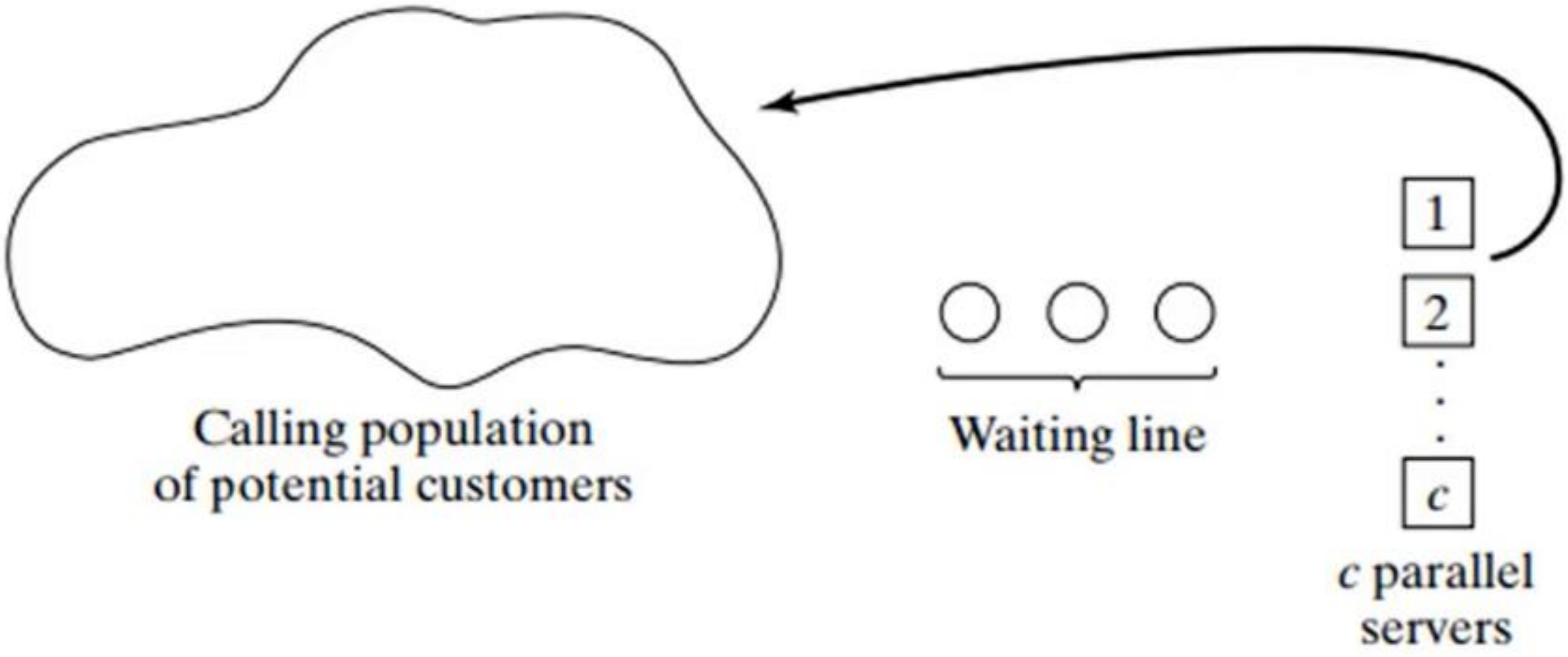
Multi-server Queuing system

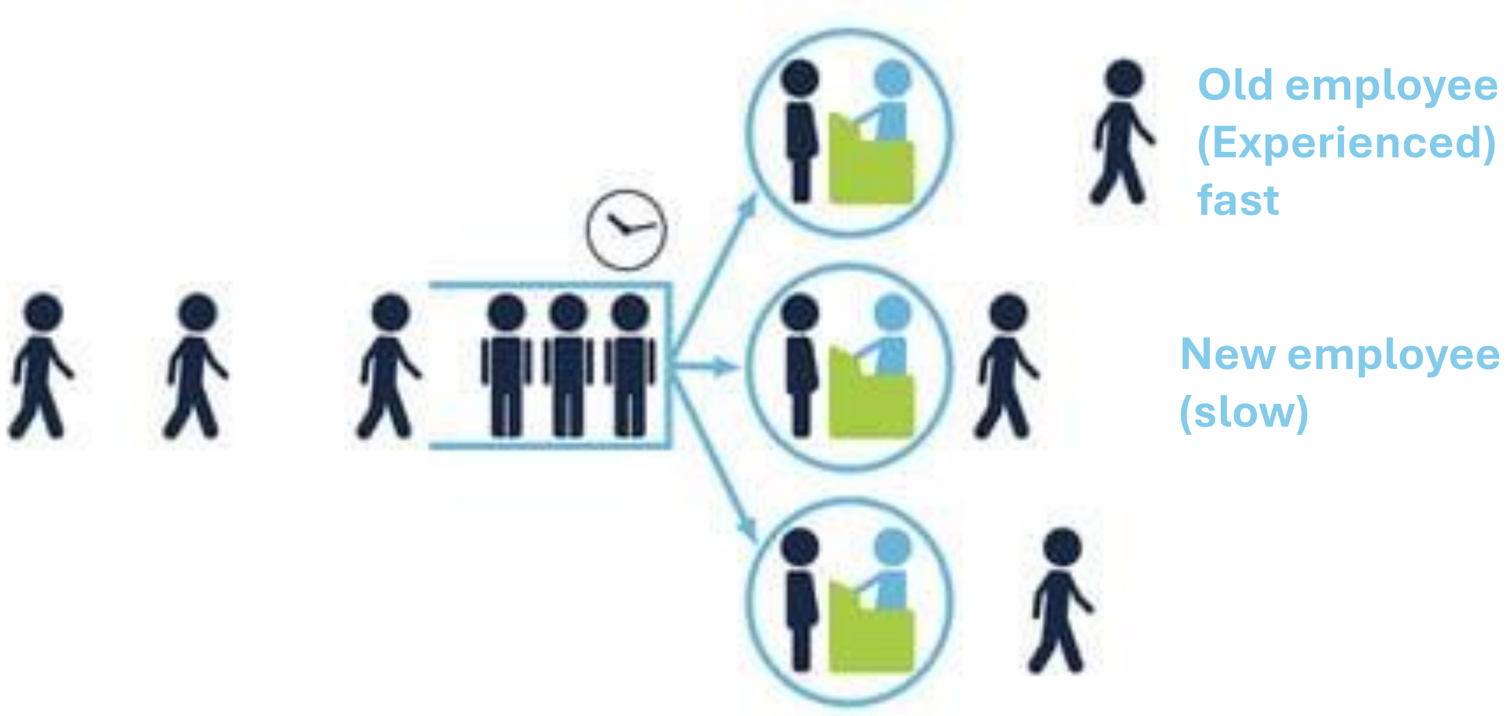
Model

- In a multi-server queuing system, we have c parallel servers and the customers enter any available server



Example

- Banks usually have multi-server queuing system.
  - Customers enter the system, go to the available server, or wait if all tellers are busy
  - Different servers might have different service-time distribution



Cont.

- Let us assume, for simplicity, that the system has two servers as follows
  - Able: More experienced and provide a service faster
  - Baker: New employee and slower than Able

- We constraint our system with a rule
  - **The customer goes to Able if both are idle**
  - we can make this as a random distribution of selecting the server (with different or same probability)
  - You can assume that the customer go to the nearest server
  - etc.

Cont.

- Let us start with formulating the interarrival-time of customers

Time between Arrivals (Minutes)	*Probability*	*Cumulative Probability*	*Random-Digit Assignment*
1	0.25	0.25	01–25
2	0.40	0.65	26–65
3	0.20	0.85	66–85
4	0.15	1.00	86–00

Cont.

- The service-time distribution usually differs, as different employees provide service at different speed rates
- For **Able**

Service Time (Minutes)	*Probability*	*Cumulative Probability*	*Random-Digit Assignment*
2	0.30	0.30	01–30
3	0.28	0.58	31–58
4	0.25	0.83	59–83
5	0.17	1.00	84–00

- For **Baker**

Service Time (Minutes)	*Probability*	*Cumulative Probability*	*Random-Digit Assignment*
3	0.35	0.35	01–35
4	0.25	0.60	36–60
5	0.20	0.80	61–80
6	0.20	1.00	81–00

Cont.

- Now the task is to find if the current arrangement of servers with our rules is good.
  - or do we need to modify the arrangement to make our system more reliable?
- For this we simulate for **1 hour** and calculate the performance measurements.
  - The more the simulation lasts, the better the estimation of the system's performance

Cont.

A	B	C	D	E	F	G	H	I	J	K	L
					Able			Baker			
Customer No.	Random Digits for Arrival	Time between Arrivals	Clock Time of Arrival	Random Digits for Service	Time Service Begins	Service Time	Time Service Ends	Time Service Begins	Service Time	Time Service Ends	Time in Queue
1											
2											
3											
4											
5											

Cont.

- Generate a random number for the service time

A	B	C	D	E	F	G	H	I	J	K	L
					Able			Baker			
Customer No.	Random Digits for Arrival	Time between Arrivals	Clock Time of Arrival	Random Digits for Service	Time Service Begins	Service Time	Time Service Ends	Time Service Begins	Service Time	Time Service Ends	Time in Queue
1	–	–	0	95							
2											
3											
4											
5											

Cont.

- The service time is selected from the distribution of the server
  - Able, as we have a rule saying that the customer goes to Able when both are idle

A	B	C	D	E	F	G	H	I	J	K	L
						Able			Baker		
Customer No.	Random Digits for Arrival	Time between Arrivals	Clock Time of Arrival	Random Digits for Service	Time Service Begins	Service Time	Time Service Ends	Time Service Begins	Service Time	Time Service Ends	Time in Queue
1	-	-	0	95	0	5	5				0
2											
3											
4											
5											

Cont.

A	B	C	D	E	F	G	H	I	J	K	L
						Able			Baker		
Customer No.	Random Digits for Arrival	Time between Arrivals	Clock Time of Arrival	Random Digits for Service	Time Service Begins	Service Time	Time Service Ends	Time Service Begins	Service Time	Time Service Ends	Time in Queue
1	–	–	0	95		5					
2	26	2		21							
3	98	4		51							
4	90	4		92							
5	26	2		89							

Simulation table

- We completed the simulation in 1 hour or 60 minutes
- here we finished at 62 which is closer to 60

A	B	C	D	E	F	G	H	I	J	K	L
					Able			Baker			
Customer No.	*Random Digits for Arrival*	*Time between Arrivals*	*Clock Time of Arrival*	*Random Digits for Service*	*Time Service Begins*	*Service Time*	*Time Service Ends*	*Time Service Begins*	*Service Time*	*Time Service Ends*	*Time in Queue*
1	—	—	0	95	0	5	5				0
2	26	2	2	21				2	3	5	0
3	98	4	6	51	6	3	9				0
4	90	4	10	92	10	5	15				0
5	26	2	12	89				12	6	18	0
6	42	2	14	38	15	3	18				1
7	74	3	17	13	18	2	20				1
8	80	3	20	61	20	4	24				0
9	68	3	23	50				23	4	27	0
10	22	1	24	49	24	3	27				0
11	48	2	26	39	27	3	30				1
12	34	2	28	53				28	4	32	0
13	45	2	30	88	30	5	35				0
14	24	1	31	01				32	3	35	1
15	34	2	33	81	35	4	39				2
16	63	2	35	53				35	4	39	0
17	38	2	37	81	39	4	43				2
18	80	3	40	64				40	5	45	0
19	42	2	42	01	43	2	45				1
20	56	2	44	67	45	4	49				1
21	89	4	48	01				48	3	51	0
22	18	1	49	47	49	3	52				0
23	51	2	51	75				51	5	56	0
24	71	3	54	57	54	3	57				0
25	16	1	55	87				56	6	62	1
26	92	4	59	47	59	3	62				0
						56			43		11

Findings

1. Over the 62-minute period Able was busy 90% of the time.
2. Baker was busy only 69% of the time. The seniority rule keeps Baker less busy (and gives Able more tips).
3. Nine of the 26 arrivals (about 35%) had to wait. The average waiting time for all customers was only about 0.42 minute (25 seconds), which is very small.

One can extract more findings and plot more diagrams to visualize the performance of the system

Summary

- Queuing systems have vast applications, and simulating them gives us a better understanding of the system under different configurations and conditions.
- Some metrics, such as the average waiting time, idle and busy times of the servers, and the number of customers in the queue, can be used to observe the performance of the queuing system.
- We can simulate a single server of many servers and put the simulation rules as needed
- The queuing simulation is mainly dependent upon the random distributions.