PART II – SPE Models

System Execution Models:
Queuing Networks

West Virginia University

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

Queuing networks (QN)

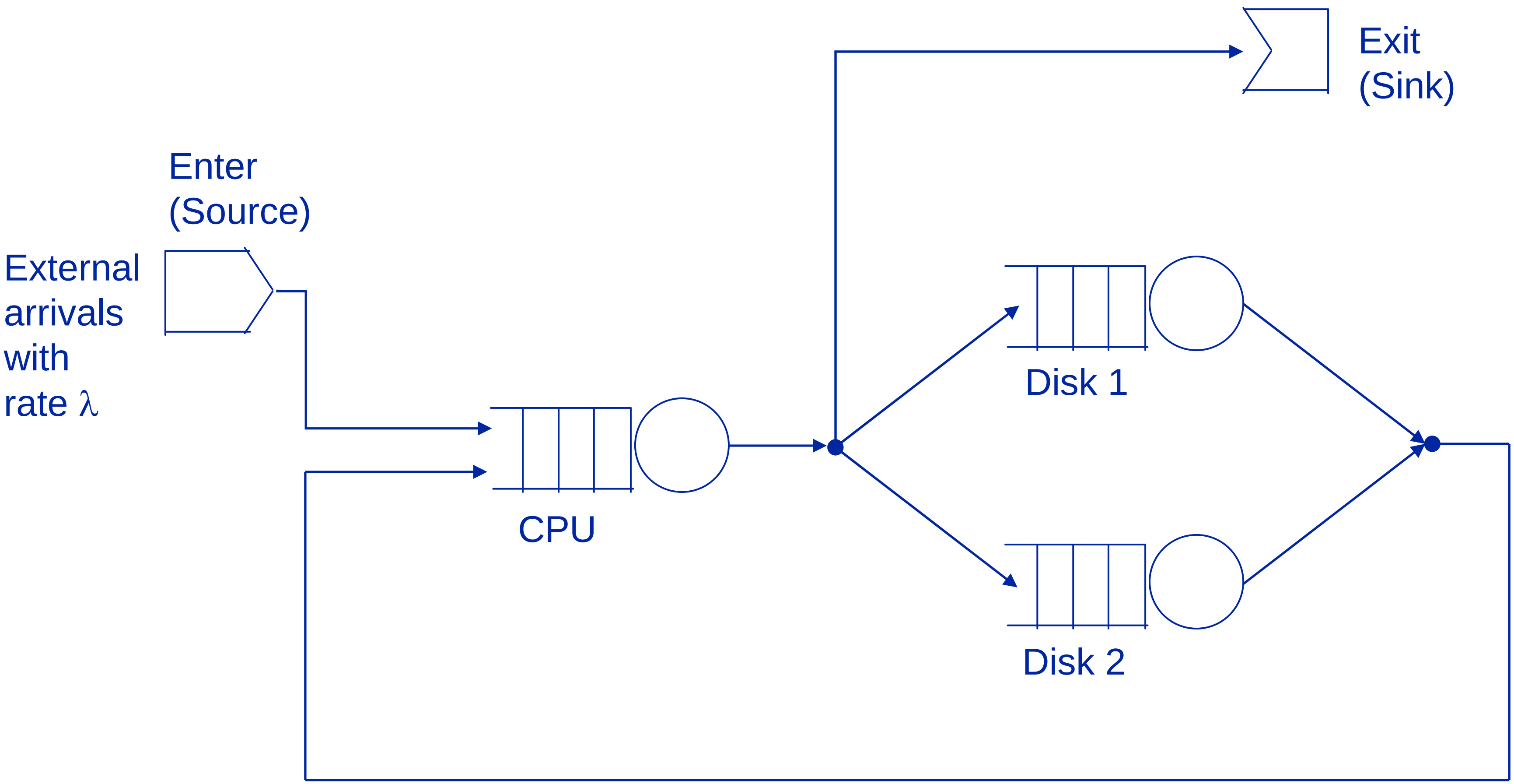
- Open queuing network – number of requests in the QN is unbounded; requests arrive, go through various resources, and leave the system
- Closed queuing network - number of requests in the QN is fixed

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)
  - Queuing networks
    - Open queuing networks
    - Closed queuing networks
- Case study

West Virginia University

Open queuing network



West Virginia University

Input parameters

- Transaction workload; population varies over time; requests that have completed service leave the model.
  - λ - arrival rate of requests to the QN
- K - number of queues (service centers, devices)
- For each device i
  - V_i (average number of visits to device i by a request) and S_i (average service time of a request at device i per visit)

OR

  - $D_i = V_i \cdot S_i$ (service demand)

West Virginia University

Solution to open QN

- System throughput X. In the case of open system with operational equilibrium, the average throughput is the same as the average arrival rate λ (Flow balance property) $X = \lambda$

- Device throughput $X_i = V_i\, X$ (Forced flow law). If only service demand D_i is known the average device throughput can not be estimated

- Device utilization $U_i = X_i \cdot S_i$ (Utilization low)

$$U_i = X_i \cdot S_i = V_i\, X \cdot S_i = V_i\, \lambda\, S_i = \lambda \cdot V_i\, S_i = \lambda\, D_i$$

Forced Flow low

Flow Balance property

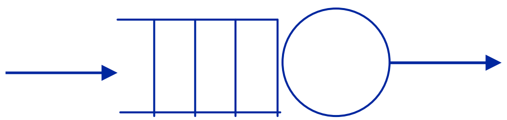
West Virginia University

Solution to open QN

- Response time of a request at a queueing device *i* is the total time spent at the device for one visit (both queueing and receiving service): $R_i = S_i + W_i$

  Arrival Theorem for open queue: average number of requests in the queue i as seen by an arriving request is equal to the average number of requests $\overline{N}_i$

  $$R_i = S_i + W_i = S_i + \overline{N}_i \cdot S_i$$



  from Little's law we have $\overline{N}_i = X_i R_i$

  $$R_i = S_i + X_i R_i \cdot S_i$$

  from Utilization law we have $U_i = X_i S_i$

  $$R_i = S_i + R_i \cdot U_i$$

  It follows that $R_i = \dfrac{S_i}{1 - U_i}$

Solution to open QN

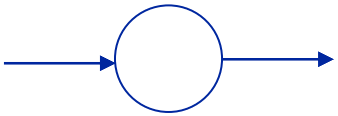
- Residence time of a request at a queueing device *i* (over all visits to device *i*):

$$R'_i = V_i R_i = \frac{V_i S_i}{1 - U_i} = \frac{D_i}{1 - U_i}$$

West Virginia University

Solution to open QN

- Response time of a request at a delay device *i* does not have queueing component; It is simply a service time $R_i = S_i$



- Residence time of a request at a delay device i (over all visits to device i)

$$R_i' = V_i R_i = V_i S_i = D_i$$

- System response time – sum of the residence times over all devices

$$R = \sum_{i=1}^{K} R'_i$$

West Virginia University

Solution to open QN

- Average number of request at device *i*:

  Utilization law

  - Queueing device $\bar{N}_i = X_i R_i = \frac{X_i S_i}{1-U_i} = \frac{U_i}{1-U_i}$

  Little's law

  - Delay device $\bar{N}_i = X_i R_i = X_i S_i = U_i$

- Average number of request in the system:

$$\bar{N} = X \cdot R = \sum_{i=1}^{K} \bar{N}_i$$

Open queuing networks - Summary

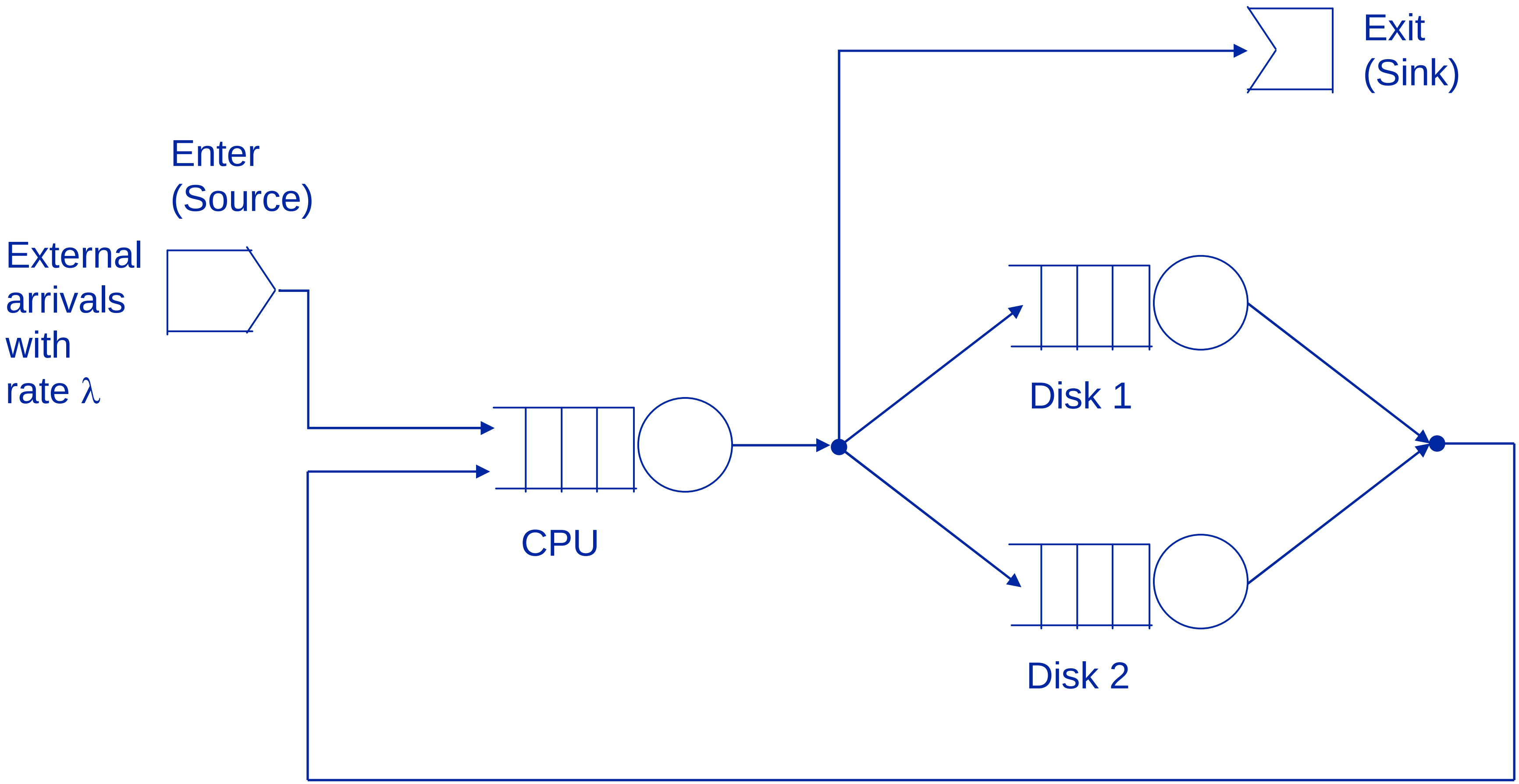
West Virginia University

- System throughput $X = \lambda$
- Device throughput $X_i = V_i X$
- Device utilization $U_i = \lambda D_i$
- Residence time at device $R_i' = \begin{cases} D_i & \text{delay device} \\ \dfrac{D_i}{1-U_i} & \text{queueing device} \end{cases}$
- System response time $R = \sum_{i=1}^{K} R'_i$
- Queue length at device $\bar{N}_i = \begin{cases} U_i & \text{delay device} \\ \dfrac{U_i}{1-U_i} & \text{queueing device} \end{cases}$
- Average number in system $\bar{N} = \sum_{i=1}^{K} \bar{N}_i$

Open QN - bounds

- Processing capacity , that is, maximum theoretical value of the arrival rate λ
  - for all resources $U_i = \lambda D_i$, that is, $\lambda = \frac{U_i}{D_i}$
  - because the utilization of any resource cannot exceed 100% it follows that $\lambda \leq \frac{1}{D_i}$
  - maximum value of λ is limited by the resource with the highest value of the service demand, called the bottleneck resource $\lambda \leq \frac{1}{\max\limits_{1 \leq i \leq K} D_i}$

Example: Open queuing network



West Virginia University

Example: Open queuing network

A DB server has one CPU and two disks and receives requests at a rate of 1,080 request per hour. Each request needs 605 msec of CPU and performs seven I/Os on disk 1 and five I/Os on disk 2 on average. Each I/O takes an average of 300 msec on disk 1 and 270 msec on disk 2. What are the average response time per request, average throughput of the DB server, utilization of the CPU and disks, and the average number of requests at the server? What is the maximum theoretical arrival rate of requests sustained by this DB server?

λ=1,080/3,600 = 0.3 request/sec

D_{CPU}=0.605 sec

V_1=7; S_{disk1}=0.3 sec $D_{disk1} = V_1 S_{disk1} = 7 \cdot 0.3 = 2.1$ sec

V_2=5; S_{disk2}=0.27 sec $D_{disk2} = V_2 S_{disk2} = 5 \cdot 0.27 = 1.35$ sec

West Virginia University

Example: Open queuing network

- Average throughput of the DB server is equal to the average arrival rate (Flow balance law)

  $X = \lambda = 0.3$ request /sec

- Device throughput (Forced flow low)

  X_{CPU} cannot be estimated

  $X_{disk1} = V_1 X = 7 \cdot 0.3 = 2.1$ request /sec

  $X_{disk2} = V_2 X = 5 \cdot 0.3 = 1.5$ request /sec

- Utilization of the CPU and disks (Service Demand law)

  $U_{CPU} = D_{CPU} X = D_{CPU} \lambda = 0.605 \cdot 0.3 = 0.1815 = 18.15\%$

  $U_{disk1} = D_{disk1} X = D_{disk1} \lambda = 2.1 \cdot 0.3 = 0.63 = 63\%$

  $U_{disk2} = D_{disk2} X = D_{disk2} \lambda = 1.35 \cdot 0.3 = 0.405 = 40.5\%$

West Virginia University

Example: Open queuing network

- Residence times of a request at device

$R'_{CPU} = D_{CPU}/(1 - U_{CPU}) = 0.605/(1 - 0.1815) = 0.740$ sec

$R'_{disk1} = D_{disk1}/(1 - U_{disk1}) = 2.1/(1-0.63) = 5.676$ sec

$R'_{disk2} = D_{disk2}/(1 - U_{disk2}) = 1.35/(1-0.405) = 2.269$ sec

Total response time

$R = R'_{CPU} + R'_{disk1} + R'_{disk2} = 0.740 + 5.676 + 2.269 = 8.685$ sec

Example: Open queuing network

- Average number of requests at each device

$$\bar{N}_{CPU} = U_{CPU}/(1- U_{CPU}) = 0.1815/(1-0.1815) = 0.222$$

$$\bar{N}_{disk1} = U_{disk1}/(1- U_{disk1}) = 0.63/(1-0.63) = 1.703$$

$$\bar{N}_{disk2} = U_{disk2}/(1- U_{disk2}) = 0.405/(1-0.405) = 0.681$$

Total number of request at DB server

$$\bar{N} = \bar{N}_{CPU} + \bar{N}_{disk1} + \bar{N}_{disk2} = 0.222 + 1.703 + 0.681 = 2.606 \text{ request}$$

West Virginia University

Example: Open queuing network

- Maximum theoretical arrival rate of requests sustained by this DB server

$\lambda = 1/\ max\{0.605, 2.1, 1.35\} = 0.476$ request / sec

Disk 1 is the bottleneck – the resource with the highest value of service demand

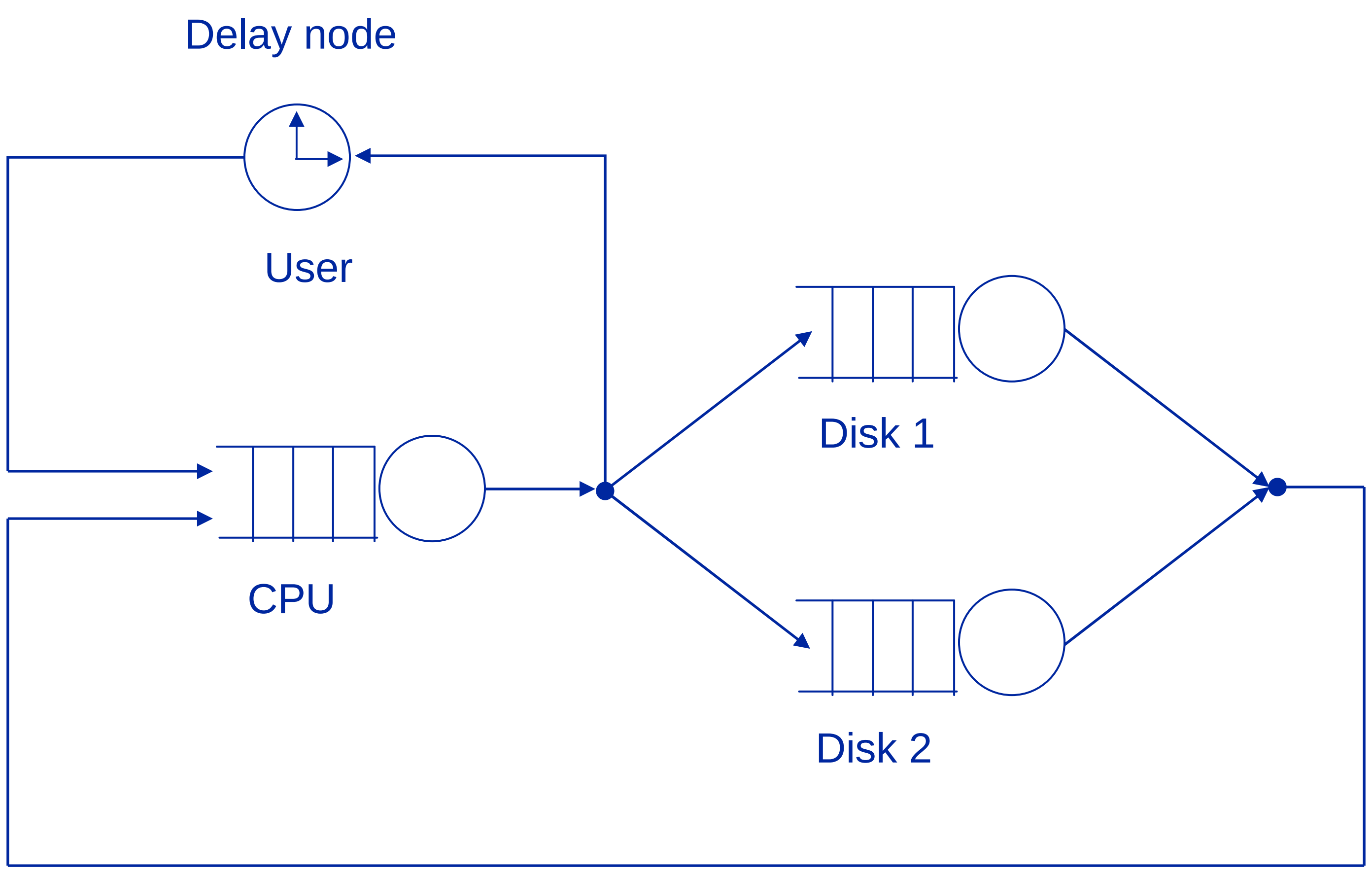
Outline

- 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)
  - Queuing networks
    - Open queuing networks
    - Closed queuing networks
- Case study

Closed queuing networks

- Finite fix number of request in the system; No external arrivals or departures
- Examples:
  - maximum degree of multiprogramming under heavy load
  - client/server system with a known number of clients sending request to a server

Closed queuing network



Input parameters

- Terminal workload; population is fixed over time;
  - N – population (number of requests) in the QN
  - Z – think time (average delay between the receipt of response and the submission of the next request)

OR

- Batch workload; population is fixed over time;
  - N – population (number of requests) in the QN; $(Z = 0)$
- K - number of queues (service centers, devices)
- For each device i
  - V_i (average number of visits to device i by a request) and S_i (average service time of a request at device i per visit)

  OR

  - $D_i = V_i \cdot S_i$ (service demand)

West Virginia University

Solution to closed QN

- Solving closed queuing network models is more complex because the throughput depends on the response time
- Mean Value Analysis (MVA) - solution technique for closed queuing networks with only infinite queue and load-independent service time for each node (device)
- For closed queuing networks variables are functions of the number of requests N in the system
- MVA is based on recursively using three equations
  - Residence time equation
  - Throughput equation
  - Queue length equation

MVA - Residence time equation

West Virginia University

- Residence time equation

Response time per visit to device i:

$$R_i(N) = S_i + W_i(N)$$

Denote by $A_i(N)$ the average number of requests found in the device i by an arriving request

$$R_i(N) = S_i + A_i(N) \cdot S_i = S_i[1 + A_i(N)]$$

MVA - Residence time equation

West Virginia University

Arrival Theorem for closed QN: average number of requests in the queue i as seen by an arriving request when there are N requests in the QN is equal to the average number of requests in queue i in the QN with N-1 requests (arriving request cannot find itself in the queue)

$$A_i(N) = \bar{N}_i(N-1)$$

It follows that

$$R_i(N) = S_i[1 + A_i(N)] = S_i[1 + \bar{N}_i(N-1)]$$

Multiplying both sides by V_i we get

$$R'_i(N) = D_i[1 + \bar{N}_i(N-1)]$$

For the response time *R(N)* we get $R(N) = \sum_{i=1}^{K} R'_i(N)$

MVA – Throughput equation

- Throughput equation

  Applying Little's law for the entire QN we get

$$X(N) = \frac{N}{Z + R(N)} = \frac{N}{Z + \sum_{i=1}^{K} R'_i(N)}$$

MVA – Queue length equation

- Queue length equation

  Appling the Little's law and the Forced Flow law we get

$$\bar{N}_i(N) = X_i(N) \cdot R_i(N) = X(N) \cdot V_i \cdot R_i(N) = X(N) \cdot R'_i(N)$$

MVA – Summary

- Residence time equation
  - Queuing resource $R'_i(N) = D_i[1 + \bar{N}_i(N-1)]$
  - Delay resource $R'_i(N) = D_i$
- Throughput equation

$$X(N) = \frac{N}{Z + R(N)} = \frac{N}{Z + \sum_{i=1}^{K} R'_i(N)}$$

- Queue length equation

$$\bar{N}_i(N) = X(N) \cdot R'_i(N)$$

MVA – Summary

- We start with $N=0$ and work our way up to the value of N we are interested in
- Results for $N=0$ are trivial because when there are no requests in the QN, the queue lengths are 0 for all queues, that is, $\bar{N}_i(0) = 0$ for all i's
- Sequence of computations for MVA

$$\bar{N}_i(0) \rightarrow R'_i(1) \rightarrow X(1) \rightarrow \bar{N}_i(1) \rightarrow R'_i(2) \rightarrow X(2) \rightarrow \bar{N}_i(2) \cdots$$

West Virginia University

MVA - Other model outputs

- System response time: $R = N/X - Z$
- Average number in system: $\bar{N} = N - XZ$
- Throughput of device i: $X V_i$
- Utilization of device i: $X D_i$

West Virginia University

Closed queuing network - bounds

- $U_i(N) = X(N) \cdot D_i \leq 1$ (no utilization can exceed 1)

  Since the bottleneck device is the first to saturate, it restricts the system throughput most severely

$$X(N) \leq \frac{1}{\max_{1 \leq i \leq K} D_i}$$

Closed queuing network - bounds

$$X(N) = \frac{N}{Z + \sum_{i=1}^{K} R'_i(N)}$$

Since

- Queuing resource $R'_i(N) = D_i[1 + \bar{N}_i(N-1)]$
- Delay resource $R'_i(N) = D_i$

It follows that $R'_i(N) \geq D_i$

that is $X(N) \leq \dfrac{N}{Z + \sum_{i=1}^{K} D_i}$

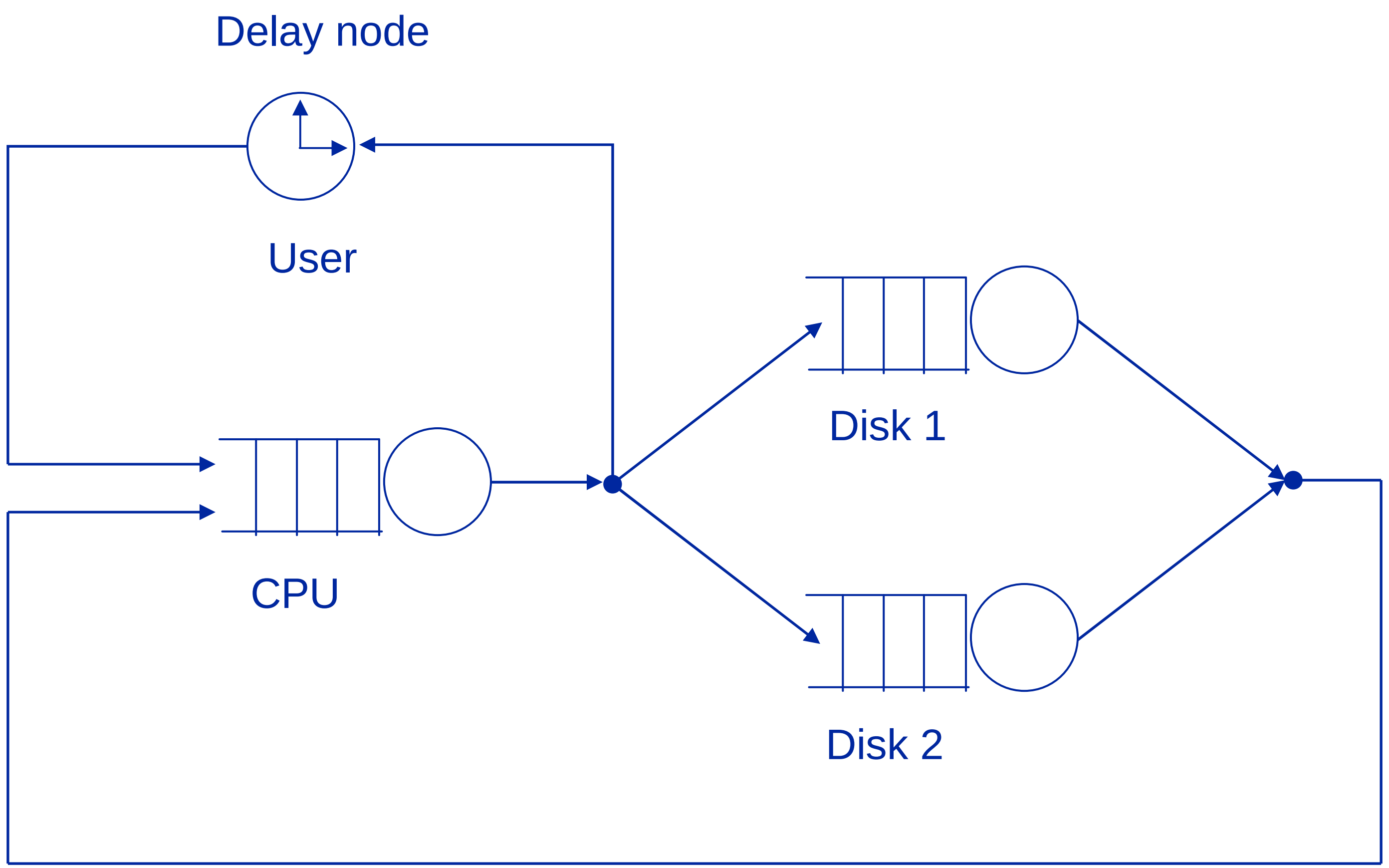
Closed queuing network - bounds

- The observations from the two previous slides can be summarized for the bound on the throughput of the closed queuing network as

$$X(N) \le min\left[\frac{1}{\max\limits_{1 \le i \le K} D_i}, \frac{N}{Z + \sum\limits_{i=1}^{K} D_i}\right]$$

West Virginia University

Example: Closed queuing network



Example: Closed queuing network

West Virginia University

Use the same example as on Slide 14 with transaction workload replaced by terminal workload. The terminal class have three customers ($N=3$) and average think time of 15 seconds ($Z=15$ sec).

$D_{CPU} = 0.605$ sec

$D_{disk1} = 2.1$ sec

$D_{disk2} = 1.35$ sec

Example: Closed queuing network

	i	N=0	N=1	N=2	N=3
R'_i	CPU	-	0.605	0.624	0.644
	Disk 1	-	2.1	2.331	2.605
	Disk 2	-	1.35	1.446	1.551
X		-	0.0525	0.1031	0.1515
$\bar{N}_i$	CPU	0	0.0318	0.0643	0.0976
	Disk 1	0	0.1102	0.2403	0.3947
	Disk 2	0	0.0708	0.1490	0.2350

R = 4.8 sec

X = 0.1515 request/sec

$\bar{N}$ = 0.7273 request

Why average number of request in the system does not equal the population?

In the class of terminal type some of the customers are "thinking" (average number XZ)

Example: Closed queuing network

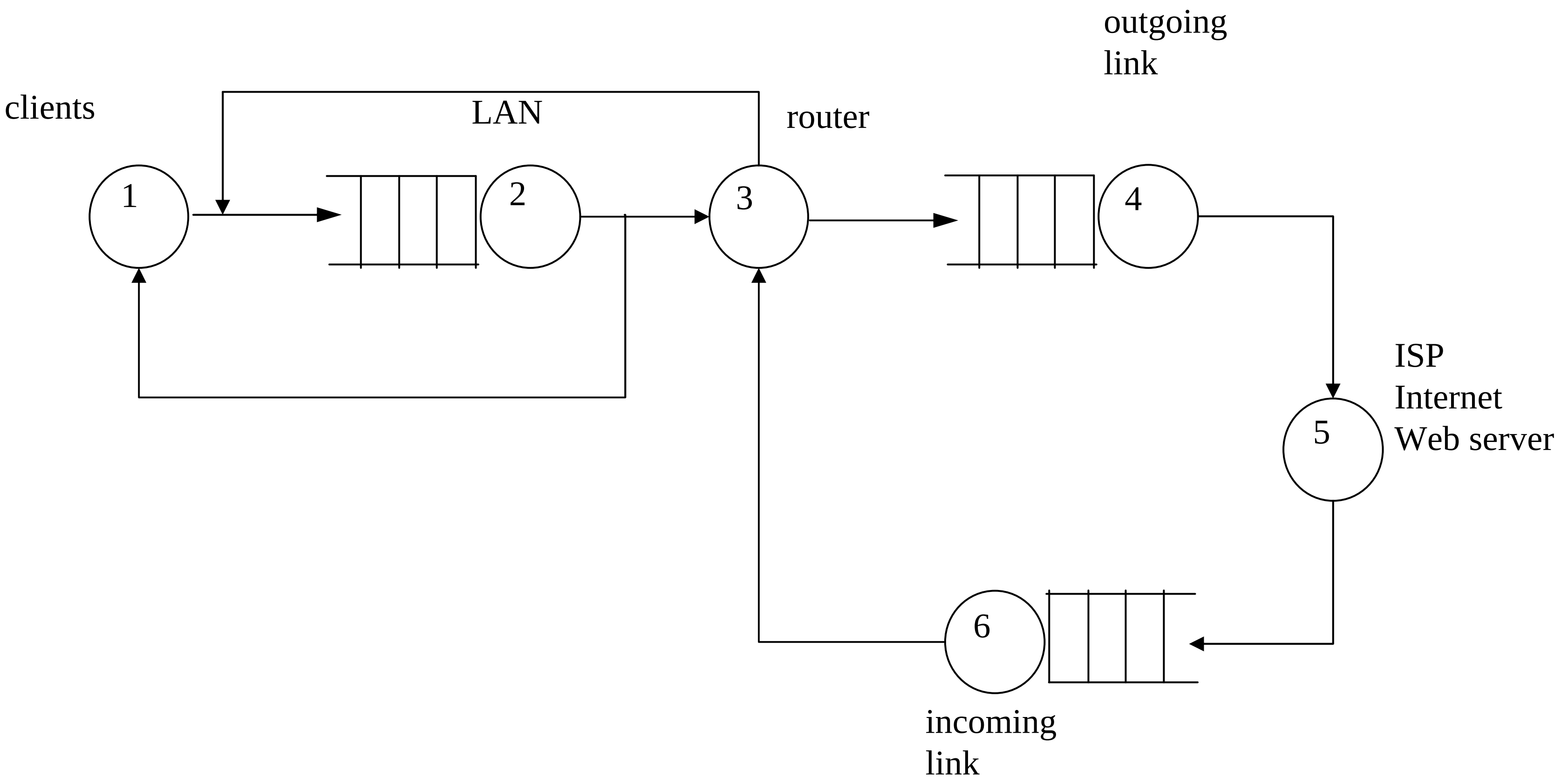
- $1/\ max\{0.605, 2.1, 1.35\} = 0.476$ request / sec

  Disk 1 is the bottleneck – the resource with the highest value of service demand

- $$\frac{N}{Z + \sum_{i=1}^{K} D_i} = \frac{3}{15 + 4.055} = 0.157 \text{ request /sec}$$

- $X(3) \leq min\{0.476, 0.157\} = 0.157$ request/sec

QN model of the clients accessing Web server



PART II – SPE Models

System Execution Models:
Case Studies

West Virginia University

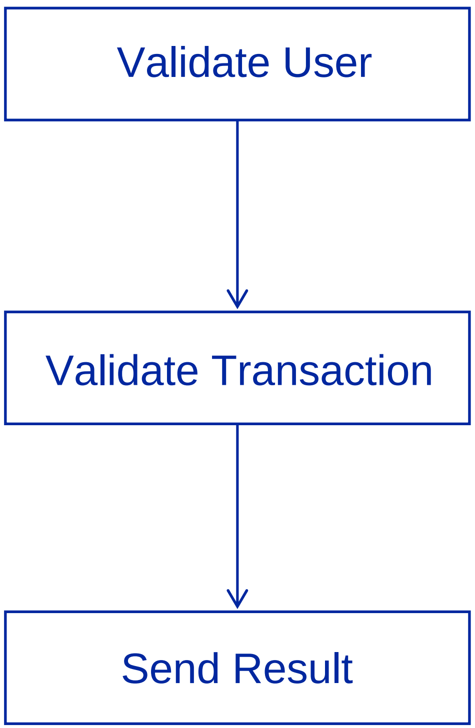
Outline

- 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)
  - Queuing networks
    - Open queuing networks
    - Closed queuing networks
- Case studies

Case study 1 - **authorize Transaction**

West Virginia University

Consider the software execution model of the **authorize Transaction** from the lecture Software execution models



Validate User:

Work Units	1
DB	2
Messages	0

Validate Transaction:

Work Units	2
DB	3
Messages	0

Send Result:

Work Units	2
DB	1
Messages	1

Work Units specify relative CPU consumption. Range of values from 1 to 5 (1 - simple task, 5 - the most complex task). The processing overhead then specifies an approximate number of machine instructions for the simple task

Software resources

Case study 1 - **authorize Transaction**

West Virginia University

Devices	CPU	Disk	Network
Quantity	1	1	1
Service Units	KInstr.	Phys. I/O	Msgs.

Name, quantities, and service units of the computer devices

Work Unit	20	0	0
DB	500	2	0
Massages	10	2	1

Connection between software resources and computer device usage; for example DB requires 500K CPU instructions, 2 I/Os, and 0 network messages

Service Time	0.00001	0.02	0.01

Service time

Processing step	CPU KInstr	Phys. I/O	Network Msgs
Validate User	1,020	4	0
Validate Transaction	1,540	6	0
Send Result	550	4	1
Total	3,110	14	1

Step 1: Estimate total computer resources for each step in the software execution model

Step 2: Estimate total computer resources using reduction rules

Case study 1 - **authorize Transaction**

- **Step 3:** Estimate elapsed time by multiplying total resource requirements for each computer resource by the service time for that resource (from the last row), and summing the result for each resource

$$3{,}110 * 0.00001 + 14 * 0.02 + 1 * 0.01 = 0.3211 \textit{ sec}$$

- This is an optimistic estimate since it does not consider the queuing delays due to contention for system resources

Case study 1 - **authorize Transaction**

West Virginia University

- Software execution model provides the following input parameters for the system execution model
  - K - number of queues (devices): CPU, disk, and network
  - For each device i
    - V_i (average number of visits to device i by a request) and
    - S_i (average service time of a request at device i per visit)

OR

- $D_i = V_i \cdot S_i$ (service demand)

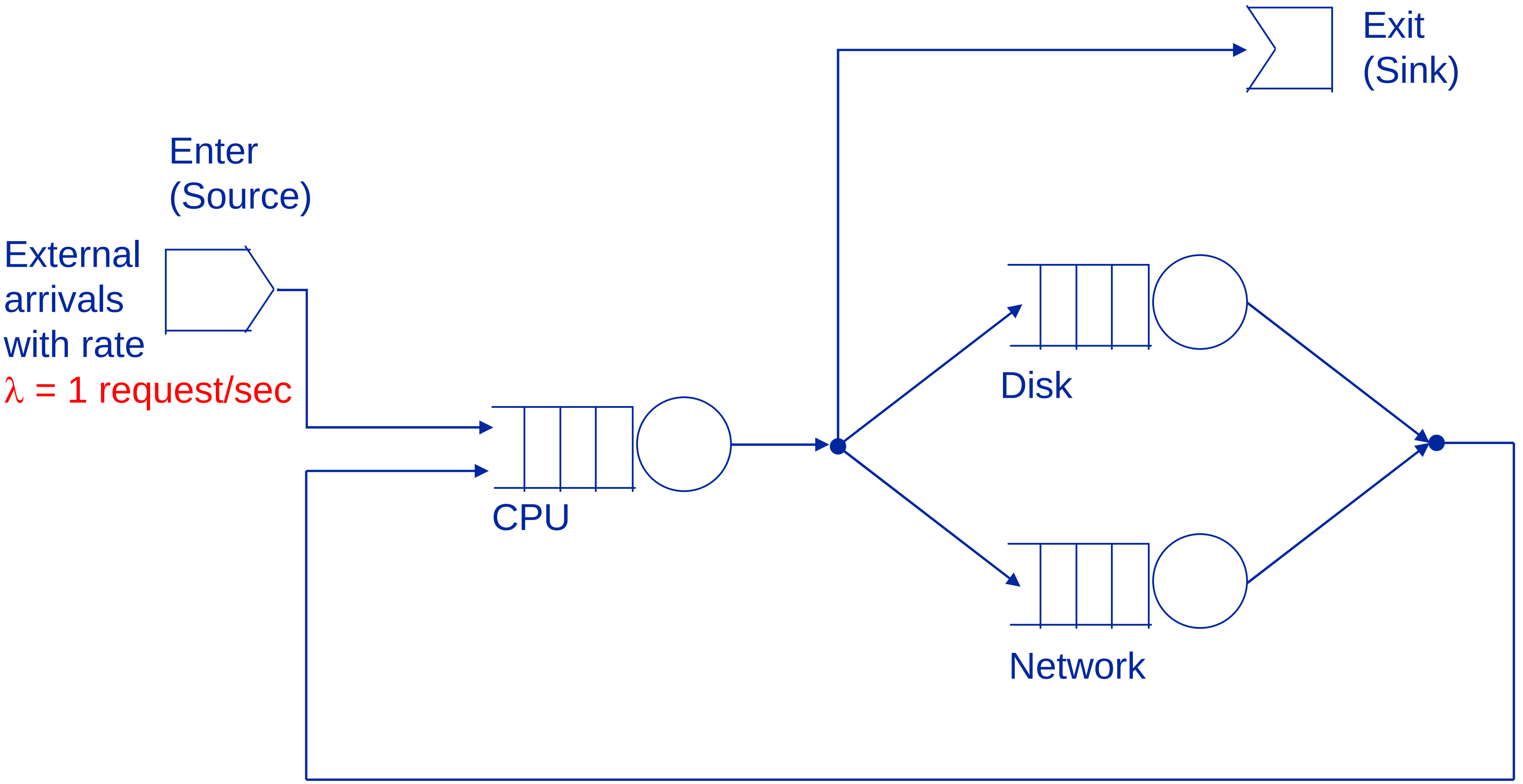
Device	Average number of visits V_i	Average service time per visit S_i	Service demand $D_i = V_i \cdot S_i$
CPU	-	-	0.0311
Disk	14	0.02	0.28
Network	1	0.01	0.01

West Virginia University

Case study 1 - **authorize Transaction**

- Next step is to decide whether the system is best modeled as an open or closed QN
  - Open QN is suitable for the situations such as transaction processing system, where requests arrive, receive some service, and leave the system
    - Input parameter: λ (arrival rate of requests to the QN)
  - Closed QN are more appropriate for interactive systems where the users enter a request, receive the results, and then enters another request
    - Input parameters: N (population in the QN) and Z (think time, could be 0)
- This is a typical example of transaction system. Therefore, we will use an open queuing network. Assume that the arrival rate is λ = 1 request / sec.

Case study 1 - **authorize Transaction**



West Virginia University

Case study 1 - **authorize Transaction**

- System throughput $X = \lambda$
- Device throughput $X_i = V_i X$
- Device utilization $U_i = \lambda D_i$
- Residence time at device $R'_i = \frac{D_i}{1 - U_i}$
- System response time $R = \sum_{i=1}^{K} R'_i$
- Queue length at device $\bar{N}_i = \frac{U_i}{1 - U_i}$
- Average number in system $\bar{N} = \sum_{i=1}^{K} \bar{N}_i$

West Virginia University

Case study 1 - **authorize Transaction**

- Average throughput of the system is equal to the average arrival rate (Flow balance law)

  $X = \lambda = 1$ request /sec

- Device throughput (Forced flow low)

  X_{CPU} cannot be estimated

  $X_{disk} = V_{disk}\, X = 14 \cdot 1 = 14$ request /sec

  $X_{network} = V_{network}\, X = 1 \cdot 1 = 1$ request /sec

- Utilization of the devices (Service Demand law)

  $U_{CPU} = D_{CPU}\, X = D_{CPU}\, \lambda = 0.0311 \cdot 1 = 0.0311 = 3.11\%$

  $U_{disk} = D_{disk}\, X = D_{disk}\, \lambda = 0.28 \cdot 1 = 0.28 = 28\%$

  $U_{network} = D_{network}\, X = D_{network}\, \lambda = 0.01 \cdot 1 = 0.01 = 1\%$

West Virginia University

Case study 1 - **authorize Transaction**

- Residence times of a request at device

  $R'_{CPU} = D_{CPU}/(1- U_{CPU}) = 0.0311/(1- 0.0311) = 0.0321$ sec

  $R'_{disk} = D_{disk}/(1- U_{disk}) = 0.28/(1-0.28) = 0.3889$ sec

  $R'_{network} = D_{network}/(1- U_{network}) = 0.01/(1-0.01) = 0.0101$ sec

- Total response time

  $R = R'_{CPU} + R'_{disk} + R'_{network} = 0.0321 + 0.3889 + 0.0101$

  $= 0.4311$ sec

- Compare with the time obtained from software execution model 0.3211 sec which excludes queuing delays when multiple processes want to use the same computer resources in the same time

Case study 1 - **authorize Transaction**

West Virginia University

- Average number of requests at each device

$$\overline{N}_{CPU} = U_{CPU}/(1- U_{CPU}) = 0.0311/(1- 0.0311) = 0.0321$$

$$\overline{N}_{disk} = U_{disk}/(1- U_{disk}) = 0.28/(1-0.28) = 0.3889$$

$$\overline{N}_{network} = U_{network}/(1- U_{network}) = 0.01/(1-0.01) = 0.0101$$

- Total number of request at the system

$$\overline{N} = \overline{N}_{CPU} + \overline{N}_{disk} + \overline{N}_{network} = 0.0321 + 0.3889 + 0.0101$$
$$= 0.4311 \text{ request}$$

West Virginia University

Case study 1 - **authorize Transaction**

- Maximum theoretical arrival rate of requests sustained by this system

  $\lambda_{max} = 1/ \, max\{0.0311, 0.28, 0.01\} = 3.57$ request / sec

  Disk is the bottleneck – the resource with the highest value of service demand. In this example the maximum arrival rate significantly exceeds the actual arrival rate $\lambda = 1$ request / sec

West Virginia University

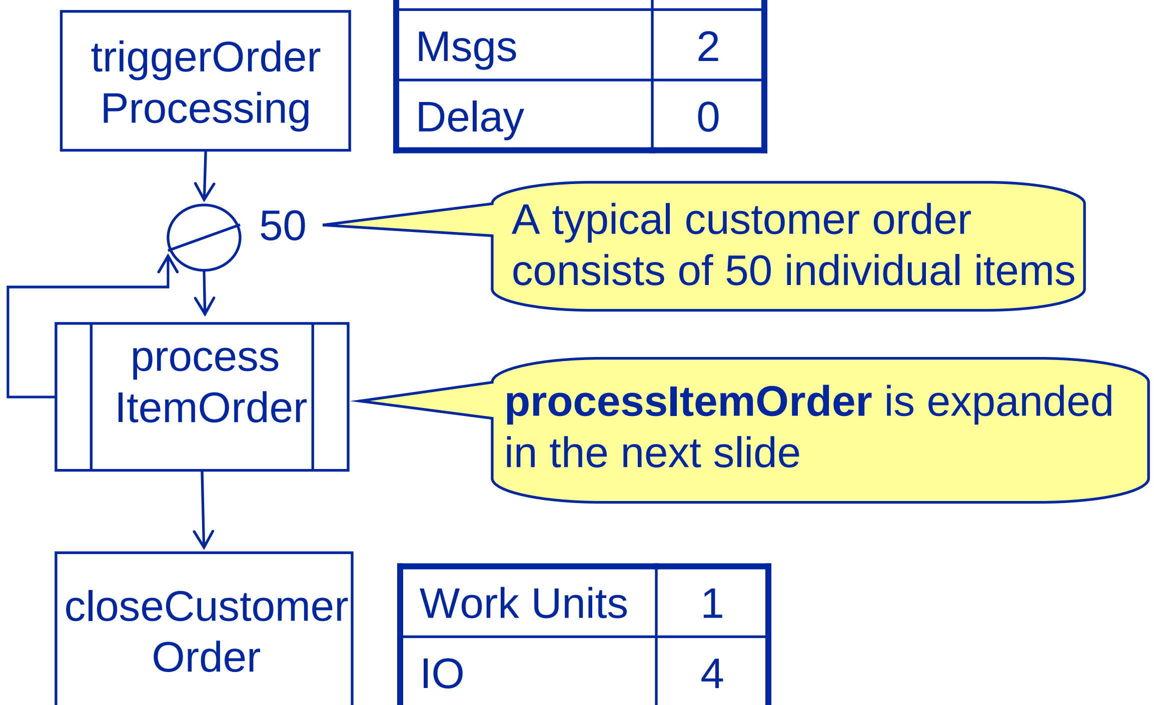
Case study 2: Distributed system

- E-commerce application
- We consider the use case "**processing a new order**" whose sequence diagram is given on the Figure 6-6 (page 154) in the book

Case study 2: Distributed system

Software execution graph

Work Units	1
IO	4
Msgs	2
Delay	0

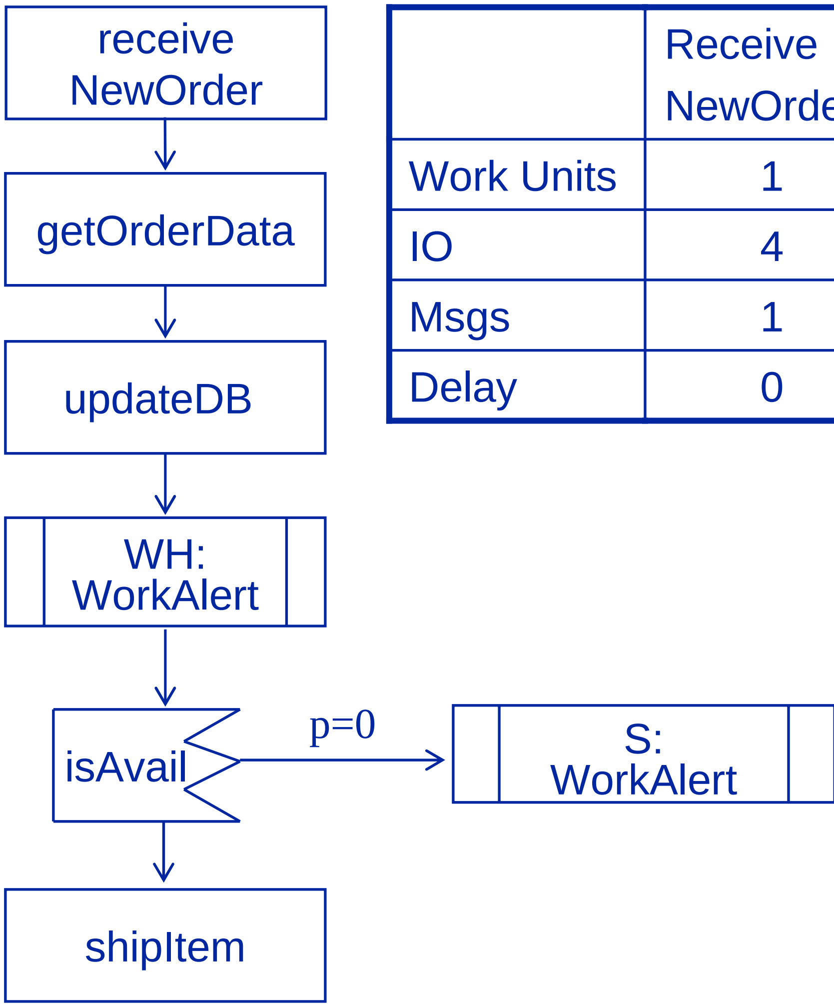


Work Units	1
IO	4
Msgs	1
Delay	0

Case study 2: Distributed system

Expansion of the **processItemOrder**

	Receive NewOrder	getOrder Data	updateDB	isAvail	shipItem
Work Units	1	0	5	0	15
IO	4	0	35	0	170
Msgs	1	2	0	2	8
Delay	0	1	0	0.1	4



Optional step if the item is not in stock; The best case model assumes that the items are available, that is, p=0

West Virginia University

Case study 2: Distributed system

Specify the computer resource requirements

Devices	CPU	Disk	Delay	LAN
Quantity	6	3	1	1
Service Units	Sec.	Phys. I/O	Units	Msgs.

Work Unit	0.01			
DB		1		
Massages	0.0005	1		1
Delay			1	

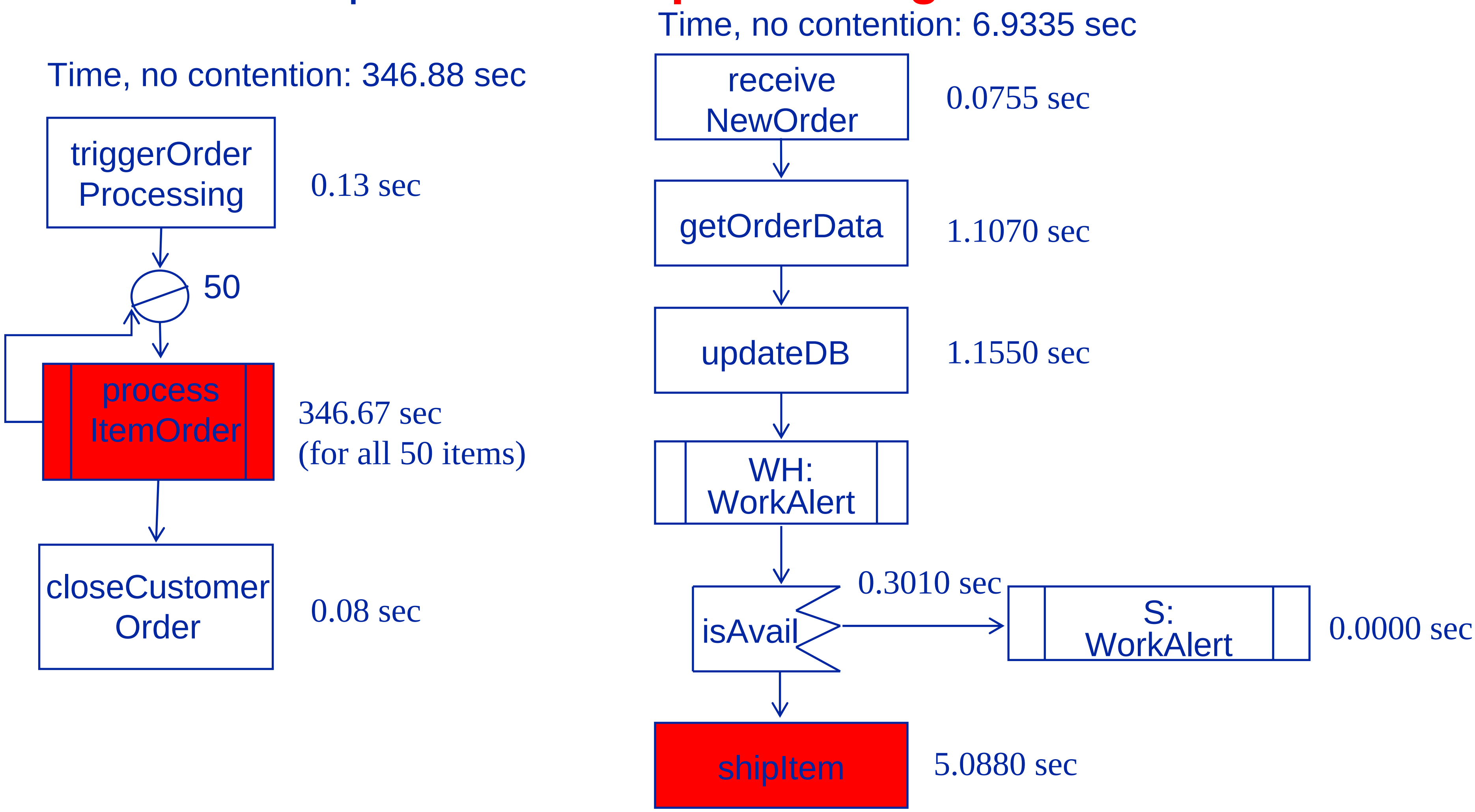
Service Time	1	0.003	1	0.05

Note: The Work Units are derived from measurements that include the processing time for the database, so the DB row has no requirement to CPU resource.
Also, it is assumed that the disk visits are equally distributed over the three Disk devices.

Case study 2: Distributed system

West Virginia University

Best case elapsed time for "**processing a new order**" scenario

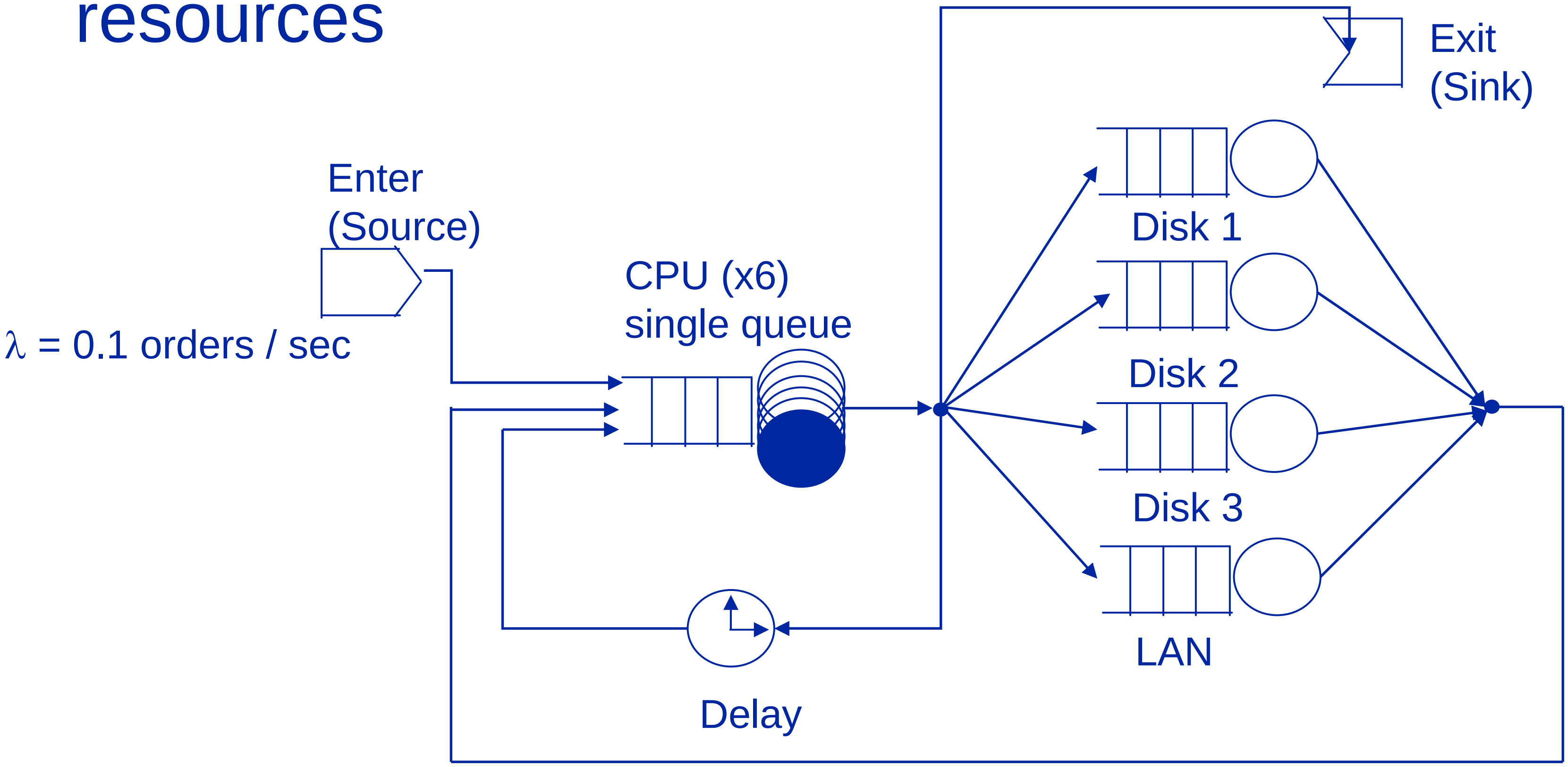


Case study 2: Distributed system

- Best case elapsed time estimated from the software execution model is far worse than the performance objective
- An alternative is considered that processes batches of items, rather than individual items in an order
- It is important to resolve performance problems in the simple software execution model before proceeding to the advanced models

Case study 2: Distributed system

- Next step is to construct and evaluate system execution model that considers contention for resources



Case study 2: Distributed system

- Software execution model for the alternative design provides the input parameters to the system execution model (the open QN on the previous slide)
  - The response time, including the resource contention, is estimated to be 14.9 seconds
  - Expected utilization for the computer resources are
    - 2% for the CPU
    - 5% for the disks
    - 17% for the LAN

West Virginia University

Summary and Modeling Hints

- System models account for multiple users by specifying the workload (either by the arrival rate or the number of users and think time)
- QN models calculate average values. The actual behavior can differ from the average; for example the behavior in a peak hour is not likely to be the same as the average behavior
- Stay with synchronous and asynchronous communication when possible to simplify software implementation and testing

Summary and Modeling Hints

- Bottleneck device is the one with the highest demand or highest utilization; This is the device that will limit the scalability
- Remove the bottleneck by changes to software design or hardware configuration (use faster device or add more devices)
- Determine the scalability of the system by solving the model using projected future workload. If the response time is not acceptable, identify the bottlenecks