Queuing Theory
A Quick View

Ray Wicks
561-236-5846
RayWicks@us.ibm.com

Bibliography

Ray has spent most of his career at IBM in the performance analysis and capacity planning end of the business in Poughkeepsie, London, and now at the Washington Systems Center. He is the major contributor to IBM's internal PA & CP tool zCP3000. This tool is used extensively by the IBM services and technical support staff world wide to analyze existing zSeries configurations (Processor, storage, and I/O) and make projections for capacity expectations.

Ray has given classes and lectures worldwide. He was a visiting scholar at the University of Maryland where he taught part time at the Honors College.

He won the prestigious Computer Measurement Group's A.A. Michelson award in 2000. His recent virtual sessions "Getting Started in Performance Analysis & Capacity Planning" workshop held for attendees in China and India was well accepted.

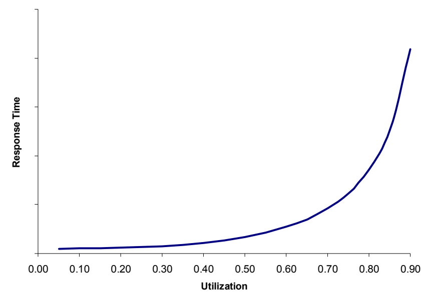
Queuing Theory

This session reviews some of the basics of queuing theory – the terminology, the assumptions, some statistics and some simple model implementations.
Although one may not do queuing theory, in Performance Analysis and Capacity planning discussions, it is very important to know what the terms mean.
Included will be Little's Law and M/M/1 and M/M/c equations. And then begins the slippery slope: once the basic equations are understood, the reality of non Markovian distributions make the match with reality a bear. Enter M/M/c/k models.
Excel graphics will be used to see what the equations are telling us. This will then be followed by a taste of the real implementation in larger queuing models: Mean Value Analysis.

Trade Marks, Copyrights & Stuff

- **On foils that appear in this presentation are not in the handout. This is to prevent you from looking ahead and spoiling my jokes and surprises.**

The Approach

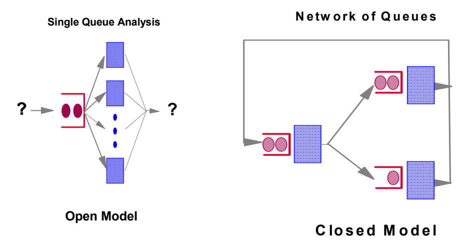


Queuing Theory

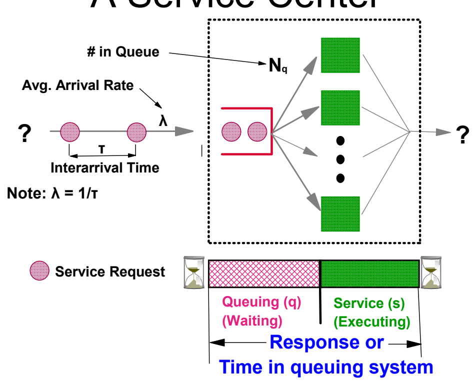
- **Shape of curve?**
- **Quantification?**
- **Exceptions?**

As the utilization increases, response time gets worse.

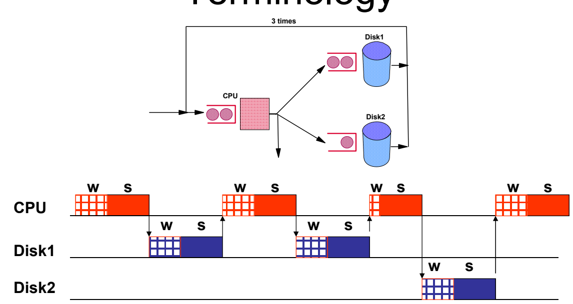
Queuing Analysis



A Service Center

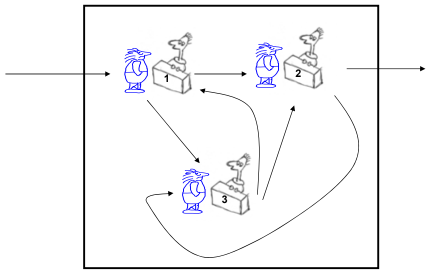


Terminology



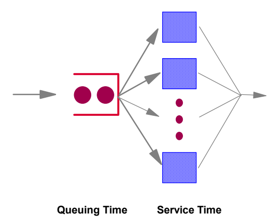
Service Demand at CPU = D = s1 + s2 + s3 +s4
Residence Time at CPU = w1 + s1 + w2 + s2 +w3 + s3 + w4 + s4
Response Time = Residence Time at CPU + Residence time at Disks

Total Response Time



RT = V1R1 +V2R2 +V3R3
Where Vi= # visits to Service Center i
Si = service time at Service Center i
Ri = Response at Service Center i

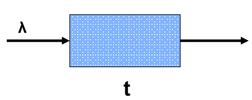
Service Center Considerations



- **Arrival Rate**
- **Total Population (Closed or ∞?)**
- **Number of Servers**
- **Speed of Servers**
- **Queuing Discipline**
- **Time Distributions**

- **Queuing Time**
- **Service Time**

Utilization Law



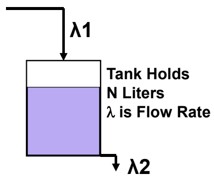
λ =	**Throughput (transactions/sec)**	**24**
t =	**Service time (sec/trans)**	**0.053 Secs**
u =	**utilization of server (Utilization= % = Secs/sec busy)**	
N =	**number users in server "system"**	
c =	**number of servers**	**2**

λt = 24 * 0.053 = 1.272 seconds/second (Traffic)
u = λt/c = 1.272/2 = 63.6%
λ*Service_Time = Traffic

Examples

- **Workload uses 32% of a 4 way processor for 10 transactions / second. What's the service time?**
  - **Total CPU seconds = #CPs x Busy**
    = 4 x 32%
    = 4 x 0.32 = 1.28 Seconds
  - **Service time = CPU seconds/Transaction**
    = 1.28/10 = 0.128 seconds
- **A 5 Server system is busy for 23 seconds over a minute. What's the average System utilization?**
  - **23/60 = 0.38 seconds/second**
  - **Utilization = total CPU seconds/second / # servers = 0.38/5 = 0.076**
  - **7.6%**

Little's Law



λ1 < **λ2**?
λ1 > λ2?
λ1 = λ2 (steady state)

At Steady State:
Average Residency Time = T = N/λ
Response Time = T
λ*Response_Time = Intensity

Example

A server processes 630,000 request in a half hour. The average number of requests in the server is 2. What's the average Response time?

λ = 630000/1800 = 350/second
N = λT
T = N/λ = 2/350 = 0.0058 seconds

→Know ST & number in system, you can compute RT
→ Q = RT - ST

Philosophical Remark

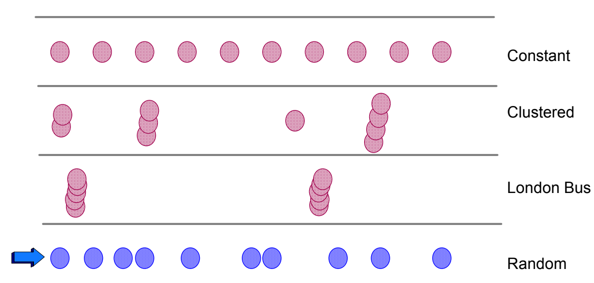
We Understand a law by trying to break it

$$\forall x \Phi x \ .{\approx}.\ \sim\exists y \sim\Phi y$$

We Generalize on limited information

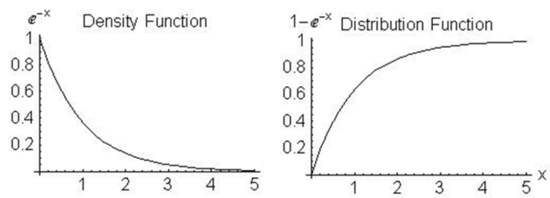
$$\exists y \Phi y\ .{\approx}.\ \forall x \Phi x$$

Distributions



Each circle represents an arrival. If the interval is one second, notice that the average inter-arrival time in all cases is 100Ms or 0.1 seconds. What's the impact on response time for various service times?

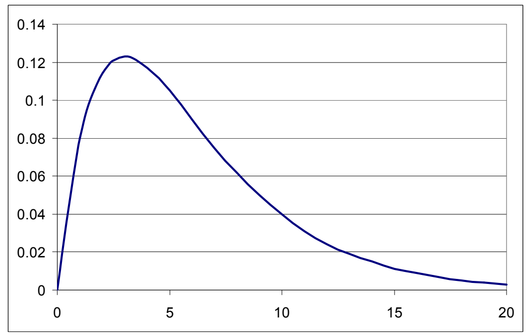
Exponential Function $f(x)=e^{-x}$



Full Function is: $\lambda e^{-\lambda x}$ where $E[f(x)]=1/\lambda$

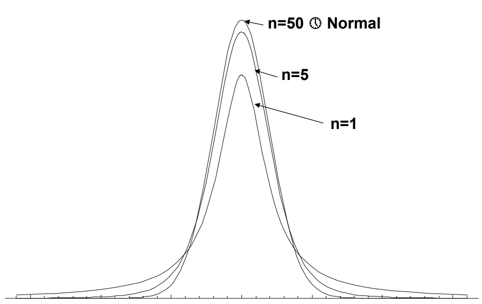
Erlang K=3 Θ=2

$$\frac{x^{K-1}\, e^{-x/\Theta}}{\Theta^{k}(K-1)!}$$



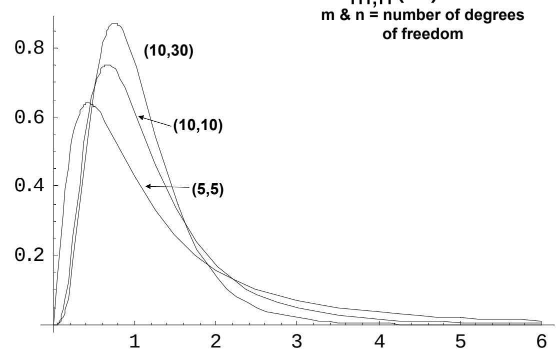
T-Distribution $f_n(t)$

n= number of degrees of freedom

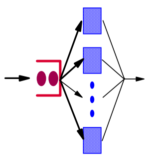


F-Distribution $f_{m,n}(F)$

m & n = number of degrees of freedom



Kendal Notation A/B/c/K/m/Z



A describes inter-arrival time
B describes the service time distribution
c the number of servers
K maximum number of users allowed in system
m number of users in population
Z is the queuing discipline

Common distributions
M exponential
D constant
Ek Erlang-k

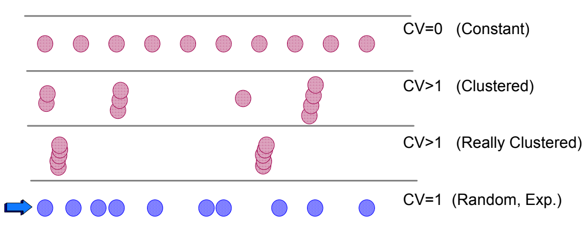
M/M/1 = M/M/1/∞/∞/FCFS

Queuing Theory Paradox

Buses pass a certain corner with an average time between them of 20 minutes. What is the average time that one would expect to wait?

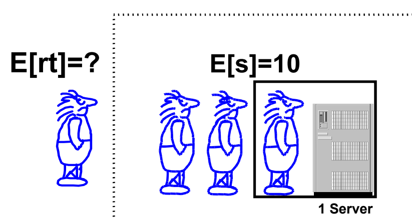
Queuing Theory Paradox

Buses pass a certain corner with an average time between them of 20 minutes. What is the average time that one would expect to wait?



Coefficient of Variation, CV = standard deviation / mean or σ/μ

Expected Response New Arrival

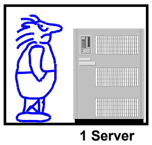


E[rt]=E[s]*(1+N[q])

If the **service time is distributed exponentially (CV=1)**, the expected value of the time left of the one getting service is the expected value!

Expected Time to Completion

E[s]=10

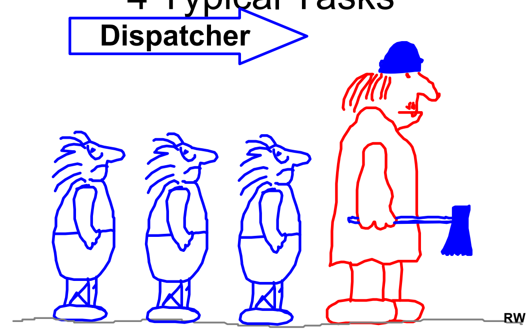


CV	Time Left
0	5
1	10
2	15

Upon arrival of a new request, how much time is remaining for someone in service?

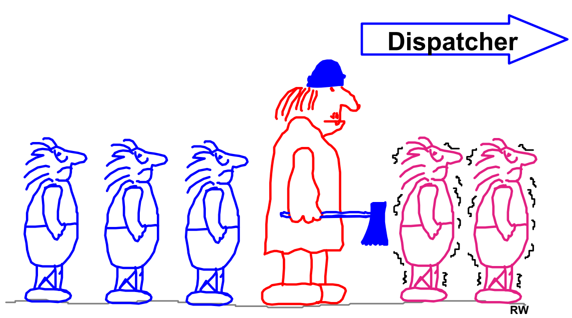
$E[ST] \times (1+CV^2)/2$

4 Typical Tasks

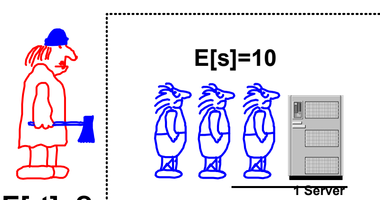


Guess who gets all they want? How much do we, the little people, get if there's one server? Two servers? Four servers? What's best?

A More Complicated Situation



Expected Response Big Shot?

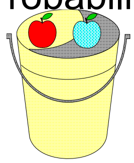


E[rt]=?

E[rt]=E[s]

If the arriving request is a "Big Shot" (high priority) and he can jump to the head of the line and preempt the one getting service, what's the expected RT?

Probability



50/50

- What's the probability of drawing a Red apple from a bucket?
- What's the probability of finding a server busy if it is averaging 50% busy?

Probability

50/50 50/50

- What's the probability of drawing a Red apple from each bucket?
- What's the probability of finding both servers busy if they are both averaging 50% busy?

An Ugly Formula

Erlang's C formula (M/M/c) for the probability of finding c Servers Busy (a Variation)

$$\boxed{C(c, T) = \frac{\frac{T^c}{c!}}{\frac{T^c}{c!} + (1 - U)\sum_{n=0}^{c-1} \frac{T^n}{n!}}}$$

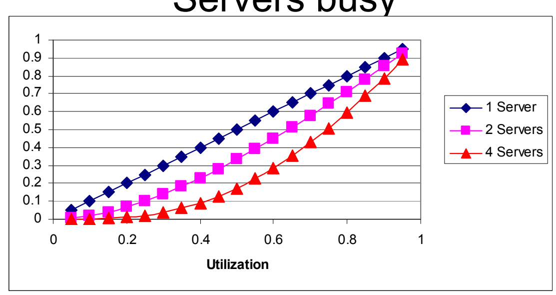
$$C(1, T) = U$$

$$C(2, T) = \frac{2\,U^2}{(1 + U)}$$

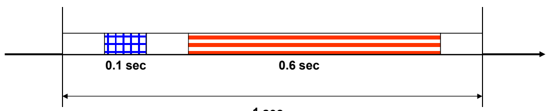
$$C(4, T) = \frac{32\,U^4}{3 + 9\,U + 12\,U^2 + 8\,U^3}$$

T=total Traffic
U=Average Utilization = T/c

Probability of 1, 2, and 4 Servers busy



True if



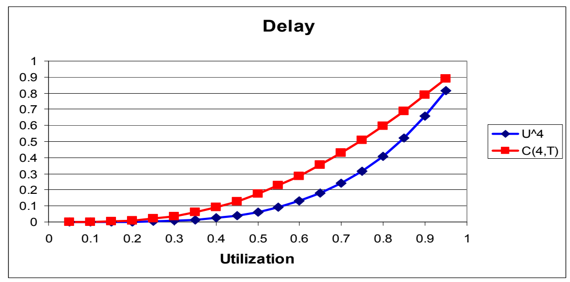
Probability of finding server busy?
Pr Server busy = busy time / clock = 0.7 secs/ 1 second
Pr(busy)=0.7?

Pr(server busy) = 0.6 / 0.9 = 0.67

Pr(server busy) = 0.1 / 0.4 = 0.25

Probability of 4 Servers busy

$C(4,T) = (32U^4)/(3+9U+12U^2+8U^3)$

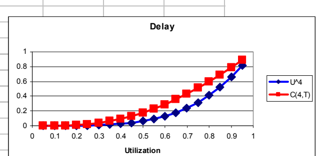


The intuition of U^4 would be optimistic compared to Erlang's C(4,T). Which would you use?

Excel Implementation

=POWER(A4,4) =(32*B4^4)/(3+(9*B4)+(12*B4^2)+(8*B4^3))

Utilization	U^4	C(4,T)
0.05	0.00000625	5.74548E-05
0.1	0.0001	0.000794439
0.15	0.00050625	0.00348612
0.2	0.0016	0.009580838
0.25	0.00390625	0.020408163
0.3	0.0081	0.037049743
0.35	0.01500625	0.060303906
0.4	0.0256	0.090699734
0.45	0.04100625	0.128533647
0.5	0.0625	0.173913043
0.55	0.09150625	0.226798854
0.6	0.1296	0.287043189
0.65	0.17850625	0.354420798
0.7	0.2401	0.428654318
0.75	0.31640625	0.509433962
0.8	0.4096	0.596432472
0.85	0.52200625	0.689316222
0.9	0.6561	0.787753264
0.95	0.81450625	0.891418995

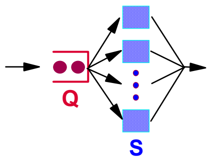


Example

If a system has 2 servers. For what utilization threshold I might expect the probability of both servers being busy to be less than 0.1?

$C(2,T) = (2U^2) / (1 + U)$
$0.1 = (2U^2) / (1 + U)$
$20U^2 - U - 1 = 0$
$U = -0.2, +.25$
Answer = at 25% busy the probability of finding both busy is 0.1. Or 90% of the time a request will not wait.

Erlang's M/M/c



$$E[RT] = E[S] + E[Q]$$

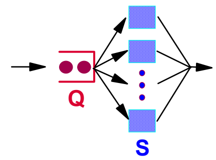
$$E[RT] = E[S] + \frac{C(c,T)E[S]}{c(1-U)}$$

$$E[RT] = E[S]\left\{1 + \frac{C(c,T)}{c(1-U)}\right\}$$

c = Number of CPs
U = Utilization
T= traffic or c*U
C(c,T) is Erlang's C formula
E[s] is expected service time

From any queuing theory book: Arnold or Jain for example.

Erlang's M/M/c



$$E[RT] = E[S] + E[Q]$$

$$E[RT] = E[S] + \frac{C(c,T)E[S]}{c(1-U)}$$

$$E[RT] = E[S]\left\{1 + \frac{C(c,T)}{c(1-U)}\right\}$$

Read as Contention Factor.
When CF=0, E[RT] = E[ST]
When CF=1, E[RT] = 2* E[ST]

c = Number of CPs
U = Utilization
T= traffic or c*U
C(c,T) is Erlang's C formula
E[s] is expected service time

M/M/1

$$E[RT] = E[S] + E[Q]$$

$$E[RT] = E[S] + \frac{C(c,T)E[S]}{c(1-U)}$$

$$E[RT] = E[S]\ \{1 + \frac{C(c,T)}{c(1-U)}\}$$

.... C=1

$$E[RT] = \frac{E[S]}{1-U}$$

IF E[S] = 30 Ms. And U=80%
Then E[RT] = 30/(1-0.8) = 150

M/M/1 Exercise

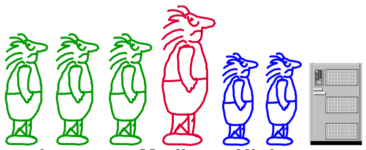
Workload	Service Time	Workload Utilization
Hi	0.05 sec	32%
Medium	0.25 sec	45%
Low	1.32 Min	12%

Assume M/M/1.

(1) What is the expected RT for Medium?

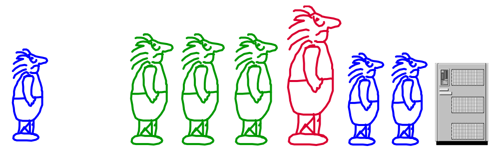
(2) At what effective utilization would the response time for medium exceed 1.2 seconds?

With priority



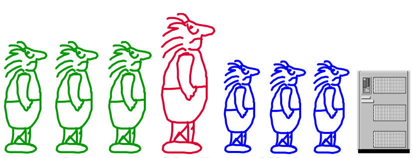
Workload	Workload Utilization
High	32%
Medium	45%
Low	12%

New arrival with priority



Workload	Workload Utilization
High	32%
Medium	45%
Low	12%

New arrival with priority



Workload	Workload Utilization	Perceived Utilization
High	32%	32%
Medium	45%	77%
Low	12%	89%

M/M/1 Exercise

Workload	Service Time	Utilization
Hi	0.05 sec	32%
Medium	0.25 sec	45%
Low	1.32 Min	12%

Assume M/M/1.

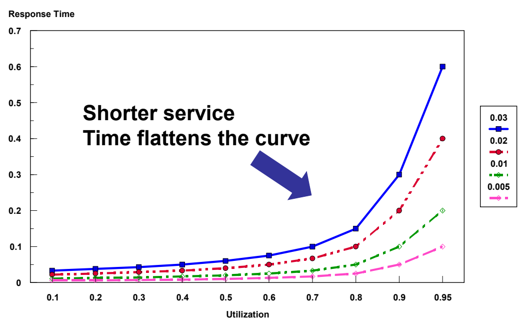
(1) What is the expected RT for Medium?

(2) At what effective utilization would the response time for medium exceed 1.2 seconds?

RT = ST / 1-U
RT = 0.25 / 1- .77
RT= 1.1

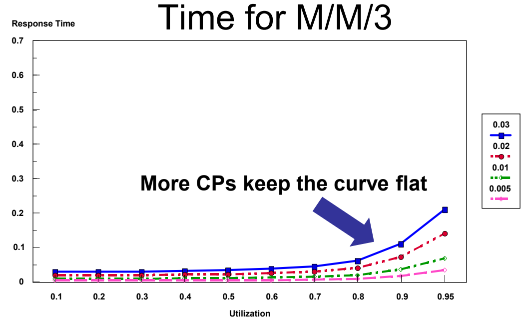
RT = ST / 1-U
1.2 = 0.25 / 1-U
U= 79%

The Effect of E[S] on Response Time for M/M/1



With 1 CP, the service time is a significant factor.

The Effect of E[S] on Response Time for M/M/3



Message: Many CPs are good IF an important task doesn't exhaust a server

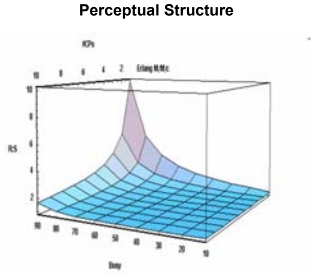
Erlang-C RT:ST

Conceptual Structure

$$C(c, T) = \frac{\frac{T^c}{c!}}{\frac{T^c}{c!} + (1 - U)\sum_{n=0}^{c-1}\frac{T^n}{n!}}$$

$$\frac{E[RT]}{E[ST]} = \frac{1 + C(c,T)}{c(1-U)}$$

Perceptual Structure



Overall, as the average busy (utilization) gets above 60 or 70, the number of CPs has a significant impact on the Response time to Service time ratio.

Simple Capacity Plan

Base
 # LCPs 4
MIPS 1000
MIPS/LCP 250

Target
 # LCPs 6
MIPS 1200
MIPS/LCP 200

Base → Target
- **More MIPS**
- **More Engines**
- **Slower Engines**

Problem: For the following workload, find the future workload utilization by month if the per annum growth rate is 30%.

Hi	**40.0**
Middle	**25.0**
Low	**20.0**
Total	85.0

Growth Computations

	Utilization Input		Projected
Workload	Jun-06	*F	Jul-06
Hi	40.0		40.9
Middle	25.0		25.6
Low	10.0		10.2
Total	75.0		76.7

$U_{t+1} = 1.022U_t$

PA Growth **G**	30	**Per Annum**
Period Length **L**	1	**in Months**
Period **F**	1.022104451	

$$(0.01 * G)^{L/12}$$

$$1.02210445^{12} = 1.3$$

Growth Computations

	Utilization Input		Projected
Workload	Jun-06	*F	Sep-06
Hi	40.0		42.7
Middle	25.0		26.7
Low	10.0		10.7
Total	75.0		80.1

PA Growth **G**	30	**Per Annum**
Period Length **L**	3	**in Months**
Period **F**	1.067789972	

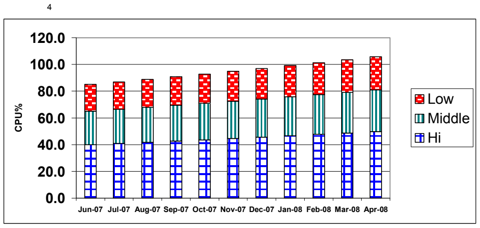
$$(0.01 * G)^{L/12}$$

$$1.06778997^{4} = 1.3$$

Base Utilization

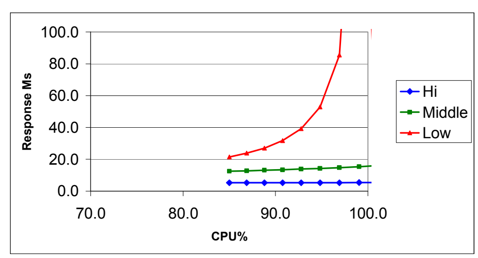
Workload	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-07	Dec-07	Jan-08	Feb-08	Mar-08	Apr-08
Hi	40.0	40.9	41.8	42.7	43.7	44.6	45.6	46.6	47.6	48.7	49.8
Middle	25.0	25.6	26.1	26.7	27.3	27.9	28.5	29.1	29.8	30.4	31.1
Low	20.0	20.4	20.9	21.4	21.8	22.3	22.8	23.3	23.8	24.3	24.9
Total	85.0	86.9	88.8	90.8	92.8	94.8	96.9	99.1	101.2	103.5	105.8

PA Growth	30
Period	1
Period F	1.022104
#CPs	4



Base Response Time

Workload	Service T	Jun-06	Jul-06	Aug-06	Sep-06	Oct-06	Nov-06	Dec-06	Jan-07	Feb-07	Mar-07	Apr-07
Hi	5	40.0	40.9	41.8	42.7	43.7	44.6	45.6	46.6	47.6	48.7	49.8
Middle	10	65.0	66.4	67.9	69.4	70.9	72.5	74.1	75.7	77.4	79.1	80.9
Low	10	85.0	86.9	88.8	90.8	92.8	94.8	96.9	99.1	101.2	103.5	105.8
Hi		5.2	5.2	5.2	5.2	5.3	5.3	5.3	5.3	5.4	5.4	5.4
Middle		12.5	12.8	13.1	13.4	13.8	14.3	14.8	15.4	16.1	17.0	18.0
Low		21.5	23.8	27.0	31.7	39.2	52.8	85.6	269.6	0.0	0.0	0.0



Migration Options

- **Migrate to same MIPS and more CPs.**
- **Migrate to same MIPS and fewer CPs.**
- **Migrate to more MIPS and fewer CPs.**
- **Migrate to more MIPS and more CPs.**

Utilization?
Service Time?
Evaluation?

Computations

Base

 # LCPs 4
MIPS 1000
MIPS/LCP 250

Target

 # LCPs 6
MIPS 1200
MIPS/LCP 200

$$\text{UTILtarget} = \frac{\text{POWERbase}}{\text{POWERtarg}} * \text{UTILbase}$$

$$= \frac{1000}{1200} * \text{UTILbase}$$

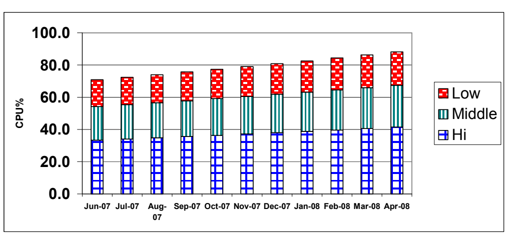
$$\text{SERVtarget} = \frac{\text{PUSPEEDBase}}{\text{PUSPEEDtarg}} * \text{SERVbase}$$

$$= \frac{250}{200} * \text{SERVbase}$$

Target Utilization

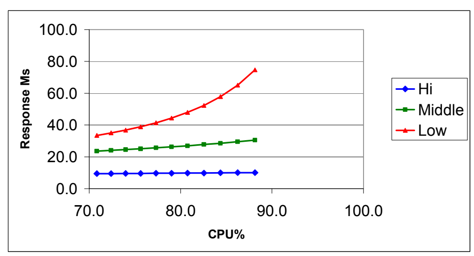
Workload	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-07	Dec-07	Jan-08	Feb-08	Mar-08	Apr-08
Hi	33.3	34.1	34.8	35.6	36.4	37.2	38.0	38.8	39.7	40.6	41.5
Middle	20.8	21.3	21.8	22.2	22.7	23.2	23.8	24.3	24.8	25.4	25.9
Low	16.7	17.0	17.4	17.8	18.2	18.6	19.0	19.4	19.9	20.3	20.7
Total	70.8	72.4	74.0	75.6	77.3	79.0	80.8	82.5	84.4	86.2	88.1

PA Growth	30
Period	1
Period F	1.022104
#CPs	2

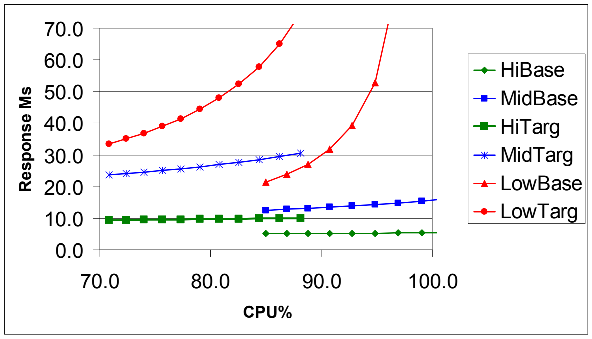


Target Response Time

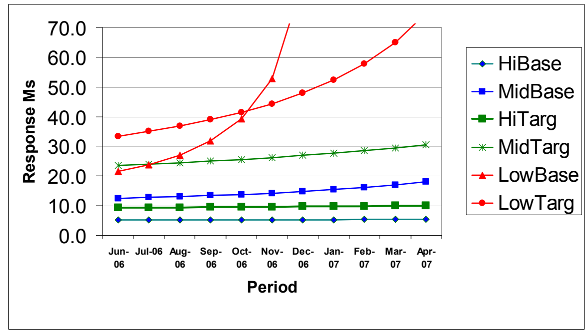
Workload	Service T	Jun-07	Jul-07	Aug-07	Sep-07	Oct-07	Nov-07	Dec-07	Jan-08	Feb-08	Mar-08	Apr-08
Hi	8.333333	33.3	34.1	34.8	35.6	36.4	37.2	38.0	38.8	39.7	40.6	41.5
Middle	16.66667	54.2	55.4	56.6	57.8	59.1	60.4	61.8	63.1	64.5	65.9	67.4
Low	16.66667	70.8	72.4	74.0	75.6	77.3	79.0	80.8	82.5	84.4	86.2	88.1
Hi		9.4	9.4	9.5	9.5	9.6	9.7	9.7	9.8	9.9	10.0	10.1
Middle		23.6	24.0	24.5	25.0	25.6	26.3	26.9	27.7	28.6	29.5	30.5
Low		33.4	35.0	36.8	38.9	41.4	44.4	47.9	52.3	57.8	65.0	74.7



Compare Utilization

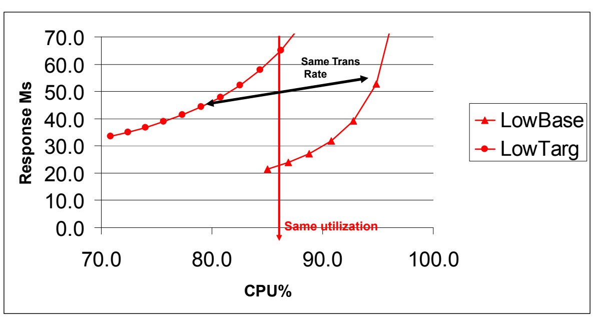


Compare Transaction Rate

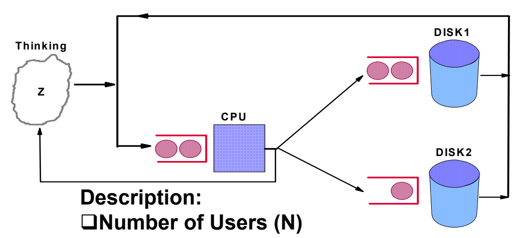


For each period, Base and Target have same transaction rate.

How to Compare?

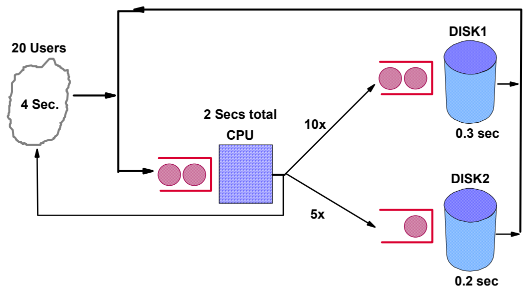


Mean-Value Analysis (MVA)



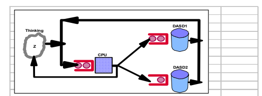
Description:
- **Number of Users (N)**
- **Think time (Z)**
- **By Request, for each device**
  - **Service time per visit (Si)**
  - **Number of visits (Vi)**

Mean-Value Analysis (MVA) – Closed Model



(Ref. Jain section 34.2)

Minimum RT (1 user)

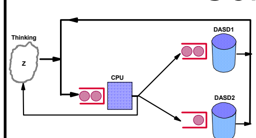


Number of Users (N)	1
Think time (Z)	4
Number of Devices (3)	
Total CPU Service	2
CPU Service per Visit	0.1
Disk1 Service per Visit (S1)	0.3
Number of Visits (V1)	10
Disk2 Service per Visit (S2)	0.2
Number of Visits (V2)	5

Minimum Residency (Response Time) is time without queuing.

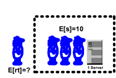
Minimum RT = Total residency for CPU + Disk1 + Disk2
= 2 + 10*0.3 + 5*0.2 = 6.0

Computations



For each server i:
Ri = response time
Si = service time
Qi = queue length (included one in service)
Vi = visits

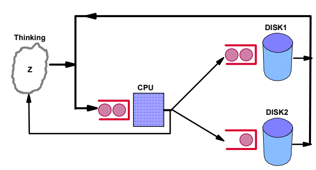
$R_i = S_i * (1 + Q_i)$



$R = \sum_{i=1}^{3} R_i * V_i$

$\lambda = N/(Z+R)$ (Little's Law)

Input



Number of Users (N)	20
Think time (Z)	4
Number of Devices	3
Total CPU Service	2
CPU Service per Visit	0.125
DISK1 Service per Visit (S1)	0.3
Number of Visits (V1)	10
DISK2 Service per Visit (S2)	0.2
Number of Visits (V2)	5

Minimum RT = Total residency for CPU + Disk1 + Disk2
= 2 + 10*0.3 + 5*0.2 = 6.0

Algorithm

- Initialize Qi = 0 for all i.
- Iterate by the number of users (N)

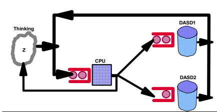
For n = 1 to N

- Iterate by the number of Service Centers (M)

For i = 1 to M

1. Ri = Si*(1+Qi)
2. R = Sum_over_i = Ri*Vi
3. Thruput = N/(Z+R)
4. Set Qi =λ*Vi*Ri

Results

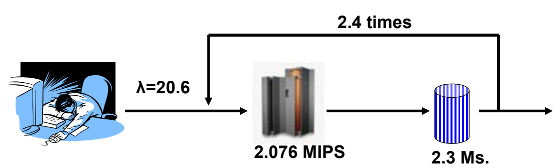


		Iteration	CPU	Disk1	Disk2	System	Thruput	CPU QL
		0						0
		1	0.13	0.30	0.20	6.00	0.10	**0.20**
		2	0.15	0.39	0.22	7.40	0.18	0.42
		3	0.18	0.51	0.24	9.09	0.23	0.65
		4	0.21	0.65	0.25	11.05	0.27	0.88
		5	0.23	0.82	0.27	13.26	0.29	1.09
		6	0.26	1.01	0.28	15.66	0.31	1.27
Number of Users (N)	20	7	0.28	1.22	0.28	18.22	0.32	1.43
Think time (Z)	4	8	0.30	1.46	0.29	20.89	0.32	1.56
Number of Devices (3)		9	0.32	1.71	0.29	23.65	0.33	1.67
Total CPU Service	2	10	0.33	1.97	0.30	26.47	0.33	1.75
CPU Service per Visit	0.125	11	0.34	2.24	0.30	29.34	0.33	1.82
DASD1 Service per Visit (S1)	0.3	12	0.35	2.51	0.30	32.24	0.33	1.86
Number of Visits (V1)	10	13	0.36	2.80	0.30	35.18	0.33	1.90
DASD2 Service per Visit (S2)	0.2	14	0.36	3.08	0.30	38.13	0.33	1.93
Number of Visits (V2)	5	15	0.37	3.37	0.30	41.09	0.33	1.95
		16	0.37	3.67	0.30	44.06	0.33	1.96
		17	0.37	3.96	0.30	47.04	0.33	1.97
		18	0.37	4.26	0.30	50.03	0.33	1.98
		19	0.37	4.56	0.30	53.02	0.33	1.99
		20	0.37	4.85	0.30	56.02	0.33	1.99

Mean Value Analysis in Excel

Workload Modeling

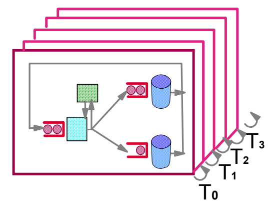
Description	TrRate	CPU%/CR	Single CP%	MIPS	MIPS/Tr	DASD I/O	IO/Tr	MIPS/IO	IO Resp
DB2C	20.6	1.3	28.7	103.2	5.0	49.7	2.4	2.076	2.3



Flip Book Animation

Simulation an Alternate to Queuing Equations

- Initialize Model for T0
- Schedule first event
- At each Ti for each Queue
  - Gather Stats on Ti-Ti-1
  - Busy? Population?
  - New Arrivals?
  - Start Counters
  - Departures?
  - Compute Behavior
  - Send to next queue
  - Schedule Next Event
- End Simulation
- Report



Simulation Demo(?)

A **simulation is an imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviors of a selected physical or abstract system.**

Modeling Issues

- **Analytic Queuing theory (and simulation) is difficult to apply in more than simple cases (Single server Unix).**
- **M/M/c can approximate (bound more complicated) cases of M/G/c/k cases. It's a good approximation at less than 100%.**
- **z/OS is complicated: WLM, priority, IRD, specialized PUs (zIIPs, zAAPs, IFLs).**
- **zSeries hardware behaves differently**
- **Packages & Services are available but it helps to know what's being done and what the terms mean.**
- **There are Single Task Multi Thread applications**

Bibliography

The Art of Computer Systems Performance Analysis, by Raj Jain, Wiley. I like this one. It is thorough and complete. A very good reference. Hard to find.
Capacity Planning for Web Performance, by Daniel A. Menasce and Virgilio A.F. Almeida, Prentice Hall. A good book on queuing theory, network structure and terminology and introduction to the topic.
Probability, Statistics, and Queuing Theory, by Arnold O. Allen, Academic Press Inc. This is the classic in queuing theory.
Computer Performance Analysis with Mathematica, Arnold O. Allen, AP Press 1994. This is a clearly written textbook even if you don't have access to the Mathematica package.
Excel Data Analysis, by Jinjer Simon, Wiley Publishing. This is a step by step visual approach to data analysis with Excel.