Capacity Planning Boot Camp

Part III: Going Guerrilla

Dr. Neil J. Gunther
Performance Dynamics Company
Castro Valley, California
`www.perfdynamics.com`

CMG-T Session 425

CMG 2008 Conference, Las Vegas, Nevada

Going Guerrilla

In this section, I am going to show you how some of the preceding CaP methods have been applied to real-life situations.

Table of Contents

Case Study: Sizing Against a Mainframe

How Many Horses?

Circa 1992 at Pyramid Technology [1].

A customer wants to replace an IBM legacy mainframe (IBM 3090-600S) with a Pyramid unix multiprocessor (MIServer ES Series).

How many processors are required to match the CPU horsepower used by the current mainframe?

Find a Common Starting Point

Model 3090-600S is rated at 105 MIPS (IBM LSPR rating).

Pyramid MIServer ES-Series uses 40 MHz R3000 microprocessors. How many IBM MIPS is that?

Historical information:

- Oracle Corp. sponsored a TP1 [2] benchathon in 1990.
- Best in class using ORACLE 6.0 database.
- Amdahl Corp. won the mainframe class.
- Pyramid won the mid-range class with an MIServer T-Series.

Research Public Performance Data

All benchmark runs were audited was Tom Sawyer (whose job was to prevent vendors from white-washing their benchmarks). ☺

Toms consulting company was called Performance Metrics Inc.

They had formal written reports which documented the audited TP1 results.

From those reports I learned that Amdahl used a 4-way model 5995-1440 rated at 105 IBM MIPS equivalent (running UTS Unix).

Pyramid used an MIServer 12-way T-Series.

Both platforms ran ORACLE 6.0 RDBMS.

Building the Bridge

Analyze common TP1 data.

Extract TPS conversion factor for mainframe and unix server.

Use as a baseline horsepower rating. Important metric is TPS/MIP.

Scale up across Pyramid machine generations:

- T-series used a proprietary TTL processor @ 10 MHz
- S-series used R3000 @ 25 MHz
- ES-series used R3000 @ 40 MHz

Don't forget to factor in overhead effects on scalability.

Amdahl TP1 Configuration

Part	Description	Quantity
5990	1440 proc board	4
5990	512MB RAM	1
5990	64 channels	1
6100	100 Storage Proc	4
6100	8 Chan Adapts.	4
6380	DASD spindles	60
6880	DASD Cntlr Unit	6
3064	UTS 2.1 O/S	1
3200	Network s/w	1
3200	ORA 6 RDBMS	1
3200	TP Option s/w	1
3200	SQL*Net	1
3200	TCP/IP option	1

Pyramid TP1 Configuration

Part	Description	Quantity
4152	T proc board	12
4091	128MB RAM	1
4008	32MB RAM	1
4062	IPX Cntlr Brd	1
4068	IOP Cntrl Brd	10
6012	SMD 1.1 GB Dsk	40
3064	OSx 5.1a O/S	1
3200	Network s/w	1
3200	ORA 6 RDBMS	1
3200	TP Option s/w	1
3200	SQL*Net s/w	1
3200	TCP/IP option	1

Construct an Excel spreadsheet.

Sizing By Numbers (new)

	A	B	C	D	E	F
1		S390			RISC	
2	**Baseline 1990**	**Amdahl**			**Pyramid**	
3	Model	**5995-1440**			**T-Series**	
4	N-way	4			12	
5	Spindles	61			37	
6	MIPS per CPU	28	IBM		10	SPEC
7	g-factor %	0.050	ORA 6.0		0.030	ORA 6.0
8	Aggregate MIPS	105	IBM		80.4	SPEC
9	Measured TP1	**416.0**	TPS		**208.5**	TPS
10						
11	**Platform 1991**					
12	Model				**S-Series**	
13	N-way				12	
14	Spindles				37	
15	MIPS per CPU				21	SPEC
16	g-factor %				0.035	ORA 6.0
17	Aggregate MIPS				168.84	SPEC
18	Measured TPC-B				370	TPS
19	Predicted TP1s				407	TPS
20						
21	**Platform 1992**					
22	Model				**ES-Series**	
23	N-way				24	
24	Spindles				56	
25	MIPS per CPU				32	SPEC
26	g-factor				0.025	ORA 7.1
27	Aggregate MIPS				326.4	SPEC
28	Measured TPC-B				720	TPS
29	Predicted TP1s				792	TPS
30	Predicted TPC-A				648	TPS
31	Published TPC-A				**645.2**	TPS
32						
33	**Self-Consisten**	use of g-scaling to get 1 CPU value				
34	TP1 per CPU				**77.6**	
35	N-way				24	
36	Predicted TP1s				791.52	
37		**N-way**	**TP1**			
38		2	151			
39		4	287			
40		6	407			
41		**8**	512	match 390 + 25% headroom		
42		10	601			
43		12	675			

Sizing By Numbers (new)

Pyramid ES-Series Sizing

Each ES-Series CPU board had 2-CPUs per board.

A 6-way ES server would give similar TP1 rating.

But propose an 8-way includes 25% headroom plus I/O MIPS.

Case study: Data Warehouse Optimization

Data Warehousing and Mining

Data mining is a huge business, e.g., Lexus-Nexus, Google, Amazon.

Requires organized data in a database; a data warehouse.

Efficient mining (e.g., DSS, OLAP) relies on optimized performance from well-planned data storage and access.

The best access performance relies on a form a parallelism for massive queries.

Queries are read-only and can be cached without consistency problems. No locking required (cf. OLTP: shared writes).

Parallel reads give the shortest query times.

How short is short?

Query Performance

Definitions:

FEP: Front-end processors

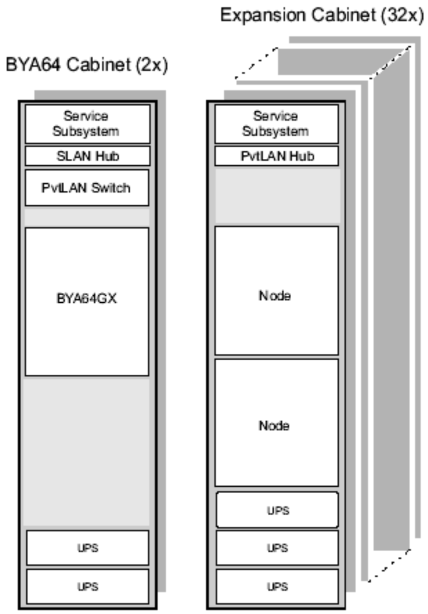
BEP: Back-end processors with local disks

In general, for N homogeneous sub-queries executing on the BEP, expect parallel query time $R(p)$ to scale hyperbolically with the number of processing nodes p as:

$$R(p) = \frac{R(1)}{p} \qquad (1)$$

where $R(1)$ is the query-time on a *uniprocessor*.

NCR Worldmark MPP Architecture



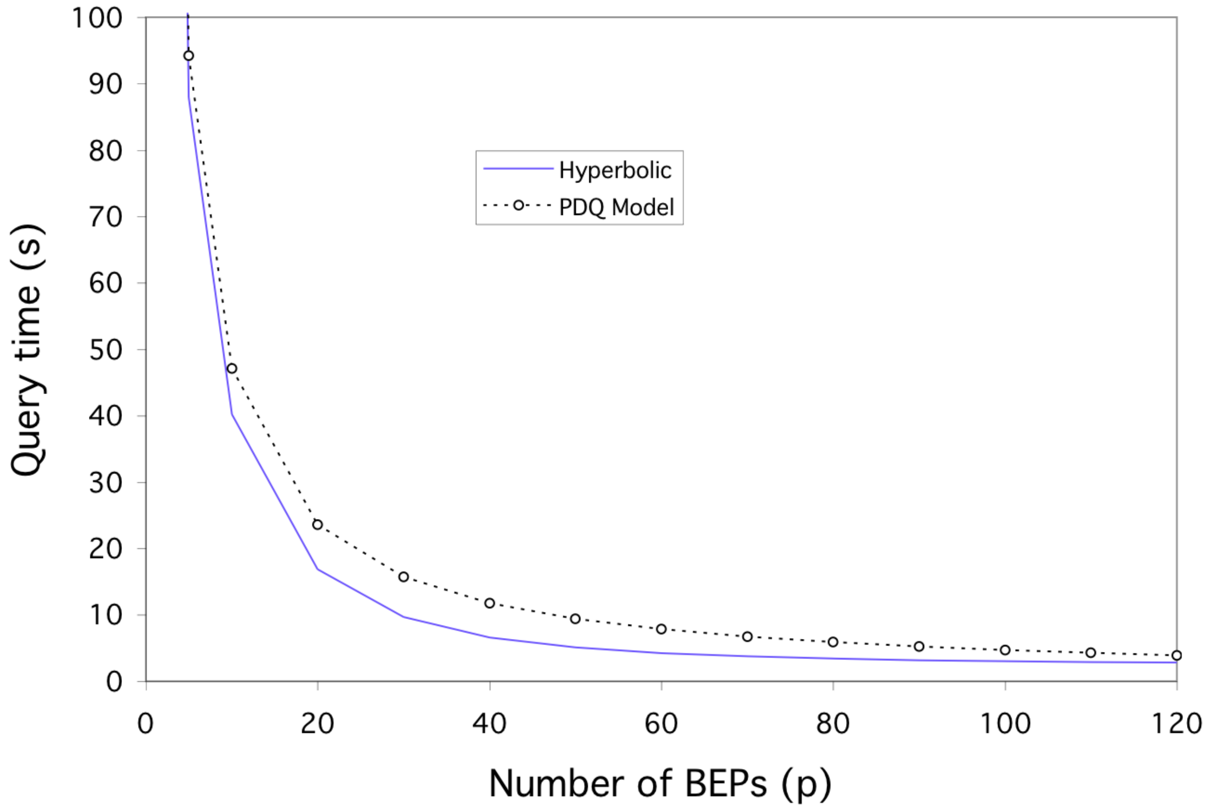
The WorldMark 5150 system can have from 2 to 128 nodes (2 per rack).

The SMP nodes communicate with each other through the BYNET high-speed interconnect.

Parallel Query-Time Models

	Predicted $R(p)$ (s)	
BEPs	Hyperbolic	PDQ
1	471.23	471.23
10	40.18	47.12
20	16.85	23.56
30	9.68	15.71
40	6.60	11.78
50	5.09	9.42
60	4.25	7.85
70	3.75	6.73
80	3.42	5.89
90	3.18	5.24
100	3.02	4.71
110	2.89	4.28
120	2.79	3.93

Predicted Query Times



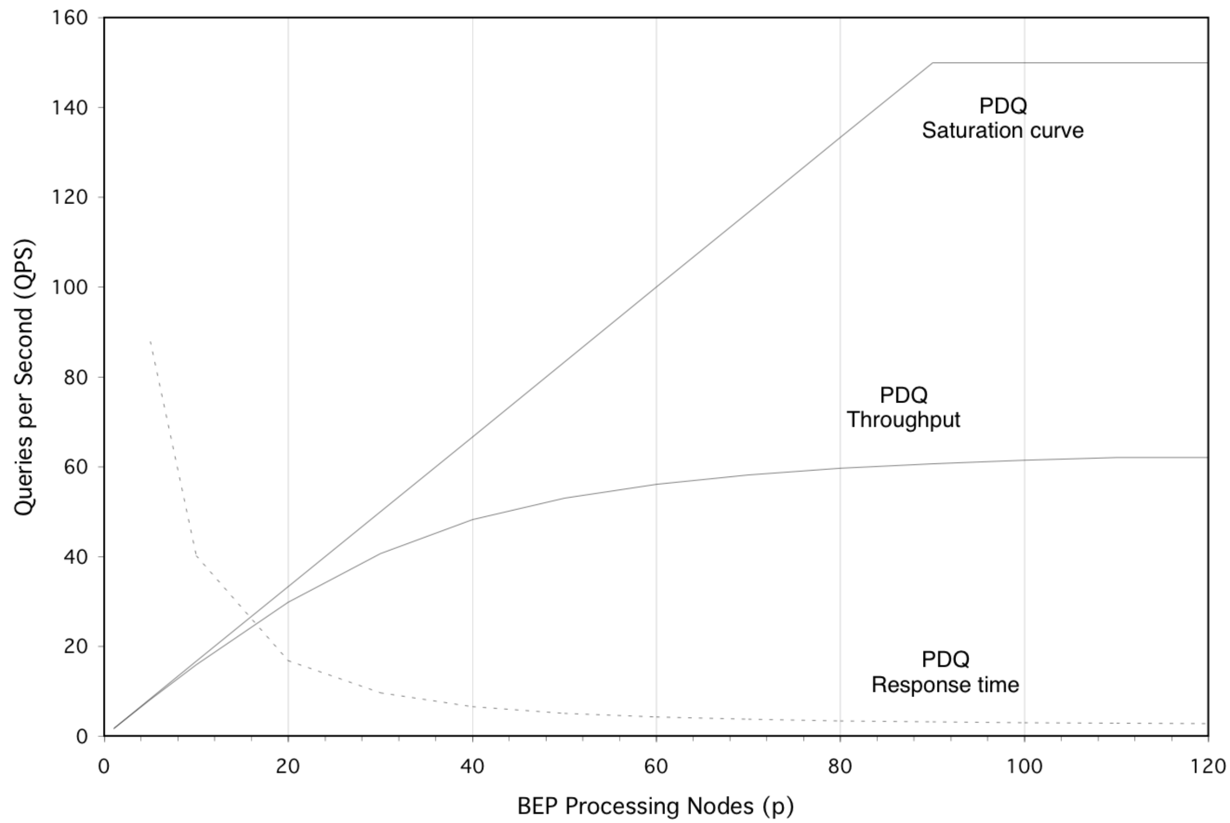
Parallel Query-Throughputs

BEP(p)	$X(p)$ QPS	$X_{\text{sat}}(p)$ QPS
1	1.66	1.67
10	15.94	16.67
50	53.03	83.33
70	58.19	116.67
90	**60.69**	**150.00**
100	61.45	150.00
120	62.07	150.00

The surprising things here is, we can best determine the BEP configuration by examining the predicted query throughput, rather that the response times (as would be naively expected).

This is the power of performance models.

Optimal BEP Processor Configuration



Note the knee in the saturation throughput curve at 90 BEPs.

How Does It Work?

Uniprocessor BEP: The preprocessing time on the FEP $<$ query time on the BEP

Intermediate BEPs: The preprocessing time on the FEP $\simeq$ query time on the BEP

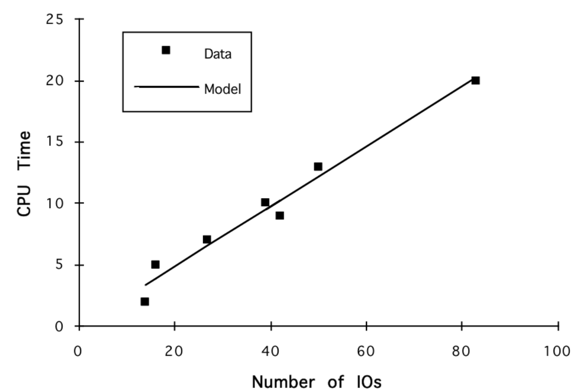
Parallel BEPs: The preprocessing time on the FEP $>$ query time on the BEP

Case study: Forecasting Web-Server Demand

Multivariate Linear Regression

In Part 1 we looked at a 2-dimensional regression model:

$$\hat{y} = m_1 x_1 + c \tag{2}$$



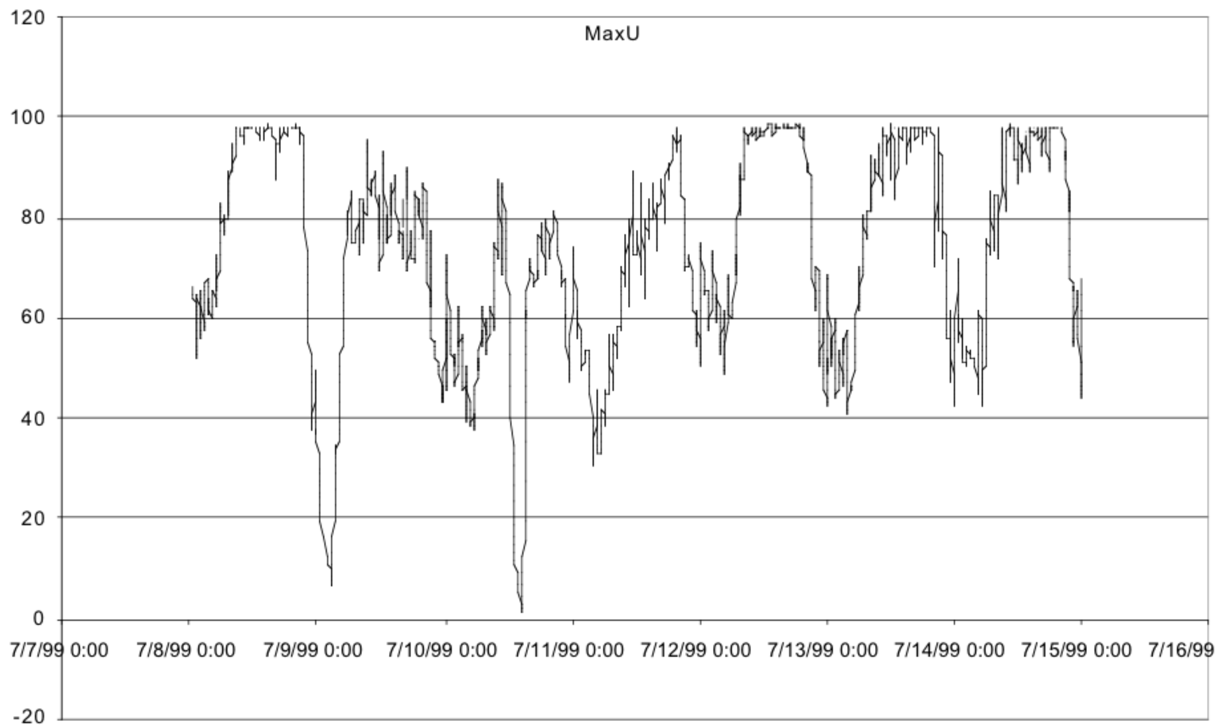
This time were going to make a 7-dimensional model:

$$\hat{y} = m_1 x_1 + \cdots + m_4 x_4 + c \tag{3}$$

The 4 random variables x_i are called *regressors*, and y is called the *response* variable [3]. The "hat" denotes an estimate based on the x's.

The Measured CPU Utilization Data

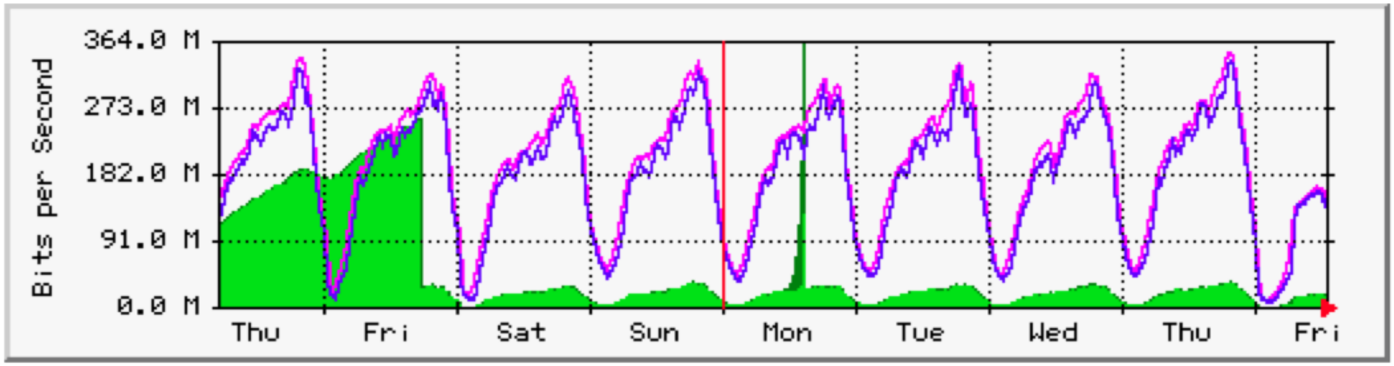
Heres a back-end database server for a major web site.



It's a 64-way E10K Starfire and it's saturated almost every day. How much CPU capacity is effectively being used?

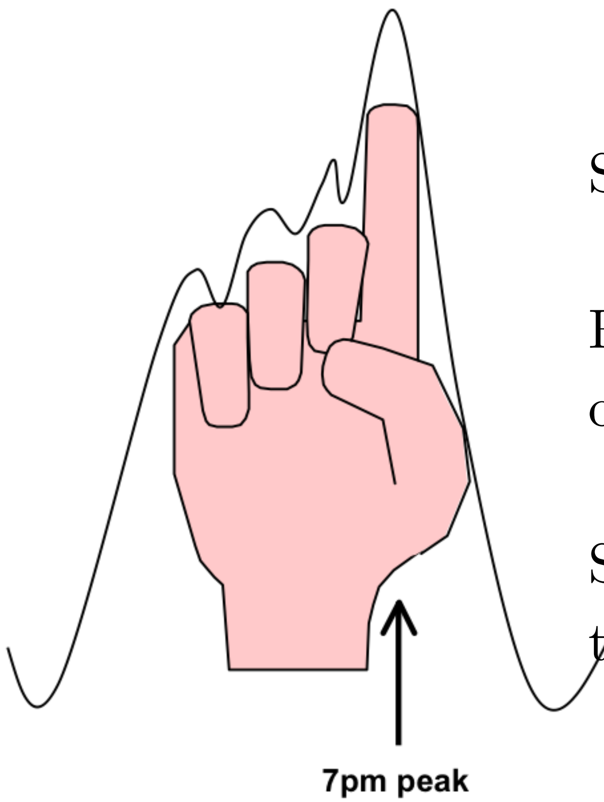
Other Data Sources

How much CPU capacity is effectively being used?
We can't tell because CPU utilization is bounded by 100



But other data sources (MRTG in this case) such as network bandwidth, which is not (yet) saturated, reveal a certain kind of daily traffic profile.

“Numero Uno” Signature



Shows a distinct peak around 7pm PDT.

Renditions of this load characteristic seen on all ma jor website systems.

Servers are more likely to be throttled than networks.

Noisy Fingers

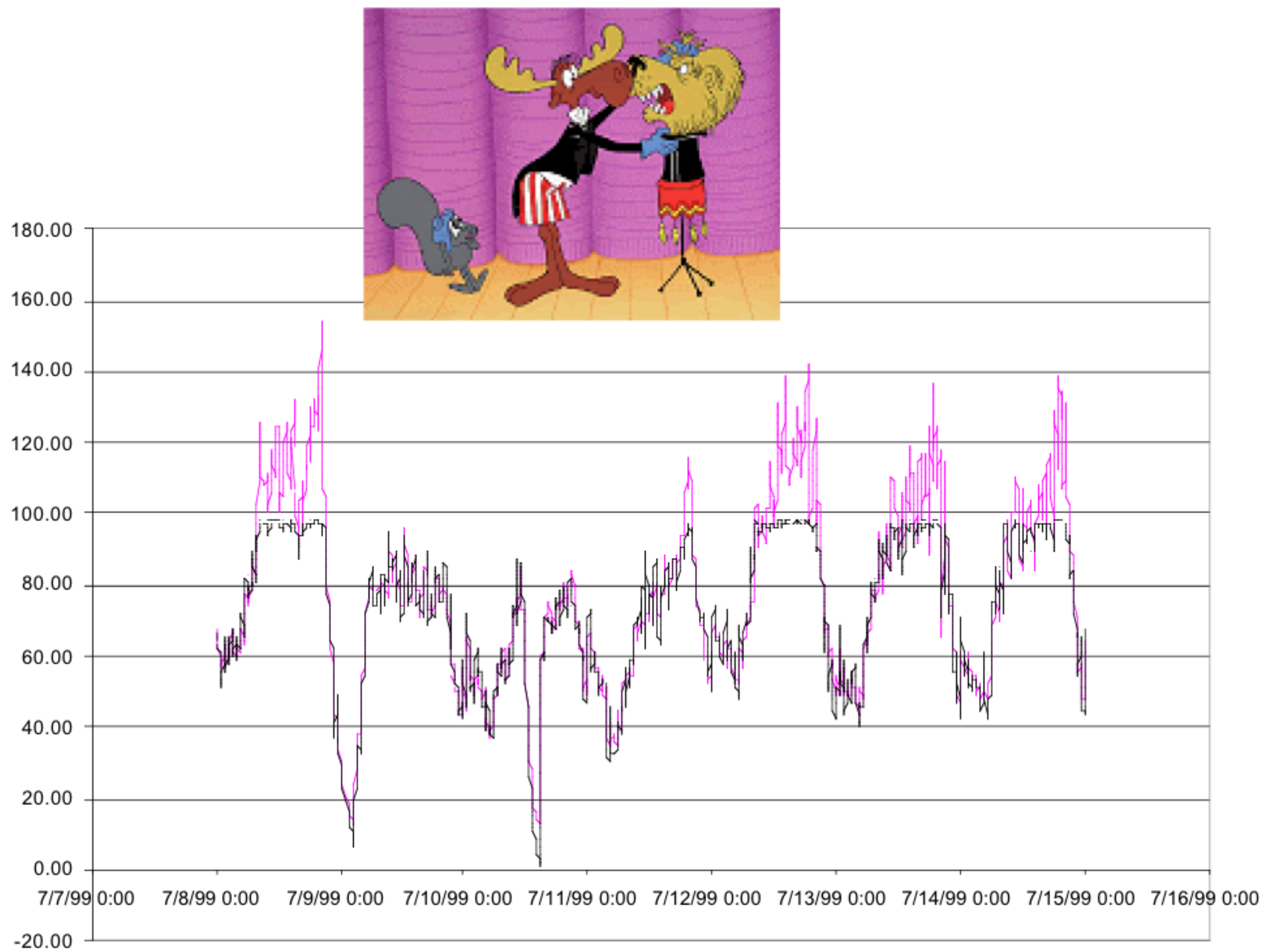
Its as if the numero uno fingers have been amputated in the raw CPU performance data. We would like to surgically re-attach those fingers.

But how could that be possible?
Like Bullwinkle, it would require pulling a rabbit out of the hat.

That's where the multivariate regression comes in.

But because it's a statistical technique, the reconstruction will produce *noisy* fingers.

Bullwinkle's Rabbit



Performance Dynamics Company℠

Excel Spreadsheet of Selected Metrics

MultiVarRegression.xls

	A	B	C	D	E	F	G	H	I	J
1	DateTime	AvgU	MaxU	MinU	AvgQ	MaxQ	MinQ	SumQ		
2	9/29/99 0:00	25.25	32	19	16.45	18.96	15.04	131.56		SUMMARY OUTPUT
3	9/29/99 0:16	27.25	45	11	17.01	22.49	14.18	136.08		
4	9/29/99 0:32	47.12	50	42	29.52	33.32	27.07	236.13		*Regression St*
5	9/29/99 0:48	45.88	53	38	27.29	32.09	24.62	218.29		Multiple R
6	9/29/99 1:04	45.25	53	40	29.88	34.12	27.42	239.04		R Square
7	9/29/99 1:20	50.88	56	45	32.84	35.76	30.58	262.73		Adjusted R Squa
8	9/29/99 1:36	52.25	60	48	33.45	37.02	30.88	267.56		Standard Error
9	9/29/99 1:52	54.5	72	46	36.43	42.95	30.75	291.42		Observations
10	9/29/99 2:08	57.12	63	49	37.5	42.51	33.7	299.98		
11	9/29/99 2:24	56.75	62	49	37.69	40.25	34.95	301.55		ANOVA
12	9/29/99 2:40	48.25	52	45	32.43	37.21	28.72	259.43		
13	9/29/99 2:56	54.25	63	48	35.21	38.14	32.82	281.71		Regression
14	9/29/99 3:12	44.75	57	34	29	38.76	24.06	232.02		Residual
15	9/29/99 3:28	54.5	70	39	32.44	40.85	23.54	259.51		Total
16	9/29/99 3:44	48.25	60	37	33.68	40.62	28	269.41		
17	9/29/99 4:00	43.38	64	35	27.91	33.89	24.81	223.25		
18	9/29/99 4:16	45.25	60	32	28.86	33.58	24.09	230.9		Intercept
19	9/29/99 4:32	45.88	52	42	30.38	35.03	26.98	243.04		MaxU
20	9/29/99 4:48	50.88	65	41	30.78	37.69	26.21	246.27		MinU
21	9/29/99 5:04	48.12	60	41	31.04	34.74	27.77	248.35		AvgQ
22	9/29/99 5:20	56.5	73	42	34.81	43.73	28.41	278.5		MaxQ
23	9/29/99 5:36	45.25	52	35	29.57	37.26	25.06	236.6		MinQ
24	9/29/99 5:52	46.88	59	37	28.94	33.6	23.46	231.5		SumQ
25	9/29/99 6:08	46.38	52	40	31.45	34.93	28.96	251.63		
26	9/29/99 6:24	46.12	51	40	30.51	34.2	29.16	244.11		
27	9/29/99 6:40	47.5	56	39	28.96	31.63	27.87	231.66		

Rgn Coeffs | Hourlys | Dailys | Weeklys | Bk Figs | Bk Envelopes | Bk Bul

Ready Sum=0

Excel ANOVA Table Excel ANOVA Table

MultiVarRegression.xls

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.99216266
R Square	0.98438674
Adjusted R Squa	0.98325807
Standard Error	1.11808743
Observations	90

ANOVA

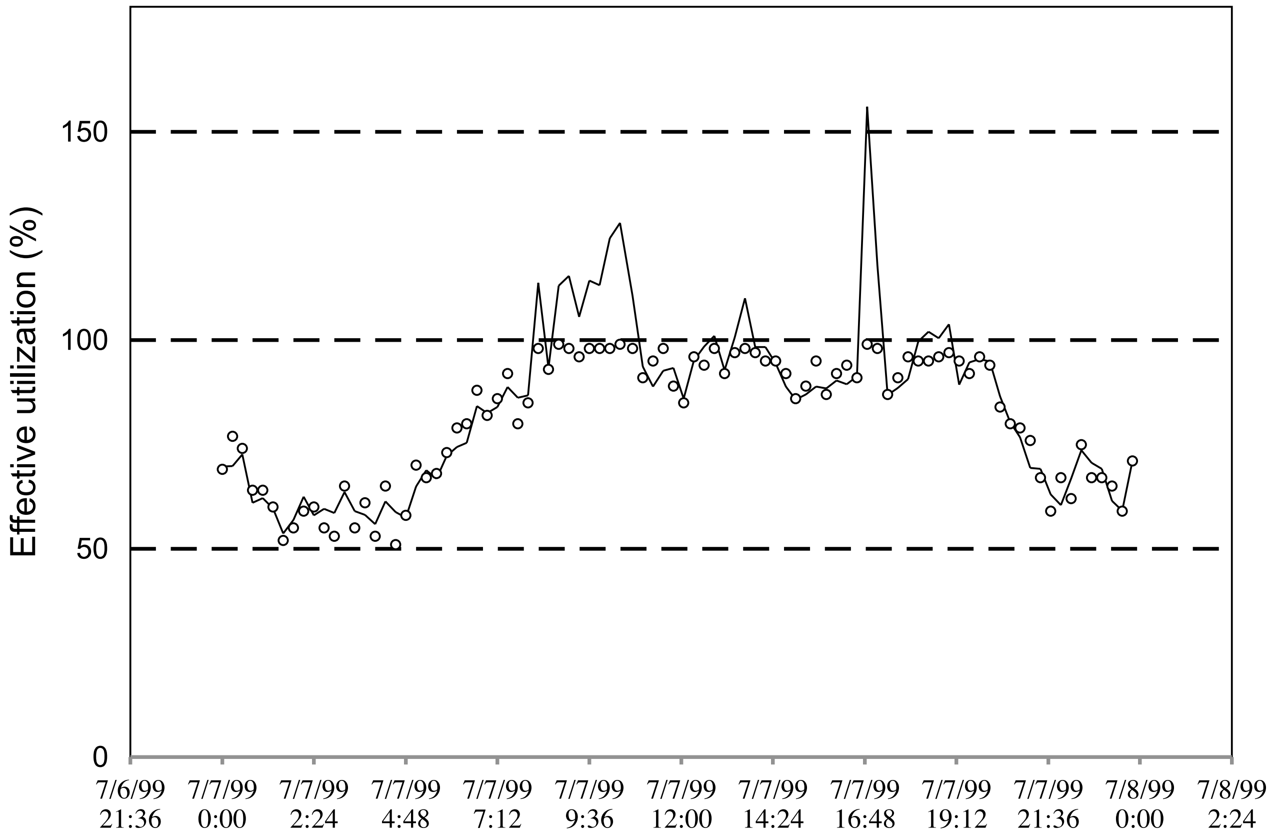
	df	*SS*	*MS*	*F*	*Significance F*
Regression	6	6541.86692	1090.31115	872.165542	9.6049E-73
Residual	83	103.759919	1.2501195		
Total	89	6645.62684			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0*
Intercept	1.3240455	0.75942325	1.74348822	0.08495121	-0.18641722	2.83450822	-0.1864172
MaxU	0.22818773	0.02407508	9.47817141	7.2199E-15	0.18030335	0.27607211	0.1803033
MinU	0.25784123	0.02501249	10.3084996	1.6063E-16	0.20809238	0.30759008	0.2080923
AvgQ	26.0945498	39.6325319	0.65841238	0.51209543	-52.7329836	104.922083	-52.732983
MaxQ	-0.05100836	0.09029699	-0.56489552	0.57366849	-0.23060549	0.12858876	-0.2306054
MinQ	-0.15435786	0.11931225	-1.29373015	0.19934731	-0.3916652	0.08294948	-0.391665
SumQ	-3.14335044	4.95763341	-0.63404253	0.52779665	-13.0038866	6.7171857	-13.003886

Rgn Coeffs | Hourlys | Dailys | Weeklys | Bk Figs | Bk Envelopes | Bk Bı

Ready Sum=0

One Day Sample



Plot of $\hat{y}$ for a 24-hour period shows demand over 1.5 servers at around 16:48 hours.

Summary of Part III

Being able to dig into historical performance information can be very important.

It was critical to being able to build a bridge between mainframe and unix server MIPS (slide 4).

Thats why keeping your own personal performance database was mentioned in Part I.

As the DSS example (slide 13) showed, optimizing response time is sometimes best done by understanding the knees in the throughput characteristic.

Details concerning multivariate regression (slide 23) can be found in Chap. 8 of [4].

References

[1] en.wikipedia.org/wiki/PyramidTechnology,

[2] The TP1 benchmark was a precursor of the current TPC benchmarks. See www.tpc.org/information/about/history.asp

[3] Box, G. E. P., Hunter, W. G., and Hunter, J. S., *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, Wiley 1978

[4] N. J. Gunther, *Guerrilla Capacity Planning*, Springer-Verlag, 2007