# Operational Laws

## IN4390 Quantitative Evaluation of Embedded Systems

Koen Langendoen

# Queueing theory

## Yet another take at performance evaluation

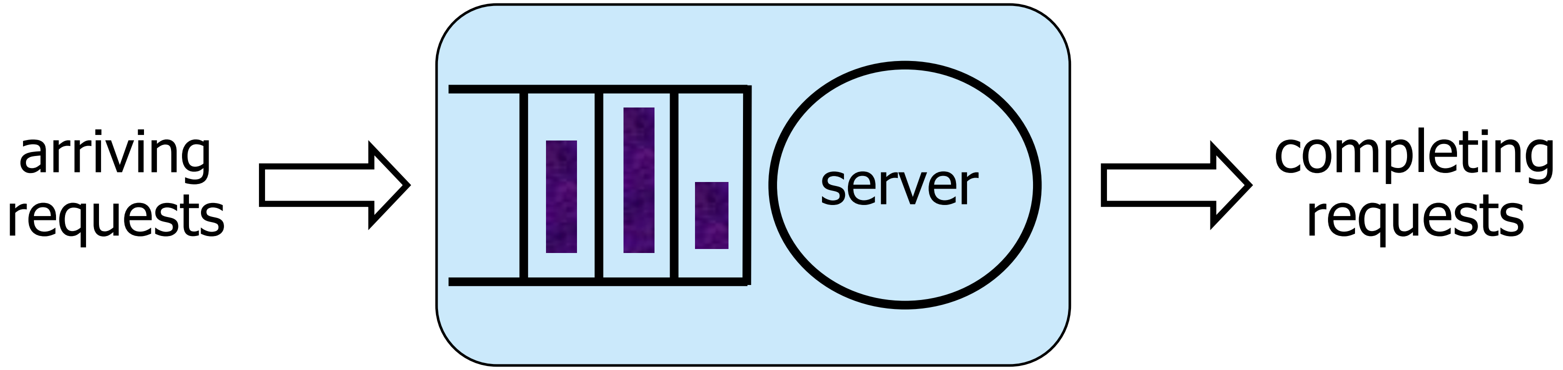
- Measurements
  - DoE
- Simulations
  - …
- Modeling
  - Petri nets
  - Markov modeling
  - **Queueing theory**

memoryless
makes for
easy math

# Queues are everywhere
## **When demand exceeds supply**


- impacts response time
- signals bottleneck(s)

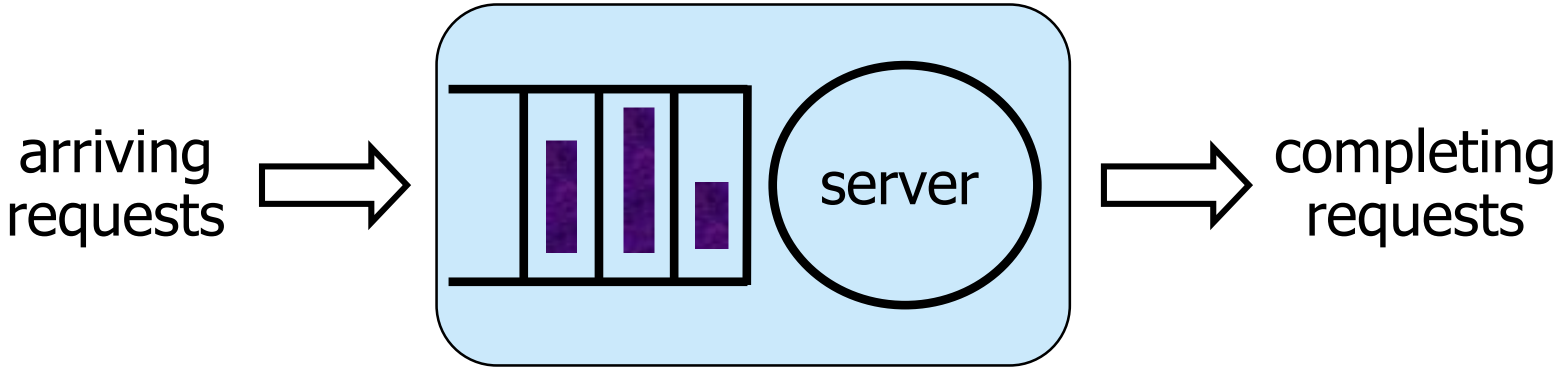
# A queue




Characterized by
- arrival distribution (rate)
- request size (service time)
- scheduling discipline

# A queueing system

## Kendall notation




Characterized by **A/S/m/B/K/SD**

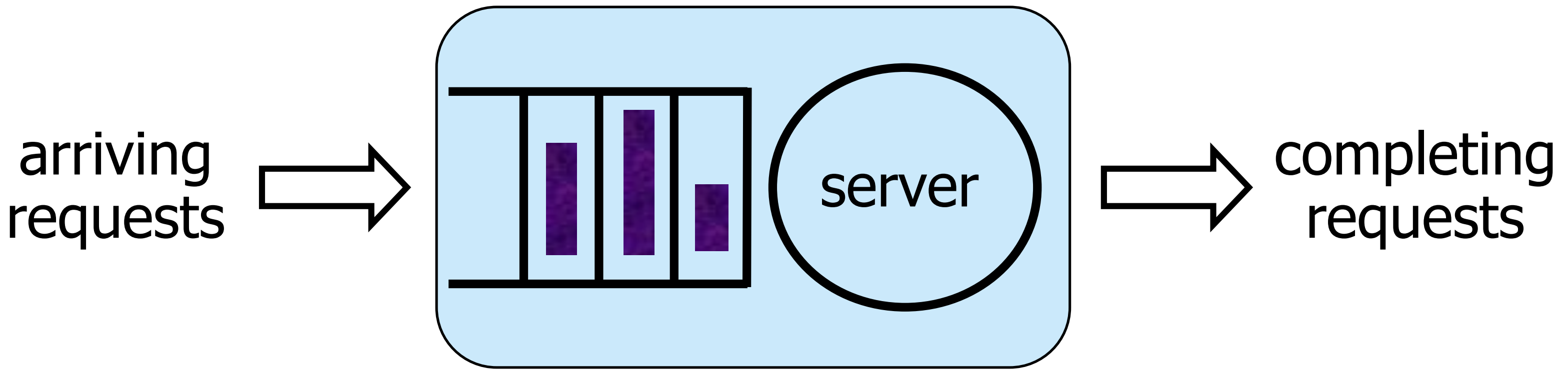
- **A**: interarrival time distr.
- **S**: service time distr.

} M | E | D | … | G

- **m**: #servers
- **B**: #buffers
- **K**: population size
- **SD**: Service Discipline

# A queueing system

## Kendall for dummies




Characterized by **A/S/m**

- **A**: interarrival time distr.
- **S**: service time distr.
- **m**: #servers

M | E | D | ... | G

# Operational laws

## Hold for any distributions and scheduling

- Operational ⇒ directly measurable
- Testable assumptions
  - #arrivals = #completions [job flow balance]
- Observable variables
  - arrival rate
  - service time
  - waiting time
  - …

what happens when violated?

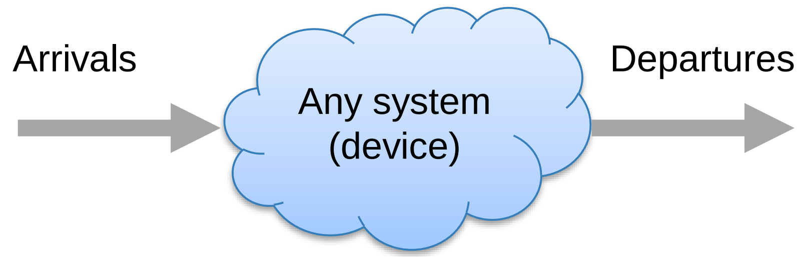
= random variable?

operational law = relation between observables

# Operational quantities

large or small?

- Measure during a time period t
  - A: #arrivals
  - C: #completions
  - B: busy time




- Derive **mean** values
  - arrival rate: $\lambda = A / t$
  - throughput: $X = C / t$
  - utilization: $U = B / t$
  - service time: $S = B / C$
  - service rate $\mu = 1 / S$

# Operational laws

| | |
|---|---|
| **UTILIZATION LAW** | $U = X S$ |
| **LITTLE'S LAW** | $N = X R$ |
| **FORCED FLOW LAW** | $X_k = V_k X$ |
| **BOTTLENECK LAW** | $U_k = D_k X$ |

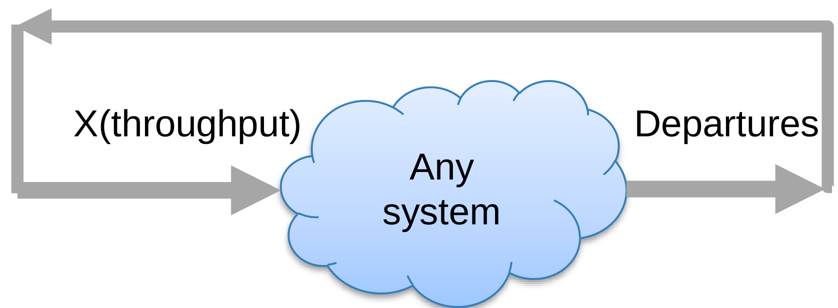
Good for

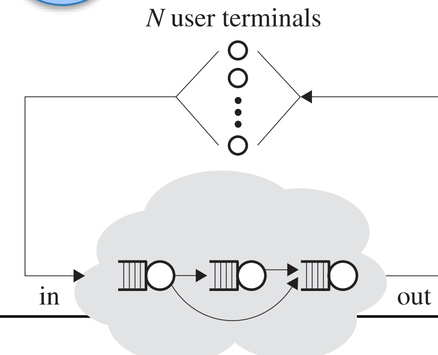
- "back of the envelope" calculations
- determining performance bounds

# System model

- Open
  - external arrivals
  - e.g. web server


- Closed
  - fixed number of jobs
  - e.g. thread pool
  - e.g. time-sharing system



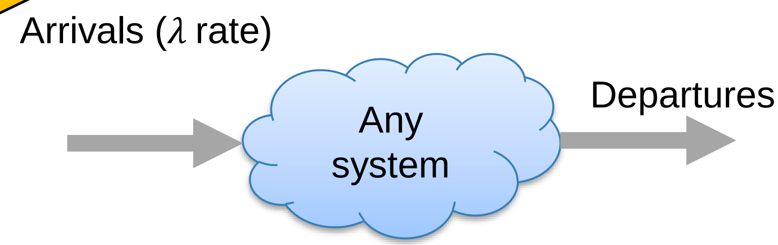



# Utilization law

implicit assumption?

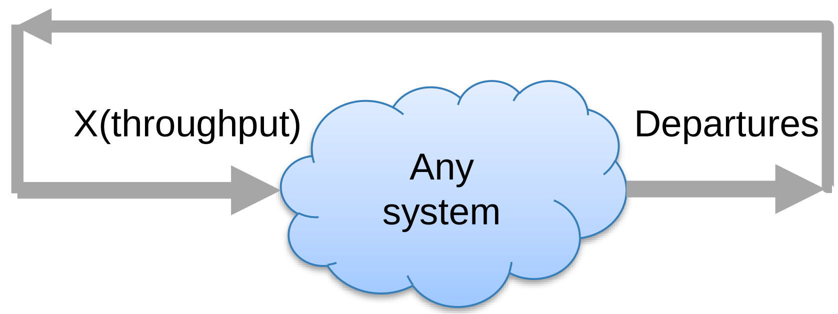
- Open

$$\mathbf{U} = \lambda\, \mathbf{S}$$




- Closed

$$\mathbf{U} = \mathbf{X}\, \mathbf{S}$$



# Little's law

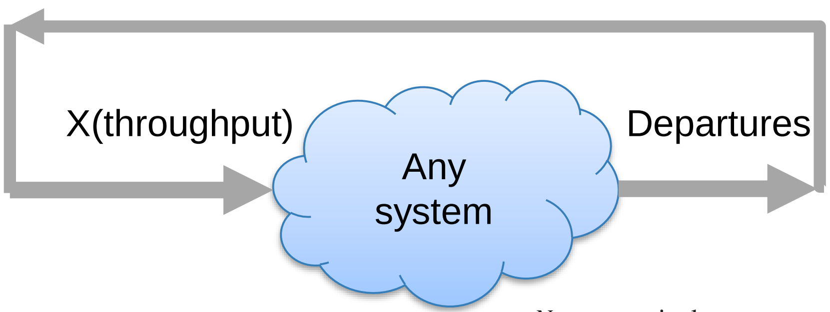
how to proof?

- Open

$$\boxed{N = X\,R}$$

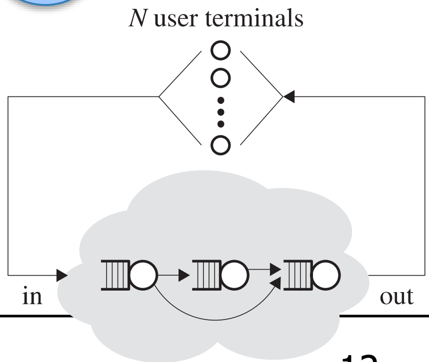

- Closed

$$\boxed{N = X\,(R+Z)}$$

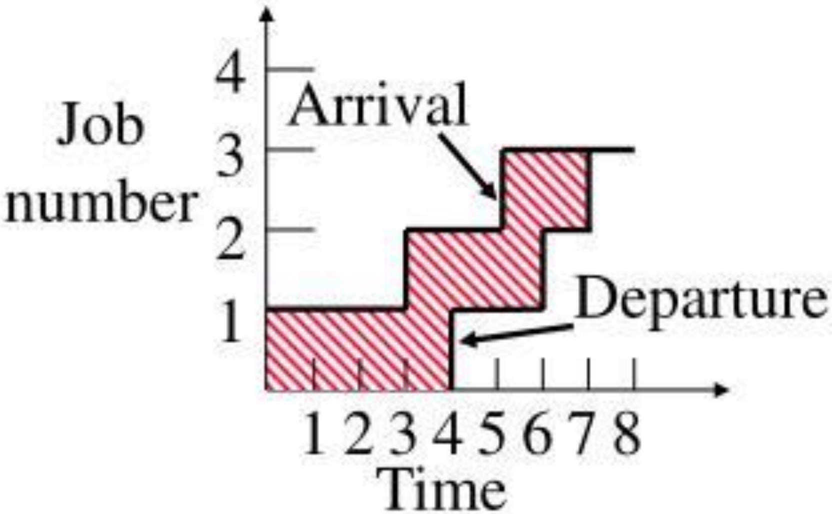


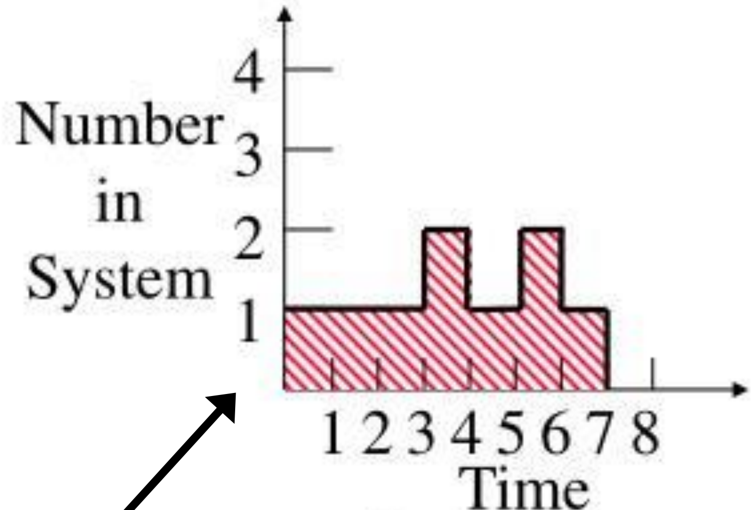

- N: #jobs in the system
- R: Response time
- Z: think time

**mean** values

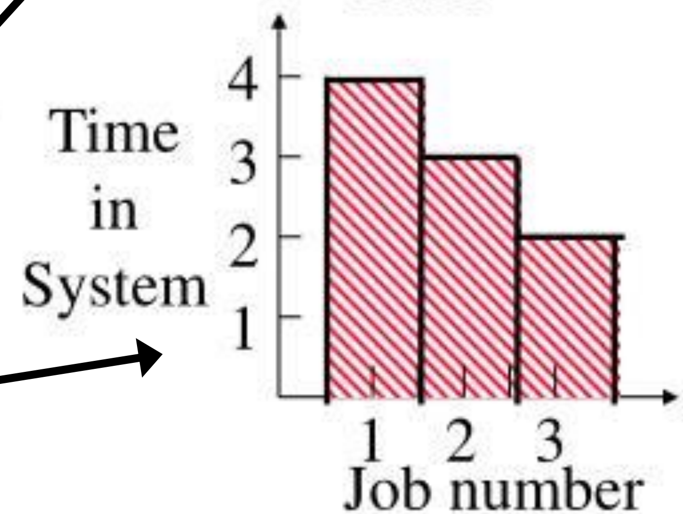


# Proof of Little's Law



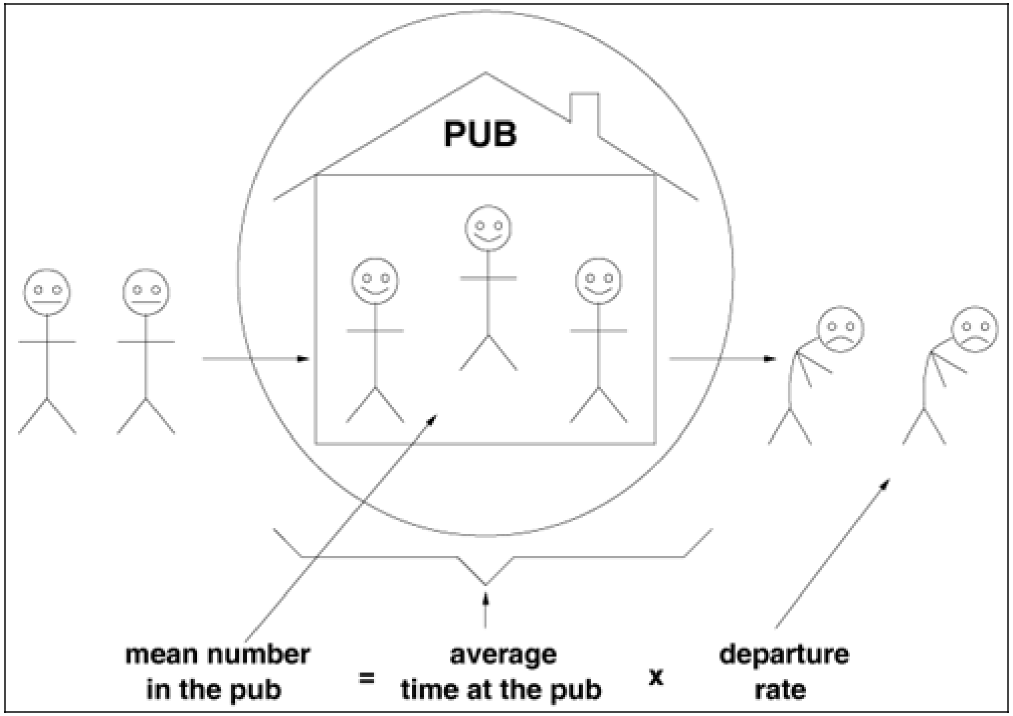




- J = hatched area = total time spent in the system by all jobs
- X = A/t
- N = J/t vertical view
- R = J/C horizontal view
- N = X R Little's Law



# Little’s law

## Cheers!




**N = R X**

# Operational laws

| | |
|---|---|
| UTILIZATION LAW | $U = X\,S$ |
| LITTLE'S LAW | $N = X\,R$ |
| FORCED FLOW LAW | $X_k = V_k\,X$ |
| BOTTLENECK LAW | $U_k = D_k\,X$ |

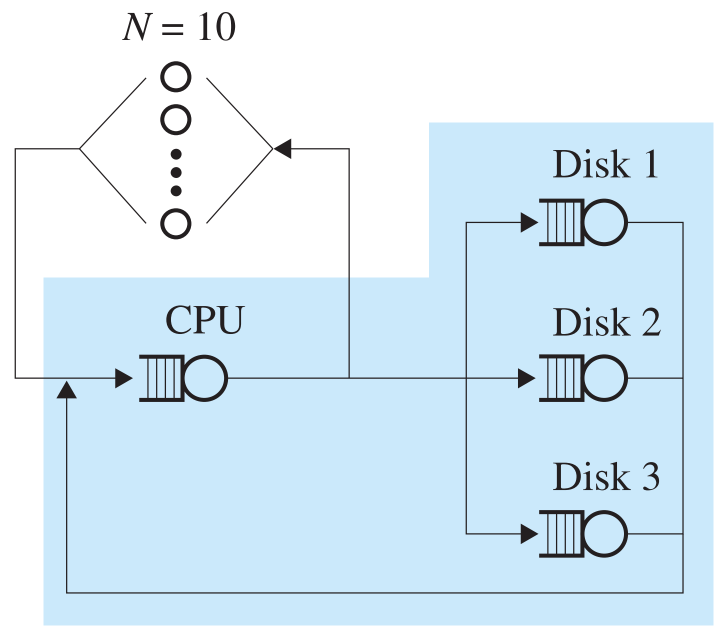
Good for

- "back of the envelope" calculations
- determining performance bounds

# Systems of systems

- Multiple resources
- Connected queues

- Operational laws apply
  - individual queues
  - system as a whole



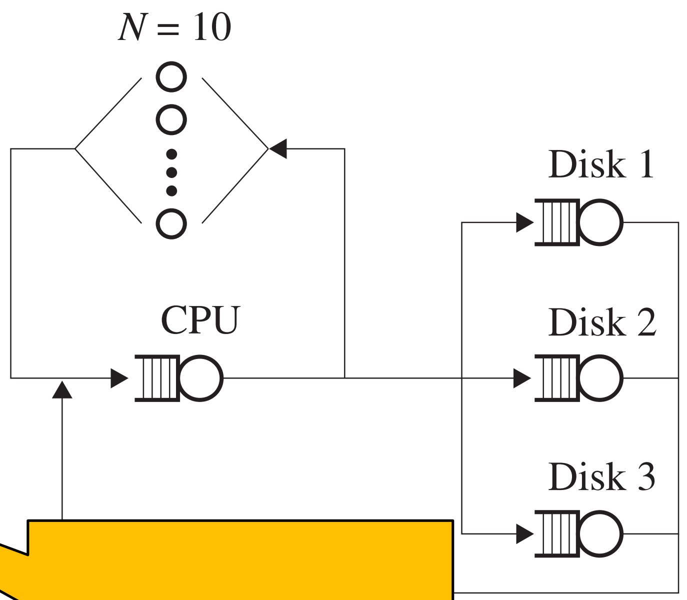

  - number devices (1 .. #resources) $\quad N_k = X_k R_k$
  - top-level system is device 0 $\quad N_0 = X_0 (R_0+Z)$

# Forced Flow law

- Relate system throughput to device throughput

$$\boxed{X_k = V_k\, X}$$

any intuition?




  - $V_k$: #visits to device k per job

- Proof by counting
  - $X_k = \frac{C_k}{t} = \frac{C_k}{C_0}\frac{C_0}{t} = V_k\, X$

# Example

## What one rattling disk reveals

Suppose a system has:

- 30 terminals
- 18 seconds average think time
- 20 visits to a specific disk/interaction
- 30% utilization of that disk
- 25 ms is the average time for a disk access

$N = 30$
$Z = 18$
$V_{disk} = 20$
$U_{disk} = 0.30$
$S_{disk} = 25$ ms

Compute the system throughput and response time.

# Example

## What one rattling disk reveals

Compute the

- system throughput
- response time

$$X_k = V_k X$$

$N = 30$
$Z = 18$
$V_{disk} = 20$
$U_{disk} = 0.30$
$S_{disk} = 25$ ms

# Operational laws

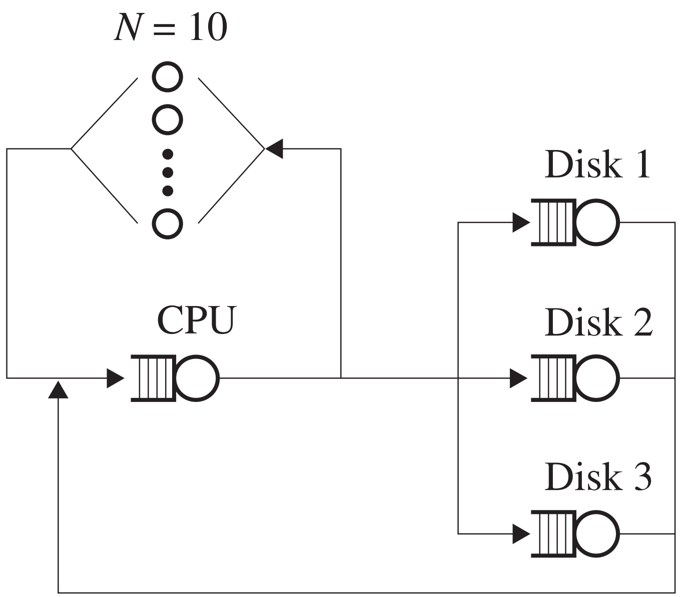
| | |
|---|---|
| UTILIZATION LAW | $U = X\,S$ |
| LITTLE'S LAW | $N = X\,R$ |
| FORCED FLOW LAW | $X_k = V_k\,X$ |
| BOTTLENECK LAW | $U_k = D_k\,X$ |

Good for

- "back of the envelope" calculations
- determining performance bounds

# Bottleneck law

- Relate system throughput to device utilization

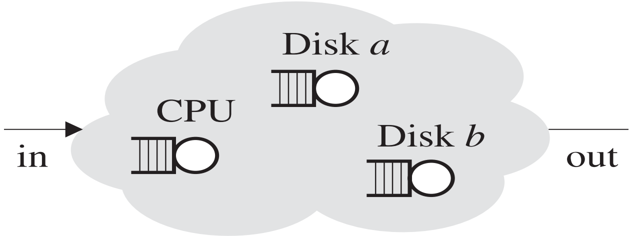



$$\boxed{\mathbf{U_k = D_k\ X}}$$

  - $D_k = V_k\ S_k$: total service demand on device k for all visits of a job

- The device with the highest utilization (demand) is the bottleneck in the system

# Webserver example




- Measurements taken during one hour from a Web server indicate that the utilization of the CPU and the two disks are: $U_{CPU} = 0.25$, $U_{diskA} = 0.35$, and $U_{diskB} = 0.30$. The server log shows that 216,000 requests were processed during the measurement interval.

- What are the service demands (the time used by each request) at the CPU and both disks?
- Which component is the bottleneck?
- What is the maximum (potential) throughput?
- What was the response time of the Web server?

tricky!

# Asymptotic Bounds for Closed Systems

$$X \le \boldsymbol{min}\left(\frac{\mathbf{1}}{\boldsymbol{D_{max}}}, \frac{\boldsymbol{N}}{\boldsymbol{D+Z}}\right)$$

$$R \ge \boldsymbol{max}(\boldsymbol{D}, \boldsymbol{N} \times \boldsymbol{Dmax} - \boldsymbol{Z})$$

1 or 2 proofs required?

$$\mathbf{D} = \sum_k \mathbf{D}_k$$

$$\mathbf{D}_{\boldsymbol{max}} = max_k\ \mathbf{D}_\mathrm{k}$$

**Upper bounds** on throughput

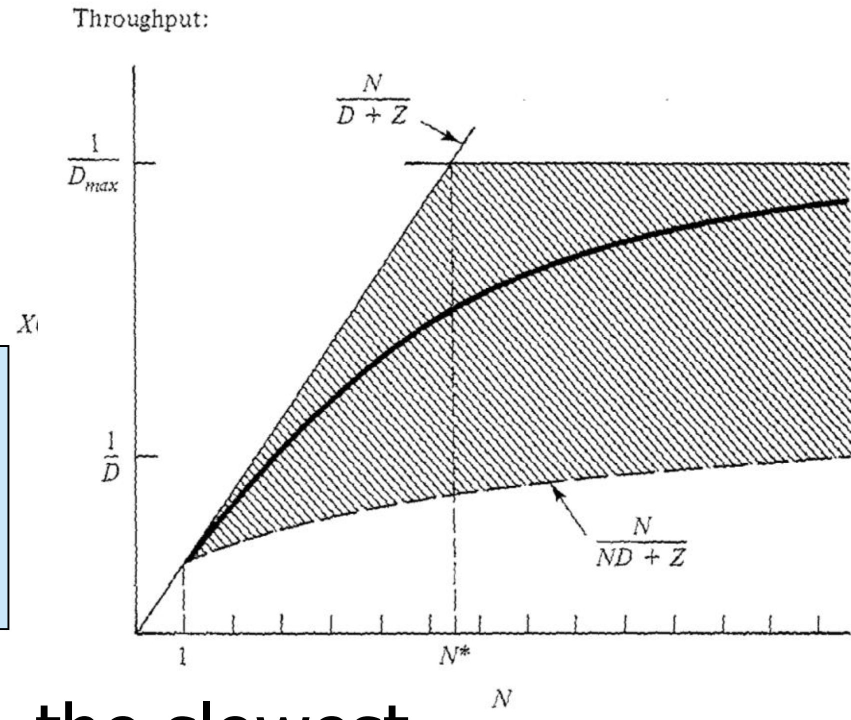
**lower bounds** on response time

- can be obtained by considering the service demands only (i.e., without solving any underlying model)

# Proof

## Throughput bounds

$$X \le \min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right)$$




- when loading the system, the slowest device becomes the bottleneck:
  - $X = U_k / D_k \le 1/D_k \quad X \le 1/D_{max}$

$$U_k = D_k X$$

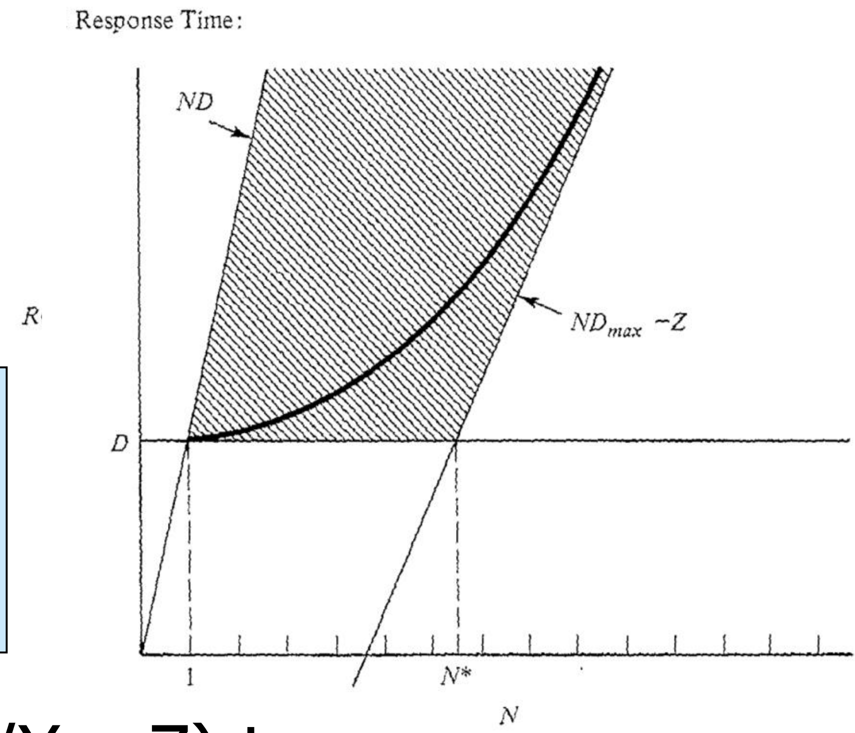
- max throughput when no queueing occurs ($R \ge D$):
  - $X = N / (R+Z) \le N / (D+Z)$

$$N = X (R+Z)$$

# Proof

## Response time bounds

$$R \geq max(D, N \times Dmax - Z)$$



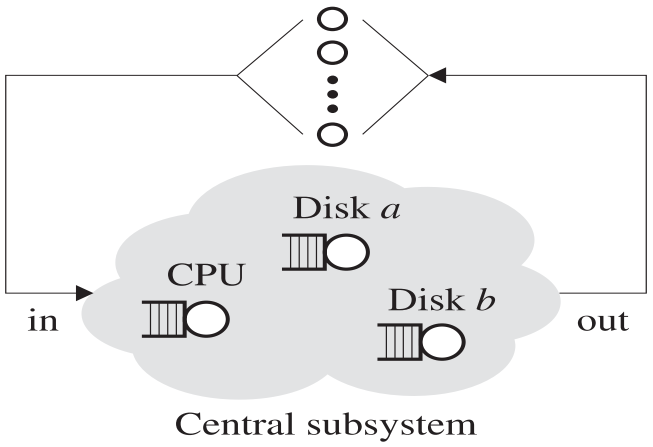

- Apply Little's Law (R = N/X – Z) to previous proof

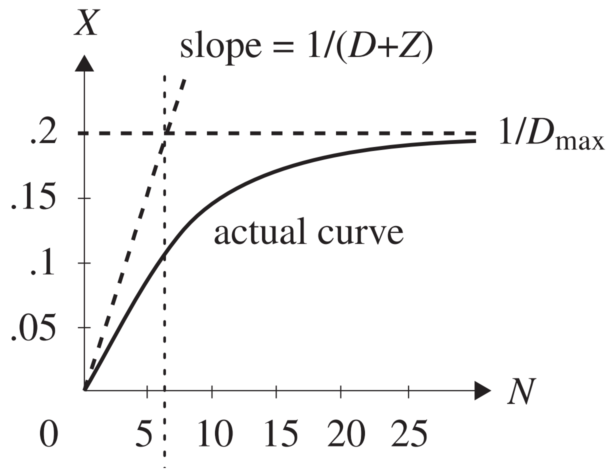
$$X \leq min\left(\frac{N}{D+Z}, \frac{1}{D_{max}}\right)$$

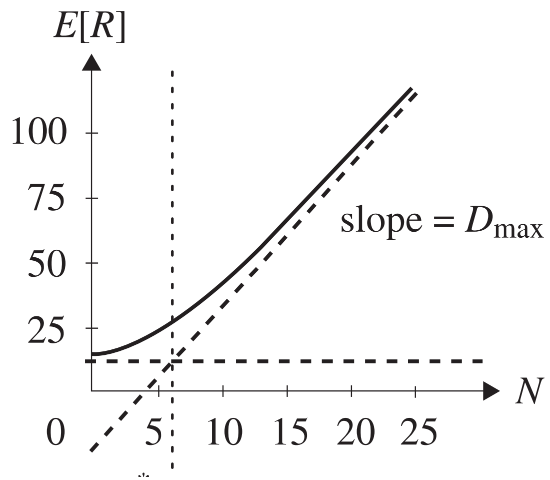
# Webserver example

## Closed system

- $Z = 18$
- $D_{CPU} = 5$
- $D_{disk\ a} = 4$
- $D_{disk\ b} = 3$



# Operational laws and bounds

| | |
|---|---|
| UTILIZATION LAW | $U = X\,S$ |
| LITTLE'S LAW | $N = X\,R$ |
| FORCED FLOW LAW | $X_k = V_k\,X$ |
| BOTTLENECK LAW | $U_k = D_k\,X$ |

$$X \leq min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right)$$

$$R \geq max(D, N \times Dmax - Z)$$