

# **Parallel Programming**

## **HW2**

### **Java Atomic Library**

**Meyssam Rostamzadeh**

**1402/09/05**



# Variable Definition:

```
//for Non-Atomic version
```

```
public static ArrayList<Integer> array = new ArrayList<>();
```

```
//for Atomic version
```

```
private static final AtomicIntegerArray sharedArray = new AtomicIntegerArray(ARRAY__SIZE);
```

```
private static final AtomicInteger arrayLength = new AtomicInteger(0);
```



# Thread body:

```
static class ArrayUpdater implements Runnable {
    @Override
    public void run() {
        // non-atomic
        int length= array.size();
        //      System.out.println(Thread.currentThread().threadId() + " current L
        while (length < ARRAY__SIZE) {
        //      System.out.println(Thread.currentThread().threadId() + " addin
            array.add(length);
            length= array.size();
        //      System.out.println(Thread.currentThread().threadId() + " now a

        ////      //atomic version
        int currentLength = arrayLength.get();
        while (currentLength < ARRAY__SIZE) {
        ////      System.out.println(Thread.currentThread().threadId()+ " arra
            sharedArray.set(currentLength , currentLength + 1);
            currentLength = arrayLength.getAndIncrement();
        ////      System.out.println(Thread.currentThread().threadId()+ " arra
```



# Final Array Content:

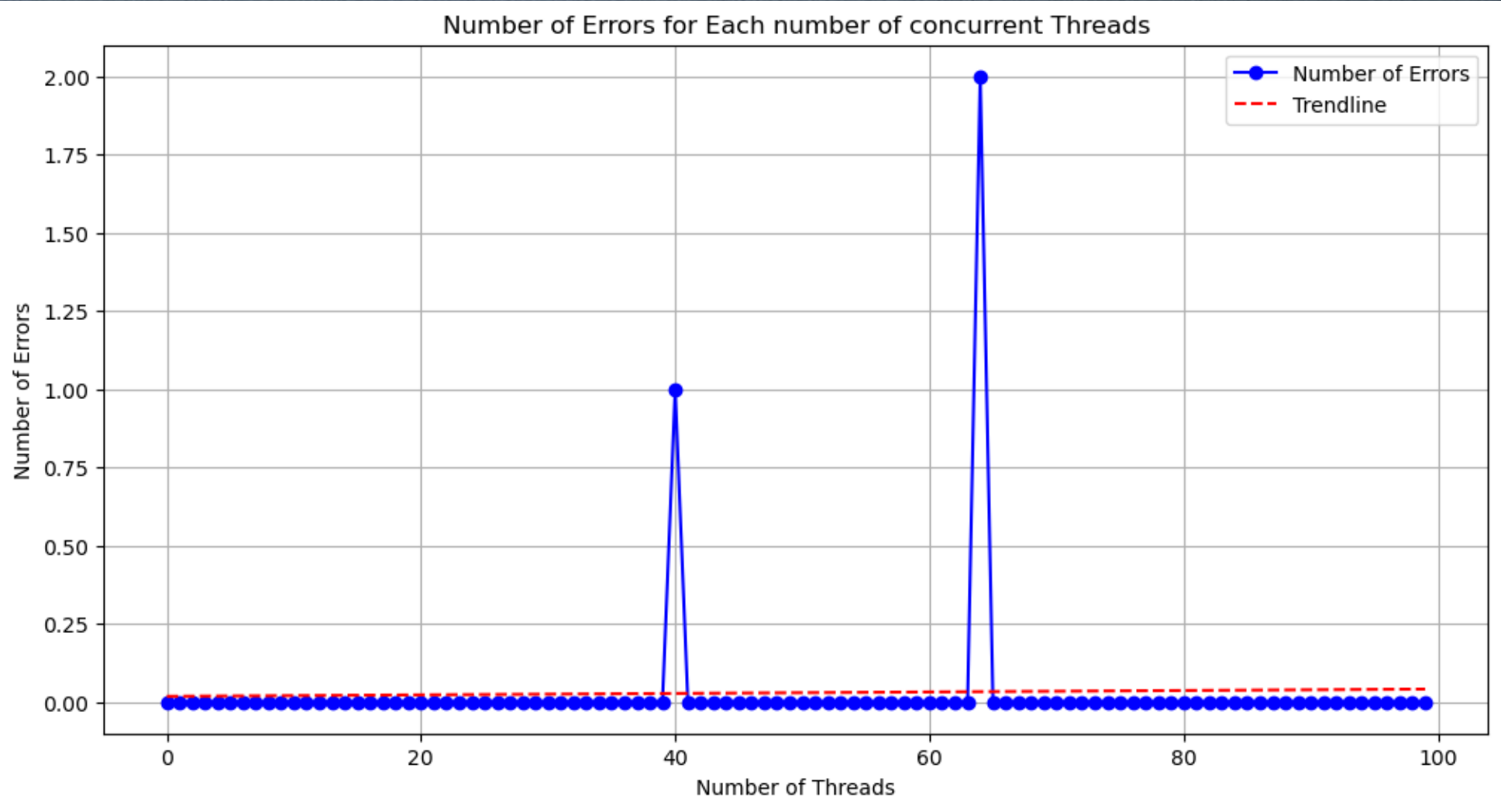
Final Array Length after 3 threads: 100												
Array contents	?- ?:	1	2	3	4	5	6	7	8	9	10	
Array contents	??- ??:	11	12	13	14	15	16	17	18	19	20	
Array contents	??- ??:	21	22	23	24	25	26	27	28	29	30	
Array contents	??- ??:	31	32	33	34	35	36	37	38	39	40	
Array contents	??- ??:	41	42	43	44	45	46	47	48	49	50	
Array contents	??- ??:	51	52	53	54	55	56	57	58	59	60	
Array contents	??- ??:	61	62	63	64	65	66	67	68	69	70	
Array contents	??- ??:	71	72	73	74	75	76	77	78	79	80	
Array contents	??- ??:	81	82	83	84	85	86	87	88	89	90	
Array contents	??- ??:	91	92	93	94	95	96	97	98	99	100	
Final Array Length after 4 threads: 100												

# Final Array Content:

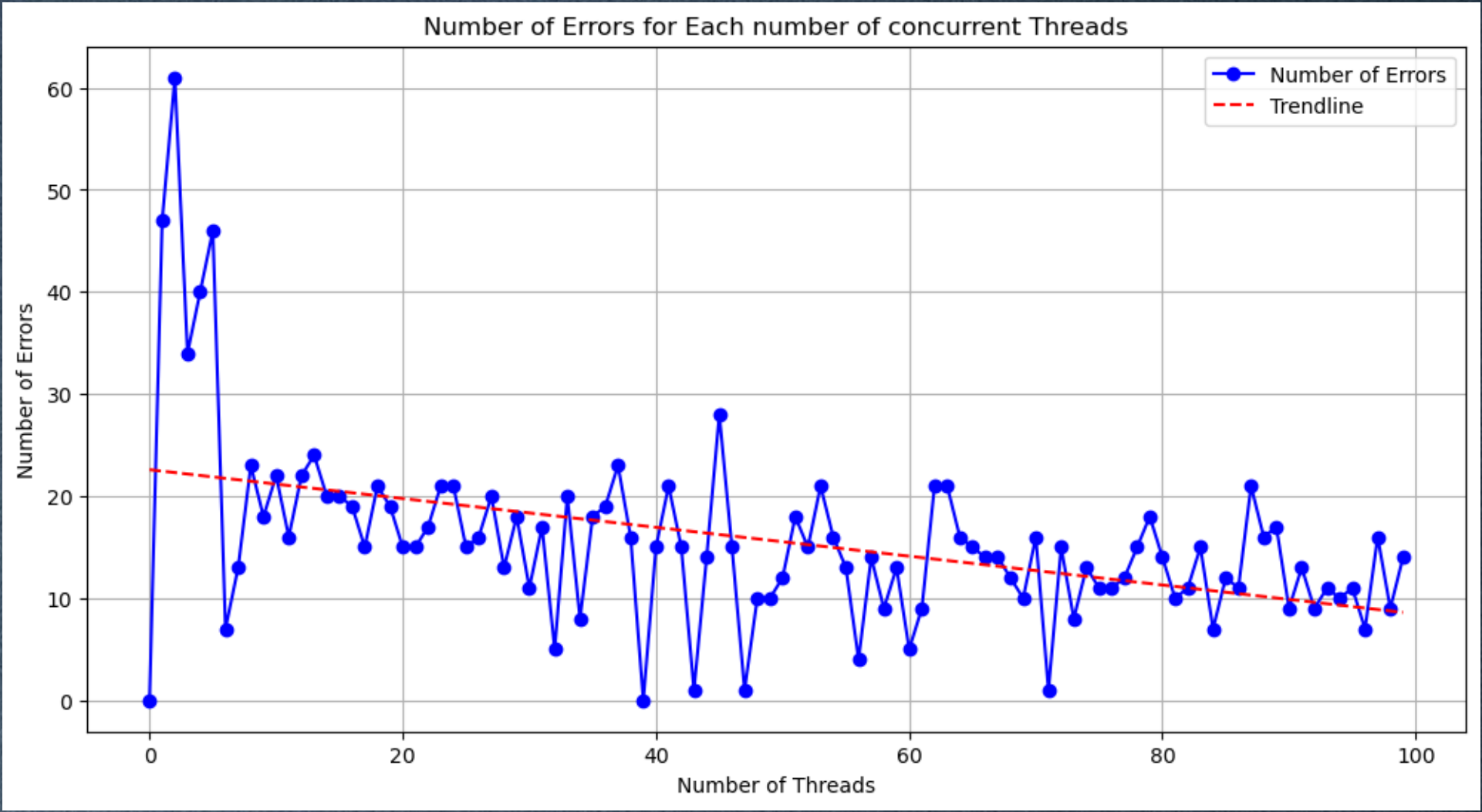
Final Array Length after 3 threads: 102												
Array contents	?- ?:	0	1	2	3	4	5	6	7	8	9	
Array contents	??- ??:	10	11	12	13	14	15	11	16	17	18	
Array contents	??- ??:	19	20	21	0	22	23	24	27	28	25	
Array contents	??- ??:	26	31	29	30	32	35	34	33	38	37	
Array contents	??- ??:	36	41	40	39	43	42	46	44	48	47	
Array contents	??- ??:	49	51	52	53	54	55	56	57	58	59	
Array contents	??- ??:	60	61	62	63	64	65	66	67	68	69	
Array contents	??- ??:	70	71	72	73	74	75	76	77	78	79	
Array contents	??- ??:	80	81	82	83	84	85	86	87	88	89	
Array contents	??- ??:	45	50	90	91	94	92	93	97	95	96	
Array contents	???-???:	98	99									



**Solutions:**      **Non-Atomic**      **0\_print function inside Thread**

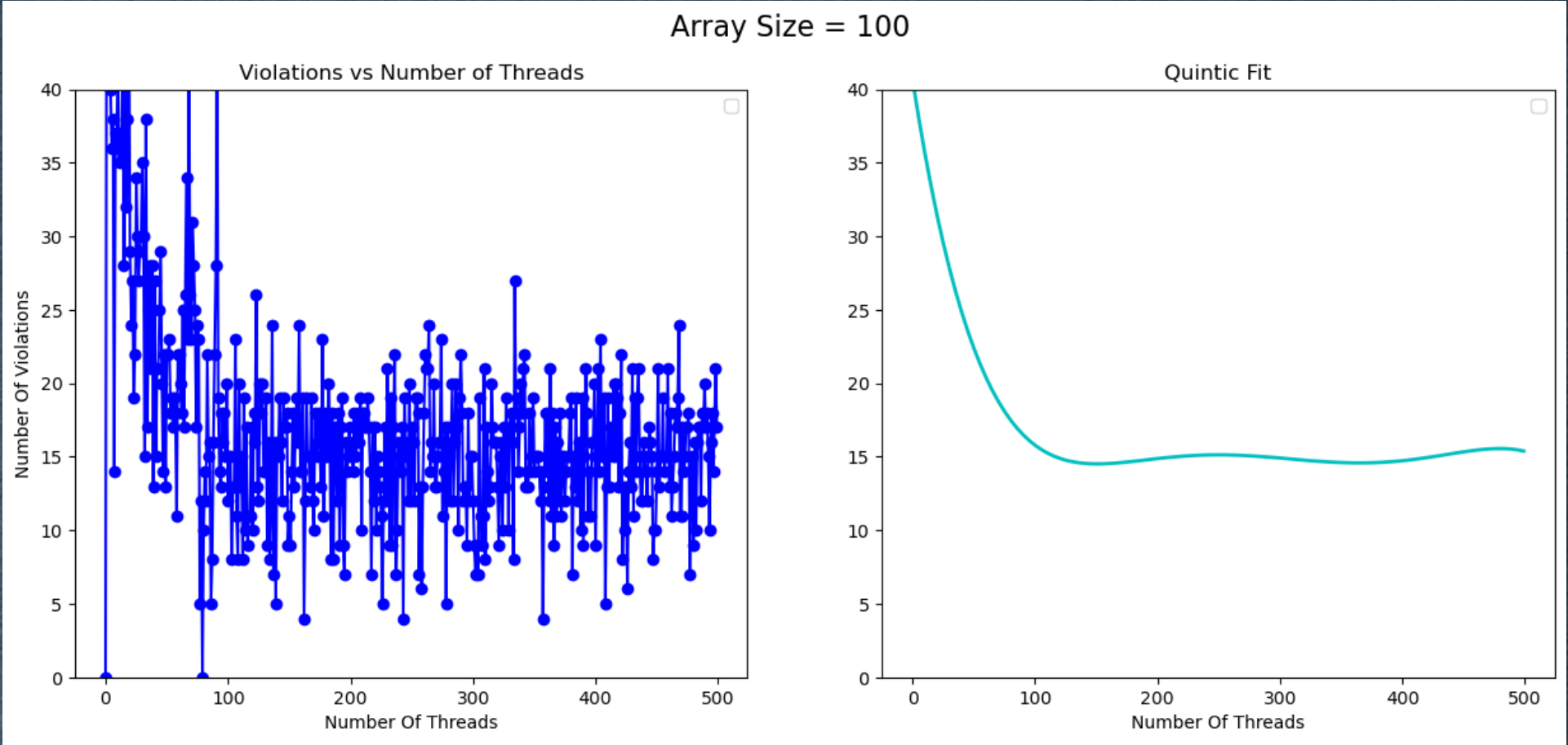


# Solutions: Non-Atomic 1\_print function inside Thread



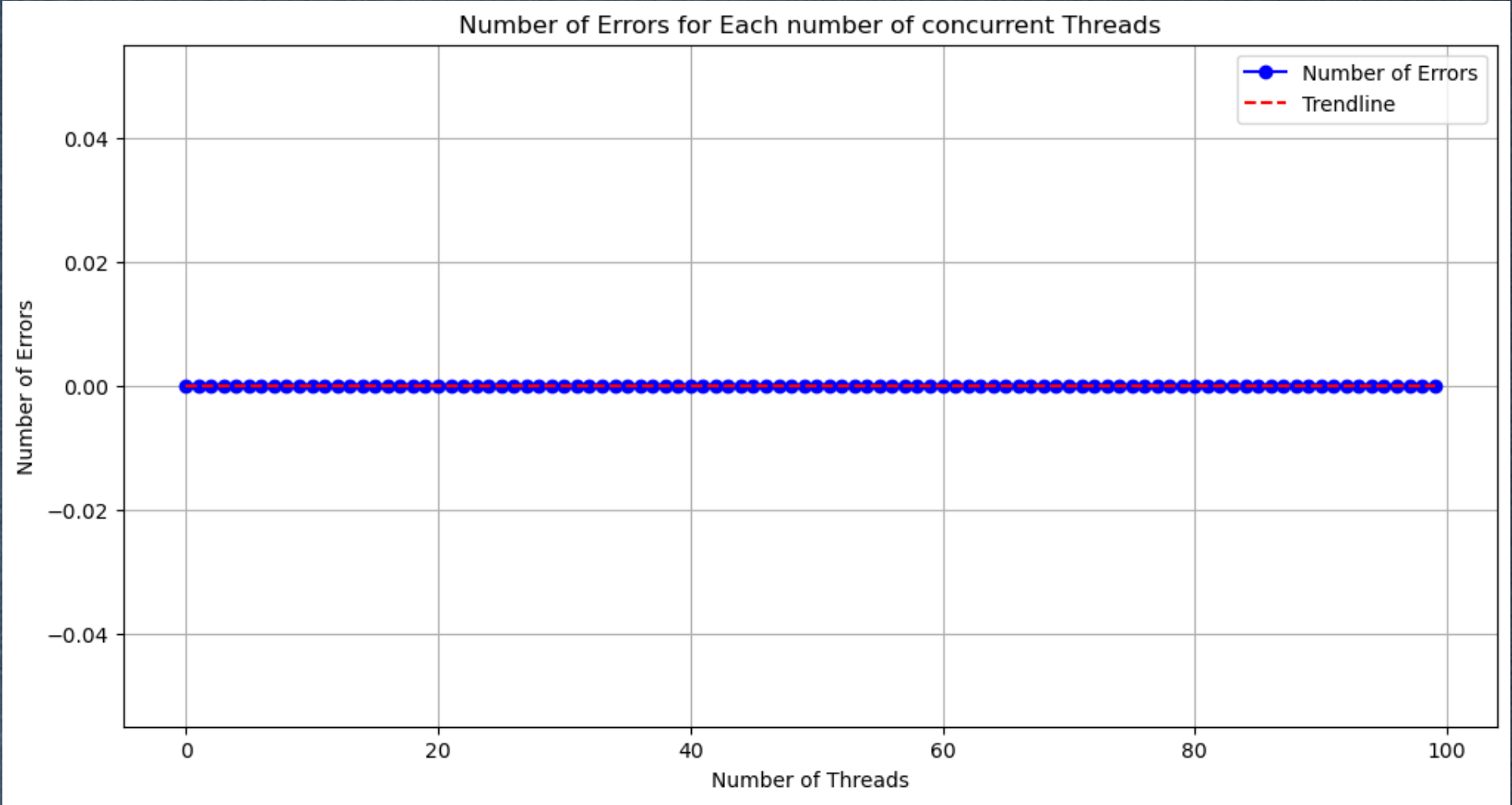


# Solutions: Non-Atomic 2\_print function inside Thread





# Solutions: Atomic 0\_1\_2\_print function inside Thread



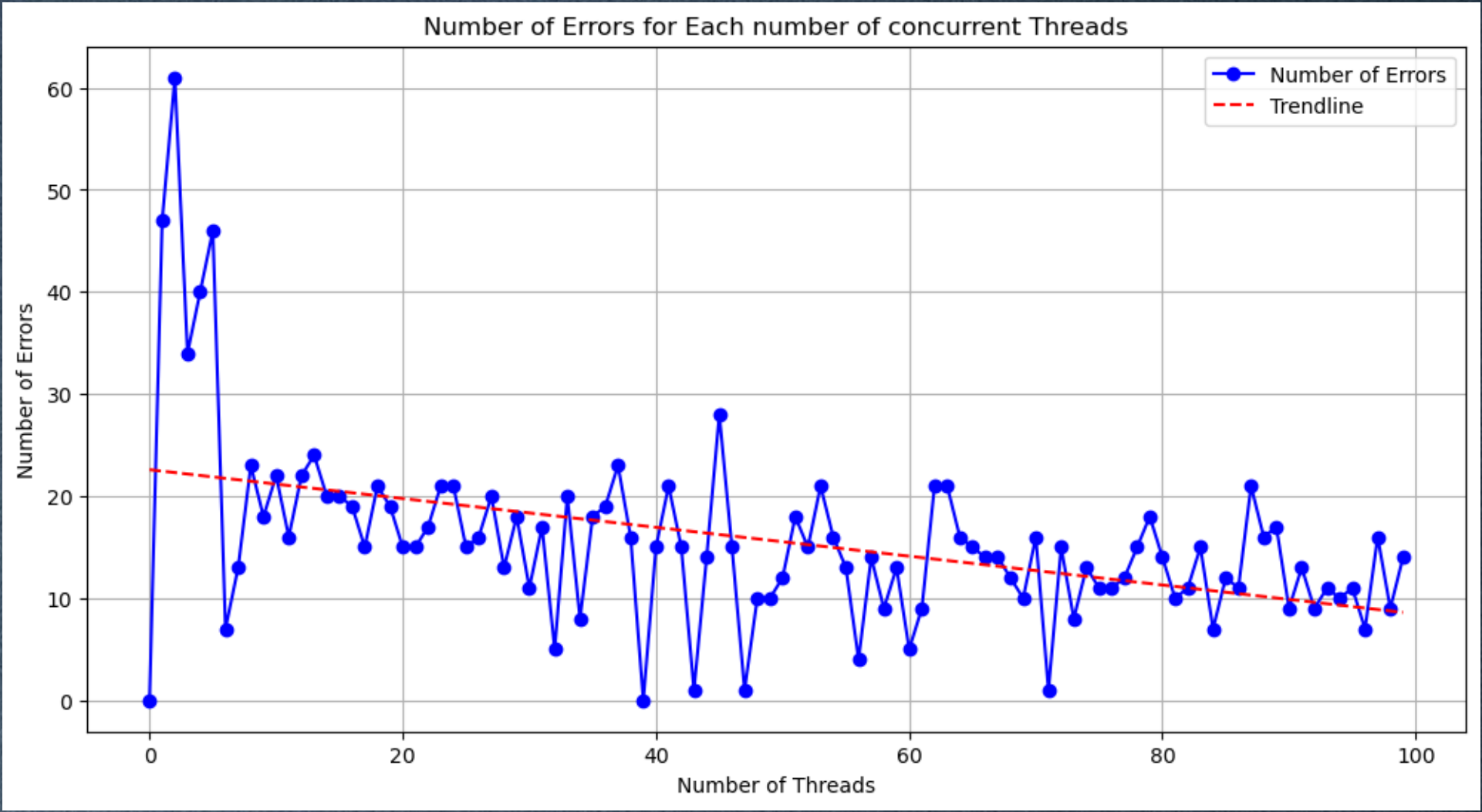
**Running Time: 1\_Thread 500k\_ArrayLength**





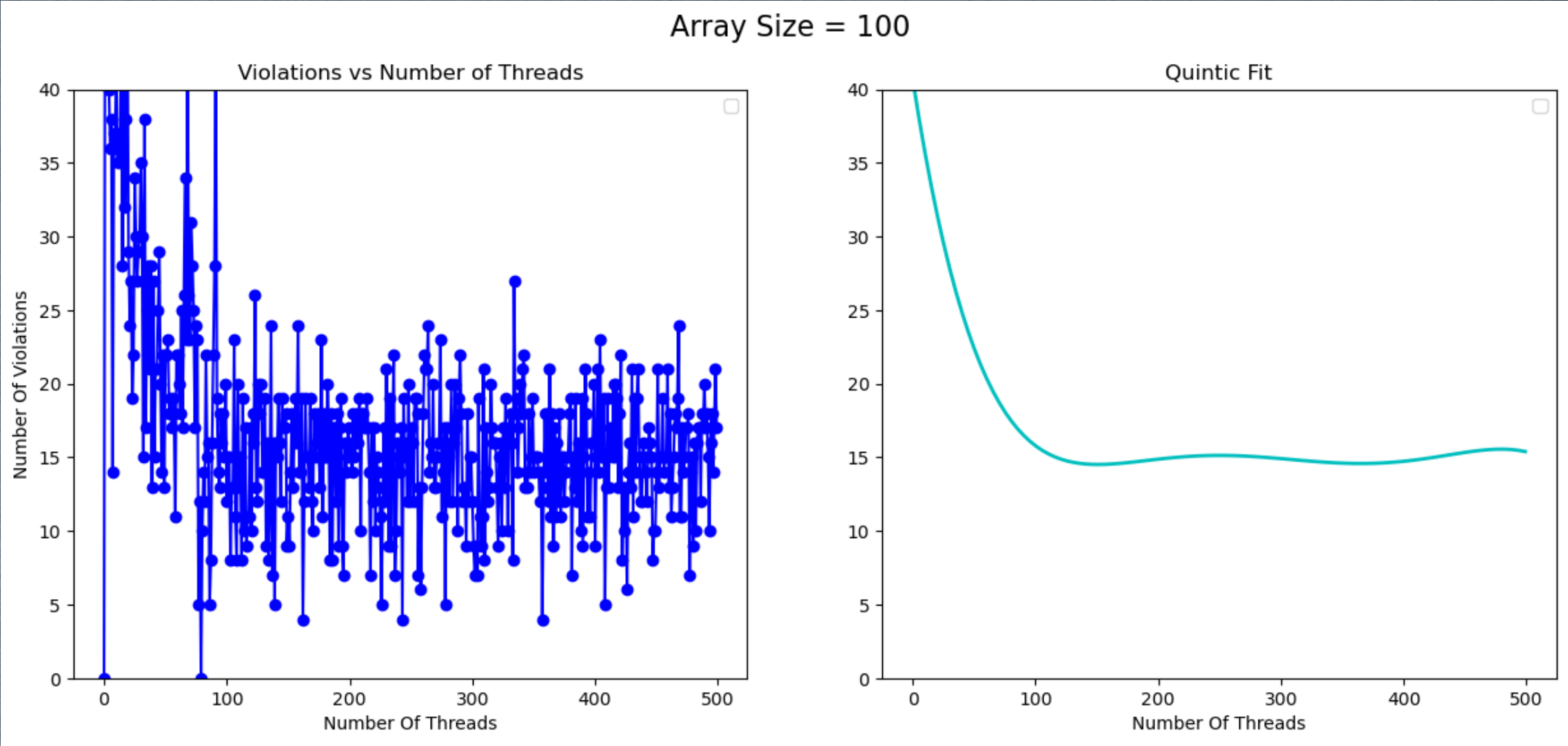
**Investigating Correlation of**  
**Number Of Violations**  
**With**  
**Number Of Threads**

# Solutions: Non-Atomic 1\_print function inside Thread

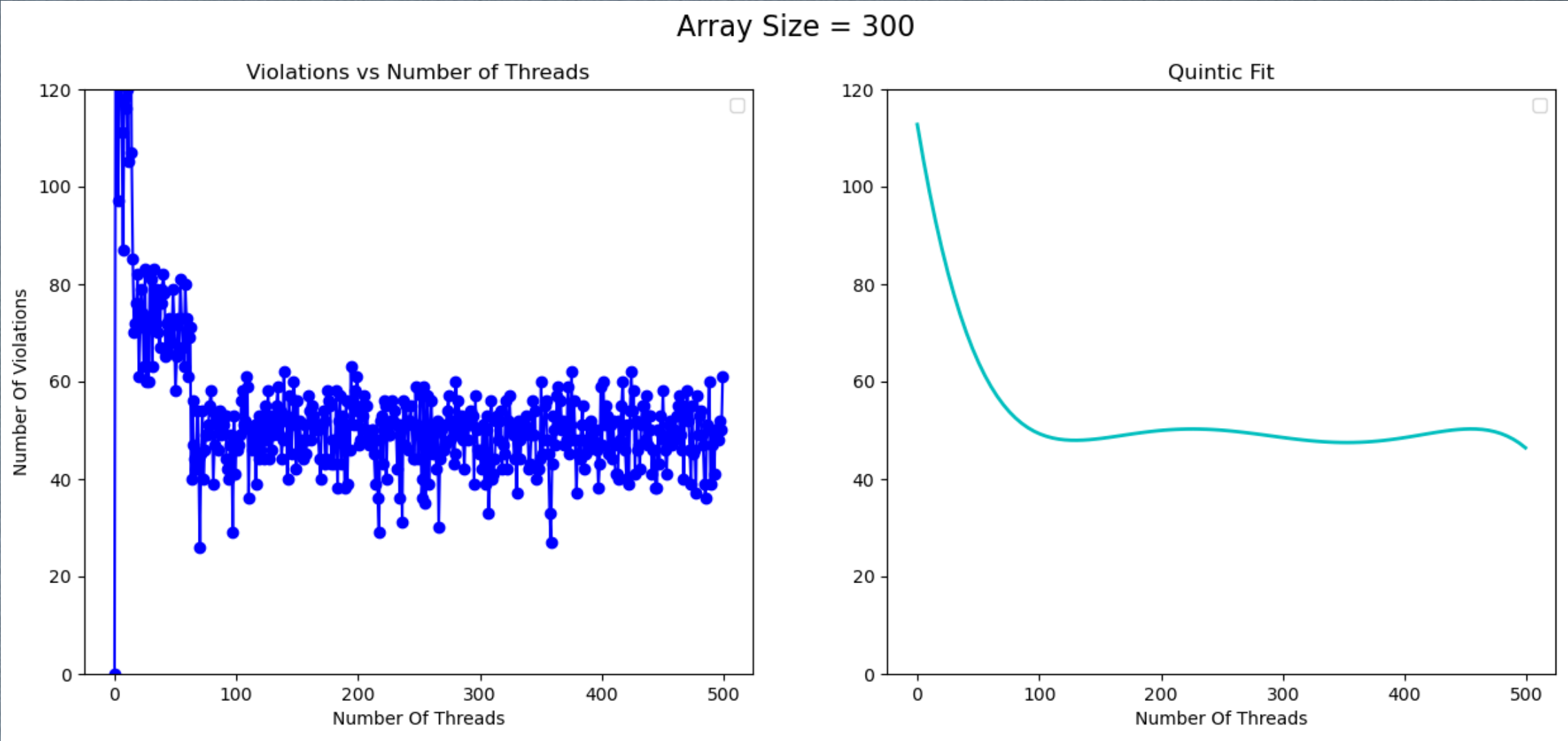




# Number Of Violations:

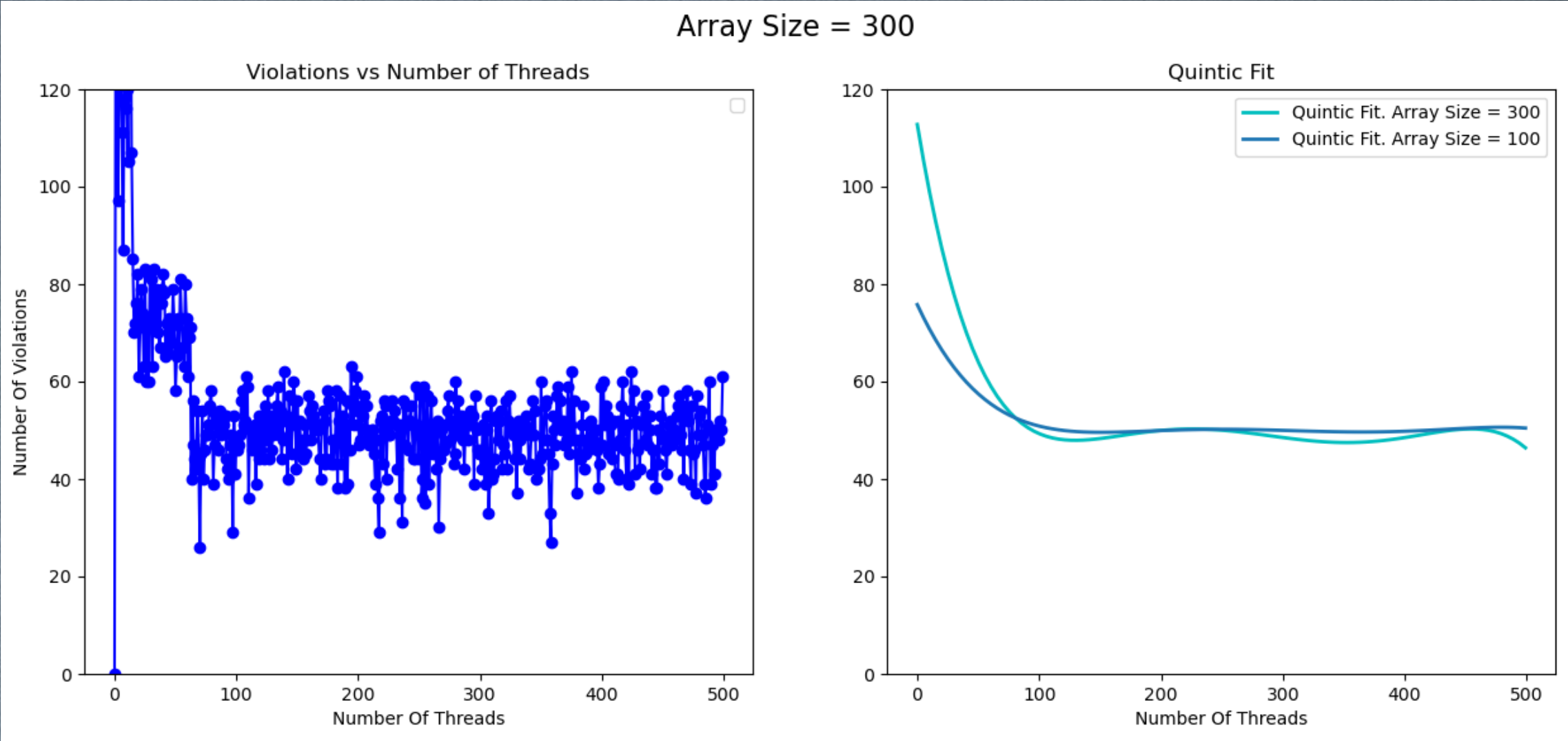


# Number Of Violations:

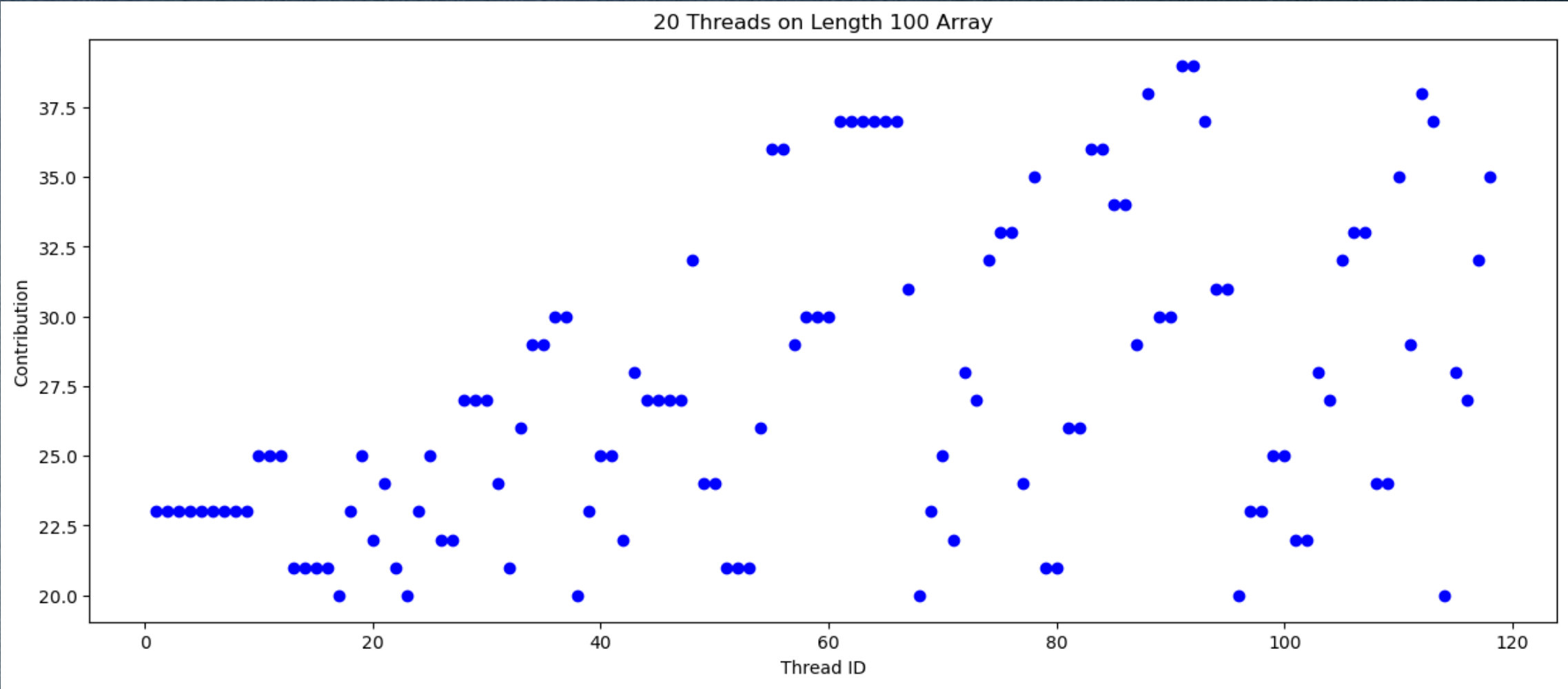




# Number Of Violations:

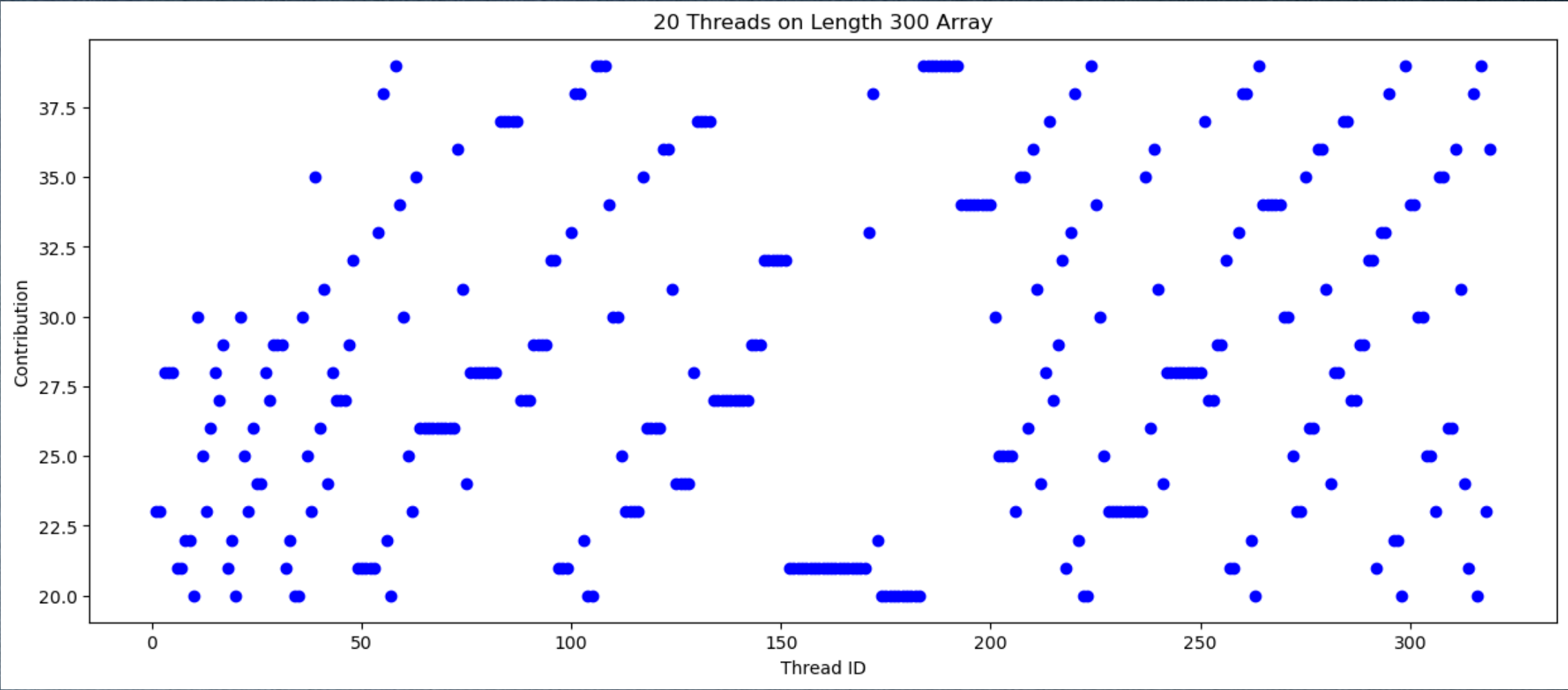


# Thread Occurrences : 20 Threads

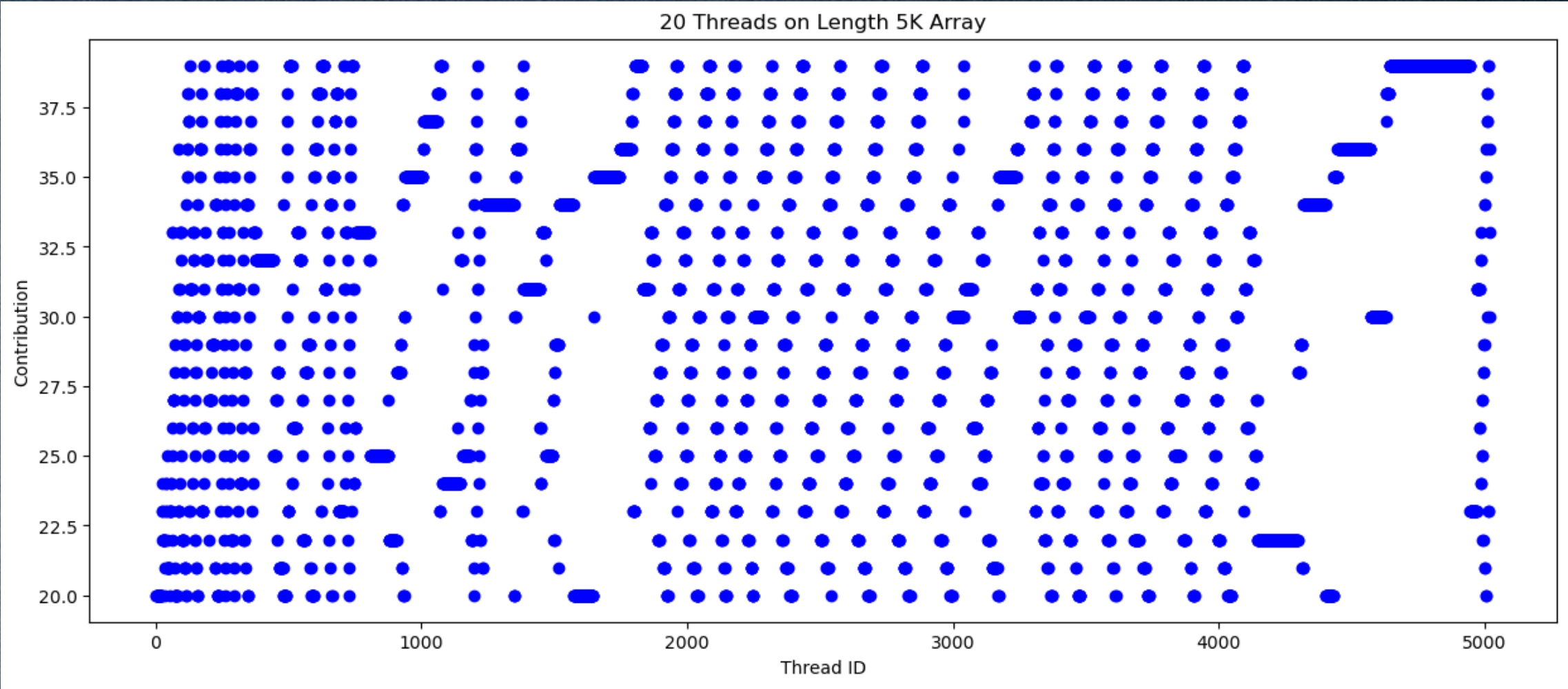




# Thread Occurrences: 20 Threads

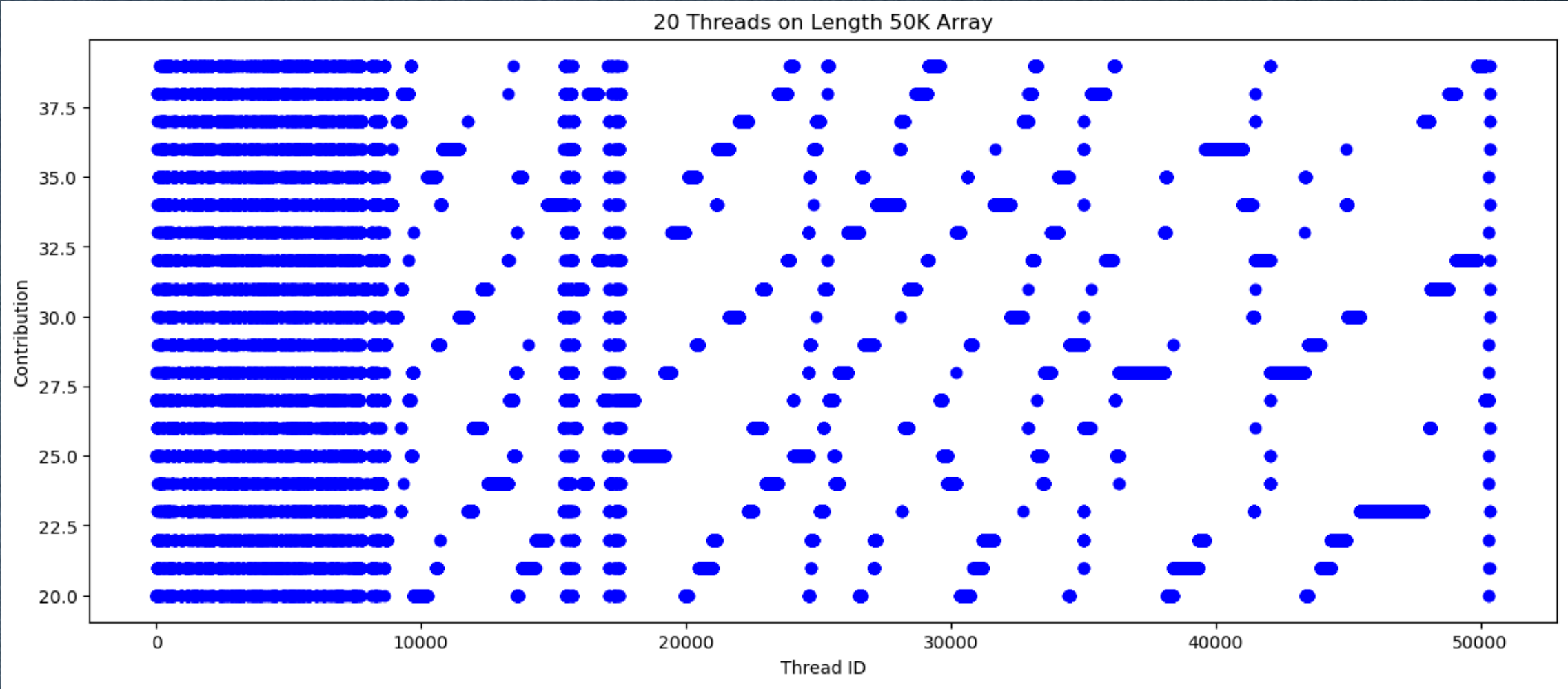


# Thread Occurrences: 20 Threads

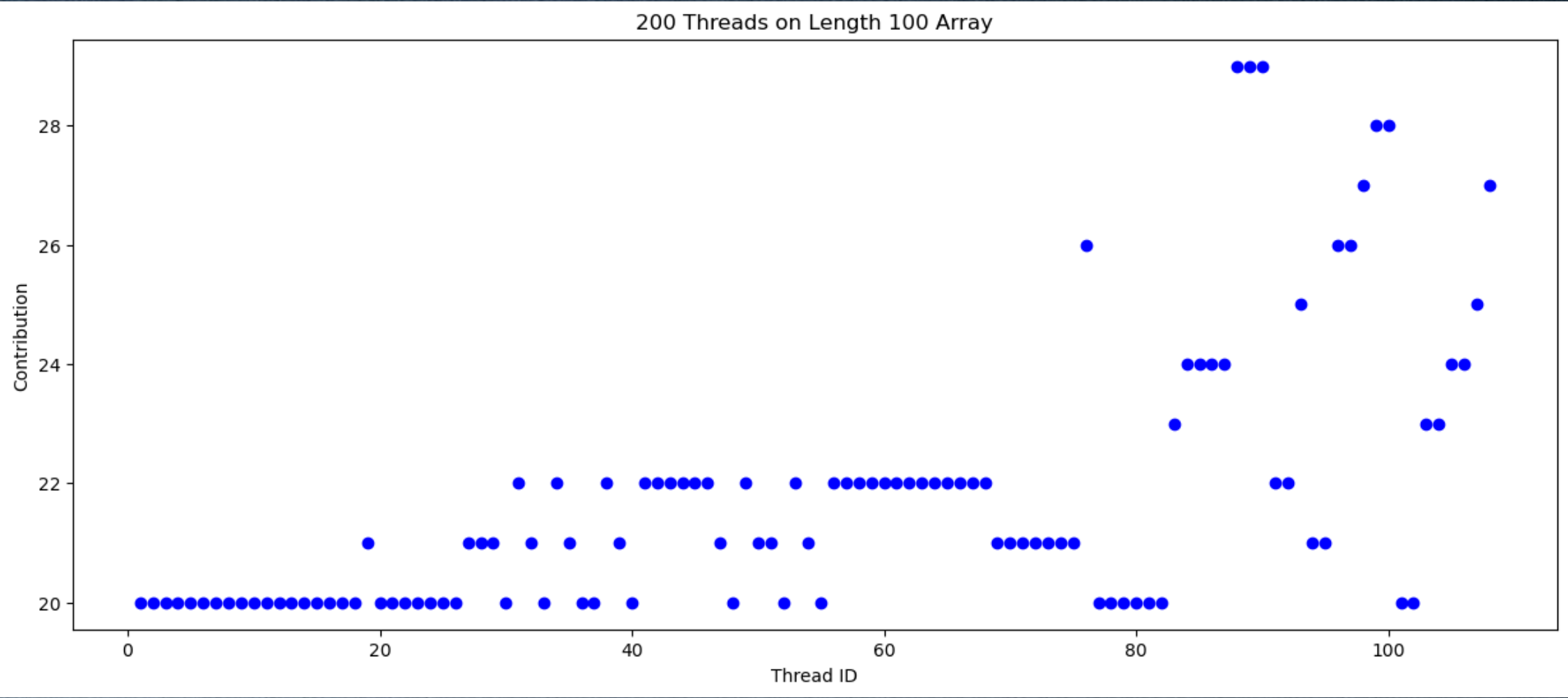




# Thread Occurrences : 20 Threads

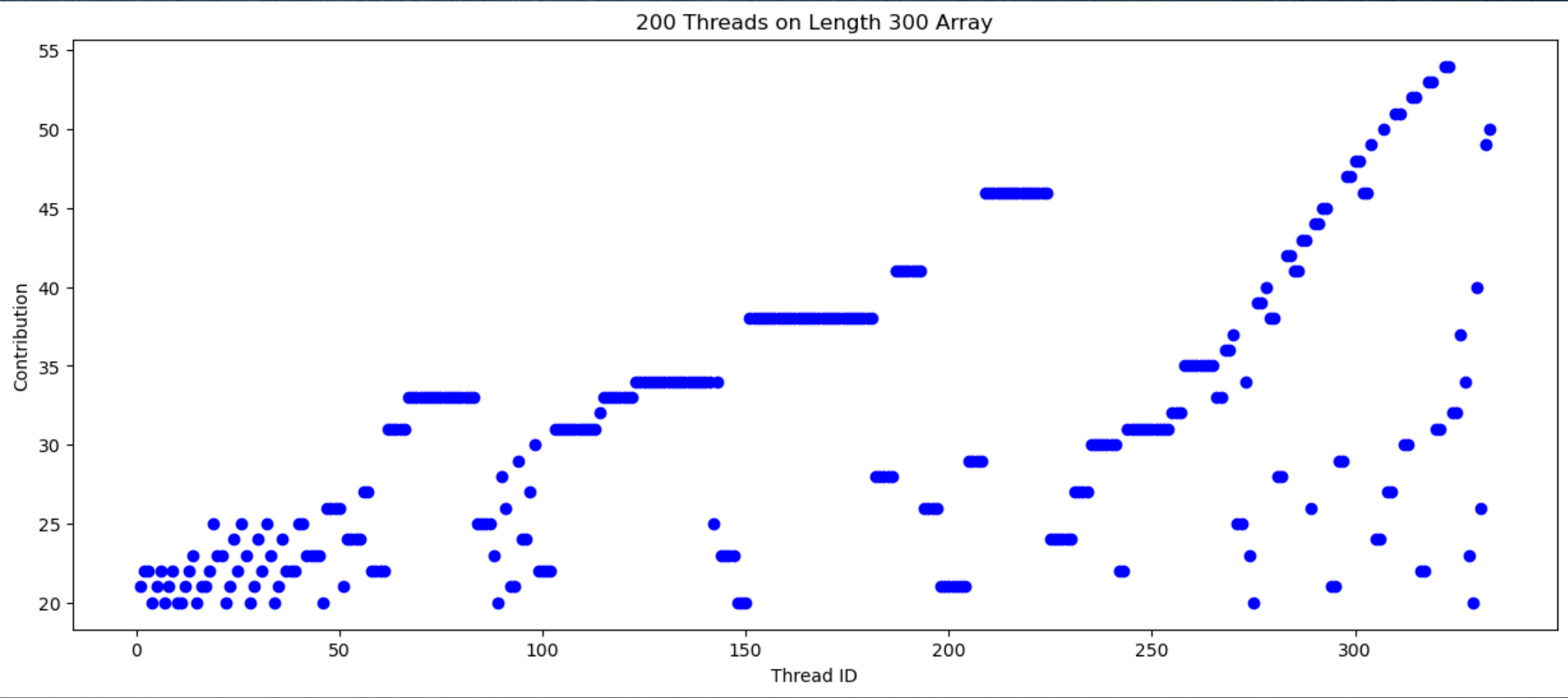


# Thread Occurrences: 200 Threads

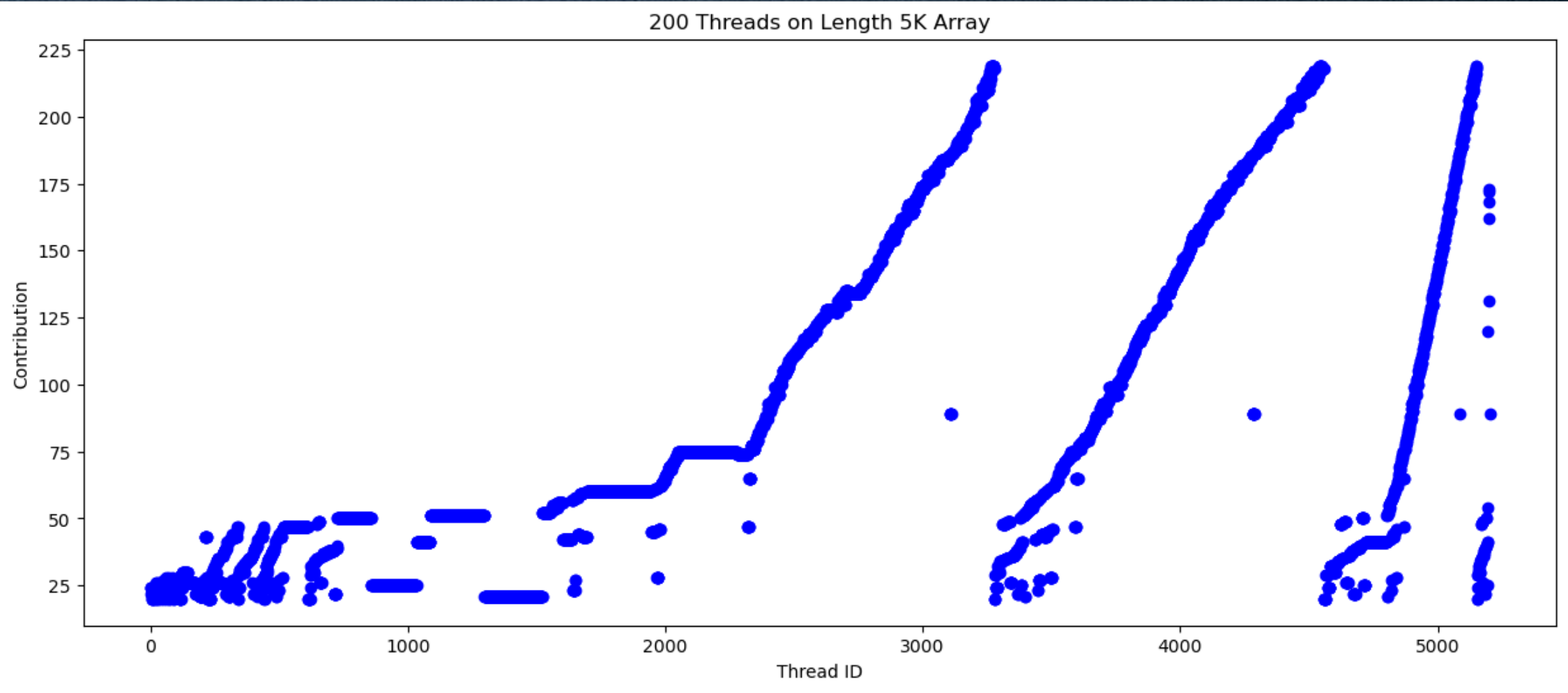




# Thread Occurrences : 200 Threads



# Thread Occurrences : 200 Threads





# Thread Occurrences : 200 Threads

