001
				producer	50001	50001
		True	**irregular**	**consumer**	50001	50001
				producer	50001	50001
			uniform	**consumer**	50001	50001
				producer	50001	50001
	1000+1000	**False**	**irregular**	**consumer**	50001	50001
				producer	50001	50001
			uniform	**consumer**	50001	50001
				producer	50001	50001
		True	**irregular**	**consumer**	50001	50001
				producer	50001	50001
			uniform	**consumer**	50001	50001
				producer	50001	50001

In [4]:
```python
data.groupby(["Bufsize", "PC_config", "Is_fair", "Randomization", "Prod_or_cons", "Size", "Accesses"]).count
```

Out[4]:

Bufsize	PC_config	Is_fair	Randomization	Prod_or_cons	Size	Accesses
10000	**100+100**	**False**	**irregular**	**consumer**	**0**	**51**
					1	**42**
					2	**35**
					3	**35**
					4	**30**
...	**...**	**...**	**...**	**...**	**...**	**...**
100000	**1000+1000**	**True**	**uniform**	**producer**	**49996**	**48**
					49997	**38**
					49998	**35**
					49999	**35**
					50000	**0**

880031 rows × 0 columns

Teraz postaramy się przedstawić wszystkie konfiguracje na wykresie typu box-plot. Do przedstawienia mamy 16 konfiguracji (razy 2 bo producenci i konsumenci).
Warto zacząć od przygotowania danych oraz funkcji tworzącej wykresy.

In [13]:
```python
tmp_data = data.groupby(["Bufsize", "PC_config", "Is_fair", "Randomization", "Prod_or_cons"])
data_to_draw = []
```

In [14]:
```python
for info, group_data in tmp_data:
    data_to_draw.append((info, group_data))
```

In [108…
```python
plt.rcParams["figure.figsize"] = (15,5)

def draw_plot(config, data):
    x = []
    y = []
    for i,j in data.items():
        x.append(str(i))
        y.append(j)
    plt.scatter(x,y)
#     plt.figure(figsize=(10,5))
    plt.show()

def draw(draw_data, index, multiplier=1, print_more=False):
    print("Configuration number " + str(index+1))
    print("Buffer size: " + str(draw_data[0][0]))
    print("Type: " + str(draw_data[0][4]))
    print("PC Config: " + str(draw_data[0][1]))
    print("Is buffer fair: " + str(draw_data[0][2]))
    print("Distribution: " + str(draw_data[0][3]))
    print("Number of buckets: " + str(50 / multiplier))

    min_size = 0
    max_size = 5000
    interval_size = 100
    if draw_data[0][0] > 10000:
        interval_size = 1000
        max_size = 50000

    if interval_size == 100:
        draw_data[1]['bin'] = pd.cut(draw_data[1]['Size'], [x for x in range(min_size, max_size + 99, interv
    else:
        draw_data[1]['bin'] = pd.cut(draw_data[1]['Size'], [x for x in range(min_size, max_size + 999, inter
    a = draw_data[1].groupby('bin')["Accesses"].sum()
    a.plot.box()
    plt.show()
    if print_more:
        print()
        draw_plot(draw_data[0] ,a)
```

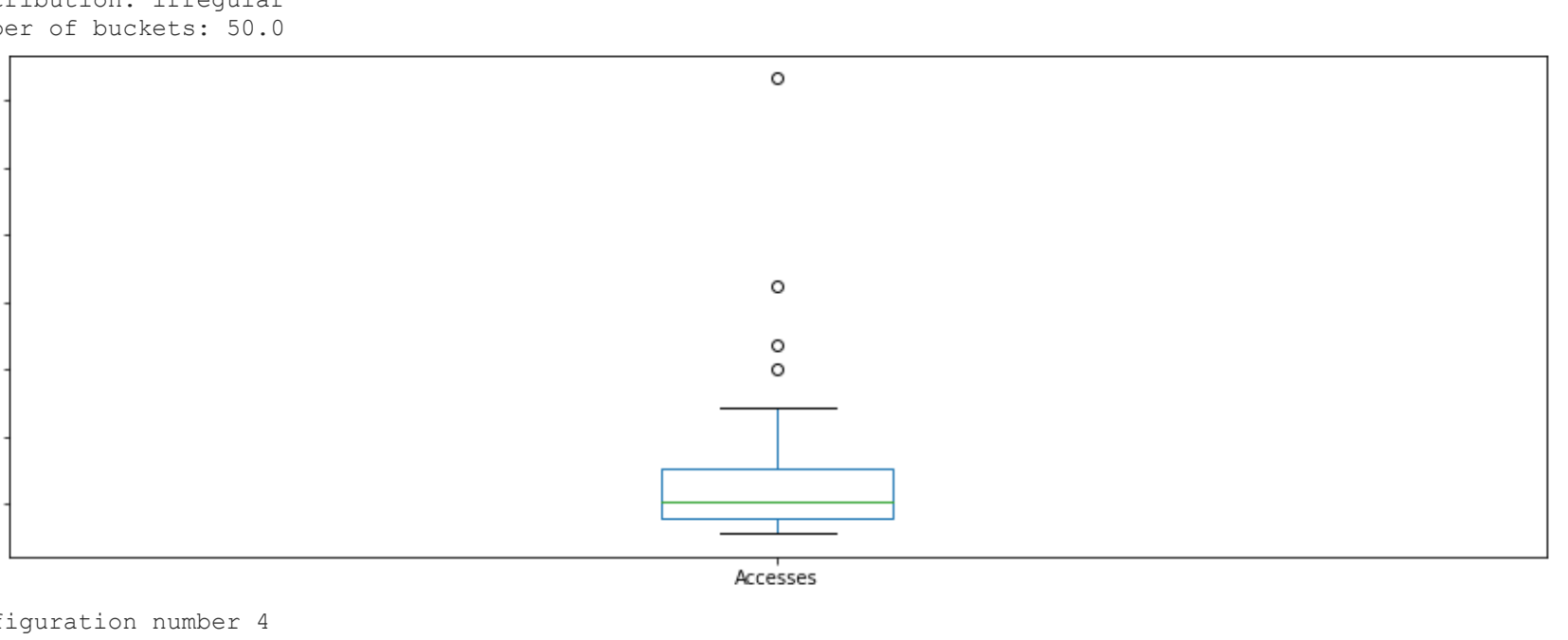
Wszystkie eksperymenty przeprowadzone są dla 50 kubełków.
Postaramy się pokazać na wykresach różnice między konfiguracjami.

1. Producenci

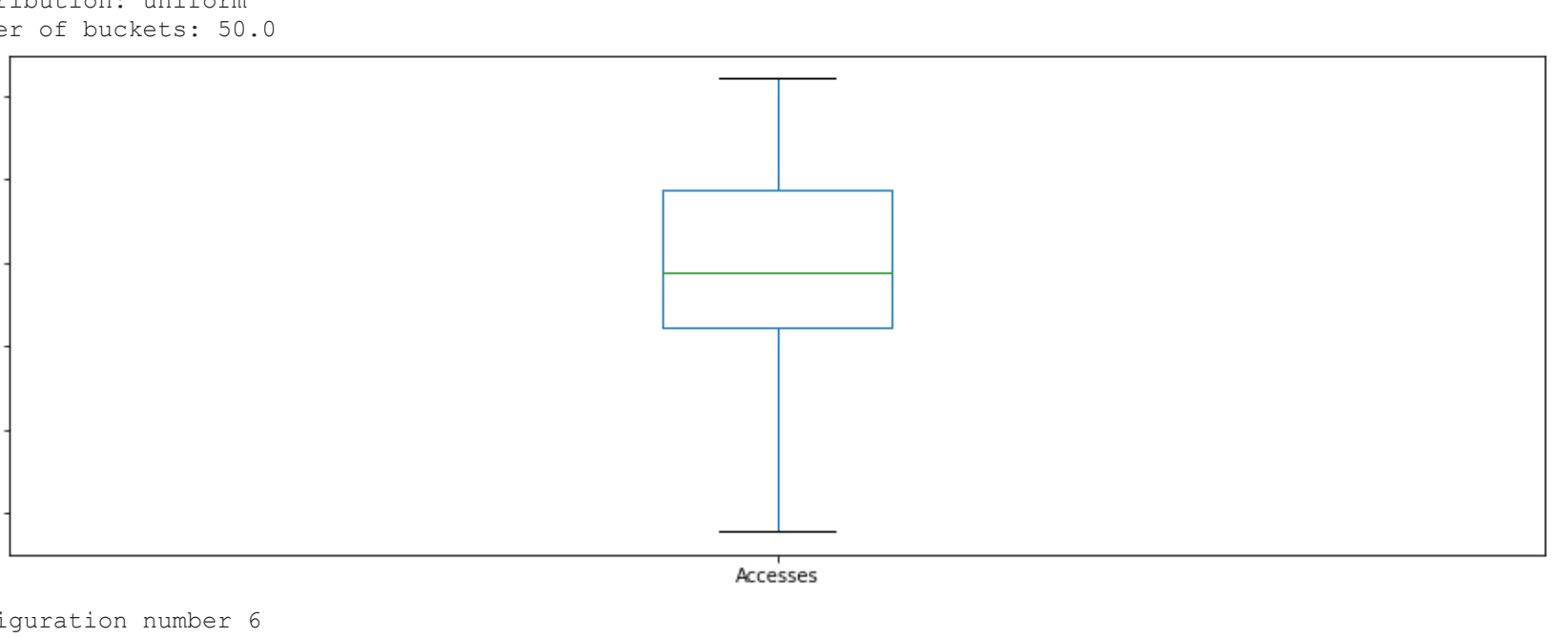
Zaczniemy od narysowania wszystkich wykresów pudełkowych dla producentów (16 wykresów).

In [103…
```python
draw(data_to_draw[1], 1)
draw(data_to_draw[3], 3)
draw(data_to_draw[5], 5)
draw(data_to_draw[7], 7)
draw(data_to_draw[9], 9)
draw(data_to_draw[11], 11)
draw(data_to_draw[13], 13)
draw(data_to_draw[15], 15)
draw(data_to_draw[17], 17)
draw(data_to_draw[19], 19)
draw(data_to_draw[21], 21)
draw(data_to_draw[23], 23)
draw(data_to_draw[25], 25)
draw(data_to_draw[27], 27)
draw(data_to_draw[29], 29)
draw(data_to_draw[31], 31)
```

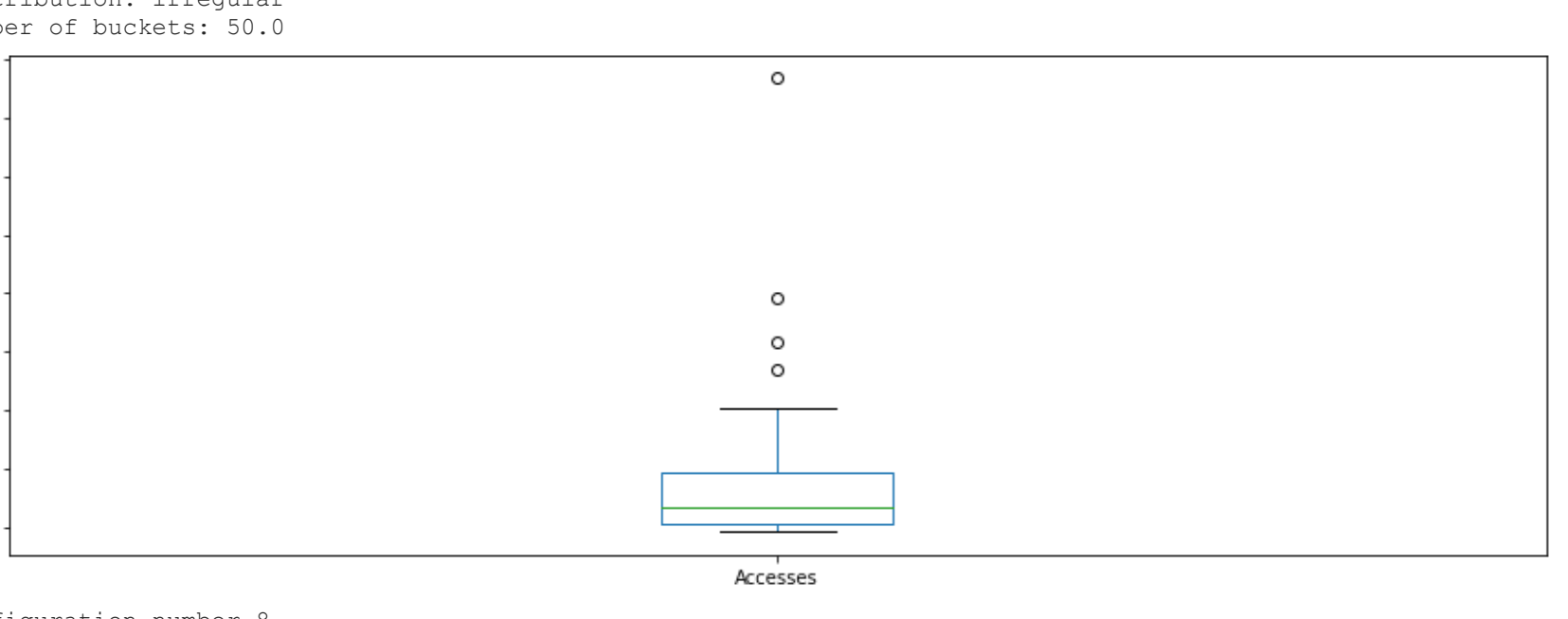
```
Configuration number 2
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



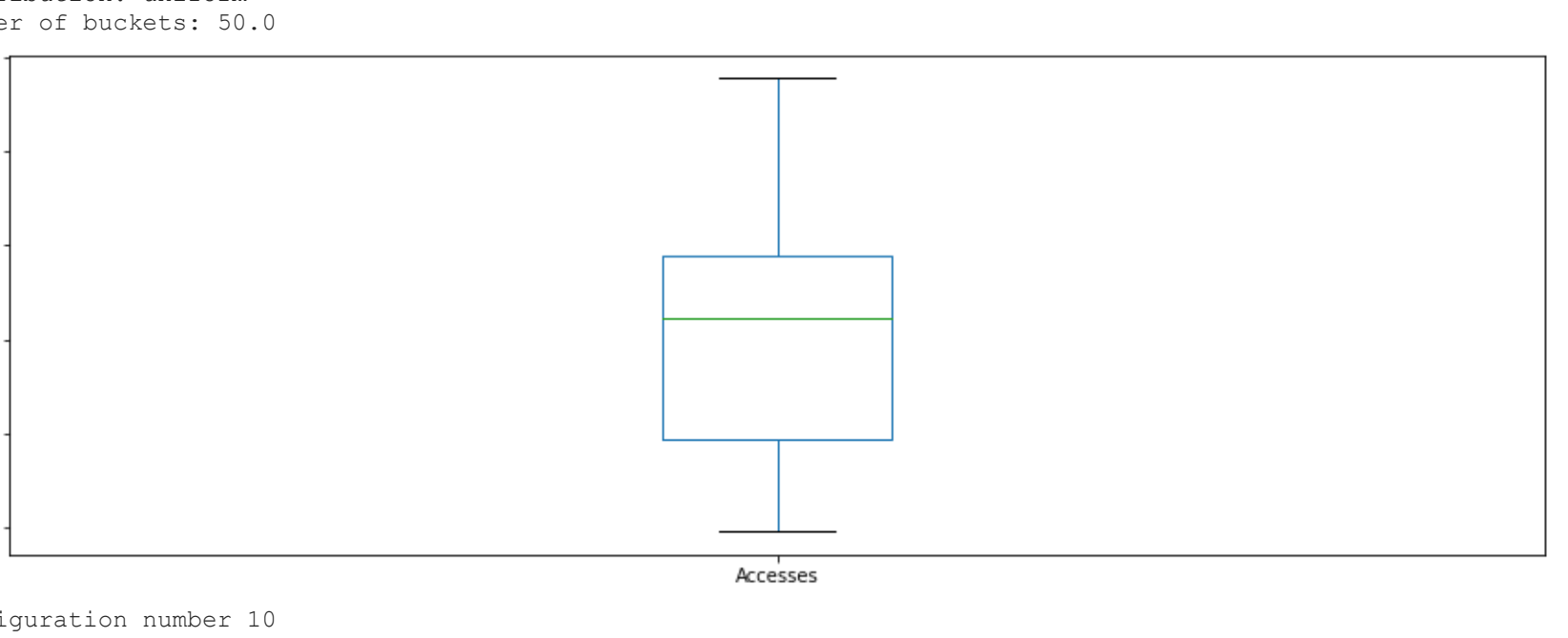
```
Configuration number 4
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



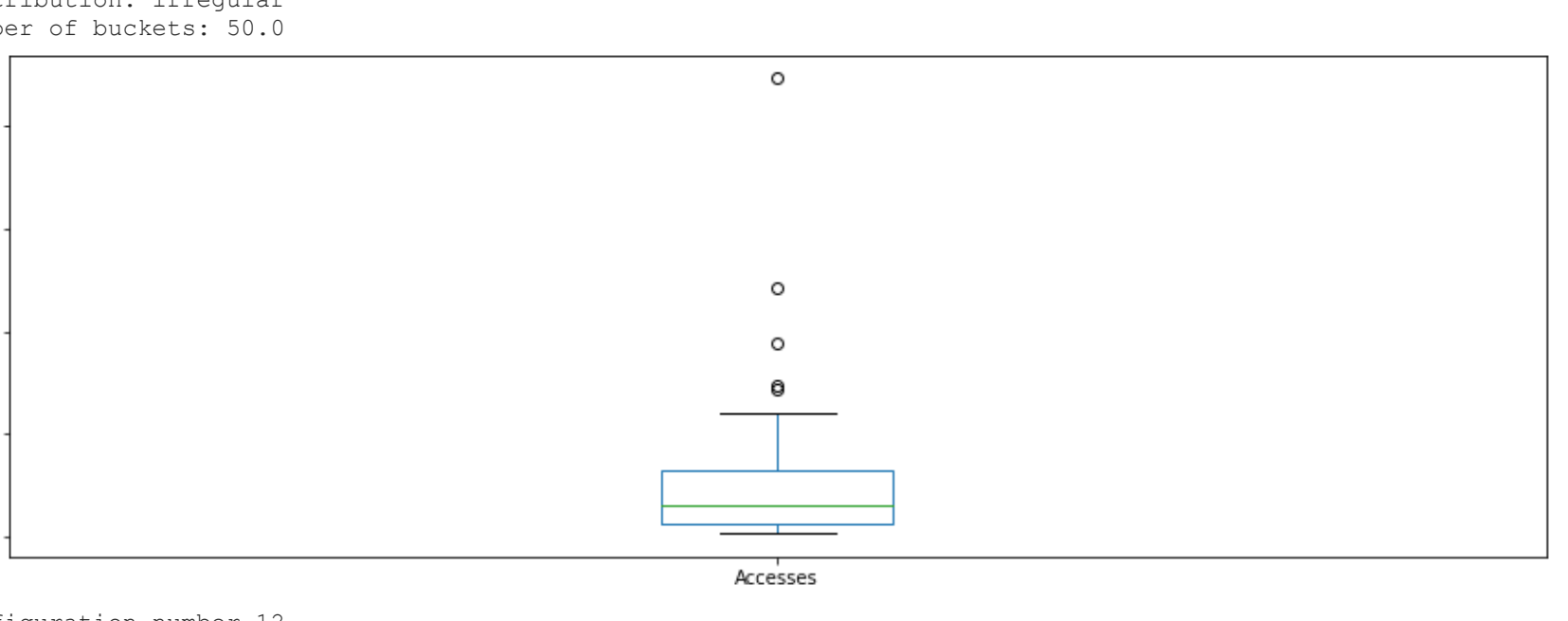
```
Configuration number 6
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



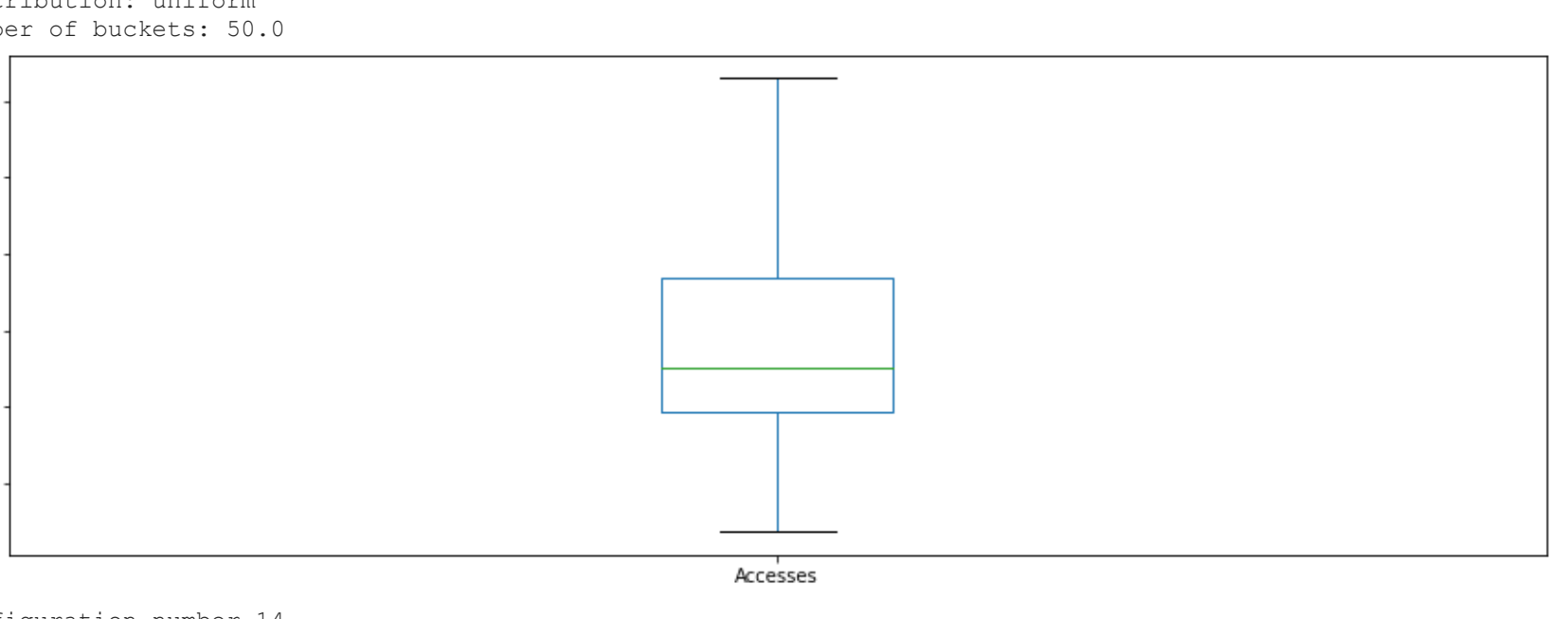
```
Configuration number 8
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



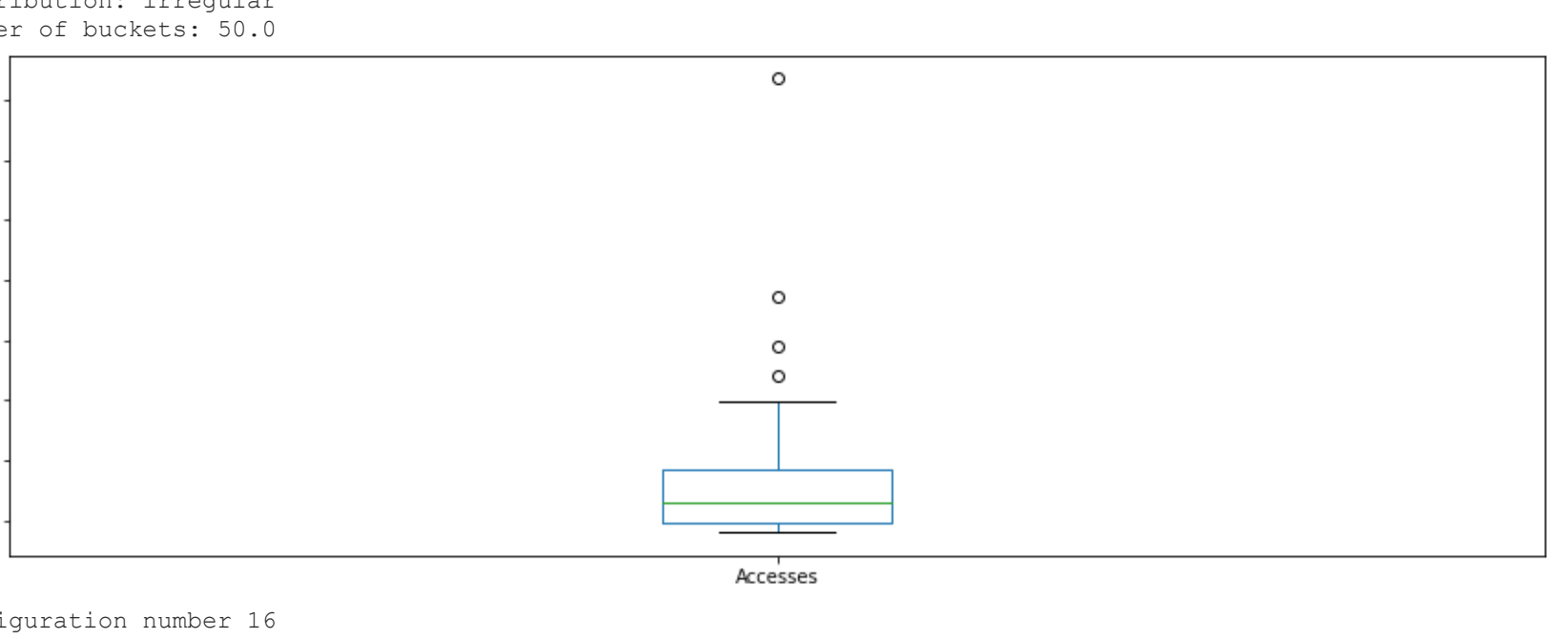
```
Configuration number 10
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



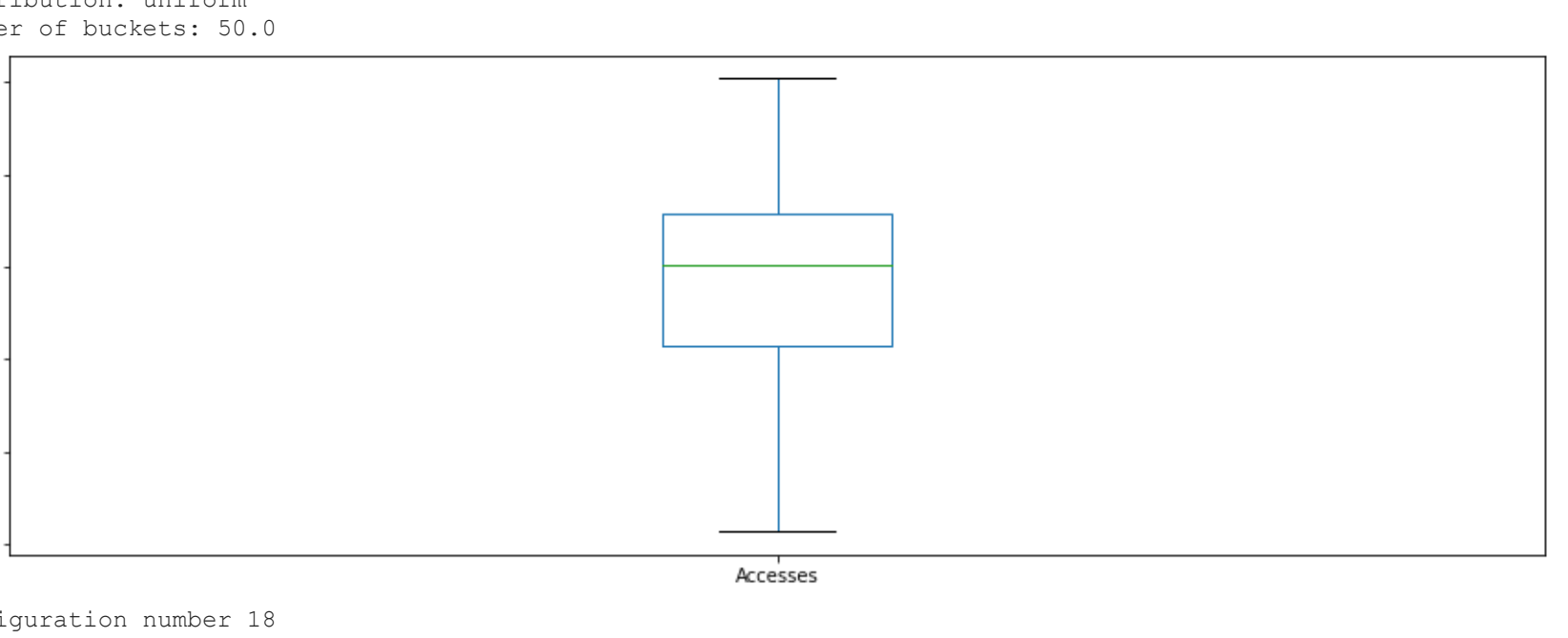
```
Configuration number 12
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



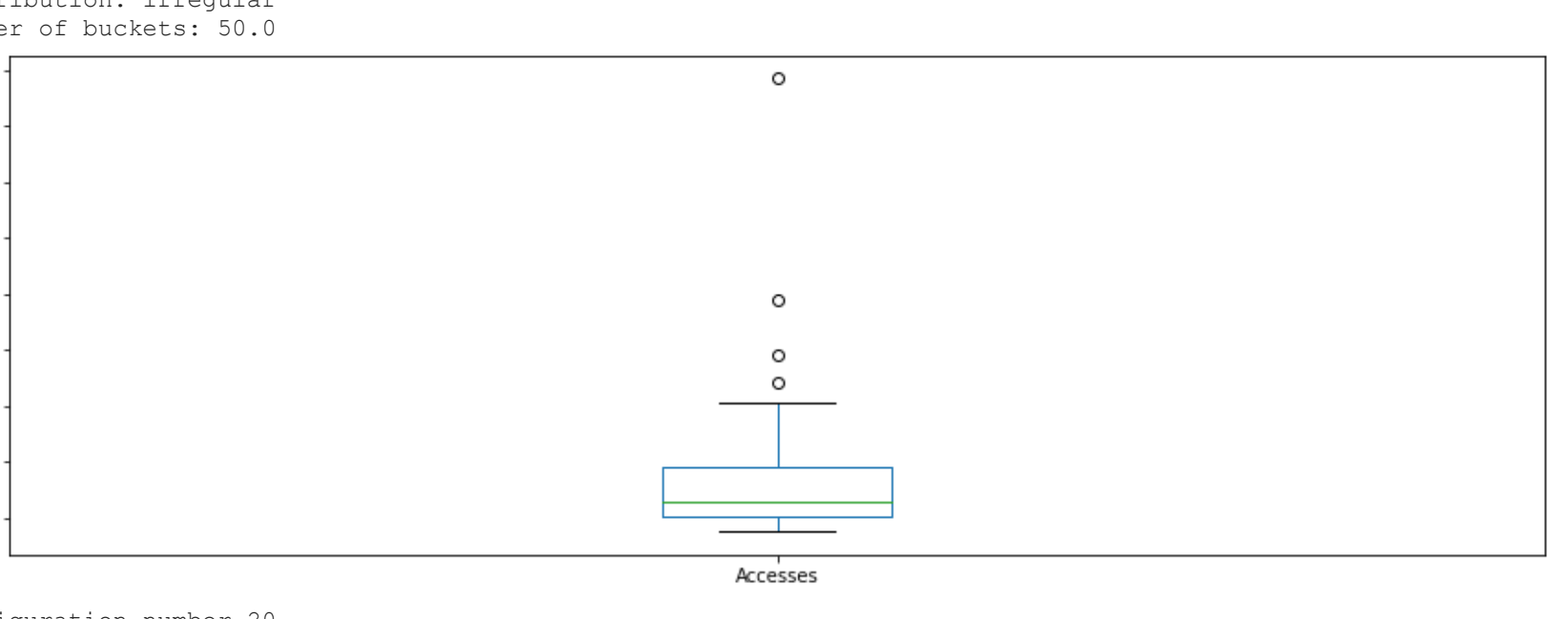
```
Configuration number 14
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



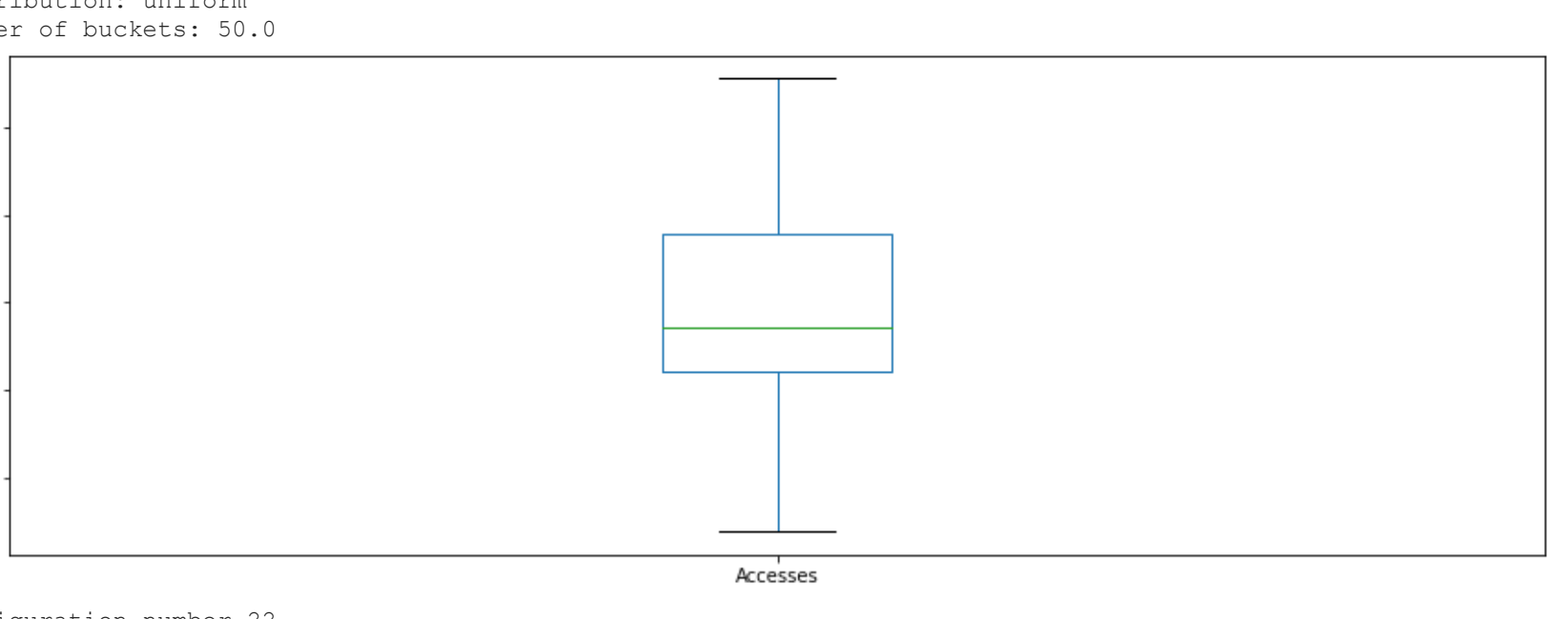
```
Configuration number 16
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



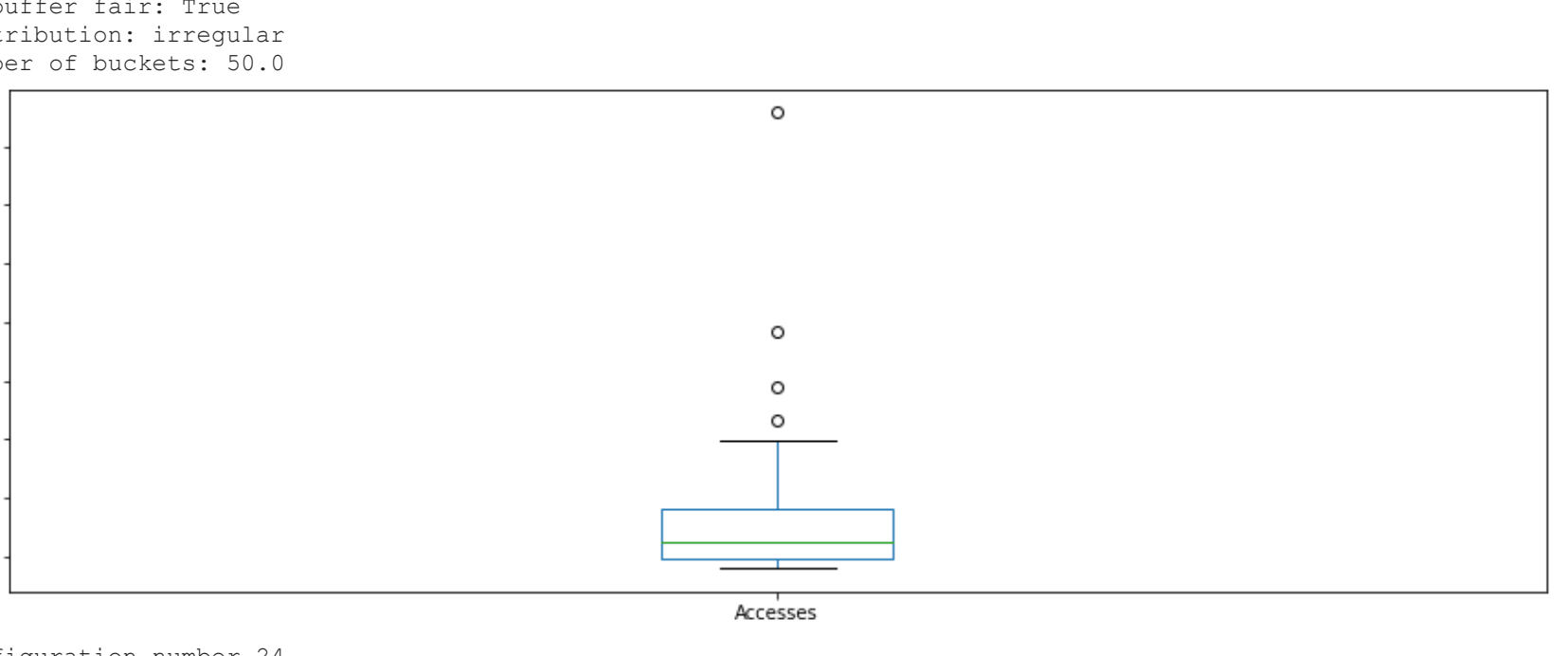
```
Configuration number 18
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



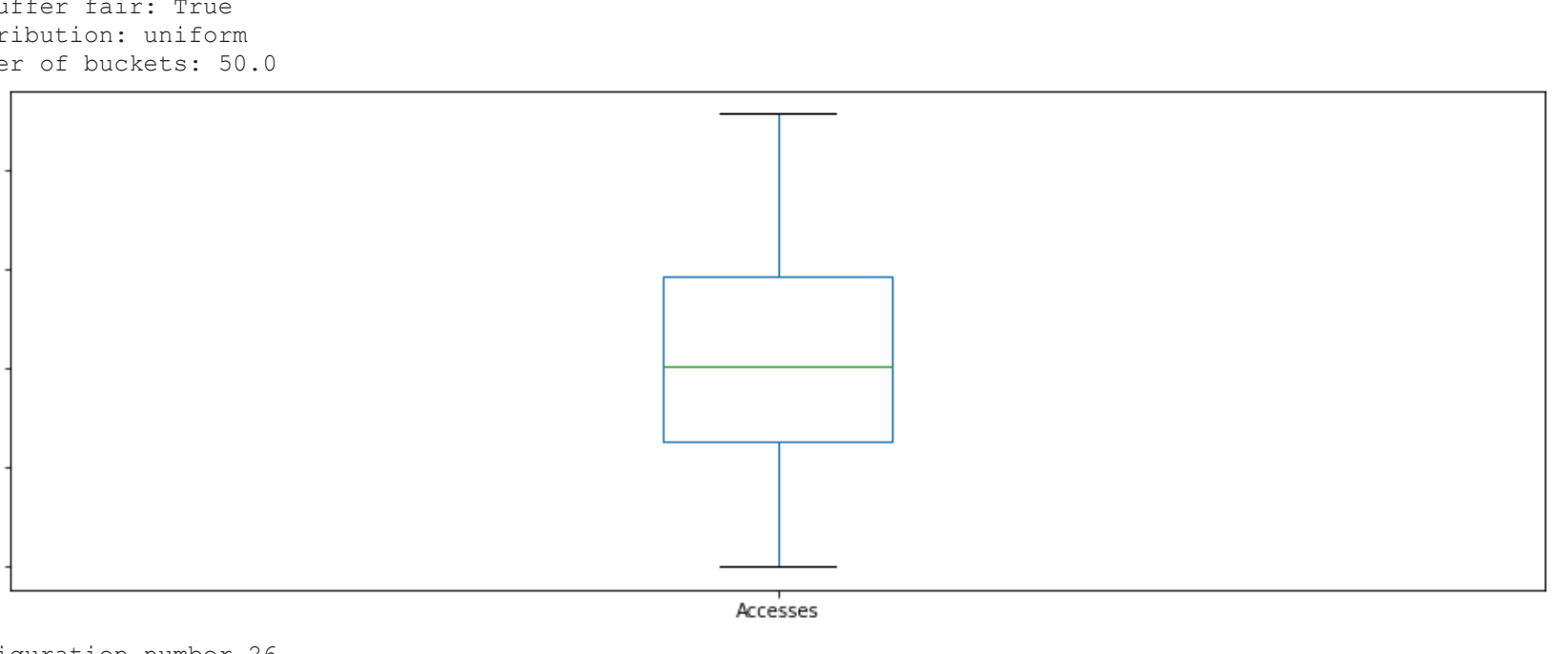
```
Configuration number 20
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



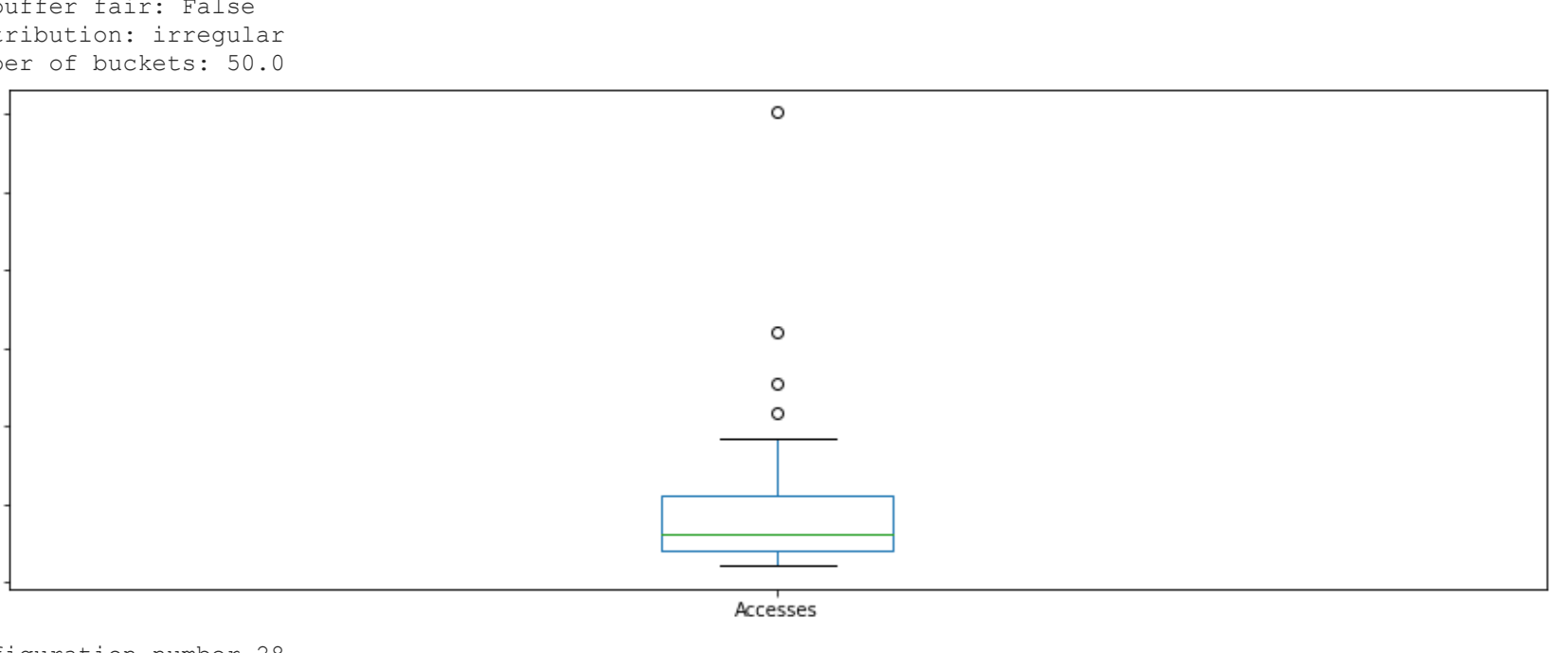
```
Configuration number 22
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



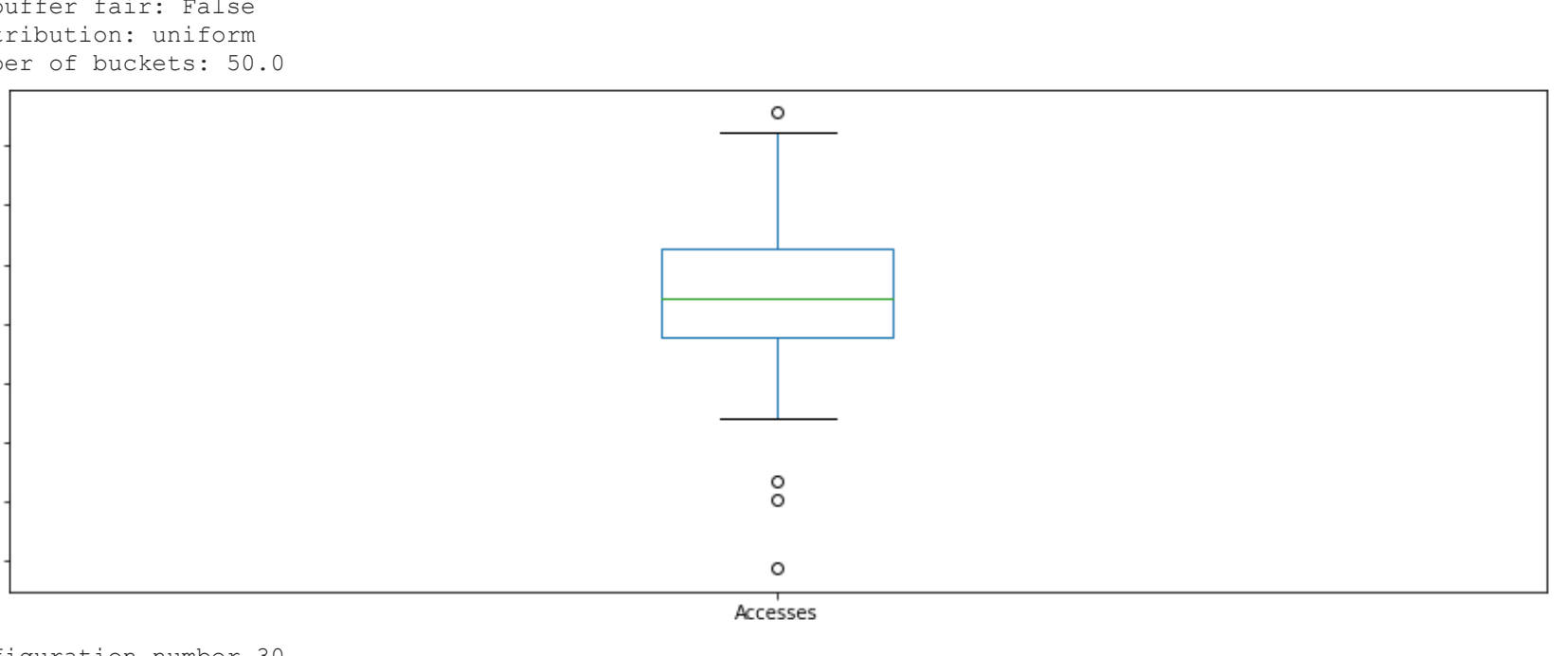
```
Configuration number 24
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



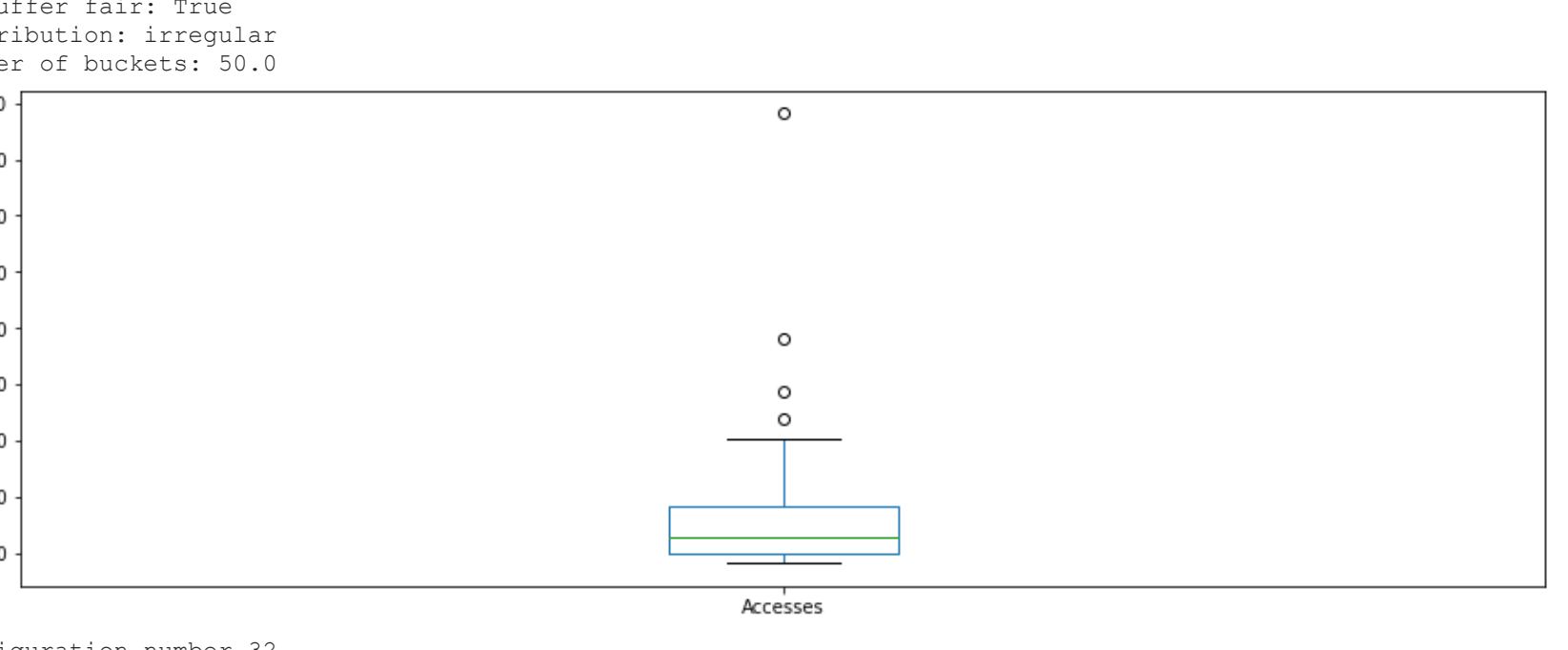
```
Configuration number 26
Buffer size: 100000
Type: producer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



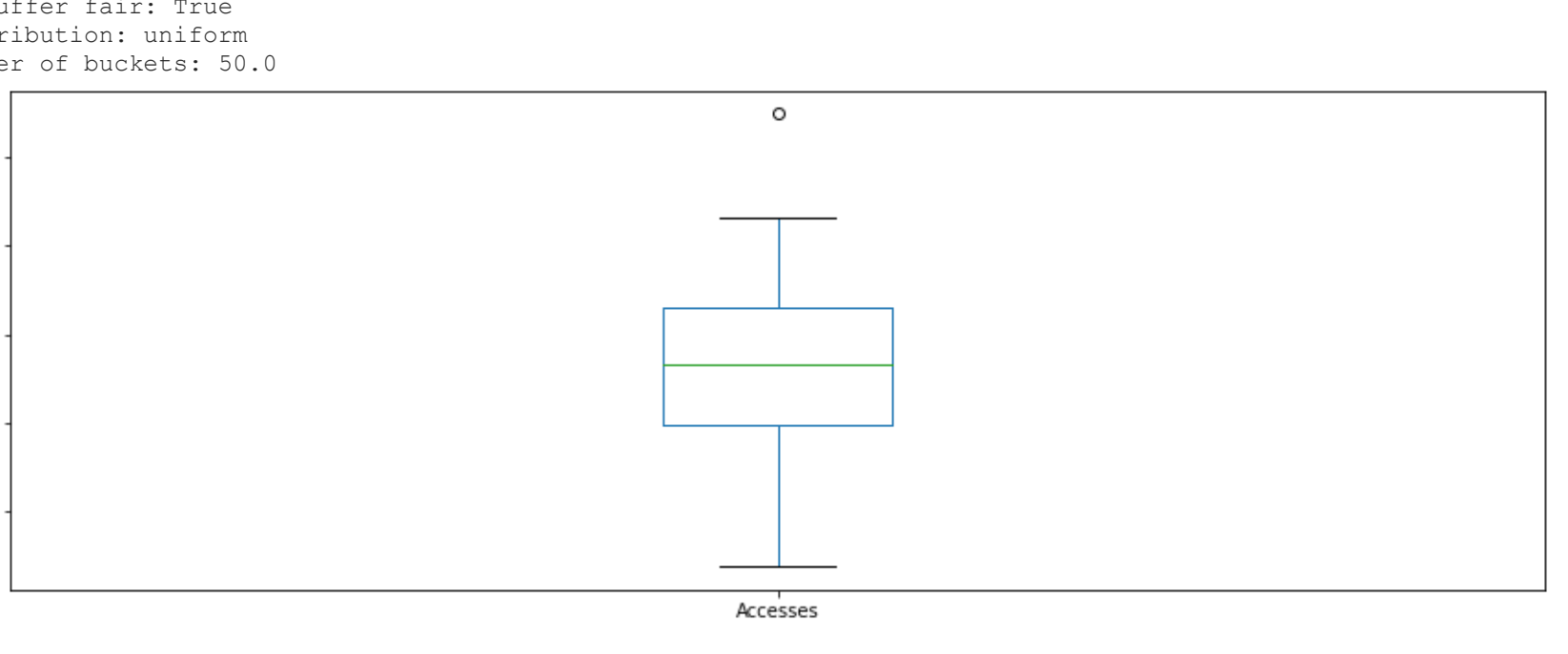
```
Configuration number 28
Buffer size: 100000
Type: producer
PC Config: 1000+1000
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



```
Configuration number 30
Buffer size: 100000
Type: producer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



```
Configuration number 32
Buffer size: 100000
Type: producer
PC Config: 1000+1000
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```

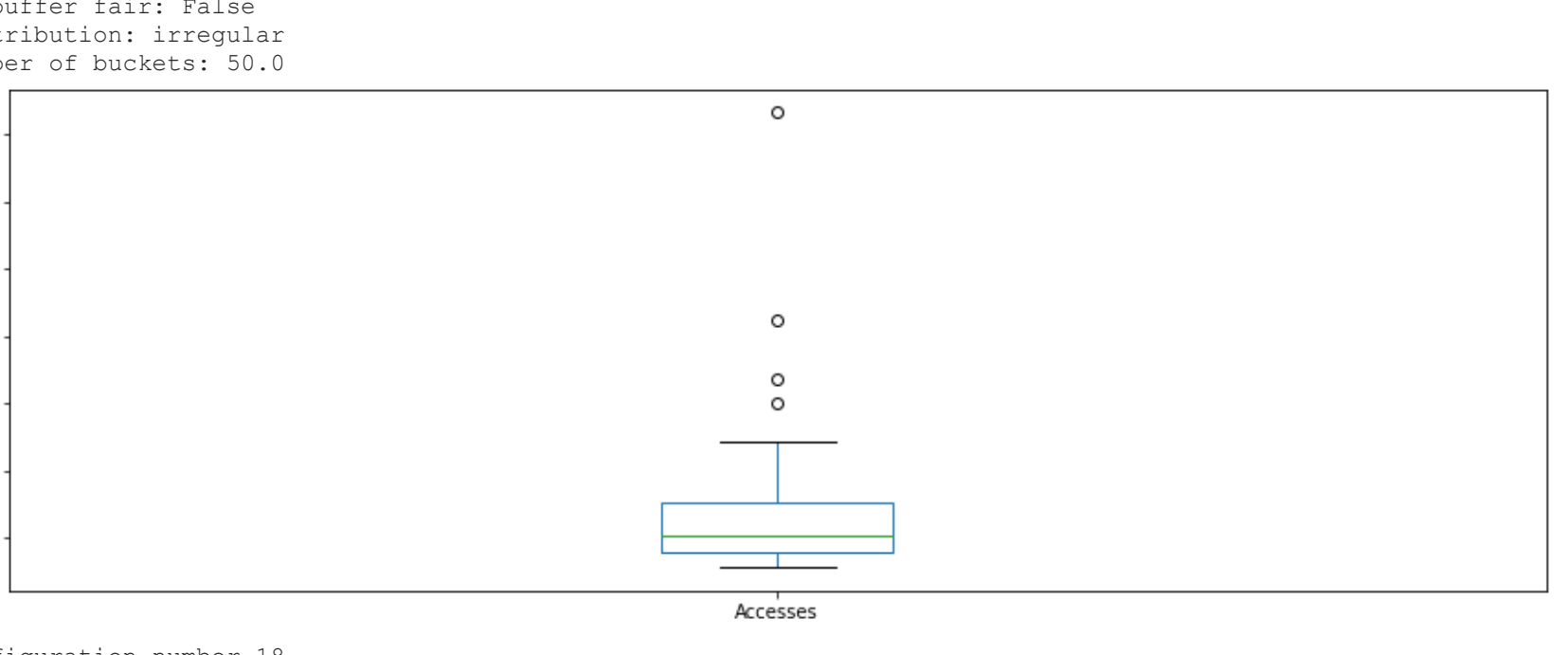


Na pierwszy rzut oka widać, że dla dystrybucji uniform pudełka są zazwyczaj większe i nie mają (lub mają mało wartości odstających).
Przyjrzyjmy się teraz bliżej poszczególnym parametrom.

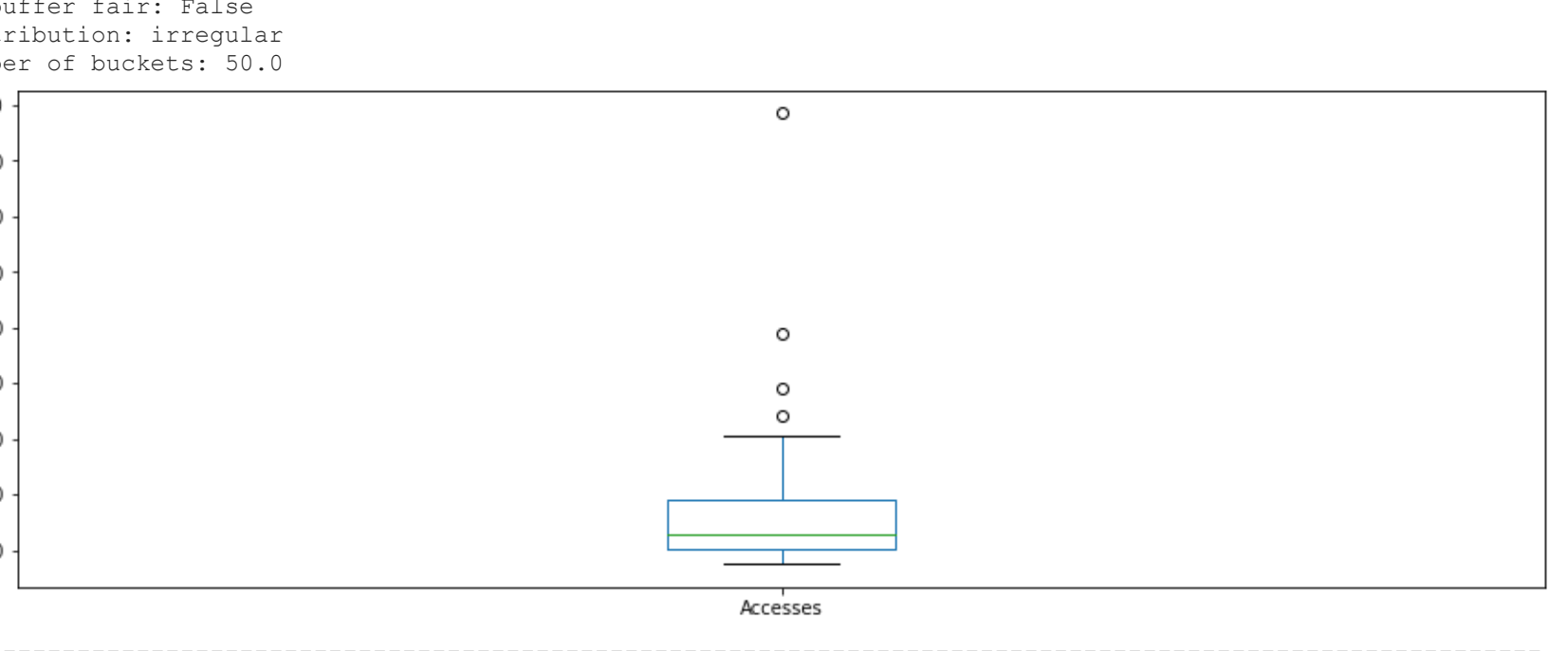
Wielkość bufora

In [104…
```python
draw(data_to_draw[1], 1)
draw(data_to_draw[17], 17)
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
draw(data_to_draw[5], 5)
draw(data_to_draw[21], 21)
```

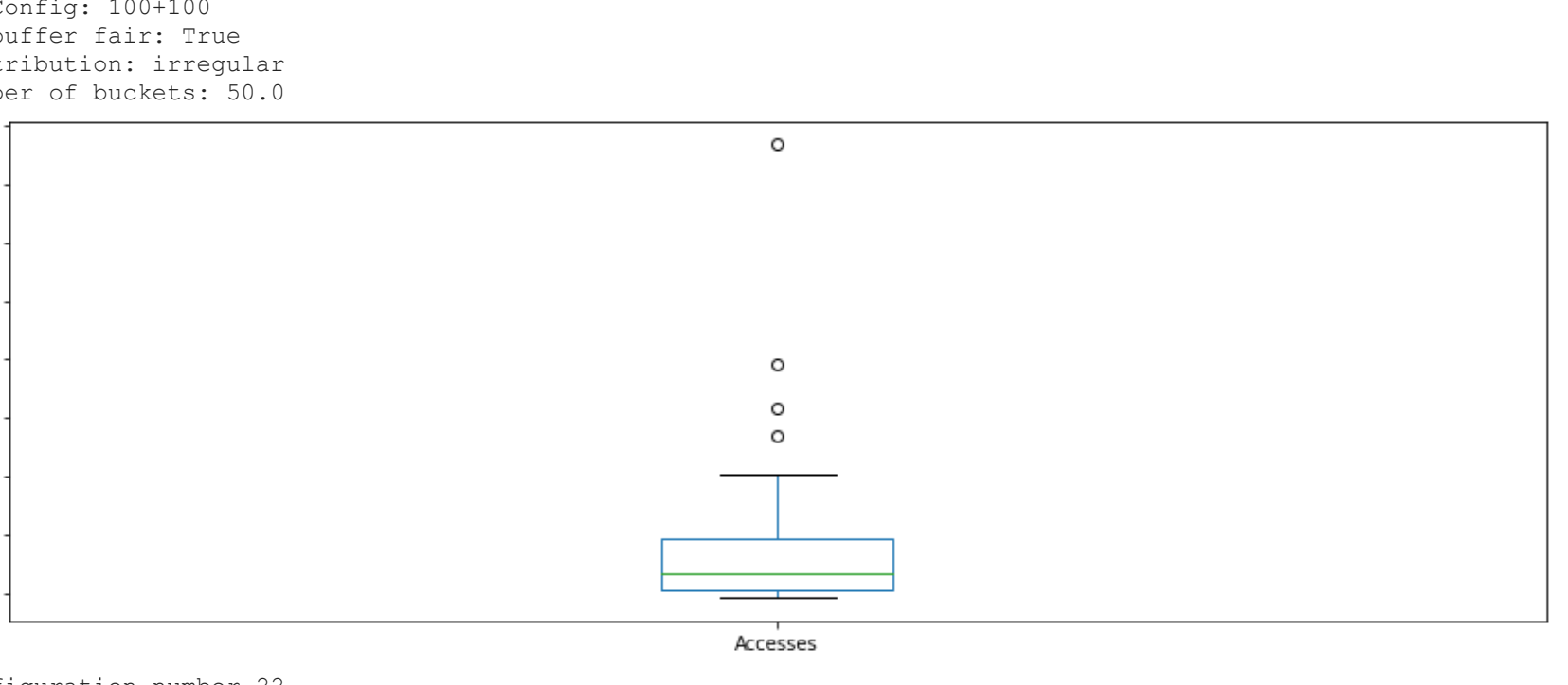
```
Configuration number 2
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



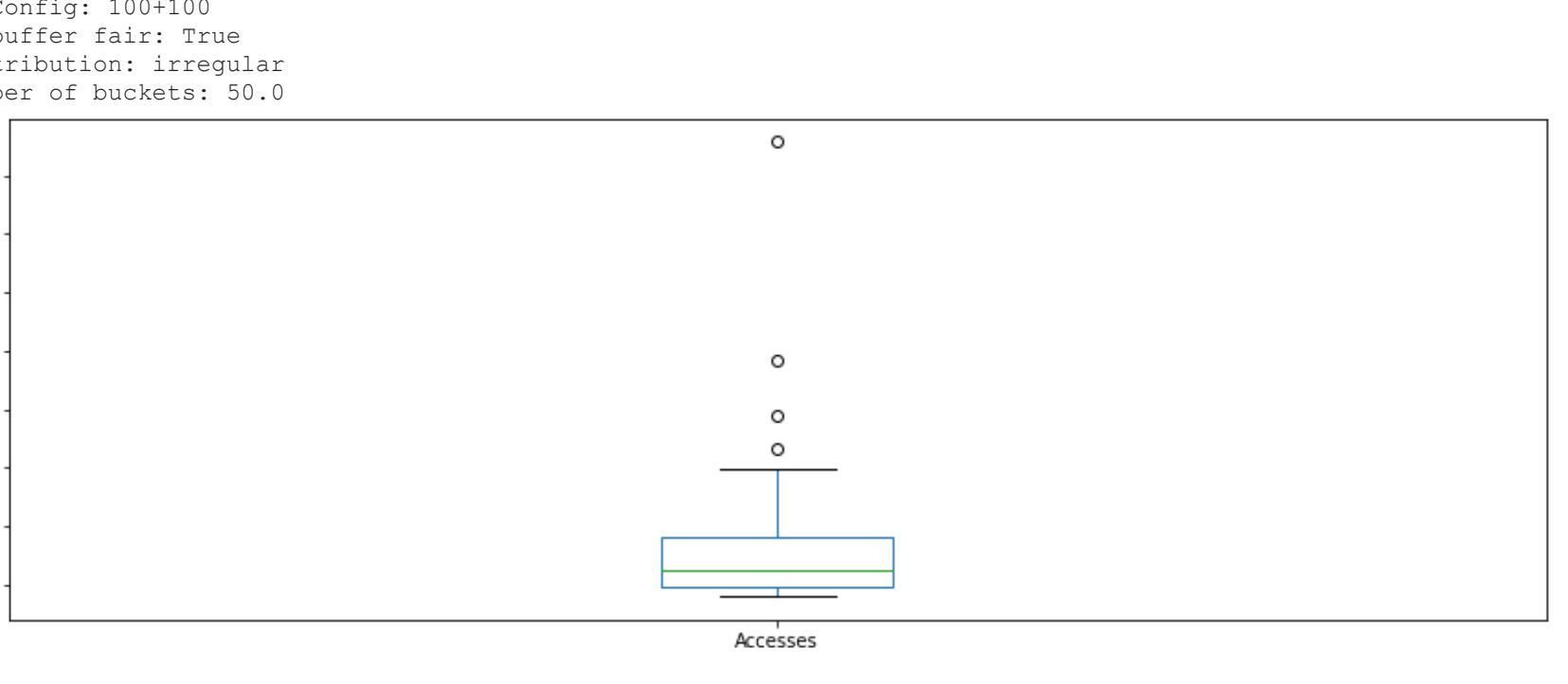
```
Configuration number 18
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



```
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
Configuration number 6
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



```
Configuration number 22
Buffer size: 100000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```

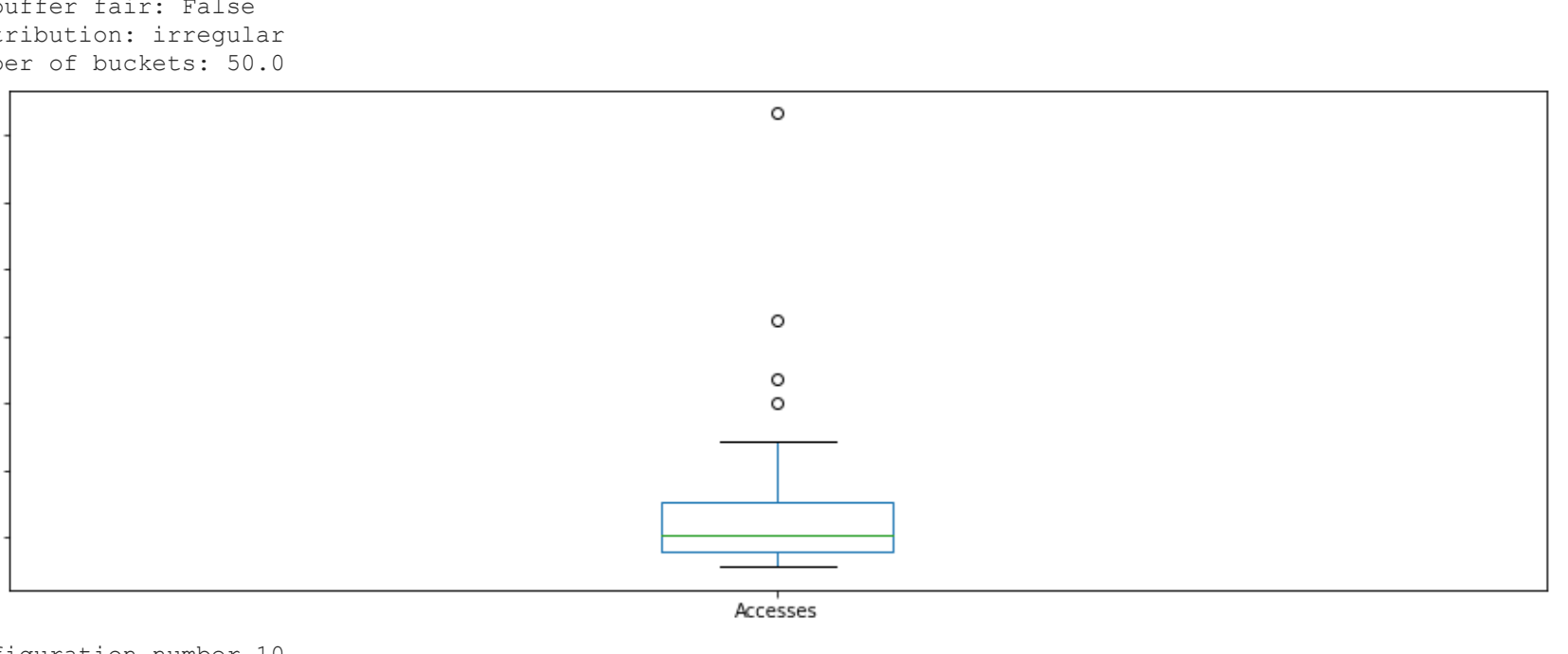


Wniosek: Czym większa większość bufora tym większa średnia ilość dostępów do bufora. Kształt wykresu jest jednak bardzo podobny - ilość buforów rozkłada się zatem podobnie (oczywiście z przeskalowaniem).

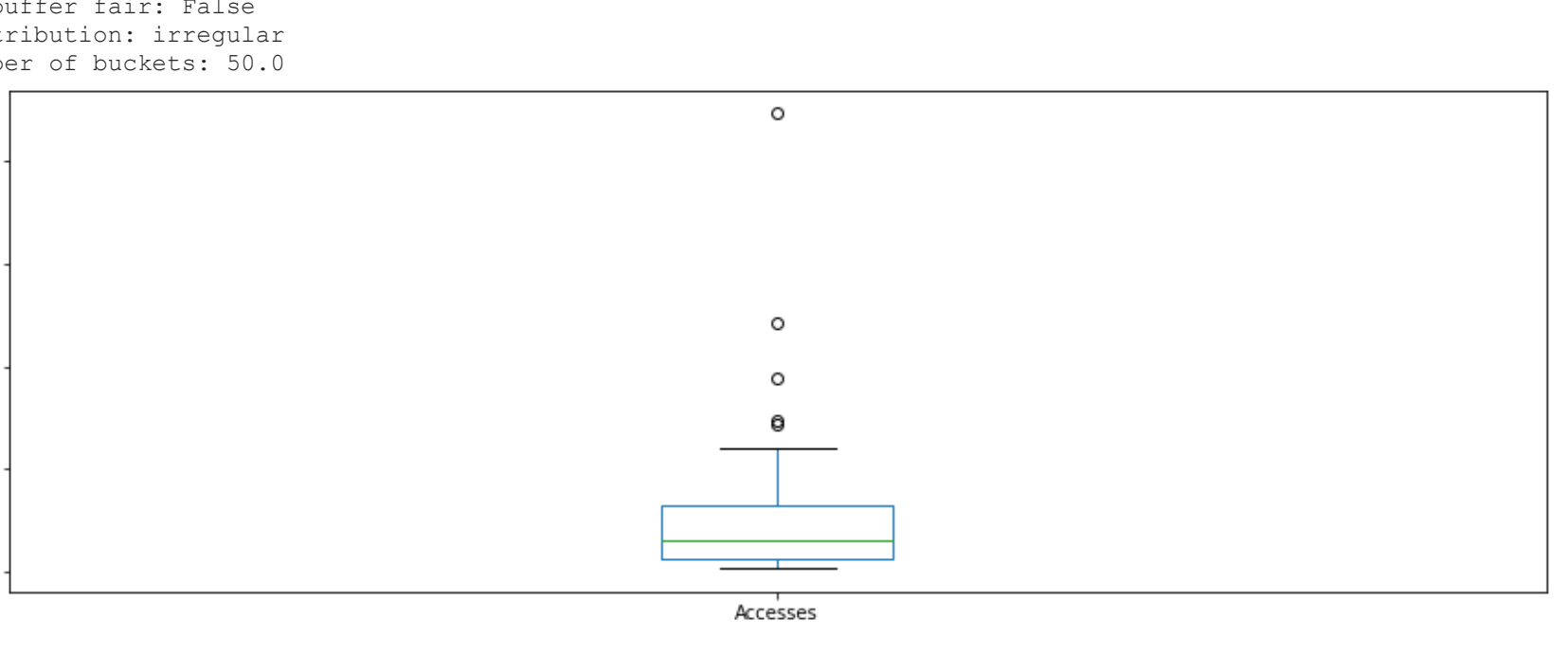
PC Config

In [105…
```python
draw(data_to_draw[1], 1)
draw(data_to_draw[9], 9)
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
draw(data_to_draw[5], 5)
draw(data_to_draw[13], 13)
```

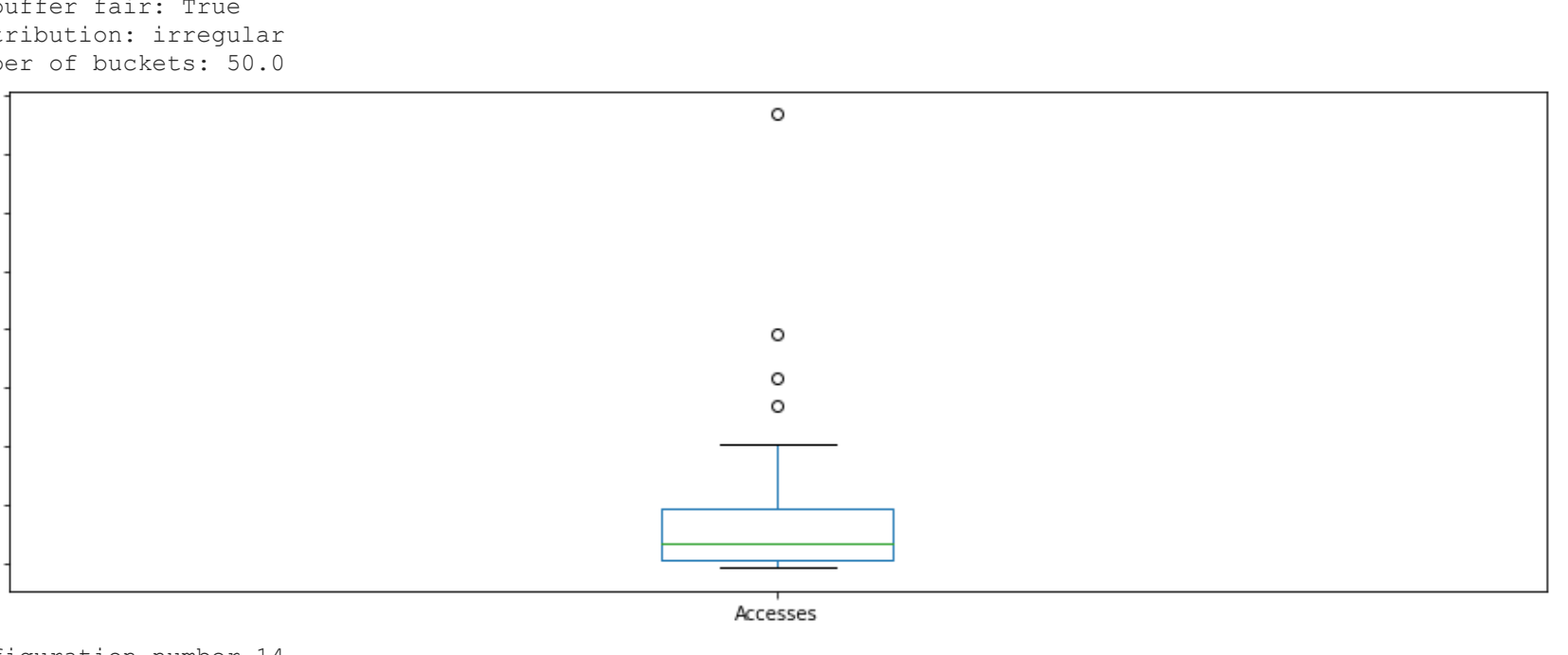
```
Configuration number 2
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



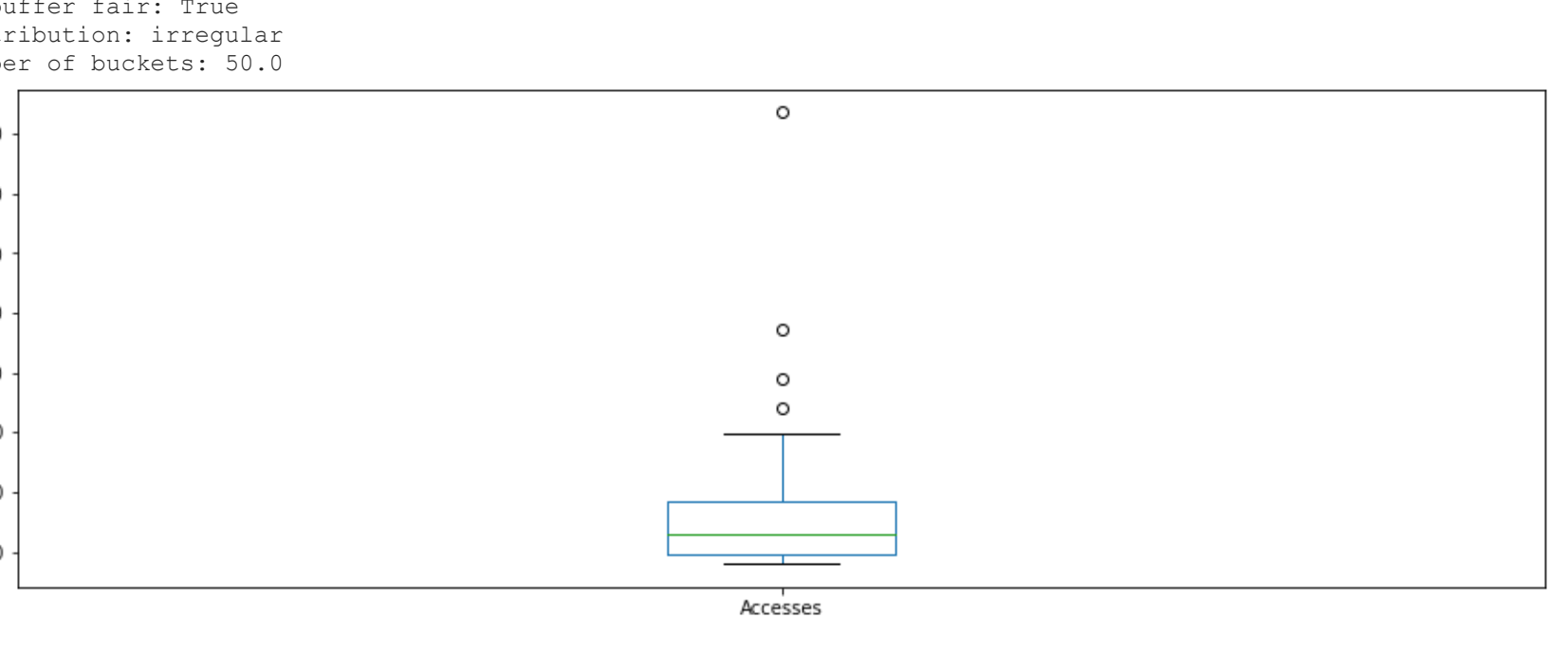
```
Configuration number 10
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



```
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
Configuration number 6
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



```
Configuration number 14
Buffer size: 10000
Type: producer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```

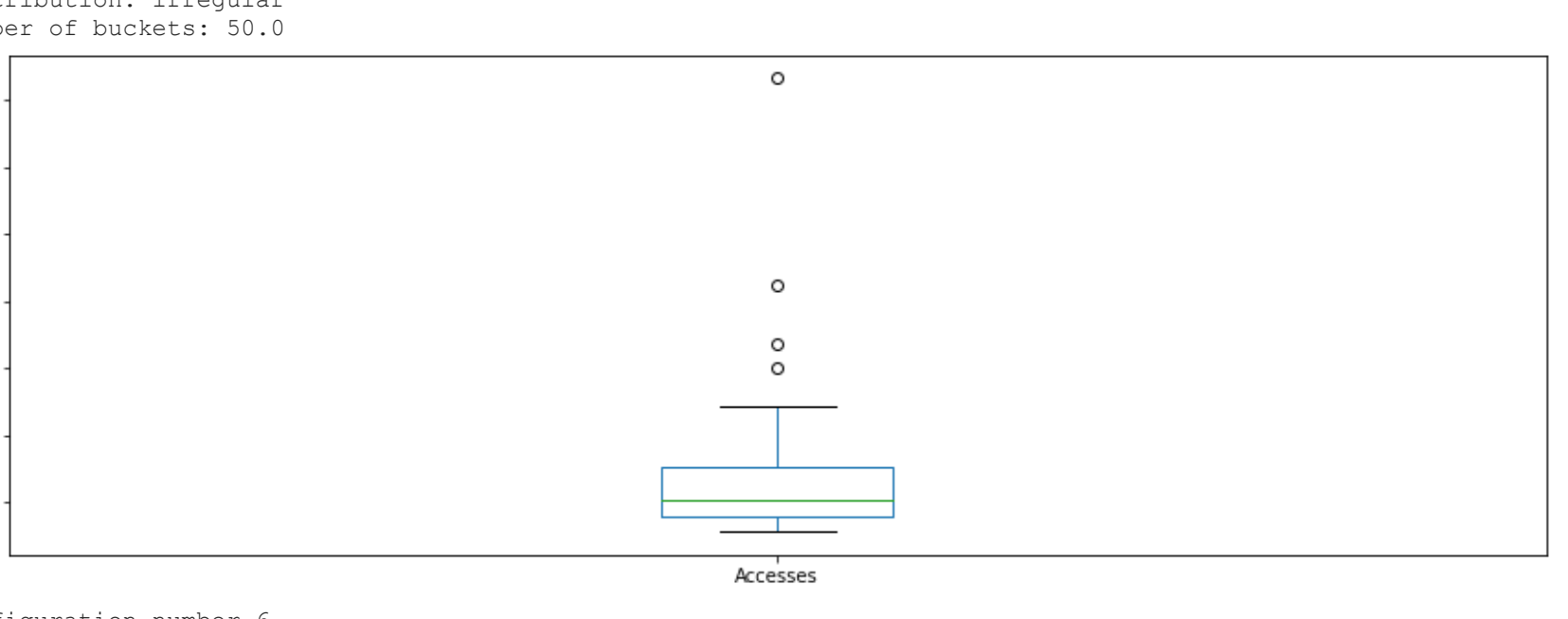


Wniosek: Czym więcej producentów (i konsumentów) tym większa średnia ilość dostępów do bufora. Kształt wykresu jest jednak bardzo podobny - ilość buforów rozkłada się zatem podobnie (oczywiście z przeskalowaniem).

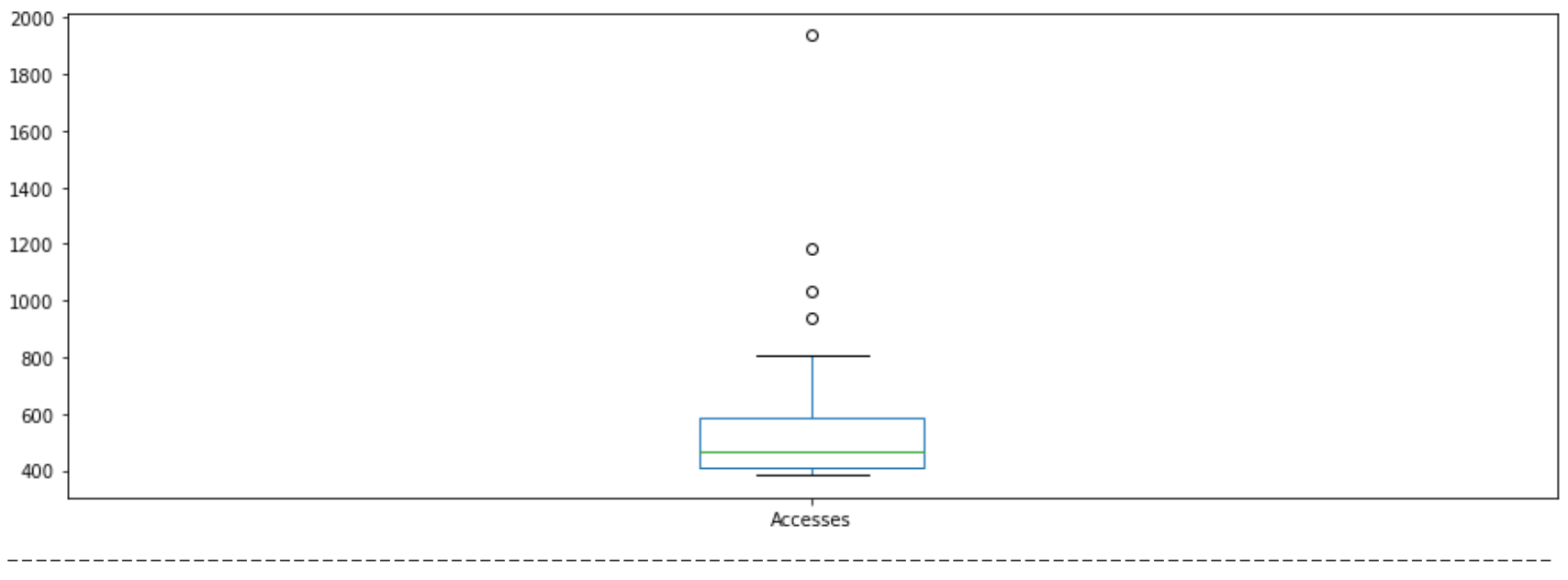
Wariant sprawiedliwy/naiwny

In [106…
```python
draw(data_to_draw[1], 1)
draw(data_to_draw[5], 5)
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
print('------------------------------------------------------------------------------------------------
draw(data_to_draw[3], 3)
draw(data_to_draw[7], 7)
```

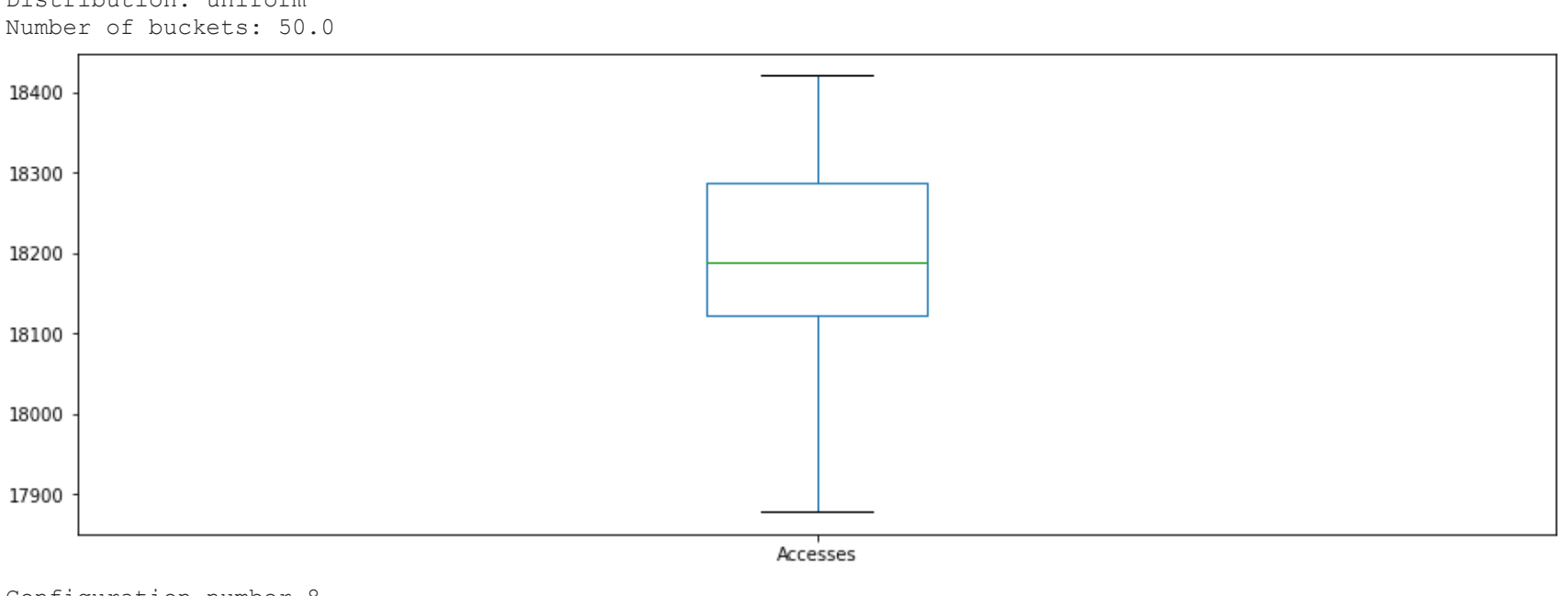
```
Configuration number 2
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



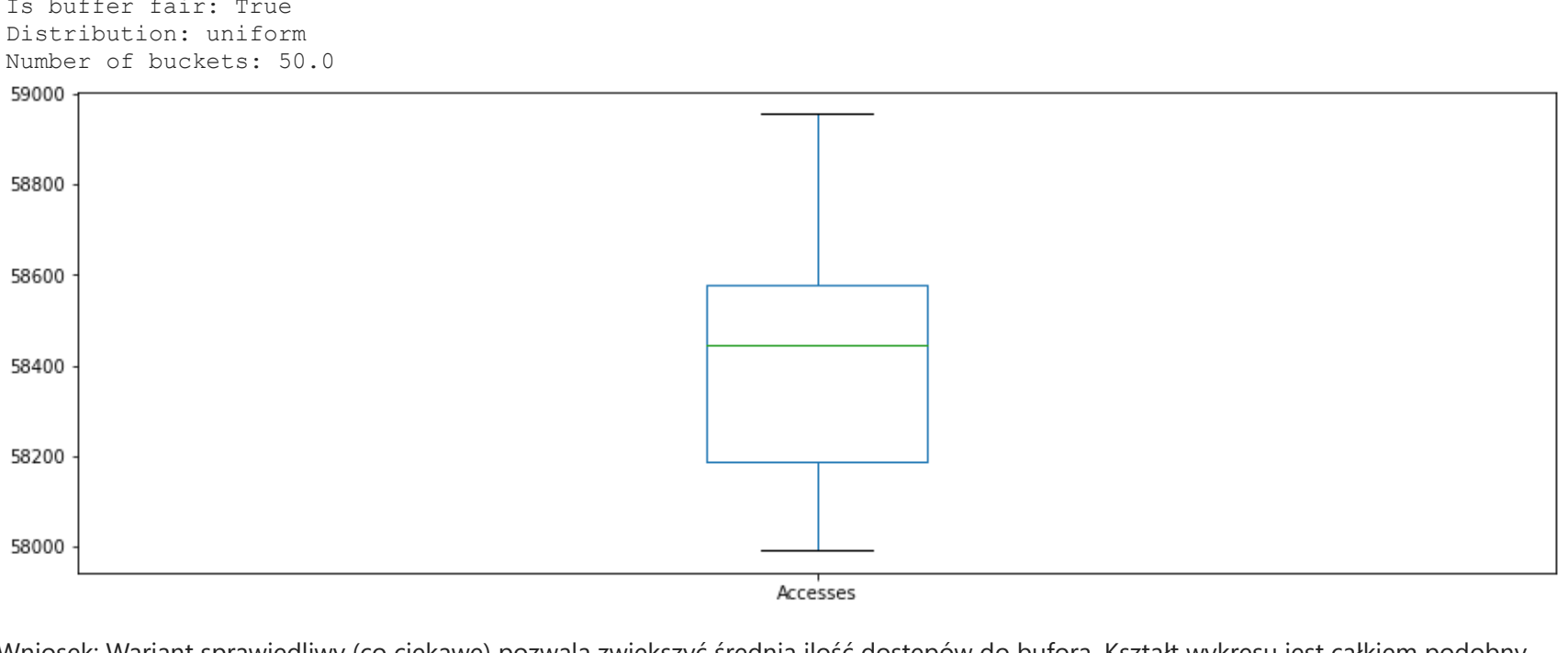
```
Configuration number 6
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```




```
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
Configuration number 4
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```




```
Configuration number 8
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```

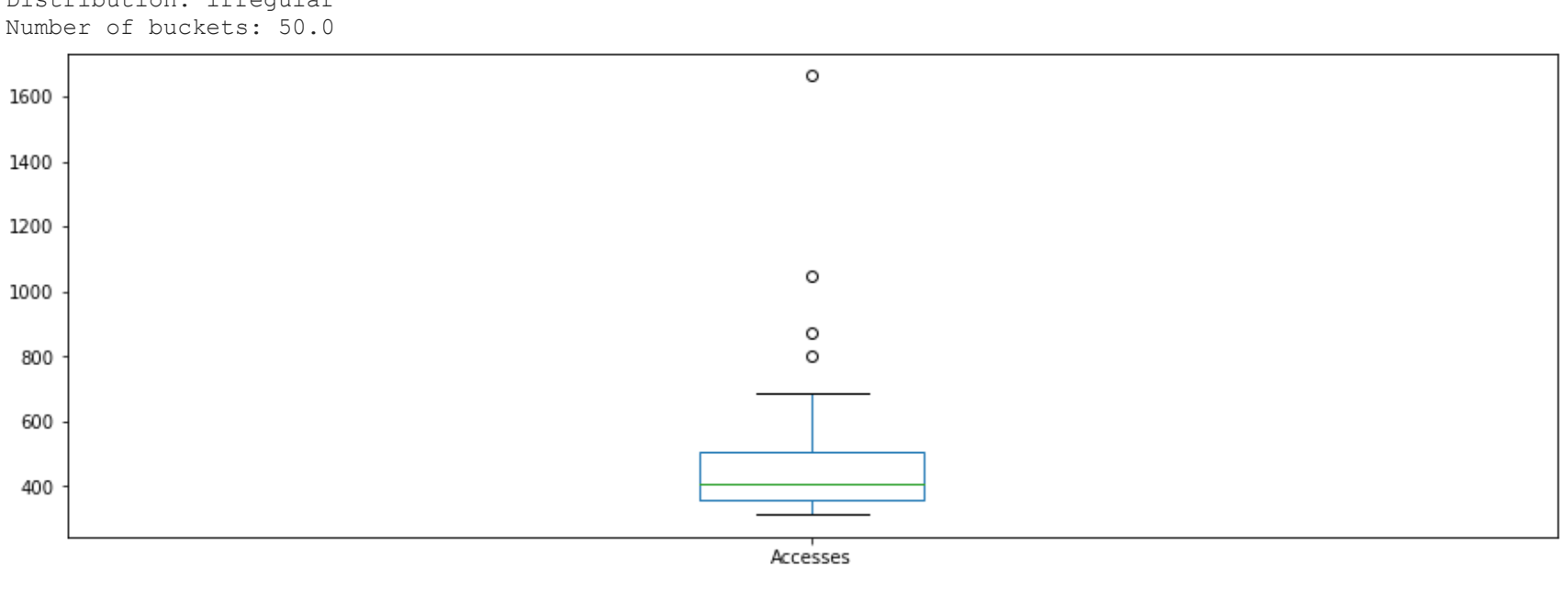


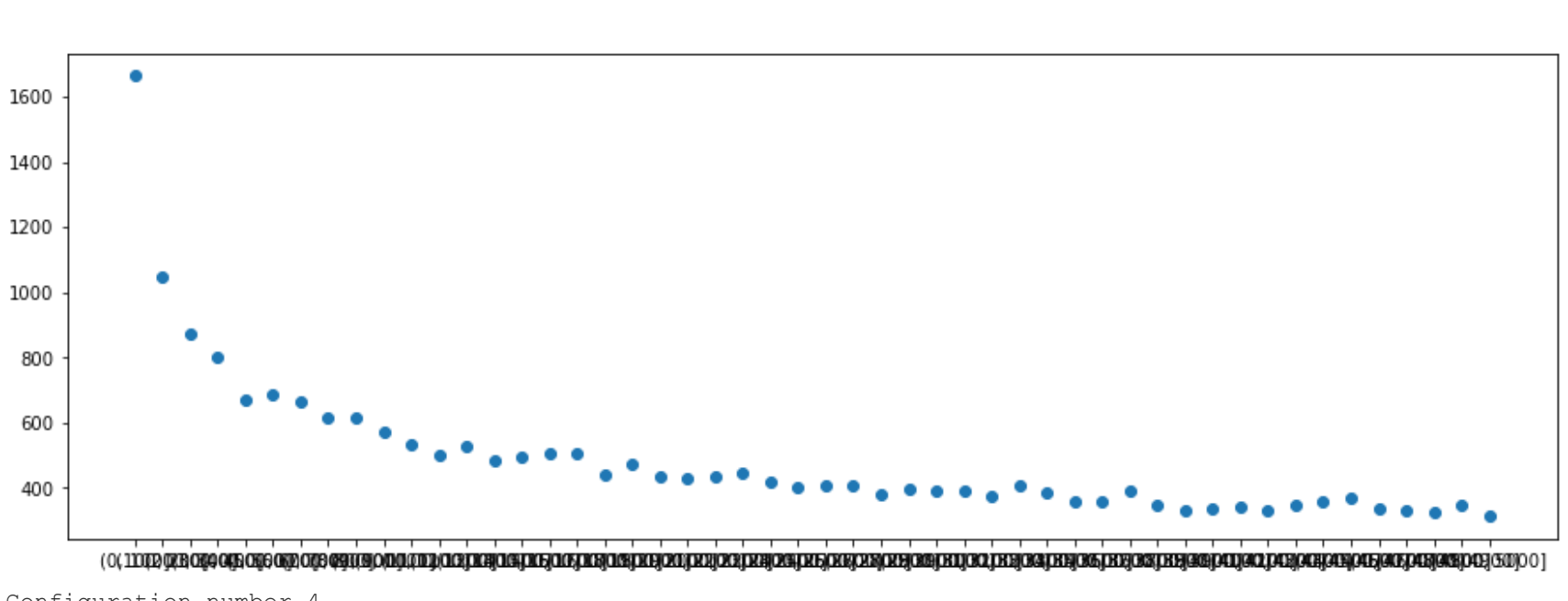
Wniosek: Wariant sprawiedliwy (co ciekawe) pozwala zwiększyć średnią ilość dostępów do bufora. Kształt wykresu jest całkiem podobny - ilość buforów rozkłada się zatem stosunkowo podobnie (oczywiście z przeskalowaniem).

Dystrybucja

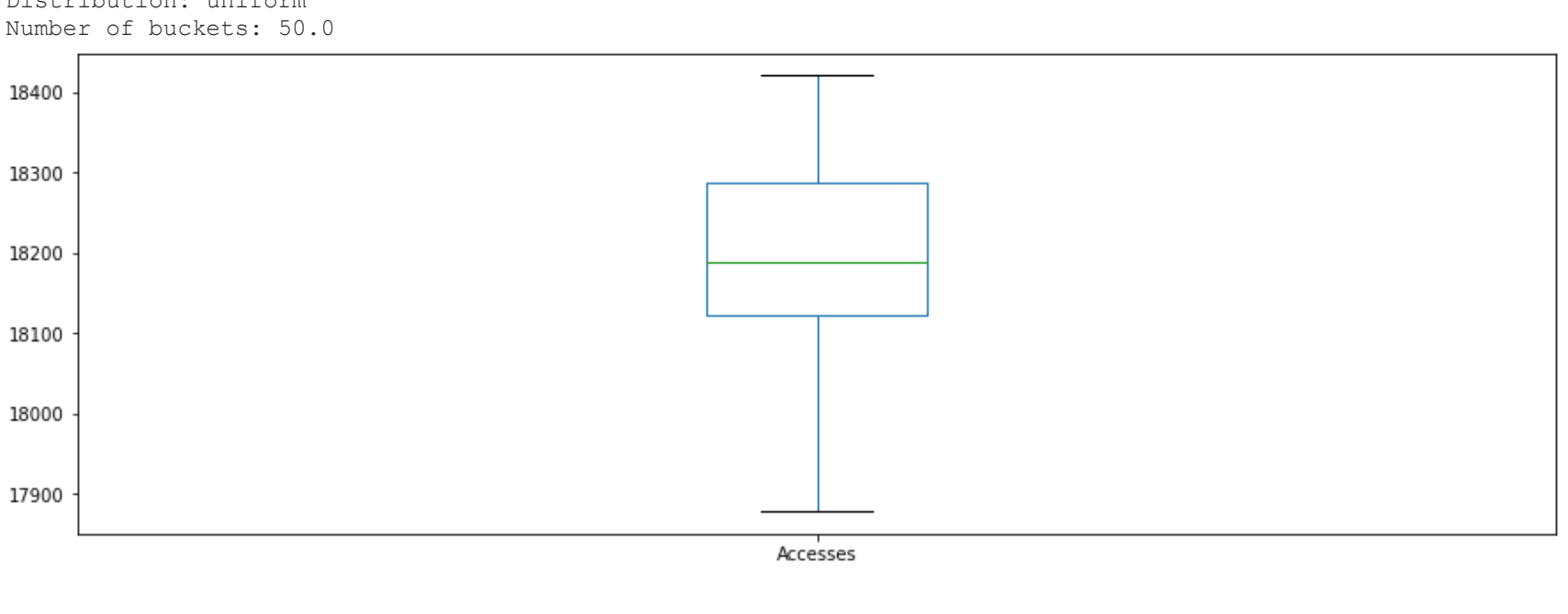
In [109…
```python
draw(data_to_draw[1], 1, print_more = True)
draw(data_to_draw[3], 3, print_more = True)
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
draw(data_to_draw[5], 5, print_more = True)
draw(data_to_draw[7], 7, print_more = True)
```

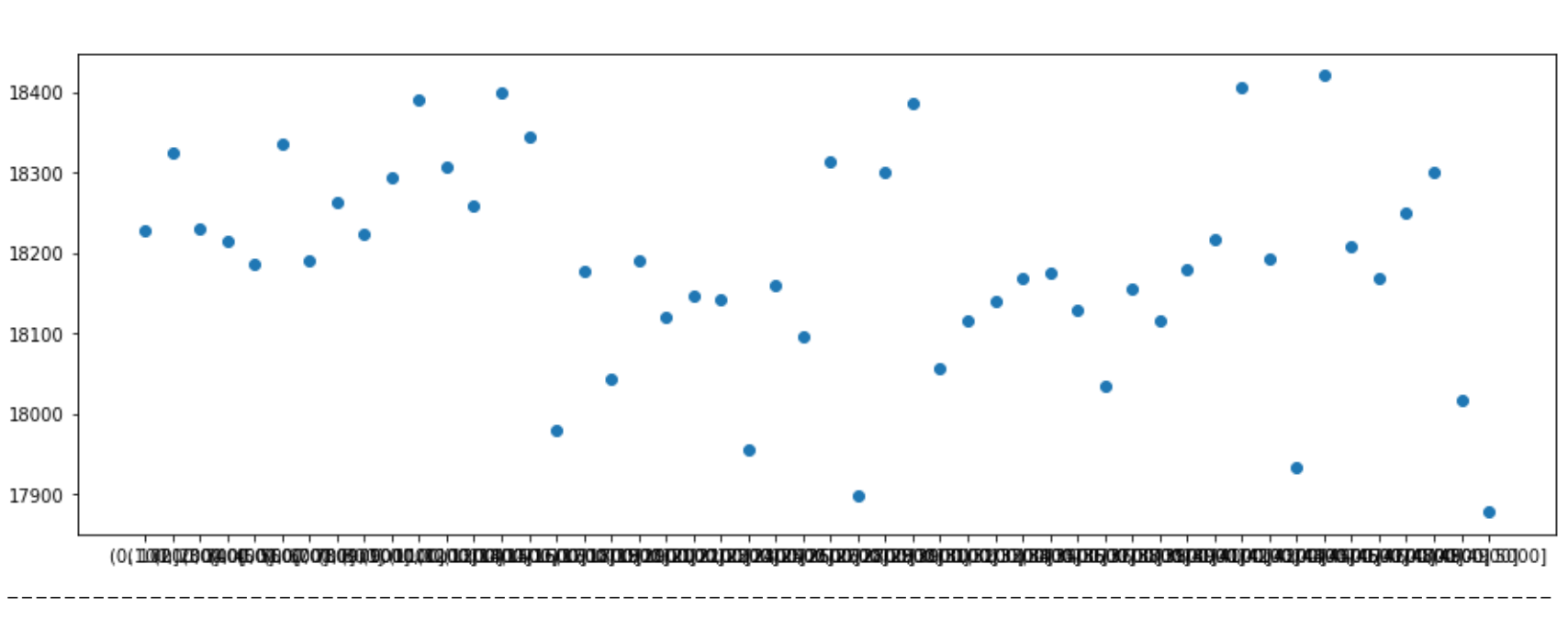
```
Configuration number 2
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



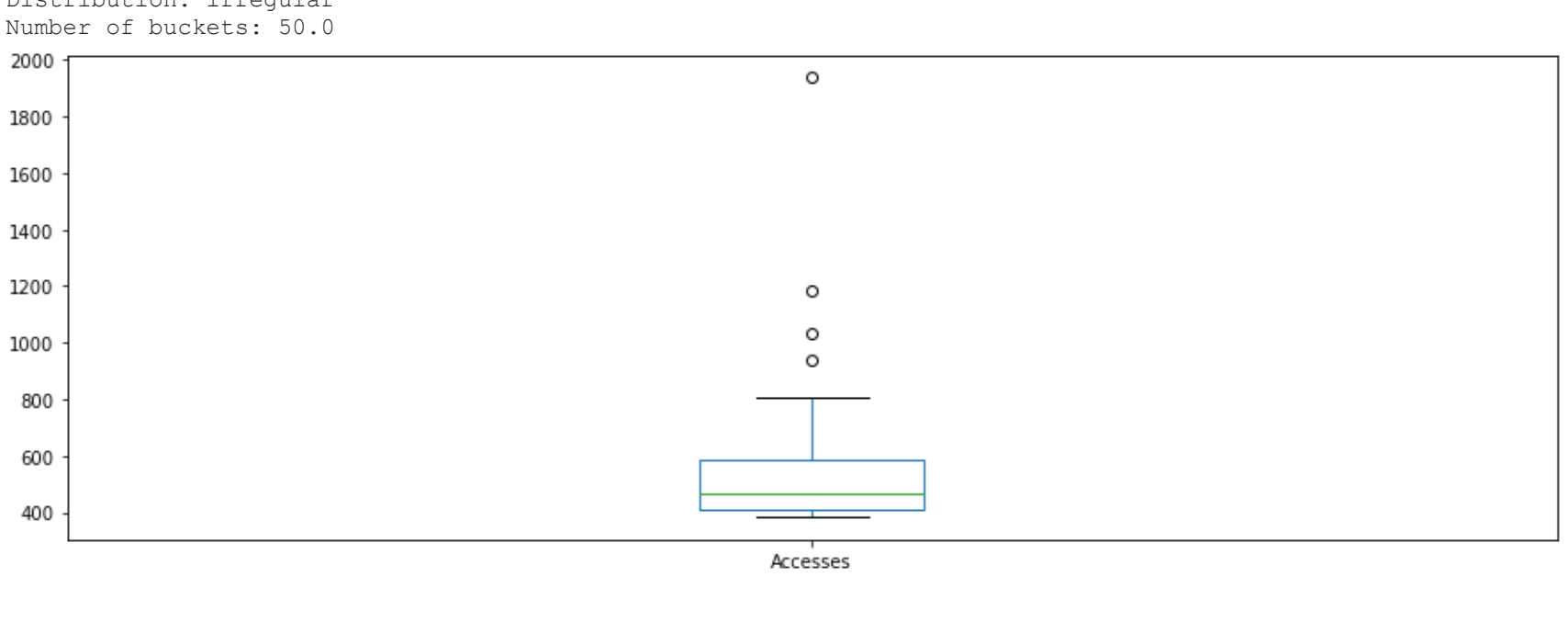


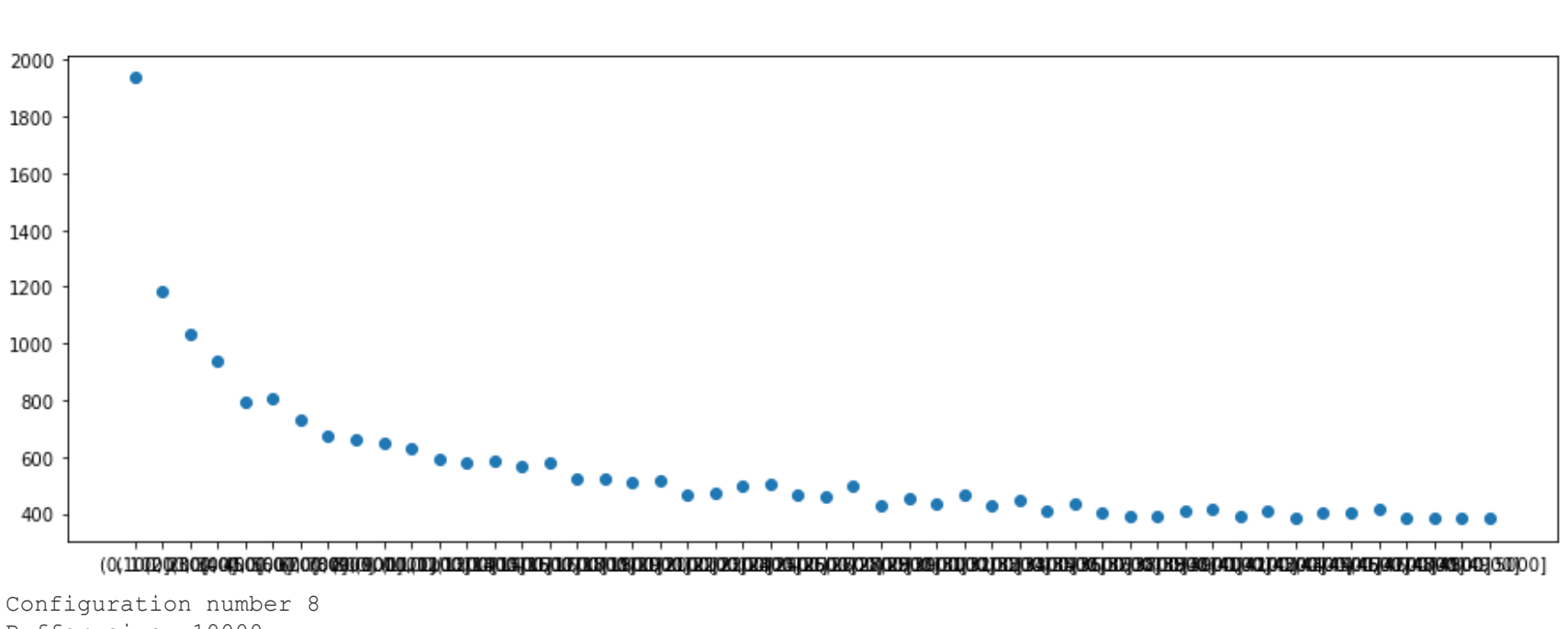
```
Configuration number 4
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



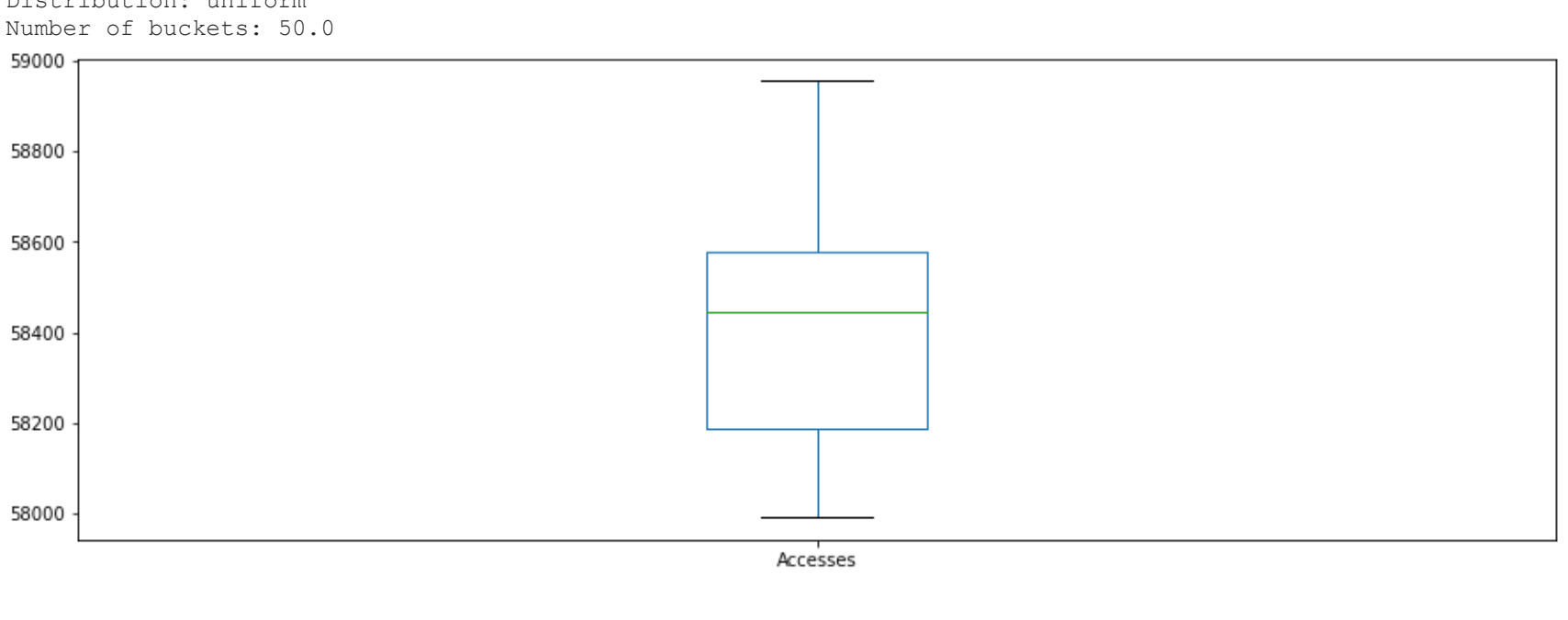


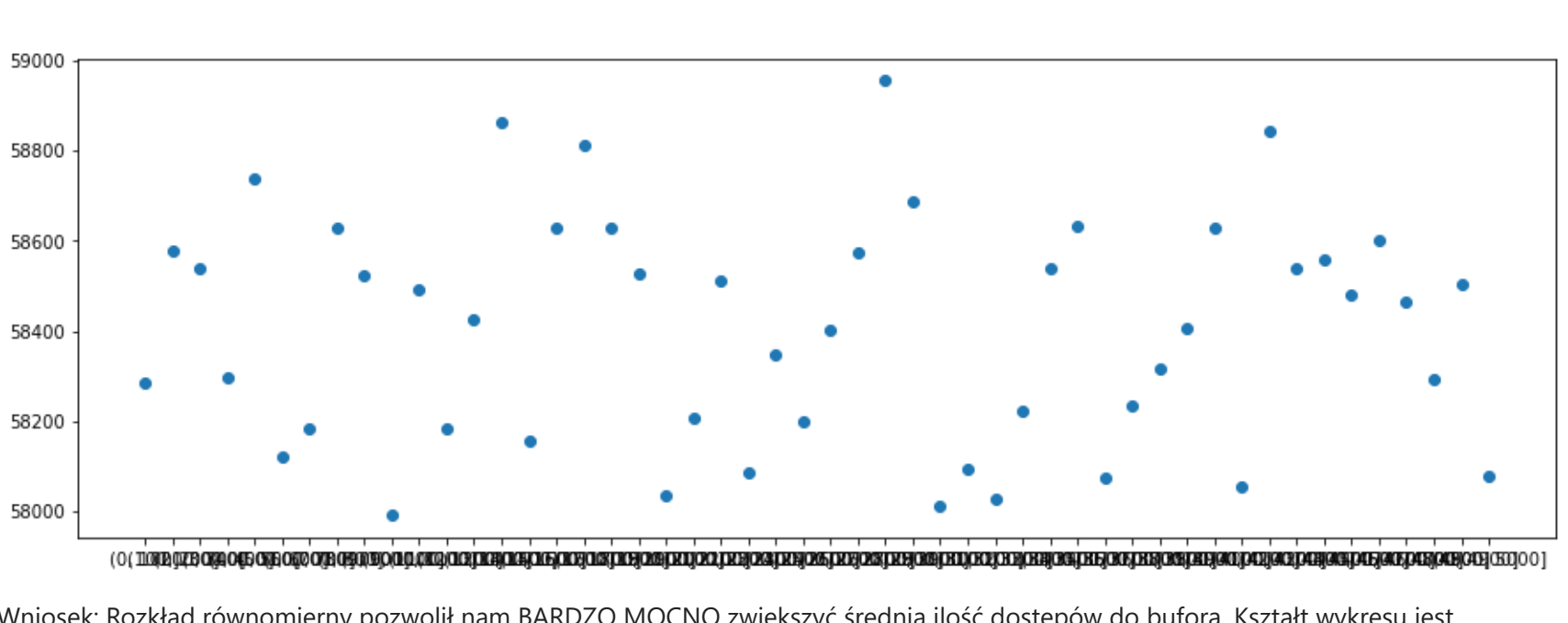
```
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
Configuration number 6
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```





```
Configuration number 8
Buffer size: 10000
Type: producer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```





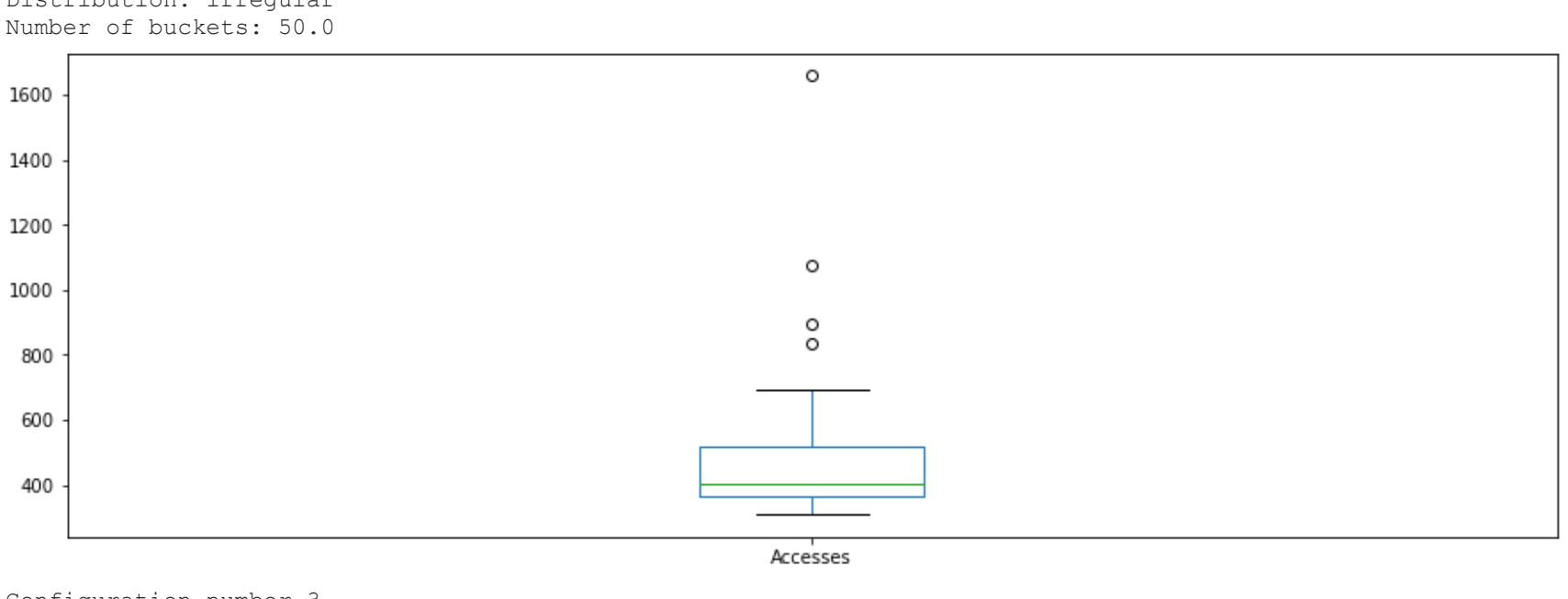
Wniosek: Rozkład równomierny pozwolił nam BARDZO MOCNO zwiększyć średnią ilość dostępów do bufora. Kształt wykresu jest zupełnie inny - wynika to z tego, że przy rozkładzie równomiernym więcej wartości jest dopuszczonych (a więc pudełko większe). Pomocniczy wykres pokazuje, że faktycznie rozkład w jednym wypadku jest równomierny a w drugim mniejsze liczby mają większe prawdopodobieństwo.

2. Konsumenci

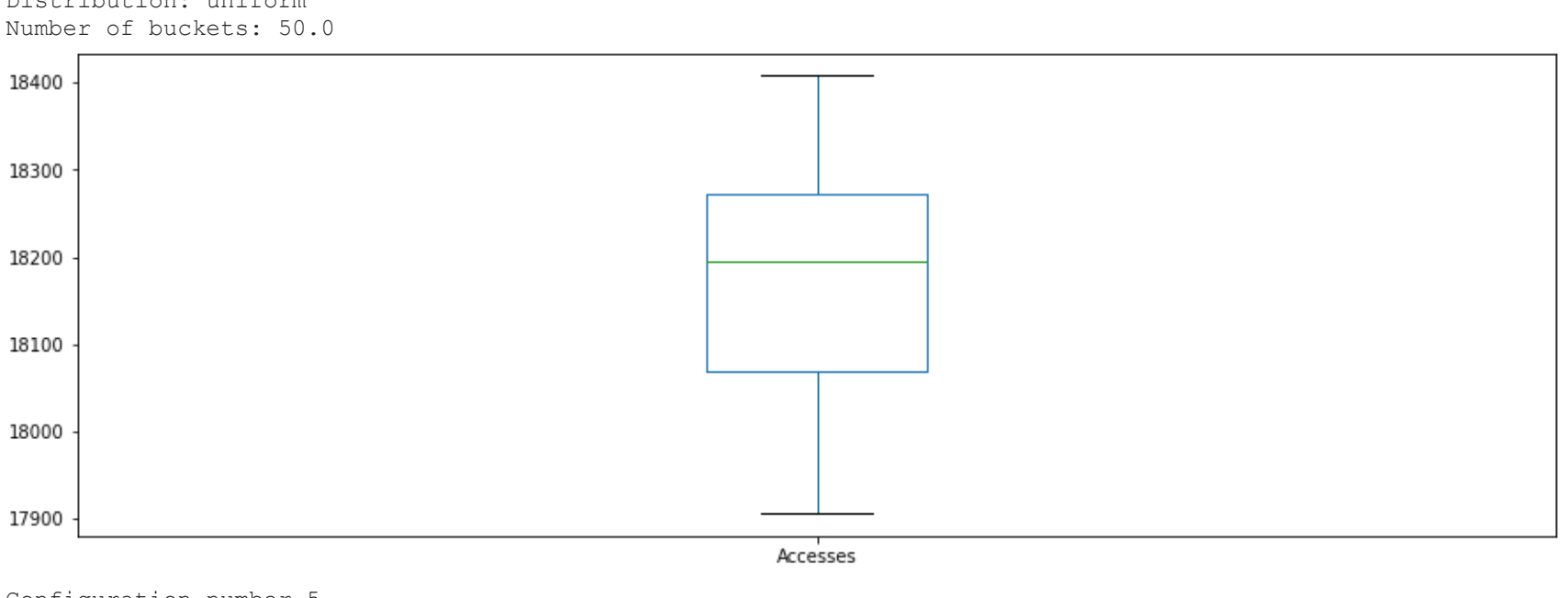
Zaczniemy od narysowania wszystkich wykresów pudełkowych dla konsumentów (16 wykresów).

In [111…
```python
draw(data_to_draw[0], 0)
draw(data_to_draw[2], 2)
draw(data_to_draw[4], 4)
draw(data_to_draw[6], 6)
draw(data_to_draw[8], 8)
draw(data_to_draw[10], 10)
draw(data_to_draw[12], 12)
draw(data_to_draw[14], 14)
draw(data_to_draw[16], 16)
draw(data_to_draw[18], 18)
draw(data_to_draw[20], 20)
draw(data_to_draw[22], 22)
draw(data_to_draw[24], 24)
draw(data_to_draw[26], 26)
draw(data_to_draw[28], 28)
draw(data_to_draw[30], 30)
```

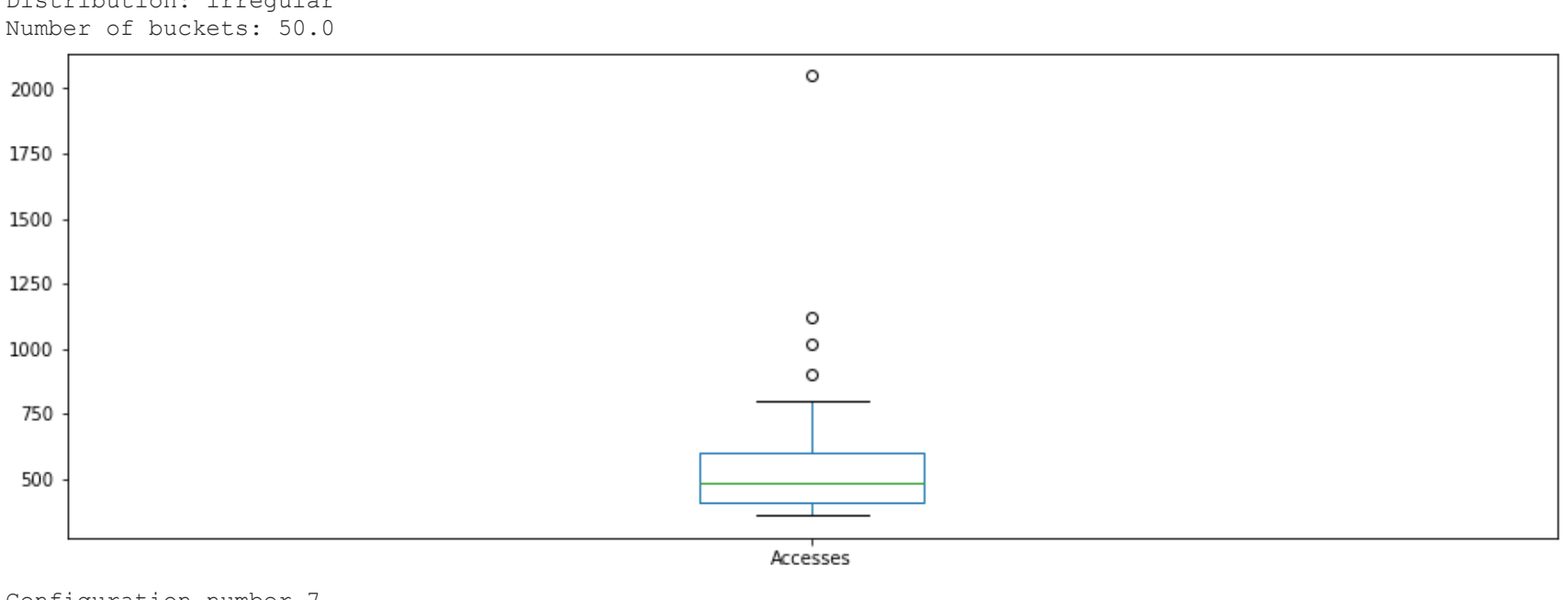
```
Configuration number 1
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



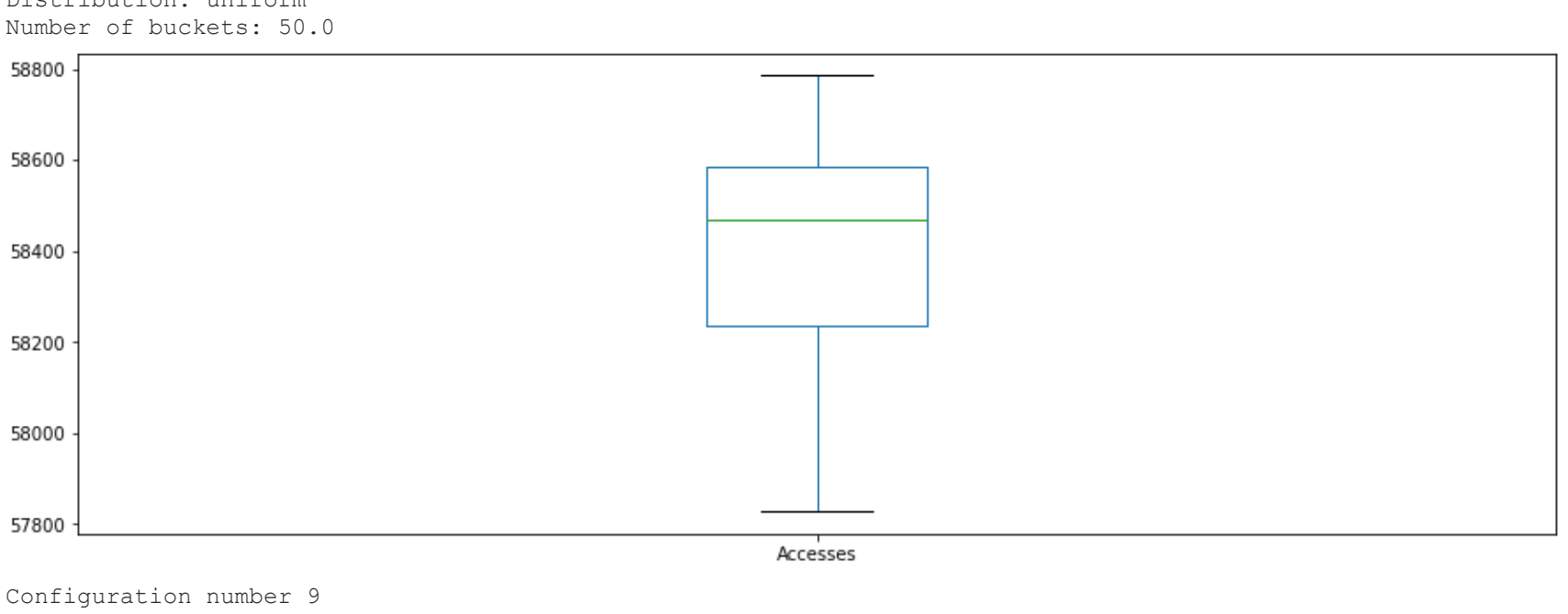

```
Configuration number 3
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



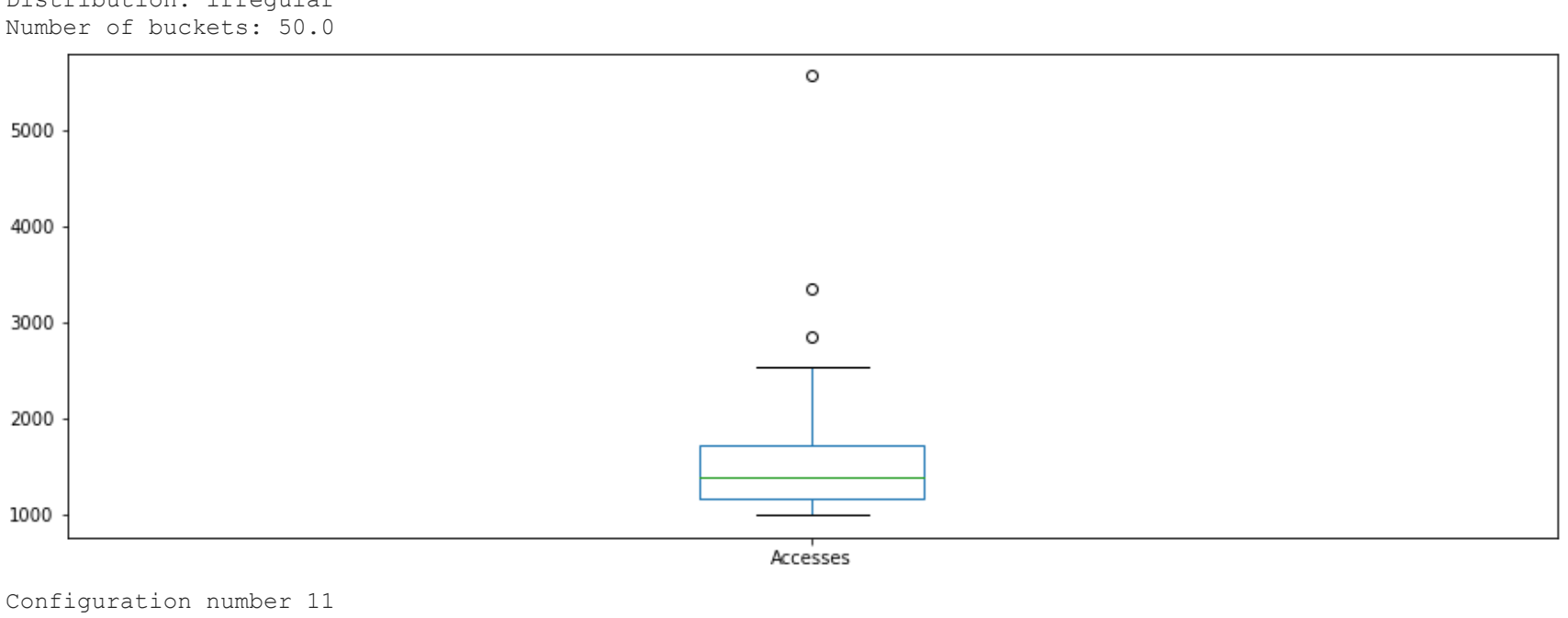

```
Configuration number 5
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



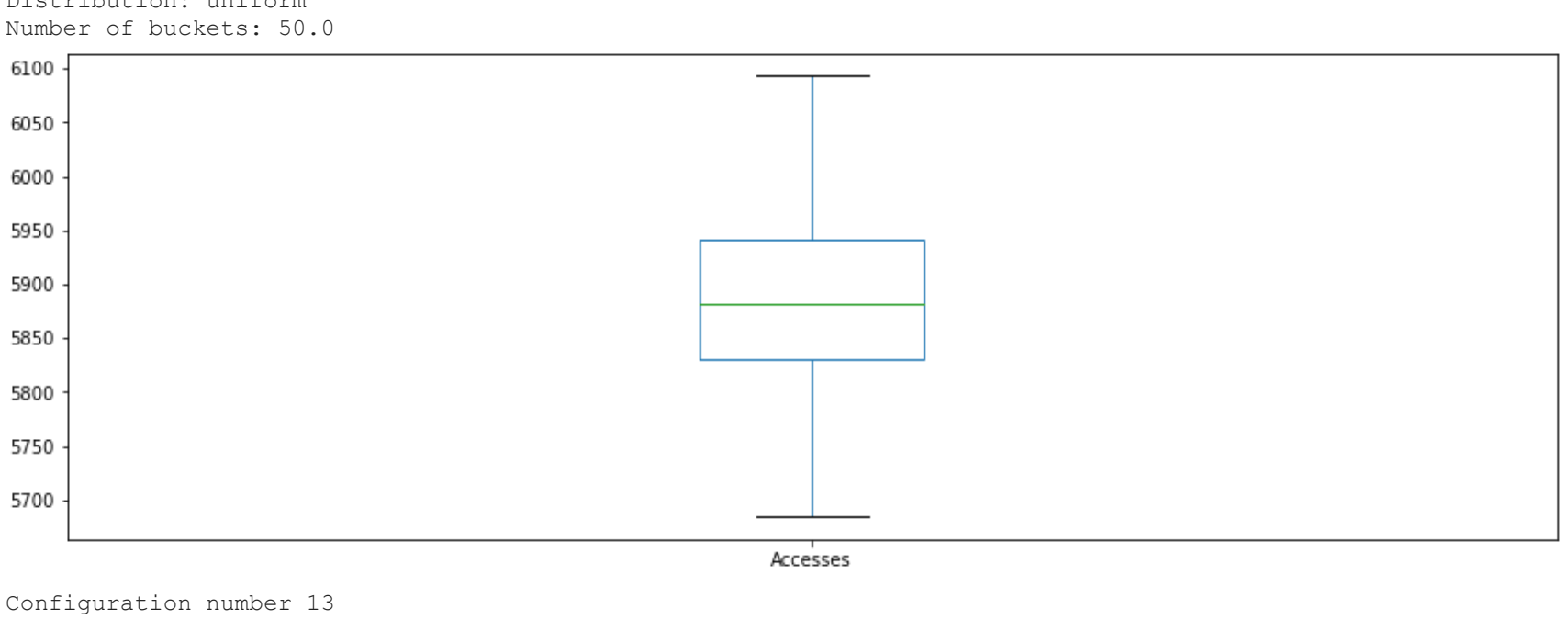

```
Configuration number 7
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



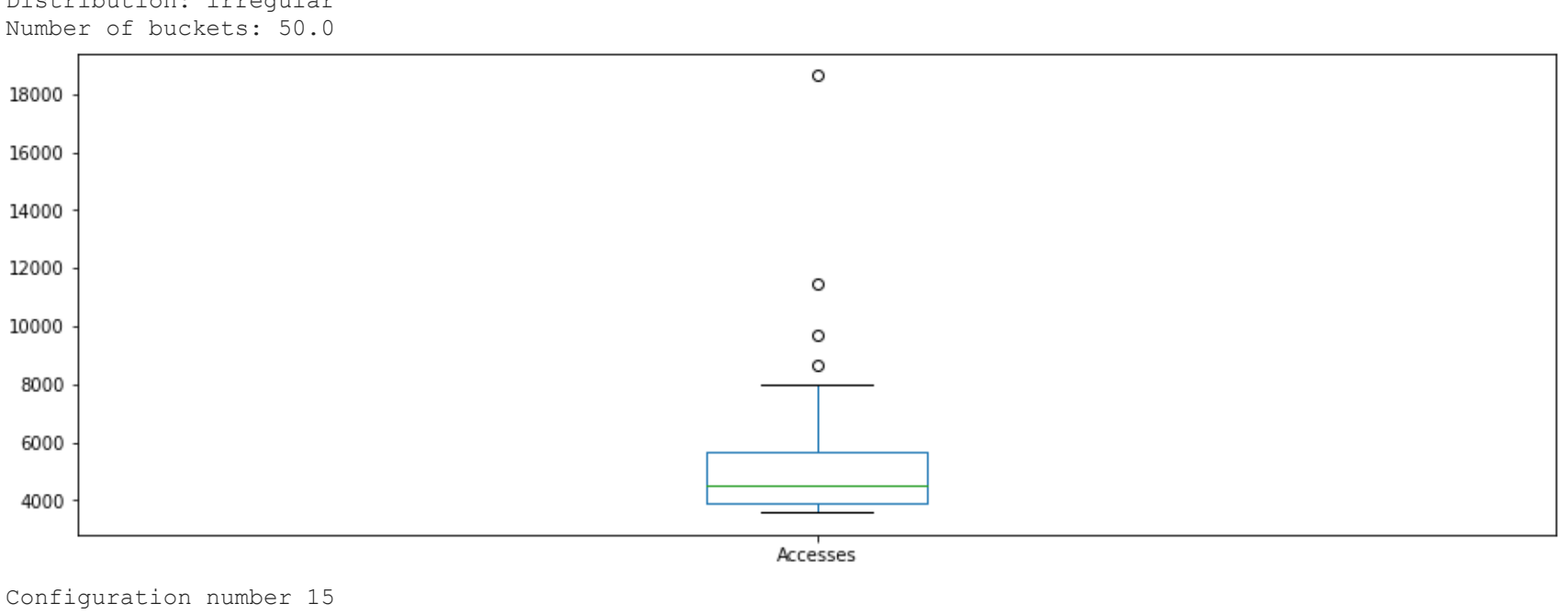

```
Configuration number 9
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



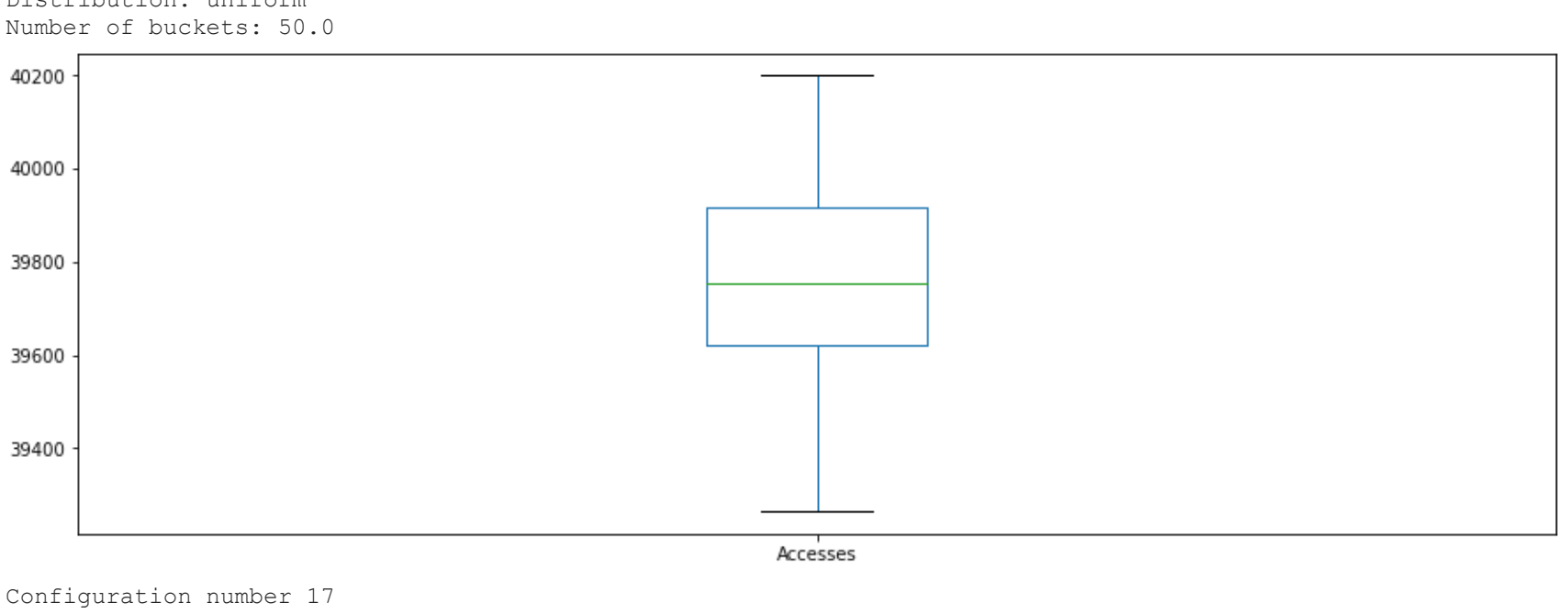

```
Configuration number 11
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



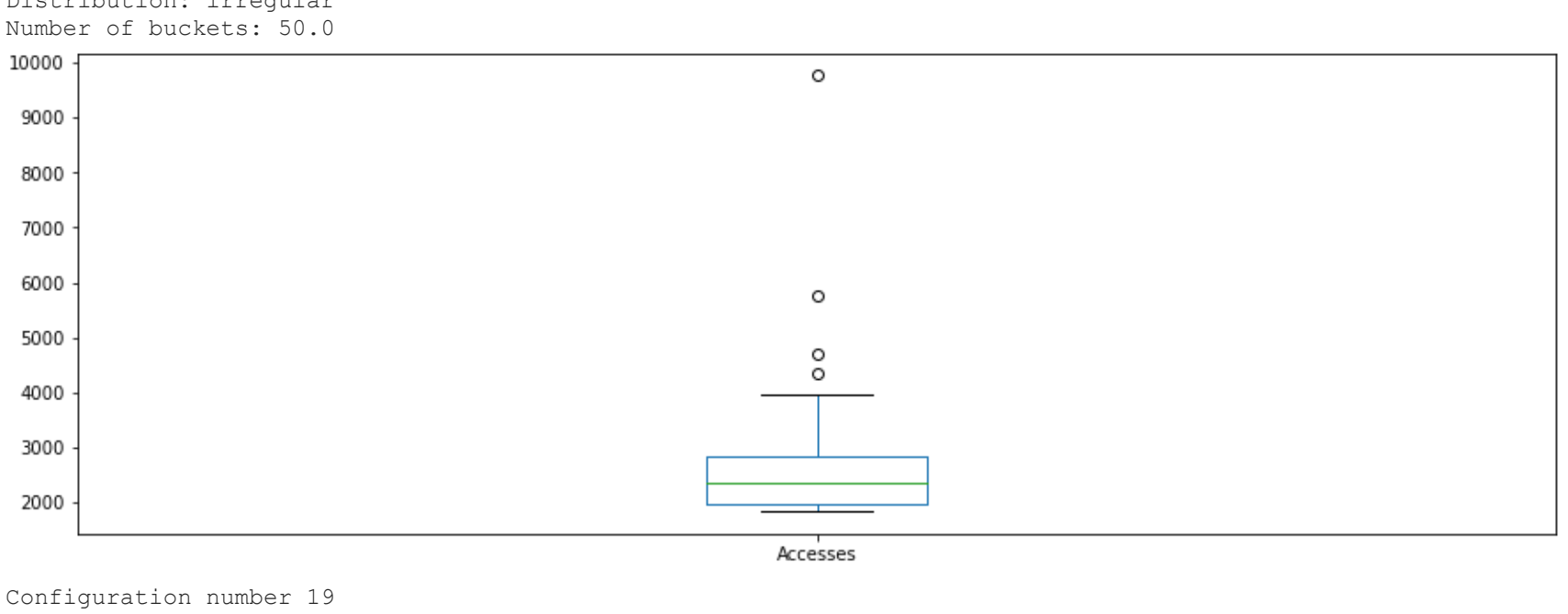

```
Configuration number 13
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



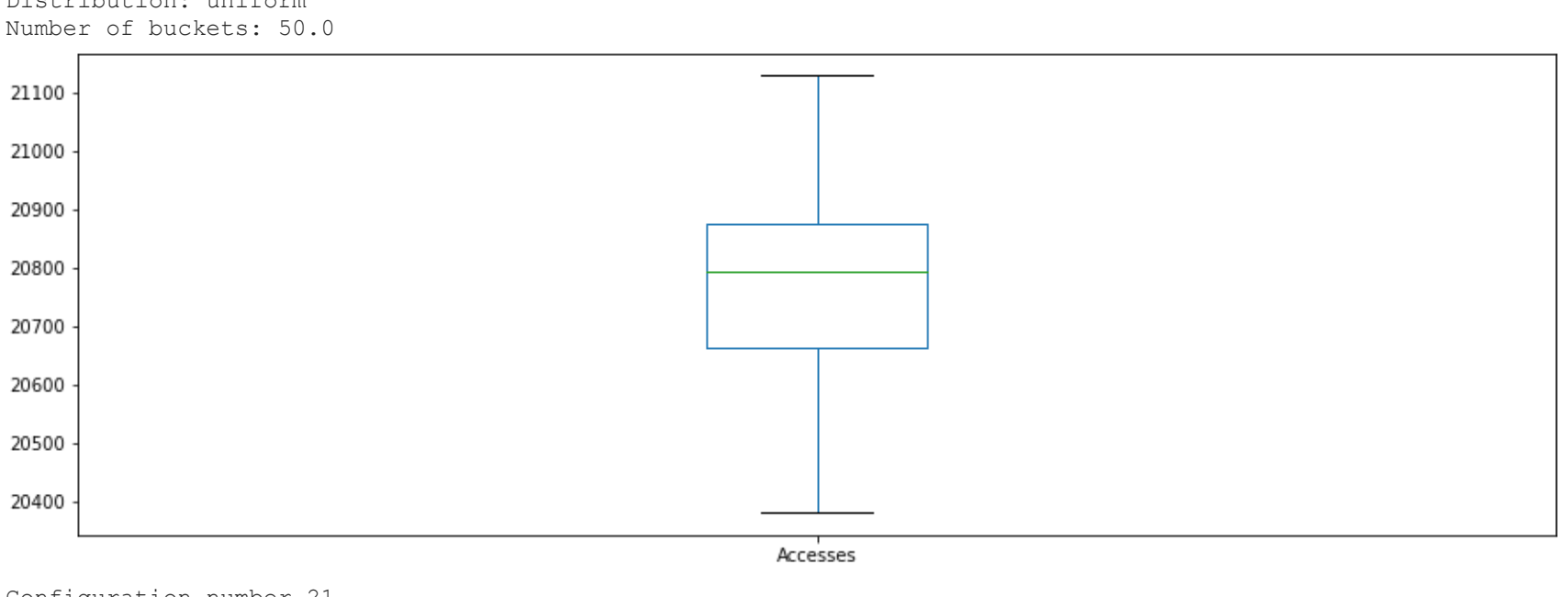

```
Configuration number 15
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



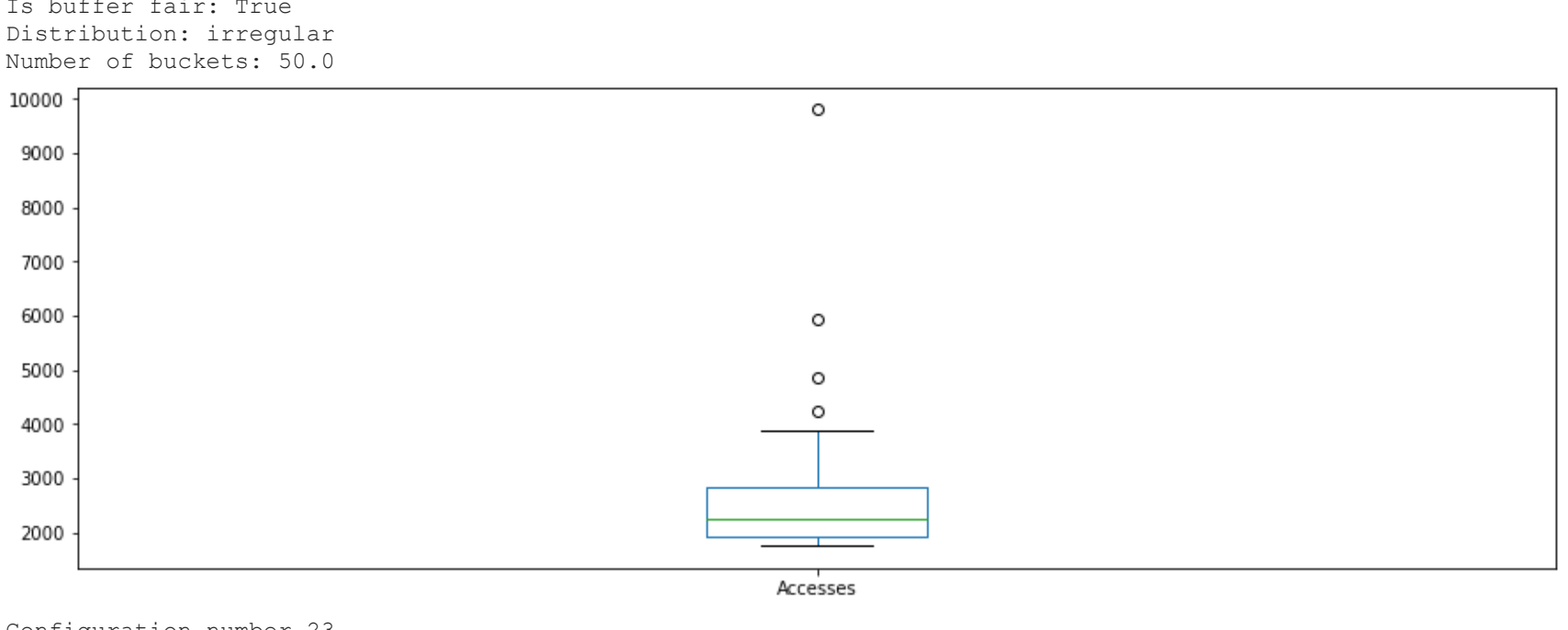

```
Configuration number 17
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



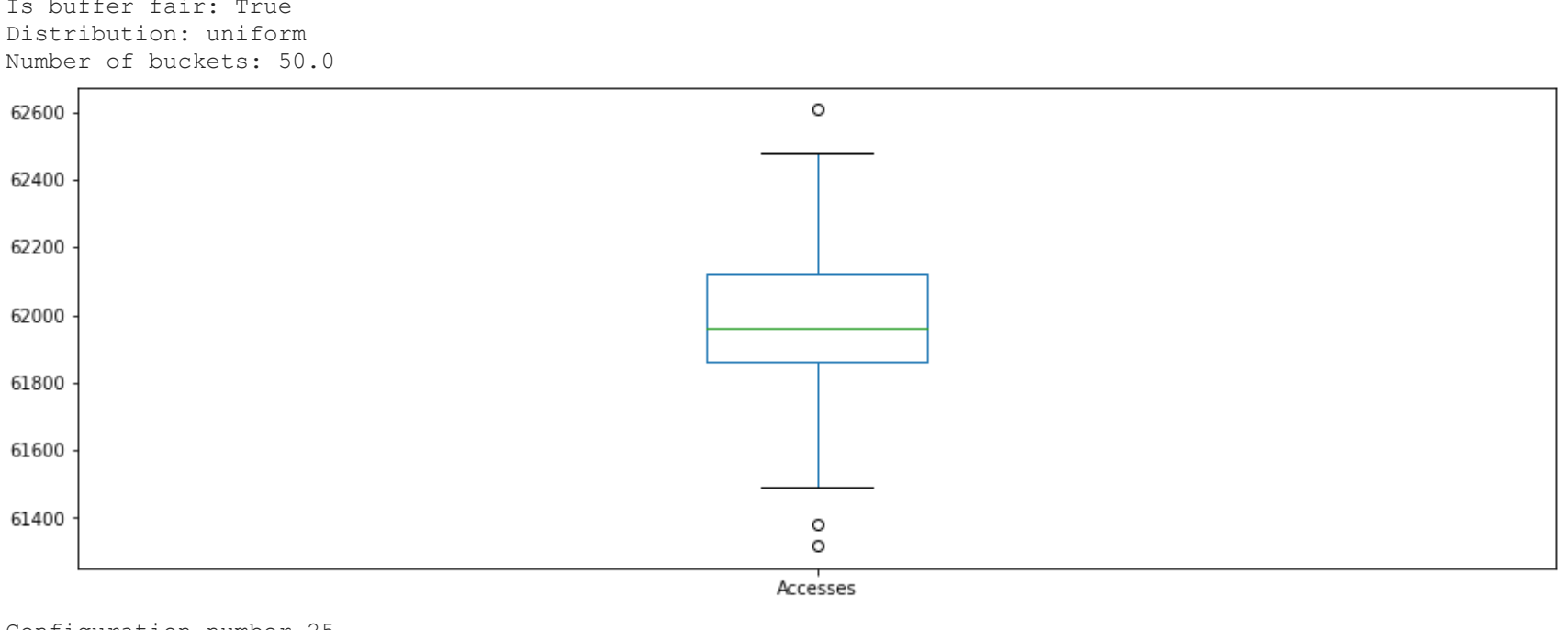

```
Configuration number 19
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



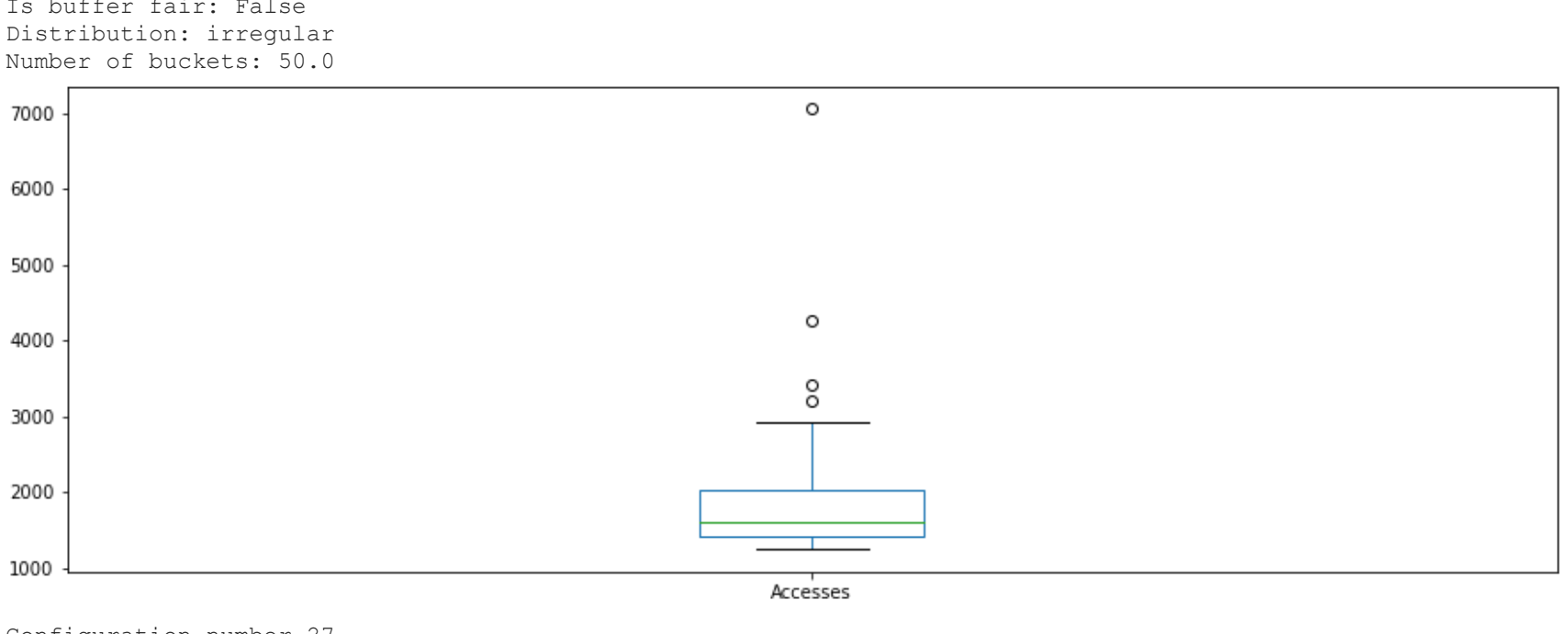

```
Configuration number 21
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



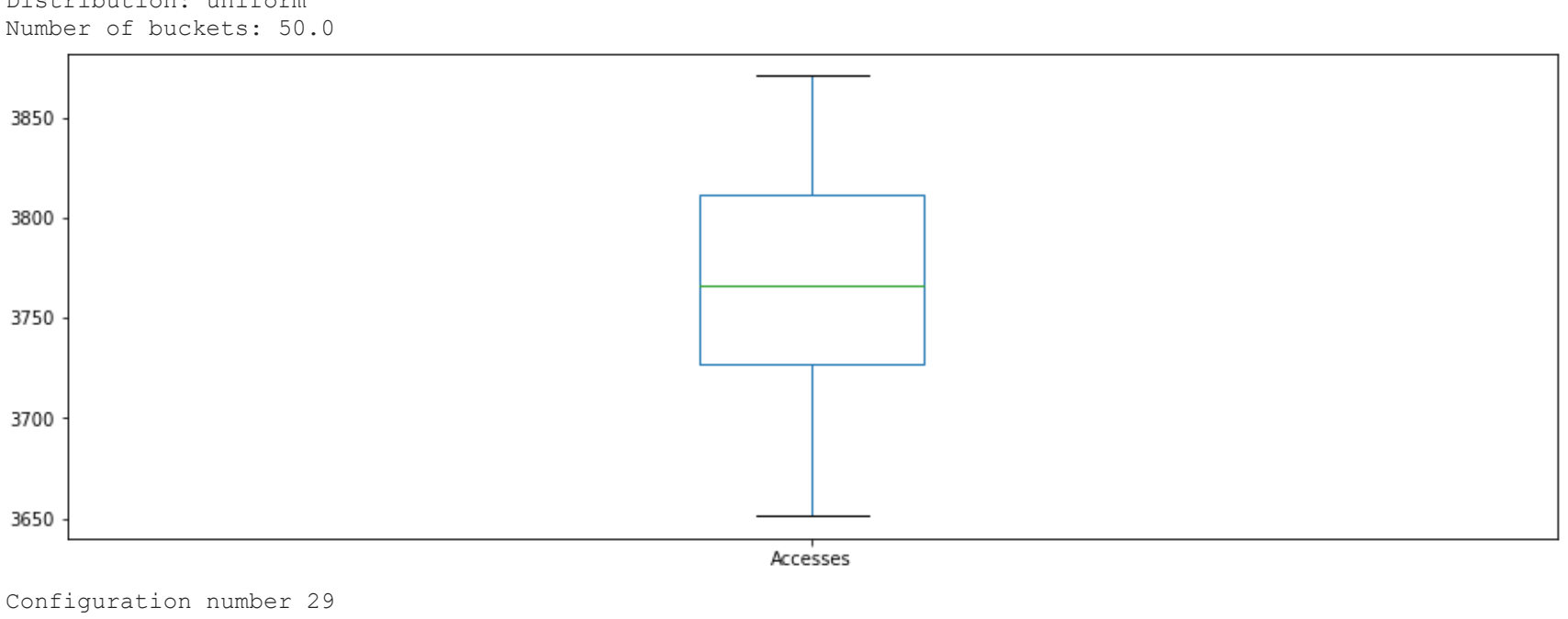

```
Configuration number 23
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



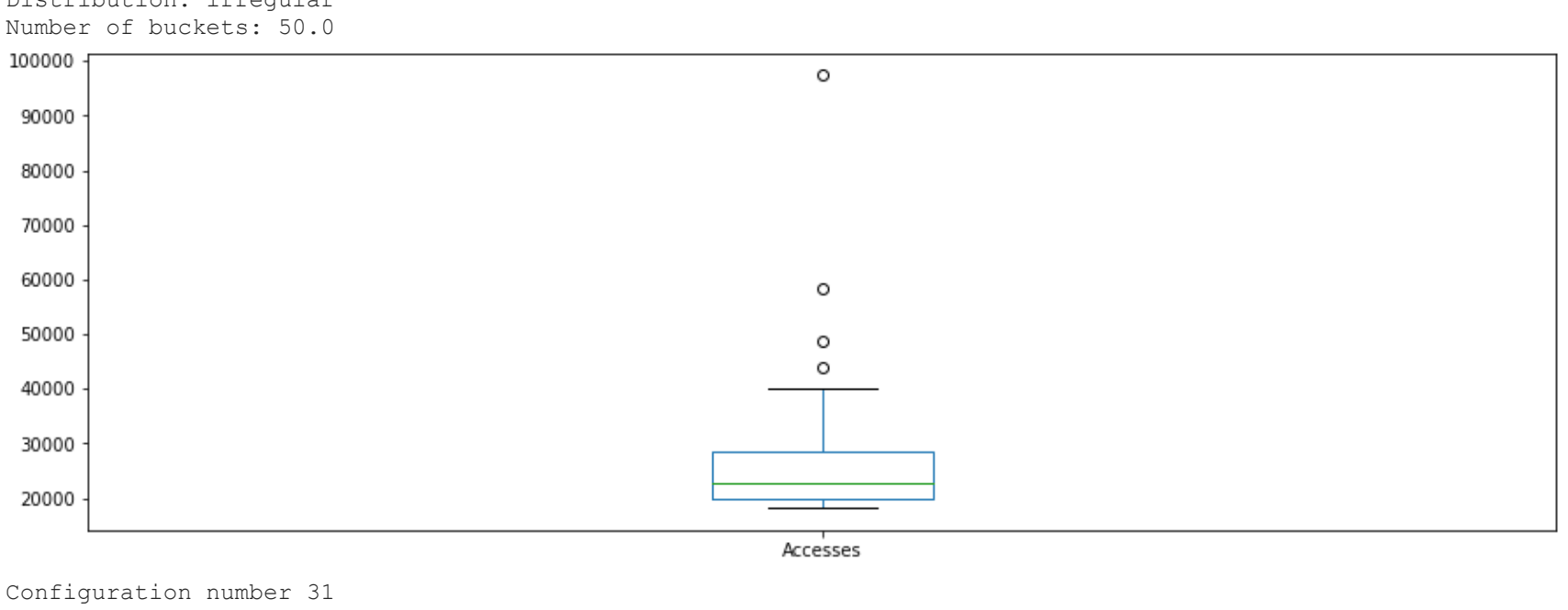

```
Configuration number 25
Buffer size: 100000
Type: consumer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



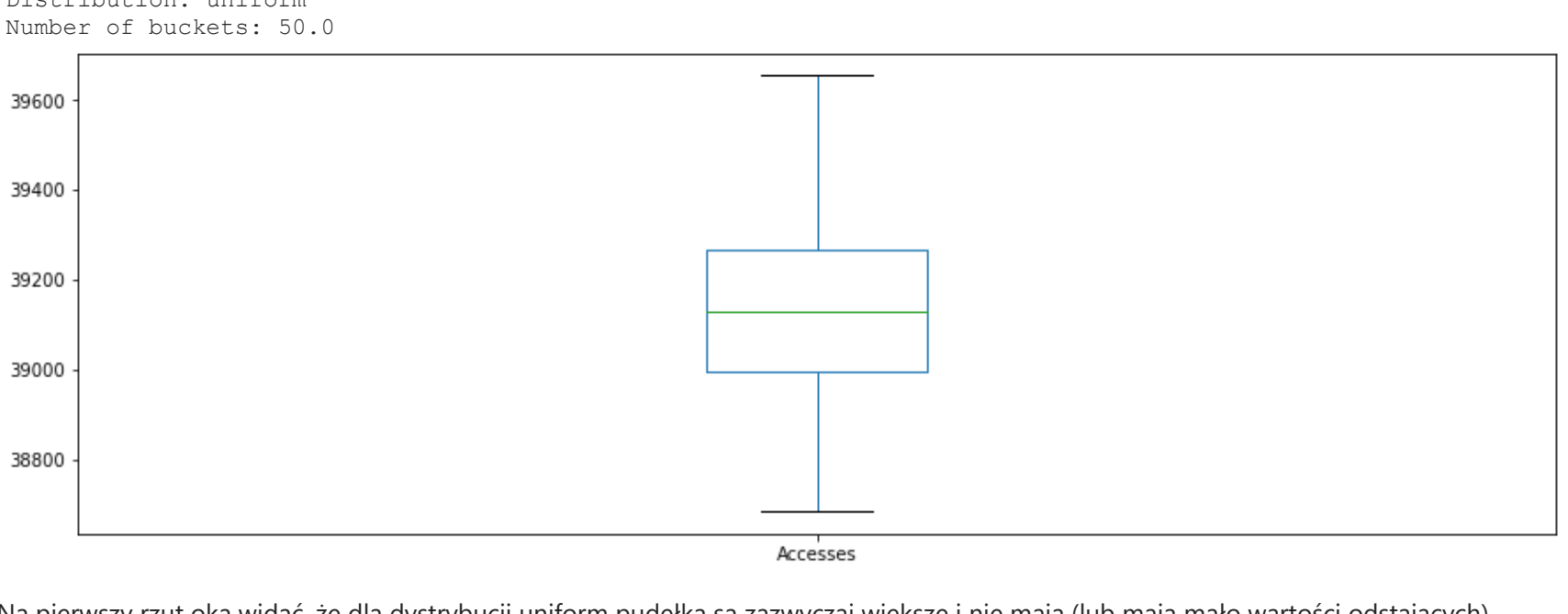

```
Configuration number 27
Buffer size: 100000
Type: consumer
PC Config: 1000+1000
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```




```
Configuration number 29
Buffer size: 100000
Type: consumer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```




```
Configuration number 31
Buffer size: 100000
Type: consumer
PC Config: 1000+1000
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```

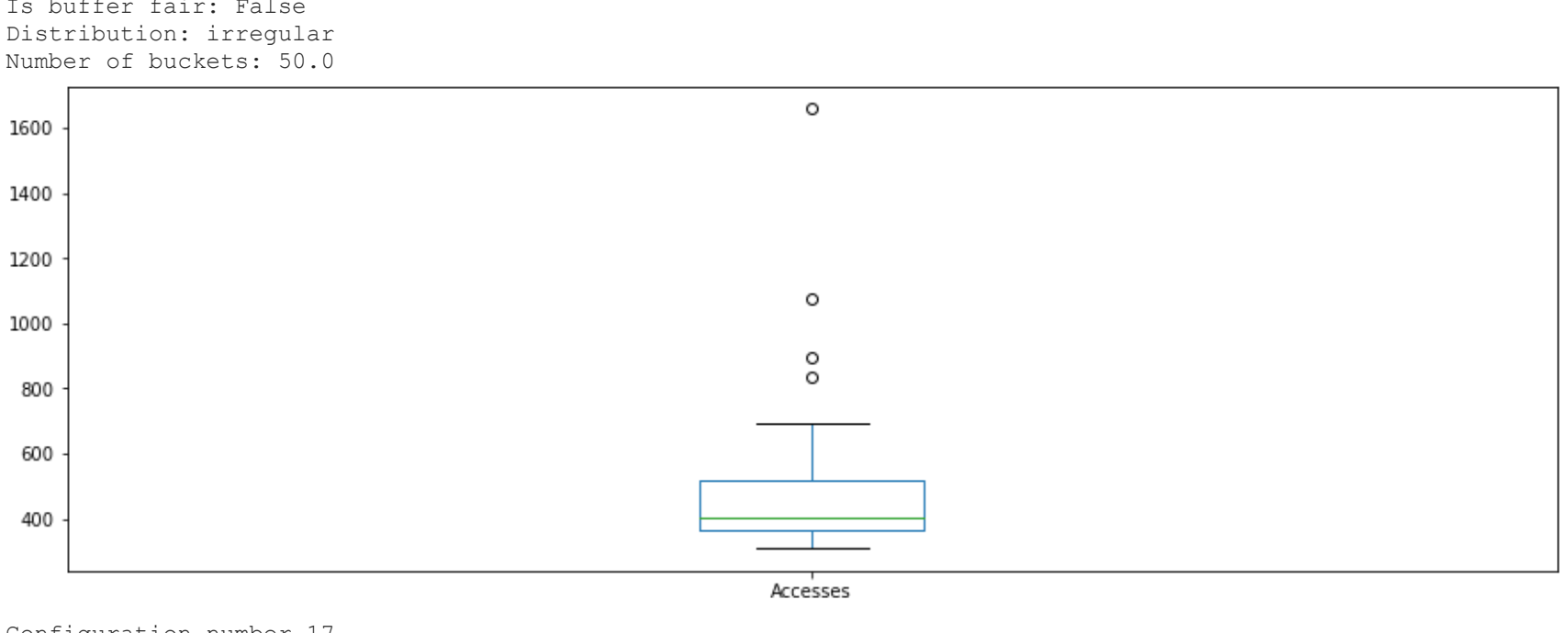


Na pierwszy rzut oka widać, że dla dystrybucji uniform pudełka są zazwyczaj większe i nie mają (lub mają mało wartości odstających). Przyjrzyjmy się teraz bliżej poszczególnym parametrom.

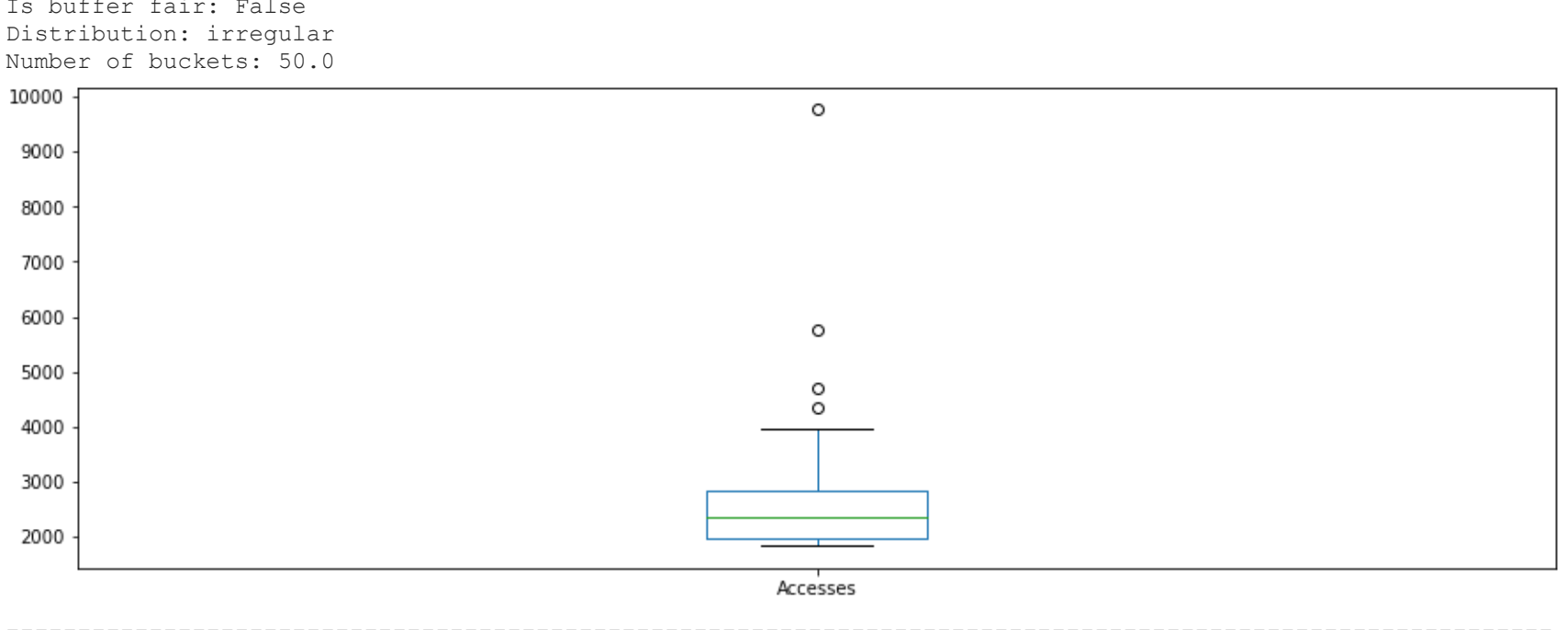
Wielkość bufora

In [113…
```python
draw(data_to_draw[0], 0)
draw(data_to_draw[16], 16)
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
draw(data_to_draw[4], 4)
draw(data_to_draw[20], 20)
```

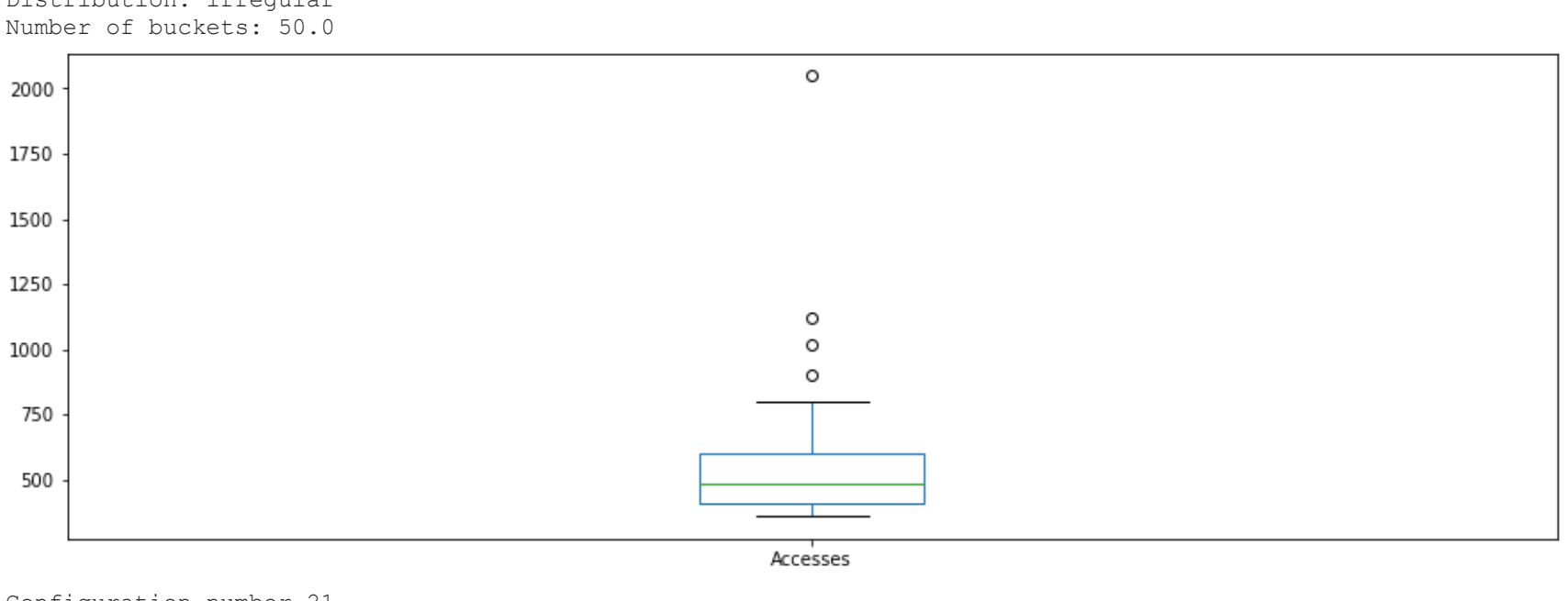
```
Configuration number 1
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



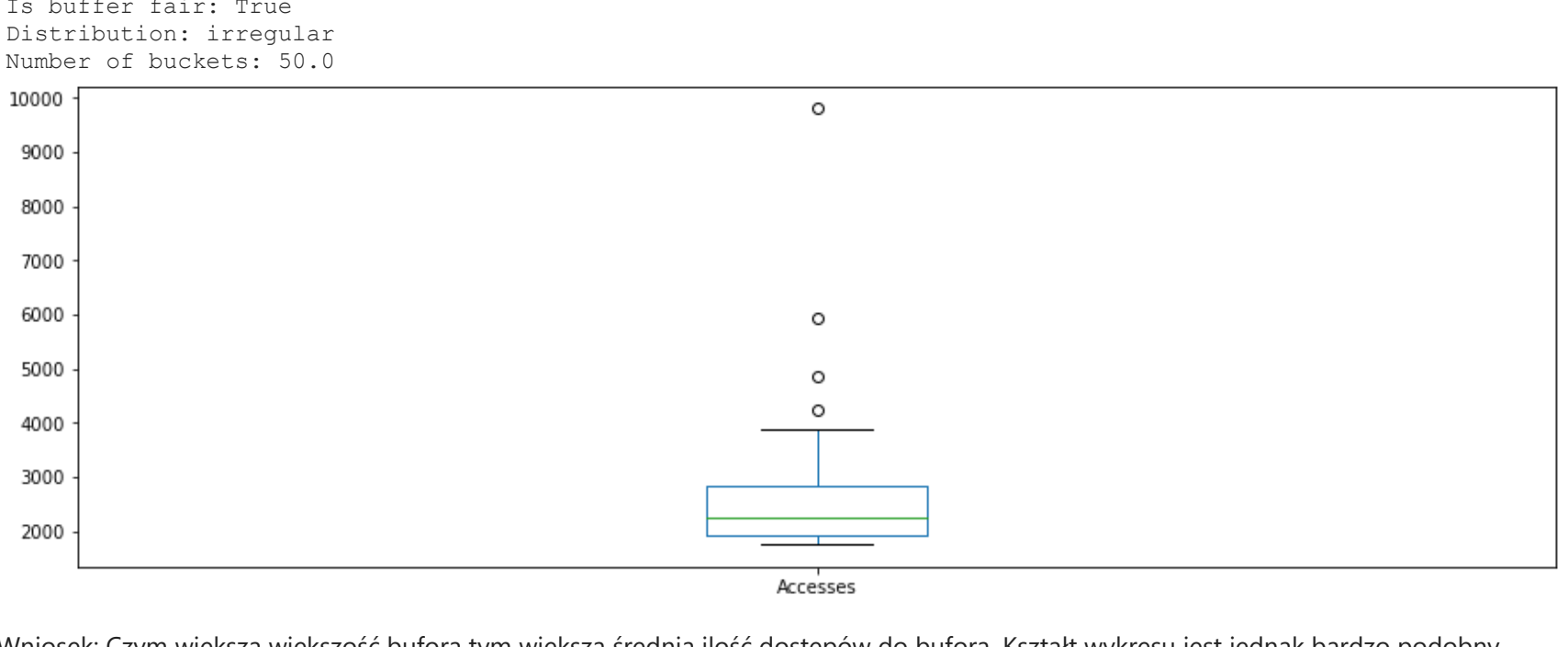

```
Configuration number 17
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```




```
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
Configuration number 5
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```




```
Configuration number 21
Buffer size: 100000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```

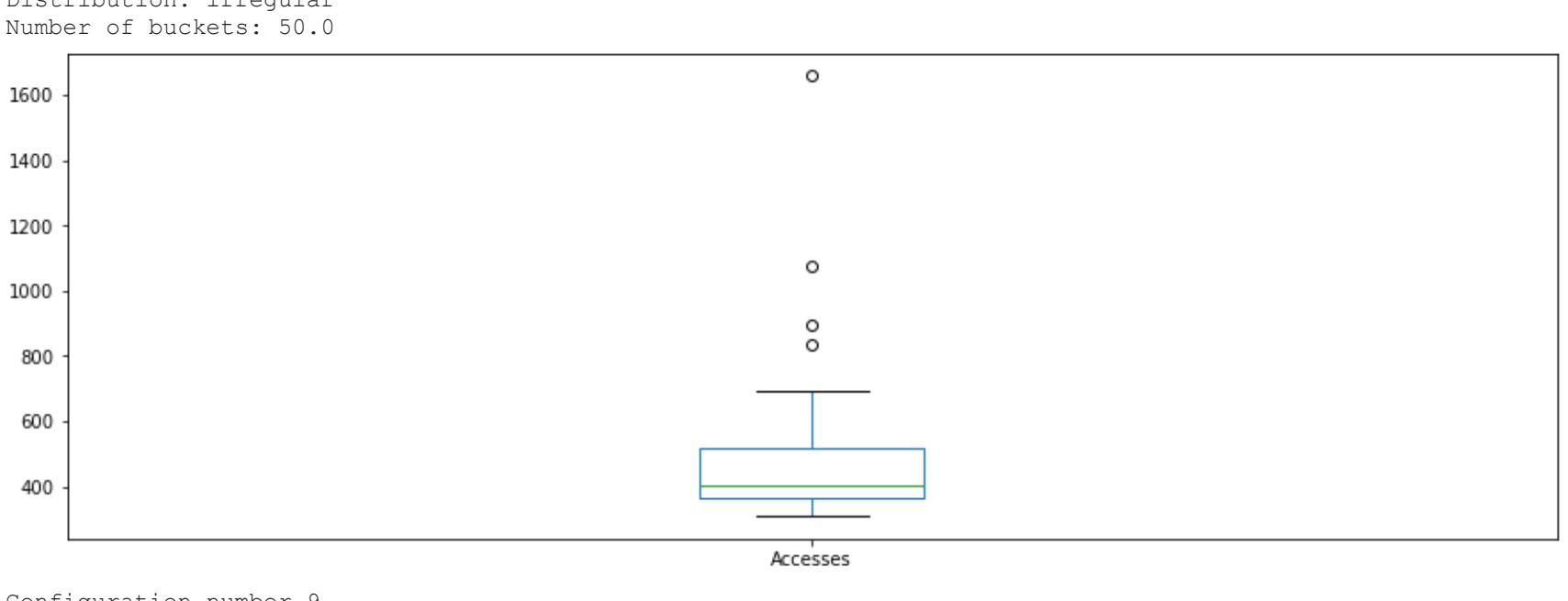


Wniosek: Czym większa większość bufora tym większa średnia ilość dostępów do bufora. Kształt wykresu jest jednak bardzo podobny - ilość buforów rozkłada się zatem podobnie (oczywiście z przeskalowaniem).

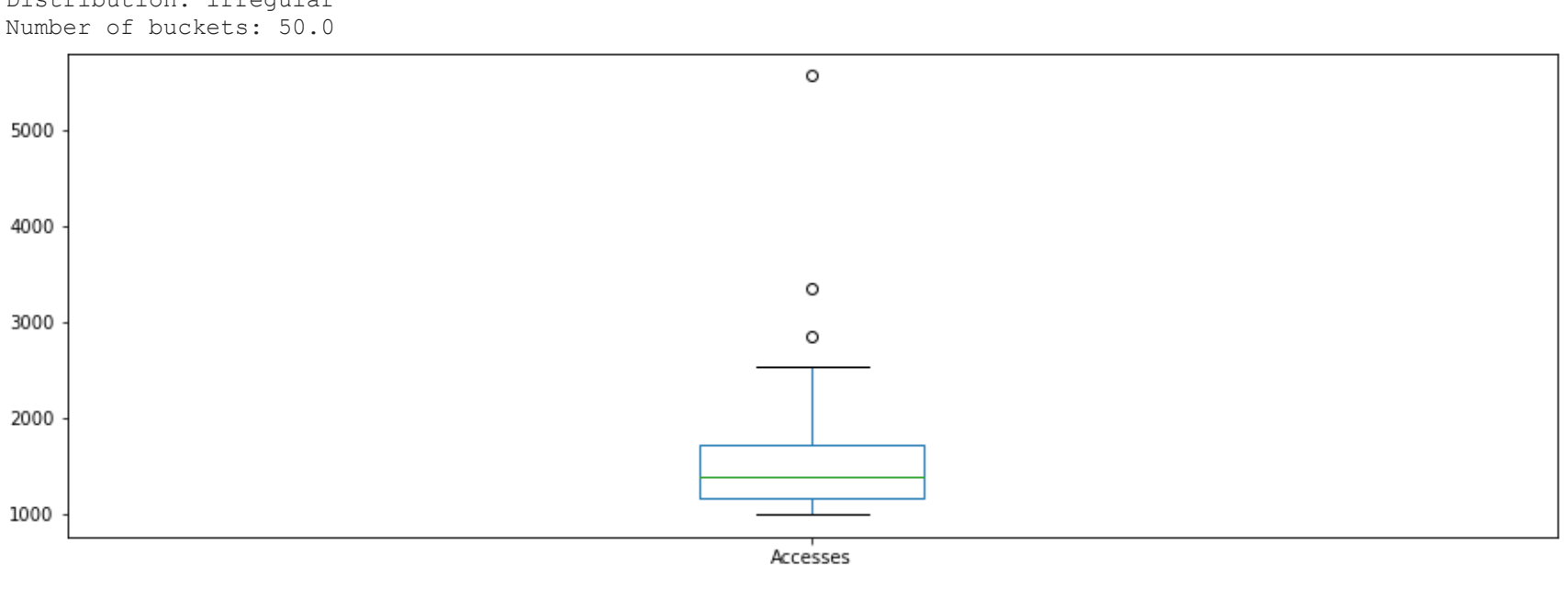
PC Config

In [114…
```python
draw(data_to_draw[0], 0)
draw(data_to_draw[8], 8)
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
draw(data_to_draw[4], 4)
draw(data_to_draw[12], 12)
```

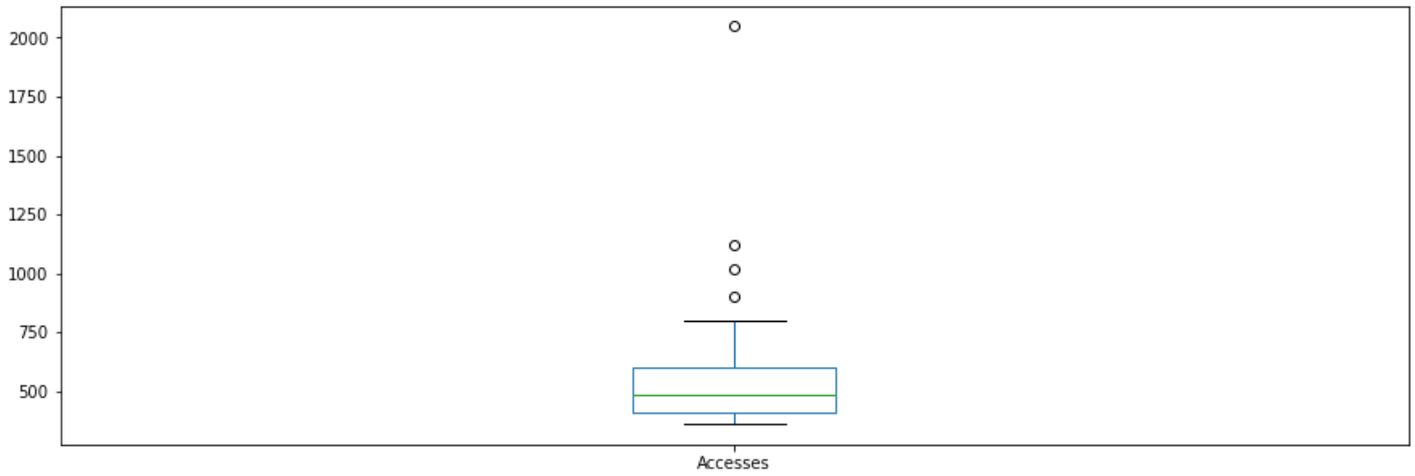
```
Configuration number 1
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



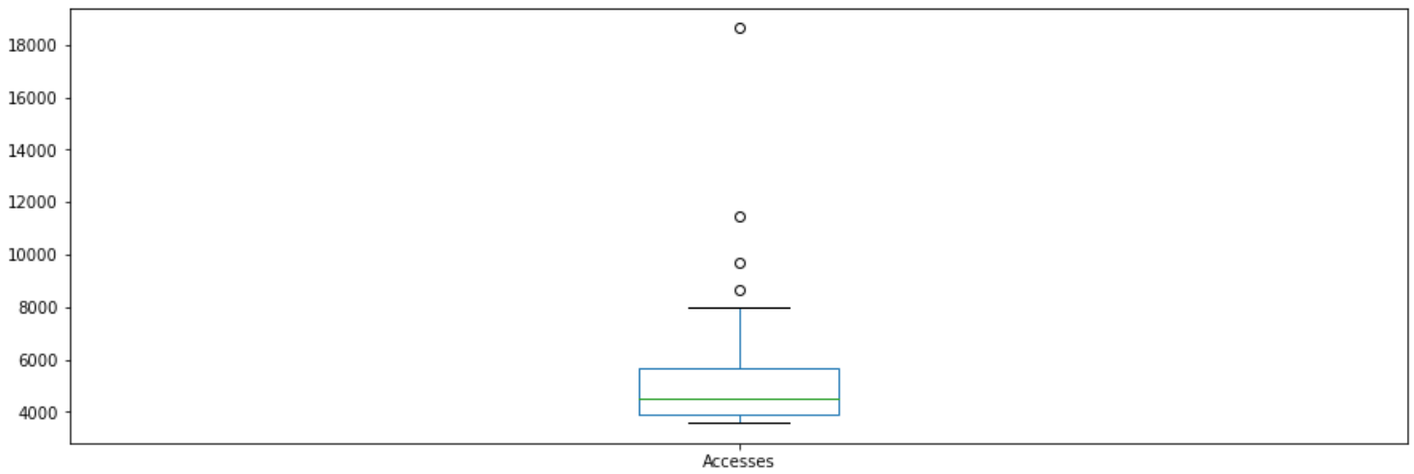

```
Configuration number 9
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```




```
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
----------------------------------------------------------------------------------------------------
--------
Configuration number 5
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```




```
Configuration number 13
Buffer size: 10000
Type: consumer
PC Config: 1000+1000
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```



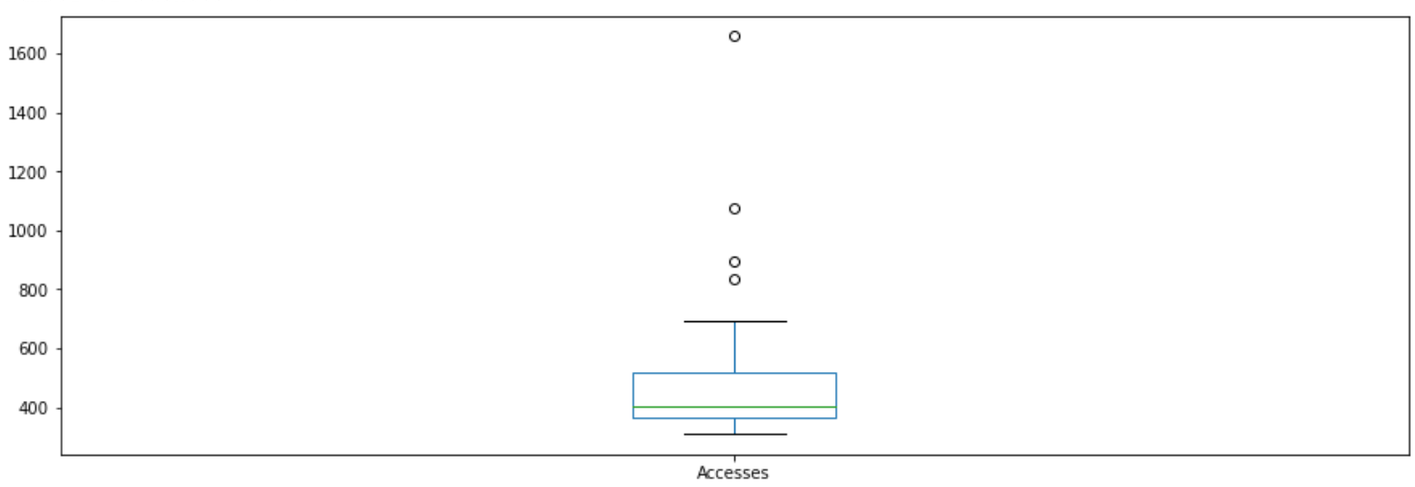
Wniosek: Czym więcej producentów (i konsumentów) tym większa średnia ilość dostępów do bufora. Kształt wykresu jest jednak bardzo podobny - ilość buforów rozkłada się zatem podobnie (oczywiście z przeskalowaniem).

Wariant sprawiedliwy/naiwny

In [115…

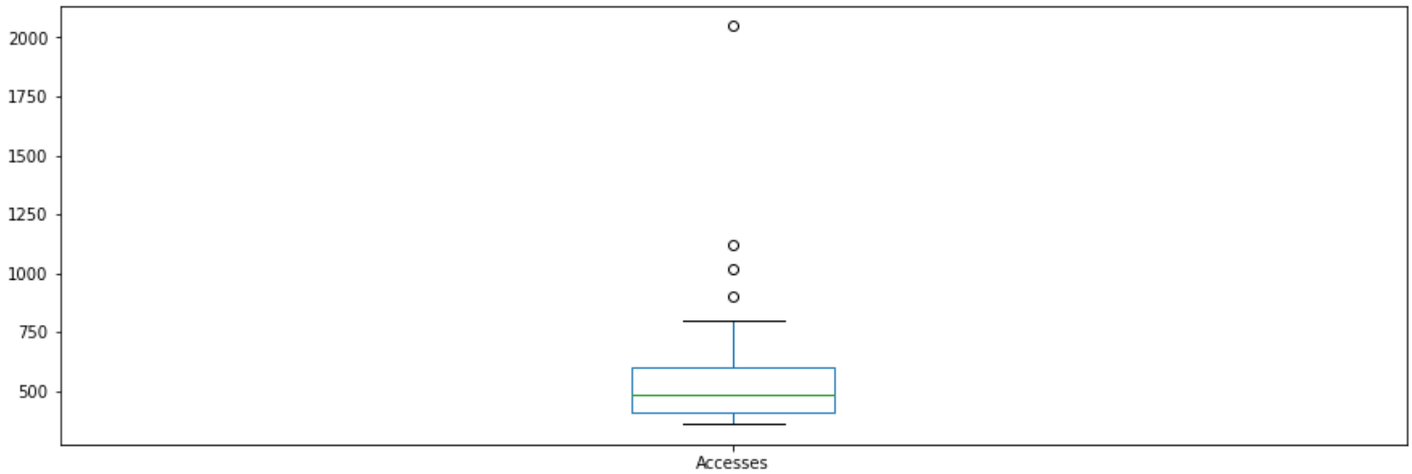
```
draw(data_to_draw[0], 0)
draw(data_to_draw[4], 4)
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
draw(data_to_draw[2], 2)
draw(data_to_draw[6], 6)
```

```
Configuration number 1
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```

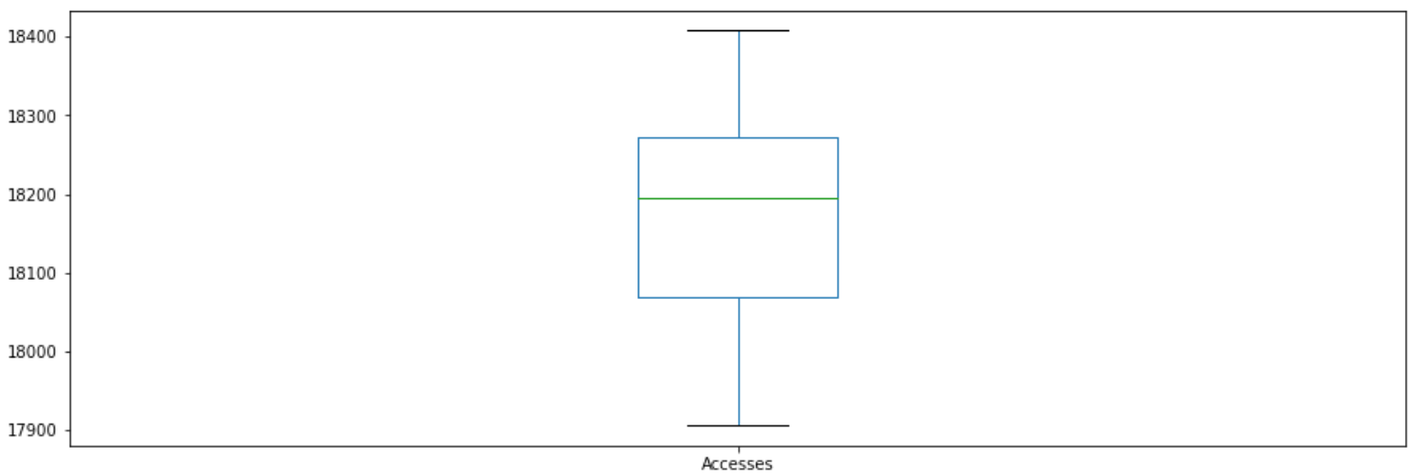



```
Configuration number 5
Buffer size: 10000
```

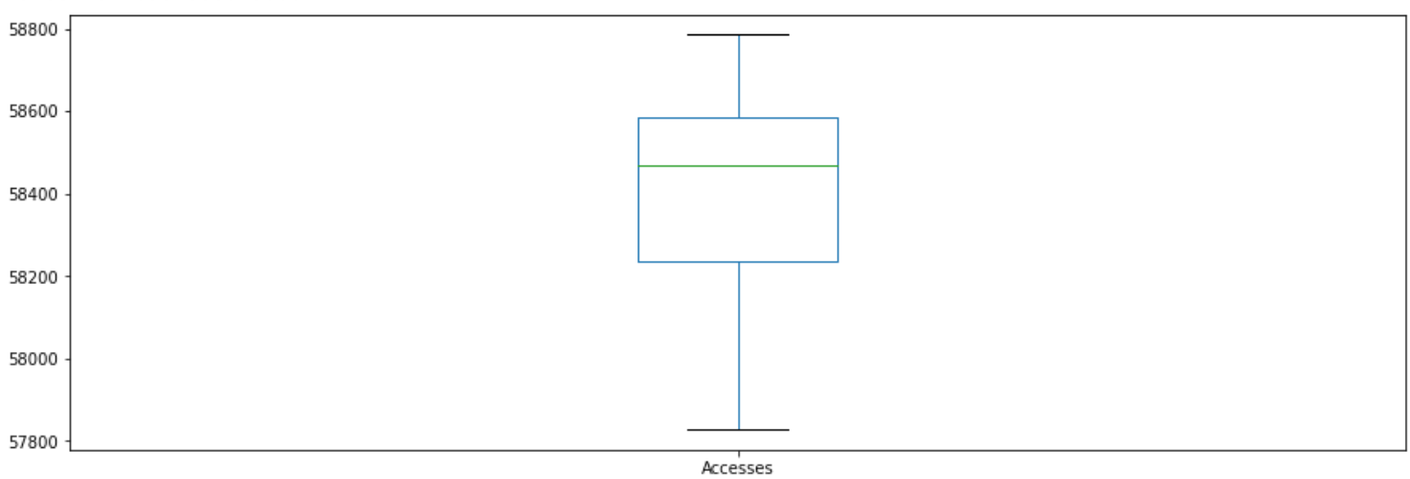
```
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0
```




```
--------------------------------------------------------------------------------------------------------------
--------
--------------------------------------------------------------------------------------------------------------
--------
--------------------------------------------------------------------------------------------------------------
--------
Configuration number 3
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```




```
Configuration number 7
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0
```



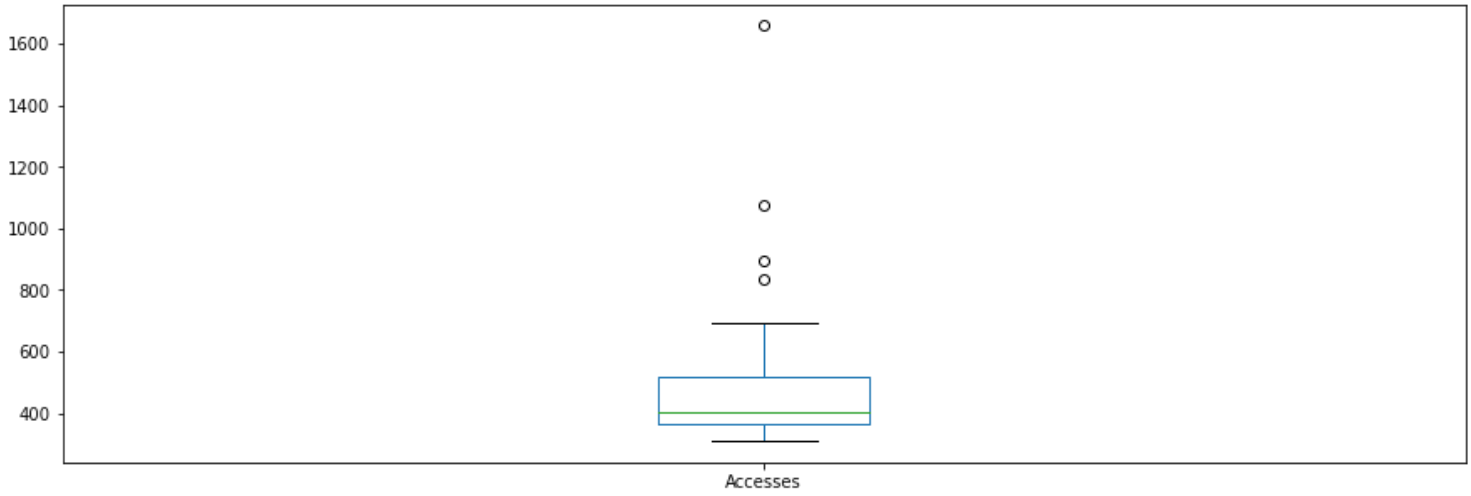
Wniosek: Wariant sprawiedliwy (co ciekawe) pozwala zwiększyć średnią ilość dostępów do bufora. Kształt wykresu jest całkiem podobny - ilość buforów rozkłada się zatem stosunkowo podobnie (oczywiście z przeskalowaniem).

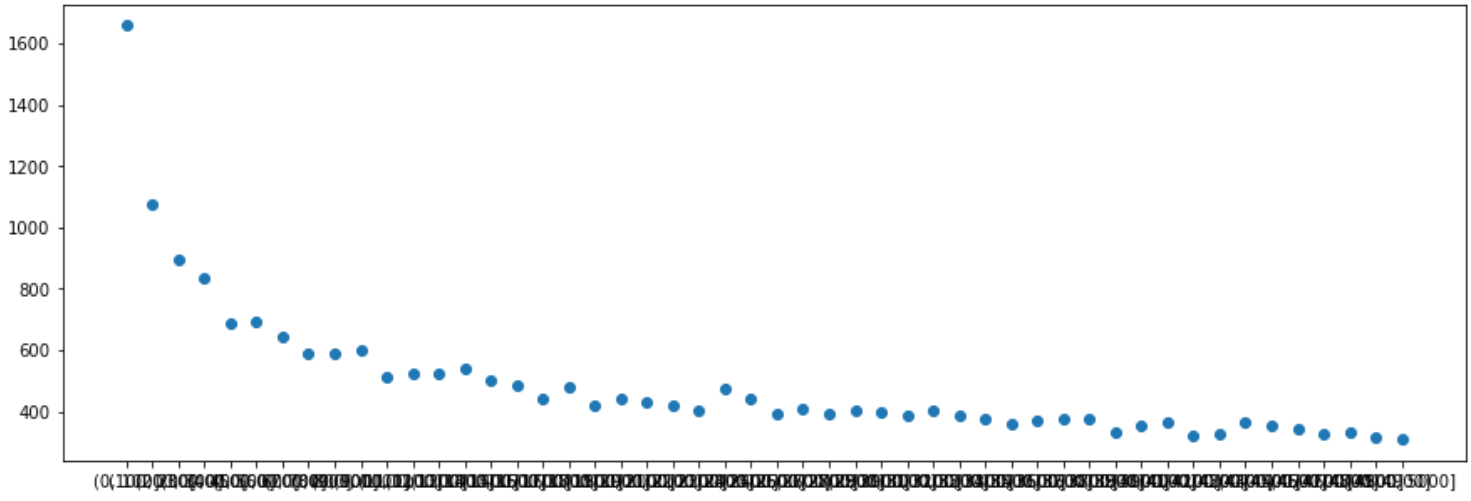
Dystrybucja

In [116…

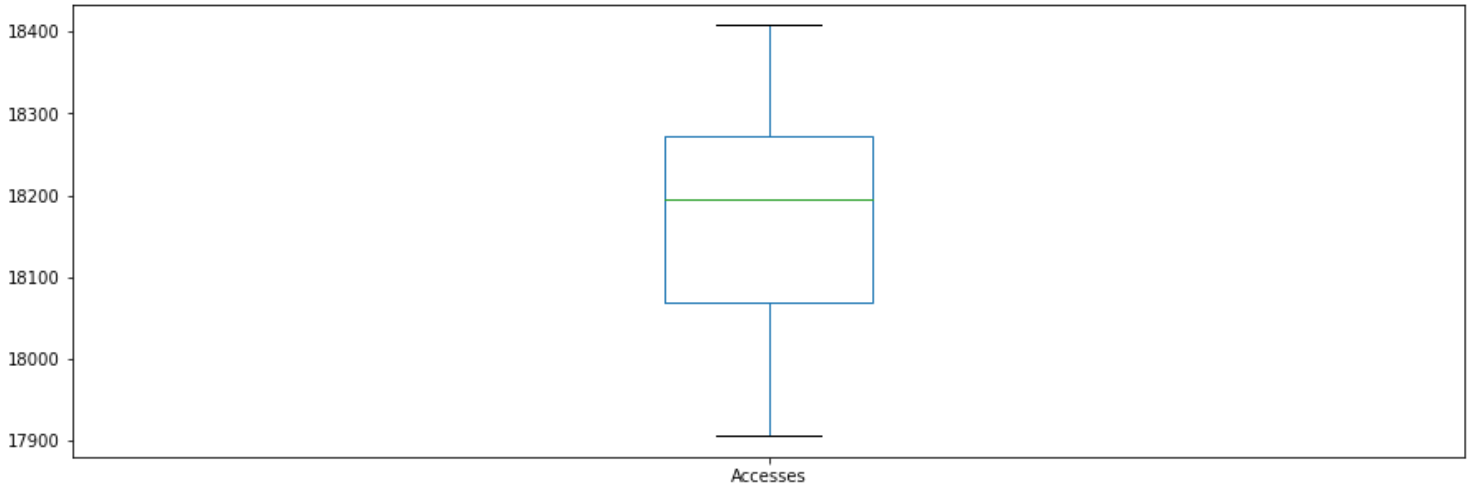
```
draw(data_to_draw[0], 0, print_more = True)
draw(data_to_draw[2], 2, print_more = True)
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
print('----------------------------------------------------------------------------------------------------
draw(data_to_draw[4], 4, print_more = True)
draw(data_to_draw[6], 6, print_more = True)
```

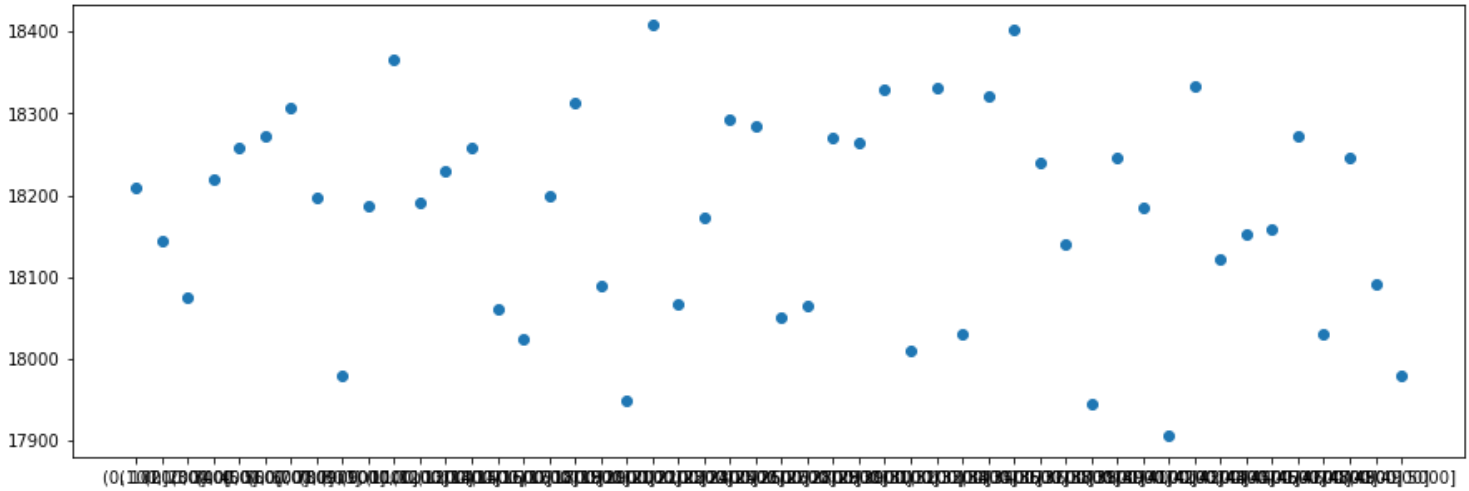
```
Configuration number 1
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: irregular
Number of buckets: 50.0
```



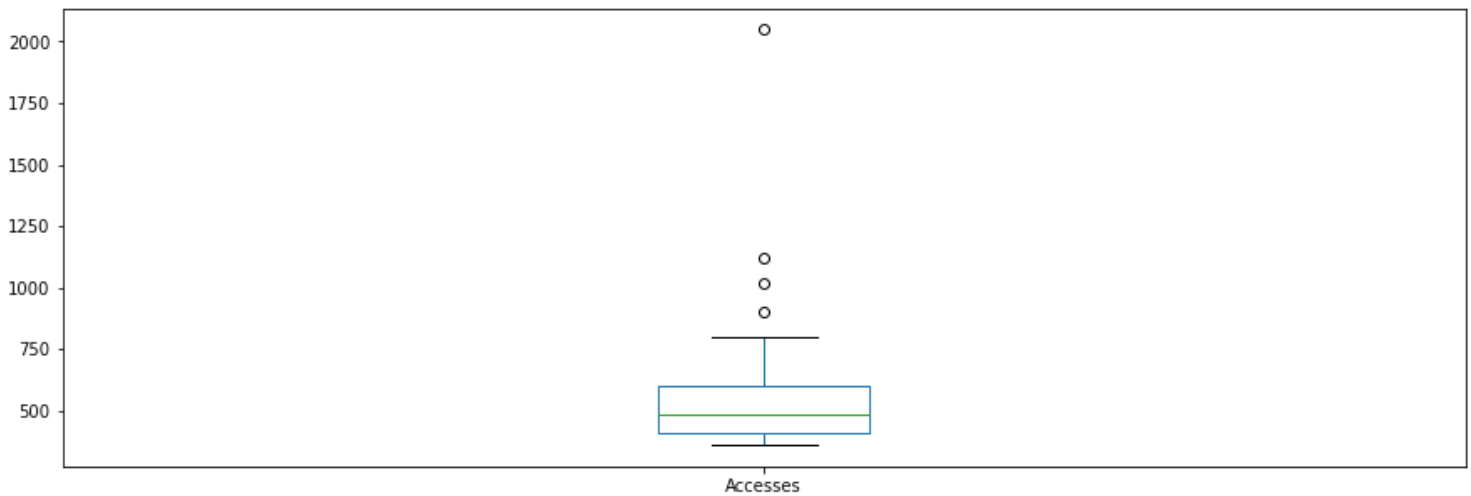


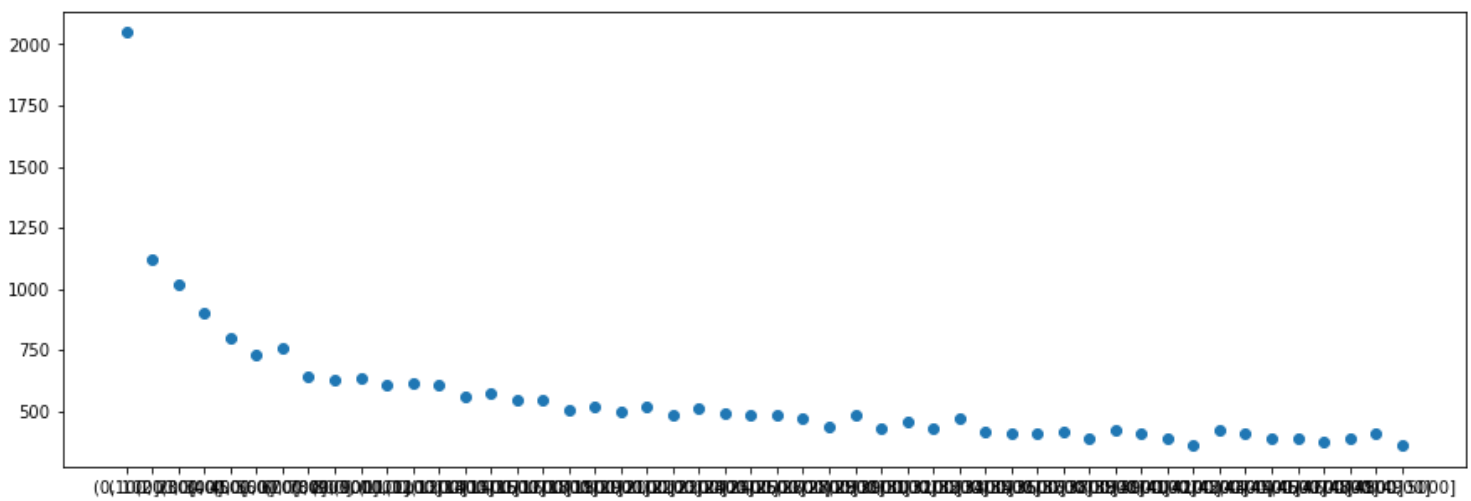

```
Configuration number 3
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: False
Distribution: uniform
Number of buckets: 50.0
```



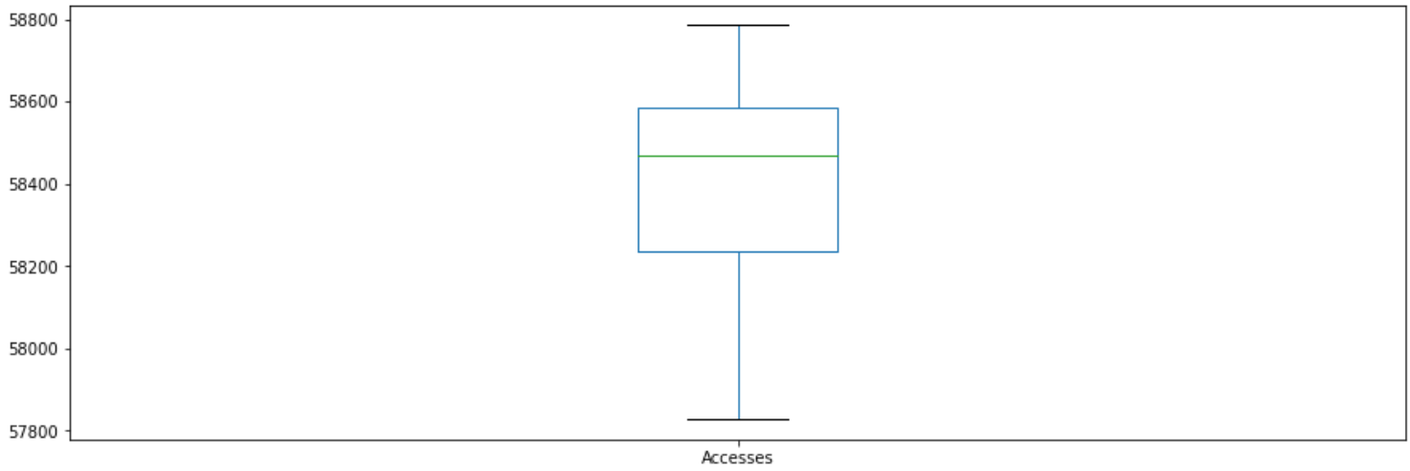


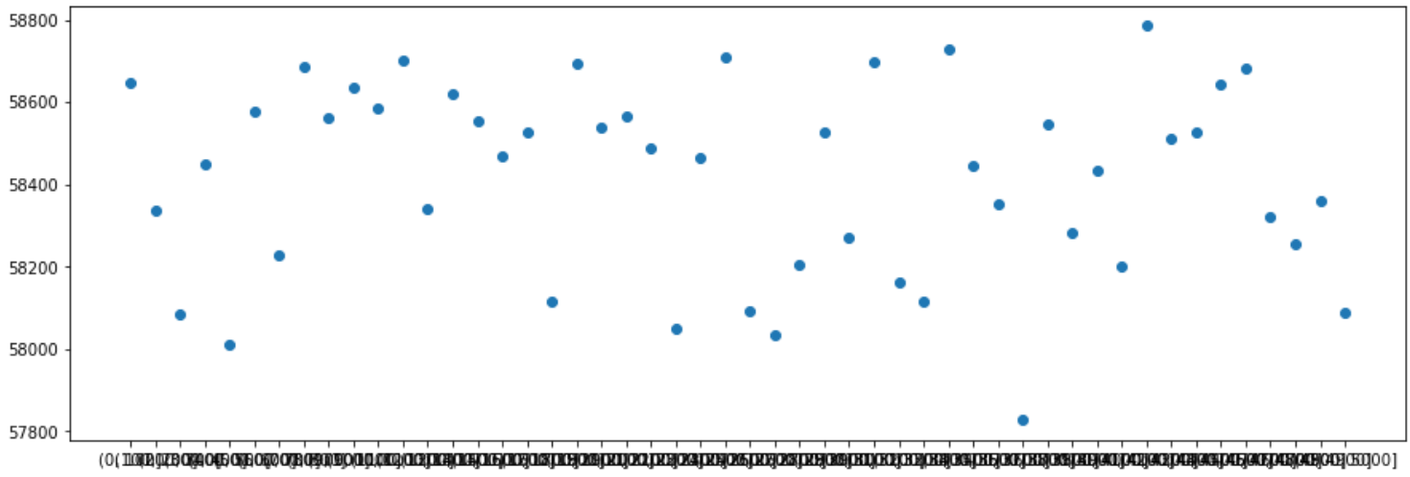
Configuration number 5
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: irregular
Number of buckets: 50.0





Configuration number 7
Buffer size: 10000
Type: consumer
PC Config: 100+100
Is buffer fair: True
Distribution: uniform
Number of buckets: 50.0





Wniosek: Rozkład równomierny pozwolił nam BARDZO MOCNO zwiększyć średnią ilość dostępów do bufora. Kształt wykresu jest zupełnie inny - wynika to z tego, że przy rozkładzie równomiernym więcej wartości jest dopuszczonych (a więc pudełko większe). Pomocniczy wykres pokazuje, że faktycznie rozkład w jednym wypadku jest równomierny a w drugim mniejsze liczby mają większe prawdopodobieństwo.

3. Wnioski

- Problem producentów oraz konsumentów pozwala dobrze zapoznać się z pojęciem współbieżności
- Problem producentów oraz konsumentów pozwala zrozumieć różnicę między naiwnym a sprawiedliwym podejściem
- Wykonane doświadczenie pozwoliło zobaczyć jak poszczególne parametry naszego problemu wpływają na ilość dostępów wątków do wspólnego bufora
- Ilość dostępów do bufora zwiększa się wraz z:
  - wielkością bufora
  - ilością wątków (producentów i konsumentów)
  - wariantem sprawiedliwym (względem naiwnego co wydaje się nielogiczne)
  - rozkładem równomiernym