PART II – SPE Models

System Execution Models

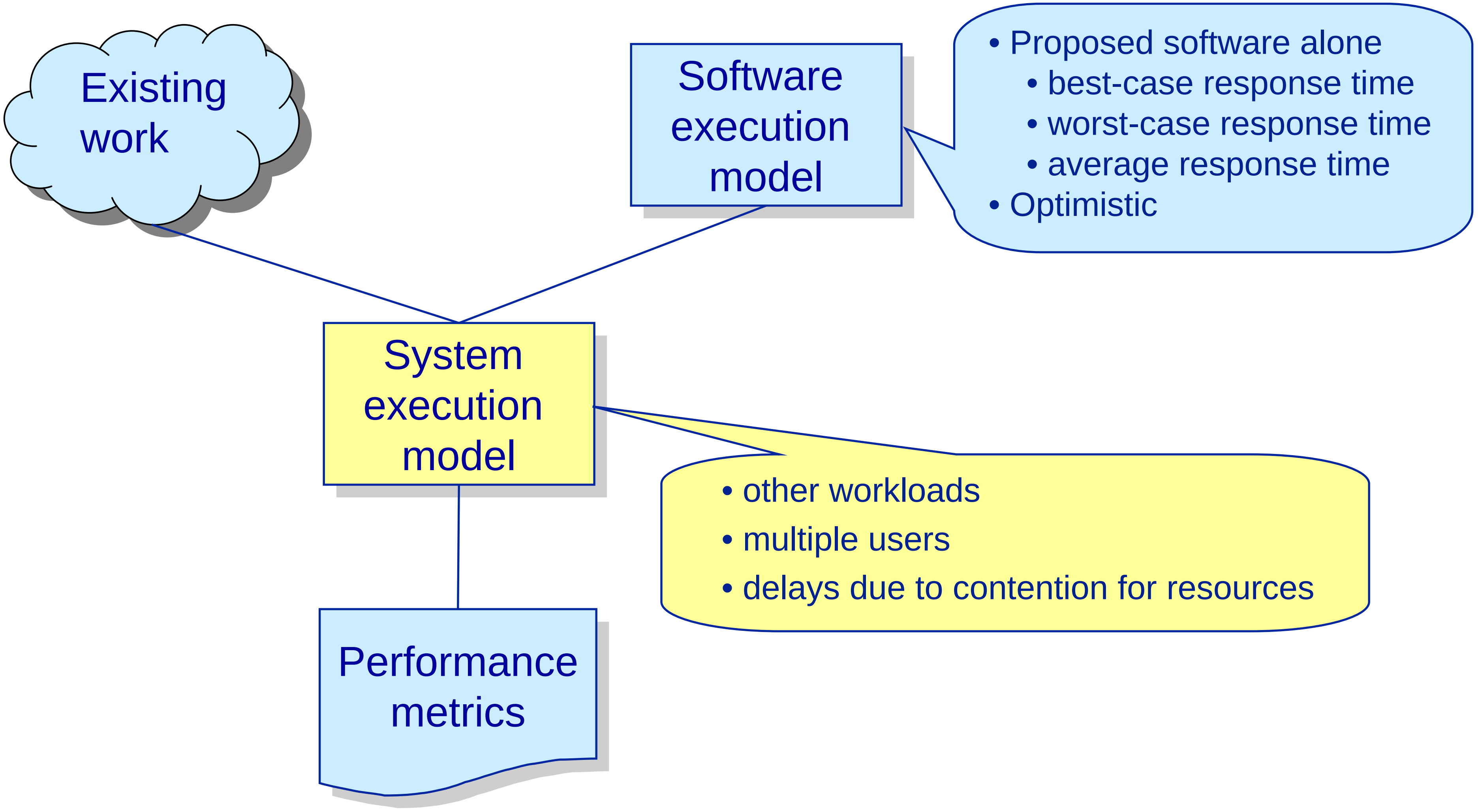
Basics

Outline

- Introduction
- System execution model basics
- Some basic performance results
- Different system execution models
  - M/M/1 queue (infinite population / infinite queue)
  - M/M/1/n queue (infinite population / finite queue)
  - M/M/m queue (infinite population / infinite queue / m servers)
  - Queuing networks
    - Open queuing networks
    - Closed queuing networks
- Case studies

West Virginia University

SPE Models



Software execution models

- Software execution models can identify serious performance problems at early design phases
  - If the predicted performance is unsatisfactory there is no need to build system execution model
  - If the predicted performance is satisfactory then build the system execution model
    - The absence of problems in software execution model does not mean that there are no problems
      - contention for system resources could cause problems; these problems may be corrected with
        - software design alternatives
        - hardware configuration alternatives

West Virginia University

System execution models

- Provide the following additional information
  - More precise metrics that account for resource contention
  - Identification of bottleneck resources
  - Sensitivity of performance metrics to variations in workload
  - Effect of new software on service level objectives of other software that executes on the same system
  - Scalability of hardware and software to meet future demands
  - Data on improving performance via workload changes, software changes, hardware upgrades, and various combinations of these

West Virginia University

Outline

- Introduction
- System execution model basics
- Some basic performance results
- Different system execution models
- Case study

West Virginia University

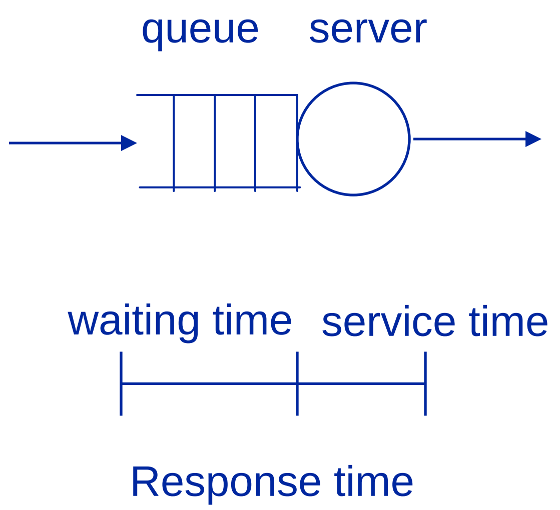
System execution model basics

- Sources of contention for resources
  - Multiple users of the system - several customers at different ATM machines may request transactions from the host bank computer at the same time
  - Other applications executing on the same hardware resources – in addition to the application that handles ATM transactions, at the bank host may execute applications that handle teller transactions, payroll system, etc.
  - Application may consists of several concurrent processes or threads – often case in embedded real-time systems

West Virginia University

System execution model – single resource

- Computer resources are represented as queues and servers
  - Server – component that provides some service to the software (CPU, disk, network)
  - Queue – jobs waiting for service



West Virginia University

System execution model – input parameters

- Job arrival
- Amount of service they need
- Time required for the server to process individual jobs
- Policy used for server to process individual jobs from the queue

West Virginia University

System execution model – Performance metrics

- Response time – average time that jobs spend at the server (includes service time and waiting time)
- Utilization – average percent of time that the server is busy providing service
- Throughput – average number of jobs that complete service per time unit (average rate at which jobs complete service)
- Queue length – average number of jobs at the server (include both those receiving service and those waiting)

Outline

- Introduction
- System execution model basics
- Some basic performance results
- Different system execution models
- Case study

West Virginia University

Some basic performance results

- Let
  - T – length of time we observe the system
  - A – number of request arrivals
  - C - number of request completions
  - B – length of time the resource was busy
- Arrival rate $\lambda=A/T$
- Throughput $X=C/T$
- Utilization $U = B/T$
- Service time per request $S=B/C$

Some basic performance results

- Simple and general relationships known as fundamental laws
  1. Utilization law
  2. Forced flow law
  3. Service demand law
  4. Liitle’s law
  5. Flow balance assumption

West Virginia University

Some basic performance results - Utilization law

- $U = B/T = C/T \cdot B/C$
- $U = X \cdot S$

  Utilization of a resource is equal to the product of the throughput of that resource and the average service time
- Example: A network segment transmits 1,000 packets/sec. Each packet has an average transmission time equal to 0.15 msec. What is the utilization of the LAN segment?

  $U = 1{,}000 \cdot 0.00015 = 0.15 = 15\%$

West Virginia University

Some basic performance results - Forced flow law

- Establishes relationship between individual resource view and entire system view
- Define visit count of a resource as a ratio of the number of completions at that resource to the number of system completions (i.e., average number of visits that a system level request makes to that resource) $V_i = C_i / C$

$$C_i = V_i \, C$$

$$C_i/T = V_i \, C/T$$

$$X_i = V_i \, X$$

West Virginia University

Some basic performance results - Forced flow law

- Example: Database transactions perform an average of 4.5 I/O operations on the database server. The database server was monitored during one hour and during this period 7,200 transactions were executed. What is the average throughput of the disk? If each disk I/O takes 20 msec on the average, what was the disk utilization?

Database throughput $X = 7{,}200 / 3{,}600 = 2$ trans/sec

Average number of visits to the disk $V_d = 4.5$

Disk throughput $X_d = V_d X = 4.5 \cdot 2 = 9$ trans/sec

Disk utilization $U_d = X_d S_d = 9 \cdot 0.02 = 0.18 = 18\%$

West Virginia University

Some basic performance results – Service demand law

- Define service demand as $D_i = V_i S_i$
- Combining the Utilization ($U_i = X_i \cdot S_i$) and Forced Flow ($X_i = V_i \cdot X$) laws we get

  $D_i = V_i \cdot S_i = (X_i / X) \cdot (U_i / X_i)$

  $D_i = U_i / X$
- Example: What is the service demand of the disk in the previous example?

  $D_d = U_d / X = 0.18/2 = 0.09$ sec

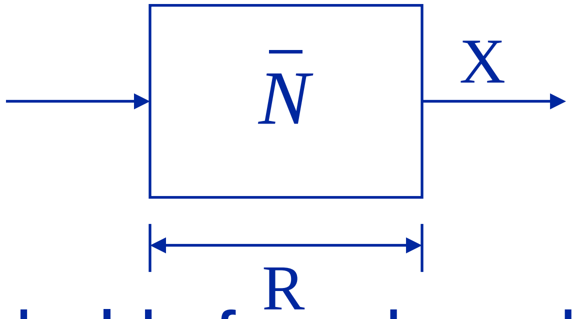
  (also $D_d = V_d S_d = 4.5 \cdot 0.02 = 0.09$ sec)

West Virginia University

Some basic performance results - Little's law

- Little's formula - mean number of jobs in the queuing system in steady state is equal to the product of the mean departure rate (throughput) and the mean response time

$$\bar{N} = X \cdot R$$



- Note: Little's formula holds for a broad variety of queuing systems - the box could contain a simple device such as disk, or complex queuing system such as an entire intranet provided that it does not create or destroys customers

West Virginia University

Some basic performance results - Little's law

- R=W+S

  (average response time = average waiting time + average service time)

- Appling Little's formula we get
  - Average number of jobs at the waiting queue $\bar{N}_w = X \cdot W$
  - Average number of jobs receiving service $\bar{N}_s = X \cdot S$

    (in case of a single resource queue this is a number between 0 and 1 that can be interpreted as the fraction of the time that the resource is busy, i.e., the utilization of the resource)

Some basic performance results - Flow balance assumption

- Assume that systems satisfy the flow balance property, namely, that the number of arrivals equals the number of completions, and thus the arrival rate equals the throughput

$A=C$, therefore $\lambda = X$

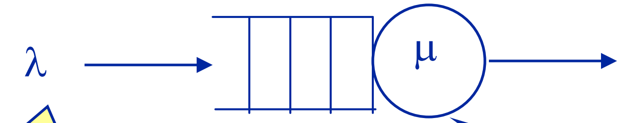
Outline

- Introduction
- System execution model basics
- Some basic performance results
- Different system execution models
  - M/M/1 queue (infinite population / infinite queue)
- Case study

West Virginia University

Simple server model – Infinite population / infinite queue

M/M/1 queue



- Customer arrival – Poisson process with rate λ (customer inter-arrival times are exponentially distributed with mean 1/λ)
- Single class (homogeneous workload)

Policy – first come first served (FCFS)

Service times are independent identically distributed random variables - exponential distribution with mean 1/μ

Requests arrive at rate of λ request / sec,
queue for service,
get served at a rate of μ request / sec,
and depart

We want to compute:
- probability p_k that there are k jobs in the system
- average number of jobs in the system
- server's utilization and throughput
- average response time of a job

West Virginia University

Simple server model – M/M/1

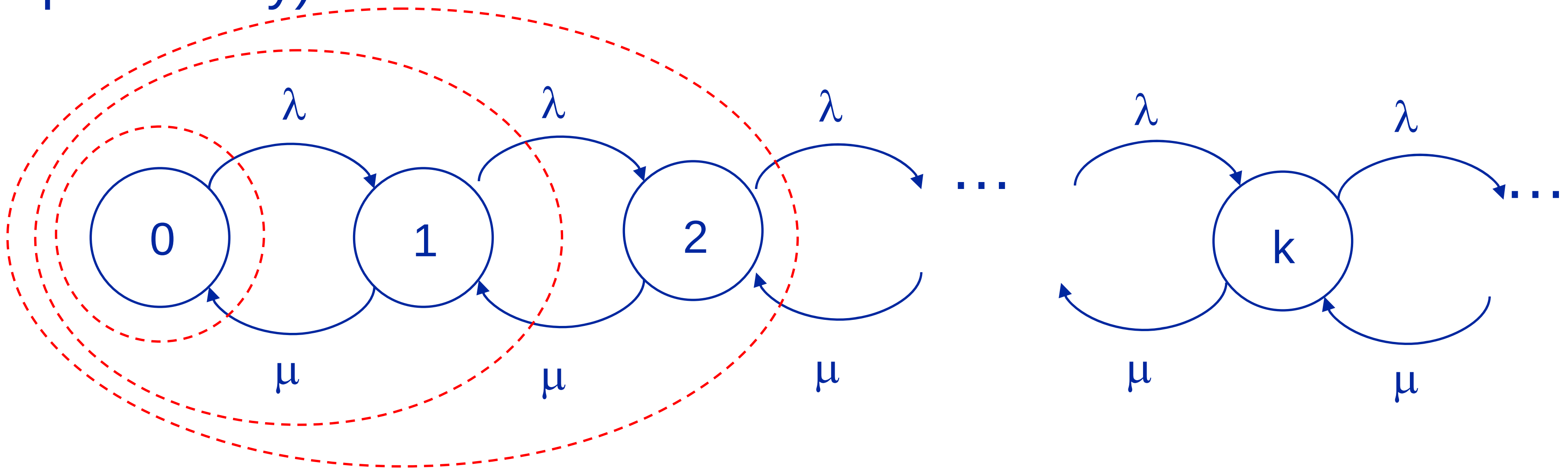
- State – number of jobs present in the server (waiting or receiving service)

$$\begin{tikzcd}
0 \arrow[r, bend left, "\lambda"] & 1 \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & 2 \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & \cdots \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & k \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & \cdots \arrow[l, bend left, "\mu"]
\end{tikzcd}$$

Continuous time Markov chain (CTMC)

M/M/1 queue – probability p_k

- Probability p_k that there are k jobs in the server (steady-state probability)



flow-in = flow-out

$\mu\, p_1 = \lambda\, p_0$

$\mu\, p_2 = \lambda\, p_1$

$\vdots$

$\mu\, p_k = \lambda\, p_{k-1}$

$$p_k = \frac{\lambda}{\mu} p_{k-1} = \frac{\lambda}{\mu}\left(\frac{\lambda}{\mu} p_{k-2}\right) = \ldots = \left(\frac{\lambda}{\mu}\right)^k p_0 \qquad for\ k = 1,2,\ldots$$

M/M/1 queue – probability p_k

$$p_0 + p_1 + \ldots + p_k + \ldots = \sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k p_0 = 1$$

$$p_0 = \frac{1}{\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k} = 1 - \frac{\lambda}{\mu}$$

$$p_k = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^k, \quad for\ k \geq 0$$

M/M/1 queue – probability p_k

- Traffic intensity

  $\rho = \lambda/\mu =$ *mean service time / mean interarrival time*

- If $\rho = \lambda/\mu < 1$ (arrival rate is smaller than service rate)
  - Probability that there are k jobs in the server is

    $$p_k = (1-\rho)\rho^k, \quad for\ k \geq 0$$

  - Probability that the server is idle (0 jobs)

    $$p_0 = 1 - \rho$$

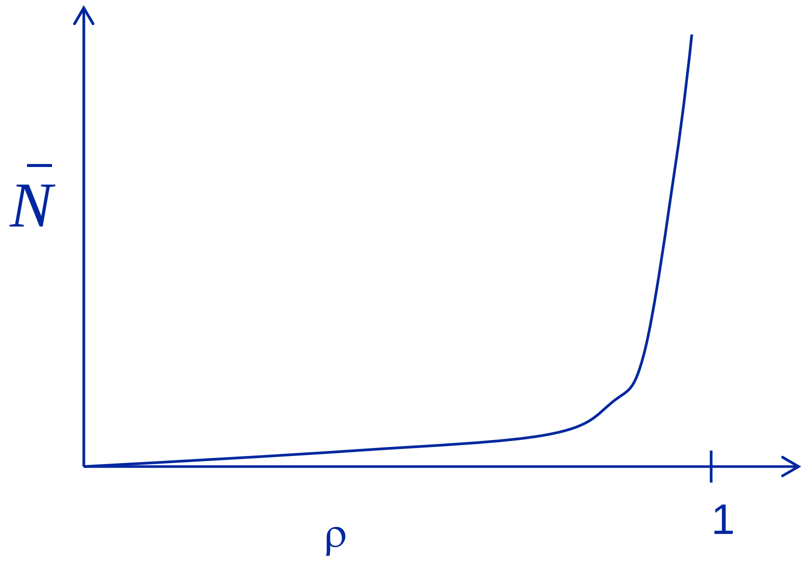
- If $\rho \geq 1$ the system is unstable – number of customers in the system tends to increase without a bound

West Virginia University

M/M/1 queue - average number of jobs in the server

Average number of jobs in the server

$$\bar{N} = E[N] = \sum_{k=0}^{\infty} k\, p_k = \sum_{k=0}^{\infty} k\,(1-\rho)\rho^k = (1-\rho)\sum_{k=0}^{\infty} k\,\rho^k = \frac{\rho}{1-\rho}$$



West Virginia University

M/M/1 queue – server utilization and throughput

- Utilization – proportion of time the server is busy

  $U = 1 - p_0 = \rho$

- Throughput - average rate at which jobs complete service

  $X = \mu \bullet U + 0 \bullet (1-U) = \mu \bullet \lambda/\mu = \lambda$

  (requests are not lost, the average arrival rate is equal to the average departure rate)

West Virginia University

M/M/1 queue – average response time

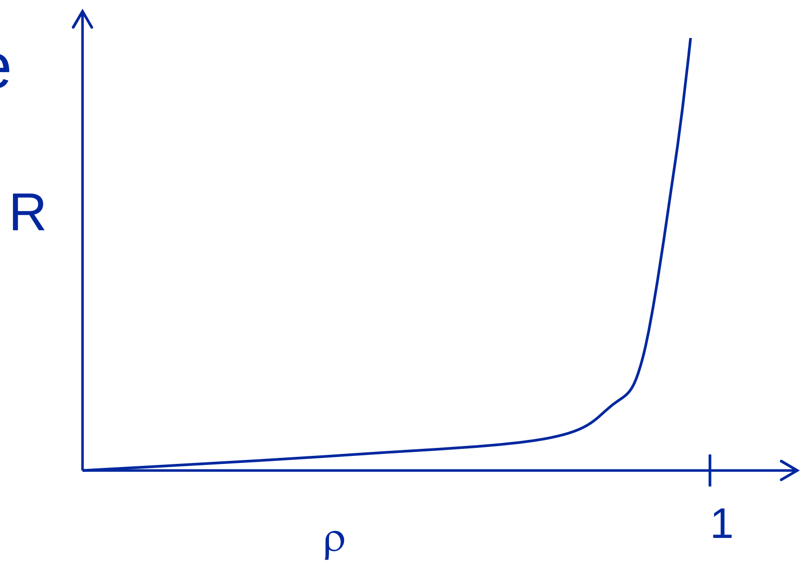
- Appling Little's formula to M/M/1 queue

$$R = \frac{\bar{N}}{X} = \lambda^{-1}\frac{\rho}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{\textit{average service time}}{\textit{probability that the server is idle}}$$

where $s = 1/\mu$ is the average service time

When the probability that server is idle is close to 1 (utilization U=ρ close to 0) the average response time is close to the average service time

When the probability that server is idle is close to 0 (utilization U=ρ close to 1) the average response time goes to infinity (delays build rapidly due to congestion)



M/M/1 queue – other measures

- Average waiting time in the queue

$$W = R - S = \frac{1}{\mu(1-\rho)} - \frac{1}{\mu} = \frac{1}{\mu}\frac{\rho}{1-\rho} = S\,\bar{N}$$

- Average number of jobs waiting in the queue (excluding those in service) – apply Little's formula

$$\bar{N}^w = X \cdot W = \frac{\rho^2}{1-\rho}$$

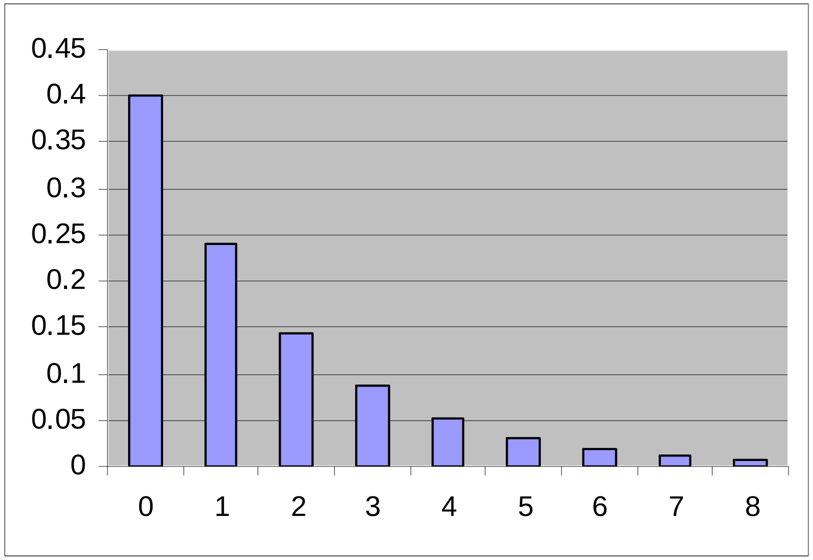
M/M/1 queue - Example

- Requests arrive at the database server at a rate of 30 requests / sec. Each request takes 0.02 sec on the average to be processed. What is the fraction of time that k (k=0,1,…) requests are found in the database server? What is the average response time of the server? What is the average response time if the server is replaced with a server twice as fast? What would be the response time if the arrival rate doubles when the server becomes twice as fast?
- λ = 30 request / sec
- S = 0.02 sec
- μ = 1 / 0.02 = 50 request /sec

West Virginia University

M/M/1 queue - Example

- Fraction of time the server is idle $p_0=1-(\lambda/\mu)=1-0.6=40\%$
- $p_k= (1-\lambda/\mu)(\lambda/\mu)^k = 0.4 \cdot 0.6^k$



- Utilization $U = 1 - p_0 = \lambda/\mu = 0.6 = 60\%$
- Throughput $X = \lambda =$ 30 request /sec
- Average number of requests at the server $\bar{N} = \frac{\rho}{1-\rho} = \frac{0.6}{1 - 0.6} = 1.5$

West Virginia University

M/M/1 queue - Example

- Average response time $R = \frac{\bar{N}}{X} = \frac{1.5}{30} = 0.05$ sec

- Server twice as fast μ = 100 request /sec; U = ρ = λ/μ = 0.3

$$R = \frac{\bar{N}}{X} = \lambda^{-1} \frac{\rho}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{1/100}{1\text{-}0.3} = 0.014 \text{ sec}$$

Using twice as fast server reduces the response time to about 28% of its original value

- Server twice as fast μ = 100 request /sec

Double arrival rate λ = 60 request / sec; U=ρ= λ/μ = 0.6

$$R = \frac{\bar{N}}{X} = \lambda^{-1} \frac{\rho}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{1/100}{1\text{-}0.6} = 0.025 \text{ sec}$$

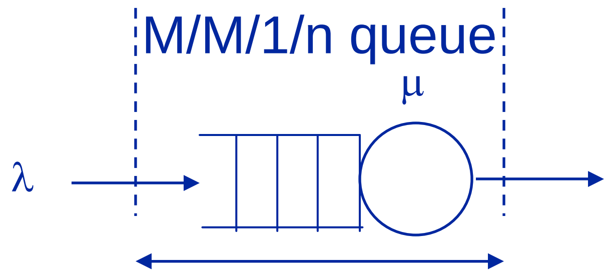
West Virginia University

Outline

- Introduction
- System execution model basics
- Different system execution models
  - M/M/1 queue (infinite population / infinite queue)
  - M/M/1/n queue (infinite population / finite queue)
- Case study

West Virginia University

Simple server model – Infinite population / finite queue



Limited buffer space – at most **n** jobs can be in the system at a time

Requests arrive at rate of λ request / sec ,
if the system is full the job is rejected,
otherwise the job enters the queue for service, gets served at a rate of μ request / sec, and departs

We want to compute:

- probability p_k that there are k jobs in the server
- average number of jobs in the server
- server’s utilization and throughput
- average response time of a job

West Virginia University

Simple server model – M/M/1/n

- State – number of jobs present in the server (waiting or receiving service)

$$\begin{tikzcd}
0 \arrow[r, bend left, "\lambda"] & 1 \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & \cdots \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & k \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & \cdots \arrow[l, bend left, "\mu"] \arrow[r, bend left, "\lambda"] & n \arrow[l, bend left, "\mu"]
\end{tikzcd}$$

Continuous time Markov chain (CTMC)

M/M/1/n queue – probability p_k

- Probability p_k that there are k jobs in the server (steady-state probability)
- Write “flow-in = flow-out” equations

$$p_k = \left(\frac{\lambda}{\mu}\right)^k p_0 \qquad for\ k = 1, 2, \ldots, n$$

M/M/1/n queue – probability p_k

$$p_0 + p_1 + \ldots + p_n = \sum_{k=0}^{n} p_k = \sum_{k=0}^{n} \left(\frac{\lambda}{\mu}\right)^k p_0 = \left[\frac{1-(\lambda/\mu)^{n+1}}{1-\lambda/\mu}\right] p_0 = 1$$

$$p_0 = \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{n+1}}$$

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{n+1}}, \quad \mathit{for}\ k \geq 0$$

West Virginia University

M/M/1/n queue - average number of jobs in the server

- Average number of jobs in the server

$$\bar{N} = E[N] = \sum_{k=0}^{n} k \, p_k = p_0 \sum_{k=0}^{n} k \, (\lambda / \mu)^k$$

Using the fact that $\sum_{k=0}^{n} k \, a^k = [n a^{n+2} - (n+1) a^{n+1} + a] / (1-a)^2$

and the equation for p_0 we get

$$\bar{N} = E[N] = \frac{(\lambda/\mu)[n(\lambda/\mu)^{n+1} - (n+1)(\lambda/\mu)^n + 1]}{[1-(\lambda/\mu)^{n+1}](1-\lambda/\mu)}$$

M/M/1/n queue – server utilization and and throughput

- Utilization – proportion of time the server is busy

$$U = 1 - p_0 = \frac{(\lambda/\mu)[1-(\lambda/\mu)^n]}{1-(\lambda/\mu)^{n+1}}$$

- Throughput - average rate at which jobs complete service

$$X = \mu \cdot U + 0 \cdot (1-U) = \frac{\lambda[1-(\lambda/\mu)^n]}{1-(\lambda/\mu)^{n+1}}$$

M/M/1/n queue

- Average response time - applying Little's formula

$$R = \frac{\bar{N}}{X} = \frac{S\left[n(\lambda/\mu)^{n+1} - (n+1)(\lambda/\mu)^{n} + 1\right]}{[1-(\lambda/\mu)^{n}]\,(1-\lambda/\mu)}$$

where $S = 1/\mu$ is the average service time

M/M/1/n queue – Example

- Consider the same database server of Slide 31, but now assume that at most four requests can be queued at the server (including requests being processed).
- λ = 30 request / sec
- S = 0.02 sec
- μ = 1 / 0.02 = 50 request /sec
- n=4
- $p_0 = 0.43 = 43\%$
- $p_k = 0.43 \cdot 0.6^k$
- Fraction of request that are lost because the queue is full $p_n = 0.43 \cdot 0.6^n = 0.0557 = 5.57\%$

M/M/1/n queue – Example

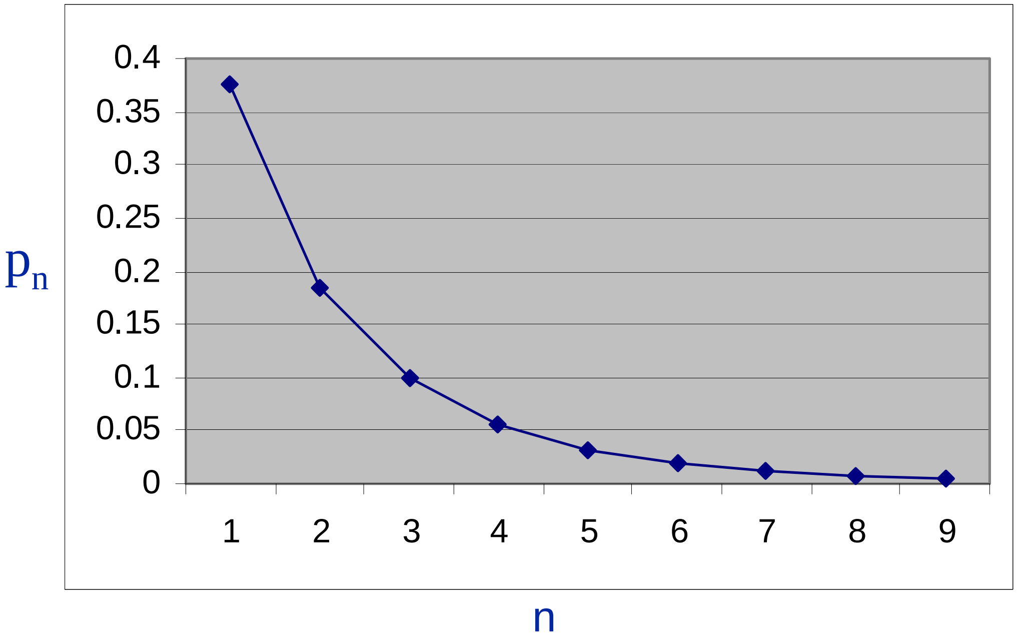
- What should be the minimum value of the buffer size (maximum number of accepted requests) so that less than 1% of the requests are rejected?

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{n+1}} = 0.4 \cdot 0.6^n / (1-0.6^{n+1}) < 0.01$$

$n > -\ln(40.6)/\ln(0.6)$

$n > 7.25055$

$n \geq 8$



West Virginia University

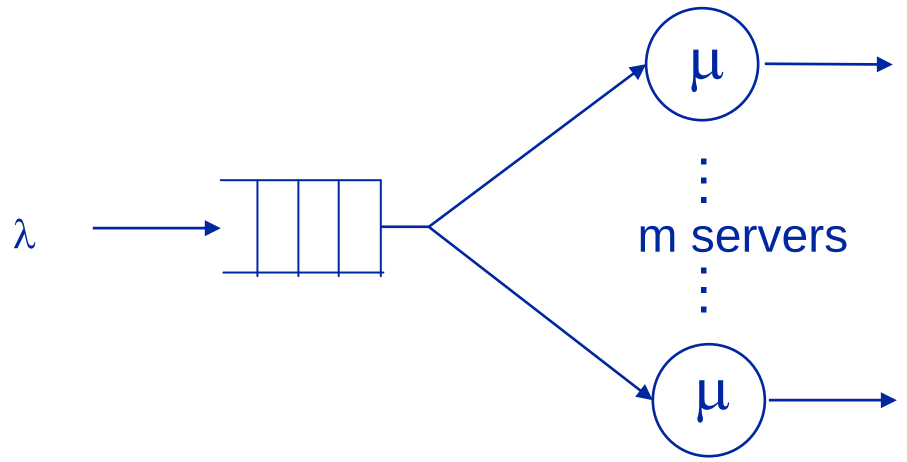
Outline

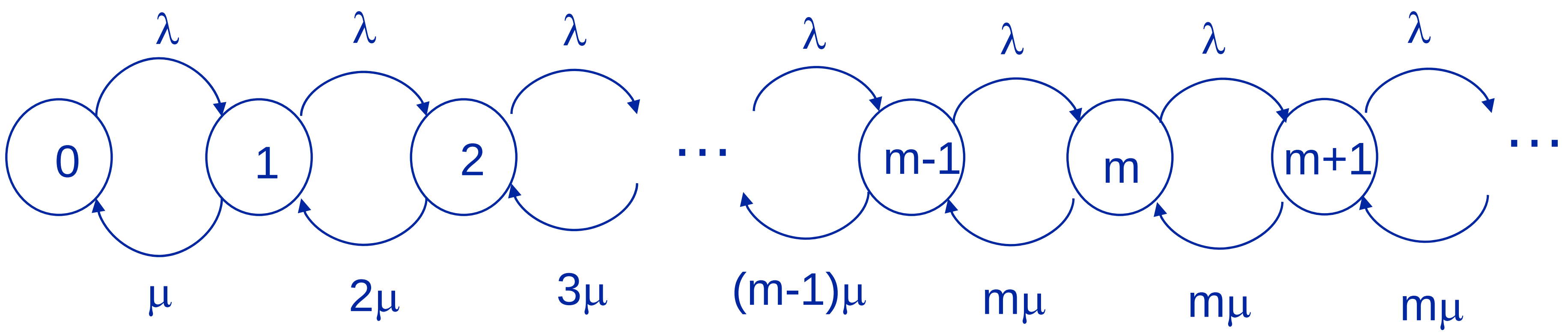
- Introduction
- System execution model basics
- Different system execution models
  - M/M/1 queue (infinite population / infinite queue)
  - M/M/1/n queue (infinite population / finite queue)
  - M/M/m queue (infinite population / infinite queue / m servers)
- Case study

West Virginia University

Simple server model – M/M/m

M/M/m queue

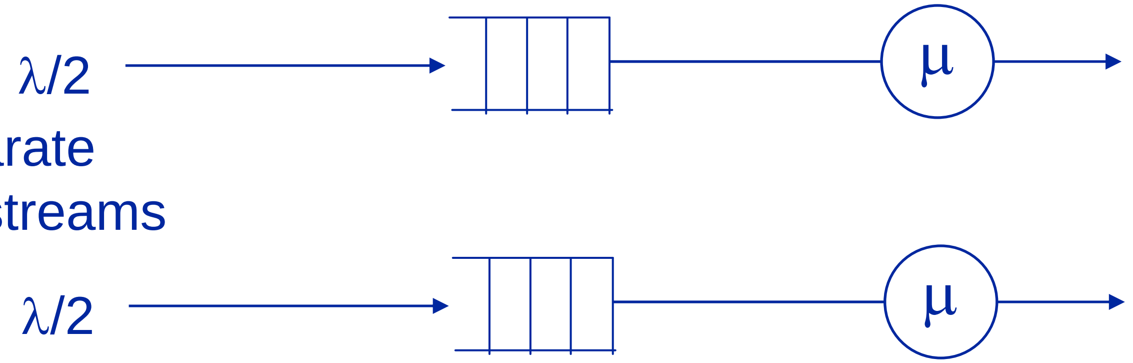


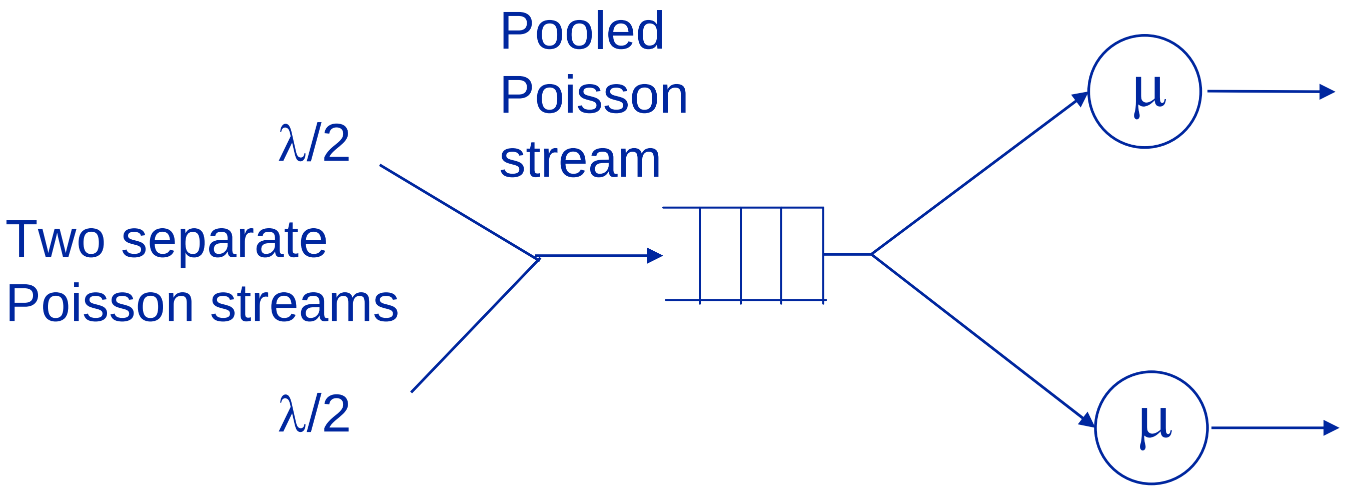


West Virginia University

Example

Compare the two different queuing schemes based on the response times.





West Virginia University

Example

- First scheme – two independent M/M/1 queues with arrival rates $\lambda/2$, service rates μ; and $\rho = \lambda/(2\mu)$

$$R_s = \frac{1/\mu}{1-\rho} = \frac{2}{2\mu - \lambda}$$

- Second scheme – M/M/2 queue with $\rho = \lambda/(2\mu)$

$$R_c = \frac{1/\mu}{1-\rho^2} = \frac{4\mu}{4\mu^2 - \lambda^2}$$

- Compare $R_s = \frac{2}{2\mu - \lambda} = \frac{4\mu + 2\lambda}{4\mu^2 - \lambda^2} > R_c$

Common-queue scheme is better than a separate-queue scheme