Oracle Meets Fractals

and Learns the Power of Power Laws

Dr. Neil Gunther

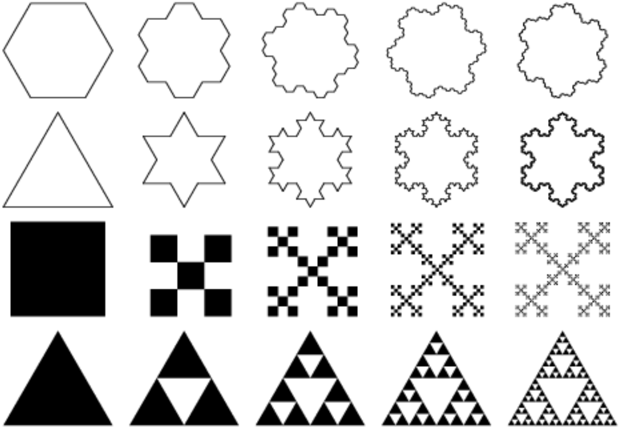
Performance Dynamics

NoCOUG Winter Conference 2012
Thursday, February 23 @ 1 pm

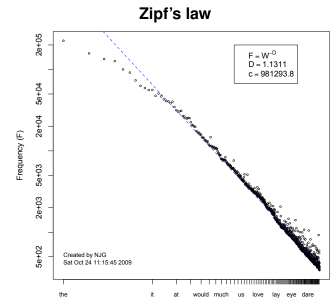
Outline

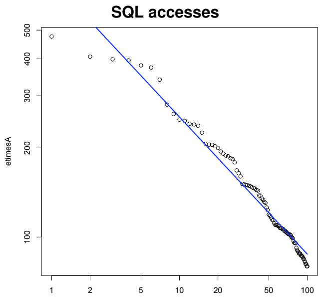
Outline

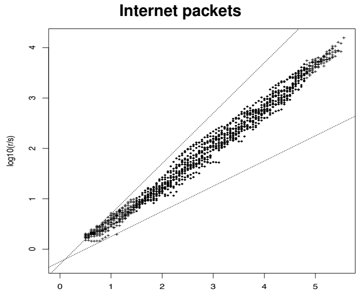
Fractals in Space

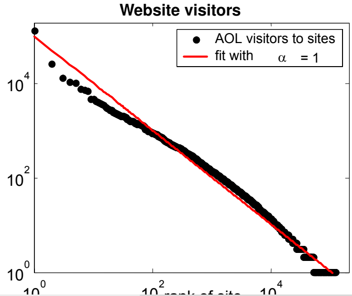


Fractals are Described by Power Laws





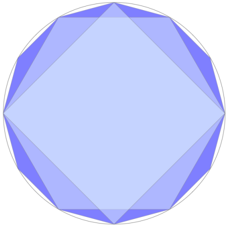




Outline

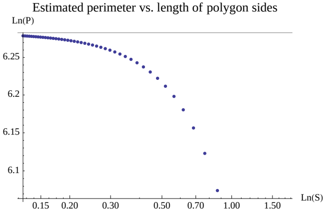
1. Fractals
   - What is a Fractal?
   - How It Works
   - Internet Traffic
2. Applications
   - Word Fractals
   - Fractal Query Times
   - Fractal Time Series
3. Conclusions

Calibrating a Circumference



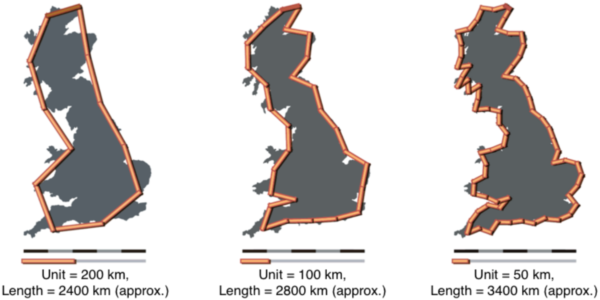
Approximating circumference by regular polygons with successively shorter sides. Polygon represents measurement device or ruler

Double-log plot of the estimated circumference (y-axis) vs. the length of the polygon side (x-axis). As the sides get shorter the perimeter of the polygon approaches the actual circumference.



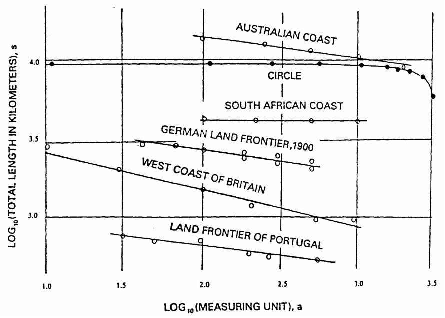
- Euclidean geometry of the circle
- Greeks knew (irrational) ratio of diameter D to circumference C: $\pi = C/D$
- Successive measurements converge to fixed value: C
- Speed of convergence is clearly seen on logarithmic axes.

Crinkly Coastlines



- What about highly irregular shapes like coastline of Britain?
- Greeks would never consider such imperfect non-Euclidean geometry.
- Successively smaller ruler size (S) produces longer coastline estimate (L)!
- Why? Smaller ruler gets into more coastal nooks and crannies.

Borders and Wars



- Plot border length (L) against ruler size (S) on log-log axes[1]
- Why do border lengths fall on **straight lines** in log-log plot?
- Any crazy country shape is then characterized by a single number: its slope!
- Reason remained obscure until Mandelbrot resurrected it as geometry of fractals[2]

[1] Lewis F. Richardson (1961) "The problem of contiguity: An appendix to Statistic of Deadly Quarrels."

[2] B. Mandelbrot, The Fractal Geometry of Nature, W. H. Freeman, New York (1983)

The Power of Power Laws

Straight lines on log-log plot have form:

$$Y = mX + c$$

But $Y \equiv \ln(L)$ and $X \equiv \ln(S)$ with negative slope:

$$\begin{aligned} \ln(L) &= -\alpha \ln(S) + \ln(k) \\ &= \ln(k\, S^{-\alpha}) \end{aligned}$$

Taking antilogs of both sides reveals general power law form:

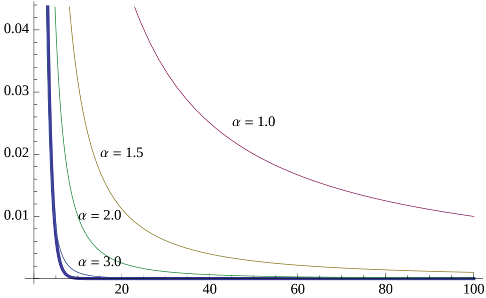
$$L = k\, S^{-\alpha}$$

$$L = \frac{k}{S^{\alpha}} \tag{1}$$

Reverse this logic

- If your data looks “linear” on a log-log plot
- Assume it signals presence of a power law like (1)
- Find the slope to characterize it
- Exponent α is the “power” in *power law*

The Shape of Power Laws

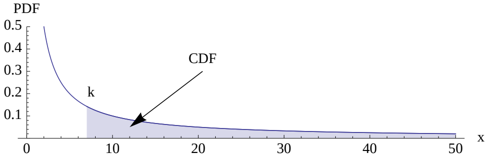


- General shape of power law eqn.(1) is a hyperbola
- Blue curve is an exponential function
- Other curves are power laws with increasing α exponents (slopes)

Big powers

Large α power laws are indistinguishable from an exponential

The Tale is in the Tail



- Power laws differ from standard statistical distributions
- Power laws carry most of the information in their tail
- Fatter tail corresponds to stronger correlations than "normal"
- Mass of tail measured by *cumulative distribution function* (CDF)

Log-log fitting

On a log-log plot we are trying to fit right-hand side data, not left side

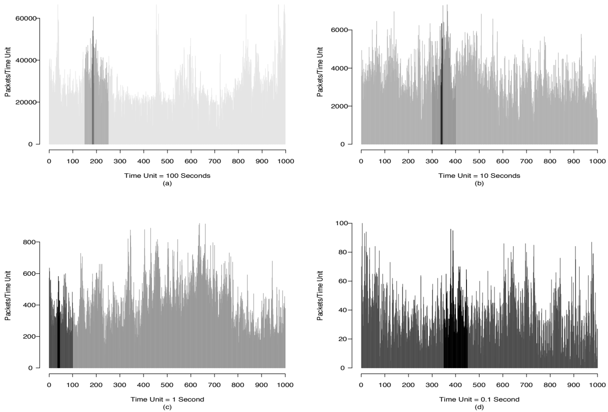
Outline

1. Fractals
   - What is a Fractal?
   - How It Works
   - Internet Traffic
2. Applications
   - Word Fractals
   - Fractal Query Times
   - Fractal Time Series
3. Conclusions

Internet Congestion

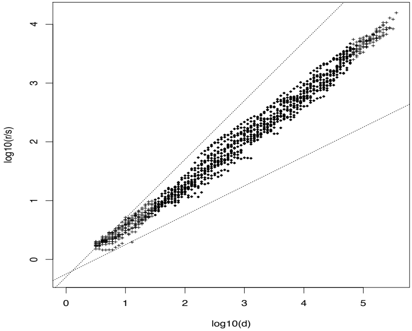
- Internet performance collapsed c.1986
- Bellcore c.1990 impact of ISDN broadband
- Packet tracing measurements at Bellcore
- Surprise: large packet trains
- Surprise: Service times file-size dependent
- Surprise: Packet arrivals not always Poisson
- Surprise: Queueing models break down
- How to do CaP for future Internet?
- Why is it happening?
- Part of the answer is power laws
- Netflix uses 33% of USA Internet BW

Strangeness in the Interpipes



Read bottom to top, left to right
Variance persists over 5 decades of time

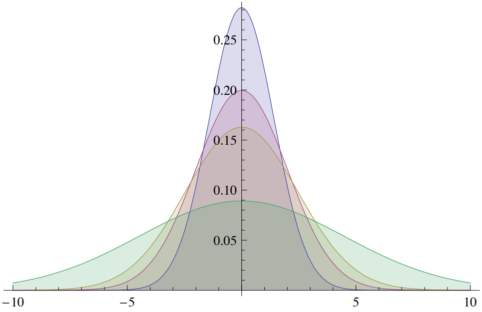
Traces on Log-Log Axes



$Y = (max - min)/$std dev ("rescaled range")

$X =$ sample size (in trace file

Fractals in Time



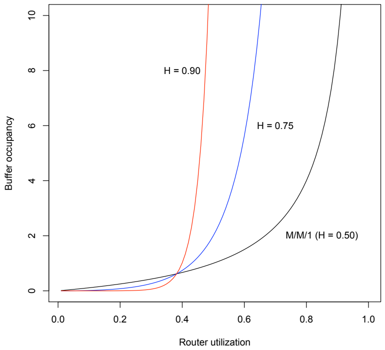
- Bellcore packet data shows fractal behavior in time (5 decades)
- Diffusion model of Brownian motion
- Solution is a normal distribution which evolves in time (t)

$$\partial_t f(x,t) = \sigma^2 \partial_x^2 f(x,t)$$

$$f(x,t) = \frac{1}{\sqrt{4\pi\sigma^2 t}} \exp\left(\frac{-(x-\mu)^2}{4\sigma^2 t}\right)$$

- $L^2 = 4\sigma^2 t$
- Diffusion length $L = \sqrt{4\sigma^2 t}$
- $L \sim t^{\frac{1}{2}}$
- Generalization $L \sim t^{\frac{1}{2}} \to t^H$ (Brownian $\to$ Levy)
- What happens if $H = \frac{1}{2}$ becomes $\frac{1}{2} < H < 1$?

Router Occupancy



- Q: queue length or buffer occupancy
- $\rho = \lambda S$: router utilization
- H: power law exponent (Hurst parameter)

$$Q = \frac{\rho^{\frac{1}{2(1-H)}}}{(1-\rho)^{\frac{1}{1-H}}}$$

- $H = 0.5$ is identical to **M/M/1 queue**
- $H = 0.9$ Internet empirical Hurst exponent
- Buffer overflow can occur at lower loads

Router model

```
x<-c(1:100)
rho<-x/100
qlen<-function(r,H){r^(1/(2*(1-H))) / ((1-r)^(H/(1-H)))}
plot(rho,qlen(rho,0.5),type="l",xlab="Router utilization",ylab="Buffer occupancy",ylim=c(0,10))
lines(rho,qlen(rho,0.75),col="blue")
lines(rho,qlen(rho,0.90),col="red")
```

Outline

1. Fractals
   - What is a Fractal?
   - How It Works
   - Internet Traffic
2. Applications
   - Word Fractals
   - Fractal Query Times
   - Fractal Time Series
3. Conclusions

Outline

Data Source — Corpus (Body of Words)

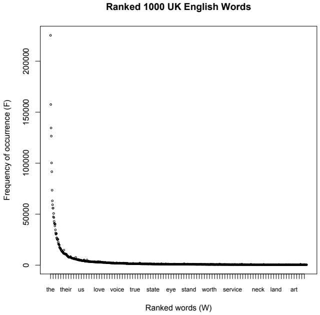
Data (already ranked) is 1000 most common wordforms in UK English based on 29 works of literature by 18 authors (4.6×10^6 words)

- **Wordform:** english word
- **Abs:** absolute frequency (total number of occurrences)
- **r:** range (number of texts in which the word occurs)
- **mod:** modified frequency as defined by Rosengren (1972)

Read data file

```
> dir<-setwd("~/../GDAT Scripts/Power Laws/")
> td<-read.table("zipf1000.txt",header=TRUE)
> head(td)
  Rank Wordform    Abs  r      mod
1    1      the 225300 29 223066.9
2    2      and 157486 29 156214.4
3    3       to 134478 29 134044.8
4    4       of 126523 29 125510.2
5    5        a 100200 29  99871.2
6    6        I  91584 29  86645.5
```

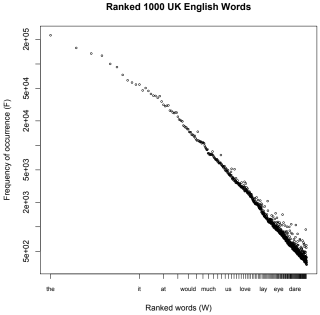
Linear Visualization



Linear plot of data

```
> plot(td$Rank, td$Abs, type="p", main="Ranked 1000 UK English Words",
xlab="Ranked words (W)", ylab="Frequency of occurrence (F)",
xaxt="n",log="xy",cex=0.5)
> ticks.at<-seq(min(td$Rank), max(td$Rank),10)
> ticks.lab<-as.character(td$Wordform[ticks.at])
> Axis(td$Rank, at=ticks.at, las=1, side=1,labels=ticks.lab, cex.axis=0.75)
```

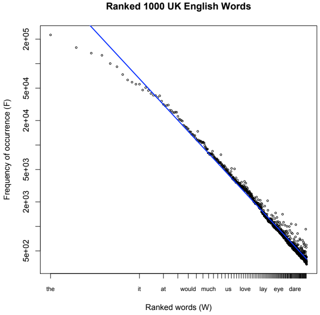
Double-Log Visualization



Data on log-log axes

```
> plot(td$Rank, td$Abs, type="p", main="Ranked 1000 UK English Words",
xlab="Ranked words (W)", ylab="Frequency of occurrence (F)",
xaxt="n",log="xy",cex=0.5)
> ticks.at<-seq(min(td$Rank), max(td$Rank),10)
> ticks.lab<-as.character(td$Wordform[ticks.at])
> Axis(td$Rank, at=ticks.at, las=1, side=1,labels=ticks.lab, cex.axis=0.75)
```

Regression Fit



Regression fit to logarithmic data

```
# regression model of Y=log(y) and X=log(x)
> z.fit <- lm(log(td$Abs) ~ log(td$Rank))
# Must transform back to log scaled coords in plot
> ly <- exp( (coef(z.fit)[2])*log(td$Rank) + coef(z.fit)[1] )
> lines(td$Rank, ly, col="blue",lty="solid",lwd=2)
```

Summary of Regression Statistics

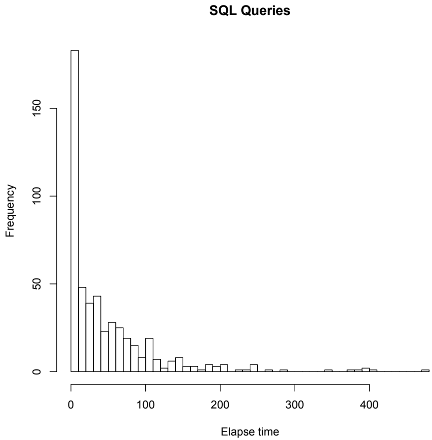
Regression summary

```
> summary(z.fit)

Call:
lm(formula = log(td$Abs) ~ log(td$Rank))

Residuals:
     Min       1Q   Median       3Q      Max
-1.47144 -0.06902 -0.01477  0.05757  0.81894

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.796627   0.024210   569.9   <2e-16 ***
log(td$Rank) -1.131084   0.004039  -280.0   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1258 on 998 degrees of freedom
Multiple R-squared: 0.9874, Adjusted R-squared: 0.9874
F-statistic: 7.841e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Outline

1. Fractals
   - What is a Fractal?
   - How It Works
   - Internet Traffic
2. Applications
   - Word Fractals
   - Fractal Query Times
   - Fractal Time Series
3. Conclusions

Interpreting Time Histogram



- Histogram of measured SQL query times
- *x*-axis is elapsed time in **seconds**
- *y*-axis is number of queries with that time
- What distribution profile is it?
- Exponential, log-normal,...
- Can't tell by just staring at it

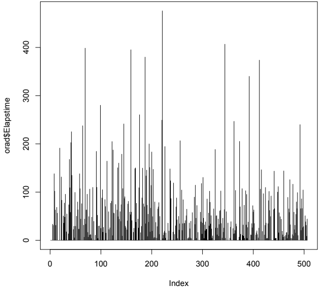
Data Source

- Original question: Craig Shallahamer's blog
- Attempted solution: Dave Abercrombie's blog
- My solution: My blog

Read data file

```
> dir<-setwd("~/Desktop/GDAT Dev 2011/GDAT Scripts/Power Laws/")
> orad<-read.table("orasql-data.txt",header=FALSE)
# Add column names
> colnames(orad) <- c(
  "SQLid", "sample", "execns", "dkReads", "buffGets", "CPUtime", "Elapstime"
)
> head(orad)
                    SQLid sample execns dkReads buffGets CPUtime Elapstime
1 8qtkxy0g5d1p3,2282376281      1      1       0        3   0.100     0.100
2 8qtkxy0g5d1p3,2282376281      2      1       0        3   0.106     0.106
3 8qtkxy0g5d1p3,2282376281      3      1       0        3   0.101     0.101
4 8qtkxy0g5d1p3,2282376281      4      1       0        3   0.098     0.098
5 8qtkxy0g5d1p3,2282376281      5      1       6      118   0.000    33.575
6 8qtkxy0g5d1p3,2282376281      6      1       8      137  10.000    31.004
```

Visualize Raw Data

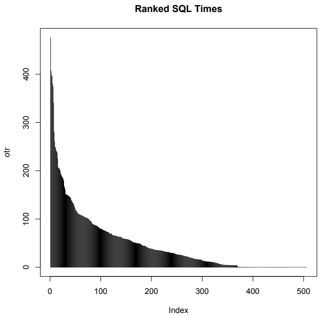


Linear plot of unranked data

```
> plot(orad$Elapstime, type="h")
```

Like Zipf's law, data must be ranked by frequency of occurrence

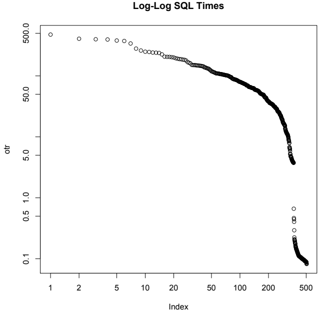
Visualize Ranked Data



Linear plot of ranked data

```
> otr <- sort(orad$Elapstime, decreasing=TRUE)
> plot(otr,type="h",main="Ranked SQL Times")
```

Double-Log Visualization



Log-log plot of ranked data

```
> plot(otr,log="xy",main="Log-Log SQL Times")
```

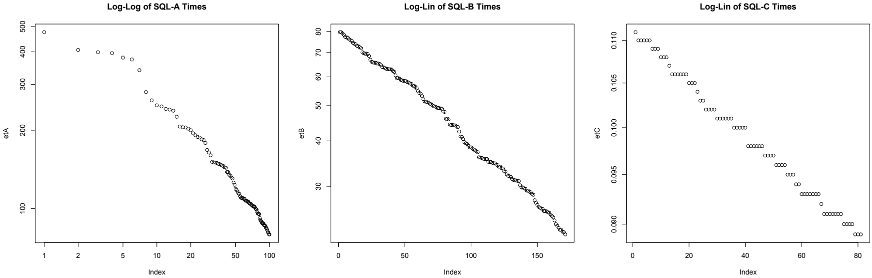
- Clearly this profile is not power law overall
- But the first 100 queries do appear to be power law

Data Regions

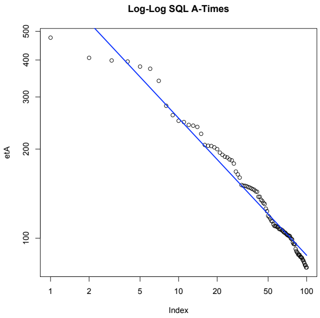
This suggests breaking data across 3 regions as follows:

Windowed plots

```
# Define data windows of ranked data
etA<-otr[1:100]
etB<-otr[100:270]
# gap...
etC<-otr[420:500]
plot(etA,type="p",log="xy",main="Log-Log of SQL-A Times")
plot(etB,type="p",log="y", main="Log-Lin of SQL-B Times")
plot(etC,type="p",log="y", main="Log-Lin of SQL-C Times")
```



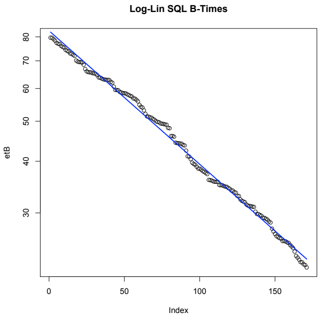
Data Region A Fit



Regression analysis for Window A

```
> xA<-seq(1:length(etA))
> zA.fit<-lm(log(etA) ~ log(xA))
> EyA<-exp(coef(zA.fit)[2]*log(xA) + coef(zA.fit)[1])
> plot(etA,log="xy",main="Log-Log SQL A-Times")
> lines(xA,EyA,col="blue",lwd=2)
```

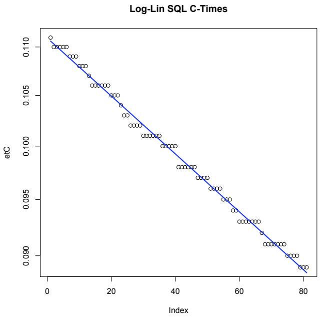
Data Region B Fit



Regression analysis for Window B

```
> xB<-seq(1:length(etB))
> zB.fit<-lm(log(etB) ~ xB)
> EyB<-exp(coef(zB.fit)[2]*xB + coef(zB.fit)[1])
> plot(etB,log="y",main="Log-Lin SQL B-Times")
> lines(xB,EyB,col="blue",lwd=2)
```

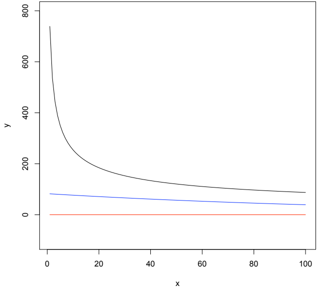
Data Region C Fit



Regression analysis for Window C

```
> xC<-seq(1:length(etC))
> zC.fit<-lm(log(etC) ~ xC)
> EyC<-exp(coef(zC.fit)[2]*xC + coef(zC.fit)[1])
> plot(etC,log="y",main="Log-Lin SQL C-Times")
> lines(xC,EyC,col="blue",lwd=2)
```

Regression Models



$$y_A \sim x^{-0.4632} \quad \text{power law}$$

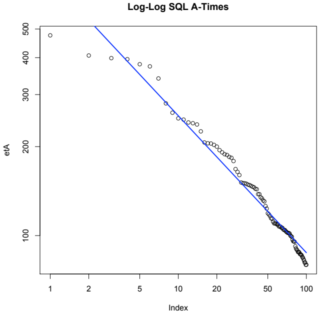
$$y_B \sim e^{-0.0074x} \quad \text{exponential decay}$$

$$y_C \sim e^{-0.0028x} \quad \text{exponential decay}$$

Regression coefficients

```
> coef(zA.fit)
(Intercept)      log(xA)
  6.6055308  -0.4632485
> coef(zB.fit)
 (Intercept)           xB
 4.416070310 -0.007438368
> coef(zC.fit)
 (Intercept)           xC
-2.198802043 -0.002782828
```

Slope Analysis



- From `coef(zA.fit)` know `log(xA) = -0.4632485`
- Empirical slope $\gamma = 0.46$ to two significant decimal digits
- About half Zipfian slope $\gamma = 1.0 \pm 0.5$
- Correlations are stronger than for Zipf

Hypothesis

Shorter query times (window A) may be associated with dictionary lookups or other structured data. That structure provides correlations. Longer queries in windows B and C are not structured (ad hoc?) and are therefore more randomized. The lack of strong correlations shows up as different exponential decay rates.

Outline

1. Fractals
   - What is a Fractal?
   - How It Works
   - Internet Traffic
2. Applications
   - Word Fractals
   - Fractal Query Times
   - Fractal Time Series
3. Conclusions

The Perfect Storm

A Power Law Storm

Before the Storm

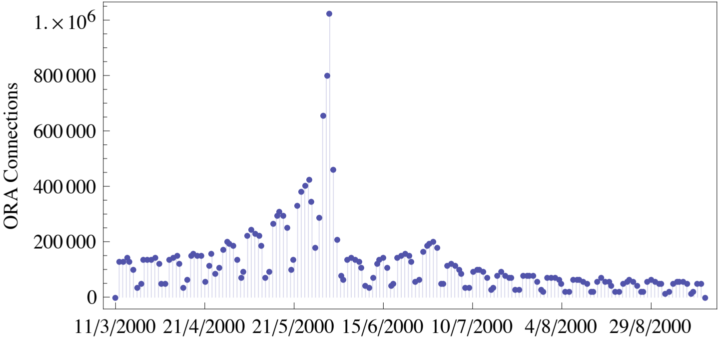
An Australian Government Initiative business.gov.au ABN Lookup

All businesses are required to register with the Australian Tax Office (ATO) for an Australian Business Number (ABN) to claim an income tax refund. The ABN was introduced in Y2K.

- Data from website hosting initial ABN registrations.
- Period covers March 27 to September 19, 2000
- Post-advertising traffic 1 March to 30 May , 2000
- Deadline spike on 31 May, 2000
- Smaller traffic peaks from 1 June to 30 June, 2000
- Post deadline period from 1 July to 19 Sept, 2000

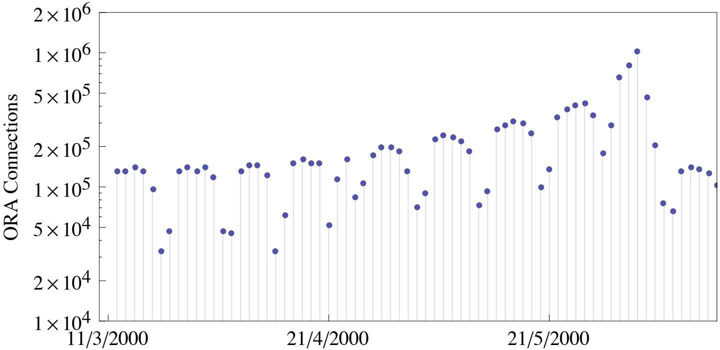
Full details can be found in my CMG-A paper `cmga-p10167.pdf` included in your GDAT class materials.

Full Data Profile



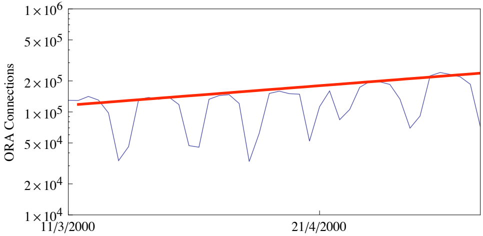
- Question: Could the “11th hour” spike have been predicted?
- Answer: Yes, but quite involved.
- How: Using a power law. What else!?

Log-Linear Plot



- y-axis is the number of Oracle RDBMS connections
- Here, the y-axis is log scaled
- Peak growth preceding spike looks linear on semi-log plot
- x-axis index (not shown) is "days from the start of data window"
- time series index range $t = 0$ to $t = 38$ days

Semi-Log Regression on Peaks

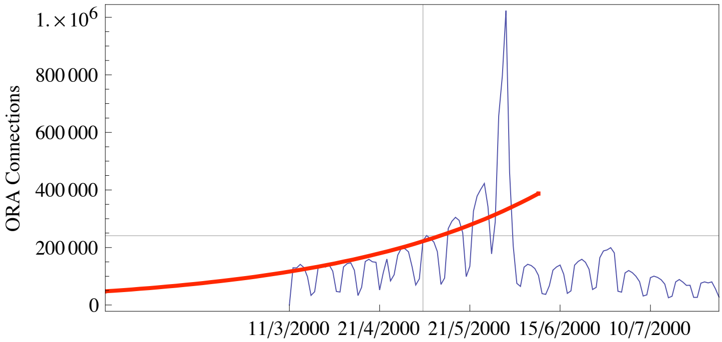


- Linear peak-growth on semi-log axes
- Curve must be an exponential function
- Use Exp as regression model
- $\hat{y}(t) = A \exp(Bt)$

Model parameters

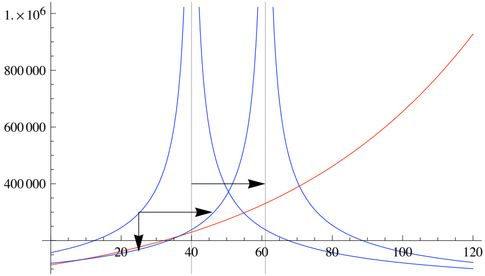
- Origin: A = 114128
- Curvature: B= 0.0175
- Doubling period: $T_2 = \dfrac{\ln(2)}{B} \sim 6$ months

Trend Overview



- Revert to linear axes to review the trend
- Exponential forecast up to the crosshairs looks valid
- But significantly underestimates onset of the “11th hour” peak
- As well as rapid drop off on RHS of the peak

Power Law Fit

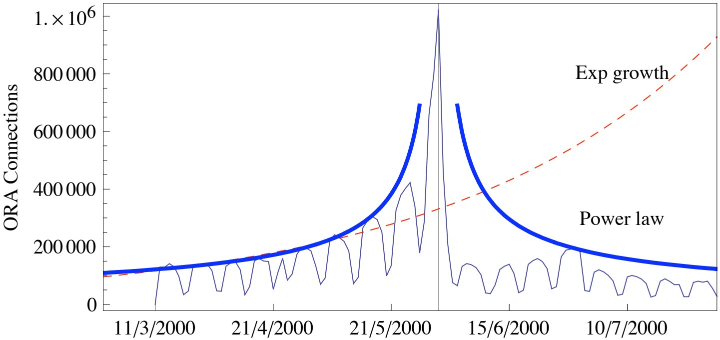


- Power law has a *critical point* t_c
- Equation: $\hat{y}(t) = k|t - t_c|^{-\gamma}$
- See far LHS curve with $t_c = 40$ (blue)
- Estimate $\hat{y}(t) \to \infty$ at $t = t_c$
- Translate $\hat{y}(t)$ rightward until lower part of curve matches Exp function (red)
- Critical point also moves to $t_c = 61$ (31 May, 2000)

Critical point

- New element is the appearance of a critical point at t_c
- Power law goes infinite and spikes at $t_c = 61$ with $\gamma = 0.6421$

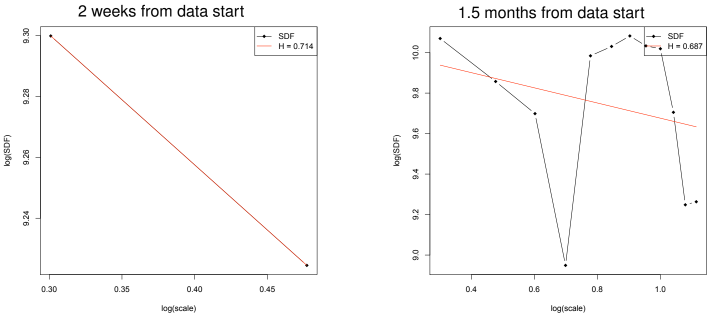
Comparison of Models



- Exponential trend is consistent with data through April 2000
- Completely underestimates onset of the "11th hour" spike
- Completely overestimates decay of traffic load beyond spike
- Data is already exceeding Exp model during April-May period
- Power law model predicts all these effects quite well
- Critical point is inclusion of critical point t_c

Look-ahead Tools

Could we have seen the spike coming without knowing t_c ?



Estimate $H = \frac{1}{2}(1 - \beta)$ from the slope β of $\ln[S(f)]$ vs $\ln[f]$ in frequency domain:

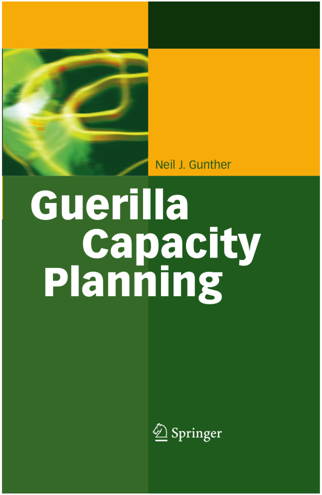
- $H \in [\frac{1}{2}, 1)$ persistent autocorrelations (increase/decrease typically followed by increase/decrease)
- $H = \frac{1}{2}$ statistically independent random fluctuations
- $H \in (0, \frac{1}{2}]$ antipersistent autocorrelations (increase/decrease typically followed by decrease/increase)

Outline

Review

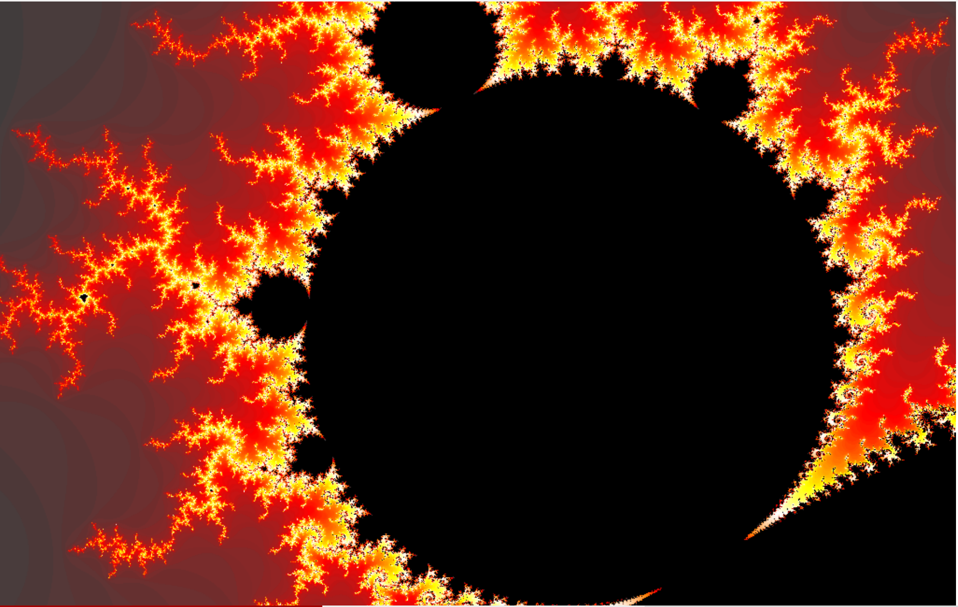
- Power laws are ubiquitous (but usually hidden)
- Need to transform your data (correctly) to see them
- Power laws are not like standard statistical distributions
- Power laws have fatter tails that carry the bulk of information
- Power laws are often easy to demonstrate with log-log plot
- Looked at 3 examples:
    - Zipf's law for word frequencies
    - ORA SQL query elapsed times
    - ORA ABN time-series spike
- Need to explain persistent correlations
- Might need more data but that's exactly how it should be

Wanna Learn More?

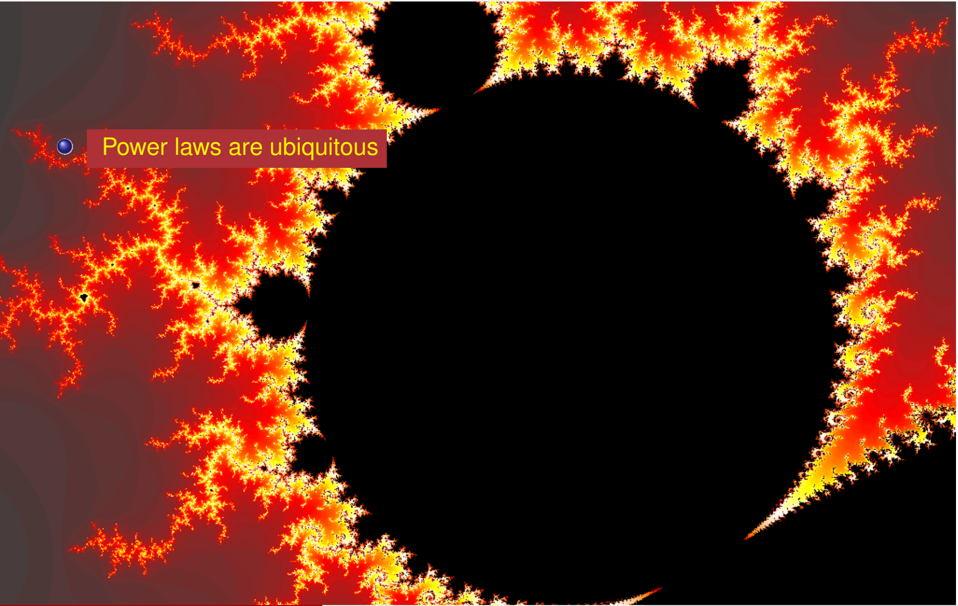


- Chapter 10 Internet Planning
- Bellcore traces
- Fractals and Self-Similarity
- Short-range Dependence
- Long-range Dependence (LRD)
- Ethernet Packetization
- LRD and Flicker Noise
- Guerrilla training classes

Why You Should Care

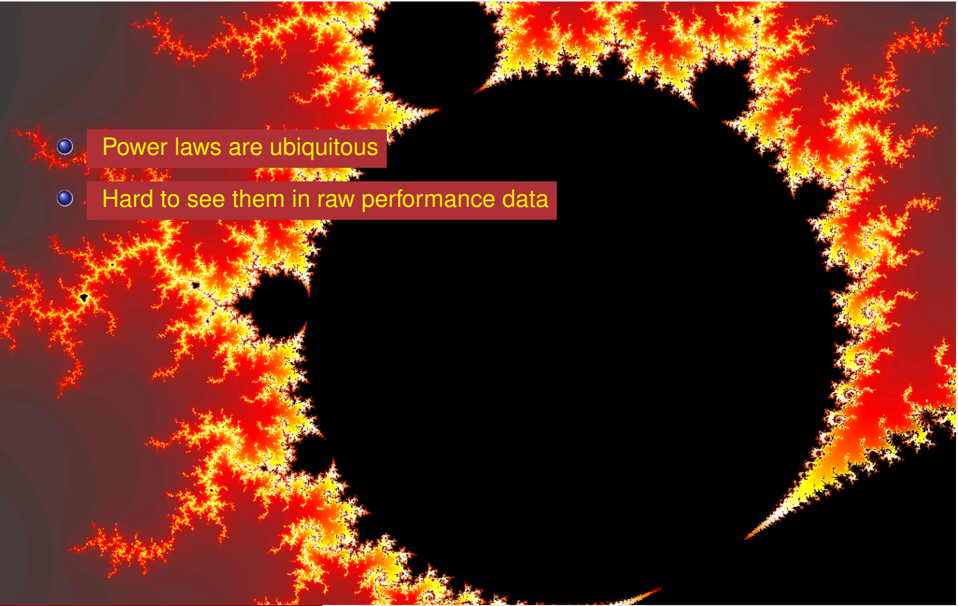


Why You Should Care



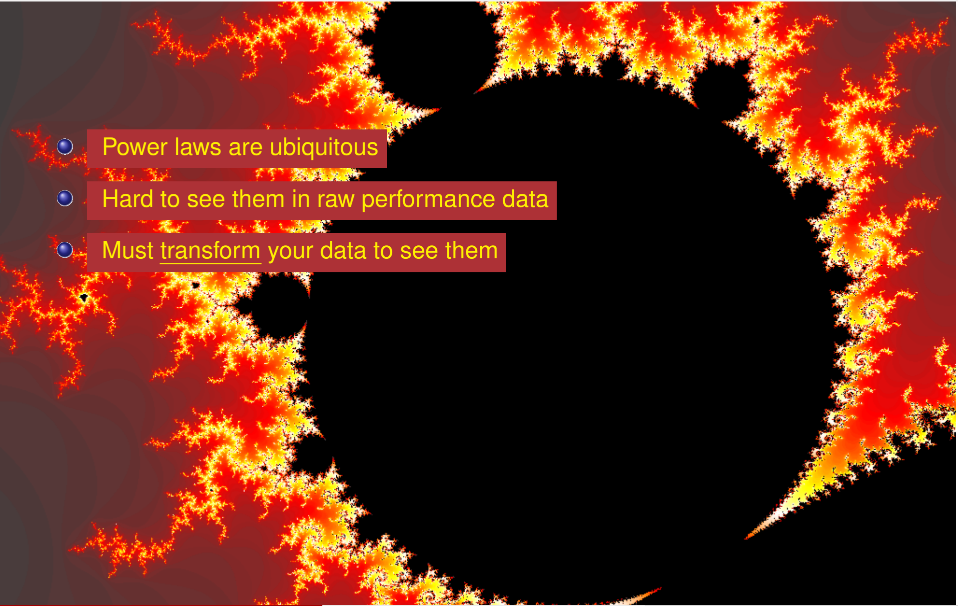
- Power laws are ubiquitous

Why You Should Care



- Power laws are ubiquitous
- Hard to see them in raw performance data

Why You Should Care



- Power laws are ubiquitous
- Hard to see them in raw performance data
- Must transform your data to see them

Why You Should Care

- Power laws are ubiquitous
- Hard to see them in raw performance data
- Must transform your data to see them
- Ranked data appears **linear** on double-log axes

Why You Should Care

- Power laws are ubiquitous
- Hard to see them in raw performance data
- Must transform your data to see them
- Ranked data appears **linear** on double-log axes
- More persistent response degradation than usual

Why You Should Care

- Power laws are ubiquitous
- Hard to see them in raw performance data
- Must transform your data to see them
- Ranked data appears **linear** on double-log axes
- More persistent response degradation than usual
- Can seriously impact overall database performance

References

- B. Mandelbrot,
  The Fractal Geometry of Nature, W. H. Freeman, 1983
- L. Liebovitch,
  Fractals and Chaos Simplified for the Life Sciences, Oxford Uni. Press, 1998
- K. Park and W. Willinger,
  Self-Similar Network Traffic and Performance Evaluation, John Wiley, 2000
- N. J. Gunther,
  Guerrilla Capacity Planning, Springer, 2007
  www.perfdynamics.com/iBook/gcap.html
- The R Project
  Tools for Statistical Computing
  www.r-project.org
- Performance Dynamics Educational Services
  Local and on-site training
  www.perfdynamics.com/Classes/schedule.html

Thank you for attending!

Performance Dynamics Company
Castro Valley, California
www.perfdynamics.com
perfdynamics.blogspot.com
twitter.com/DrQz
facebook.com/Performance-Dynamics-Company
info@perfdynamics.com
+1-510-537-5758