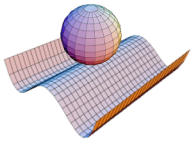
Northern California CMG Meeting
February 7, 2006

Scalability on a Stick



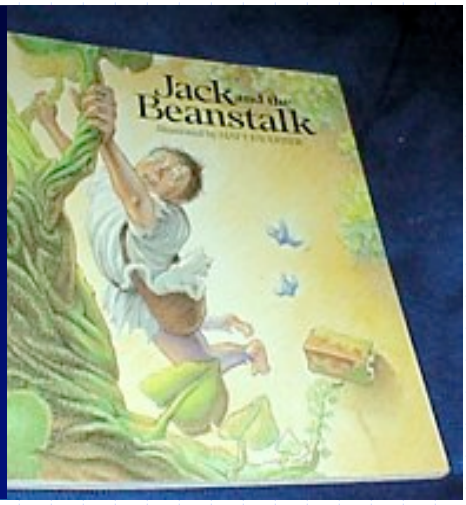
Neil Gunther
Performance Dynamics Company
www.perfdynamics.com

Universal Scalability Law

- Virtual Load Testing
  - Hardware scalability
    - Amdahl's law
    - Universal law of computer scalability
  - Software scalability
    - Same model applies
- Amdahl and the Repairman
  - Theorem of mine from 2002
  - Proves that it is a physical theory
  - Not just curve fitting games

Scaling versus Scalability

- Physical Scaling Laws
  - Load ~ mass $^{1/3}$
  - Limits to giantism
  - Phase transitions
  - Nanotechnology
- Computer Scalability
  - Diminishing returns
  - Internal overheads
  - Multiprocessors
  - Clusters, GRIDs
  - Internet congestion



Scaling Characteristics

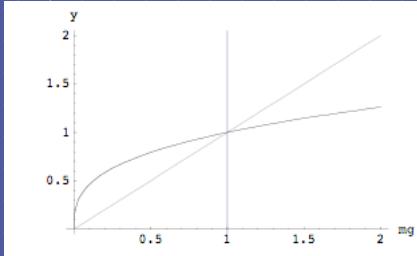


Fig. 4.1. Allometric scaling as a function of weight (mg). As long as the strength (surface area curve shown in *black*), exceeds the weight (volume curve shown in *gray*), growth is sustainable. Beyond the critical point, where the two curves cross, the system will collapse due to lack of mechanical strength

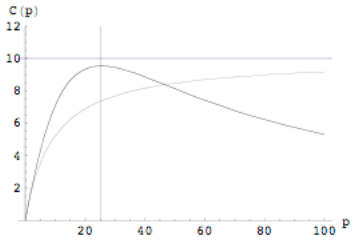


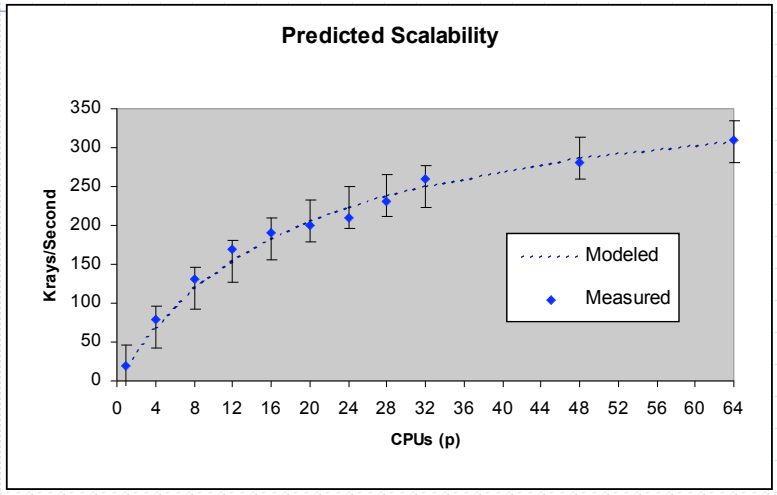
Fig. 4.9. Universal scalability characteristic (*black*) compared with Amdahl scaling (*gray*), which corresponds to a coherency value of $\kappa = 0$ in (4.27). A key feature of universal scaling is that a maximum can develop (here located at $p^* = 25$) depending on the values of σ and κ in (4.28). Comparison with Fig. 4.1 shows that although the system does not fail beyond p^*, its available capacity can degrade significantly

Nature Computer

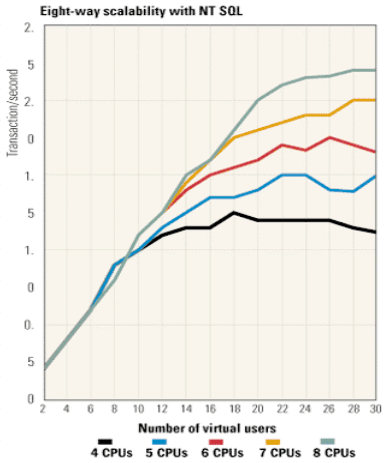
Commonality

- Both hardware and application scalability are two sides of the same coin (more later)
- Hardware
  - Hold the number of **users-per-processor** fixed
  - Capacity function of number of processors (p)
- Software
  - Hold the number of **processors** fixed
  - Capacity is a function of user load (N)

Hardware Scalability (CPUs)



Software Scaling (Users)



Scalability is Not a Number!

- "Extreme Software Scaling", ACM Queue vol.3(no.7), Sep. 2005 by R. MacDougall
  - Demonstrating Amdahl speedup
  - "…max speedup is 75% of linear…"
- Scalability is a function (not a number!)
  - Diminishing returns (due to increasing overhead) appears as **fall** from linearity
  - Want to express this fall from linearity as **quantitative function**

Why Should We Care?

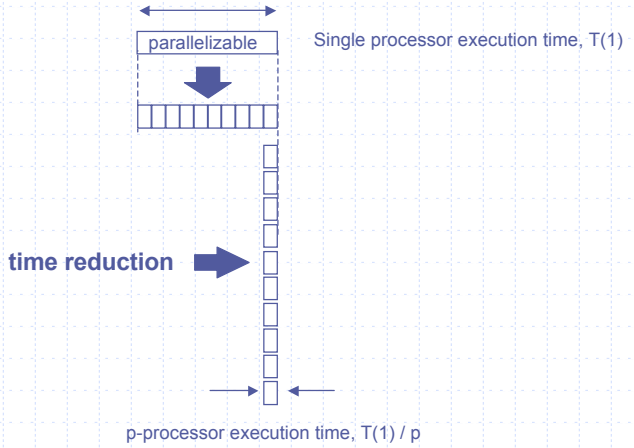
- Old reason: Should always care about performance
- New reason: Multicores have arrived!
  - Intel 'paxville'
    - 2 CPU cores (2 buffers) = 4 virtual CPUs
  - Sun UltraSPARC T
    - 8 CPU cores (4 buffers) = 32 virtual CPUs

Multiprocessing is Hard

- Welcome back to SMPs
  - Just on a smaller scale
  - Faster cycles
- Pyramid Technology
  - Spent a lot of time with ISVs
  - Teach them how to use SMPs
  - Need to understand scalability

Universal Hardware Scalability

Naive Parallelism



Speedup Metric

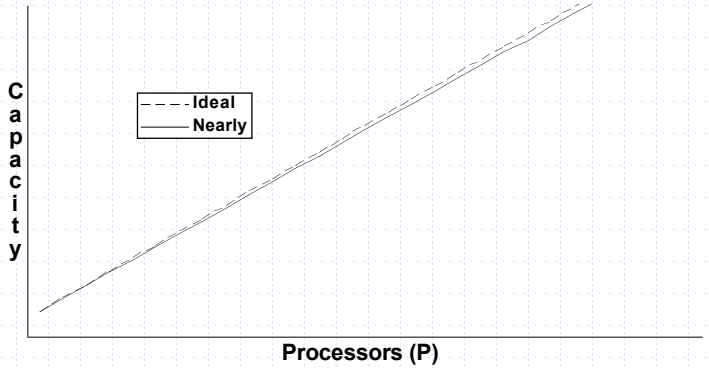
$$S_N(p) = \frac{T(1)}{T(p)}$$

$$= \frac{T}{T/p}$$

$$= p$$

- Naïve parallelism means linear scalability
  - Simple function of p-processors
  - No diminishing returns (unrealistic)
- Mathematical definition of "equal bang for the bucks"

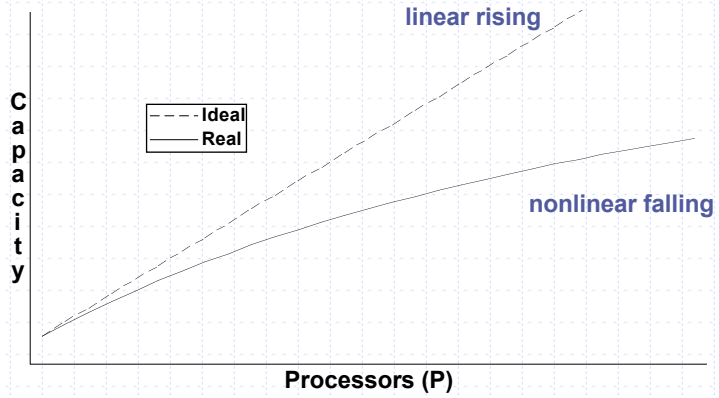
Equal Bang for the Buck



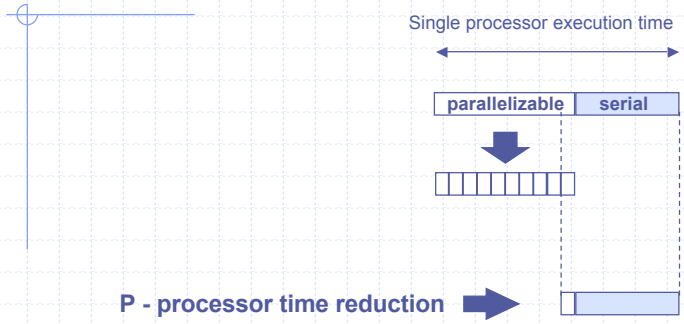
Sources of Serialization

- Ideal SMP scalability is linear but …
- Transaction may wait on:
  - Longer O/S code paths
  - Exchange of shared writeable-data between CPU caches
  - Data exchange between CPUs and main memory
  - Spin lock synchronization (serialization) of accesses to shared writeable data
  - Waiting for an I/O DMA to complete
- Nonlinear scaling more typical for OLTP
- Scalability curve is a function
  - But a function of what?

Effect of Diminishing Returns



Amdahl's Law



Serial portion can only be done by SINGLE processor

The greater the parallelism, the greater the serial fraction

Derivation of Amdahl's Law

- Uniprocessor elapsed time: $T(1) = T$
- Fixed **serial** portion: σT
- Accessable **parallel** portion: $(1 - \sigma)T$
- Elapsed time with p processors:

$$T(p) = \text{Serial portion} + \text{Parallel portion}$$

$$T(p) = \sigma T + (1 - \sigma)T/p$$

Amdahl Speedup

$$S_A(p) = \frac{T(1)}{T(p)}$$

$$= \frac{T}{\sigma T + \frac{(1-\sigma)T}{p}}$$

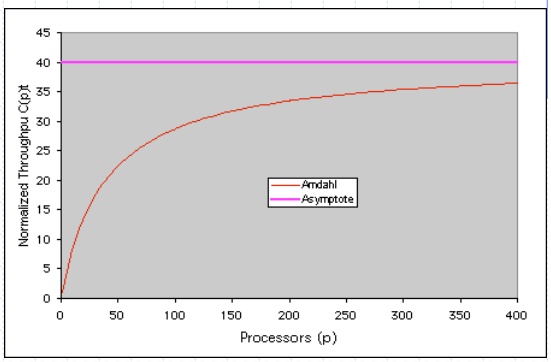
$$= \frac{p}{\sigma p + (1-\sigma)}$$

This equation is known as Amdahl's law

$$\boxed{S_A(p) = \frac{p}{1+\sigma(p-1)}}$$

Amdahl's Speedup Curve

$$S_A(p) = \frac{p}{1+\sigma(p-1)}$$

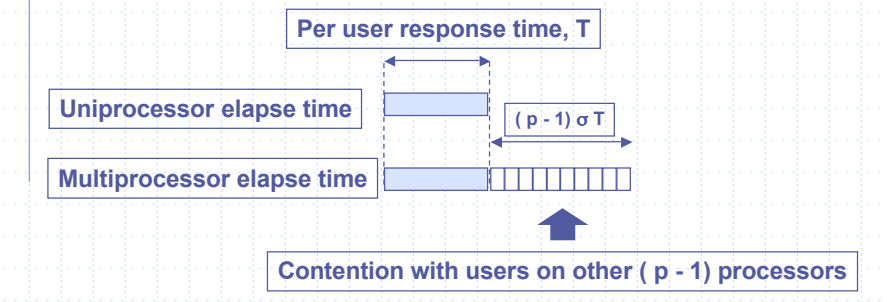


- Diminishing returns are due to the increasing relative serial fraction of the execution time: $\sigma T(1)$
- Serial component of workload dominates at large p

Parallel Scaleup

- Amdahl's law (1967) assumes there is a fixed amount of work which is partitioned across p-processors.
- One way to defeat Amdahl serialization is to increase the amount of work in proportion to the number of processors (demonstrated in 1987).
- Generally very difficult to achieve in practice. Only for special workloads. Can still be defeated by comms overhead.

Multi-user Scaleup



The greater the concurrency, the greater the contention

Derivation of Multi-user Scaleup

- Fixed number of tx's per processor: C
- Uniprocessor elapsed time: T(1) = T (as before)
- Uniprocessor throughput:
  - X(1) = C/T
- Contention from each added processor:
  - σT
- Elapsed time with p processors:
  - $T(p) = T + (p - 1)\,\sigma\,T$
- SMP throughput with p processors:
  - X(p) = p [C/T(p)]

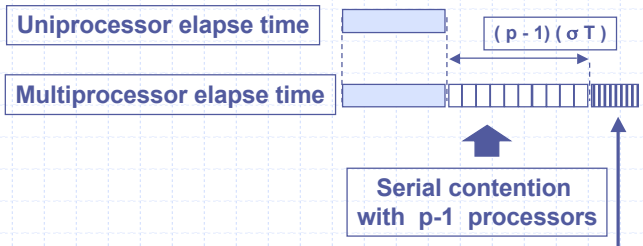
Relative SMP Capacity

$$C(p) = \frac{X(p)}{X(1)} = \frac{pC}{T(p)}\frac{T}{C} = \frac{pT}{T + (p-1)\sigma T} = \frac{p}{1 + \sigma(p-1)}$$

Identical to Amdahl's Law !!!

☹

Super Serial Scaleup



Coherency delay is smaller than serial contention but ...

it grows in proportion to (p x p) processors!

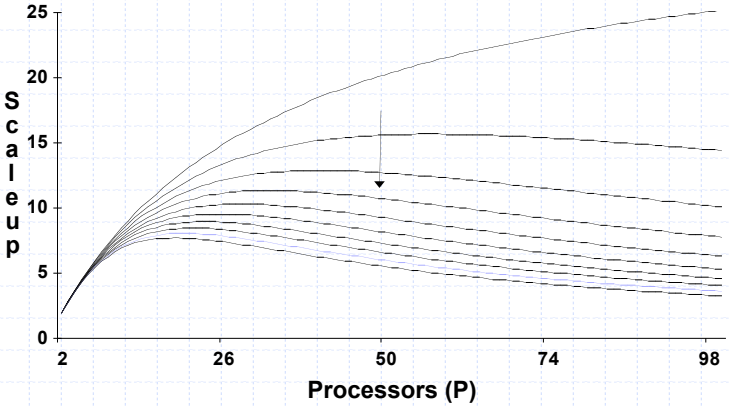
The greater the parallelism, the faster the coherency delay grows!

Super-Serial Capacity

- Fixed number of tx's per user: C
- Uniprocessor elapsed time: T
- Delay for each added processor:
  - $\sigma T + \lambda p T$
- SMP elapsed time:
  - $T(p) = T + (p - 1)\sigma T + \sigma (p - 1) \lambda p T$
- Same algebra as before produces:

$$C(\sigma,\lambda,p) = \frac{p}{1 + \sigma[(p-1) + \lambda p(p-1)]}$$

Super-Serial Characteristic



The downward arrow indicates retrograde reduction in performance as the fraction of coherency work (per processor) increases. Note the appearance of a MAXIMUM in the capacity curve.

Capacity Ceiling

- MAXIMUM throughput occurs at:

$$p_{max} = \left\lfloor \sqrt{\frac{1-\sigma}{\sigma\lambda}} \right\rfloor$$

- **If** $\lambda = 0$, **Pmax** $\rightarrow \infty$:
  Best case (Amdahl scaling; no maximum)
- **As** $\lambda \rightarrow \infty$, **Pmax** $\rightarrow 0$:
  Worst case (moves toward origin)

The Three C's

Concurrency

Degree of inherent "parallelism" in the work

Contention

Degree of serialization in the OLTP workload

Coherency

Degree of induced delay for data consistency maintenance
(analogous to page-faulting in VM memory)

How Do We Find σ and λ ?

- Might we to look at the instrunction level (like Amdahl) to determine parameters. This is the **brute force** method.
- There is a smarter way:
  - Measure throughput X(p) for a **few** (p=4-6) configurations
  - Renormalize these data into appropriate ratios
  - Fit those data as a quadratic function: {a, b, c}
  - Compute the parameters {σ, λ} for the universal scalability model
  - Use them to generate the full scaling curve

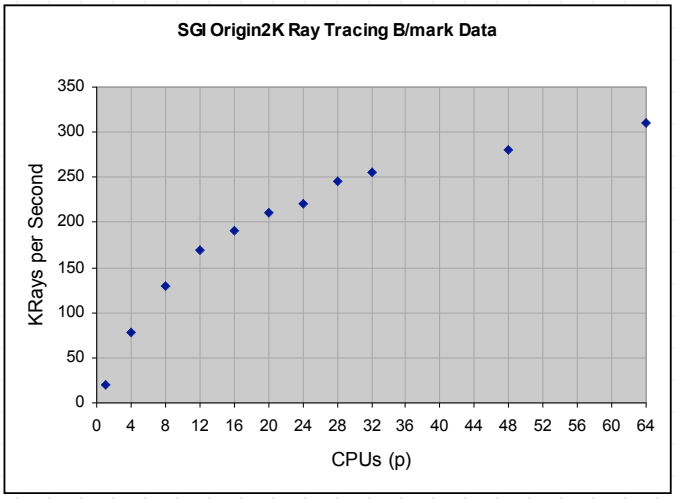
Statistics Is Your Friend

- Find **trends** in raw performance data
- Trends can be used to **predict** growth
- Methods include:
  - Regression
    - Time-independent data
  - Time Series Analysis
    - Time-dependent data ("auto-correlated")
  - ANOVA: ANalysis Of VAriance
    - Quantify degree of interaction between otherwise random performance metrics

Why I Use EXCEL?

- Ubiquitous
  - Cheap
  - Mgmt. desktops already running MS Office
  - Internet news groups for EXCEL/VBA help
- Programmable
  - VBA, object-oriented, debugging, journaling, good prototyping tool
- Statistical analysis functions (Regression, ANOVA, ARIMA)
  - Don't always know which functions are needed
  - Plotting is integrated with VBA
- Printing
  - Single push-button (also has HTML option)
- Use those statistical tools with which you're comfortable

Image Processing Benchmark



Predicting Scalability: Step 1

Measured CPU (p)	KRays/Sec X(p)
1	20
4	78
8	130
12	170
16	190
20	210
24	220
28	245
32	255
48	280
64	310

Step 2: Transformed Data

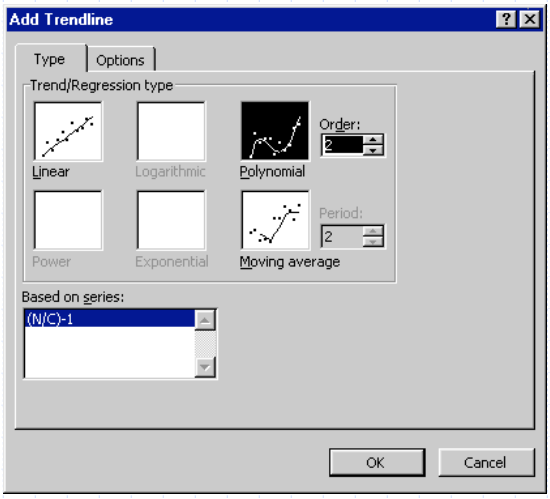
RelCap C=X(p)/X(1)	Efficiency C/p	Inverse p/C
1.00	1.00	1.00
3.90	0.98	1.03
6.50	0.81	1.23
8.50	0.71	1.41
9.50	0.59	1.68
10.50	0.53	1.90
11.00	0.46	2.18
12.25	0.44	2.29
12.75	0.40	2.51
14.00	0.29	3.43
15.50	0.24	4.13

Step 3: Regression Analysis

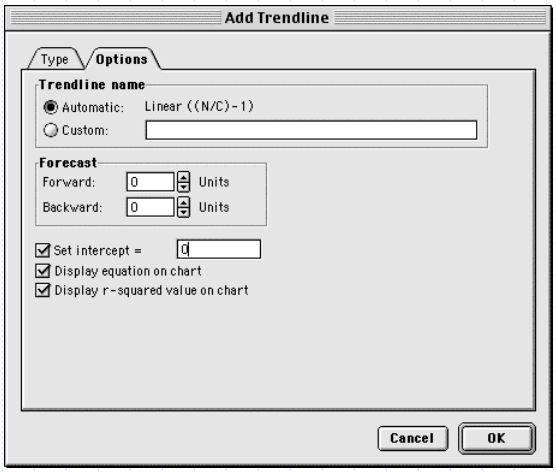
- Choose Polynomial
  - degree 2 (quadratic)
- Check Set intercept
  - Set = 0
- Check Show Equation
- Check Show R^2

Fit p-1	Transform (p/C)-1
0	0.00
3	0.03
7	0.23
11	0.41
15	0.68
19	0.90
23	1.18
27	1.29
31	1.51
47	2.43
63	3.13

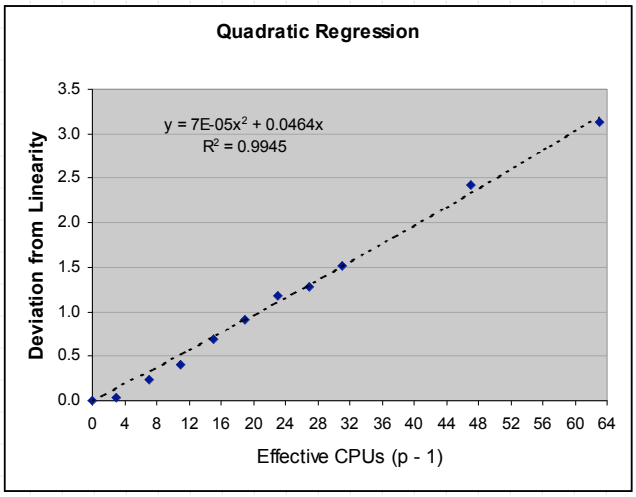
EXCEL – Trendline Type



EXCEL - Trendline Options



Deviation from Linearity



Step 4: Trendline Parameters

- Enter Trendline coefficients in EXCEL
- Note that coefficient c = 0

Trendline Quadratic	Parameters Coefficients
a	7.0000E-05
b	0.0464
c	0.0000

Parameter Relationships

$$\sigma = b - a$$

$$\lambda = \frac{a}{b - a}$$

$$p_{max} = \text{Floor}\left(\sqrt{\frac{1-\sigma}{\sigma\lambda}}\right)$$

$$p_{opt} = \frac{1}{\sigma}$$

Step 5: Model Parameters

Super Parameter	Serial Values
σ	0.0464
λ	0.0002
Pmax	377
Popt	22

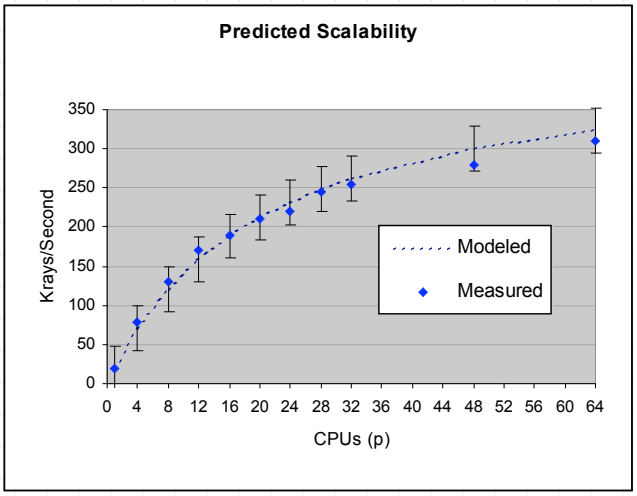
← Contention
← Coherency

Step 6: Predictions

p	Predicted C(p)	Capacity Modeled	Measured	%Error
1	1.00	20.00	20	0.00
4	3.51	70.22	78	-9.97
8	6.04	120.74	130	-7.12
12	7.94	158.81	170	-6.58
16	9.43	188.50	190	-0.79
20	10.61	212.30	210	1.10
24	11.59	231.78	220	5.36
28	12.40	248.02	245	1.23
32	13.09	261.75	255	2.65
48	15.02	300.35	280	7.27
64	16.20	323.97	310	4.51

Negative errors => model *underestimates* data

Predicted Hardware Scalability



Conclusions

- Error < 10% over 64-way (good)
- Contention (σ): 4.64%
  - This is relatively **high**
  - Due to NUMA bus, compiler, etc.
- Coherency (λ): 0.02%
  - Relatively **low**
  - Few Cache misses, no paging, etc.
- Pmax: 377 is physically unachievable
- Popt: 22 poor for this application on a 64-way SMP

Summary of Technique

- Measure appln. throughput X(p) vs. config (p)
- A sparse data sample (at least 4 data points)
- Calculate the capacity ratio C(p), the efficiency C/p, and its inverse p/C, from the data.
- Calculate the Quadratic transform
- Perform regression fit on the Quadratic transform
- Use EXCEL Trendline + Options dialog box
- Calculate the parameters {σ, λ} from the regression coefficients {a, b, c = 0}
- Use the values of σ and λ to predict the complete scalability function, $C(\sigma, \lambda, p)$

Why Do I Invert the Data?

- EXCEL is too **dumb** to fit data to a nonlinear equation like my universal scaling equation (USL)
- More sophisticated tools like Mathematica, S+ and R **can** do it
- Inverting the data corresponds to inverting my USL formula
- Just makes it easy for EXCEL to fit a 2nd degree polynomial

Universal Software Scalability

General Application Scalability

- Rename variables and parameters
  - p → N
  - σ → α
  - λ → β
- Otherwise, the same as before

$$C(N,\alpha,\beta) = \frac{N}{1+\sigma[(N-1)+\lambda N(N-1)]}$$

CASE STUDY (I): Login Certification for CRM

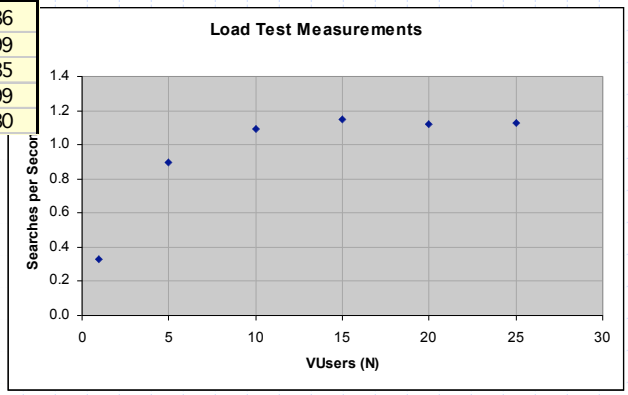
Load test measurements using Load Runner™ from Mercury-Interactive at a well-known networking company in Silicon Valley

LoadRunner™ 7.5 Settings

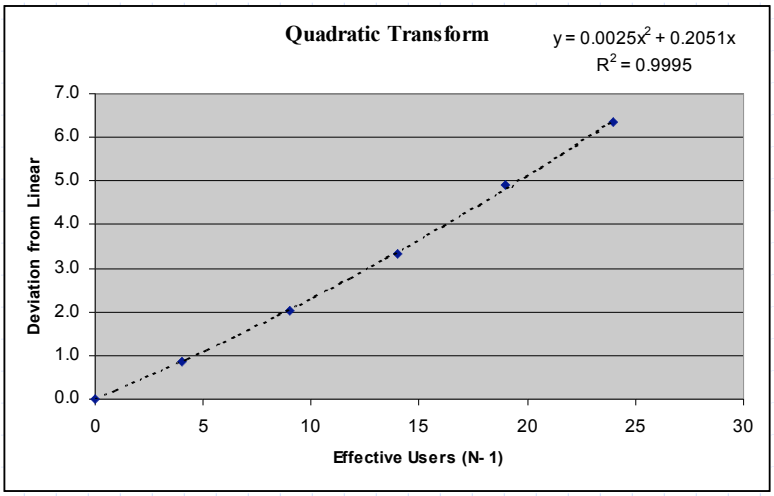
- CRM application
  - User login/authentification
  - User searches for customer parameters (ORACLE)
- Transaction definition
  - Login (1-shot) is incorporated in Init part of LR script
  - Iterate on specific searches being evaluated
  - Mean TPS calculated as Action_Tx / Duration_Seconds
- Each VU load must reach steady state
  - 10-15 mins. is common runtime per VU load
  - Use Rendezvous Start/Stop VU's @ 15 mins.
  - Use Goal mode rather than Scenario mode (new in LR 7.5)

Throughput Measurements

Measured VUs (N)	TPS X(N)
1	0.3311
5	0.8986
10	1.0899
15	1.1485
20	1.1199
25	1.1280



Regression Analysis



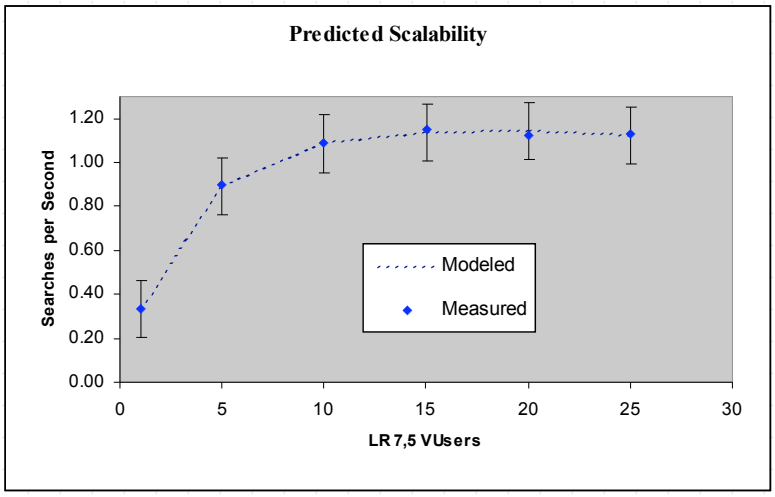
Parameters Mappings

Trendline Quadratic	Parameters Coefficients	Super Parameter	Serial Values
a	2.50E-03	α	0.2026
b	0.2051	β	0.0123
c	0.0000	**Nmax**	**19**
		Nopt	**5**

Predicted Scalability

VUs	Predicted C(N)	Capacity Modeled	Measured
1	1.00	0.3311	0.3311
5	2.69	0.8899	0.8986
10	3.28	1.0861	1.0899
15	3.44	1.1387	1.1485
20	3.45	1.1418	1.1199
25	3.40	1.1243	1.1280

Predicted and Modeled X(N)

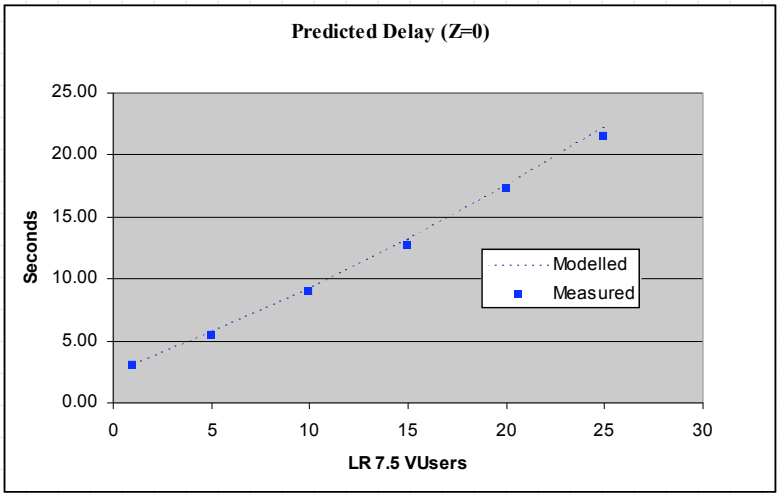


Response Time Measurements

VU	Predicted Modelled	Delay Measured	Percent Error
1	3.02	2.96	2.03
5	5.62	5.44	3.29
10	9.21	8.94	3.00
15	13.17	12.69	3.80
20	17.52	17.23	1.66
25	22.24	21.44	3.71

$$R(N) = \frac{N}{X(N)} - Z$$

Predicted Application Delay



Conclusions

- Error < 2% on **throughput** (v. good)
- Error < 4% on **latency** (excellent!)
- Contention (α): 20.26%
  - Extremely **high**
  - ODBC calls need serious revision!
- Coherency (β): 1.23%
  - Relatively **high**
  - DBMS cache misses?
- Nmax: 19 is severely bottlenecked
- Nopt: 5 users is untenable in production

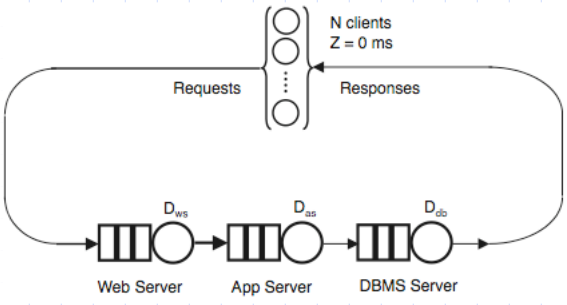
CASE STUDY (II): Multi-tier Scalability

Based on a CMG paper

Reference

- Based on CMG 2001 paper by Buch & Pentkovski, "*Experience of Characterization of Typical Multi-tier e-Business Systems Using Operational Analysis*"
- This is an excellent paper
- I analyze this model using PDQ in my other book "Analyzing Performance with Perl::PDQ", Springer 2005
- Here, I use the **Universal Scaling Law** instead (much easier but not as insightful)

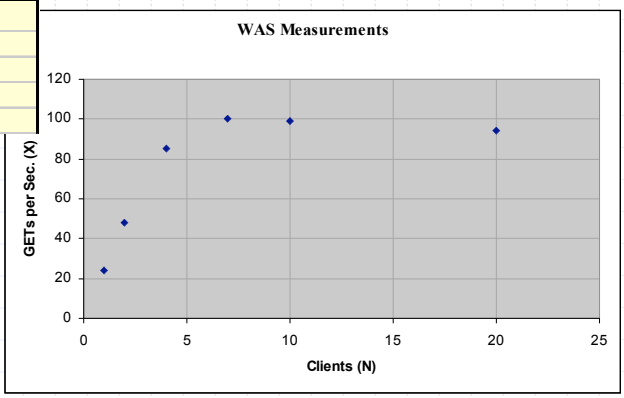
Multi-tier Model



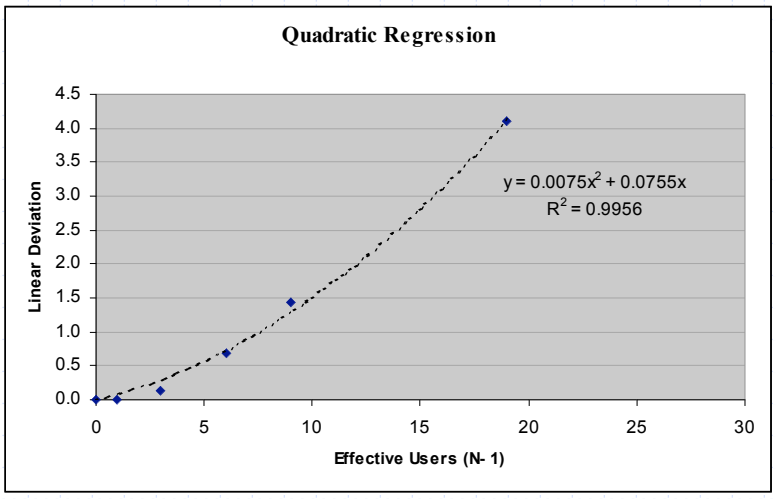
- Classic 3-tier architecture
- Only show 1 queue per tier
- Measured using MS-WAS (not LR)

Measured Throughput

VUs (N)	X(N)
1	24
2	48
4	85
7	100
10	99
20	94



Deviation from Linearity



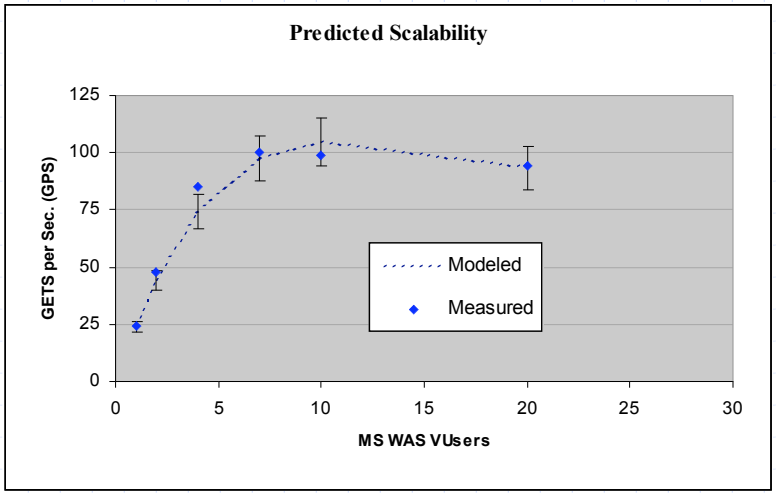
Model Parameters

Trendline Quadratic	Parameters Coefficients	Super Parameter	Serial Values
a	7.50E-03	α	0.0680
b	0.0755	β	0.1103
c	0.0000	**Nmax**	**10**
		Nopt	**15**

Predicted Throughput

VUs	Predicted C(N)	Capacity Modeled	Measured	Percent Error
1	1.00	24.00	24	0.00
2	1.85	44.32	48	7.66
4	3.09	74.19	85	12.72
7	4.06	97.50	100	2.50
10	4.37	104.94	99	-6.00
20	3.89	93.35	94	0.69

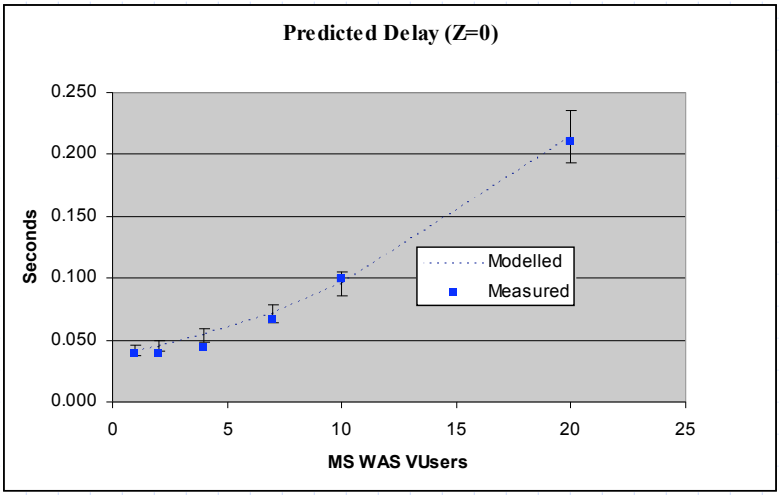
Retrograde Scalability



Predicted Response Times

VU	Predicted Modelled	Delay Measured	Percent Error
1	0.042	0.039	6.84
2	0.045	0.039	15.71
4	0.054	0.044	22.54
7	0.072	0.067	7.15
10	0.095	0.099	-3.75
20	0.214	0.210	2.02

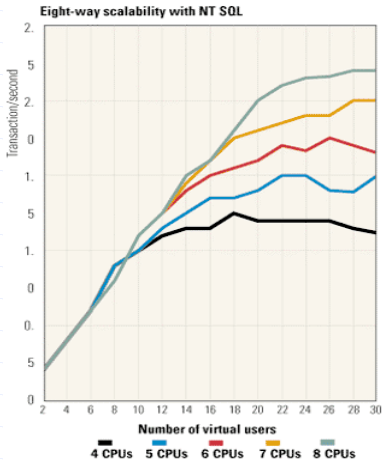
Response Scalability



Conclusions

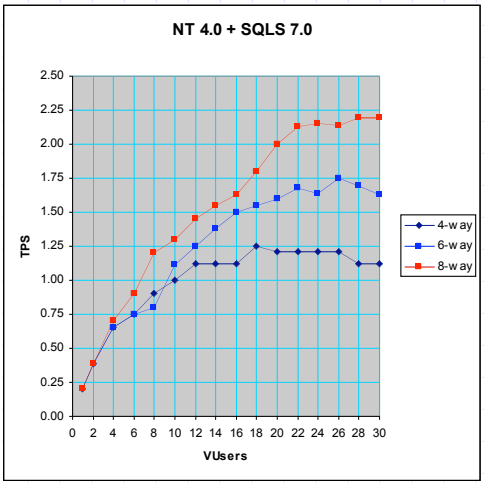
- Contention (α): 6.8%
  - Very **high**
  - Middleware need serious tuning
- Coherency (β): 11.3%
  - Incredibly **high**!
  - Some kind of thrashing is occurring
- Nmax: 10 is poor
- Nopt: 15 can't be achieved
- Recommend you compare their CMG 2001 paper with Chap. 10 in my PPDQ book

Application Scaling Example



- Wintel platform
- NT 4.0 O/S
- MS-SQLS
- Fixed CPU configs
- Vary Vuser load

SQLServer Scalability Plotted



4-Way Parameters

Trendline Quadratic	Parameters Coefficients	Super Parameter	Serial Values
a	2.30E-03	α	0.0790
b	0.0813	β	0.0291
c	0.0000	Nmax	20
		Nopt	13

- After performing the same kind of regression analysis we did for hardware scalability

6-Way Parameters

Trendline Quadratic	Parameters Coefficients	Super Parameter	Serial Values
a	2.00E-04	α	0.0802
b	0.0804	β	0.0025
c	0.0000	Nmax	70
		Nopt	13

8-Way Parameters

Trendline Quadratic	Parameters Coefficients	Super Parameter	Serial Values
a	2.00E-05	α	0.0566
b	0.0566	β	0.0004
c	0.0000	Nmax	223
		Nopt	18

Modeled Scalability Curves

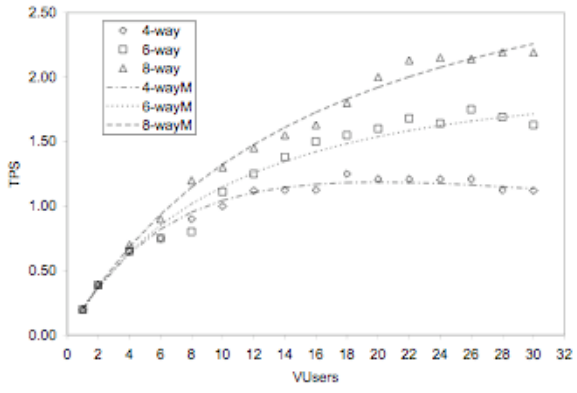


Fig. 6.5. Universal scalability models (*dashed lines*) for the data in Fig. 6.4

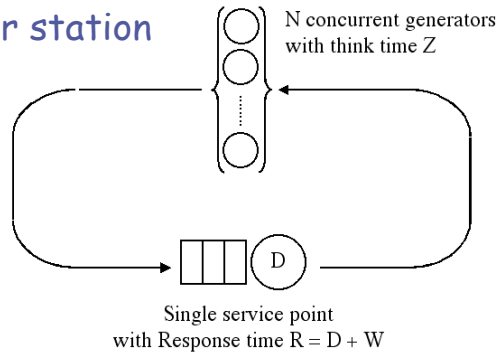
Amdahl and the Repairman

Theorem: Amdahl's law for parallel speedup is equivalent to the synchronous throughput bound of the repairman queueing model

--njg (2002)

Classic Repairman Model

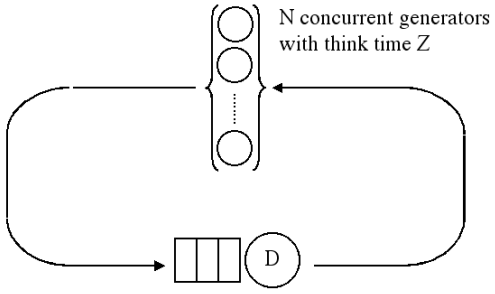
- Machines in assembly line
  - Some are "up" and working
  - Some are "down" for repairs
- Repair station



Single service point
with Response time R = D + W

Multiuser (Timeshare) Model

- Machines -> users at their terminals
- Repair station -> computing resource

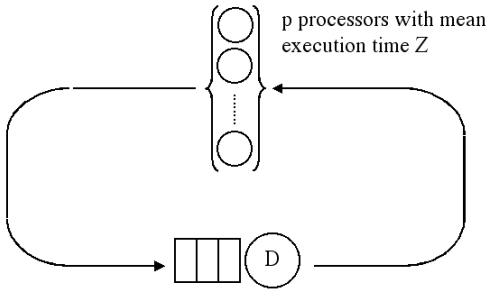


Single service point
with Response time R = D + W

79

Multiprocessor Bus Model

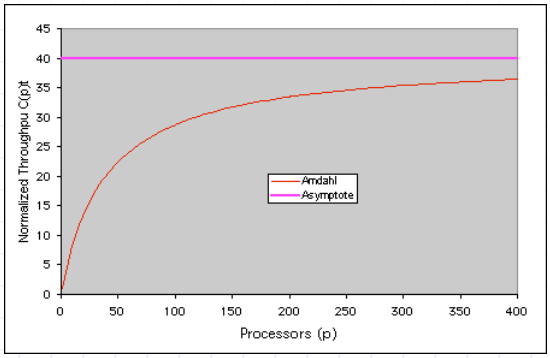
- Machines -> Processors
- Repair station -> communication bus



Bus interconnect with
latency R = D + W

80

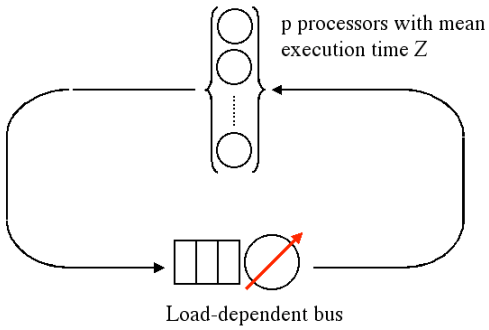
Synchronous Queueing



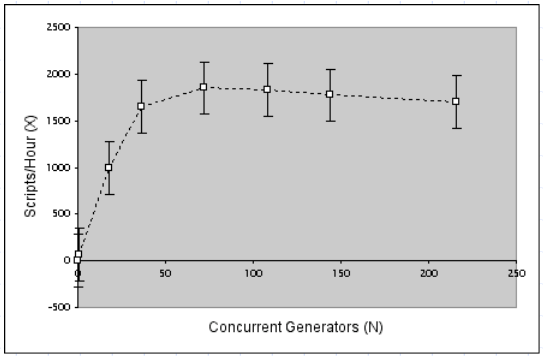
- Amdahl's law == Synchronous queueing bound
- Serial fraction: $\sigma = D / (D + Z)$

Load-Dependent Bus

- Service rate is a function of the load
- Load measured by queue length



Retrograde Throughput



Commonly seen in application load testing
Difficult to model in queueing theory

Virtual Load Testing

1. Measure appln. throughput X(N) vs. load (N)
2. A sparse data sample (> 4 load points) is OK
3. Calculate the capacity ratio C(N), the efficiency C/N, and its inverse N/C, from the data.
4. Calculate the Quadratic transform
5. Perform regression fit on the Quadratic transform
6. Use EXCEL Trendline + Options dialog box
7. Calculate the parameters {α, β} from the regression coefficients {a, b, c = 0}
8. Use the values of α and β to predict the complete scalability function, C(N)

Bibliography

- G. Amdahl, *Validity to the Single Processor Approach to Achieving Large Scale Computing Capabilities*, AFIPS Conference, 1967
- N. Gunther, *A New Interpretation of Amdahl's law and Geometric Scalability*, http://xxx.lanl.gov/ October 2002
- N. Gunther, Guerrilla Capacity Planning, Springer-Verlag, 2006 (in press)
- **The Guerrilla Manual** is always available on my website www.perfdynamics.com