Politecnico di Milano
DEIB Dept.

Course

Performance Evaluation of Computer Systems

M. Gribaudo

2020 / 2021 Projects

Project rules

Consider the system described in the project assigned to you as described below. Define a model and use it to answer the proposed question. You can use whatever tools and techniques you wish: MatLab code, Excel evaluation, Java Modelling Tools are some examples, but you are free to use other tools if you know them and feel more comfortable using them. Your model can either be a Queuing Network, a Stochastic Petri Net, or a multiformsalism model combining the two. Your models can be open or closed, single or multiple class as you think more appropriate for the problem that has been assigned to you.

Prepare a small document describing your approach, and a set of slides that can help you in presenting your results. In your analysis, try not limiting yourself just to answer the proposed questions, but use what you have learnt to derive as much insights from the system behavior from the models you have created.

Chose the project and the corresponding set of model parameters according to the last two digits on the right (the least significant) of your “Codice Persona” as specified in the table included below. **This exercise is mandatory and must be presented at the exam!**

Which						
Last digitis of "Codice Persona"					Project	Data set
00	20	40	60	80	A	3
01	21	41	61	81	C	4
02	22	42	62	82	B	2
03	23	43	63	83	B	3
04	24	44	64	84	B	1
05	25	45	65	85	A	1
06	26	46	66	86	E	1
07	27	47	67	87	D	2
08	28	48	68	88	A	2
09	29	49	69	89	D	4
10	30	50	70	90	C	3
11	31	51	71	91	E	3
12	32	52	72	92	D	1
13	33	53	73	93	C	2
14	34	54	74	94	B	4
15	35	55	75	95	A	4
16	36	56	76	96	D	3
17	37	57	77	97	E	4
18	38	58	78	98	C	1
19	39	59	79	99	E	2

Project **A**

A distributed architectural rendering algorithm creates images that shows a view of a 3D reconstruction of a building. It works in the following way. There are N processes, each one renders a view of the considered project. After a start-up time of exponentially distributed duration Z, each process first reads the data for the rendering from the storage. Then it splits each image into M smaller rectangular portions that are rendered separately on different threads. Rendering is performed by K identical servers. When all the M portions have been completed, the image is saved back on the storage. Both the servers and the storage can be considered processor sharing systems with exponential service time, and the start-up time can be modeled with a delay station. The average time required to read the data for a rendering is D_L, the time for rendering one of the M portions on one of the K servers is D_R, and the time required to save the final image on the storage is D_S.

The administrators wants to decide the minimum number of servers $K_{\min}$ and the best number of processes N to have an average rendering time less than $R_{\max}$.

Parameter sets

ID	Z	M	D_L	D_R	D_S	$R_{\max}$
1	1.5 s.	64	800 ms.	500 ms.	1200 ms.	10 s.
2	1 s.	128	500 ms.	250 ms.	800 ms.	10 s.
3	1.2 s.	256	250 ms.	130 ms.	600 ms.	15 s.
4	2 s.	192	1200 ms.	200 ms.	1500 ms.	20 s.

Project **B**

A department store is starting a new online order and delivery service. A server accepts requests from online users which arrive according to Poisson process of rate λ_{online}, that are served with and Erlang distributed time, with average $D_{serv,online}$ and a coefficent of variation $c_{serv,online} = 0.25$. The same server is used by N_{op} operators whose job is: i) check the goods that have been ordered, ii) collect them in the department store, iii) check them out at a counter, iv) bring to the delivery station. Each of the previous activities require an exponentially distributed amount of time, characterised respectively by the following rates: $\mu_{serv,op}$, $\mu_{dep,op}$, μ_{count} and $\mu_{delivery,op}$. The department store is also visited by regular costumers that arrive according to a Markov-modulated Poisson process, that alternates between a low affluence and high affluence at rates $\gamma_{LH} = 1$h. and $\gamma_{HL} = 1$h., each one characterised by arrival rates $\lambda_{cost,L} = 10$ cost./h. and $\lambda_{cost,H} = 30$ cost./h. Each costumer spends an exponential distributed time in the department store, for which its average a. They are served at the counters at the same rate μ_{count} as the operators that works on on-line orders. There are a total of $N_{counter}$ counters, served by a single queue where both costumers and operators joins.

The director of the department store wants to understand the total number of employees (N_{op}+$N_{counter}$) to run the service, with a reasonable delivery time for on-line costumers and for regular walk-in costumers.

Parameter sets

ID	λ_{online}	$D_{serv,online}$	$\mu_{serv,op}$	$\mu_{dep,op}$	μ_{count}	$\mu_{delivery,op}$	a
1	50 req./h.	0.5 min.	60 job/h.	30 job/h.	60 job/h.	100 job/h.	10 min.
2	60 req./h.	0.4 min.	60 job/h.	20 job/h.	50 job/h.	120 job/h.	15 min.
3	30 req./h.	0.75 min.	40 job/h.	30 job/h.	75 job/h.	90 job/h.	20 min.
4	100 req./h.	0.25 min.	30 job/h.	20 job/h.	90 job/h.	120 job/h.	30 min.

Project **C**

A newsletter dispatching systems, receives two kind of streams:

- A quick news information stream (on the average, λ_{news} news per hour)
- A detailed article stream (on the average λ_{artc} articles per hours)

Each stream is processed by a text and image processing systems, where quick news requires an average of D_{news}, while articles need an average of D_{artc}. Feeds are then sent to three social media, namely Facebook, Twitter and Instagram, each one using a sending queue that works in FCFS. Each feed might be sent to one or more of the considered media. In particular, each social network receives the following percentage of feeds, and requires a different average time for being processed:

Social media	news %	news time	article %	article time
Facebook	50%	8 s.	80%	10 s.
Twitter	80%	6 s.	10%	12 s.
Instagram	75%	8 s.	60%	9 s.

All service time distributions can be considered exponential, and arrivals are Poisson processes. Note that social media can be modeled by infinite server stations, since those services are generally spread over an extremely large number of servers (which is unknown to the users). The administrator would like to study the average time required to publish an article over all its media.

Parameter sets

ID	λ_{news}	λ_{artc}	D_{news}	D_{artc}
1	1 job/s.	0.1 job/s.	500 ms.	2 s.
2	1 job/s.	0.2 job/s.	300 ms.	1.5 s.
3	2 job/s.	0.1 job/s.	250 ms.	3 s.
4	1.5 job/s.	0.15 job/s.	200 ms.	3 s.

Project **D**

A wireless communication protocol uses the CSMA / CA (Carrier Sense Multiple Access / Collision Avoidance) technique to avoid conflict on the medium. We consider N users, each one transmitting a packet to a server exactly every Z seconds. When a transmission starts, it waits a random delay that can be modelled with an exponential distribution of parameter μ_W. After the delay, it hears the channel. If the channel is empty (with probability p), it starts transmission, otherwise it repeats the CA procedure, waiting another random delay until the channels is free (with probability $1-p$). Transmission occurs at rate μ_T according to a Poisson process. The transmission queue has a maximum capacity of K packets: if new packets arrive when the buffer is full, they are discarded. If the transmission succeed, the servers sends an ACK back to the server after a deterministic time Δ_A. ACKs uses the same wireless channel as normal packets, and are send according to a FCFS queue; their transfer rate is $\mu_A = \mu_T$.

We want to study the average number of access to the channel, the drop rate, and the end-to-end delay.

Parameter sets

ID	N	Z	μ_W	p	μ_T	K	Δ_A
1	50	10 s.	10 job/s.	0.9	30 job/s.	30	50 ms.
2	30	10 s.	5 job/s.	0.95	30 job/s.	10	50 ms.
3	20	5 s.	5 job/s.	0.98	30 job/s.	10	50 ms.
4	80	20 s.	30 job/s.	0.95	30 job/s.	50	50 ms.

Project **E**

A three tier web application is composed by a web server, an application server and a DB: services are called in sequence. Each server is replicated respectively in C_{WS}, C_{AS} and C_{DB} instances, each one running on a different machine that shares a common queue of finite capacity K_{WS}, K_{AS} and K_{DB}. Three types of requests populate the system: N_U user requests (characterised by a think time Z_U), N_S remote procedure call software agents, and N_B batch elaboration programs. Each type of request requires a different amount of time from each resource, according to an exponential distribution following the average shown in the table below. Requests arriving at a full station are blocked after they are served.

	U	S	B
WS	80 ms.	15 ms.	5 ms.
AS	30 ms.	25 ms.	120 ms.
DB	20 ms.	50 ms.	40 ms.

We want to determine the minimum number of replica C_{WS}, C_{AS} and C_{DB} for each server such that requests are served on the average in less than 2 sec.

Parameter sets

ID	K_{WS}	K_{AS}	K_{DB}	N_U	N_S	N_B	Z_U
1	30	60	8	50	100	10	1 min.
2	8	30	60	10	50	100	2 min.
3	8	8	60	100	10	50	1 min.
4	30	30	30	50	50	50	2 min.