CS648 Assignment - 2

Team Name:

kQiAuZC8mLskFP47e39

Soham Samaddar 200990

Aditya Tanwar 200057

January 2023

1 Maximum Load with Simple Randomization

We showed in the course that the following simple protocol $\mathbf{P_1}$ ensures that the maximum load is concentrated around $\Theta((\log n)/(\log \log n))$.

Each client selects a server randomly uniformly among all servers and gets queued there for the execution of its job.

Fill up the following table on the basis of your experimental results of $\mathbf{P_1}$. This has to be followed by a concise inference.

$n \rightarrow$	10^3	10^4	10^5	10^6
Average Value of maximum load	5.51	6.65	7.74	8.80
$(\log_e n)/(\log_e \log_e n)$	3.57	4.15	4.71	5.26
No. of cases where maximum load exceeds average by 20%	1048	1356	202	224
No. of cases where maximum load exceeds average by 30%	75	244	22	37
No. of cases where maximum load exceeds average by 50%	19	10	0	0
No. of cases where maximum load exceeds average by 100%	0	0	0	0

Table 1: Statistics of Simple Randomization

Our inferences follow:

- The ratio of average maximum load and $(\log_e n)/(\log_e \log_e n)$ is $(1.54, 1.60, 1.64, 1.67)$ respectively (for $n = 10^3, 10^4, 10^5, 10^6$). Since the ratio is almost a constant, we conclude that the expected value of maximum load grows linearly with $(\log_e n)/(\log_e \log_e n)$. Or, in other words, the expected value of maximum load is $\mathcal{O}((\ln n)/(\ln \ln n))$ asymptotically.
- The claim above is supported further, since the values of maximum load (typically) condense towards the average value as n grows.
- There are some "exceptions" to the above claim, for example, the deviation count is more for $n = 10^4$ than for $n = 10^3$ (and similarly, more for 10^6 than 10^5). However, these statistics are "exceptions" only at a surface level. These anomalies are investigated further in the *Appendix*.

2 Maximum Load with two random choices

We discussed another simple protocol $\mathbf{P_2}$ for this problem. Here, the clients arrive in a sequential order. The i^{th} client in the sequence executes the following protocol:

1. *It selects 2 servers randomly uniformly and independently.*

2. *Sends requests to each of the selected servers about their load. These 2 servers communicate back their exact loads to the i^{th} client.*

3. *The i^{th} client selects the server which has less load than the other and gets queued there for its job.*

Fill up the following table on the basis of your experimental results of $\mathbf{P_2}$. This has to be followed by a concise inference.

$n \rightarrow$	10^3	10^4	10^5	10^6
Average Value of maximum load	3.01	3.06	3.46	4.00
$\log_e \log_e n$	1.93	2.22	2.44	2.63
No. of cases where maximum load exceeds average by 20%	64	468	0	0
No. of cases where maximum load exceeds average by 30%	64	468	0	0
No. of cases where maximum load exceeds average by 50%	0	0	0	0
No. of cases where maximum load exceeds average by 100%	0	0	0	0

Table 2: Statistics of Randomization with Two Choices

Our inferences follow:

- The ratio of average maximum load and $\log_e \log_e n$ is $(1.56, 1.38, 1.42, 1.52)$ respectively (for $n = 10^3, 10^4, 10^5, 10^6$). Since the ratio is almost a constant, we conclude that the expected value of maximum load grows linearly with $\log_e \log_e n$. Or, in other words, the expected value of maximum load is $\mathcal{O}(\ln \ln n)$ asymptotically.

- The claim above is supported further, since the values of maximum load (typically) condense towards the average value as n grows.

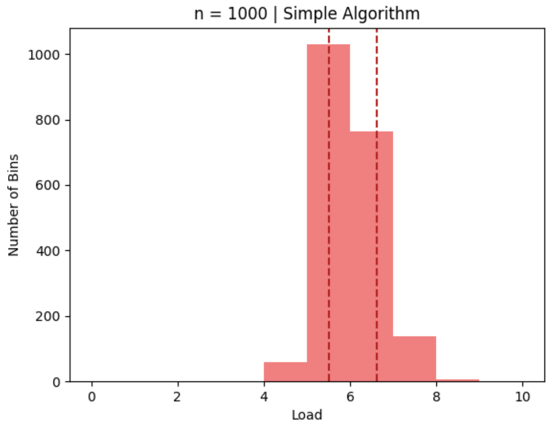
- There are some "exceptions" to the above claim, for example, the deviation count is more for $n = 10^4$ than for $n = 10^3$. However, these statistics are "exceptions" only on a surficial level. These anomalies are investigated further in the *Appendix*.

- The count of cases exceeding the average by 20% is the same as those exceeding the average by 30% for $n = 10^3$ (and for $n = 10^4$). We have tried to provide an explanation for the same in the *Appendix*.

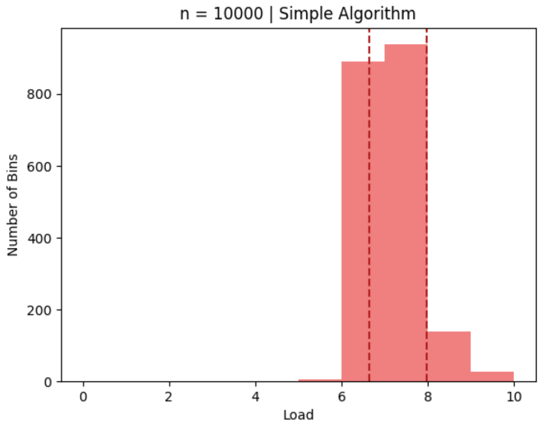
Appendix

Sudden bumps in deviation count

Let us inspect the histogram plots for $n = 1,000$ and $n = 10,000$ for the simple randomized algorithm.



(a)



(b)

For $n = 10^3 =$ 1,000, twenty percent above the mean is 6.61 while for $n = 10^4 =$ 10,000, twenty percent above the mean in 7.98. But the values attained by the load can only be integral values. Hence for $n =$ 1,000, its load is considered above twenty percent only if its load is at least 7. Loosely speaking, the interval $(6.61, 7)$ is "wasted". For $n =$ 10,000 however, only the interval $(7.98, 8)$ is "wasted" which is considerably smaller than the previous case. This is the reason we see jumps in the deviation count. It stems from the integrality of the load, and the small range of values it takes.

Same count for 20% and 30%

We take the example of $n = 10^4$; the argument is the same for $n = 10^3$ excepting the calculations.

We first calculate $3.06 \times 1.2 = 3.67$, which is 120% of the average value. Similarly, we calculate $3.06 \times 1.3 = 3.98$, which is 130% of the average value. We now observe that both of these values are strictly less than 4.

Since, the maximum load is always an integer (and thus, discrete), any instance in which the maximum load exceeds the average by 20%, must be at least 4 (by the discrete nature of the problem), and thus the same instance would exceed the average by 30% as well. Thus, the two counts are exactly the same for $n = 10^4$.

In contrast, for $n = 10^4$ in the first problem, 120% of the average is $6.65 \times 1.2 = 7.98$, while 130% of the average is $6.65 \times 1.3 = 8.64$. This allows instances to deviate by 20% of the average, while still not deviating by more than 30%. Thus, the deviation counts in the first problem differ.