Discrete Event Simulation and Queuing Theory

C. Carissimo: 10890661[a)]

(Dated: 3 December 2017)

Different queuing systems are simulated and experimented with on the basis of waiting time in queues. Results show that for constant server loads, systems with more servers have less waiting time. Furthermore, as the server load approaches 1, the observations required for accurate results increase drastically.

Keywords: Queuing Theory, Markov Chains

I. INTRODUCTION

This report will investigate Queuing Theory as a method for modelling stochastic systems, of which the precise functioning is unknown, and the only available information is the distribution of elements entering and leaving the system. Markov Chains are applied to this topic as the underlying mathematical model to describe such systems. Queuing systems will be investigated mathematically and numerically, with the use of the python package Simpy[1].

The report will discuss elements of Queuing Systems in section 1. In sections 2 the report will discuss experimental results of waiting times for different queueing systems.

II. QUEUING SYSTEMS

It is a strong fascination of people to simulate the flow of elements through a system, where the flow is a function of the rates that govern the arrival and departure from the system. Queuing theory, as the name suggests, simulates the flow of elements (conceptually often people) that arrive in a system and request the use of a service. Once serviced, the elements leave the system. Queuing systems typically have two rates that govern the flow[2]: an arrival rate, and a service rate. The relative balances of these two rates, which we will call ρ produce varied dynamics in the system.

It is rather intuitive, that if the rate at which people arrive in the system, λ and the rate at which people are serviced μ, follow this relationship

$$\lambda > \mu \tag{1}$$

then the elements in the system will tend to grow, where if given infinite time, this growth will tend to infinity.

Many interesting properties exist of these systems, of which books can and have been written. Most of these properties are not relevant for the purposes of this report, but can be investigated with papers that can be found in the bibliography.

One property that must be mentioned, which holds for a specific subset of queuing systems, is **memory-lessness**.

A. Markov Processes

Memory-lessness is a property of systems where, given a state S_k at time k, the state of the system S_{k+1} is solely determined by the state S_k. Of all continuous distributions, this is the case only for the exponential distribution.

Why is this property relevant or interesting? Well, it allows us to summarize all useful information of such a system as S_k, the state of the system at the given time. This holds true for a subset of queuing systems, known as M/M/N systems. The first two M's describe the distribution of the arrival and service times respectively, where the M's stand for Markov and the distributions are exponential. The N is the number of service points, or servers in the system. The memory-less property holds regardless of the number of servers.

B. M/M/N queues

Queues that are simulated in this form, as stated above can be summarized in terms of the state of the queue in a given time. For queues the state is the number of elements, or customers, that are in the system at a given time. From this state one of two things can happen: a new element will arrive, or an element will finish being serviced and leave.

From Little's Law, we can describe the average amount of people in the system,

$$\lim_{t\to\infty} \bar{N} = \lambda\bar{T} \tag{2}$$

where $\bar{T}$ is the mean time in the system, and we can also describe very similarly the mean waiting time in the queue,

$$\lim_{t\to\infty} \bar{N}_q = \lambda\bar{W} \tag{3}$$

where $\bar{W}$ is the mean waiting time in the queue and $\bar{N}_q$ is the mean number of people in the queue.

[a)] Master Computational Science, University of Amsterdam.

The waiting time for an M/M/N queue can be expressed with the following equation,

$$E[\bar{W}] = \Pi_W * \frac{1}{1-\rho} * \frac{1}{N * \mu} \tag{4}$$

where Π_W is the delay probability, the chance that an element upon arrival in the system has to wait to be serviced.

If we condsider the case where $N = 1$, and a case where $N = k$, it is clear to see that, ceteris paribus (everything else constant), the expression for waiting time in the queue where $N = k$ is reduced by a factor k. The only remaining term in the equation which changes other than N, is Π_W.

Π_W, as N grows larger, ceteris paribus becomes smaller. Thus, the waiting times in the queue decrease as the number of servers increases and the server load remains constant.

III. METHOD

In order to evaluate the waiting times of different queuing systems we simulated the aforementioned with python, and more specifically a python package called Simpy. Simpy allows the construction of a queuing system in a very reduced and simplified manner, by managing behind the scenes everything that is necessary to request a service, provide the service, determine whether or not the service is free to be used by the next customer, and a very easy implementation of multiple servers within the system.

All simulations are run with FIFO (first in first out) queuing, which is the most common form of a queue, particularly in commercial applications.

The object of interest from the simulations was the average waiting time for customers given a system load ρ. We focus on systems where ρ is close to 1. Each queue setup is tested for $\rho = 0.9, 0.93, 0.96, 0.99$.

For each queue, simulations are run for an arbitrarily large amount of time-steps, which in this case was $12 * 10^6$ time steps. Waiting times are saved for each customer that enters the system over the course of the simulation. The saved waiting times are then partitioned into 100 equally sized portions, where each portion contains $5 * 10^4$ observations. Between each portion, a random number of observations between 0 and 1000 is skipped, to increase the independence between observations.

Statistics are then calculated with a batching method. Mean waiting times are calculated for the portions according to their queuing system. A mean of those means is calculated and used as the mean waiting time for that particular system. The variance in waiting time for that system is calculated as the standard deviation of the means.

Confidence intervals are then calculated based on the standard deviations of the means, in the case that the means are normally distributed. Tests for normality are provided in each step. In the case that the null hypothesis that the distributions are not normally distributed can not be rejected at a 95% confidence level, the confidence intervals are not reported.

IV. RESULTS

Three main empirical results can be summarized across all simulations:

1. As ρ gets closer to 1 we observe an increase in average waiting times, alongside an increase in standard deviations.

2. For batches where the means were normally distributed, although the variance grew very large, the means are approximately precise to 10^0.

3. Waiting times are consistently shorter for queues with more servers.

The following headers in this section summarize all simulation results in tables. Each table contains four batched simulations, with a constant queuing system and a varying system load ρ. Batched means that were not normally distributed have ”na” insted of confidence intervals.

A. M/M/N

1. M/M/1

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	9.939262	9.931813	9.946712	0.849883
0.93	14.291485	14.276565	14.306405	1.702176
0.96	25.041071	24.987453	25.094688	6.116947
0.99	106.833348	106.281460	107.385235	62.962137

2. M/M/2

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	5.268784	5.265401	5.272167	0.385938
0.93	7.492170	7.485184	7.499157	0.797076
0.96	12.804119	12.782372	12.825865	2.480947
0.99	52.571383	52.201712	52.941055	42.174038

3. M/M/4

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	2.954699	2.952985	2.956413	0.195552
0.93	4.047843	4.044199	4.051487	0.415719
0.96	6.727141	6.715828	6.738453	1.290550
0.99	27.352815	27.202361	27.503269	17.164526

B. M/M/1 with priority

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	4.207256	na	na	0.209507
0.93	5.107133	5.10372	5.11054	0.389035
0.96	7.226620	7.21762	7.23562	1.026261
0.99	15.953831	15.8952	16.0125	6.690231

C. M/D/N

1. M/D/1

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	5.491195	na	na	0.298559
0.93	7.698258	na	na	0.715428
0.96	12.983305	na	na	1.669136
0.99	51.933667	51.7152	52.1521	24.920207

2. M/D/2

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	3.178352	3.17701	3.17969	0.152972
0.93	4.241886	na	na	0.287605
0.96	7.132011	7.12193	7.14209	1.150381
0.99	26.380574	26.2518	26.5094	14.694464

3. M/D/4

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	2.006269	na	na	0.068226
0.93	2.532357	2.53101	2.5337	0.153533
0.96	3.884674	3.88105	3.8883	0.413133
0.99	13.327682	13.2726	13.3827	6.278970

D. M/LT/N

ρ	$\mathbf{E}[\bar{W}]$	$\mathbf{E}[\bar{W}] - CI$	$\mathbf{E}[\bar{W}] + CI$	$\sqrt{Var}$
0.90	5073.7136	4528.0978	5619.3294	2783.7541

V. DISCUSSION

All queuing systems that were tested behaved as expected in that higher ρ produced higher average waiting times and a greater variance. It was also the case that with 50000 data points for each simulation and batches of 100 simulations we were able to obtain satisfactory confidence intervals, that pinpointed the mean waiting times to within a unit difference.

The M/M/1 queue with priority, compared to the standard M/M/1 queue had a significantly reduced waiting time, and the waiting times grew at a slower rate as ρ grew. This is an interesting property of prioritizing short jobs first. An attempt at an intuitive explanation for this result follows.

Waiting time is generated when all servers/service point are occupied and a new element joins the system. Upon joining the system the element will start generating waiting time. Any new elements that join the system will also start generating waiting time. Long jobs will on average generate more waiting time than short jobs, because the time it takes for the job to process determines the chances of generating waiting time. By prioritizing shorter jobs first, waiting time is generated at a slower pace because the short jobs that are processed are allowed to be completed and leave the system, making space for the next job which will have generated waiting time proportional to the length of the previous job. Of course, a decrease in average waiting time is obtained at the expense of increasing waiting times for long jobs.

One important difference of simulating a queuing system with priority is the assumption that the job length is known a priori. In practical applications, finding out the job length before a priori will require additional time to inspect the length of all jobs in the queue relative to all others. Furthermore, being able to precisely inspect job length might hold for a server where job length is directly proportional to package sizes, but will be an unstable assumption if humans are involved at all anywhere in the process (a bit of a joke, but you know it is true).

The M/D/N queues did not prove to be particularly different to any of the other queues, other than having a consistently small variance for lower server loads. This result can be explained quite simply as the result of stable service times. The deterministic component of these queues was the service time, and was set to 1. The variances are not only smaller, but also appear to grow at a slower rate.

Unfortunately for the M/LT/1 queue, the simulation results were very inaccurate. Running simulations for that type of queue proved to be computationally very intensive, to the point that only one batch was run, of 100 simulations with 100 data points in each simulation and a server load of 0.9. The confidence intervals that resulted from this simulation are ridiculously inaccurate.

VI. CONCLUSION

In conclusion, these different queuing systems exhibited similar characteristics when server loads increased. Variances got high, and waiting times became long. Very interesting of queuing systems is the interaction between different probability distributions in modelling queues. The memory less characteristic of Markov Processes (exponential probability distributions) makes them very fun and interesting to work with.

[1] Http://simpy.readthedocs.io/en/latest/contents.html.

[2] Willig, A. (1999). A short introduction to queueing theory. Technical University Berlin, Telecommunication Networks Group, 21.