Computer Performance Modeling

Michael Salsburg, Ph.D.

Michael@MSalsburg.com

Motivation

Business Case for Response Time Objectives

JAKOB NIELSEN

The 3 response-time limits are the same today as when I wrote about them in 1993 (based on 40-year-old research by human factors pioneers):

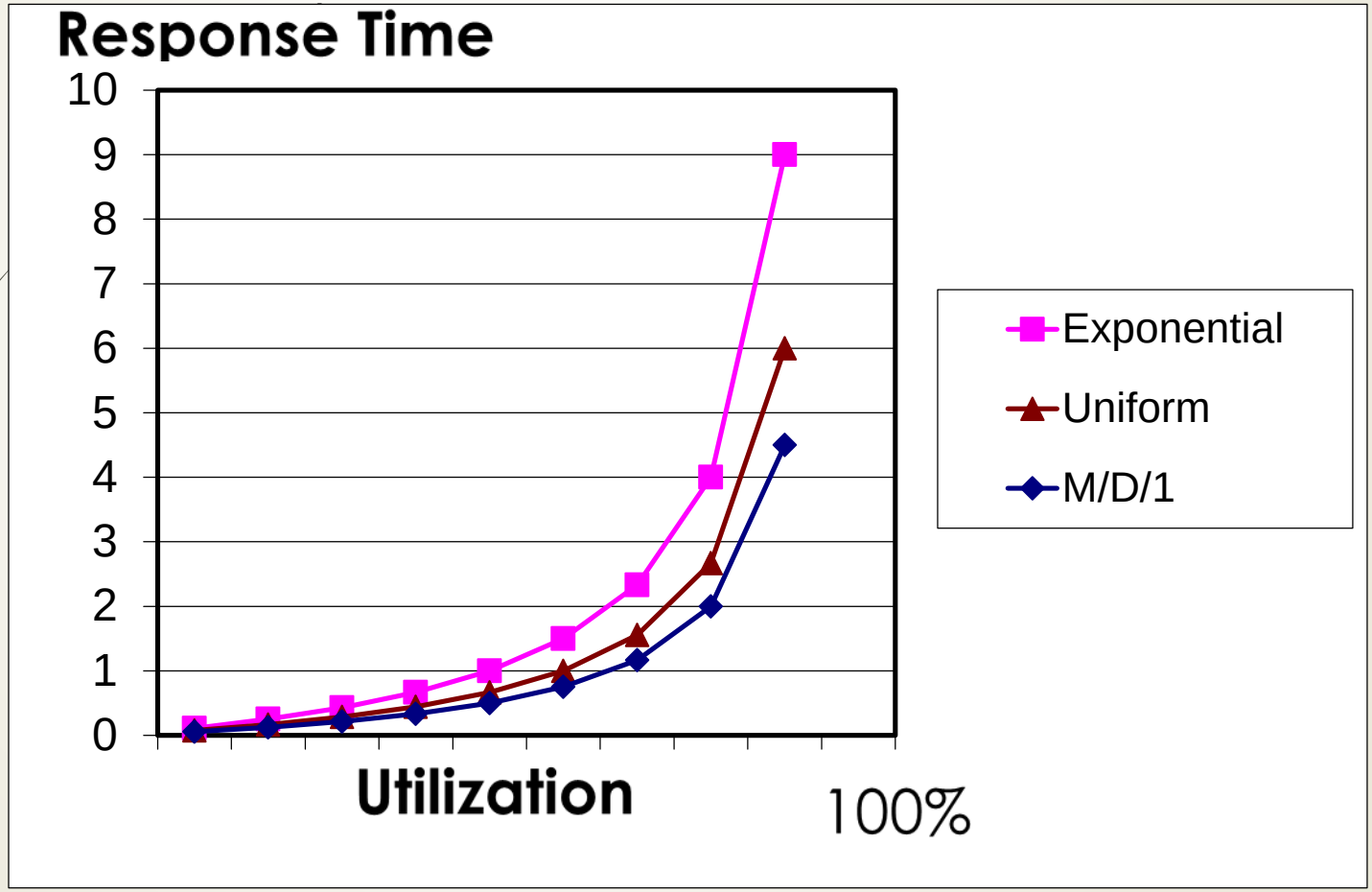
- **0.1 seconds** gives the feeling of **instantaneous** response — that is, the outcome feels like it was caused by the user, not the computer….
- **1 second** keeps the user's flow of thought **seamless**. Users can sense a delay, and thus know the computer is generating the outcome, but they still feel in control of the overall experience….
- **10 seconds** keeps the user's **attention**. From 1–10 seconds, users definitely feel at the mercy of the computer and wish it was faster, but they can handle it. After 10 seconds, they start thinking about other things….

https://www.nngroup.com/articles/website-response-times/

Response Time is a non-linear Function

- Linear
  - If one truck loader can load at the rate of 10 boxes a minute, five truck loaders can load a truck at the rate of 50 boxes a minute
- Non-Linear
  - If 5 transactions a second have a response time of 1 second, if you double the CPU power, 10 transactions a second will have a response time of 1 second

Non-Linearity of Response Time



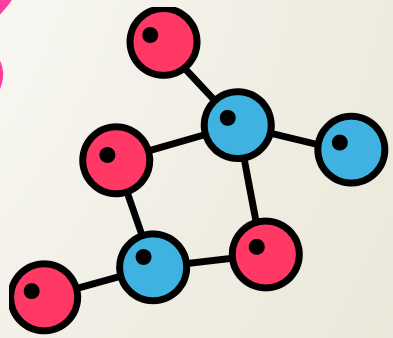
Confidence / Risk

- What is your confidence in your results?
- What is the risk involved?

What is a Computer Performance Model?

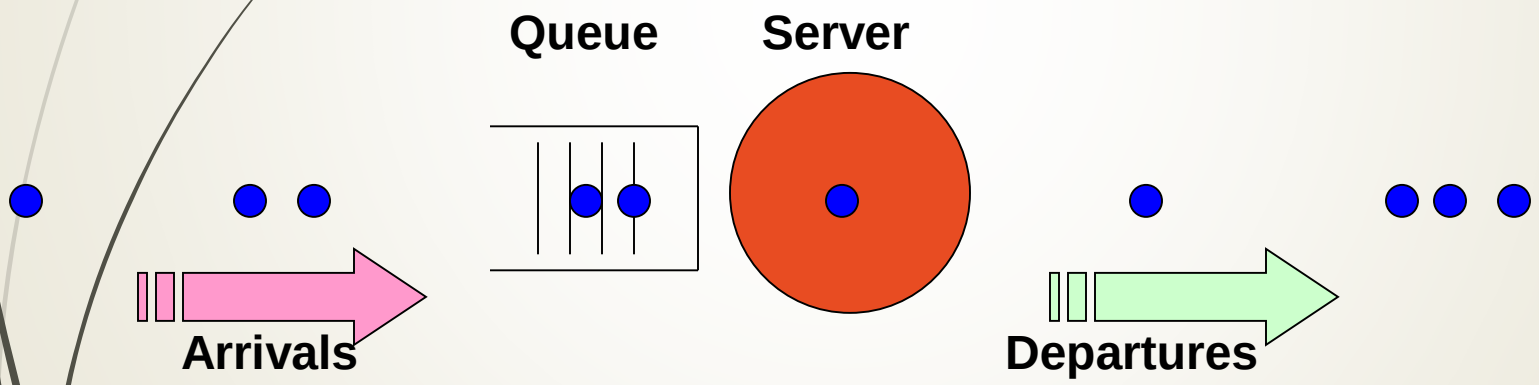
What is a Model?

- A model is an abstraction of a complex system that is constructed to observe phenomena that cannot be easily observed with the original system



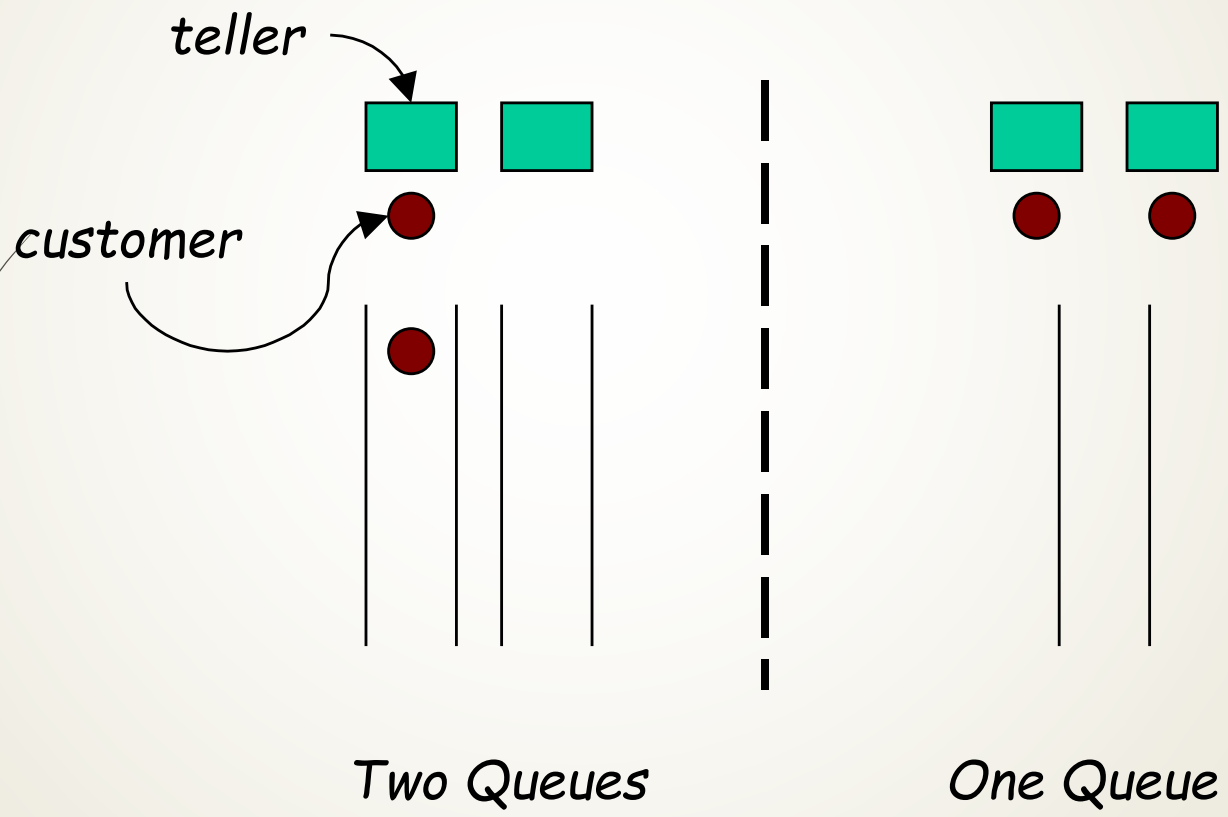
Computer Performance Model

- An Essential Building Block



Question – what is the nature of the arrival and departure pattern?

One versus Two Queues



The Example Workload

- Transaction-Oriented System
  - Average Transaction Arrival Rate and distribution
  - For Each Transaction:
    - Average CPU Time and distribution
    - Average number of overall storage I/Os
    - For Each Storage Device:

      Average number of I/Os Average Transfer Length

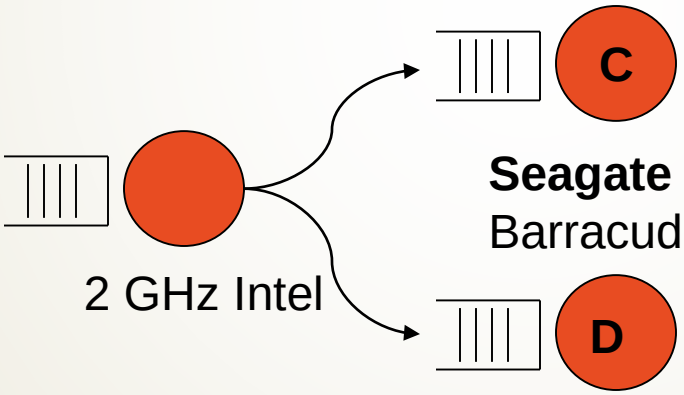
      Average % of Seeks Average Seek Time
- Example

  Checking transactions arrive at a rate of 2 / second. Each transaction uses, on average, 250 ms of CPU and requests, on average, 30 I/Os to disk “C:” For Disk C, the average transfer length is 16,000 bytes. 50% of the I/Os require a seek.

Question – have we measured and analyzed so that we can specify a workload in this manner?

The Example Configuration

- A Simple Configuration
  - One CPU Server with a “C” and “D” drive.



Seagate ST3160023AR
Barracuda (7200 RPM 160 GB Ultra ATA)

Modeling "What-If" Scenarios

- Performance models are constructed using a configuration and a workload.
- What-If Scenarios have two main areas of interest:
  - What if the workload changes?
  - What if the configuration changes?
- Examples:
  - Given the workload configuration, what if the transaction rate increases by 50%?
  - What if we replace Disk "C" with flash memory?
  - What if we decide to use a virtual server?

Question – can we accurately predict the effects of these changes?

What is a Computer Performance Model? Quiz

- A Model is the Intersection of a Workload and a Configuration. Both of these components are build using statistics that are collected and analyzed. Categorize the following statistics as “Workload” or “Configuration”
  - 5 transactions / second
  - 2 Mb Transfer rate
  - 7200 RPMs
  - 40 I/Os / transaction
  - Average transfer size

What is a Computer Performance Model? Quiz

- A Model is the Intersection of a Workload and a Configuration. Both of these components are build using statistics that are collected and analyzed. Categorize the following statistics as “Workload” or “Configuration”
  - 5 transactions / second — Workload
  - 2 Mb Transfer rate — Configuration
  - 7200 RPMs — Configuration
  - 40 I/Os / transaction — Workload
  - Average transfer size — Workload

Milestones in CPM
Let's Go to the Wayback Machine!

Probability Models
It's the LAW!

- Origins in 17th century gaming analysis
- There are Laws governing phenomena
- Statistics summarize a random sampling of the phenomenon
- Probability functions model the statistics

A. A. Markov

- Developed new branch of mathematics for probabilities
- Markov Chain - Chains of linked events where what happens next depends on the current state of the system
- Provides formulae to model
  - Queueing Systems
  - Reliability Models
  - Poetry Analysis

https://www.americanscientist.org/article/first-links-in-the-markov-chain

Agner Erlang

- He was a member of the Danish Mathematicians' Association through which he made contact with other mathematicians including members of the Copenhagen Telephone Company.
  - He went to work for this company in 1908 as scientific collaborator and later as head of its laboratory.
- Erlang applied probability models to analysis of telephone traffic and in 1909 published his first work on it "**The Theory of Probabilities and Telephone Conversations**“
- Empirical data followed Poisson's law of distribution for arrivals.
- He was often to be seen in the streets of Copenhagen, accompanied by a workman carrying a ladder, which was used to climb down into manholes.

William Feller

- He was the foremost probabilist outside of Russia.
- In the middle of the 20th century, probability was not generally viewed as a fruitful area of research in mathematics except in Russia, where Kolmogorov and others were influential.
- Contributed to the study of the relationship between **Markov chains** and differential equations. He wrote a two-volume treatise on probability that has since been universally regarded as one of the most important treatments of that subject.
- Provided necessary and sufficient conditions for the Central Limit Theorem – a limiting distribution (or LAW)

Leonard Kleinrock

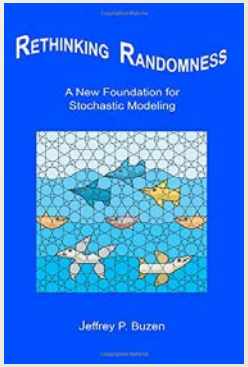
- “Queueing Systems “ (1975) - is the bible for knowledge on queueing theory
- Dr. Leonard Kleinrock created the basic principles of packet switching, the technology underpinning the Internet, while a graduate student at MIT. In this effort, he developed the mathematical theory of data networks.
- This was a decade before the birth of the Internet which occurred when his host computer at UCLA became the first node of the Internet in September 1969.
- He was listed by the Los Angeles Times in 1999 as among the "50 People Who Most Influenced Business This Century".

Arnold Allen

- IBM Systems Science Institute Instructor (1978)
- Invaluable in its clarity and simplicity of exposition
- Former CMG Member

CPM Specific Milestones

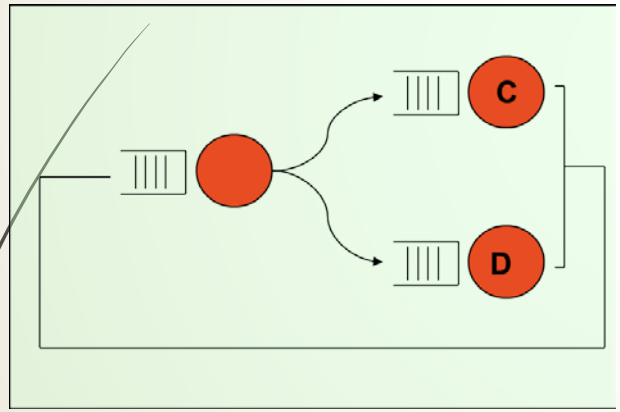
- 1957 – **Jackson** publishes an analysis of a multiple device system with parallel servers and jobs
- 1967 – **Gordon & Newell** simplified the models for “closed systems”
- 1967 – **Scherr** – used “**Machine Repairman**” problem to model an MIT Time sharing system
- 1971 – **Buzen** – introduces the “Central Server Model” and fast computational algorithms (and more..)



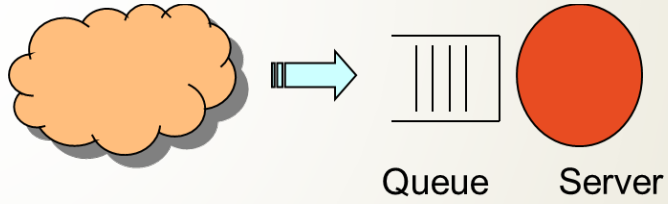
Foundational CPM Models

Open and Closed Networks

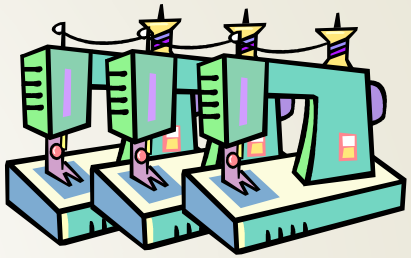
- Closed networks have a finite population



- Open Networks can theoretically have an infinite population

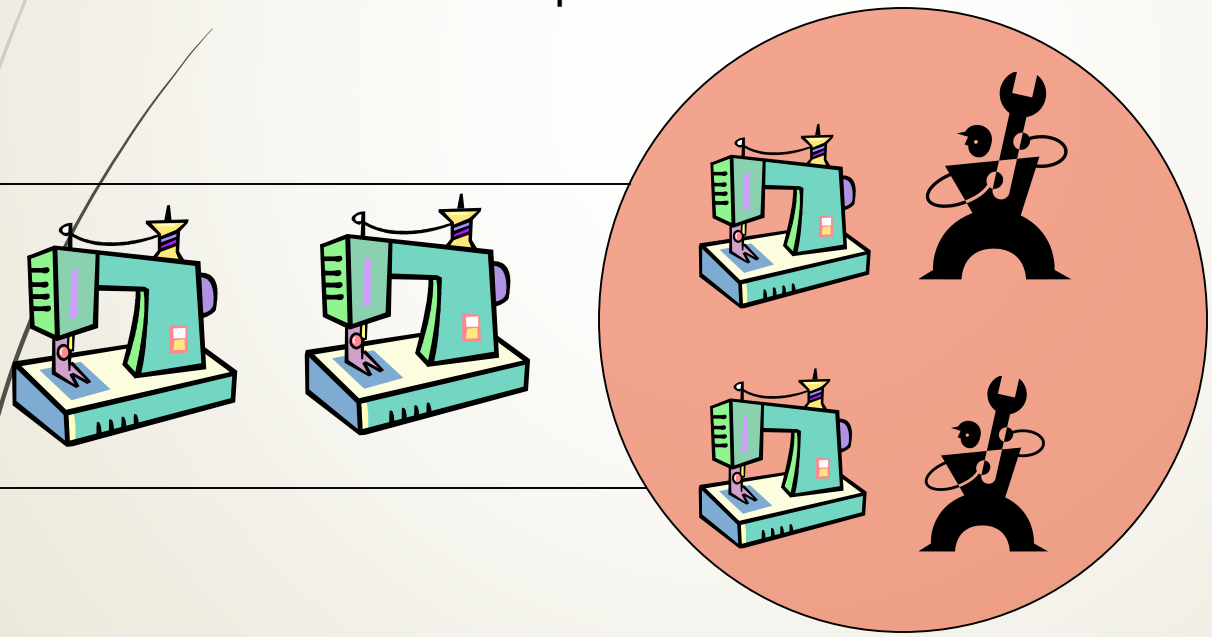


Machine Repairman Model



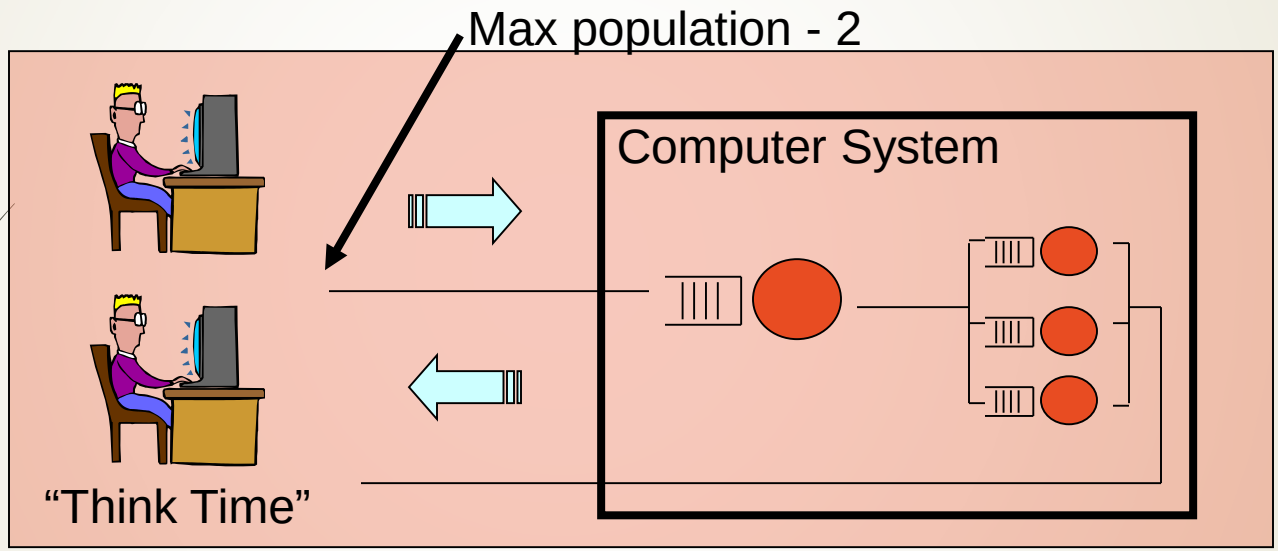
- Closed Networks have a finite population

K = 7 = machines
C = 2 = repairmen



Models for Closed Networks

- Scherr – used machine repairman problem to model a time-shared system



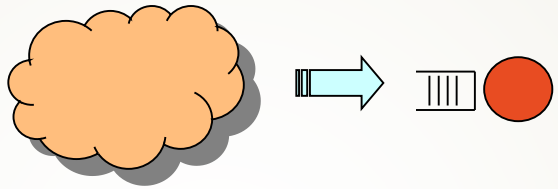
- Original Time-Sharing systems
- Batch workloads
- Multi-Threaded Applications

Kendall Notation

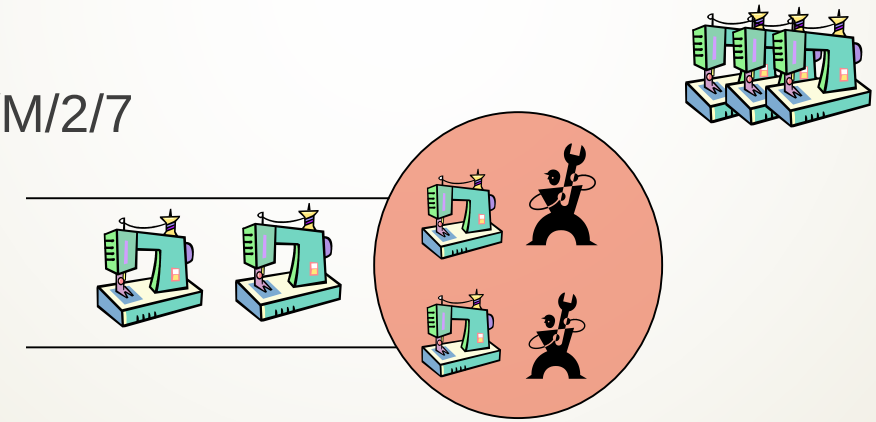
- A/B/c/k
- Distributions
  - A/ & B/ are:
    - GI – general independent interarrival time
    - G general service time distribution
    - Hk – k-state hyper exponential interarrival of service time distribution
    - Ek – Erlang-k interarrival or service time distribution
    - M – Exponential interarrival time or service time distribution
    - D – Deterministic (constant) interarrival or service time distribution
- Additional Parameters
  - /c – number of servers
  - /k – population count

Kendall Notation

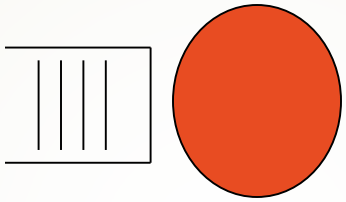
- M/M/1



- M/M/2/7



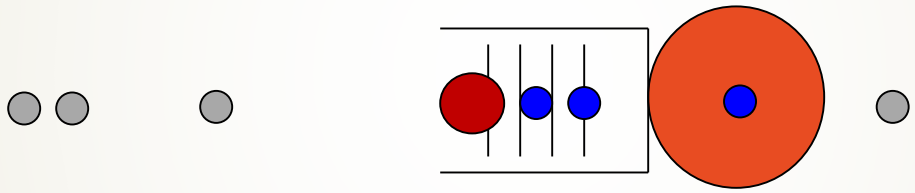
Essential Statistics



E[a]	expected arrival rate (λ)
E[s]	expected service time (1/μ)
E[w]	expected wait time
ρ	probability of being busy (x 100 = utilization)
L_q	length of queue
L	average number of “customers” in system (queue + service)
W_q	waiting time in queue
W	waiting time in the system (W_q +E[s])

Little's Law

- **The average number of objects in a steady-state queue/system is the product of the arrival rate and the average time spent in the queue/system.**
- In a "steady state" queueing system (completion rate = arrival rate):

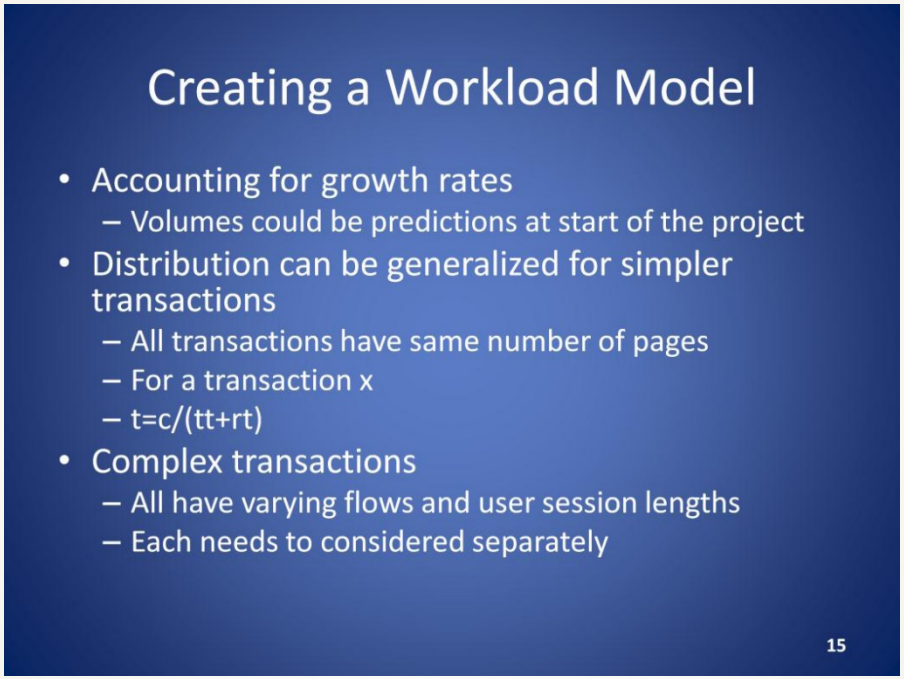


$$L = E[a] \bullet W$$

Example – iostats (Unix/Linux) and Windows performance monitor provide the average number of I/Os in the Disk queue and the arrival rate

Using Little's Law, you can estimate the average waiting time in the queue
Note – this is independent of *distribution*
See Buzen interview at http://ubiquity.acm.org/article.cfm?id=2986329

Real World Enterprise Workload Modeling, Priyanka Arora – CMG 2012

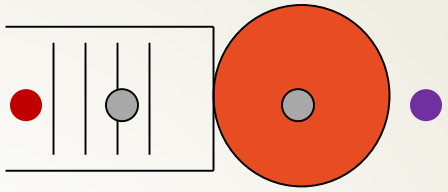


Where T is the throughput (tps) in page views per second
C is the concurrency
tt is the think time between pages in seconds ; rt is response time

$$T = c/(tt+rt) \Leftrightarrow E[a] = L / w \Leftrightarrow L = E[a] * w$$

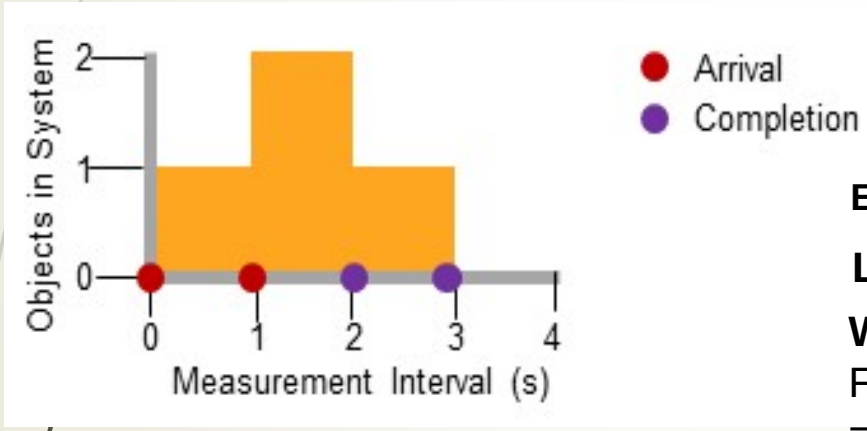
Little's Law

$$L = E[a] \bullet W$$



Observational Stochastics from Rethinking Randomness (Buzen)

Buzen Trajectory Model



Time	Objects
0-1	1
1-2	2
2-3	1
3-4	0

E[a] = 2 arrivals/4 seconds = **.5/s**

L = (1+2+1+0)/4 = **1**

W = Avg Wait time (s)
For (A_1, A_2) either: $(2_1, 2_2)$ or $(3_1, 1_2)$
= **2s**

$$1 = .5\frac{arrival}{s} \bullet 2\frac{s}{arrival}$$

Pop Quiz!!!! What is E[s]?

What is E[s]?

E[a]	expected arrival rate (λ)
E[s]	expected service time (1/μ)
E[w]	expected wait time
ρ	probability of being busy (x 100 = utilization)
L_q	length of queue
L	average number of "customers" in system (queue + service)
W_q	waiting time in queue
W	waiting time in the system (W_q +E[s])

Modeling & Forecasting

- E[s] is average service time per arrival
- We know Utilization and E[a] (Arrivals / sec)
- Utilization = E[a] *E[s] => E[s] = Utilization / E[a]
- Utilization = 3 sec / 4 seconds = 75% (=> ρ = .75)
- The server was busy for 3 seconds
- There were 2 arrivals
- E[s] = 1.5 s

Modeling Response Time
It's all a matter of probabilities

CPM is driven from Operating System Metrics

- You do not often get measurements of number of objects in the system
- ALL operating systems measure utilization arrival rates
  - sometimes at the transaction / software level
  - Always at the hardware level
- E[s] and E[a] are sufficient to model response time as long as certain basic assumptions are met
- These assumptions are made at a probabilistic level

Discrete Distribution

Uniform

p.d.f

$$f(x) = P[X = x]$$

c.d.f

$$F(x) = P[X \leq x]$$

for $x = k_1, k_2, \ldots k_n$

$$f(x) = \frac{1}{n}$$

$$F(x) = \frac{x}{n}$$

$$E[X] = \sum_{i=1}^{6} F(i) = \frac{1}{6}\sum_{i=1}^{6} i = 3.5$$

Continuous Distributions

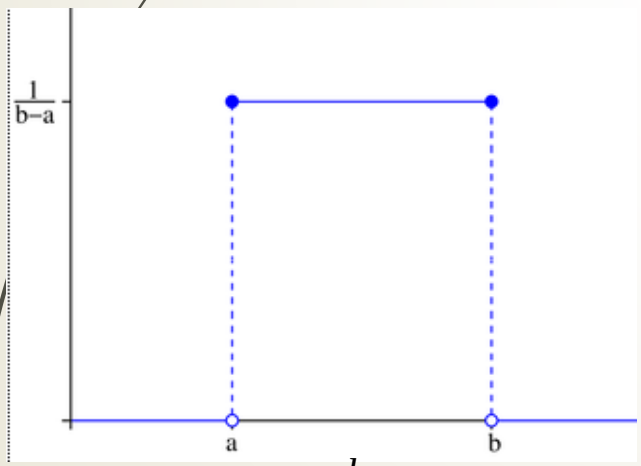
Uniform

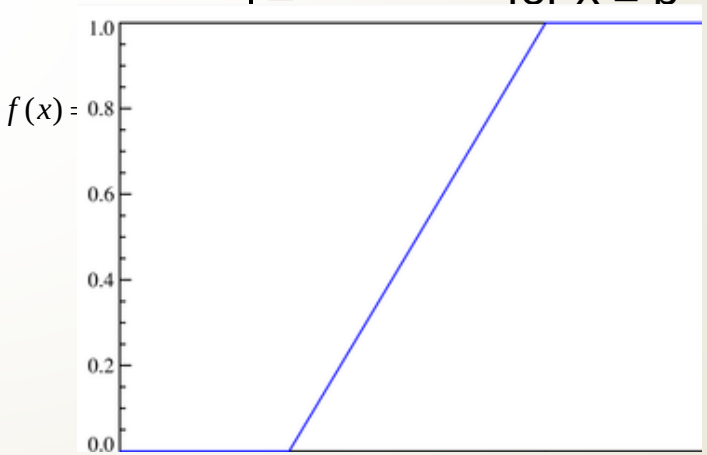
$$f(x) = P[X = x]$$

$$F(x) = P[X \le x]$$

$$f(x) = \frac{1}{b-a}$$

$$F(x) = \begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x < b \\ 1 & \text{for } x \ge b \end{cases}$$

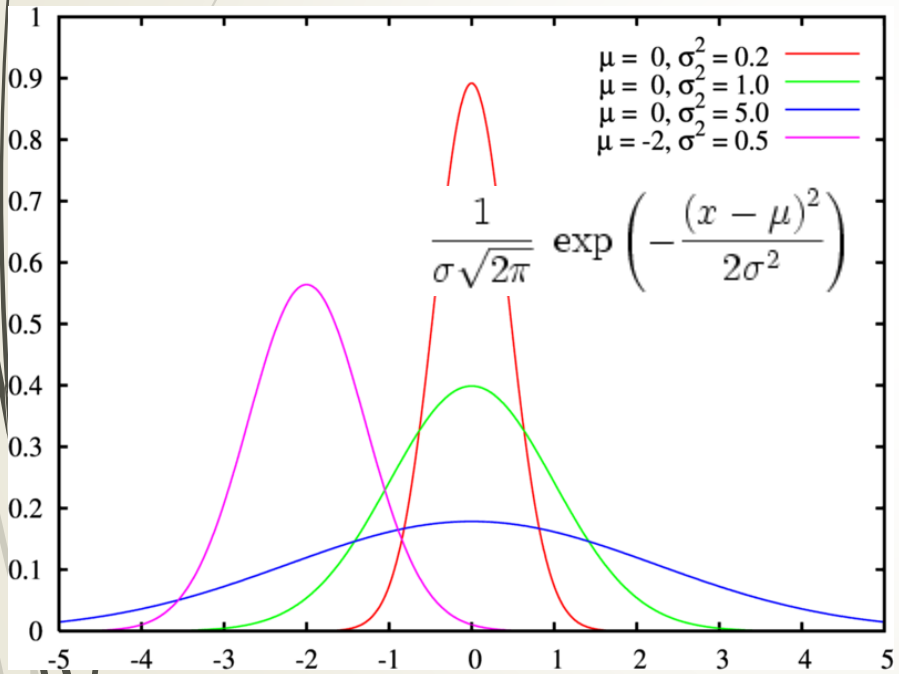


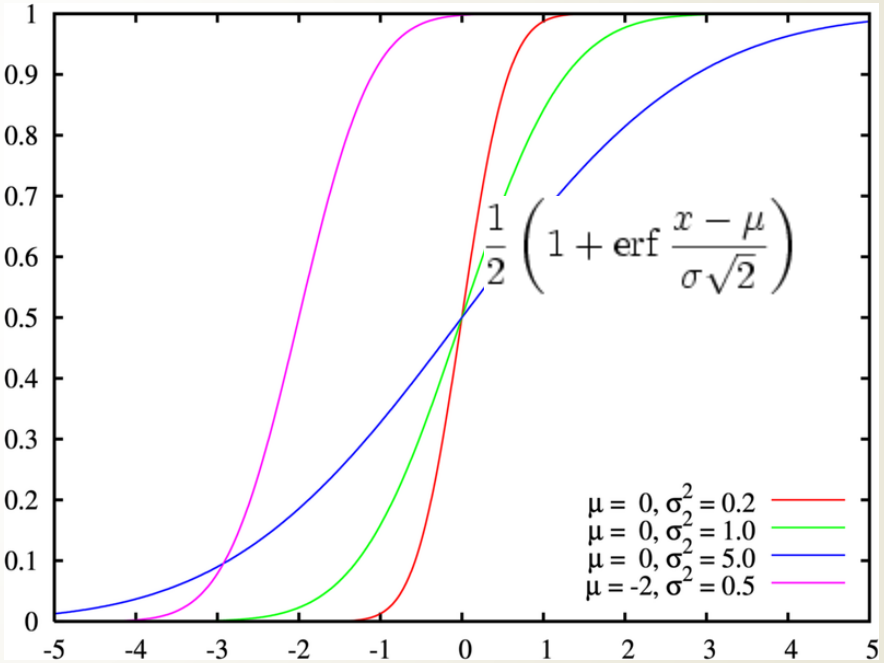


$$E[x] = \int_{x-a}^{b-x} x \bullet f(x)dx = \int_{x-a}^{b-x} X \bullet dF = \frac{b-a}{2}$$

Continuous Distributions
Normal

- The “Normal” or “Gaussian” Distribution or “Law”
  - This is a “limiting distribution”





$$f(x) = P[X = x]$$

$$F(x) = P[X \le x]$$

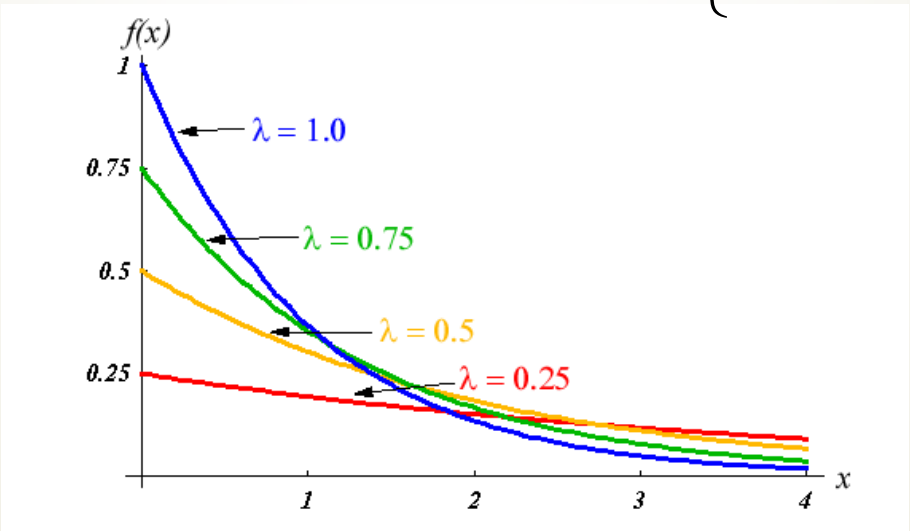
$$E[x] = \mu$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt.$$

Continuous Distributions

Exponential

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{for } x \le 0 \end{cases} \qquad F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{for } x \le 0 \end{cases}$$



$$E[x] = \lambda$$

The Memoryless Property

Only the **exponential distribution** has the memoryless property: The probability of an occurrence after time s is equal to the probability of an occurrence at time s+t

$$P(X > s+t) \mid X > t) = P(X > s) \qquad \text{for s, t > 0}$$

$$P(X > s+t) \mid X > t) = \frac{P(X > s+t, X > t)}{P(X > t)}$$

$$= \frac{P(X > s+t)}{P(X > t)}$$

$$= \frac{e^{-\lambda(s+t)}}{e^{-\lambda t}}$$

$$= e^{-\lambda s}$$

$$= P(X > s)$$

The Inspection Paradox

- (Feller) Taxis pass a specific corner with an average time of 20 minutes between taxis. You walk up at a random time. How long will you wait for the taxi?

- Answer – 20 minutes

The Poisson Process

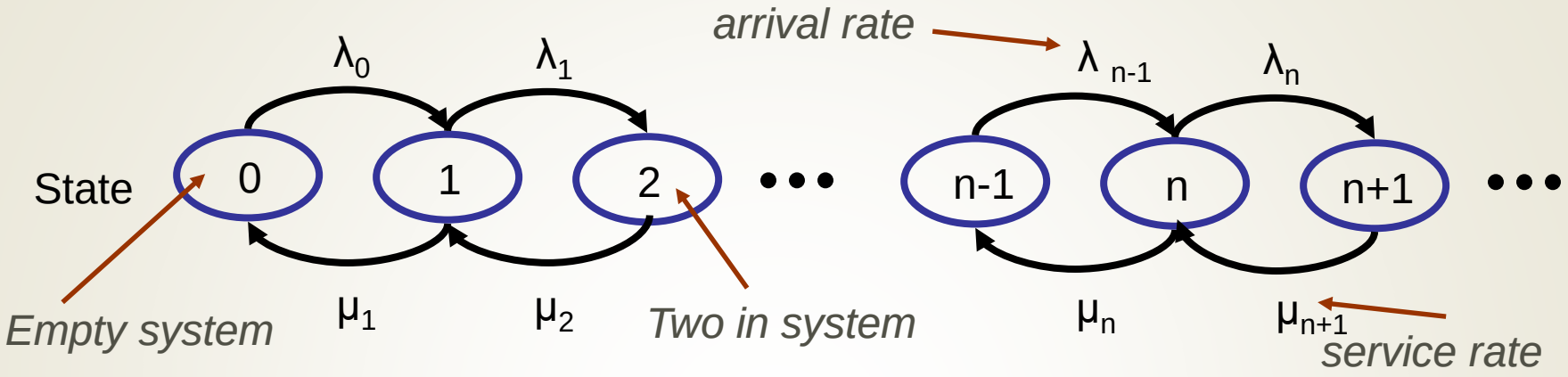
- Let (N(t), t ≥ o} be a Poisson process with rate
- The random variable Y describing the number of events in any time interval of length t > 0 has a Poisson distribution with parameter λt

$$P[Y = k] = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \quad \text{K = 0, 1, 2, ...} \quad \lambda > 0$$

- Key attribute – the interarrival times are exponentially distributed
- Memoryless – the number of arrivals occurring in any bounded interval of time t is independent of the number of arrivals occurring before time t

Markov Birth-Death Model

M/M/1



$$\sum_{n=0}^{\infty} p_n = p_0 + p_1 + p_2 + \ldots = 1$$

$$p_n \mu_n = p_{n-1}\lambda_{n-1}; n = 2,3,\ldots \Rightarrow p_n = p_{n-1}\frac{\lambda_{n-1}}{\mu_n}$$

$$p_0\left(1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0\lambda_1}{\mu_1\mu_2} + \ldots + \frac{\lambda_0\lambda_1\ldots\lambda_{n-1}}{\mu_1\mu_2\ldots\mu_n} + \ldots\right) = 1$$

$\lambda_0 = \lambda_1 = \lambda_2 \ldots$

$\mu_0 = \mu_1 = \mu_2 \ldots$

MEMORYLESS

$$\rho = \lambda / \mu = E[a] \bullet E[s]$$

$$\rho < 1 \Rightarrow S = \left(1 + \rho + \rho^2 + \rho^3 \ldots\right) < \infty$$

$$S = 1/(1-\rho)$$

$$p_0 S = 1 \Rightarrow p_0 = (1-\rho)$$

$$p_n = P[N = n] = (1-\rho)\rho^n$$

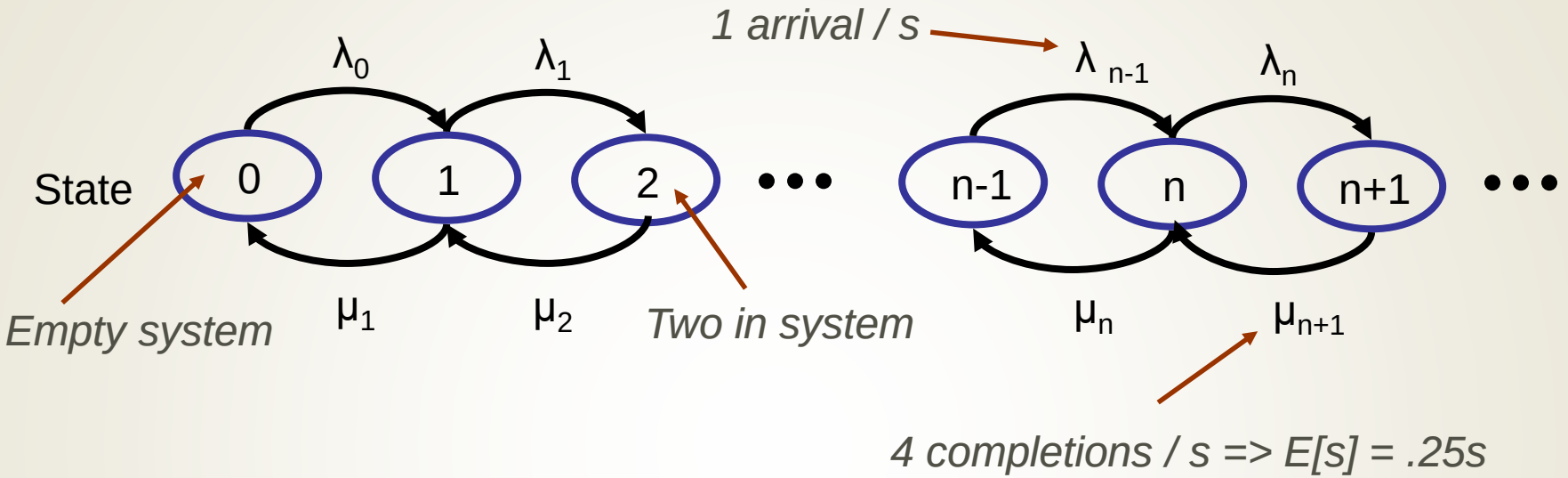
$$L = E[N] = \rho/(1-\rho) \text{ (geometric pdf)}$$

Applying Little's Law…

$$W = E[w] = L/\lambda = E[s]/(1-\rho)$$

Markov Birth-Death Model

M/M/1



$$\rho = \lambda / \mu = E[a] \bullet E[s] = 1/4 = .25$$

$$W = W_q + E[s]$$

$$W = E[s]/(1-\rho)$$

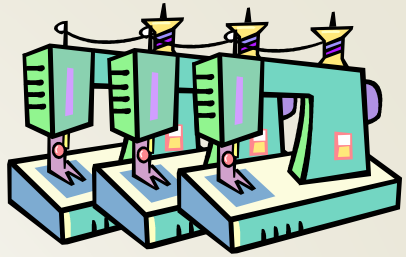
$$W_q = E[s] \bullet \frac{\rho}{(1-\rho)}$$

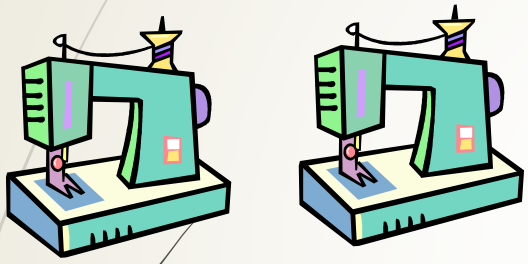
$$L = E[N] = \rho/(1-\rho) = .25/.75 = 1/3$$

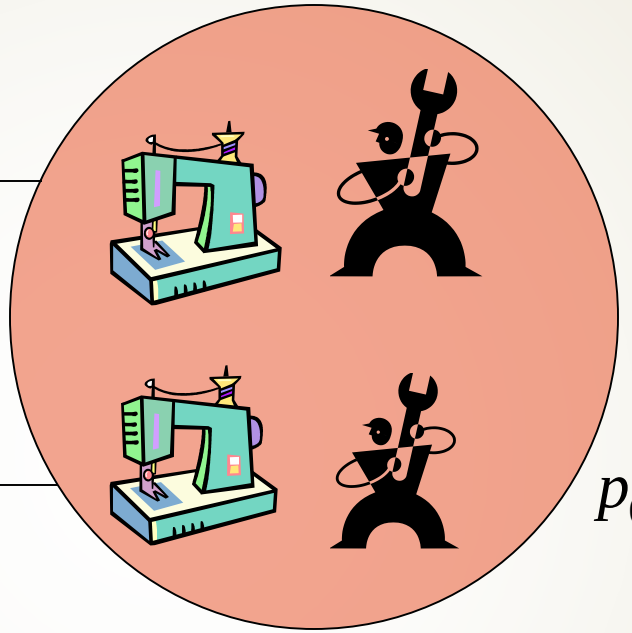
$$W = E[w] = L/\lambda = E[s]/(1-\rho) = .333s$$

$$W_q = .25/3$$

M/M/2/7







$E[O] = 1/\alpha$ - Avg operation time

$E[s] = 1/\mu$ - Avg repair time

$p_n = P[N = n]$

$$L_q = \sum_{n=c+1}^{K} (n-c)\, p_n$$

$$W_q = L_q (E[O] + E[s]) / (K - L_q)$$

$$p_0 = \left[1 + \sum_{k=1}^{K} \left(\frac{p_k}{p_0} \right) \right]^{-1}$$

$$\frac{p_n}{p_0} = \begin{cases} \binom{K}{n} \left(\frac{E[s]}{E(O)} \right)^n & n = 1, 2, \ldots, c \\ \frac{n!}{c!\, c^{n-c}} \binom{K}{n} \left(\frac{E[s]}{E(O)} \right)^n & n = c+1, \ldots, K \end{cases}$$

Here's the Short Cuts....

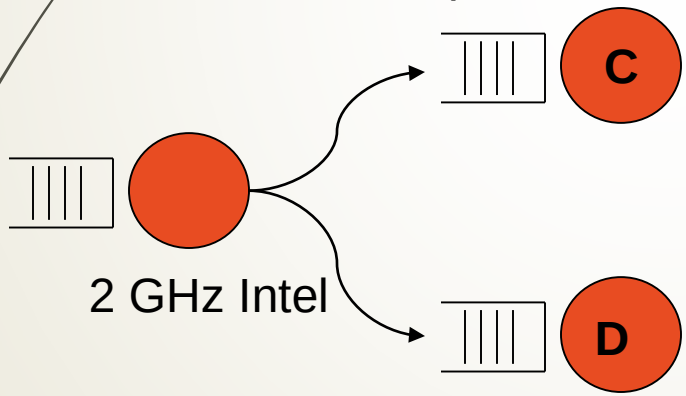
- See Arnold Allen's Book – Appendix C (either edition)
- Modeling Packages
  - Integrated with workload monitoring
  - Translates raw statistics to statistics that can be used for modeling
  - Visualizes modeled results

Back to The REAL Model

When theory is too theoretical

Back to our Model Example

Checking transactions arrive at a rate of **2 / second**. Each transaction uses, on average, **250 ms** of CPU and requests, on average, **30 I/Os** to disk "C:" For Disk C, the average transfer length is **16,000 bytes**. **50%** of the I/Os require a seek.



Seagate ST3160023AR
Barracuda (7200 RPM 160 GB Ultra ATA
Avg Seek Time = **8.5 ms**
Avg Transfer Rate = **58 Mb/s**
Avg Rotation = 60 sec / 7200 revolutions
= **8.3 ms**/ revolution
Avg Latency = **4.15**

Back to our Example
The Central Server Model

$E[a]=2 / s$

$E[s]_C$ = Seek + Latency + Transfer

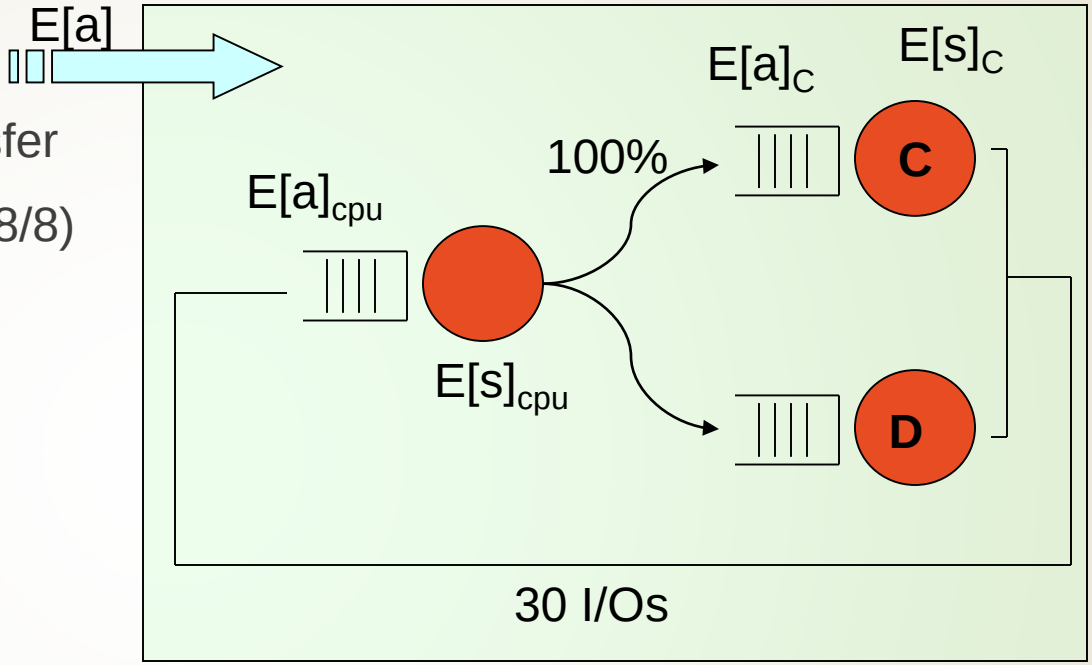
= (.5*8.5)+ 4.15 + .016/(58/8)

= (4.25 + 4.15 + 2.2)

= 10.6ms

$E[a]_{cpu}$ = 2*30 = 60/s

$E[s]_{cpu}$ = 250 / 30 = 8.333 ms

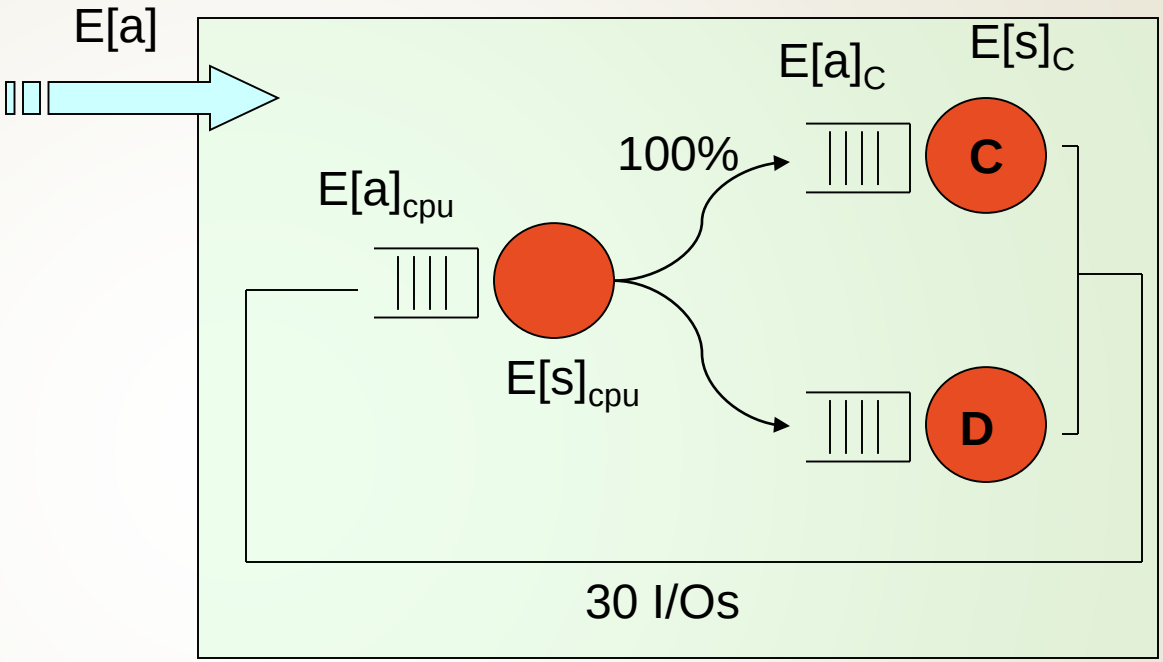


Utilization (%) = ρ*100

ρ_{CPU} =(2/ s)*250 ms = 500 ms/s = 50%

ρ_C = (2*30/s)*10.6 ms = 636ms /s = .636 = 63.3%

Back to our Example
The Central Server Model



$E[s]_{CPU}$=8.333

$E[s]_C$=10.6

ρ_{CPU} =(2/ s)*250 ms = 500 ms/s =.5

ρ_C = 636 / 1000 = .636

W_{CPU}= E[s]/(1-p) = 16.66

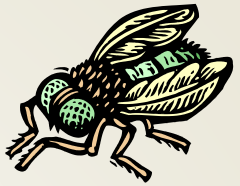
W_C= E[s]/(1-p) = 29.12

Transaction Time = 30*(16.66+29.12) = 1373.6 ~ **1.4 seconds**

What Next?

- Validate the Model with real observations
  - There could be other effects, such as software queueing
- Now you are ready for the fun part – “What IF”?
  - The transaction rate doubles? (E[a] increases)
  - The CPU is upgraded to a faster model ($E[s]_{CPU}$ decreases)

Fly in Ointment



- For disk I/Os, the random variables are Uniformly distributed, not exponentially distributed – not M/M/1
- M/G/1 – Poisson process with general distribution
  - For All Distributions – $W = E[s] + W_q$
  - For M/G/1

$$W_q = E[s]\frac{\rho}{1-\rho}\left[\frac{1+C_s^2}{2}\right]$$

$$C_s^2 = \frac{Var[s]}{E[s]^2}$$

$$Var[s] = E[S^2] - E[S]^2$$

Variance and Queueing

Exponential Distribution

$$f(x) = \lambda e^{-\lambda x}$$

$$E[x] = \frac{1}{\lambda}; Var[x] = \frac{1}{\lambda^2}$$

$$C_s^2 = 1$$

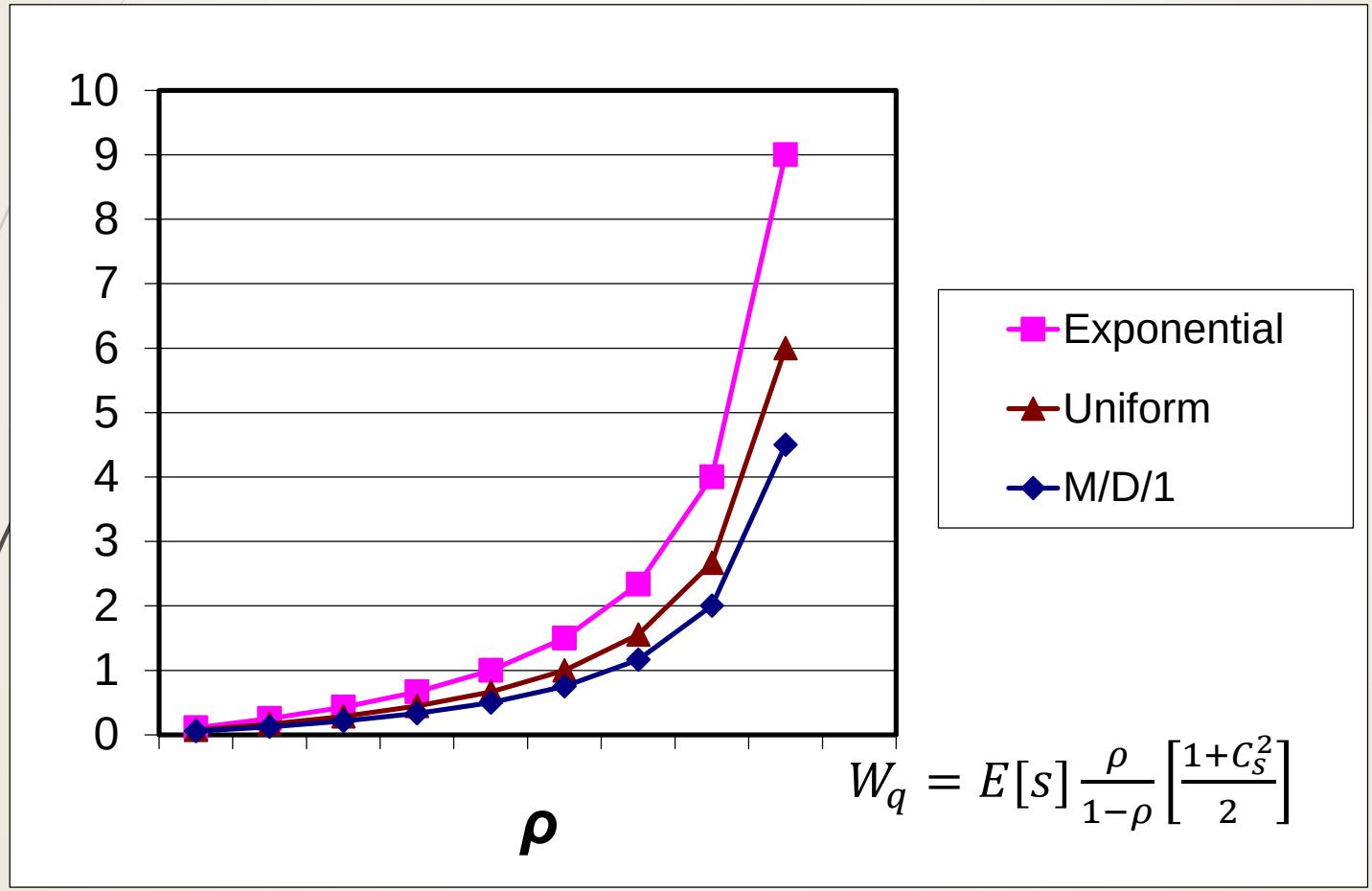
Uniform Distribution

$$f(x) = \begin{cases} \frac{1}{b-a} \\ 0 \end{cases}$$

$$E[x] = \frac{b-a}{2}; Var[x] = \frac{(b-a)^2}{12}$$

$$C_s^2 = 1/3$$

Estimated Waiting Time in Queue - M/G/1



Adding Accuracy to our Example

- M/M/1

$$30\left(16.66 + 10.6 + 10.6\frac{.636}{(1-.636)}\right)$$

$$30(16.66 + 10.6 + 18.5)$$

30 (16.66 + 29.12)

=1. 3736 seconds

- M/G/1

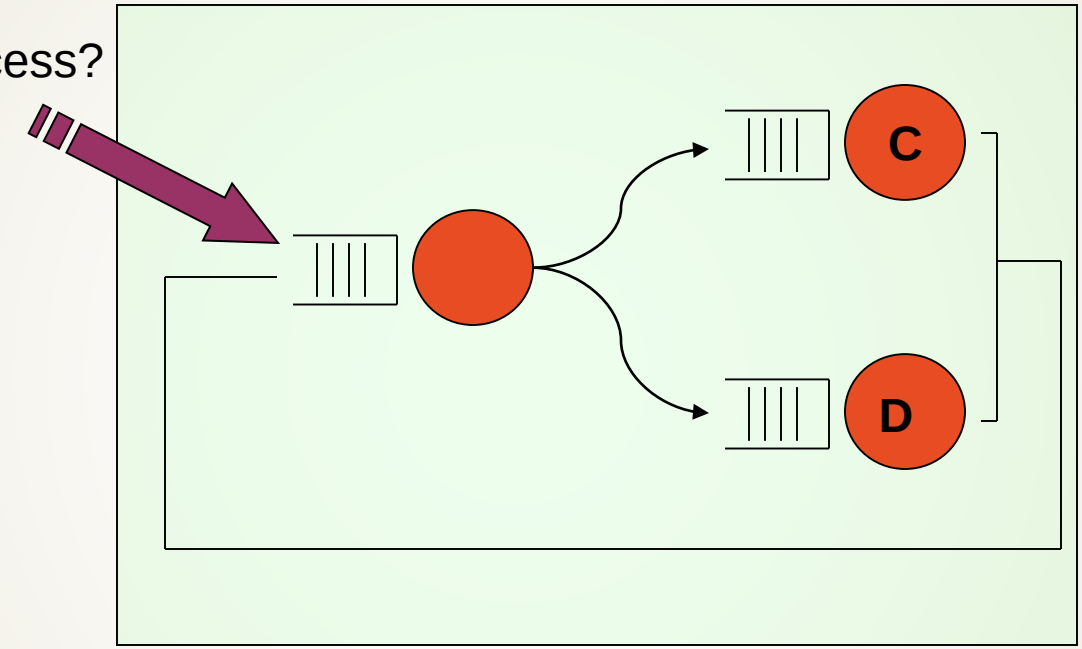
$$30\left(16.66 + 10.6 + 10.6\frac{.636}{(1-.636)}\left[\frac{1+\frac{1}{3}}{2}\right]\right)$$

$$30(16.66 + 22.947)$$

=1. 188 seconds

Another Fly…

Poisson Process?



CPM using Simulation

Modeling Using Simulation

- Simulation is not a four letter word
  - In the past, simulation was considered too CPU intensive
- Simulation emerged as a viable alternative in the 1990s
  - Fast, cheap CPU cycles (SimCity)
  - Networking protocols violate a number of traditional assumptions
  - These protocols are easily specified
- Various packages are available, from inexpensive (roll your own) to highly sophisticated, with animation
- Best reference – Averill Law

Back to the “Wayback”..

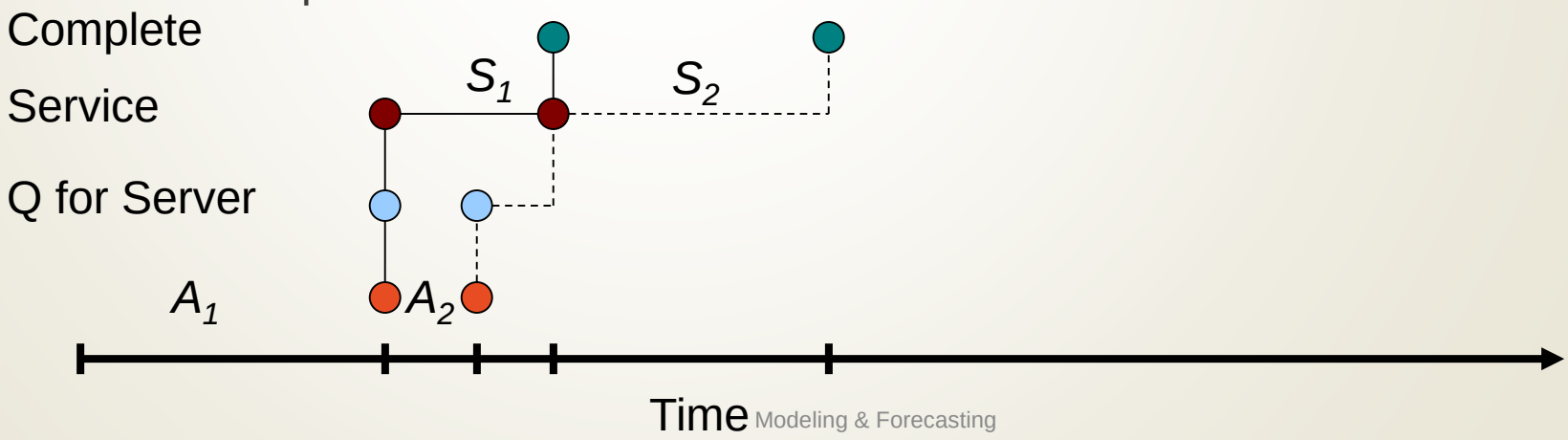
- 1944 – Manhattan Project – John Von Neumann’s simulations of hydrodynamical systems and “Monte Carlo” techniques

  http://cms.sjtu.edu.cn/doc/reading/cms/Coming_of_Materials_Science_Chap12_(Cahn).pdf

- 1960 - GPSS developed by Geoffrey Gordon, IBM –FORTRAN-Like language
- 1961 – Simula I introduced by K. Nygaard & O. Dahl
  - It used Algol and extensions to express systems with parallel processes
  - Simula 67 is considered to be the first object-oriented language
- 1963 – Simscript – COBOL-like programming language
- 1980 – MacDougall introduces smpl – C-like language
- 1986 – Schwetman - Introduces CSIM – C-like langauge

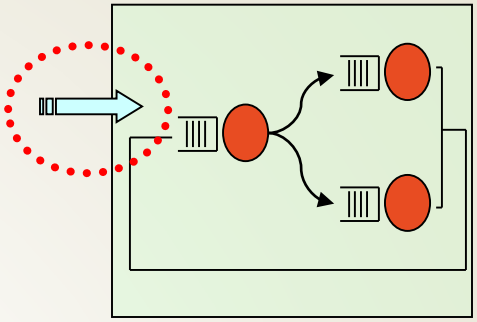
Discrete Event Simulation

- Simulated time
  - Clock progresses from one event to the next – it can skip individual time steps
- Parallel Processes
  - The language specifies independent processes that proceed in time



A CSIM Example

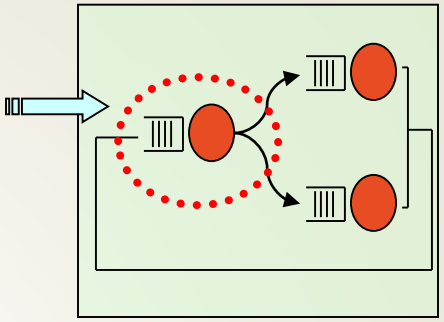
Generate Transactions



```
void generateTransaction()
/* This is a separate process that creates transactions */
{
        int i;
        double interArrivalTime;

        create("GenTrans");
        interArrivalTime = 1000.0/EATransaction;

            for(i = 1; i <= TotalTransactions; i++) {
              hold(exponential(interArrivalTime));
              ProcessTrans();
        }
        set(done);                                  /* signal
"done" event */

}
```

A CSIM Example

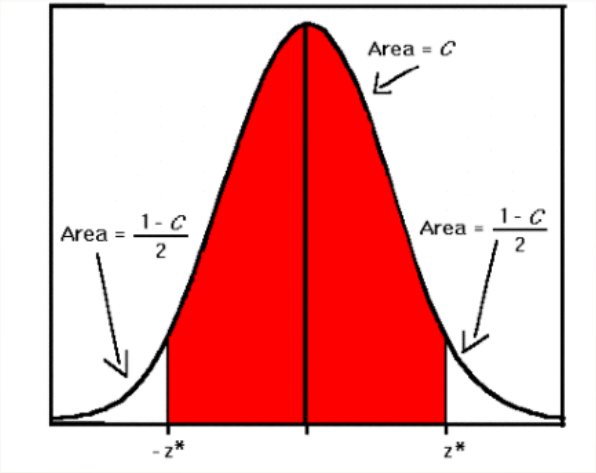
Process Transaction



```
void ProcessTrans()
/* Each time this is called, an independent process is created that
   then requests (and competes for) facilities */
{
        TIME t1;
        double cpuRequestperIO;
        long iosThisTrans;
        int ioCount;

        /* Initialize Transaction */
        create("ProcessTrans");
        t1 = simtime();
        cpuRequestperIO = EACPU / IOsPerTrans;
        iosThisTrans = (int) uniform(0.0,60.0);/* pick number of I/Os from uniform
distribution with mean of 30 */

        /* Iterate for number of I/Os in transaction */

        for (ioCount = 1; ioCount <= iosThisTrans; ioCount++) {
                reserve(cpu);                   /* reserve cpu */
                hold(exponential(cpuRequestperIO));     /* hold cpu  */
                release(cpu);                   /* release cpu */
                SimulateIO();                   /* simluate an I/O */
        }
        tabulate(totalTransTime, simtime()-t1);                 /* elapsed time for the
transaction */
} /* end of ProcessTrans */
```

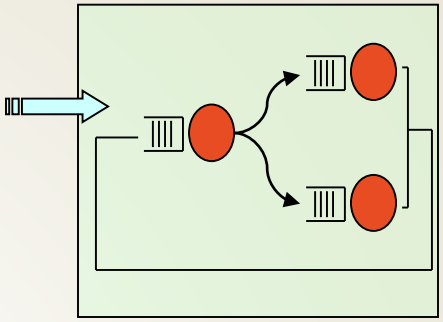
Evaluation of Simulation Statistics

- A Simulation shows the results of selecting random numbers from various distributions
- The resulting statistics must be evaluated to take the randomness into consideration



- Note – These must be *independent* samples.

CSIM Example



Ending simulation time: 5015467.206
Elapsed simulation time: 5015467.206
CPU time used (seconds): 1.522

FACILITY SUMMARY

Facility name	service	service time	util.	through-put	queue length	response time
cpu	fcfs	8.32557	0.487	0.05854	0.84592	14.45151
diskc	fcfs	10.60914	0.621	0.05853	1.57504	26.90791

M/M/1 Results

$E[s]_{CPU}$=8.33
$E[s]_C$= 10.6
U_{CPU} =(2/ s)*250 ms = 500 ms/s =.5
U_C = 636 / 1000 = .636

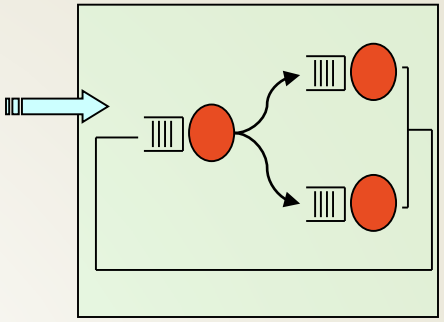
W_{CPU}=16.66
W_C= 29.12

Utilization (%) = ρ*100

ρ $_{CPU}$ =(2/ s)*250 ms = 500 ms/s = 50%

ρ $_C$ = (2*30/s)*10.6 ms = 636ms /s = 63.6%

CSIM Example



Note – CSIM evaluates autocorrelation and then adjusts the batch sizes

```
confidence intervals for the mean after 9900 observations

level                confidence interval                          rel. error

 90 %    1216.357672 +/- 49.942077 = [1166.415594, 1266.299749]    0.042817
 95 %    1216.357672 +/- 59.684293 = [1156.673379, 1276.041965]    0.051600
 98 %    1216.357672 +/- 71.126979 = [1145.230693, 1287.484651]    0.062107
```

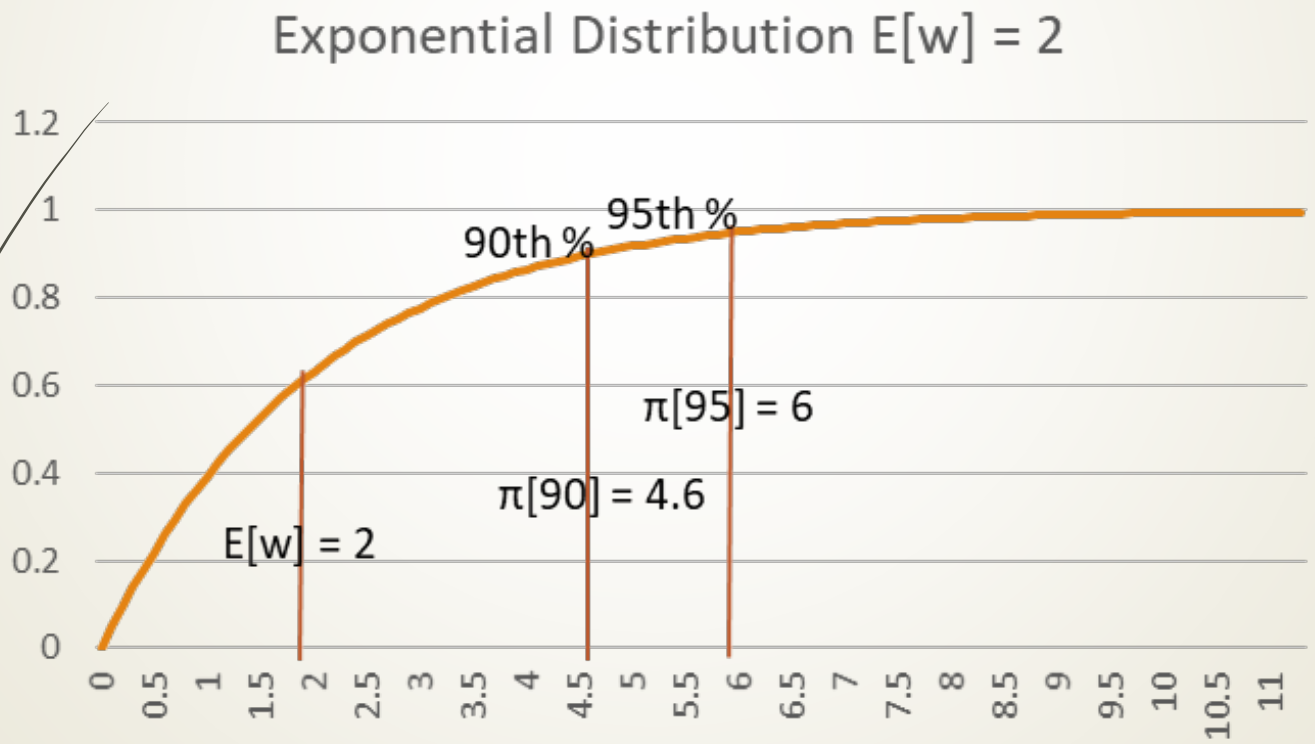
0.08 of the mean

Analytic Results

M/M/1	30*(16.66+29.12) = 1373.6	~ 1.4 seconds
M/G/1	30*(16.66+22.95) = 1188.2	~ 1.2 seconds

Simulation to Assess Risk in Estimating Percentiles

"Rules of Thumb for Response Time Percentiles -How Risky Are They?"
Michael Salsburg, Jeff Buzen



$$90\%R = \frac{\pi[90]}{E[w]}$$

M/M/1 and M/G/1formulae for estimating percentiles

For M/M/1, it can be shown that

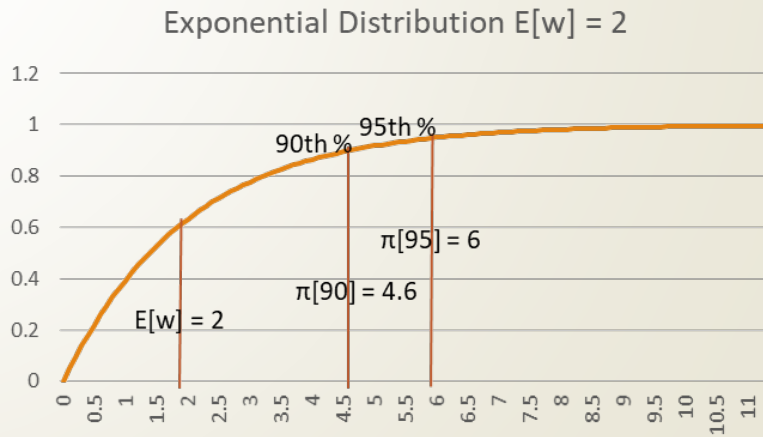
$$P[X \le \pi[90]] = .9 \Rightarrow 1 - e^{-\alpha\pi[90]} = 0.9 \Rightarrow e^{-\alpha\pi[90]} = 0.1 \Rightarrow$$

$$\pi[90] = -\frac{\ln(0.1)}{\alpha} = E[X]\ln(10) = 2.3E[X] \Rightarrow 90\%R \approx 2.3$$

Similarly, $95\%R \approx 3.0$

For M/G/1, Martin's Estimate

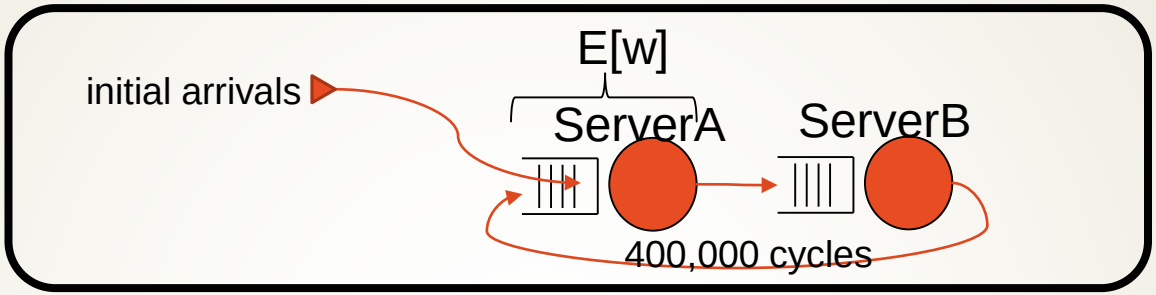
$$\pi[90] \approx W + 1.3\sigma_w \Rightarrow 90\%R = 1 + 1.3CV$$
$$\pi[95] \approx W + 2.0\sigma_w \Rightarrow 95\%R = 1 + 2.0CV$$



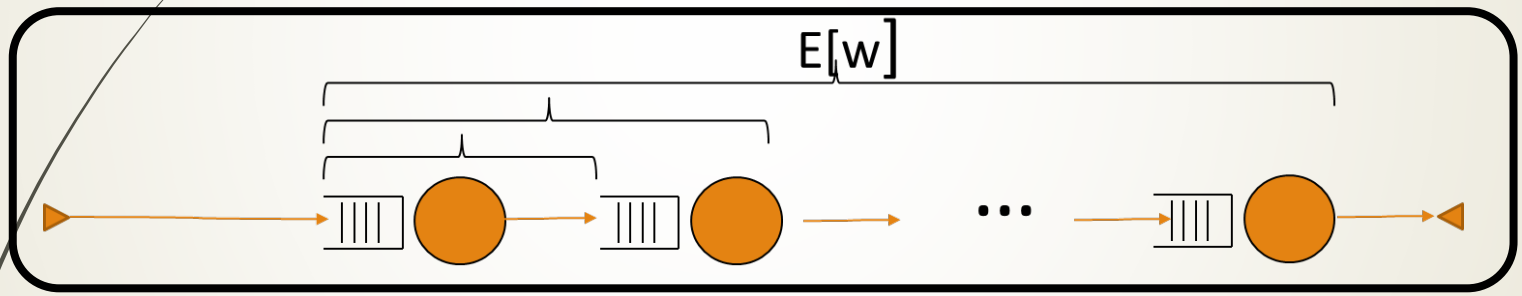
$$90\%R = \frac{\pi[90]}{E[w]}$$

Modeling Three Types of Queueing Systems

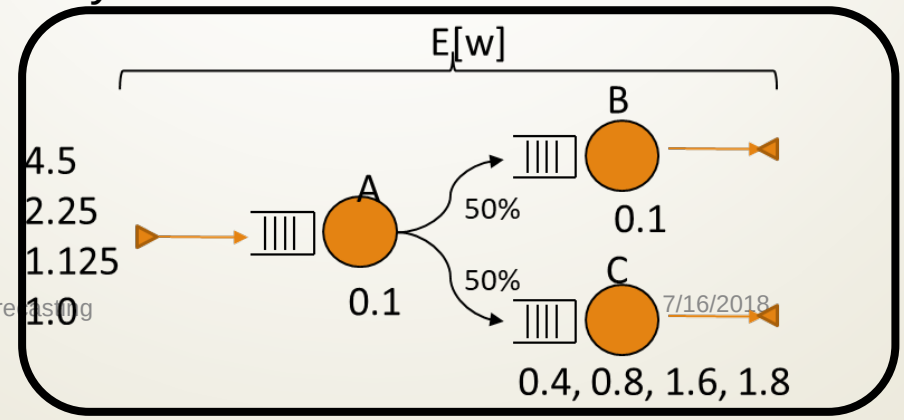
Central Server



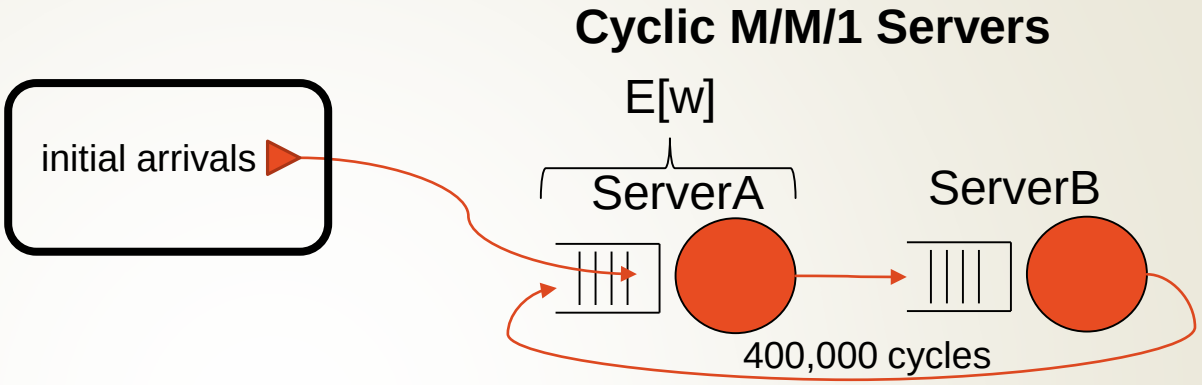
Tandem Servers



Asymmetric



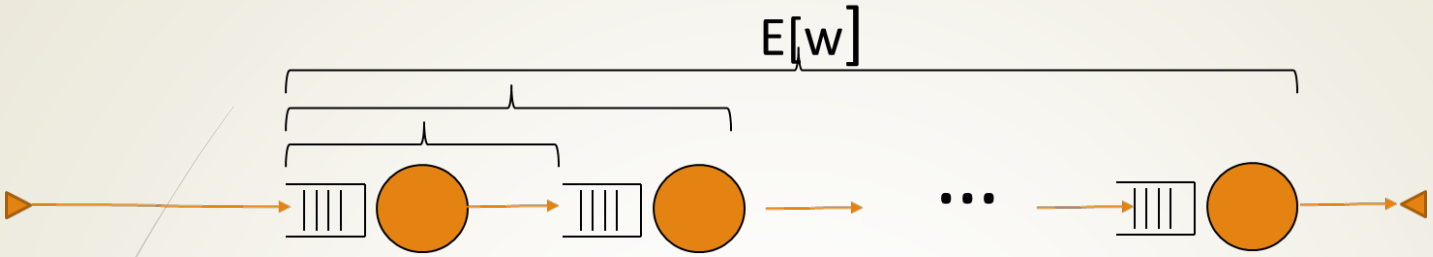
Central Server (Closed Network)



Customers	Average Service Time – ServerB						
	1	**1.25**	**1.5**	**1.75**	**2**	**4**	**8**
2	2.19	2.20	2.22	2.24	2.25	2.28	2.29
4	2.05	2.15	2.22	2.24	2.27	2.29	2.30
8	1.96	2.18	2.29	2.30	2.32	2.30	2.29
16	1.88	2.29	2.32	2.31	2.31	2.30	2.29

90%R Values

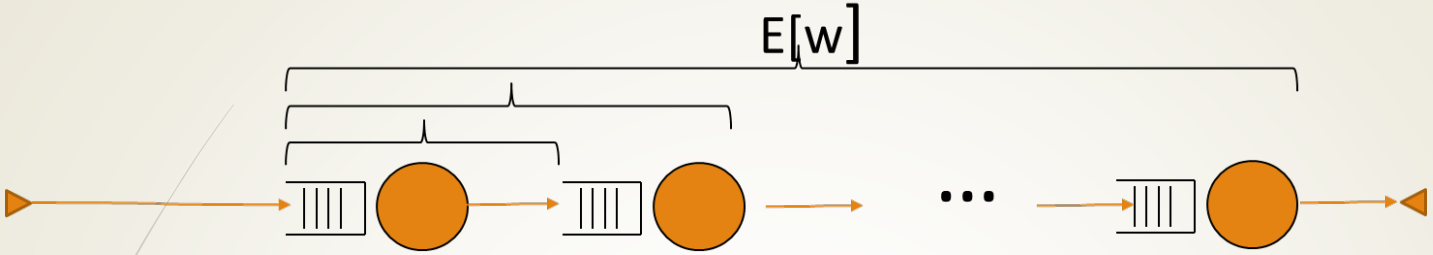
Tandem Servers

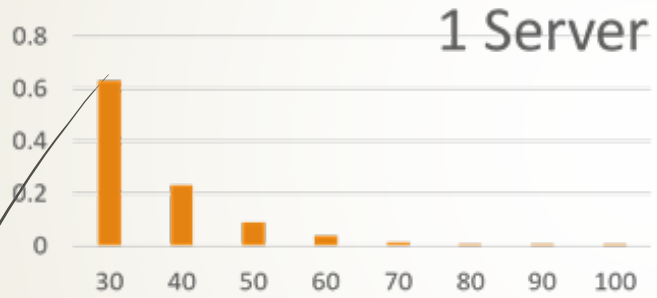


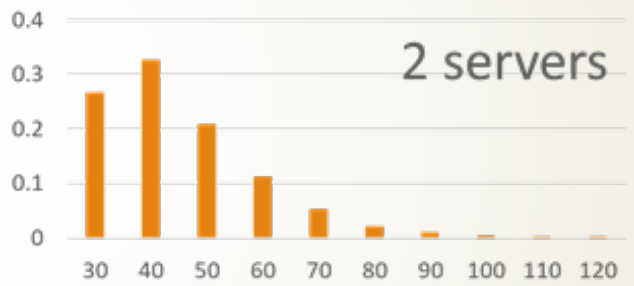
90%R Values

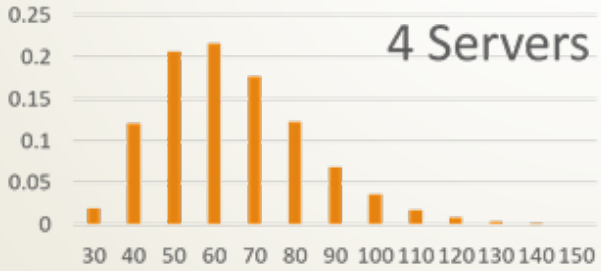
						% Error	
# Srvrs	Arrivals / Sec	CV	M/M/1	Measure	Martin's Estimate	M/M/1	Martin's Estimate
1	.2/s	1.00	3.00	2.99	3.00	0.00	0.00
2	.2/s	0.71	3.00	2.38	2.42	0.26	0.02
4	.2/s	0.50	3.00	1.94	2.00	0.55	0.03
8	.2/s	0.35	3.00	1.64	1.71	0.83	0.04
1	.4/s	0.99	3.00	2.98	2.98	0.01	0.00
2	.4/s	0.70	3.00	2.35	2.40	0.28	0.02
4	.4/s	0.50	3.00	1.93	1.99	0.55	0.03
8	.4/s	0.35	3.00	1.64	1.70	0.83	0.04
1	.8/s	1.03	3.00	2.99	3.06	0.00	0.02
2	.8/s	0.72	3.00	2.34	2.45	0.28	0.04
4	.8/s	0.51	3.00	1.95	2.01	0.54	0.03
8	.8/s	0.35	3.00	1.64	1.71	0.83	0.04
1	.9/s	0.98	3.00	3.03	2.96	-0.01	-0.02
2	.9/s	0.69	3.00	2.34	2.38	0.28	0.02
4	.9/s	0.47	3.00	1.88	1.95	0.60	0.04
8	.9/s	0.34	3.00	1.63	1.69	0.84	0.04

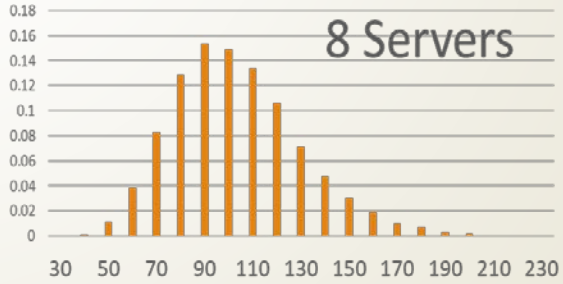
Tandem Servers and Normality





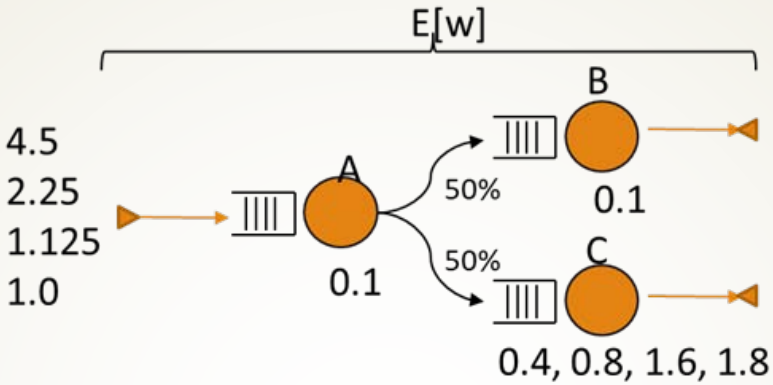






pdf for Response Times

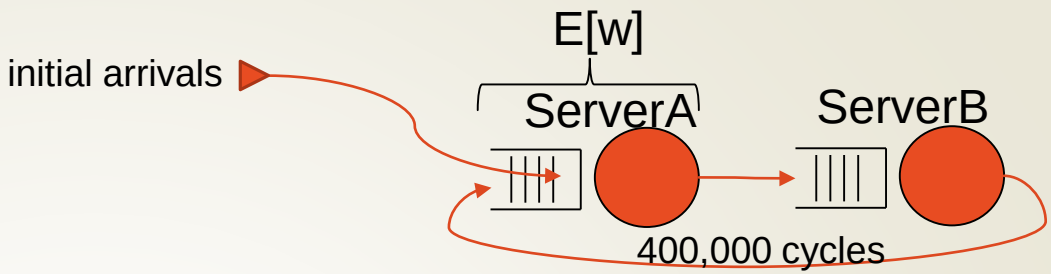
Asymetrical Servers



Utilization ~90%

					% Error	
Arrival Rate	CV	M/M/1	Measure	Martin's Estimate	M/M/1	Martin's Estimate
4.55	1.03	3.00	3.10	3.06	-0.03	-0.01
2.27	1.37	3.00	3.84	3.73	-0.22	-0.03
1.12	1.62	3.00	4.39	4.24	-0.32	-0.04
1.00	1.71	3.00	4.42	4.42	-0.32	0.00

Comparison of M/M/1 95%R and Martin's Estimate



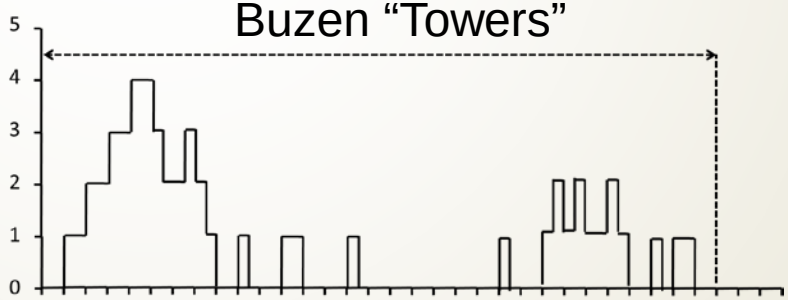
Simulation Instrumentation

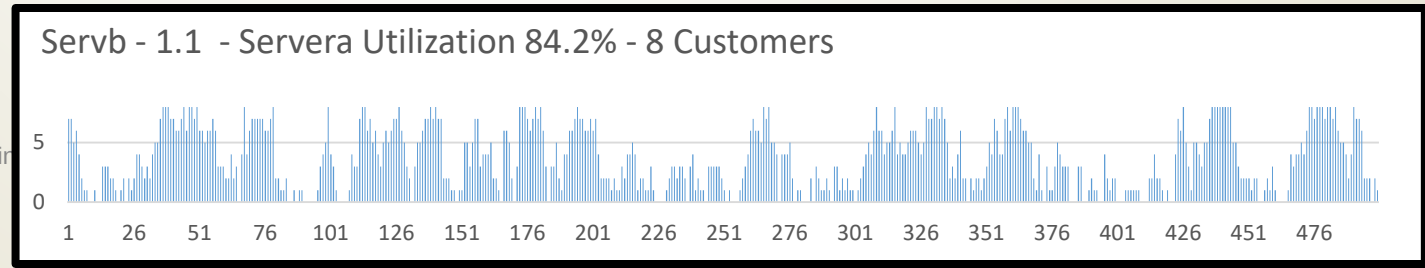
```
if (COLLECTSTATS_RESP)
        //write response time for server to fp stats file
        responsea = (clock - t1);
        fprintf(fpStats, "\t%5d\t%5d\t%10.2f\n", customer_number,
                global_customer_count, responsea);
        }

if (COLLECTSTATS_SERVA_ARRIVAL_TIME)
        // write global interarrival time for server to fpArrivals file
        {
        fprintf(fpArrivals, "\t %5d\t%5d\t%010.2f\n", customer_number,
                global_customer_count, (t1- global_arrival_time));
        global_arrival_time = t1;
        global_customer_count++;
                }
```

Simulation Instrumentation

```
void snapshotTower()
 // process that takes snapshots of number in system (server a)
{
        int num_in_system;
        create("snap");
        hold(SNAPSHOT_INTERVAL);
        for (int i = 1; i <= TOTAL_SNAPSHOTS; i++) {
                num_in_system = a->num_busy() + a->qlength();
                if (COLLECTSTATS_TOWERS)
                {
                        // write to fptowers file

                        fprintf(fpTowers, "\t%d\n", num_in_system);
                        hold(SNAPSHOT_INTERVAL);
                }
        }
}
```

Buzen “Towers”





Happy Modeling!

One More Hint – As Picasso used to say:
Never fall in love with your model

- Questions???

Bibliography

- Allen, Arnold, Probability, Statistics, and Queueing Theory, Academic Press, 1978 (1st Ed.), 1990 (2nd Ed).
- Buzen, “Queueing Network Models of multiprogramming, PhD Thesis, Harvard University.
- Buzen, Rethinking Randomness, ISBN 9781508435983
- W. Feller, An Introduction to Probability and its Applications, Vol II, 2nd edition, Wiley, 1971.
- Gordon, W. J, and Newell, G. F., “Closed queueing systems with exponential servers, Operational Research, 1967
- Graham, Denning, Buzen, Rose, Chandy, Sauer, Bard, Wong, Muntz, Acm computing surveys – “Special Issue: Queueing Network Models of Computer System Performance

Bibliography

- Jackson, J. R, “Networks of Waiting Lines”, Operational Research 5, 1957
- Kleinrock, L, Queueing Systems, Wiley, 1975.
- Law, Averil, Simulation Modeling and Analysis, McGraw-Hill, 2001
- MacDougall, M. H, “Simulating Computer Systems”, MIT Press, 1985
- MacDougall,M. H. *SMPL - A Simple Portable Simulation Language,* Amdahl Corp, Technical Report April 1, 1980
- Martin, James, Systems Analysis for Data Transmission, Prentice Hall, 1972.
- Salsburg, Michael, “Using Simulation: Major Network Decisions”, CMG Proceedings 1984.
- Salsburg, Michael, “Simulation is not a Four Letter Word”, CMG Transactions, Spring 1994.

Bibliography

- Scherr, A. L, An analysis of Time Shared computer systems, MIT Press, 1967.
- Schwetman, H. D, CSIMt: A C-BASED, PROCESS-ORIENTED SIMULATION LANGUAGE, Proceedings of the 1986 Winter Simulation Conference - http://delivery.acm.org/10.1145/320000/318464/p387-schwetman.pdf?key1=318464&key2=7776578511&coll=&dl=acm&CFID=15151515&CFTOKEN=6184618
- Schwetman, H. D, *CSIM20 User's Guide (local WWW copy),* Mesquite Software, Inc., http://www.mesquite.com/documentation