A capacity planning / queueing theory primer or How far can you go on the back of an envelope? Elementary Tutorial CMG 87 *

Ethan D. Bolker
Departments of Mathematics and Computer Science
University of Massachusetts Boston
Boston, MA 02125-3393
eb@cs.umb.edu

May 28, 1987

Abstract

Most of the benefit of capacity planning for computer systems comes from applying common sense principles to easily understood (if not easily measured) data. If you can estimate business growth you need only simple arithmetic to figure out how busy your computer will be. But sometimes the fact that the system is subtle does matter: it's hard to predict response times. Then you must use mathematics (instead of arithmetic) or software which knows some - typically a package - incorporating a queueing network model of your computer

*This is a scanned version of a reprint of the original, converted to TEX. I wrote the paper while consulting for BGS Systems. The Computer Measurement Group probably owns the copyright, but the material is not available electronically. I doubt that they would object to this version.

system. You can (and should) think of the package as a black box whose sophisticated mathematical contents need not concern you. In this top down view such products are worth what they cost because they're cost effective.

But from time to time you may want to peek into the box – to satisfy a healthy curiosity, or to convince yourself that the mathematics there makes sense, so that you can use its output with confidence or explain its function to your manager. (But exercise care– you can sell your services to your manager only by showing s/he needs them, not by describing the mathematical magic that helps you do your job.

This elementary tutorial moves from simple capacity planning arithmetic to a glimpse of queueing theory – how it solves relevant problems, and why it's subtle. Planners need to know this material. But the tutorial is intended for a less technical audience too – managers, secretaries, planners' husbands and wives – anyone who wants to know a little about how a queueing theory package earns its keep. It's essentially what I tell my friends when they ask what I do.

1 The Numbers Game

Imagine that you are managing a bank's Cash Machines, and that you know something about how much business a particular machine will be called upon to do. To be specific, suppose you anticipate that

30 customers will visit the Machine in an hour
Each will use it for a minute and a half.

You need answers to two questions:

How busy is the Cash Machine?
How long do customers wait?

Before we answer them, we can note a pattern. The numbers we know concern customer behavior: arrival rate (30 customers/hour) and service requested (1.5 minutes/customer). The numbers we want depend on the how the customers interact with the system: the percentage of time the Machine is busy and the average customer wait.

You can measure all of these numbers by hiring a High School student to watch the Machine, count customers and time their transactions. Or you can

program the Cash Machine to keep track of the first three while it does the business it was bought to do, and program the door to the room in which the Machine lives to find the last one. That's "Computer Measurement." But CMG members don't just measure, they plan. In a planning scenario the first two numbers – the customer arrival rate and the service each customer requires – are part of the business forecast. The planner needs to predict the second two: how busy the machine will be, and the waiting times. That's where mathematics comes in.

Simple arithmetic tells us how busy the Machine will be:

$$\frac{30 \text{ customers}}{\text{hour}} \times \frac{1.5 \text{ minutes}}{\text{customer}} = \frac{45 \text{ minutes}}{\text{hour}}.$$

The Cash Machine is in use for 45 minutes each hour, so it is busy 75% of the time. The underlying algebra is quite straightforward, and tedious.

2 The Waiting Game

The second question is harder. Let's reflect on it a bit. Suppose the 30 hourly customers came at scheduled times 2 minutes apart, and that each customer was an average customer and did just 1.5 minutes of banking. Then no one would have to wait, and the Cash Machine would even have 30 seconds to cool down between jobs. But common sense and our own banking experience tell us these are unrealistic assumptions. Customers come one every two minutes on the average. But sometimes there will be long idle times; sometimes several customers will arrive nearly at the same time. And although on the average customers spend 1.5 minutes at the Cash Machine some are just getting FastCash while others do a week's worth of banking all at once. The more variability of this kind the more time customers will spend waiting while someone else is using the Cash Machine. Predicting the average waiting time is harder than calculating 75% utilization. The formula that helps us out is

$$\text{Average number of customers at the Cash Machine} = \frac{\text{utilization}}{1 - \text{utilization}}$$

In our example, since the Cash Machine is 75% busy,

$$\text{Average number of customers at the Cash Machine} = \frac{75}{100-75} = 3.$$

The formula we've just used is not straightforward. To discover it requires some mathematics at about the college (Calculus) level. Fortunately, we needn't discover it – we need only understand it. And to see why it's reasonable all that's required is some High School algebra, which we'll see in the tutorial. To use the formula, recall that each of the 3 customers at the machine needs 1.5 minutes to transact his business. Therefore the average new arrival will have to wait for

$$3 \text{ customers} \times \frac{1.5 \text{ minutes}}{\text{customer}} = 4.5\text{minutes}$$

The answers to our two questions sharply focus the capacity planning dilemma you face as the Cash Machine Manager: dissatisfaction both because

the 75% utilization is too low.
the 4.5 minute wait is too long.

3 What if?

A sad consequence of the formula we found which connects the utilization and the average number of waiting customers tells you exactly how decreasing your unhappiness with one of the answers will increase it with the other.

For example, if you expect Cash Machine customers will soon be arriving at a rate of 36 rather than 30 per hour (a 20% increase) then Machine utilization will increase to 90% (which makes you happy) but the average number of people at the Machine will increase to 9 and the average customer's wait will be 13.5 minutes. This is three times the 4.5 minutes it is now, which makes you – and the customer – unhappy.

So far all the arithmetic fits on the back of an envelope.

You know your customers will not tolerate those long waits (they don'tappreciate the fact that you like to keep the Machine busy) so you decide to install a second Machine next to the first.

What are the answers now to the two central capacity planning questions? Will each machine be busy 45% of the time? Will the average wait be half of 13.5 minutes? Again, the first question is easy; the second is hard.

Each of the two Machines will indeed be in use just 45% of the time. But the average number of customers present at the pair of Machines turns out to be a mere 1.3, and, on the average, an arriving customer will wait less than two minutes before his turn comes. We will briefly discuss the mathematics which produces these numbers. The resulting formulas are still simple enough to evaluate with a calculator, but showing where the formulas come from is a little too complicated for an elementary tutorial.

4 Parallel Processing

Moreover if we install the second Cash Machine and measure waiting times we will discover that our predictions are too optimistic (and that's the wrong direction in which a planner should err.) To understand why we must think more carefully about what happens when both Cash Machines are in use.

Let us analyze the 1.5 minutes each customer spends standing at the Cash Machine when there is just one Machine in use. Some of that time, say, 1 minute, is time spent thinking about which buttons to press, and pressing them. The remaining 0.5 minute is the time it takes the computer to which the Cash Machine is linked to process the transaction.

To study that half minute, first imagine the bank in the days before computers, when the Cash Machines were tellers. The teller took the customer's request to the bookkeeper, (suppose there was just one bookkeeper) who asked a clerk to bring him the customer's account record from the appropriate filing cabinet. When the record arrived the bookkeeper did the necessary arithmetic, sent the teller back with an answer for the customer and sent the clerk back to file the updated account record. This sequence of jobs, which used to take, say, 3 minutes, accounts for the 30 seconds we measure now when computers do the bookkeeping and record management.

Observe how wasteful it is. The bookkeeper is (presumably) better paid than the clerk, but is idle most of the time (since his arithmetic uses a small fraction of the 30 seconds' processing time while the retrieval of the record takes up the rest). That suggests hiring several tellers and clerks, to keep one bookkeeper busier.

When two Cash Machines are installed side by side and are simultaneously in use each customer still spends 1 minute thinking. But once each has told the computer to process a transaction it will take the computer more than 30 seconds to comply. The teller/bookkeeper/file cabinet analogy explains

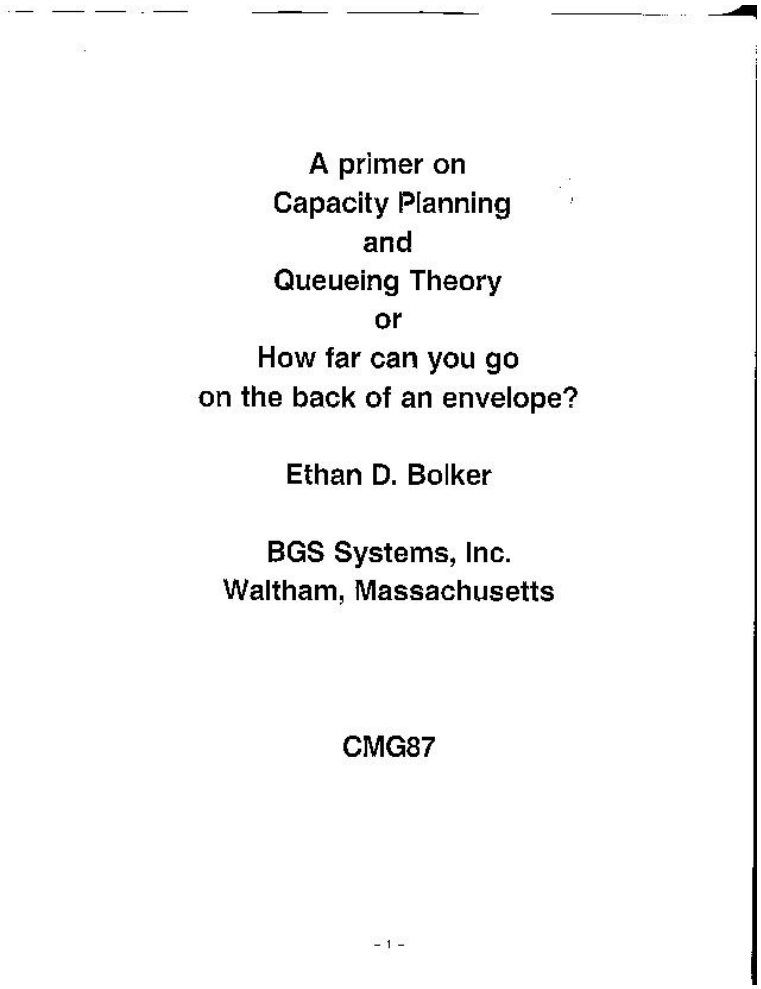
why. Although several tellers can serve several customers simultaneously the bookkeeper can work on only one record at a time. And only one clerk can use a filing cabinet at a time. (Although two clerks can get accounts from separate filing cabinets while the bookkeeper works on a third account.) Therefore some of the time one customer's work will have to wait while the bookkeeper is busy with another's, or while the right filing cabinet is in use. These internal delays mean that when there are several tellers serving customers simultaneously each customer will wait longer than the 3 minutes of real processing time his job requires. Moreover, our analogy works even when the teller is a Cash Machine and the bookkeeper and file cabinets have been replaced by a computer. The computer can process only one job at a time at the CPU (the Central Processing Unit, which is the computer's bookkeeping hardware) and can look up only one account at a time on each disk (the computer equivalent of a file cabinet), although it can look up one account while it is processing work for another. Therefore, as before, some of the time one customer's work will have to wait while another's is being done. Therefore, as before, when there are two (or 200) Cash Machines active simultaneously the computer needs more than 1/2 minute to process each customer's work. Just how much more is the crucial question.

5 Black Boxes

In the tutorial we will see some of how one might calculate how long each job will wait at the disks and at the CPU and thus predict how long the average customer will have to spend between the time he arrives to wait for the Cash Machine and the time his banking is complete. The calculations resemble the ones we have just looked at but they are so much more complicated that they can be done rapidly and economically only by a computer program – a software package which knows queueing theory. As a planner, you speak to the package in the the language you use to describe your computer system – not the mathematical language it uses internally to predict how the system will behave. It answers you in your terms too, telling you about utilizations and response times. You use those numbers to plan computer capacity to cope with expected business needs. And now, when you do, you'll know a little more about where those numbers come from.

The tutorial

The next pages contain images of the transparencies I used when I presented the tutorial at the CMG meeting. If I were to do the talk again I would convert them to powerpoint. Then the software would manage the pages I used as overlays (marked as such).



baking bread,
fixing bicycles
providing service
giving advice

. . . well
. . . on time
. . . economically

. . . profitably

requires planning

the planner's

customer:
the company

product:
advice

data:
the current state of affairs,
business plans,
service objectives,
financial constraints

tools:
common sense,
. . . mathematics

an ATM example:

	growth	arrivals (cust/hr)
now		30
March	20%	36
May	25%	45

can the ATM do the job?

Analysis:

how busy is the ATM now?

1.5 min/cust service time (measured)

30 cust/hr * 1.5 min/cust

= 45 min / hr

= 45 min / 60 min = 0.75

busy 75% of the time

Utilization law:

U = arrival rate * service time

more likely : measure U ; deduce

service time = U / arrival rate

= 0.75 / (30 cust/hr)

= 1.5 min/cust

	arrivals (cust/hr)	utilization
now	30	75%
March	36	90%
May	45	112%

Trouble in May !

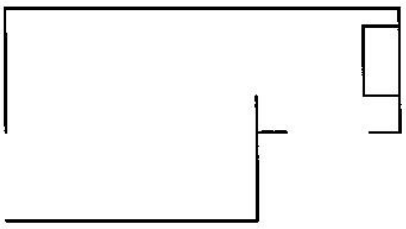
Is March OK ?

I used the picture of the ATM on this page to simulate customer traffic - coins of different sizes placed on top of the transparency. I moved them from the waiting room into the ATM room while I spoke.

Analysis:

how long do customers spend?

response time = service + wait



if scheduled:

0 wait

response = service

= 1.5 minutes

if random:

. . . magic formula

Response Time Law

response = service / (1 – U)

1.5 / (1 – 0.75) = 6

	arrivals (cust/hr)	utilization	response (minutes)
now	30	0.75	6
March	36	0.90	15
May	45	1.12	?

now ATM is 75 % busy

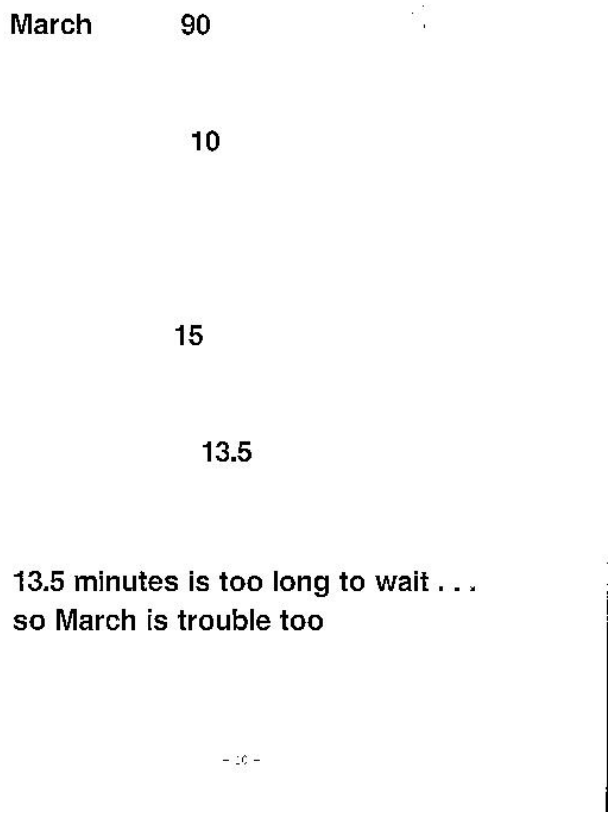
1 customer in 4 has no wait at all

but the average customer

spends 6 minutes

waits for 4.5

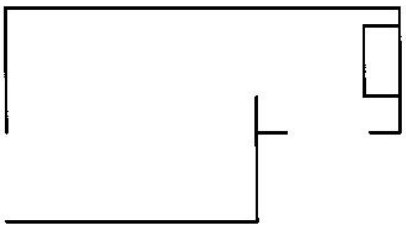
This page fits over the previous one.



Use the coins again as simulated customers to think about Little's Law.

Why trust a magic formula ?

Q = average # of customers at ATM
(in the room)



Little's Law
Q = arrival rate * response time

how does this help?
why is it true?

Little's Law => Response Time Law

resp = service + wait
= service + service * Q
(for random arrivals)
= service + service * arr * resp,
(that's Little's Law)
= service + U * resp

now solve for resp :

resp * (1 – U) = service

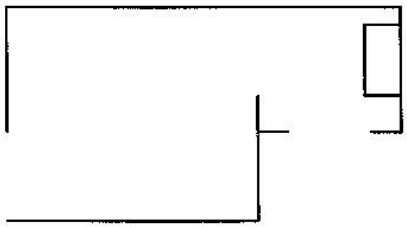
so

resp = service / (1 – U)

Why is Little's Law true?

U = arrival * service
Q = arrival * response

U = 0.75 means
1 customer 75% of the time, or
0.75 customers all the time



response = service in ATM room

Q = 30 cust/hr * 6 min/cust
= 180 min/hr
= 3 customers at a time (avg)

Back to the drawing board for March
Install a second ATM

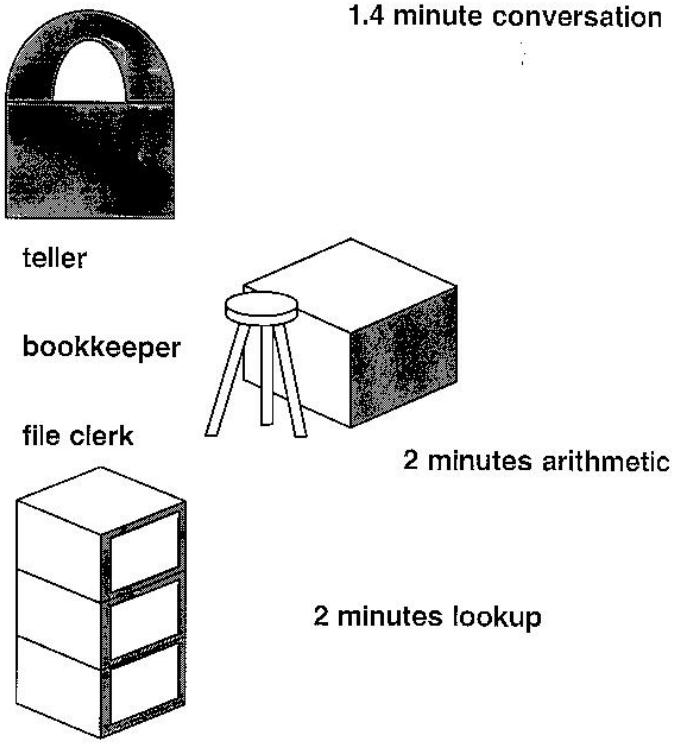
	arrivals (cust/hr)	utilization	response (min)
March	36/2	45%	2.7
May	45/2	61%	3.8

(response time law overestimates)

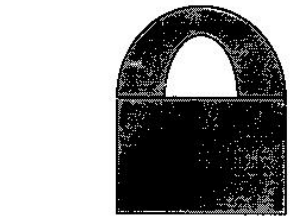
good news:
good response

bad news:
1.5 minute service increases,
so we've underestimated

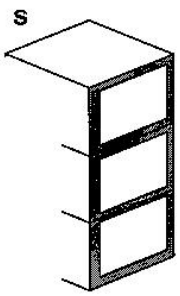
Imagine the bank with a single teller.



This transparency on top of the previous one creates the two teller bank, introducing waits for the bookkeeper and the file clerk.



s

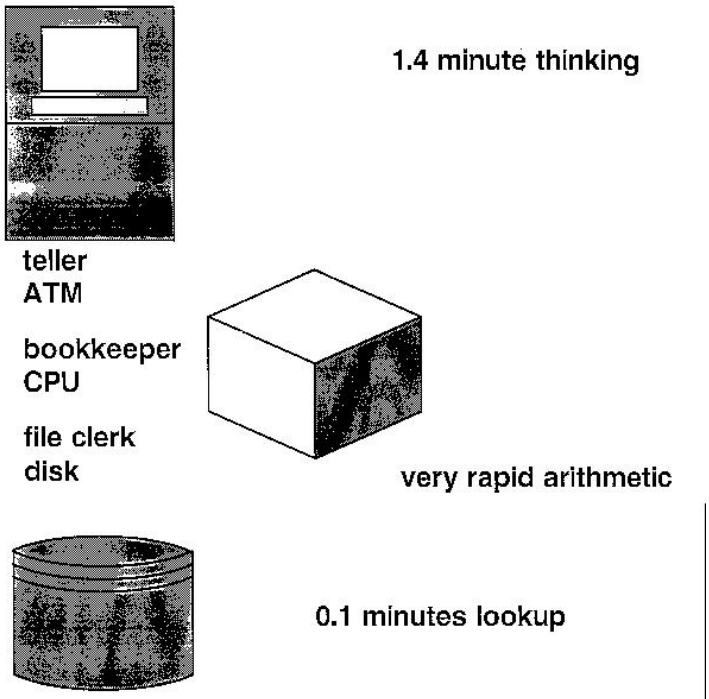


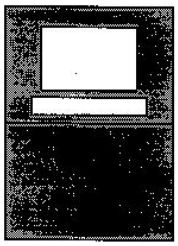
s

wait, then

wait, then

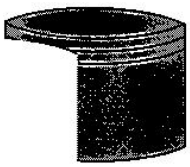
The next two transparencies replace teller/bookkeeper/file clerk with ATM/CPU/disk.





S
S

S wait, then
S



wait, then

calculate internal waiting times
predict performance,
recommend solutions

but not on the back of an envelope

to understand a complicated system
well enough to plan its future you need
mathematical models, and
tools to
    build them
    – measure the system now
    modify them
    – predict future performance
    – test effectiveness of plans

Are your software tools black boxes?

. . . yes

your queueing theory oracle tells you
of trouble to come
which solutions will work

. . . and no

you must understand it well enough
to trust it yourself
to show your manager why
s/he can trust you

Three minutes' thought would suffice to find this out, but thought is irksome and three minutes is a long time.

– A. E. Houseman,
Juvenalis Saturae