

1 – Performance Analysis and Simulation

List of Slides

- 1 Performance Analysis and Simulation
- 4 Queueing Systems, the Big Picture
- 5 Kendall's Notation
- 6 Kendall's Notation Continued
- 7 Queueing System Parameters
- 9 The Meaning of τ
- 10 Queue Structure and Parameters
- 11 Basic Queueing Relationships
- 12 Service Time Distributions
- 13 $M/M/1$ Steady State Formulas
- 14 An Example, the Situation
- 15 An Example, The Meeting
- 16 Solving Part 1 (i)

- 17 Solving Parts 1 (ii)
- 18 Solving Parts 1 (iii)
- 19 Part 2 — The Next Meeting
- 20 Part 2 case 1
- 21 Part 2 case 1 Continued
- 23 Part 2 case 2
- 25 Part 2 case 3
- 26 $M/M/N$ Steady State Formulas
- 28 Part 2 case 3
- 31 Analysis Back To Boss
- 32 Response from the Boss
- 33 The $M/M/1/B$ Formulas 1 of 3
- 34 The $M/M/1/B$ Formulas 2 of 2
- 35 The $M/M/1/B$ Solution 1 of 3
- 36 The $M/M/1/B$ Solution 2 of 3
- 37 The $M/M/1/B$ Solution 3 of 3
- 38 The $M/M/m/B$ Formulas 1 of 3

- 40 The $M/M/m/B$ Formulas 2 of 3
- 41 The $M/M/m/B$ Formulas 3 of 3
- 42 Some Observations

2 – Queueing Systems, the Big Picture

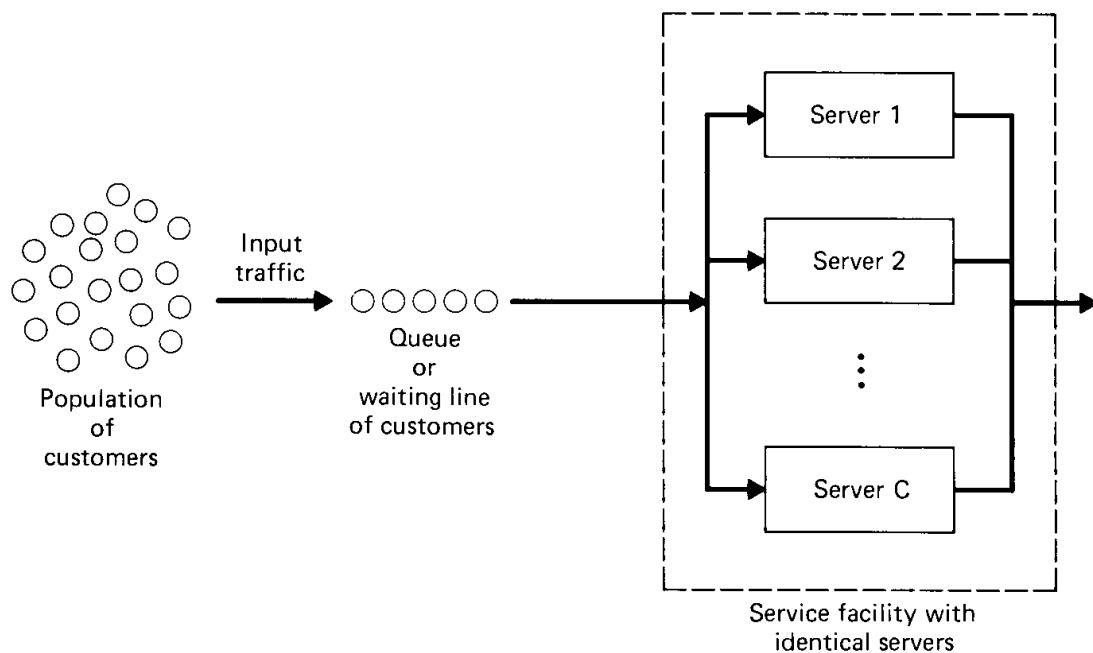


Figure 1: Elements of a Queueing System [1]

3 – Kendall's Notation

Kendall's notation [1, 2] for describing queueing systems is of the form $A/B/c/K/m/Z$.

Symbol	Meaning
A	Interarrival Time Distribution
B	Service Time Distribution
c	Number of servers
K	System's Queue Capacity
m	the number of items in the source
Z	the queue discipline

Table 1: Kendall's notation [2, 1]

An abbreviated notation $A/B/c$ is used when:

1. $K = \infty$, queue length is unbounded,
2. $m = \infty$, the number of "customers" is unbounded and
3. $Z = \text{FCFS}$, the queueing discipline is FCFS.

4 – Kendall's Notation Continued

The arrival and service time distributions are:

GI	generally independent interarrival time
G	generally independent service time
E_k	Erlang- k distribution
M	exponential distribution
D	deterministic distribution
H_k	hyperexponential distribution

Generally independent distributions mean that the exact distribution is unknown.

5 – Queueing System Parameters

Symbol	Meaning
B	The number of buffers in the system
β	the probability all servers are busy
λ	mean arrival rate
m	number of servers
N	number of servers
μ	mean service rate per item, $\mu = \frac{1}{s}$
q	mean number of items in system
ρ	server utilization
s	mean service time for each arrival
σ_q	standard deviation of q
σ_{t_q}	standard deviation of t_q
σ_s	standard deviation of s
σ_w	standard deviation of w
t_q	mean time an item spends in the system
t_w	mean time an item waits for service
τ	mean interarrival time
u	traffic intensity (for $M/M/m$ and $M/M/m/B$)
w	mean number of items waiting in the queue

Table 2: Queueing System Parameters [1, 3]

6 – The Meaning of τ

Suppose that requests arrive in our queuing system at times t_1, t_2, \dots, t_k where $0 < k \leq \infty$.

The i th *interarrival time* between requests is:

$$\Delta t_i = t_{i+1} - t_i \quad \text{for } i \geq 1 \quad (1)$$

τ is the mean interarrival time, so:

$$\tau = E[\Delta t] = \frac{\sum_{i=1}^k \Delta t_i}{k} \quad (2)$$

If Δt_{i+1} depends on Δt_i then (by definition) there is some function f satisfying: $\Delta t_{i+1} = f(\Delta t_i)$.

The interarrival time distribution is said to be *Memoryless* if there is no function f satisfying $\Delta t_{i+1} = f(\Delta t_i)$.

7 – Queue Structure and Parameters

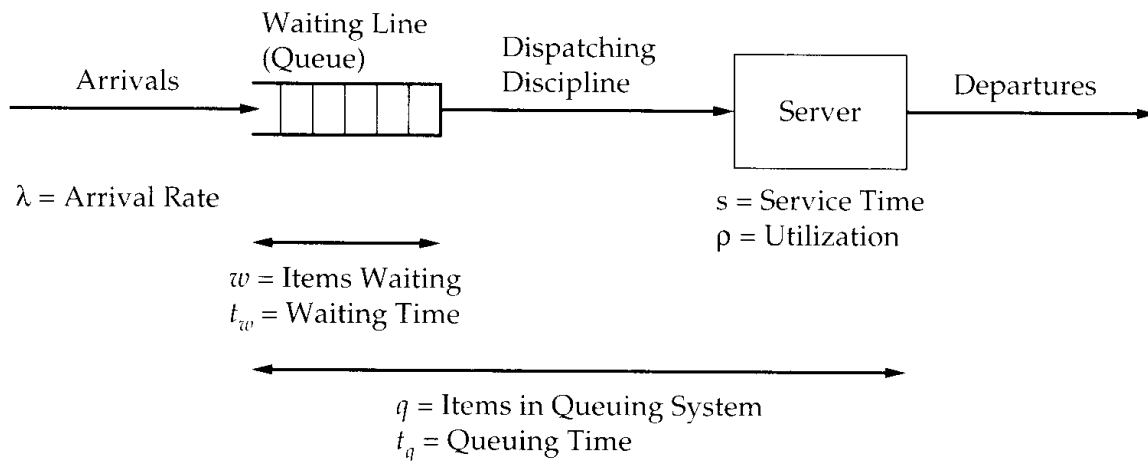


Figure 2: Queueing System Structure

8 – Basic Queueing Relationships

The following hold true regardless of queueing discipline:

$\lambda < \mu N$	Stability condition
$\rho = \lambda s$	single server case
$\rho = \frac{\lambda s}{N}$	general case
$q = \lambda t_q$	Little's Law for System
$w = \lambda t_w$	Applied to Queues
$q = w + \rho$	single server case
$q = w + N\rho$	general case

9 – Service Time Distributions

Service times are modeled as R.V.s with mean s using distributions from Table 3.

B	Meaning	Properties
G	General Independent	Less Informative
D	Deterministic	Higher Performance
M	Exponential	Worse Performance

Table 3: Common Service Time Distributions (B)

M stands for *Memoryless*, Stallings [3] uses N for the number of servers.

The $M/M/1$ and $M/M/N$ models are the most frequently used, with $M/D/N$ a close second.

Let's begin with $M/M/1$ and $M/M/N$ models.

10 – $M/M/1$ Steady State Formulas

The parameters governing the $M/M/1$ steady state formulas are:

- $\lambda = \frac{1}{\tau}$ — Mean interarrival rate
- $\mu = \frac{1}{s}$ — Mean service rate

Recalling that $\rho = \lambda s$ it can be shown that:

$$q = \frac{\rho}{1 - \rho} \quad (3)$$

$$w = \frac{\rho^2}{1 - \rho} \quad (4)$$

$$t_q = \frac{s}{1 - \rho} \quad (5)$$

$$t_w = \frac{\rho s}{1 - \rho} \quad (6)$$

$$m_{t_q}(r) = t_q \ln\left(\frac{100}{100 - r}\right) \quad (7)$$

$$m_{t_w}(r) = \frac{t_w}{\rho} \ln\left(\frac{100\rho}{100 - r}\right) \quad (8)$$

11 – An Example, the Situation

Your company is an internet provider with a Web server, which has a recently increased load of 120 users. Each user (on average) hits your site every 250 seconds (exponentially distribution). The hits are processed in FIFO order and are not preempted. Your server can service each request on average in 2 seconds, with service times being exponentially distributed. Your server has 128MB of memory, the O/S and server software uses 32MB and each request uses 4 MB, and has recently become unreliable.

Users are complaining about excessively long delays in service, and are citing quality of service contracts. These contracts specify that 90% of the users get service in no more than 8 seconds and that the average time for waiting for a response from your server does not exceed 4 seconds.

Additionally the server has begun to run out of memory and crash, and the load is suspected.

12 – An Example, The Meeting

Part 1) Your manager suspects the users are exaggerating, and asks you to look into it. (i) What is the mean current system response time, (ii) the 90th percentile of response time? (iii) Additionally since your server crashes frequently could these problems reflect an out of memory condition?

13 – Solving Part 1 (i)

We first need to solve for the parameters used to compute (i) t_q the mean time to service a hit.

We know $t_q = \frac{s}{1-\rho}$.

$s = 2$ is given, but the utilization, $\rho = \lambda s$, is unknown.

The average time between requests is $\tau = \frac{250}{120}$.

Solving for the arrival rate:

$$\lambda = \frac{1}{\tau} = \frac{120}{250} = 0.48 \quad (9)$$

So the utilization is:

$$\rho = \lambda s = 0.96 \quad (10)$$

which is very high, our time in the system is:

$$t_q = \frac{2}{1 - 0.96} = 50\text{sec.} \quad (11)$$

In fact jobs are waiting:

$$t_w = \frac{\rho s}{1 - \rho} = \frac{2 \times 0.96}{1 - 0.96} = 48\text{sec.} \quad (12)$$

14 – Solving Parts 1 (ii)

Solving for (ii) $m_{t_q}(90)$ the 90th percentile:

$$m_{t_q}(90) = t_q \ln\left(\frac{100}{100 - 90}\right) = 50 \ln 10 \approx 115.13\text{sec.} \quad (13)$$

far above your allowable value. In fact 90% percentile for queueing times is:

$$\begin{aligned} m_{t_w}(90) &= \frac{t_w}{\rho} \ln\left(\frac{100\rho}{100 - 90}\right) \\ &= 50 \ln\left(\frac{48}{5}\right) \\ &\approx 113.09\text{sec.} \end{aligned} \quad (14)$$

They are about (but not exactly) s seconds apart, which is expected (since s is the mean service time).

15 – Solving Parts 1 (iii)

(iii) q , the *mean* number of requests in the system. Using Little's law we get:

$$q = \lambda t_q = \frac{\rho}{1 - \rho} = \frac{0.96}{1 - 0.96} = 24 \quad (15)$$

Letting (for this page/section only) $R = 64\text{MB}$ represent the amount of available RAM,

$R_S = 32\text{MB}$ represent the memory used by the Server and O/S and the memory required per request in the system be $R_R = \frac{4\text{MB}}{\text{request}}$ then:

$$\begin{aligned} R &> R_S + qR_R \\ &> 32\text{MB} + 24\text{requests} \left(\frac{4\text{MB}}{\text{request}} \right) = 128\text{MB} \end{aligned}$$

Which implies that the server is approaching the limits of its memory during normal operation (a bad thing).

16 – Part 2 — The Next Meeting

Your analysis has convinced your boss that the system is indeed in big trouble, and a decision has been made to purchase more hardware.

Your boss has come up with the following solutions:

1. Buy another server, mirror the data, and put 1/2 of the users on each server, for \$10,000.
2. Upgrade to a faster processor and to 128MB memory in the same box for \$10,000, so it can now process each users request in 1.3sec..
3. Upgrade buy a two processor box (using the same type of processors) with 256MB memory in the same box for \$14,000.

Which one should your company buy?

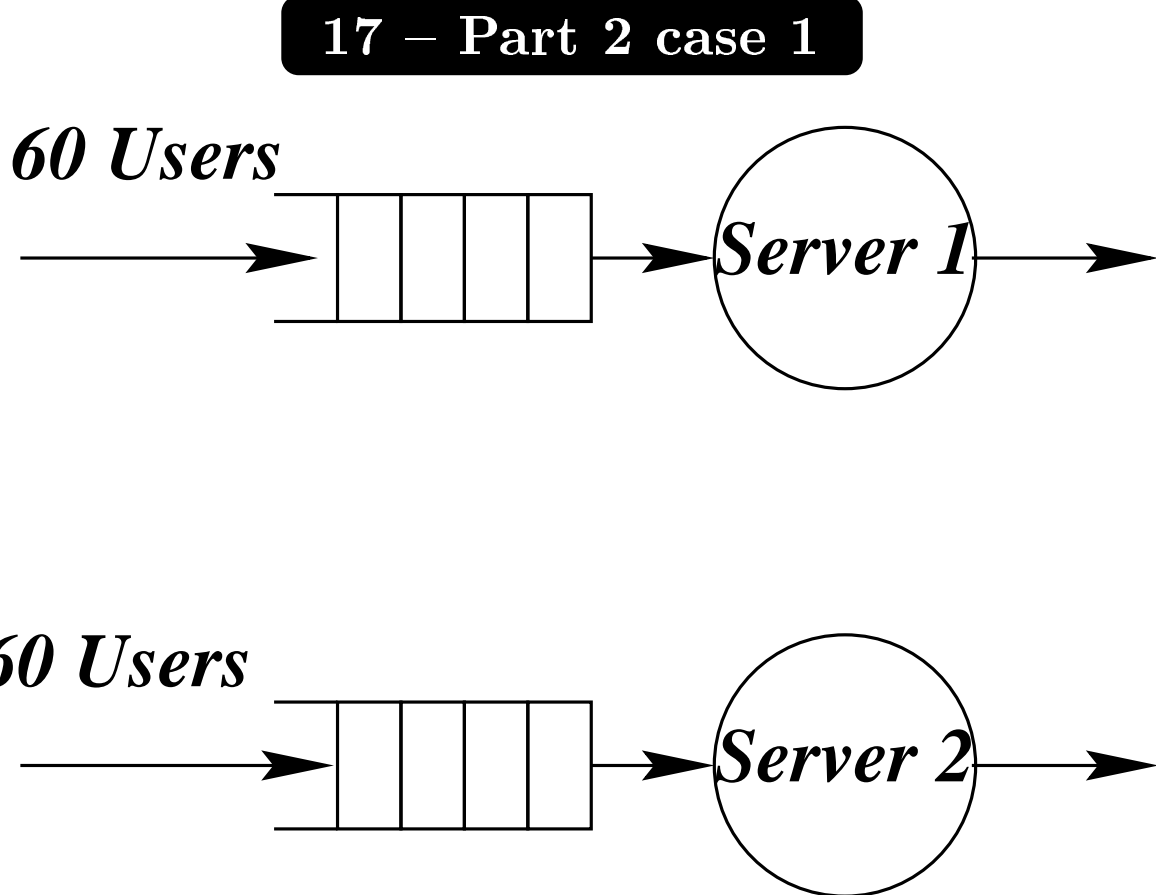


Figure 3: Partitioning the load over 2 servers with separate queues

18 – Part 2 case 1 Continued

With the reduced load on each server, memory is no longer a problem.

Repeating the steps in Part 1(i) we find that the average time between requests for each server is $\tau = \frac{250}{60}$ in this case, so:

$$\lambda = \frac{1}{\tau} = \frac{60}{250} = 0.24 \quad (16)$$

Giving each server a utilization of:

$$\rho = \lambda s = 0.48 \quad (17)$$

which is better, but the mean job's time in the system is:

$$t_q = \frac{s}{1 - \rho} = \frac{2}{1 - 0.48} = \frac{50}{13} \approx 3.85\text{sec.} \quad (18)$$

Meeting the minimum average response time criteria.

Checking the 90th percentile of the system response times shows:

$$\begin{aligned} m_{t_q}(90) &= t_q \ln\left(\frac{100}{100-90}\right) = \\ &= \frac{50}{13} \ln 10 \approx 8.85\text{sec.} \end{aligned} \quad (19)$$

Not good enough, it had to be better than 8 seconds.

19 – Part 2 case 2

With the increased memory on the server, memory is no longer a problem.

We repeat the analysis from Part 1 with the new mean service time of $s = 1.3$.

Recall the mean interarrival time between

requests was $\tau = \frac{250}{120}$, so again

$$\lambda = \frac{1}{\tau} = \frac{120}{250} = 0.48.$$

The utilization becomes:

$$\rho = \lambda s = 0.624 \quad (20)$$

The mean time a job stays in the system is:

$$t_q = \frac{s}{1 - \rho} = \frac{1.3}{1 - 0.624} \approx 3.46\text{sec.} \quad (21)$$

which is better than in case 1, and satisfies our requirement $t_q \leq 4$.

Checking the 90th percentile of response times shows:

$$\begin{aligned} m_{t_q}(90) &= t_q \ln\left(\frac{100}{100 - 90}\right) = \\ &\approx 7.96\text{sec.} \end{aligned} \tag{22}$$

which just barely satisfies the $m_{t_w}(90) \leq 8\text{sec.}$ part of the requirement.

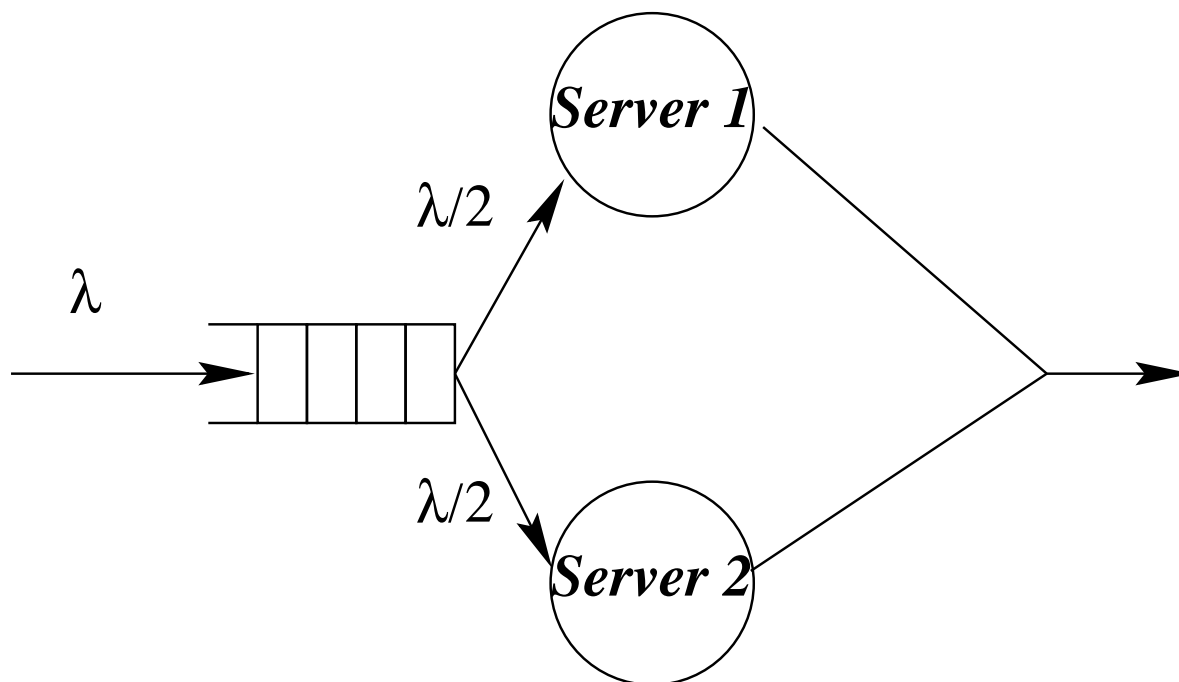
20 – Part 2 case 3

Figure 4: Partitioning the load over 2 servers with a shared queue

This is an $M/M/2$ model, and we need to apply the $M/M/N$ formulas here.

21 – $M/M/N$ Steady State Formulas

Assume that there are N servers, each with equal expected service times and load is evenly partitioned across them. We can use the formulas both Stallings [3] and Deitel [1].

Traffic intensity, u , measures global system utilization which is different from the per server utilization, ρ :

$$u = \lambda s \quad (23)$$

$$\rho = \frac{u}{N} = \frac{\lambda s}{N} \quad (24)$$

For a given u , the probability that all N servers are busy is:

$$\beta = \frac{\frac{u^N}{N!}}{\frac{u^N}{N!} + (1 - \rho) \sum_{k=0}^{N-1} \frac{u^k}{k!}} \quad (25)$$

And allows solution for the following:

$$q = \lambda t_q \quad (26)$$

$$t_q = t_w + s \quad (27)$$

$$t_w = \frac{\beta s}{N(1 - \rho)} \quad (28)$$

The r th (and 90th) percentile for t_w (waiting in a queue) is:

$$m_{t_w}(r) = \max\left(0, \frac{t_w}{\beta} \ln\left(\frac{100\beta}{100 - r}\right)\right) \quad (29)$$

$$m_{t_w}(90) = \frac{t_w}{\beta} \ln(10\beta) \quad (30)$$

22 – Part 2 case 3

We apply $M/M/2$ formulas here.

We know $s = 2$ from our given, and from Part 1

(i) we know the interarrival time, $\tau = \frac{250}{120}$, so:

$$\lambda = \frac{1}{\tau} = \frac{12}{25} = 0.48 \quad (31)$$

The traffic intensity, u , and utilization, ρ are:

$$u = \lambda s = 0.48 \times 2 = 0.96 \quad (32)$$

$$\rho = \frac{u}{N} = 0.48 \quad (33)$$

The probability of all servers being busy, β is:

$$\beta = \frac{\frac{u^N}{N!}}{\frac{u^N}{N!} + (1 - \rho) \sum_{k=0}^{N-1} \frac{u^k}{k!}} \quad (34)$$

$$= \frac{\frac{u^2}{2}}{\frac{u^2}{2} (1 - \rho)(1 + u)} = \frac{288}{925} \approx 0.311 \quad (35)$$

Solving for the mean time in the queue t_w and the mean time in the system t_q gives:

$$t_w = \frac{\beta s}{N(1 - \rho)} = \frac{288}{481} \approx 0.60\text{sec.} \quad (36)$$

$$t_q = t_w + s = \frac{1250}{481} \approx 2.60\text{sec.} \quad (37)$$

Which is well within our required $t_q < 4\text{sec.}$

Computing the 90th percentile for t_w gives:

$$\begin{aligned}m_{t_w}(90) &= \frac{s}{N(1-\rho)} \ln(10\beta) \\ &= \frac{25}{13} \ln\left(\frac{576}{185}\right) \\ &\approx 2.18 \text{sec.}\end{aligned}\tag{38}$$

We approximate $m_{t_q}(90)$ by (this may not be the official way):

$$m_{t_q}(90) \approx m_{t_w} + s \approx 4.2 \text{sec.}\tag{39}$$

Which is well below our $m_{t_q}(90) \leq 8$ requirement.

23 – Analysis Back To Boss

If the number of users is not expected to grow, and the requirements are not really tight then the upgraded uniprocessor and memory is a good bargain.

Otherwise, if expansion of service is expected, or the requirements are tight, then the \$14,000 tightly coupled dual processor box with 128MB RAM is needed. It can be shown that another 50 users can be added while still meeting the requirements.

As a short term workaround you suggest letting the server drop requests when there is no more buffer space instead of crashing.

24 – Response from the Boss

Negotiations with the customers indicate will tolerate drops if no more than 1% of requests (on average) were dropped. The boss wants to use this pending delivery of the new server, since the users hate crashes and outages. Memory can be added in $\frac{128\$}{128\text{MB}}$ increments allowing the number of buffers to be increased up to a maximum of 1GB.

The boss asks you to find if we can achieve this solution by simply adding memory and dropping packets, and if so, how much will it cost? What will the response time be?

So now what do we do?

25 – The $M/M/1/B$ Formulas 1 of 3

With your top notch systems background, you realize that the system is a $M/M/1/B$ system.

Traffic intensity is defined as $u = \lambda s$, since stability is guaranteed by automatic shedding of load (by dropping requests), we only need to consider $0 \leq u \leq 1$

The probability of n jobs being in the system at any given instant is:

$$p_0 = \begin{cases} \frac{1-u}{1-u^{B+1}}, & u \neq 1 \\ \frac{1}{B+1}, & u = 1 \end{cases} \quad (40)$$

$$p_n = \begin{cases} \frac{1-u}{1-u^{B+1}} u^n, & 0 \leq n < B \wedge u \neq 1 \\ \frac{1}{B+1}, & u = 1 \\ 0, & n > B \end{cases} \quad (41)$$

(42)

26 – The $M/M/1/B$ Formulas 2 of 2

The net mean rate of arrival, λ' , takes the probability of not having any available buffers:

$$\lambda' = \lambda(1 - p_B) \quad (43)$$

This lets us solve for the other parameters:

$$q = \frac{u}{1 - u} - \frac{(B + 1)u^{B+1}}{1 - u^{B+1}} \quad (44)$$

$$w = \frac{u}{1 - u} - u \frac{1 + Bu^B}{1 - u^{B+1}} \quad (45)$$

$$t_q = \frac{q}{\lambda'} \quad (46)$$

$$t_w = t_q - s \quad (47)$$

It can be shown that the mean percentage of dropped packets is:

$$\text{Percent Dropped} = 100\left(1 - \frac{\lambda'}{\lambda}\right) \quad (48)$$

27 – The $M/M/1/B$ Solution 1 of 3

First we estimate whether we can get the desired drop rate by adding RAM.

Recall that $\lambda = 0.48$, $s = 2$, and $B = 24$.

Then traffic intensity is $u = 0.48 \times 2 = 0.96$.

We need to estimate the net mean rate of arrival, λ' taking into account the dropped arrivals, which requires we know the probability of $n = B = 24$ jobs being in the system. Since $0 < u < 1$ we know.

$$p_B = 1 - u \frac{1 - u^{B+1}}{1 - u} \approx 0.0235 \quad (49)$$

$$\lambda' = \lambda(1 - p_B) \approx 0.469 \quad (50)$$

Numerically solving (using Maple) we see that the percentage of dropped packets is below 1% when $B \geq 40$, which means we need to buy one more 128MB RAM unit.

28 – The $M/M/1/B$ Solution 2 of 3

Adding 128 MB will give us a total of 256 MB and then:

$$\text{Buffer Capacity} = \frac{R_M - R_S}{R_R} = \frac{256 - 32}{4} = 56 \quad (51)$$

If we use all 56 available buffers we can decrease the drop rate to about 0.45% and have a higher $\lambda' = 0.478$.

Is there a trade off in using all the available buffers?

29 – The $M/M/1/B$ Solution 3 of 3

Substituting into our equations, $B = 40$,
 $\lambda' = 0.469$, and $u = 0.96$, we get:

$$q = \frac{u}{1-u} - \frac{(B+1)\rho^{B+1}}{1-u^{B+1}} \approx 14.5 \quad (52)$$

$$t_q = \frac{q}{\lambda'} \approx 30.57 \quad (53)$$

Substituting into our equations, $B = 56$,
 $\lambda' = 0.478$, and $u = 0.96$, we get:

$$q = \frac{u}{1-u} - \frac{(B+1)\rho^{B+1}}{1-u^{B+1}} \approx 17.8 \quad (54)$$

$$t_q = \frac{q}{\lambda'} \approx 37.3 \quad (55)$$

Yes, response time increased (since less load was shed).

30 – The $M/M/m/B$ Formulas 1 of 3

All $M/M/m/B$ systems are stable, since excess load is shed when buffer overflow occurs.

The parameters governing this system are:

- $\lambda = 0.48 \frac{\text{jobs}}{\text{sec}}$ is the arrival rate
- $s = 2\text{sec}$ is the mean service time
- $m = 2$ is the number of servers
- $B = 56$ is the number of buffers

The traffic intensity is $u = \frac{\lambda s}{m}$, since excess load is automatically shed, we focus on cases where $0 \leq u \leq 1$.

We need to estimate the net mean rate of arrival, λ' taking into account the dropped arrivals. This can be shown to be:

$$\lambda' = \sum_{n=0}^{n=B-1} \lambda p_n = \lambda(1 - p_B) \quad (56)$$

where p_n is the probability of n jobs being in the system.

Per processor (server) utilization is:

$$\rho = \frac{\lambda' s}{m} = u(1 - p_B) \quad (57)$$

31 – The $M/M/m/B$ Formulas 2 of 3

The general form of p_n are:

$$p_0 = \begin{cases} \frac{1-u}{1-u^{B+1}}, & u \neq 1 \\ \frac{1}{B+1}, & u = 1 \end{cases} \quad (58)$$

$$p_n = \begin{cases} \frac{1}{n!} (mu)^n p_0, & 0 \leq n < m \\ \frac{m^m u^n}{m!} p_0, & m \leq n \leq B \\ 0, & n > B \end{cases} \quad (59)$$

$$(60)$$

32 – The $M/M/m/B$ Formulas 3 of 3

This allows us to estimate:

$$q = \frac{u}{1-u} - \frac{(B+1)u^{B+1}}{1-u^{B+1}} \quad (61)$$

$$w = \frac{u}{1-u} - u \frac{1+Bu^B}{1-u^{B+1}} \quad (62)$$

$$t_q = \frac{q}{\lambda'} \quad (63)$$

$$t_w = t_q - s \quad (64)$$

The rate of packets lost is λp_B per unit time.

It can be shown that the mean percentage of dropped packets is:

$$\text{Percent Dropped} = 100 \left(1 - \frac{\lambda'}{\lambda}\right) \quad (65)$$

33 – Some Observations

- Systems with a shared queue but the same number of servers perform better than systems where separate queues are maintained.
- You do not need to double the processor speed to double the performance.
- Queueing models let us approximate a wide variety of systems.
- Increasing the number of buffers reduces the amount of load shed in $M/M/m/B$ systems which:
 - Increases Response time
 - Decreases drop rates

References

- [1] H. Deitel. *An Introduction to Operating Systems*. Addison Wesley, Reading, MA, 1989.
- [2] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, And Modeling*. Wiley, New York City, 1991.
- [3] W. Stallings. *Operating Systems: Second Edition*. Prentice Hall, Englewood Cliffs, NJ, 1995.