Analysis of Software Artifacts

System Performance II

Shu-Ngai Yeung (with edits by Jeannette Wing)
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

What we did last time

Learned some operational laws:

- They relate different system variables such as utilization, throughput and service demand.

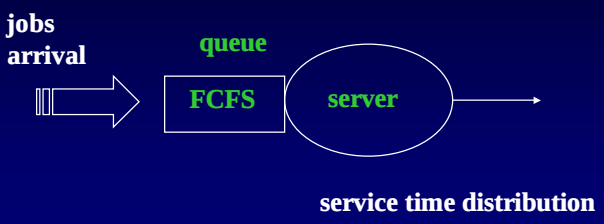
- They do not require distribution assumptions about arrival processes and service times.

What we did last lecture? (cont.)

Learned the performance bounds for closed computer systems:

- To improve performance bounds, need to improve the bottleneck device.

Single Server Open Systems



Single Queue Models

The single queue model is the simplest queueing model.

Such a model can be used to analyze individual resources in computer systems.

E.g., if all the jobs waiting for the CPU in a system are kept in one queue, then the CPU can be modeled using results that apply to single queues.

***Response time* and *queue length* are the major performance metrics.**

Lecture Outline

1. notation of M/M/1 queue

2. exponential distributions & Poisson processes

3. solution for M/M/1 queue

4. M/M/1 queue with finite buffers

5. single queue with multiple servers

6. comparison of system configurations

Queueing Notation

To specify a queueing system, we need to specify six parameters:

1. arrival process - interarrival time distribution

2. resource demands - service time distribution

3. number of servers

4. system capacity - finite buffers vs infinite buffers

5. population size

6. service disciplines - FCFS, LCFS, PS, etc.

The Arrival Pattern I

Each job is assumed to be drawn from the population or input source.

Queueing theory usually assumes an infinite population because the mathematical model is easier to analyze.

If the jobs arrive at time t_1, t_2, t_3,..., the interarrival times of jobs are defined as $t_{i+1} - t_i$, for i = 1,2,...

The Arrival Pattern II

We assume the arrivals of jobs follow a statistical pattern.

I.e., the interarrival times of jobs follow a common probability distribution with mean 1 / λ.

λ is called *arrival rate of jobs*.

The Service Distribution I

We assume the service demands of jobs are identically and independently distributed with a common distribution.

If the work demand of a job is S units, and the corresponding server has the capacity C units/sec,

then the ratio S / C is called the *service time*.

The Service Distribution II

E.g.,

- each program on the average has 200 instructions

- the CPU executes 4000 instructions per second

We say the service time follows a distribution with mean 0.05 seconds.

The Service Distribution III

Hereafter,

we only specify the service time distribution and

denote the service time by S.

The inverse of the mean service time is called the *service rate* μ.

Shorthand Notation

We use a shorthand notation to specify the six parameters :

A/S/m/B/K/SD

where

- **A is the interarrival time distribution**
- **S is the service time distribution**
- **m is the # of servers**
- **B is the # of buffers**
- **K is the population size**
- **SD is the service discipline**

Common Distribution Notation

Since the interarrival time and service time are random variables, we need notations to specify some of the common probability distributions:

M - exponential

D - deterministic (constant)

G - general (gamma, Weibull, Pareto, etc …)

Example

M/M/3/20/1500/FCFS

means:

- exponential interarrival and service time

- 3 servers and buffer size 20

- 1500 jobs that can be served

- service discipline is first come, first served

Example (cont.)

If the last 3 terms are unspecified, it means

- infinite buffers

- infinite population size

- FCFS discipline

The Exponential Distribution

The exponential distribution (Exp) is used to model the time between successive events.

The service time at devices are also modeled as exponentially distributed.

The probability density function of the Exp with rate μ is given as

$$f(x) = \mu \exp(-\mu x) \text{ for } x \geq 0.$$

Memoryless Property

The most important feature of the Exp is the memoryless property :

the residual service time of a job follows Exp(μ),

no matter how long the job has been serviced.

Poisson Processes

When the interarrival times of jobs are identically and independently distributed as Exp, the arrival process is a Poisson process.

It's called Poisson because the # of arrivals over a given time interval has a Poisson distribution.

A Poisson arrival process with rate λ means the interarrival times follow Exp with rate λ.

Properties of the Poisson Process I

Memoryless : the remaining time before the next arrival follows Exp(λ),

no matter when the last arrival took place.

Superposition : if we combine m Poisson input streams with rate λ_i into a single stream,

then we again have a Poisson process with the rate

$$\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_m.$$

Properties of the Poisson Process II

Decomposition : if we split a Poisson stream into k substreams such that the probability of a job going to the i-th substream is p_i,

then each substream is also Poisson with the rate $p_i\lambda$.

M/M/1

M/M/1 queue means a single-server system with Poisson arrivals and exponentially distributed service time.

M/M/1 queue can be used to model
- **single-processor system**
- **individual devices in computer systems**

This queueing model is analytically tractable under the exponential assumptions.

M/M/1 -Parameters

The input parameters of a M/M/1 queue include

- arrival rate λ

- service rate μ

The performance quantities are based on these two parameters and the traffic intensity $\rho = \lambda / \mu$.

Additional to the exponential assumptions, we also assume the system is stable, i.e. $\rho < 1$ or $\lambda < \mu$.

Performance Metrics

The common performance metrics include:

$E\{N_s\}$ = mean # of jobs in the system

$E\{N_q\}$ = mean # of jobs in the queue

$E\{T_s\}$ = mean time in the system for a job

$E\{T_q\}$ = mean waiting time in the queue

U = utilization of the server

Basic Relations

Since the mean # in the server is equal to the utilization, we have

$$E\{N_s\} = U + E\{N_q\}$$

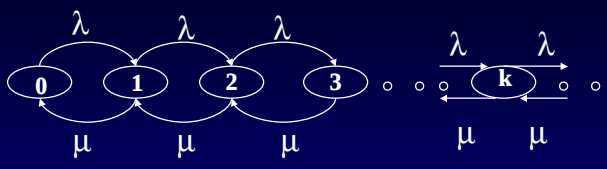
and

$$E\{T_s\} = E\{S\} + E\{T_q\}.$$

Note that E{S} = mean service time = 1 / μ.

If we know $E\{N_s\}$ or $E\{N_q\}$, by Little's Law, we can get $E\{T_q\}$ or $E\{T_s\}$.

Markov Processes



Due to the exponential assumptions, the # of jobs in the system can be modeled as a Markov process.

In probability theory, Markov processes have been well studied and the solution for computing P (# jobs in the system) exists.

Solution

Undergoing the black-box operation, we have the stationary distribution for job #:

$$P(n \text{ jobs in the system}) = (1-\rho)\,\rho^n.$$

Form this, we can deduce the mean # in the system:

$$E\{N_s\} = \rho / (1-\rho)$$

By Little's Law, the mean time in the system is:

$$E\{T_s\} = 1 / [\mu (1-\rho)]$$

System Utilization

Question : From the probability distribution of job # in the system, how do we compute the utilization?

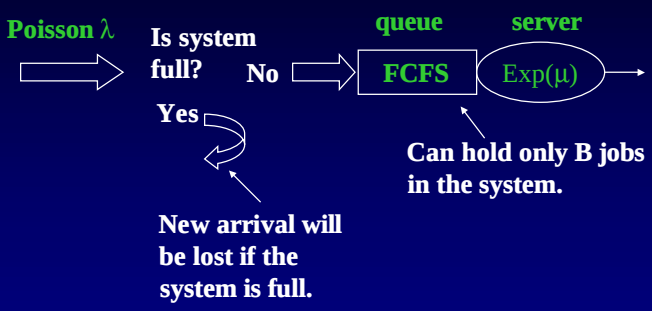
Answer : utilization = P(the server is busy)

$$= P(n>0)$$
$$= 1 - P(n=0)$$
$$= \rho.$$

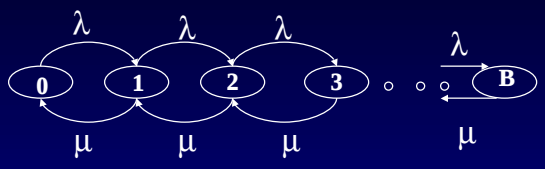
So, $E\{N_q\} = E\{N_s\} - U = \rho^2 / (1-\rho)$.

By Little's Law, $E\{T_q\} = \rho / [\mu (1-\rho)]$.

M/M/1/B finite buffer systems



Markov process for M/M/1/B



The Markov process is the same as the M/M/1 case but with a finite # of states.

Solution

After going through the black-box operation, the probability distribution for the # of jobs in the system is:

$$P(\text{n jobs in the system}) = \rho^n (1 - \rho) / (1 - \rho^{B+1})$$

for n = 0,1,2,…,B.

From the # jobs distribution, we can deduce $E\{N_s\}$ and $E\{T_s\}$.

But they are too complicated to show here.

Loss Probabilities

The more interesting quantity for the finite buffer queue is the **loss probability**, i.e., the fraction of jobs that are turned away.

Another nice property of **P**oisson processes is that the **A**rrivals **S**ee **T**ime **A**verages (**PASTA**).

i.e., P(an arrival sees n jobs in the system)
= P(n jobs in the system)

Capacity Planning

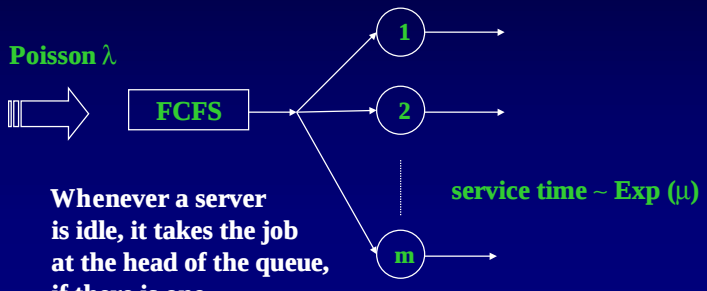
Loss probability
= P(an arrival sees B jobs in the system)
= P(B jobs in the system)
= $\rho^B (1 - \rho) / (1 - \rho^{B+1})$.

Question : For a system with traffic intensity ρ, how large a buffer size do we need in order to keep the loss probability less than 1%?

Answer : we need to solve for B numerically in
$\rho^B (1 - \rho) / (1 - \rho^{B+1}) < 0.01$

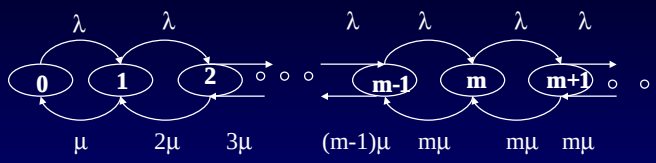
M/M/m Multiple Server Systems

M/M/m queue is used to model
- **multiprocessor systems**
- **devices that have several identical servers**



Whenever a server is idle, it takes the job at the head of the queue, if there is one.

Markov Process for M/M/m queue



The service rate is not constant due to the multiple servers.

When there are fewer than m jobs in the system, they are going at a lower rate.

The traffic intensity is defined as $\rho = \lambda / (m\mu)$.

Solution

After the black-box operation, the stationary distribution for job # in the system is:

P(n jobs in the system)

$= P_0 (m \rho)^n / n!$, if $n < m$

$= P_0 \rho^n m^m / m!$, if $n \geq m$

Probability of System Idle

P_0 is the probability of zero jobs in the system, which is found to be:

$$P_0 = \left[1 + \frac{(m\rho)^m}{m!(1-\rho)} + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} \right]^{-1}$$

Probability of Queueing

An interesting quantity in M/M/m queue is the probability of queueing

or the probability that an arriving job needs to wait before it accesses the server.

Again, by PASTA,
P_q = probability of queueing

= P(at least m jobs in the system)

= $(m\rho)^m P_0 / [m! (1- \rho)]$.

Performance Metrics

From the stationary probability distribution and Little's Law, we can deduce the following results:

$$E\{N_s\} = m\rho + \rho P_q / (1-\rho)$$

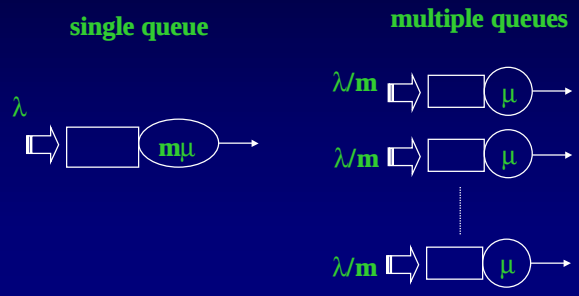
$$E\{N_q\} = \rho P_q / (1-\rho)$$

$$E\{T_s\} = 1/\mu + \rho P_q / [\lambda (1-\rho)]$$

$$E\{T_q\} = \rho P_q / [\lambda (1-\rho)]$$

Single queue vs Multiple queues I

Should we put all the resources together to form a single server or split the jobs into different queues?



Single queue vs Multiple queues II

The single queue is just an M/M/1 queue with traffic intensity $\rho = \lambda / (m\mu)$.

For the multiple queues, if we split the jobs randomly such that they have equal probability of going to each server,

then we still have Poisson arrivals processes.

Single queue vs Multiple queues III

Hence, the system can be modeled as m independent M/M/1 queues with traffic intensity $\rho = \lambda / (m\mu)$.

Since all the M/M/1 queues are identical, we only need to look at one of them.

Comparison

Multiple queue : $\rho = (\lambda/m)/\mu = \lambda / (m\,\mu)$

$$E\{T_s\} = m\,\rho / [\lambda\,(1-\rho)]$$

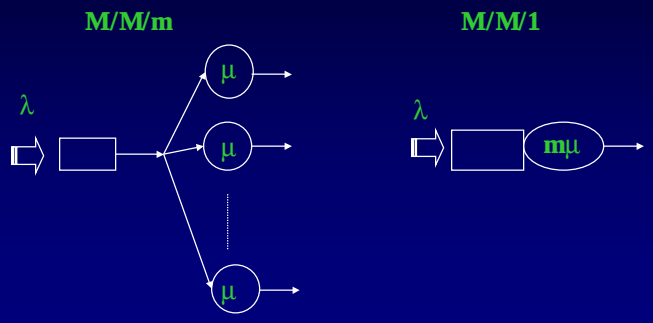
Single queue : $\rho = \lambda / (m\,\mu)$

$$E\{T_s\} = \rho / [\lambda\,(1-\rho)]$$

So the single queue is m times faster than the multiple queues.

One fast server or m slow servers?

Now we know that multiple queues is no good.
Should we use one fast server or m slow server?



Comparison I

Note that both systems have the same traffic intensity

$$\rho = \lambda / (m \mu).$$

We look at the ratio of the mean response times:

$$E\{T_s^{M/M/m}\} / E\{T_s^{M/M/1}\}$$

$$= 1/\mu + \rho P_q / [\lambda (1- \rho)] / \rho / [\lambda (1- \rho)]$$

$$= P_q + m(1- \rho).$$

Comparison II

When $\rho \approx 0$, then $P_q \approx 0$,

$$E\{T_s^{M/M/m}\} / E\{T_s^{M/M/1}\} = P_q + m(1- \rho) = m.$$

M/M/1 is m times faster than M/M/m.

When $\rho \approx 1$, then $P_q \approx 1$,

$$E\{T_s^{M/M/m}\} / E\{T_s^{M/M/1}\} = P_q + m(1- \rho) = 1.$$

M/M/m and M/M/1 have the same mean response time.

Comparison III

As the service rate of the M/M/m system is lower than M/M/1, each job has to spend more time at the server.

However, when the traffic intensity is high, each job needs to wait longer in the queue in the M/M/1 model compared to the M/M/m model.

From this result, we also know that putting all the jobs in a single queue is better than putting them in separate queues, when we have multiple servers.

Remarks

One of the advantage to use multiple queues is that we can provide guaranteed services to different classes of jobs.

The result holds only when the service time is exponentially distributed.

For general service time distribution, multiple queues may have a lower mean response time than the single queue.

References

Further reading on Markovian queueing models can be found in:

- L. Klenirock, Queueing Systems, Vol. I: Theory, John Wiley & Sons, 1975.

- R. Jain, The Art of Computer Systems Performance Analysis, John Wiley & Sons,1991.

- S.M. Ross, Introduction to Probability Models, 6-th Edition, Academic Press, 1997.

Homework

Exercises on

- applying operational laws to timesharing systems

- M/M/m queueing modeling