Randomized (Load) Balancing and Balls and Bins Problem

In general, load balancing is a problem to distribute tasks among multiple resources. We can model this problem as tossing balls into bins. There are two policies: random dropping and random load balancing.

1. The list of all the assumptions:

we assume that number of balls and bins are equal, n. the tossing process of balls are uniformly at random and independent from each other.

For the first policy, the probability of any ball falling into any bin is $\frac{1}{n}$. The expected number of collisions (average number of balls in one bin) is $\frac{1}{n}\binom{n}{2}$. The probability of a particular bin being empty is $\left(1-\frac{1}{n}\right)^n$ in which the probability of a ball not fall into a particular bin is $1-\frac{1}{n}$. the maximum occupancy is $\frac{\log n}{\log log\ n}(1+O(1))$.

In terms of random load balancing, every time we pick d bins independently and uniformly at random, and put a ball into the bin with least balls. We then get an expected max load of $\frac{\log\log n}{\log d}$.

2.The list of all the input parameters:

- number of bins and balls: n
- number of bins to pick for random load balancing: d

3.The list of all the output metrics:

- Minimum number of balls in one bin
- Maximum number of balls in one bin
- Average number of balls in bins

4.The main data structures:

- List of bins with number of occupied balls
- List of balls
- Dictionary to store results for any specific number of bins/balls

5.The main algorithms:

- Random dropping:

  For each ball,

  1. choose uniformly at random one bin

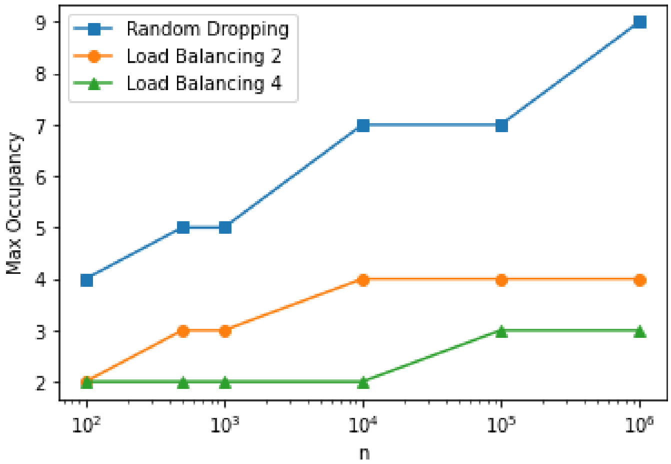
2. Put the ball into it

- Random load balancing:

  For each ball,

  1. Choose uniformly at random d bins
  2. Put the ball in the least occupied one

It should be noted that based on theorical results, random dropping has the highest maximum occupancy compared to the other ones. Regarding random load balancing, the number of maximum occupancies decreases as d increases.



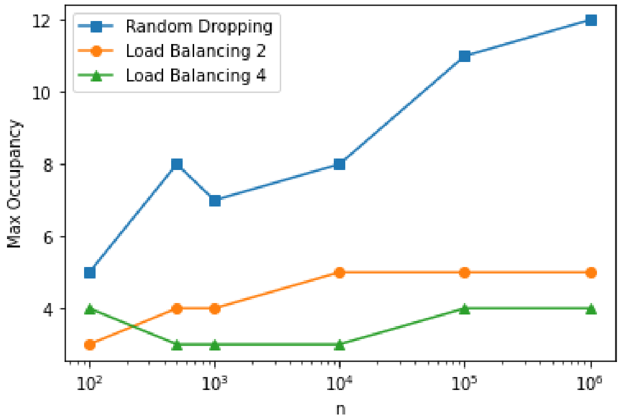
In terms of comparison between theorical and empirical results, we compare maximum occupancy of random dropping policy with its upper bound, and maximum occupancy of random load balancing with its sharply concentrated values. As it can be seen in tables below, they are reasonably satisfied.

	n	max_drop	theory_max_upper_drop
0	100	4	9.046421
1	500	5	10.205155
2	1000	5	10.722750
3	10000	7	12.444574
4	100000	7	14.135132
5	1000000	9	15.784393

	n	max_l2	theory_max_load2
0	100	2	2.203254
1	500	3	2.635663
2	1000	3	2.788217
3	10000	4	3.203254
4	100000	4	3.525183
5	1000000	4	3.788217

	n	max_l4	theory_max_load4
0	100	2	1.101627
1	500	2	1.317832
2	1000	2	1.394108
3	10000	2	1.601627
4	100000	3	1.762591
5	1000000	3	1.894108

Regarding extension, there are other load balancing techniques like hashing in which a data structure is maintained for a set of elements for fast look-up. However, we double the number of balls in order to check whether the number of maximum occupancies is doubled or not. Results show that there is not a linear relationship between these two parameters.



	n	max_drop	theory_max_upper_drop		n	max_l2	theory_max_load2		n	max_l4	theory_max_load4
0	100	5	9.046421	**0**	100	3	2.203254	**0**	100	4	1.101627
1	500	8	10.205155	**1**	500	4	2.635663	**1**	500	3	1.317832
2	1000	7	10.722750	**2**	1000	4	2.788217	**2**	1000	3	1.394108
3	10000	8	12.444574	**3**	10000	5	3.203254	**3**	10000	3	1.601627
4	100000	11	14.135132	**4**	100000	5	3.525183	**4**	100000	4	1.762591
5	1000000	12	15.784393	**5**	1000000	5	3.788217	**5**	1000000	4	1.894108