Scheduling: Proportional Share

Annalise Tarhan

February 1, 2021

1 Compute the solutions for simulations with 3 jobs and random seeds of 1, 2, and 3.

1.1 ./lottery.py -j 3 -s 1

Job 0 (length = 1, tickets = 84)
Job 1 (length = 7, tickets = 25)
Job 2 (length = 4, tickets = 44)
Random numbers: 651593, 788724, 93859, 28347, 835765, 432767

Total number of tickets: 153 (0-83, 84-108, 109-152)
1. 651593 % 153 = 119 Job 2 - 3 left
2. 788724 % 153 = 9 Job 0 - DONE
Tickets remaining: 69 (x, 0-24, 25-68)
3. 93859 % 69 = 19 Job 1 - 6 left
4. 28347 % 69 = 57 Job 2 - 2 left
5. 835765 % 69 = 37 Job 2 - 1 left
6. 432767 % 69 = 68 Job 2 - DONE
Tickets remaining: 25 (x, 0-24, x)
7-12. Job 1 runs until it is finished, six more times.

1.2 ./lottery.py -j 3 -s 2

Job 0 (length = 9, tickets = 94)
Job 1 (length = 8, tickets = 73)
Job 2 (length = 6, tickets = 30)
Random numbers: 605944, 606802, 581204, 158383, 430670, 393532, 723012, 994820, 949396, 544177, 444854, 268241, 35924, 27444, 464894, 318465, 380015, 891790, 525753, 560510, 236123, 23858

Total number of tickets: 197 (0-93, 94-166, 167-196)
1. 605944 % 197 = 169 Job 2 - 5 left
2. 606802 % 197 = 42 Job 0 - 8 left
3. 581204 % 197 = 54 Job 0 - 7 left

4. 158383 % 197 = 192 Job 2 - 4 left
5. 430670 % 197 = 28 Job 0 - 6 left
5. 393532 % 197 = 123 Job 1 - 7 left
6. 723012 % 197 = 22 Job 0 - 5 left
7. 994820 % 197 = 167 Job 2 - 3 left
8. 949396 % 197 = 53 Job 0 - 4 left
9. 544177 % 197 = 63 Job 0 - 3 left
10. 444854 % 197 = 28 Job 0 - 2 left
11. 268241 % 197 = 124 Job 1 - 6 left
12. 35924 % 197 = 70 Job 0 - 1 left
13. 27444 % 197 = 61 Job 0 - DONE
Tickets remaining: 103 (x, 0-72, 73-102)
14. 464894 % 103 = 55 Job 1 - 5 left
15. 318465 % 103 = 92 Job 2 - 2 left
16. 380015 % 103 = 48 Job 1 - 4 left
17. 891790 % 103 = 16 Job 1 - 3 left
18. 525753 % 103 = 41 Job 1 - 2 left
19. 560510 % 103 = 87 Job 2 - 1 left
20. 236123 % 103 = 47 Job 1 - 1 left
21. 23858 % 103 = 65 Job 1 - DONE
Tickets remaining: 30 (x, x, 0-30)
22. Job 2 runs one more time, then done.

1.3 ./lottery.py -j 3 -s 3

Job 0 (length = 2, tickets = 54)
Job 1 (length = 3, tickets = 60)
Job 2 (length = 6, tickets = 6)
Randoms: 13168, 837469, 259354, 234331, 995645 Random 470263 Random 836462 Random 476353 Random 639068 Random 150616 Random 634861

Total number of tickets: 120 (0-53, 54-113, 114-119)
1. 13168 % 120 = 88 Job 1 - 2 left
2. 837469 % 120 = 109 Job 1 - 1 left
3. 259354 % 120 = 34 Job 0 - 1 left
4. 234331 % 120 = 91 Job 1 - DONE
Tickets remaining: 60 (0-53, x, 54-59)
5. 995645 % 60 = 5 Job 0 - DONE
Tickets remaining: 6 (x, x, 0-5)
6-11. Job 2 runs until it is finished, six more times.

2 Now run with two specific jobs: each of length 10, but one (job 0) with just 1 ticket and the other (job 1) with 100. What happens when the number of tickets is so imbalanced? Will job 0 ever run before job 1 completes? How often? In general, what does such a ticket imbalance do to the behavior of lottery scheduling?

Because the lottery is random, any order is possible, but in the vast majority of cases job 1 will finish long before job 0 does. This large of an imbalance effectively orders them, allowing one to finish before the other starts.

3 When running with two jobs of length 100 and equal ticket allocations of 100, how unfair is the scheduler? Run with some different random seeds to determine the (probabilistic) answer; let unfairness be determined by how much earlier one job finishes than the other.

Seed: 100 Difference: 15 Unfairness: .925
Seed: 150 Difference: 10 Unfairness: .95
Seed: 200 Difference: 3 Unfairness: .985
Seed: 250 Difference: 8 Unfairness: .96
Seed: 300 Difference: 2 Unfairness: .99

In this small sample, the unfairness was very close to 1, which makes the scheduler quite fair.

4 How does your answer to the previous question change as the quantum size gets larger?

Seed: 100 Difference: $15 \rightarrow 42 \rightarrow 39 \rightarrow 24 \rightarrow 20$
Seed: 150 Difference: $10 \rightarrow 12 \rightarrow 38 \rightarrow 40 \rightarrow 35$
Seed: 200 Difference: $3 \rightarrow 2 \rightarrow 6 \rightarrow 28 \rightarrow 25$

As the quantum size increases, unfairness increases as well, but the correlation isn't perfect.

5 Can you make a version of the graph that is found in the chapter? What else would be worth exploring? How would the graph look with a stride scheduler?

The graph in this chapter shows the relationship between job length and unfairness. Presumably this question is asking what the relationship between quantum size and unfairness would look like in graph form, and the answer is that the slope would be decreasing, but not monotonically. Other relationships to explore would be the relationships between unfairness and the length or number of jobs. With a stride scheduler, all of these graphs would look a lot flatter and closer to 1, since the stride scheduler is very fair.