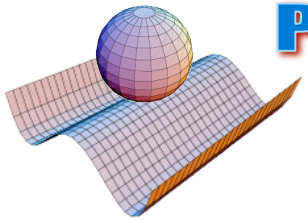
Why Are There No Giants?

The Data Analytics of Scalability

Neil J. Gunther @DrQz

Performance Dynamics

Bay Area R Users Group (BARUG) Meetup
September 18, 2018

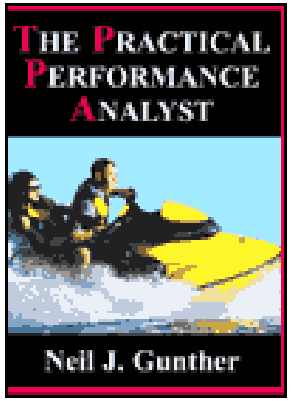


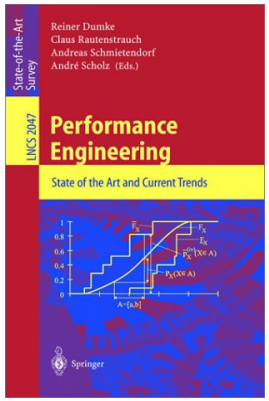
Performance Dynamics Company℠

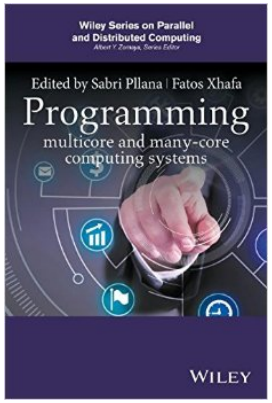
My vitals

- Consultant, researcher, author, teacher
- Ph.D. (U.K.) in theoretical physics
- NASA/JPL Voyager (still going!), Galileo deep space missions
- Xerox PARC
- Pyramid Technology Corp
- Performance Dynamics Company
- Using R for past 10 years — practically every day!
- Developed PDQ (pretty damn quick) queueing analyzer — now in R
- Developed the USL (universal scalability law) — R pkgs used in this talk

Wrote some things







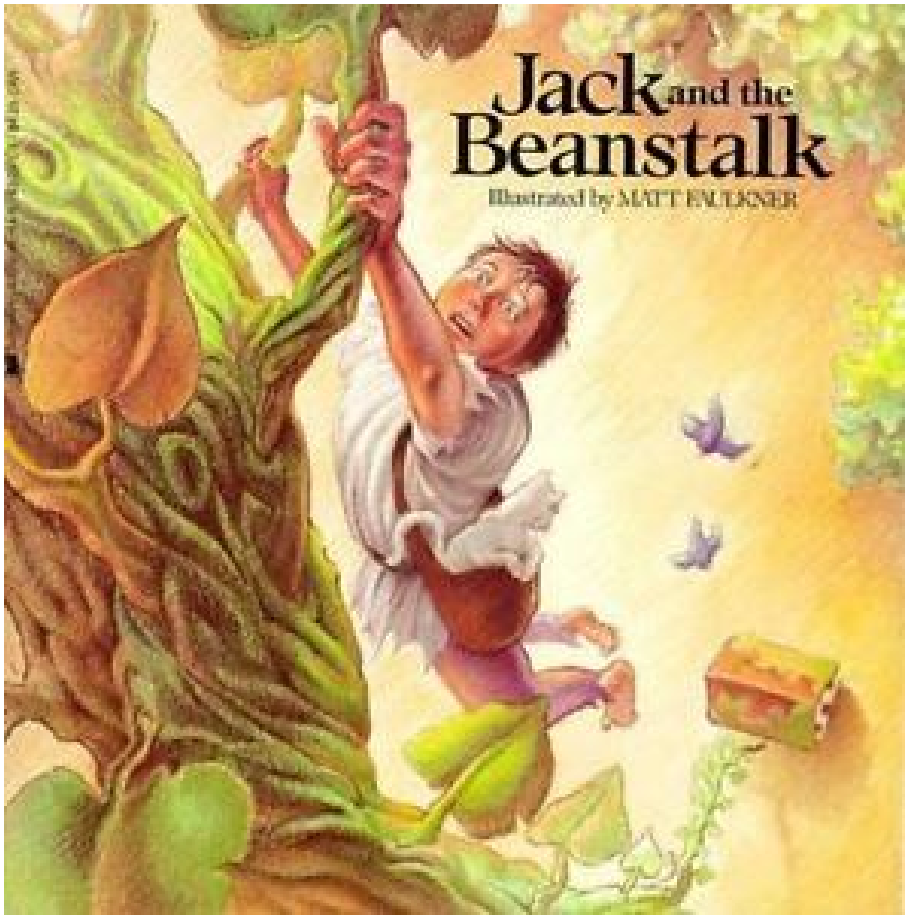
HOME | BLOG | TRAINING | BOOKS | PAPERS | PDQ | GAPHORISMS | SCALABILITY | FACEBOOK | CONTACT

Performance Ponderings

Contents

Outline

Mythical giants (and beanstalks)



- J&B giant was reputedly about 30′ tall in some accounts (no data)
- Let's not even get into beanstalks in the clouds!

Galileo was onto this

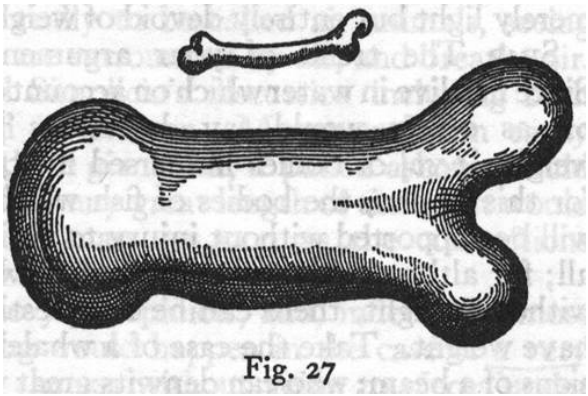


Fig. 27

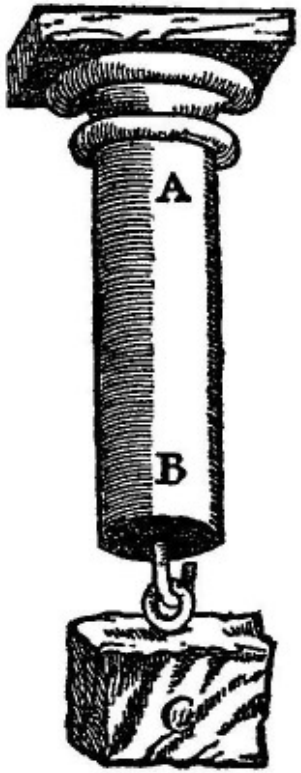
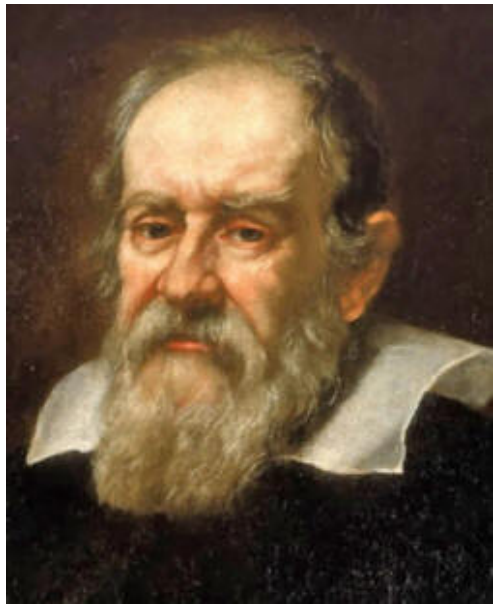


Fig. 1

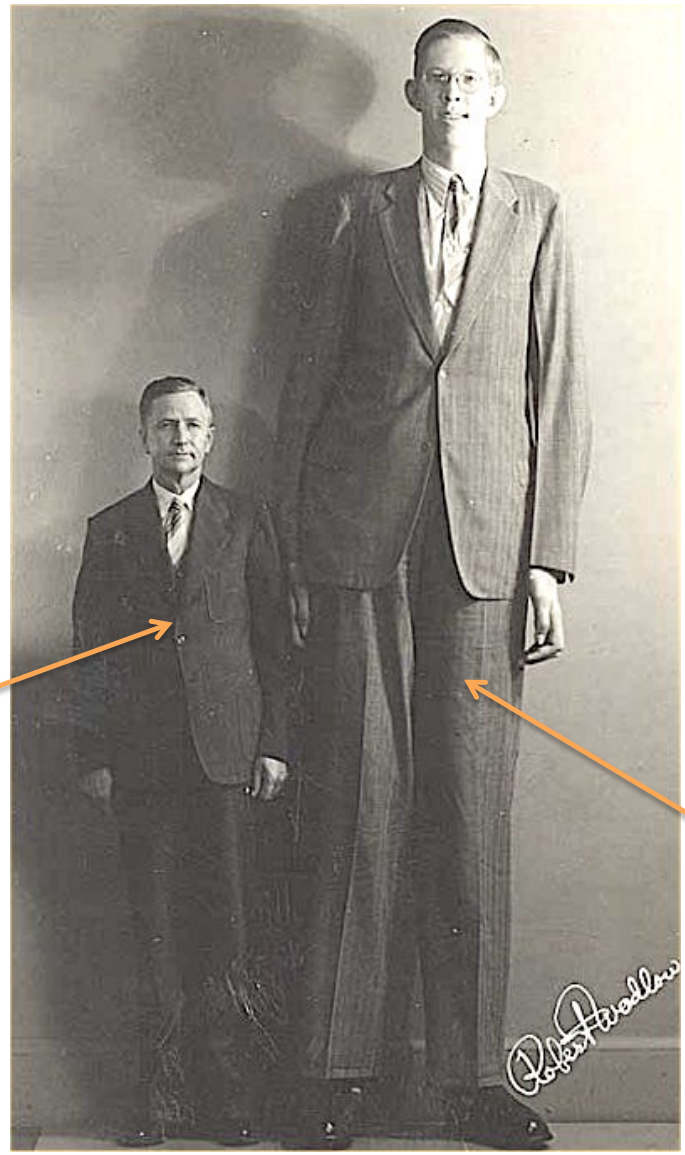


1. Discorsi e Dimostrazioni Matematiche Intorno a Due Nuove Scienze[1], Galileo Galilei (1638)
2. On Being the Right Size, J. B. S. Haldane (1928)

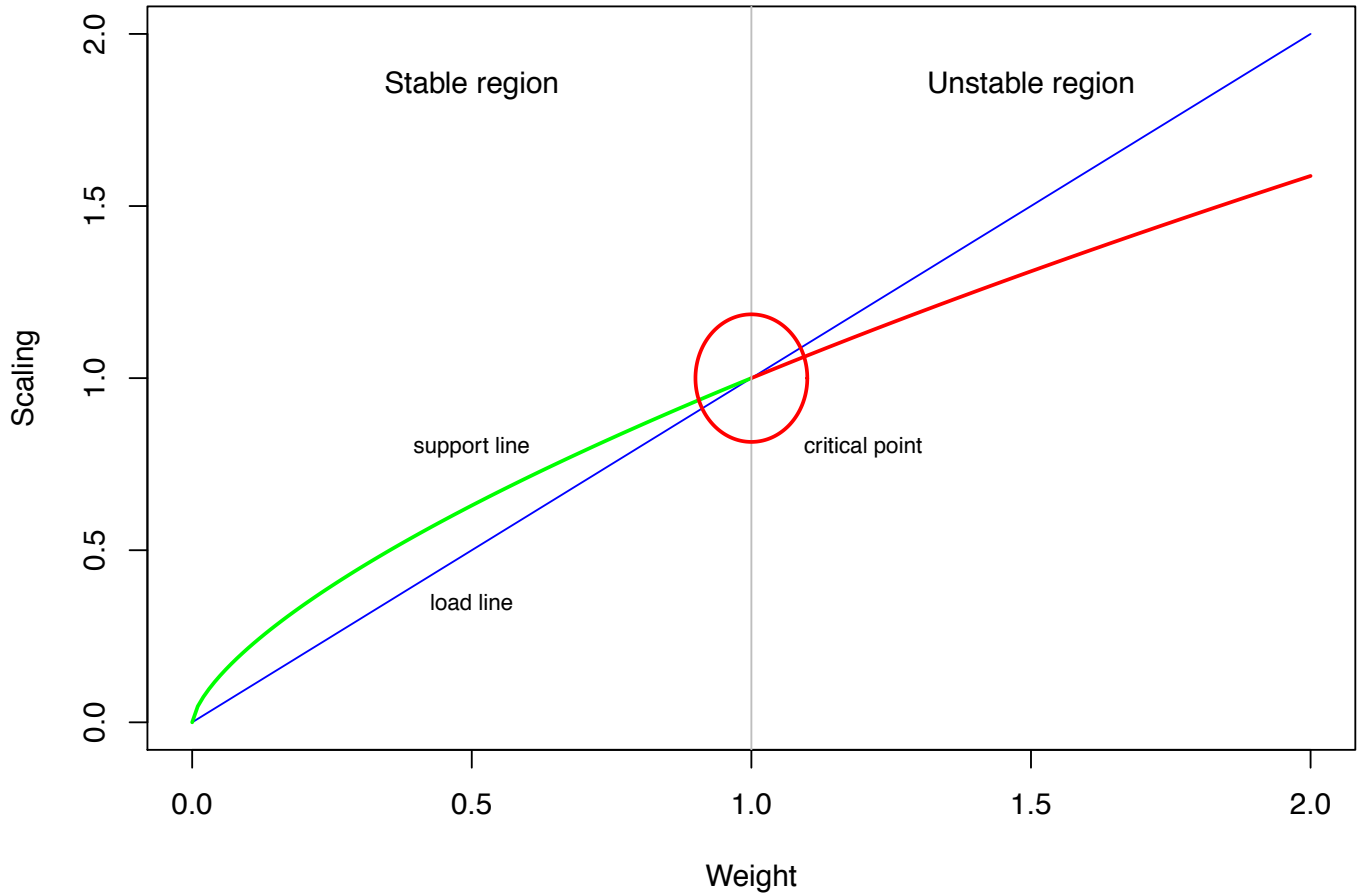
[1] The two new sciences are (i) strength of materials and (ii) projectile motion.

Allometric scaling

- Robert P. Wadlow
- Tallest human giant
- b. Alton, Illinois in 1918
- Reached $8'\ 11.1''$ (2.72 meters)
- Guinness world record
- Died in 1940 at 22 yo
- Father was $5'\ 11.5''$ (1.82 meters)



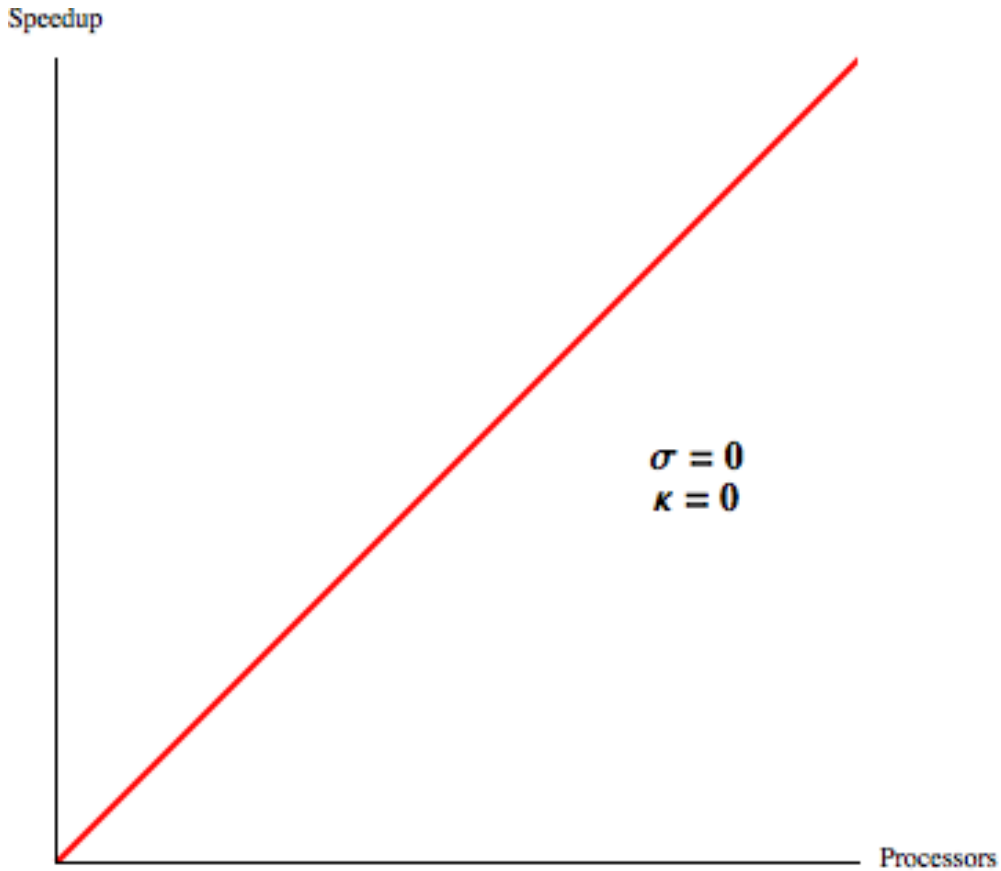
Why there are no giants



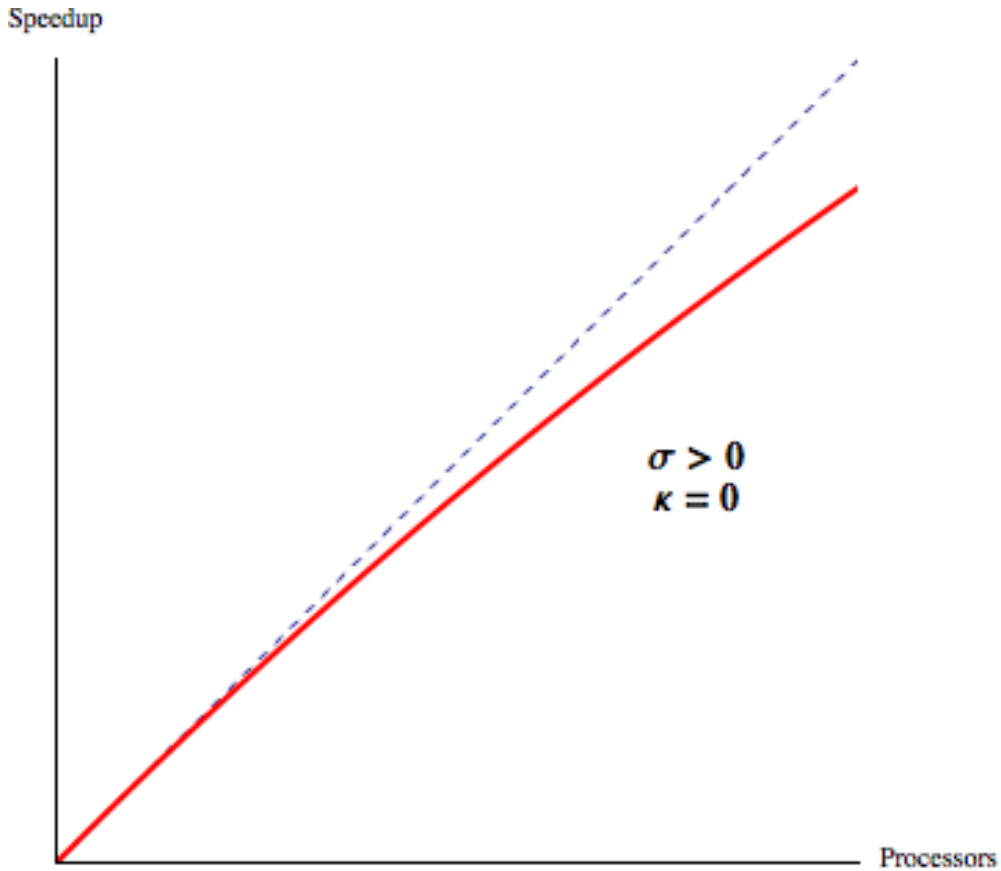
1. Weight (mass) grows like volume: $\sim L^3$
2. Support only grows like cross-sectional area: $\sim L^2$

Outline

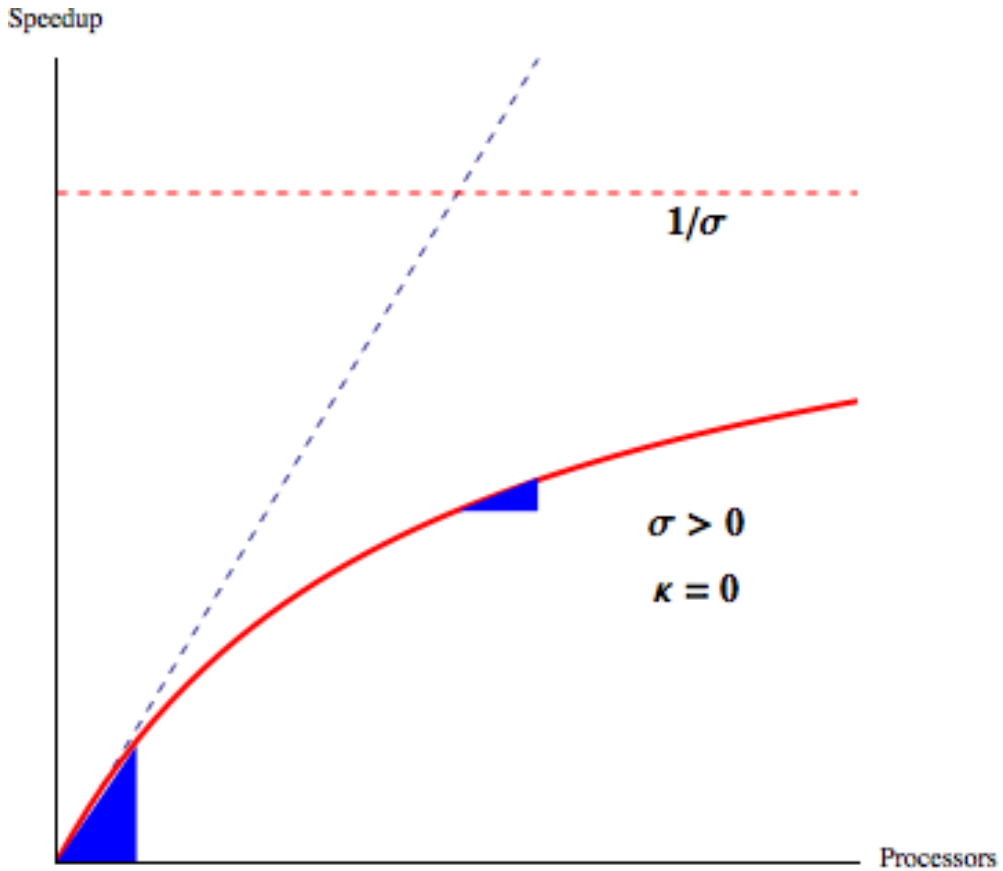
Equal bang for your buck: Concurrency



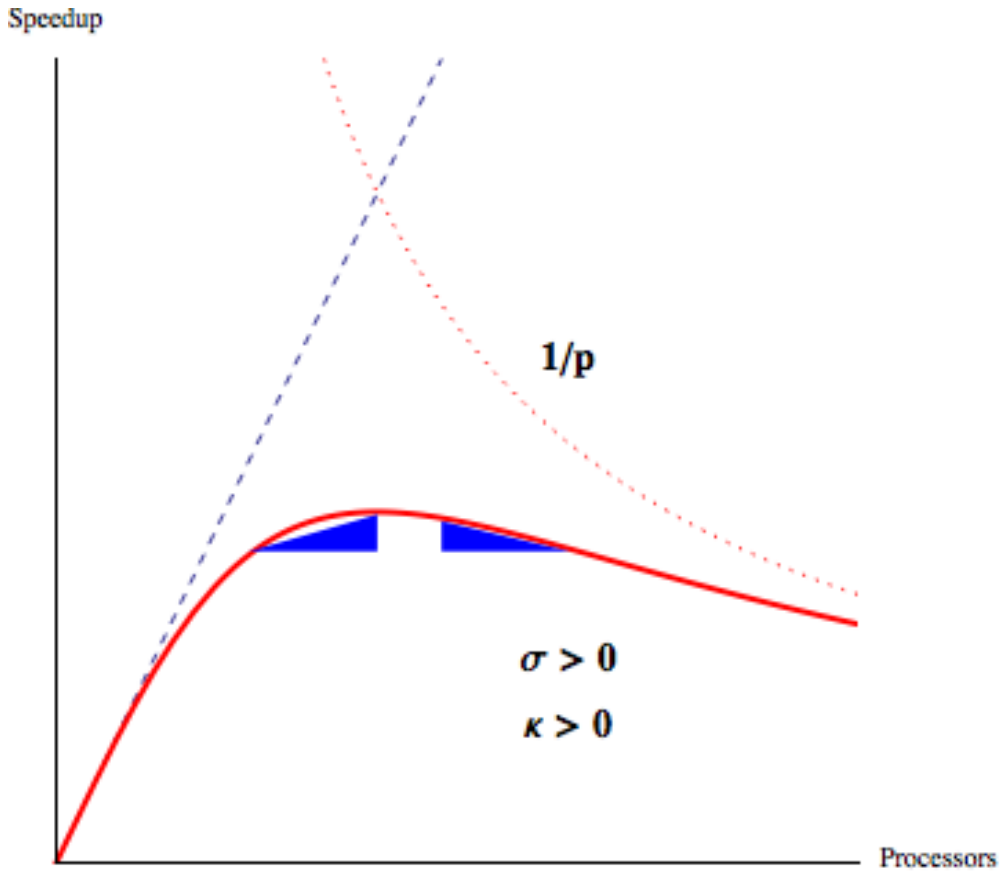
Diminishing returns: Contention cost



Resource saturation: More contention cost



Negative returns: Coherency of distributed data



Outline

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha\,(N-1) + \beta\,N(N-1)}$$

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha\,(N-1) + \beta\,N(N-1)}$$

The three Cs:

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha(N-1) + \beta N(N-1)}$$

The three Cs:

1. **C**oncurrency ($1 < \gamma < \infty$)

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha(N-1) + \beta N(N-1)}$$

The three Cs:

1. **C**oncurrency ($1 < \gamma < \infty$)
2. **C**ontention ($0 < \alpha < 1$)

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Model: Universal scalability law (USL)

N: processes provide system stimulus or **load**

C_N: response function or **relative capacity**

Question: What kind of function?

Answer: Nonlinear rational function[2]

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha(N-1) + \beta N(N-1)}$$

The three Cs:

1. **C**oncurrency ($1 < \gamma < \infty$)
2. **C**ontention ($0 < \alpha < 1$)
3. **C**oherency ($0 < \beta < \infty$)

[2] NJG. "A Simple Capacity Model of Massively Parallel Transaction Systems," *CMG Conf.* (1993)

Measurement meets model

$$\underbrace{\frac{X(N)}{X(1)}}_{\text{Thruput data}}$$

Measurement meets model

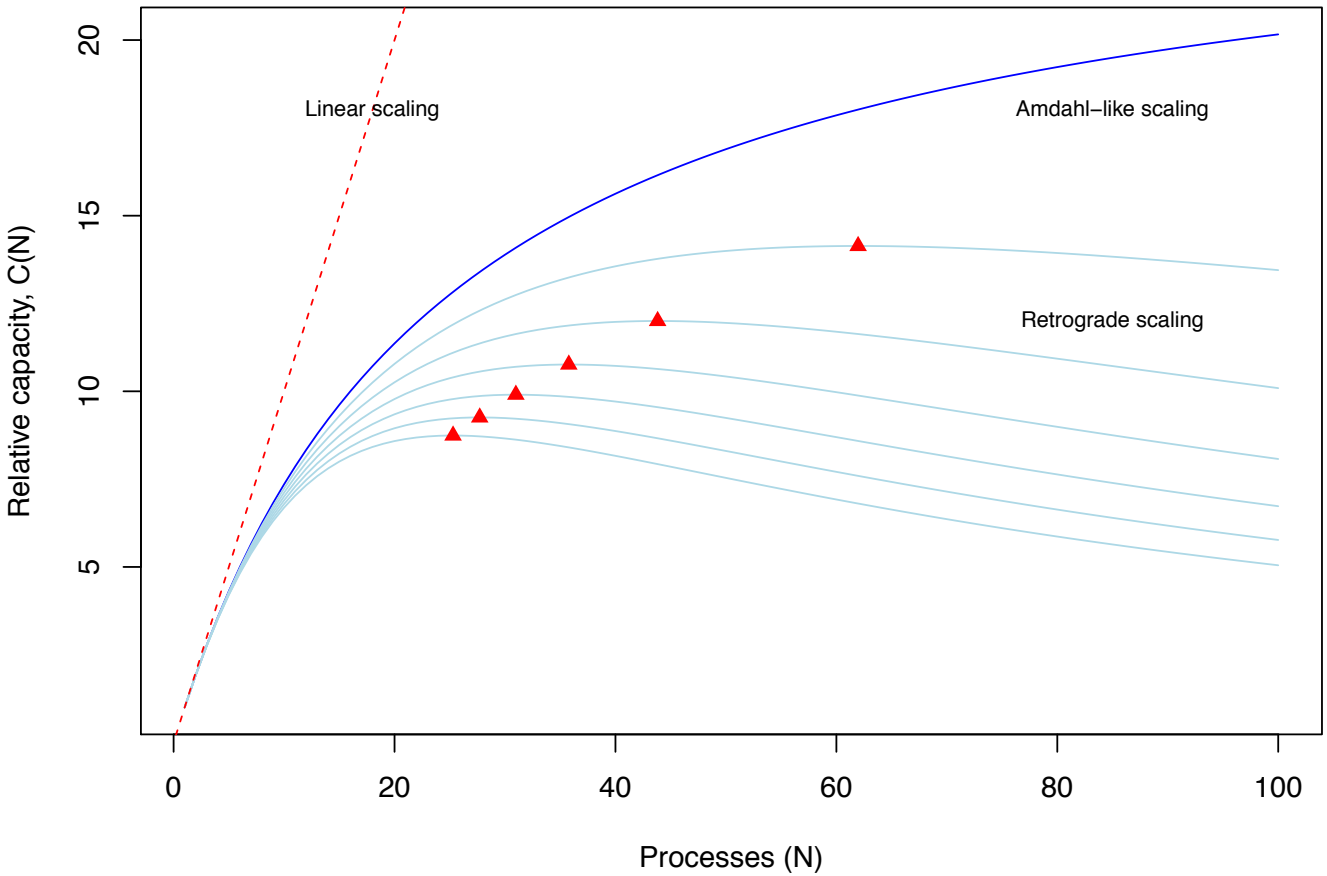
$$\underbrace{\frac{X(N)}{X(1)}}_{\text{Thruput data}} \longrightarrow \underbrace{C_N}_{\text{Scalability}}$$

Measurement meets model

$$\underbrace{\frac{X(N)}{X(1)}}_{\text{Thruput data}} \longrightarrow \underbrace{C_N}_{\text{Scalability}} \longleftarrow \underbrace{\frac{\gamma N}{1 + \alpha (N-1) + \beta N(N-1)}}_{\text{USL model}}$$

Measurement meets model

$$\underbrace{\frac{X(N)}{X(1)}}_{\text{Thruput data}} \longrightarrow \underbrace{C_N}_{\text{Scalability}} \longleftarrow \underbrace{\frac{\gamma N}{1 + \alpha (N-1) + \beta N(N-1)}}_{\text{USL model}}$$



Outline

Determining α, β, γ coefficients

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha (N-1) + \beta N(N-1)}$$

- Brute force measurements (good luck!)

Determining α, β, γ coefficients

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha (N-1) + \beta N(N-1)}$$

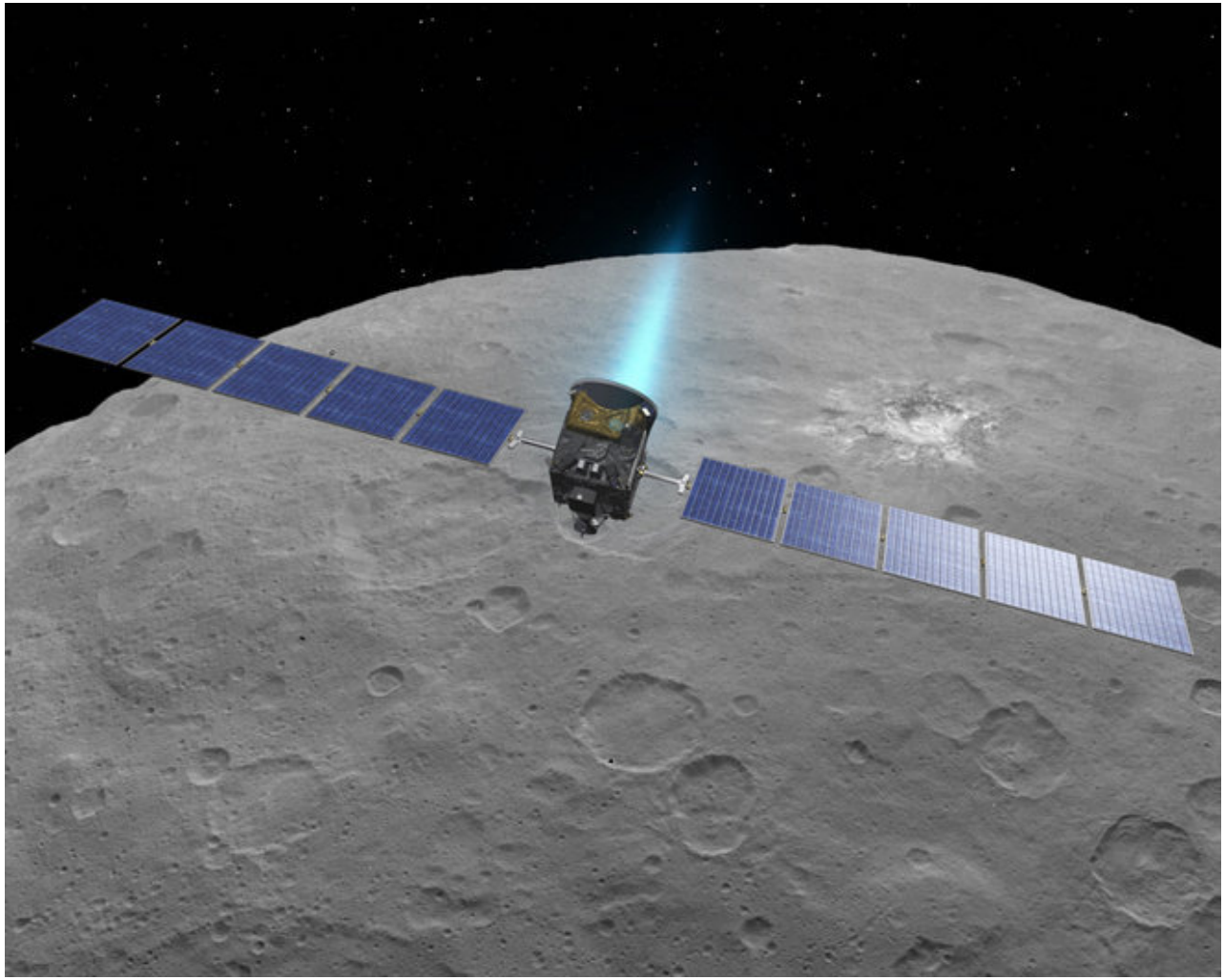
- Brute force measurements (good luck!)
- **Clever way:** Apply statistical regression

Determining α, β, γ coefficients

$$C_N(\alpha, \beta, \gamma) = \frac{\gamma N}{1 + \alpha (N-1) + \beta N(N-1)}$$

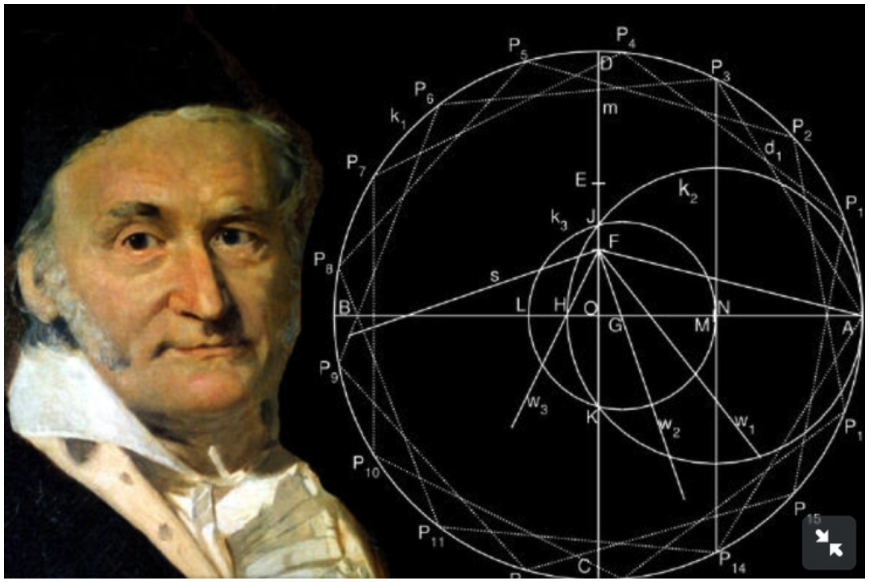
- Brute force measurements (good luck!)
- **Clever way:** Apply statistical regression
- Magic functions in **R**:
  - `nls()` **nonlinear regression** $\rightarrow$ α, β, γ in one swell foop
  - `predict()` smooth interpolation/extrapolation from data
  - `plot()` cleanest for iterative assessment
  - `ggplot()` with all its bells and whistles

Least squares history



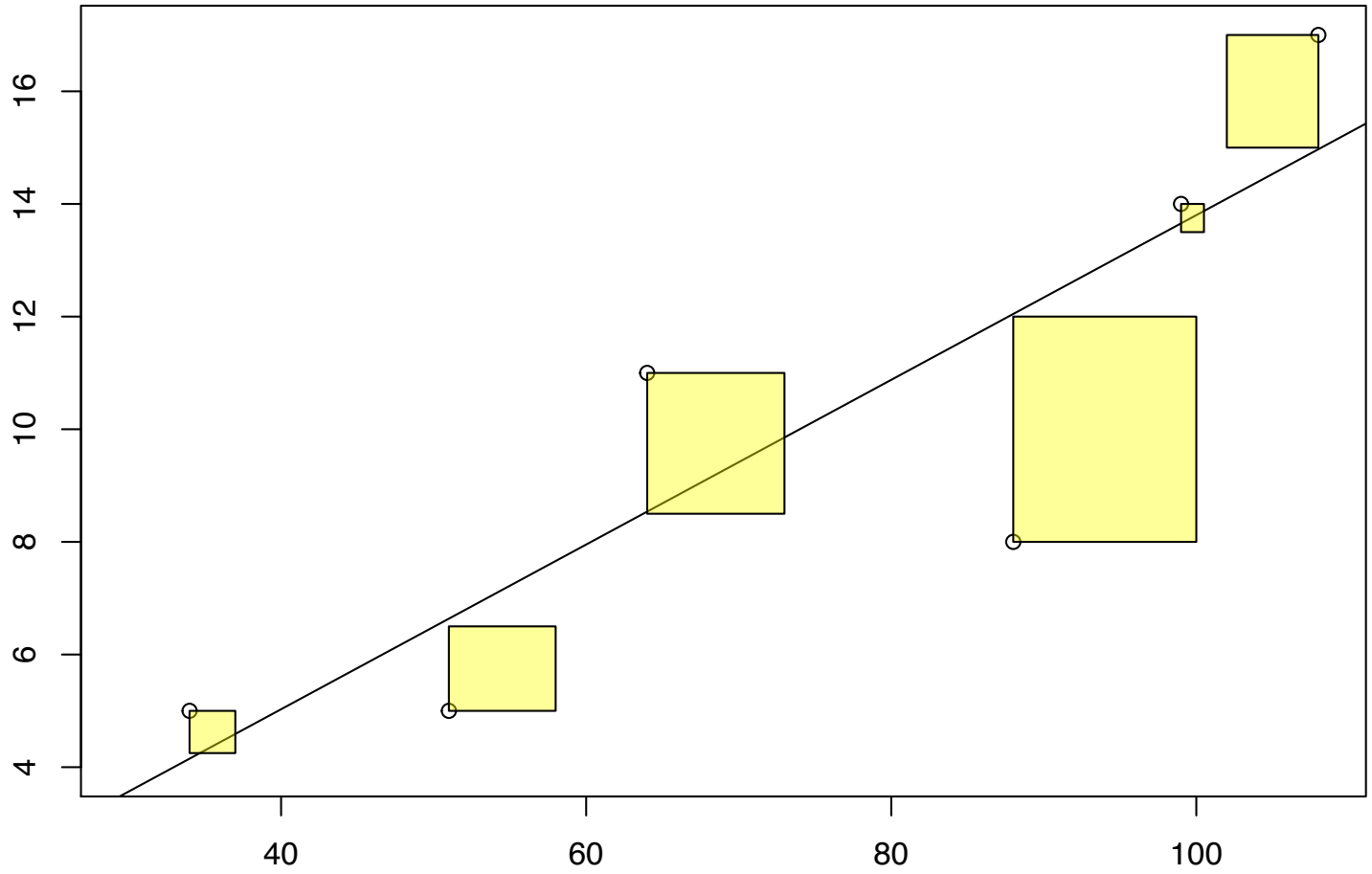
NASA Dawn spacecraft is currently orbiting dwarf planet Ceres

Least squares history



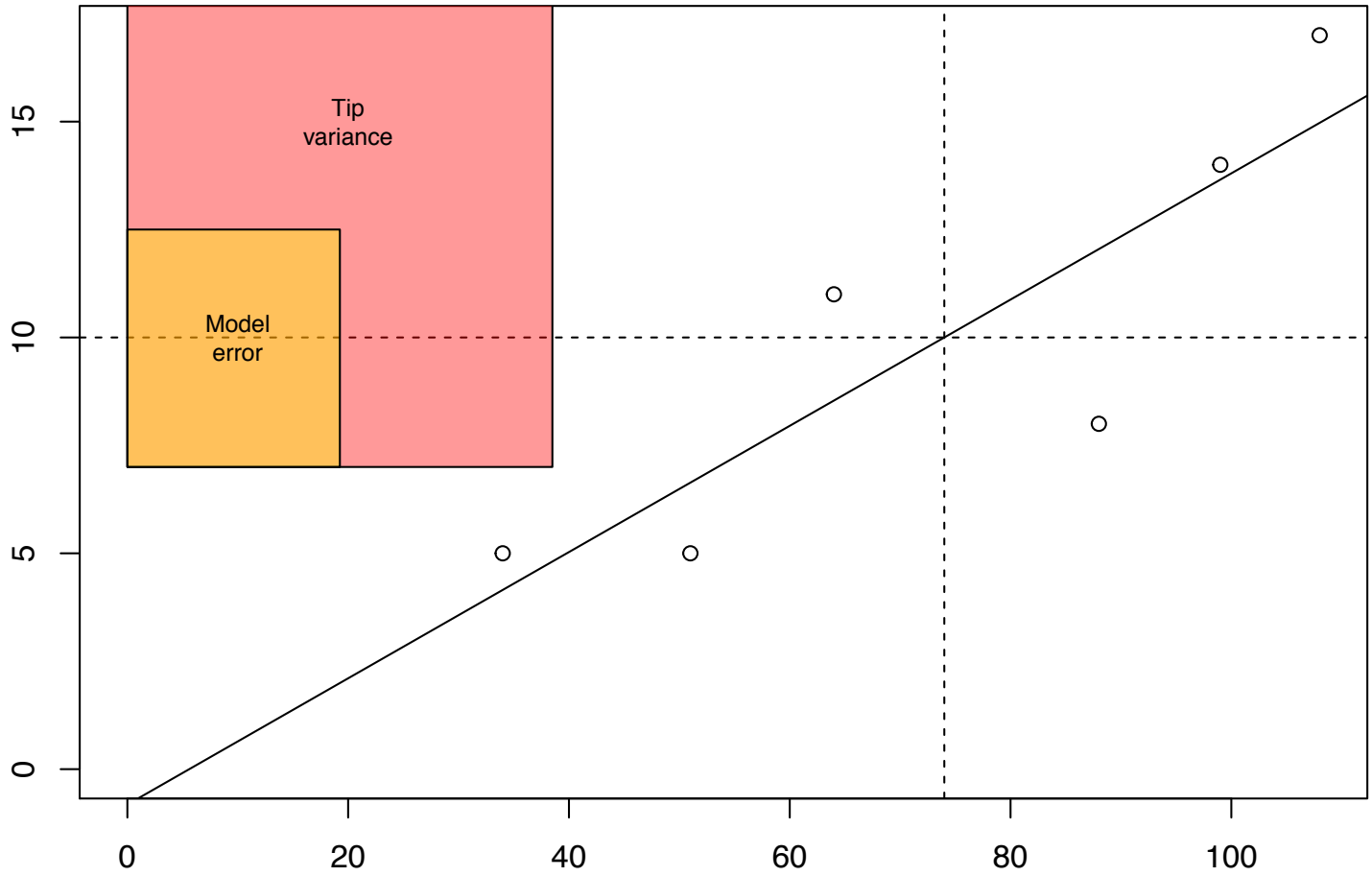
- Ceres was first observed more than 200 years ago, viz., 1801
- Mathematician C.F. Gauss most accurately estimated the (then unknown) orbit of Ceres
- He developed least squares statistical regression for this purpose
- Data errors represented by Gaussian (“normal”) distribution

Those squares are real



Calculated by “linear model” `lm()` base function in R

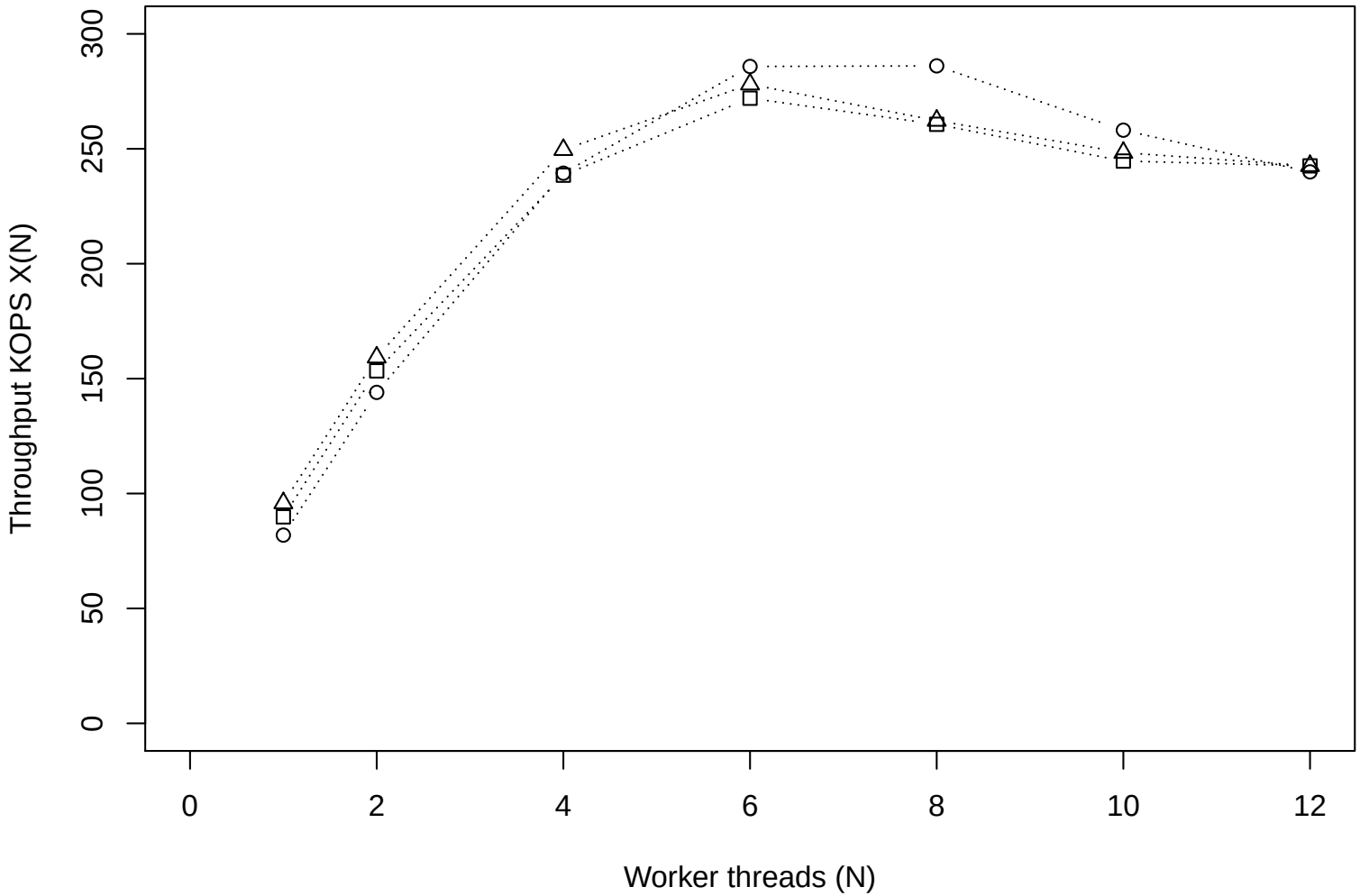
Coefficient of determination



`Multiple R-squared: 0.7494` from $R^2 = 1 - SSE/SST$

Outline

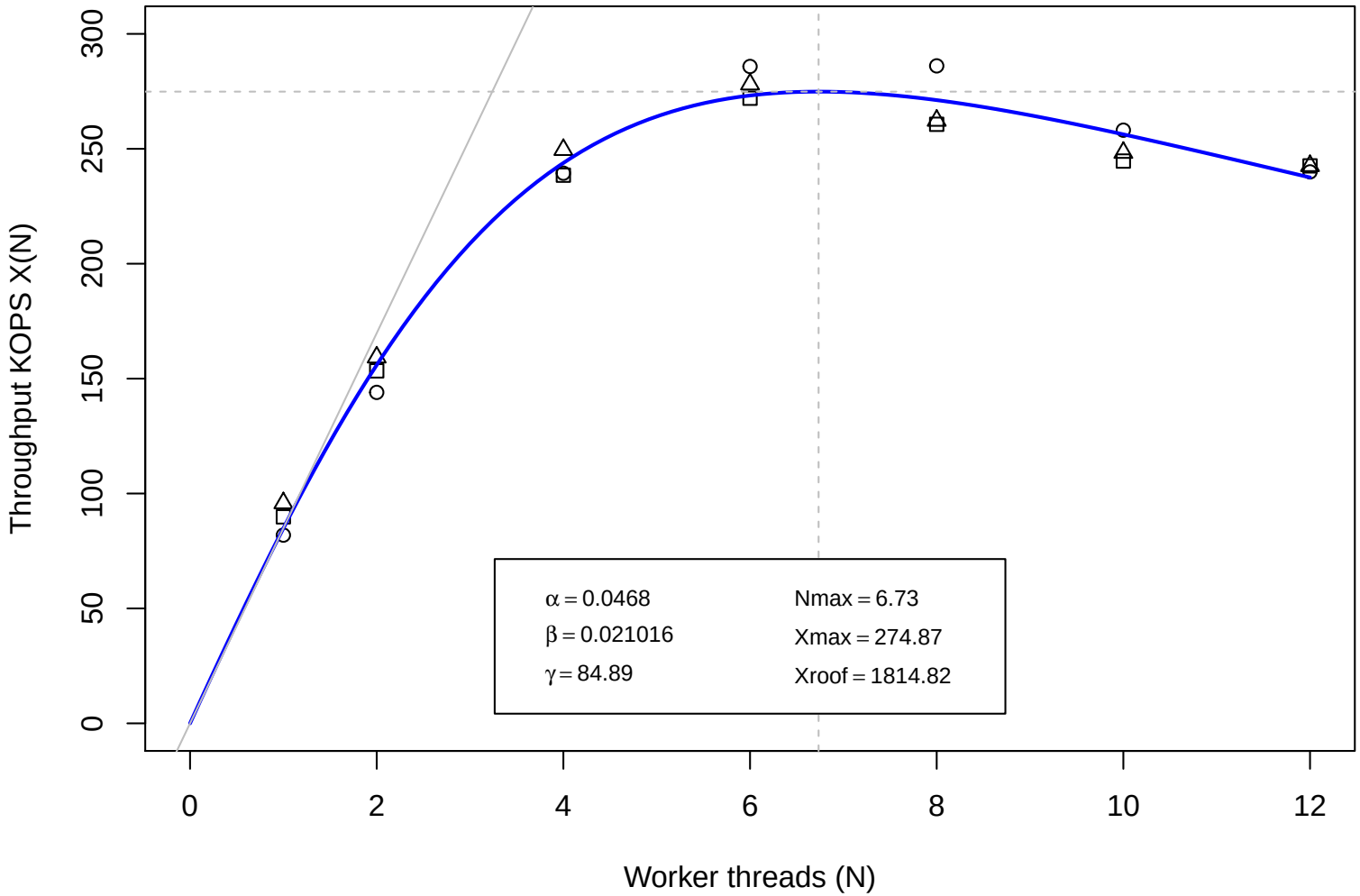
Memcache scalability data



Example (Nonlinear regression)

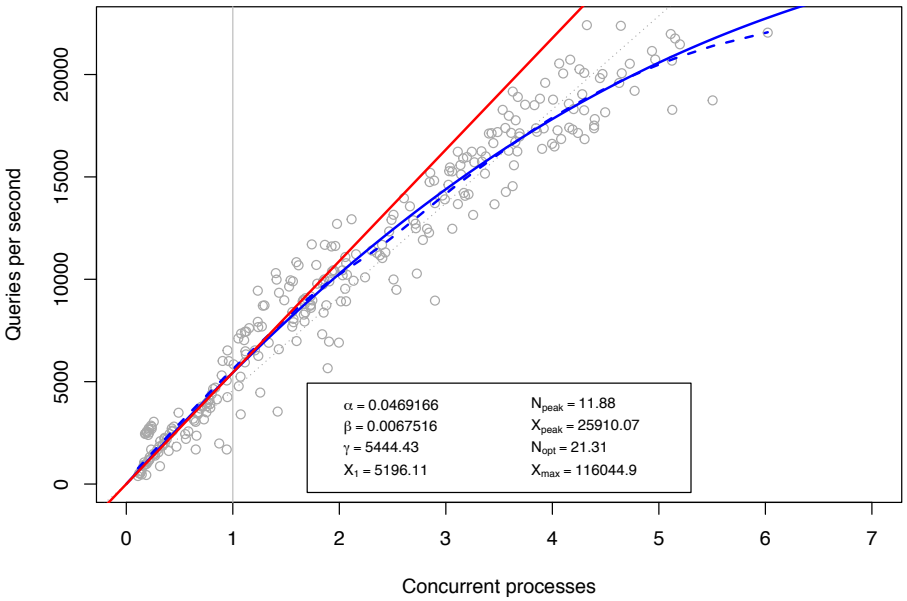
```
# plot the X(N) in KOPS against active worker threads N
plot(df.mc$N,df.mc$mc128,
     xlim=c(0,12), ylim=c(0,300),
     xlab="Worker threads (N)",ylab="Throughput KOPS X(N)"
     )
points(df.mc$N,df.mc$mc141,pch=0)
points(df.mc$N,df.mc$mc145,pch=2)

usl.fit <- nls(X.mean ~ gamma * N /
                 (1 + alpha * (N - 1) + beta * N * (N - 1)),
               data=df.mc, start=list(gamma=100, alpha=1e-2, beta=1e-2),
               algorithm="port",
               lower=c(50,0,0), upper=c(200,5e-1,5e-1)
               )

# overlay continuous USL curve in blue
curve(coef(usl.fit)['gamma'] * x /
      (1 + coef(usl.fit)['alpha'] * (x - 1) +
      coef(usl.fit)['beta'] * x * (x - 1)),
      from=0, to=12, add=TRUE, col="blue", lwd=2
      )

# compute USL scaling metrics
Nmax <- sqrt((1 - coef(usl.fit)['alpha']) / coef(usl.fit)['beta'])
Xmax <- coef(usl.fit)['gamma'] * Nmax /
  (1 + coef(usl.fit)['alpha'] * (Nmax - 1) +
     coef(usl.fit)['beta'] * Nmax * (Nmax - 1))
Nopt <- abs(1 / coef(usl.fit)['alpha'])
Xroof <- coef(usl.fit)['gamma'] * Nopt
```

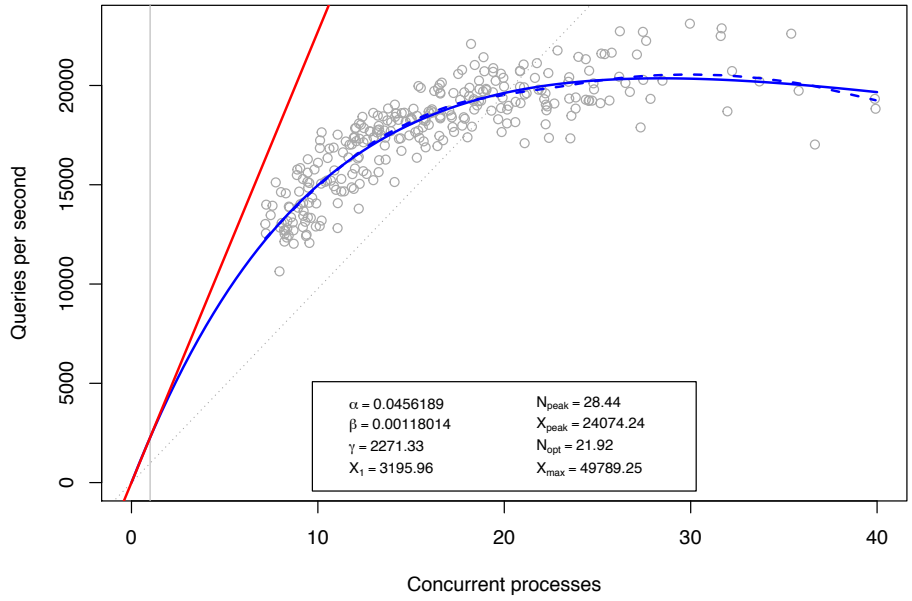
Memcache scalability model



MySQL big data

Analyzed some 2,000 production database logs
I resorted to animation as a visualization tool
About 500,000 data points in aggregate





Comparison of USL parameters
USL found unexpected progressive changes in scalability

R packages for the USL

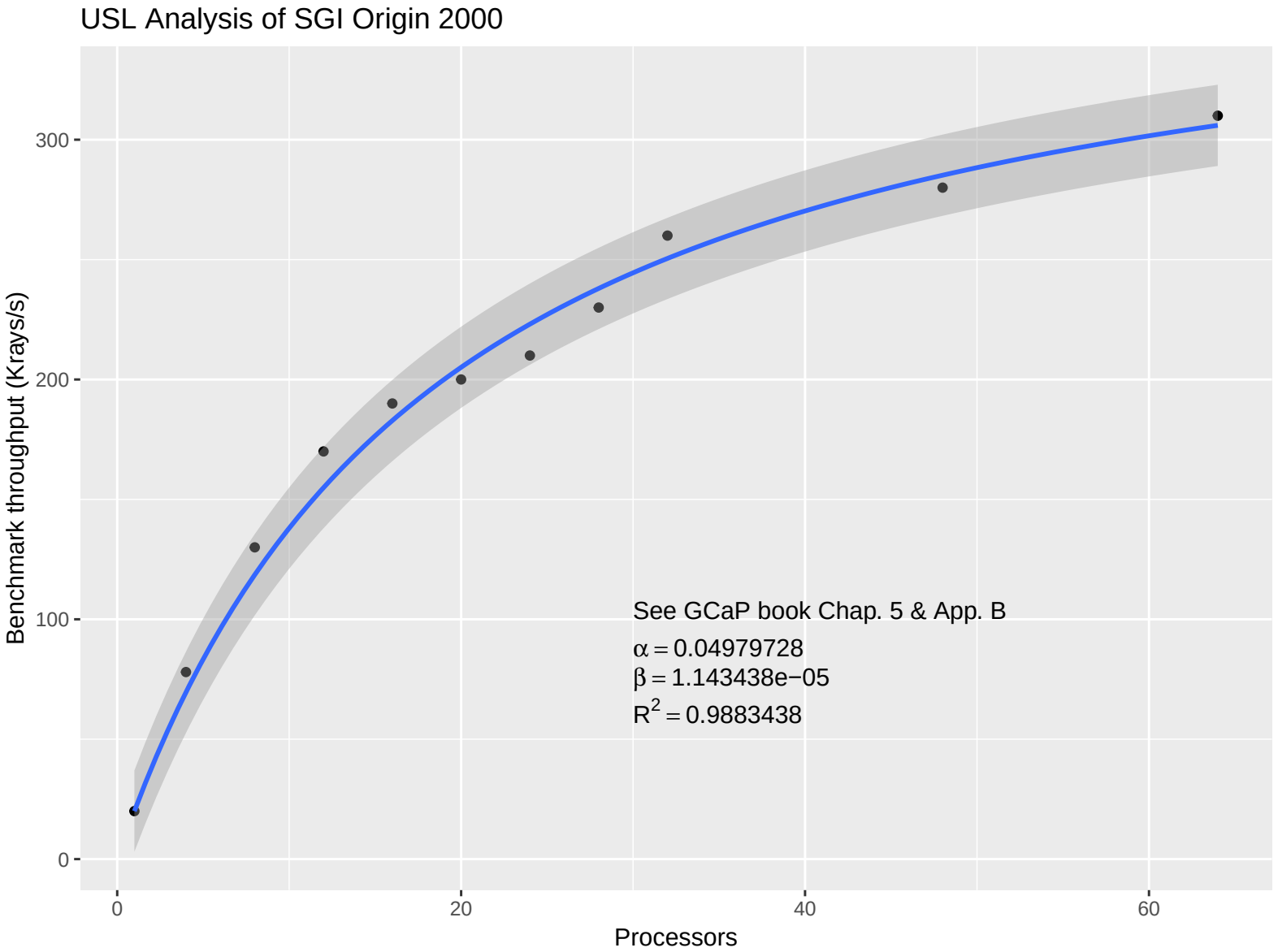
1. SATK on R-Forge
   - Author: Paul Puglia (Guerrilla graduate)
   - Applies multiple USL coefficient models for best fit

```
install.packages("SATK", repos="http://R-Forge.R-project.org")

library(SATK)
data(USLcalc)
uslcalc.zones <- zones(USLcalc)
plot(uslcalc.zones)
```

2. usl on CRAN
   - Author: Stefan Möding
   - Uses both `nls()` from base and `nlxb()` from nlsr package

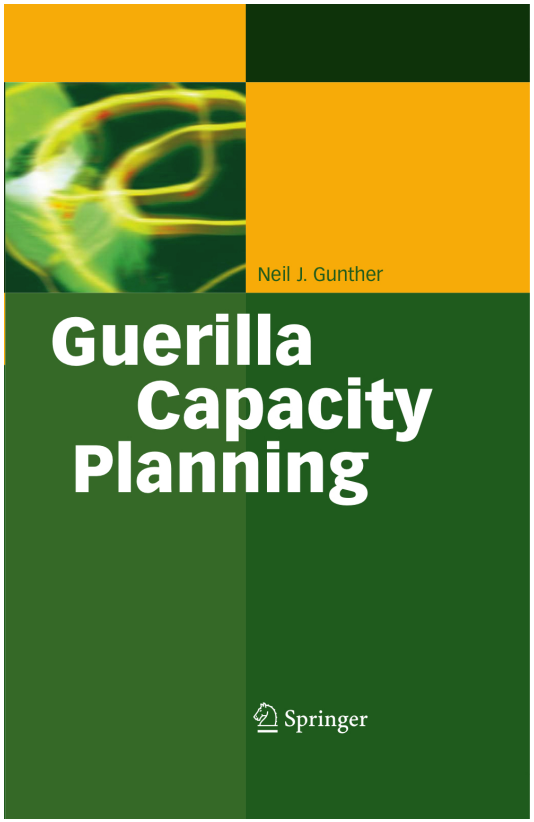
```
install.packages("usl")

library(usl)
data(specsdm91)
usl.model <- usl(throughput ~ load, specsdm91)
summary(usl.model)
peak.scalability(usl.model)
plot(specsdm91, pch=16)
plot(usl.model, col="red", add=TRUE)
```

Applying ggplot enhancements



Outline

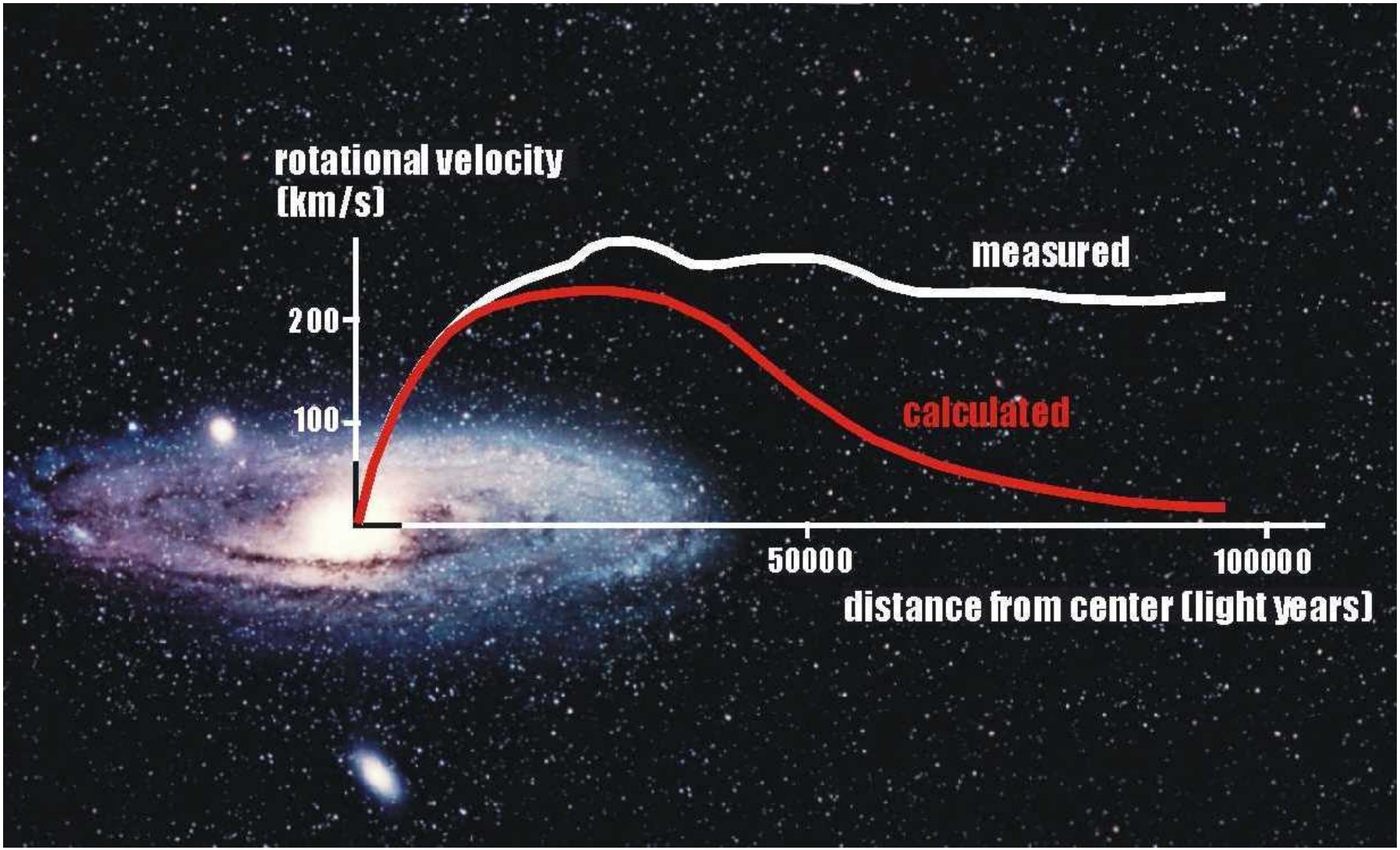
Wanna learn more?



- Chap. 4 — Scalability: A Quantitative Approach
- Chap. 5 — Evaluating Scalability Parameters
- Chap. 6 — Software Scalability

- Guerrilla training classes — textbooks included
- Guerrilla Data Analytics class — linear regression to machine learning in R

A “darker” universal law?



Questions?

www.perfdynamics.com
Castro Valley, California

Twitter
Facebook
Blog
Training classes
info@perfdynamics.com
+1-510-537-5758