typical statistics in computer systems – guilty until proven innocent

(and they mostly aren't)

what is the average of 1, 3, 2, 1000 ? and does it mean anything?

charles loboz

I speak for myself only – not for my employer. No confidential data is used.

What this is about – and why

- Reasoning about performance and capacity of larger systems
- You designed an application, optimized the code, implemented the finest data compression, thoughtfully minimized network usage.
- Now you face thousands of users hitting hundreds of servers 24x7 – you have to know how soon you will run out of serves, storage, face networking bottlenecks.

Resource usage dynamics

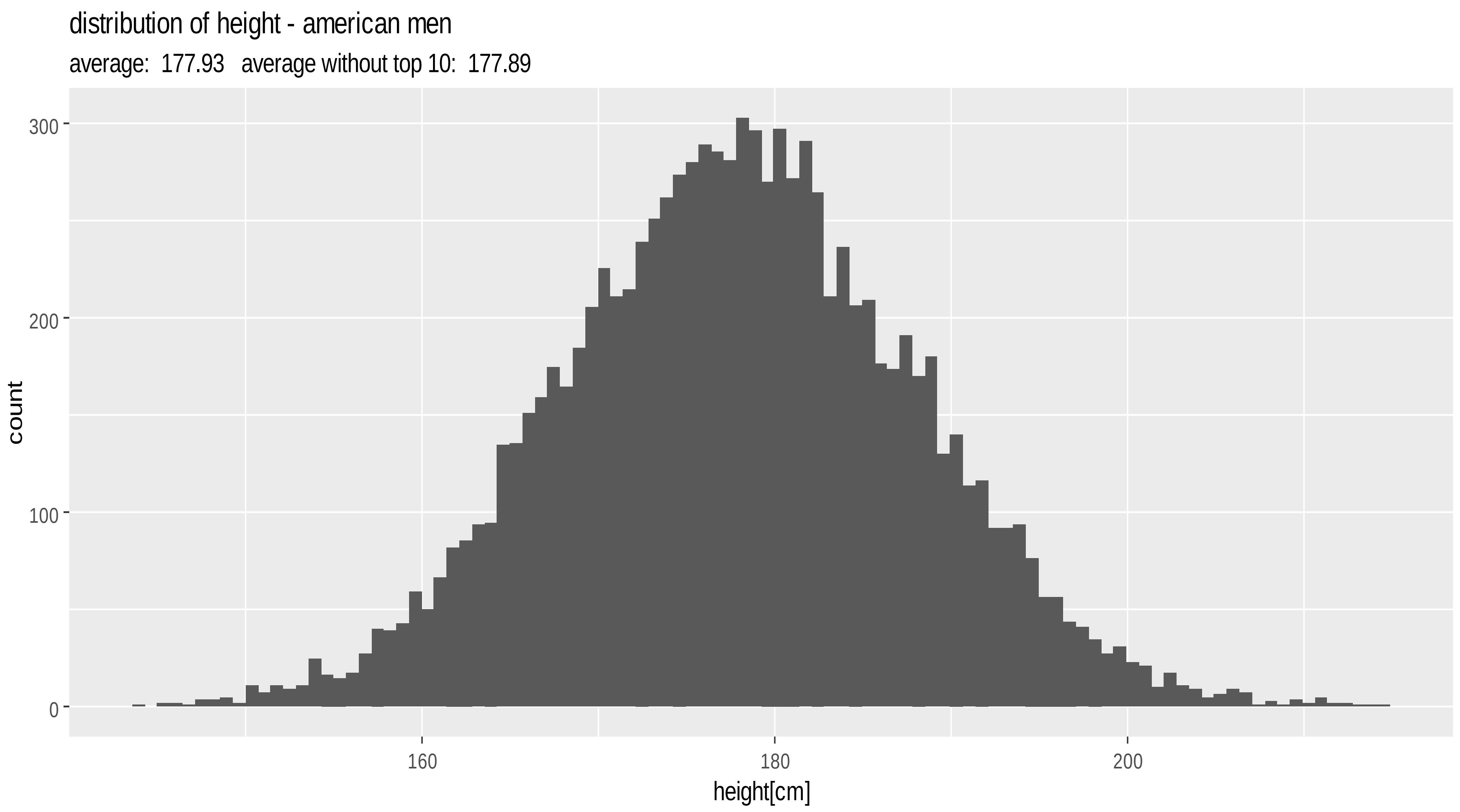
- how do you reason about performance of real-world production systems?
- Bases are the same as for smaller systems:
  - you measure/monitor resource usage
  - you do some statistical modeling to estimate needed capacity
- except that now you are playing with thousands of servers and tens of thousands of customers
- big numbers, big bucks, potential for big wastage

- unused hardware costs money, hardware shortage loses customers (trust and money)
- cloud computing will *not* solve all these problems automatically – at least not on that scale:
  - How many servers, storage you need on long-term commitment?
  - Is overflow small enough for dynamic resizing?
  - What are your customers going to do next week? month? quarter?

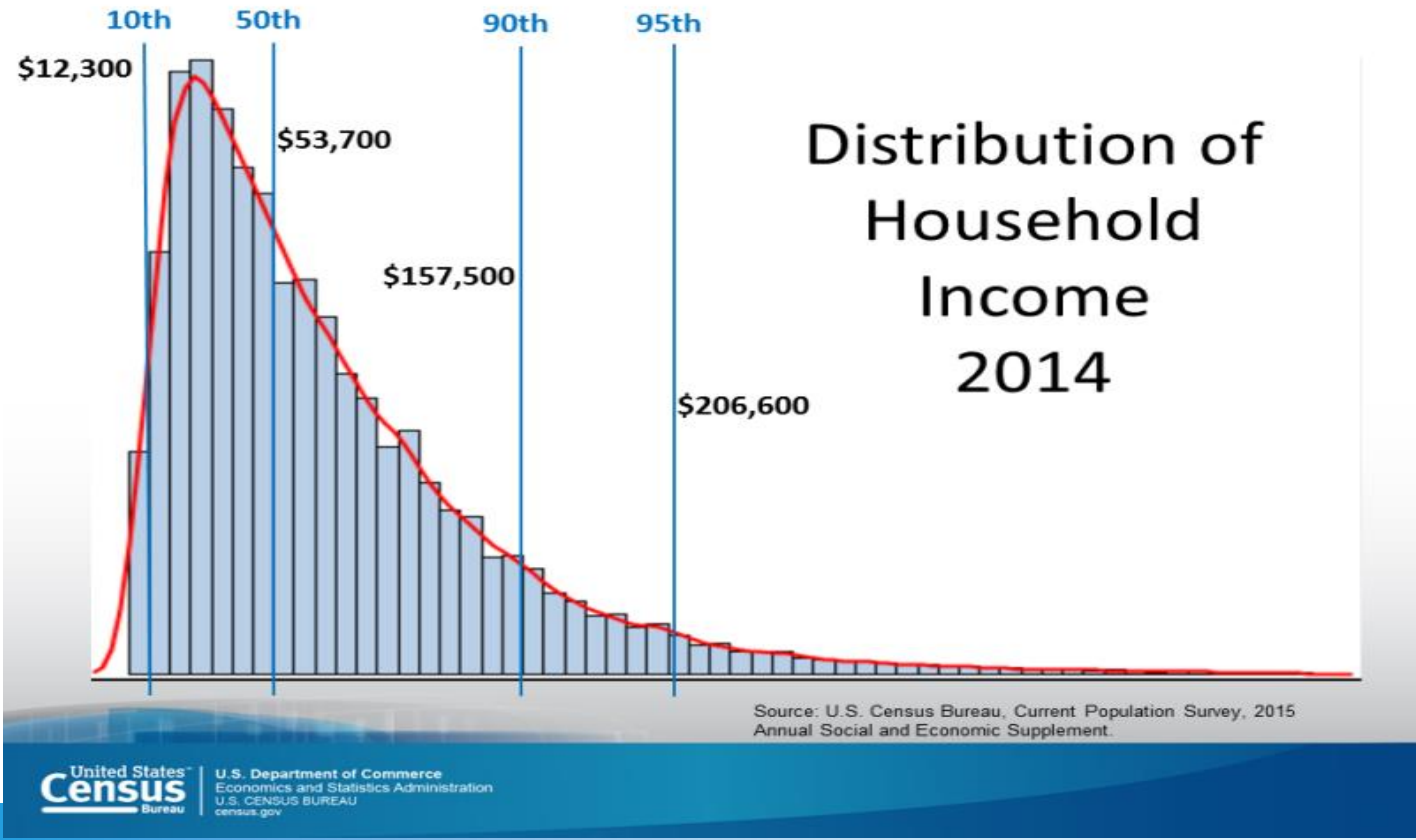
Our analytical tools

- Traditional statistics 101, 201 (cf. Raj Jain)
- Queuing theory (Kleinrock, Menasce)
- What is missing and leading to misleading reasoning:
  - Not-so-hidden assumptions – distributions are normal-like, or exponential or Poisson
    - Any other distributions lead to multi-page queuing formulas
    - Modelled systems not that simple (queuing theory was invented for telephony) – practical problems with CPU slicing, disk caching
  - Central limit theorem – small print – we need finite standard deviation - *and our distributions mostly do not have it!*
- (disclaimer: that does NOT mean we should forget traditional statistics and queuing theory!)

Non-computer example - height



Non-computer example: income

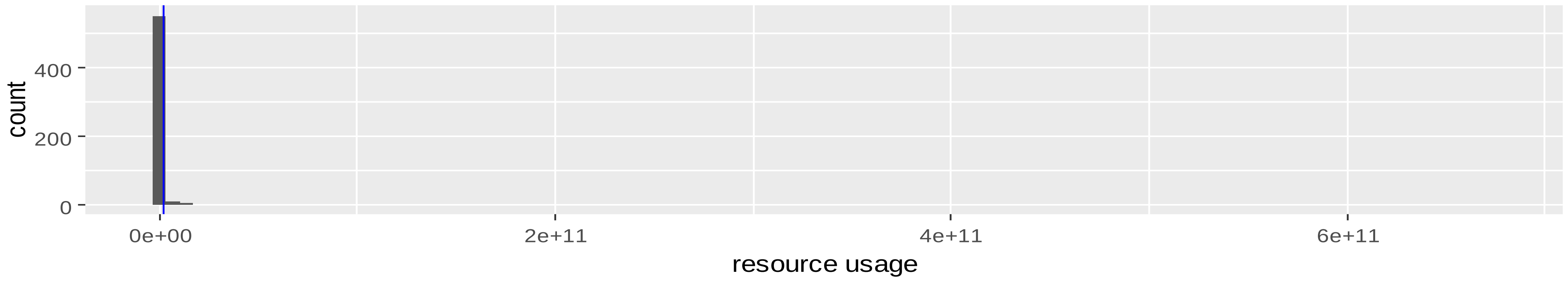


Resource usage by account

Extreme version of income distribution

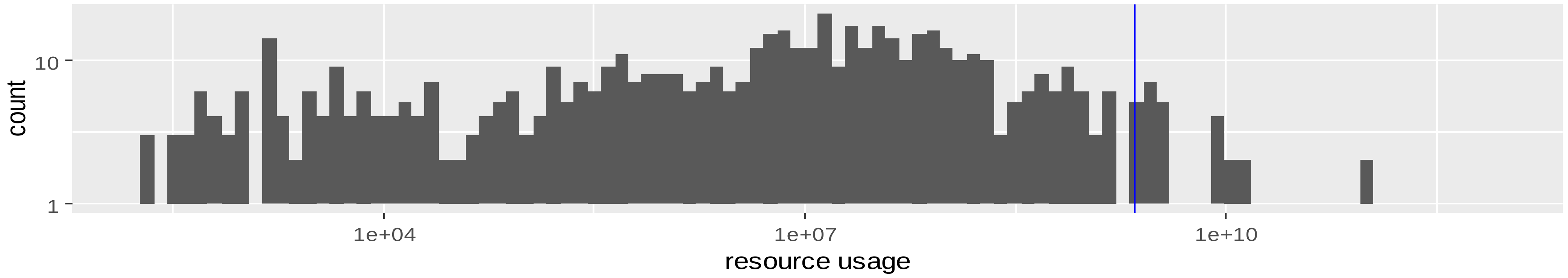
distribution of resource usage

average: 2211720713.42   average without top 1: 1039105860.16



distribution of resource usage - log-log histogram

average: 2211720713.42   average without top 1: 1039105860.16

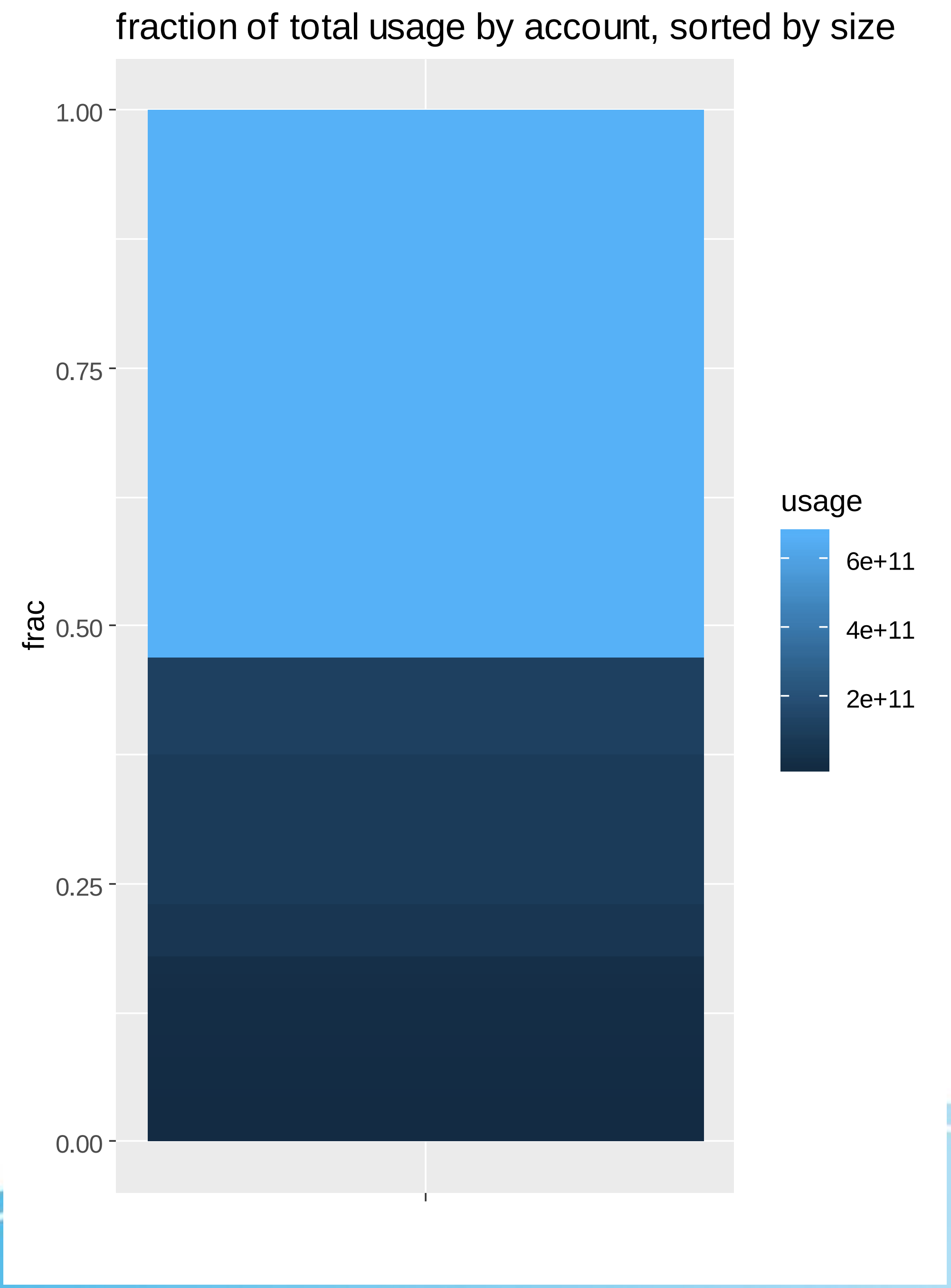


resource usage by account in one cluster

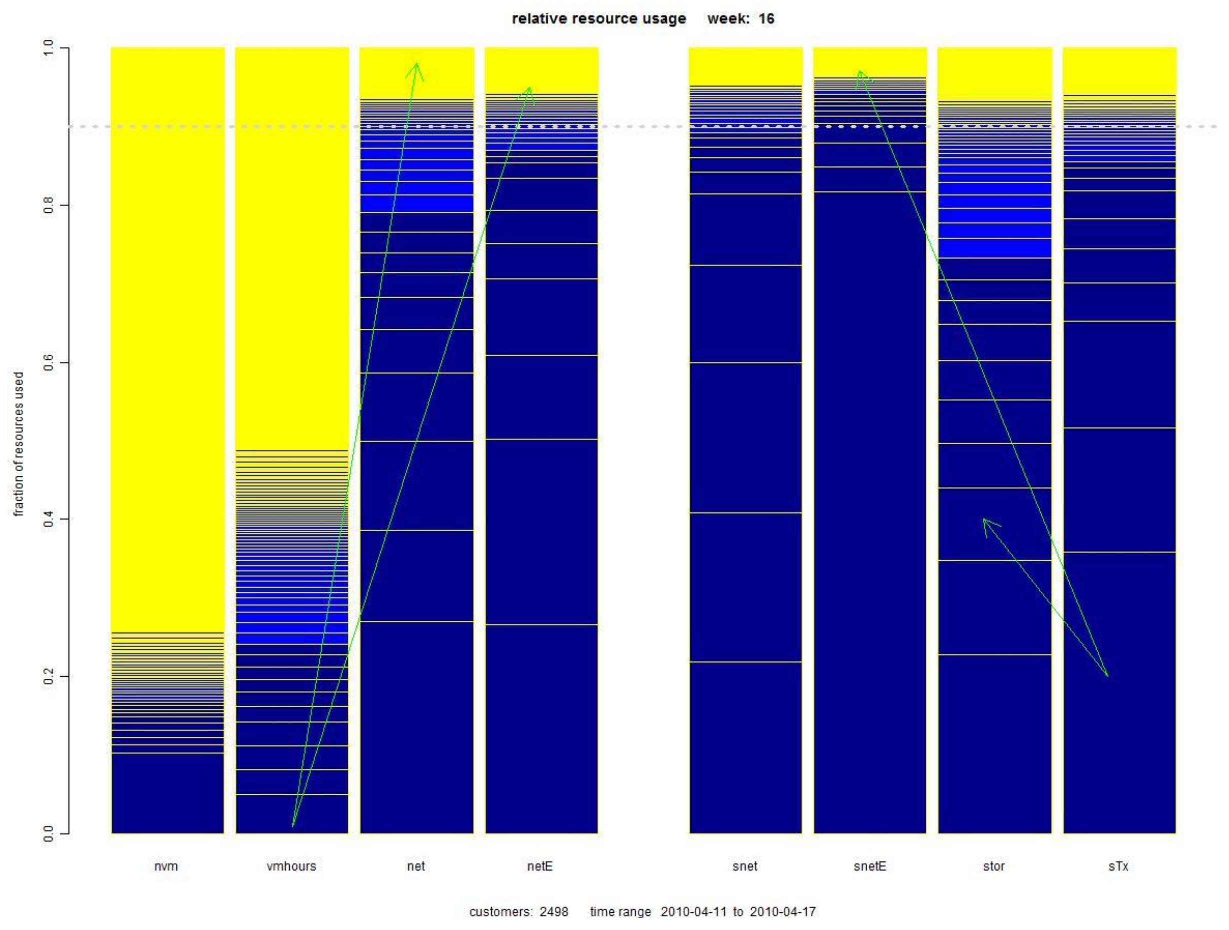
572 accounts, values (188 ,188 ,196 ,283 ,304 ,325 ,341,424 ….. 64971186176 ,90129686528 ,92741255168 ,118879485952 ,671774801920)

Of note:

- The largest account usage is 50% of the total
- The second largest account usage is 9% of the total
- Standard deviation to average is 13…
- Skewness 21, kurtosis 483 – astronomical (for normal distribution should be 0,0)



Heavy-tailed distributions abound



Resource usage by individual accounts in one compute (left) and one storage (right) cluster.

Each blue strip is one account, yellow denotes plenty of small accounts [just edges]

Further details: Loboz, C., *Cloud Resource Usage – Heavy Tailed Distributions Invalidating Traditional Capacity Planning Models*, Special Edition of Journal of Grid Computing, J. Grid Comput. 10(1): 85-108 (2012)

does it mellow with age?

Percentage of users using 95% of resources

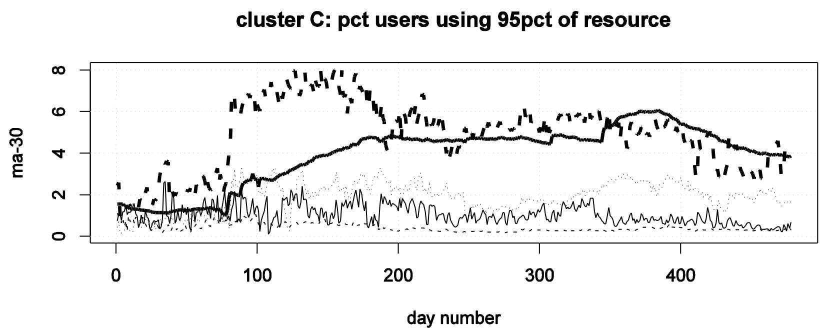
thick solid - StorageUsed,

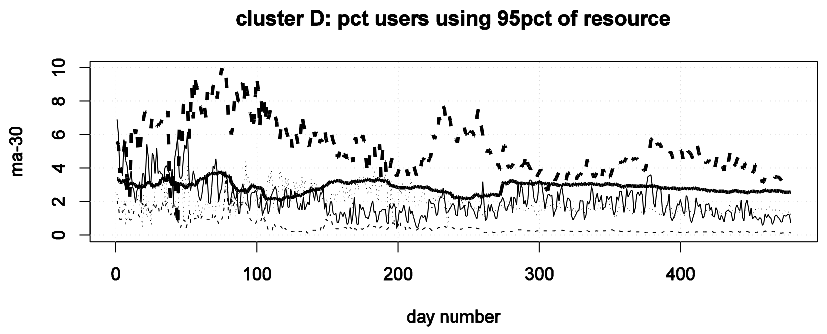
thick dashed - Tx,

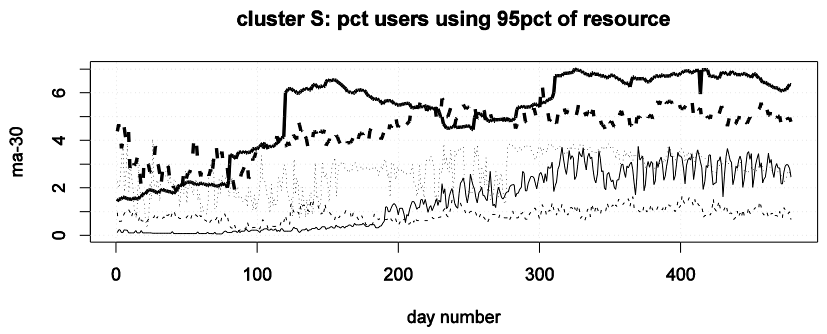
solid - EgressTransfer,

dashed - InterDCtransfer,

dotted – InternalTransfer.







Summary so far

1. Resource usage distributions are far from normal.
2. Use log plots and log-log plots to avoid losing details.
3. Use log-transformations to estimate parameters – in our sample the cofv/skew/kurt goes from 13/21/483 to 0.3/-0.3/-0.5 after log-transformation (it may not work)
4. Do *not* hope it will get better with time/maturity

- Be aware that averaging usually does not work for heavy-tailed distributions
- And most other stats are dodgy

Moving from static to time series

1. So far we considered data from a single day –’static’ distribution.
2. These distributions on a single day influence ‘dynamic’ behavior – from day to day – a time series of resource usage

System dynamics

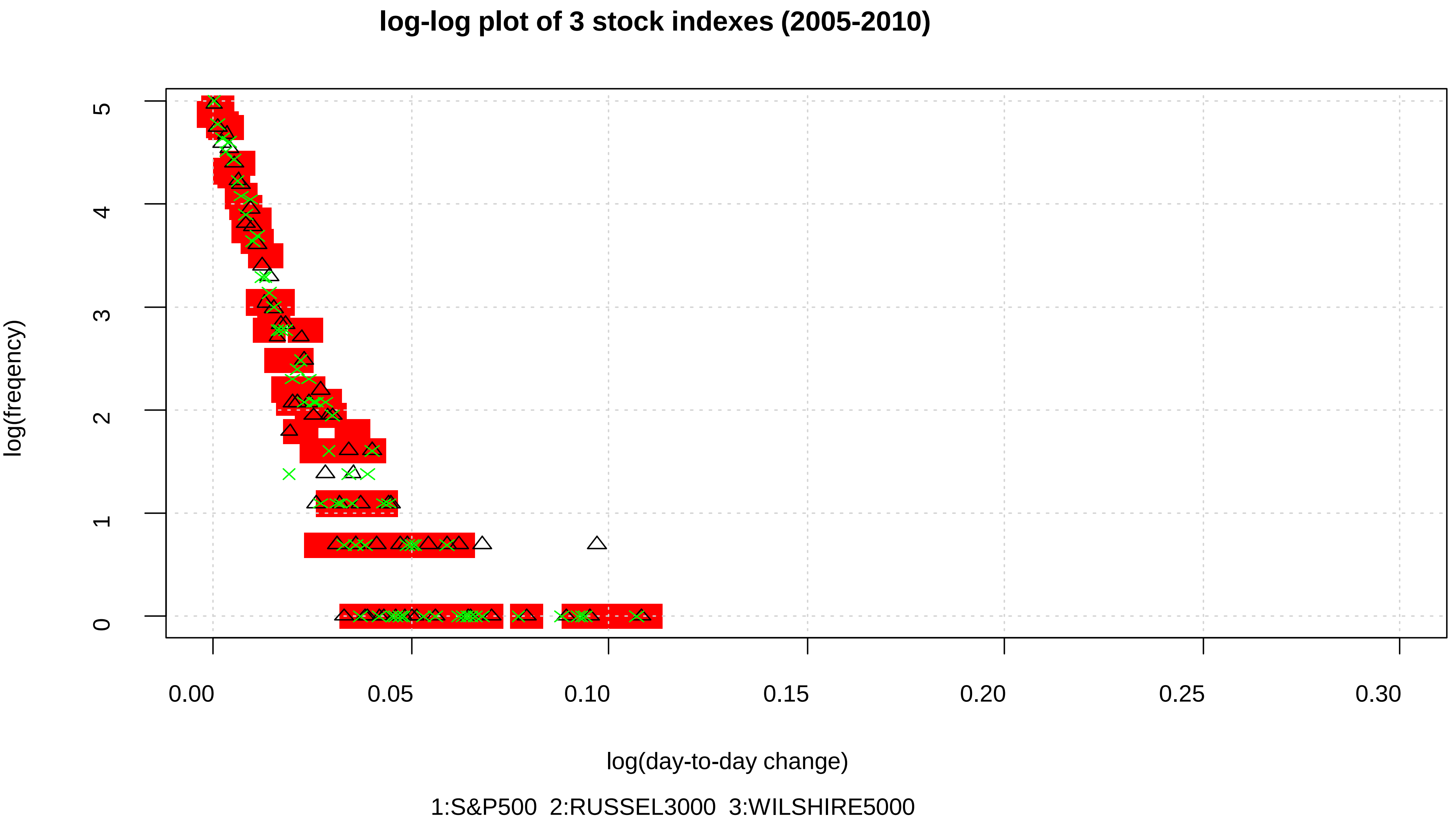
Thought experiment…

- You manage two data centers, each with 1PB storage, 1,000 customers each with p=0.1 of day-to-day usage change of 10% in any direction
- DC1 customers use on average 1TB with standard deviation of 1TB (so the largest customer uses about 4TB and the smallest are close to zero)
- DC2 - the largest customer uses 900TB and the rest use about 0.1TB each.
- What is the chance of day-to-day DC1, DC2 change in usage of 10%?
    - In round numbers: 0% and 10% respectively

Capacity planning –about the future…

- We all know that trading stocks is usually a losing proposition - the main problem is the volatility of stock prices – they change randomly
- We estimate volatility by histogramming day-to-day relative changes in stock price: log(price$_{t+1}$/price$_t$)
- Stocks are crazy to let’s take some stock indexes – they should be less volatile

Forecasting – volatility – stock indexes



Significant difference from 'normal' capacity planning

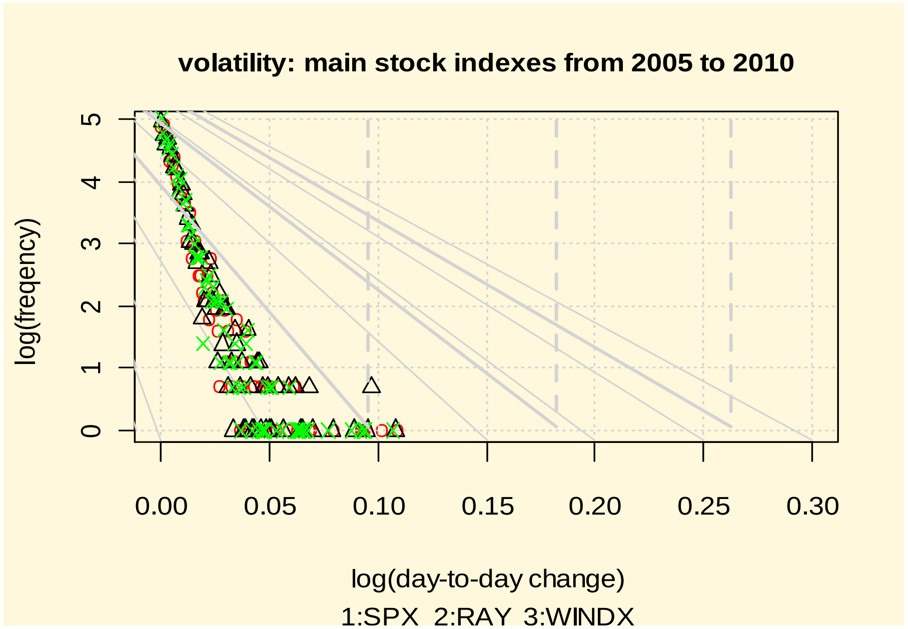
Underlying power-law (fractal) distribution of resource usage

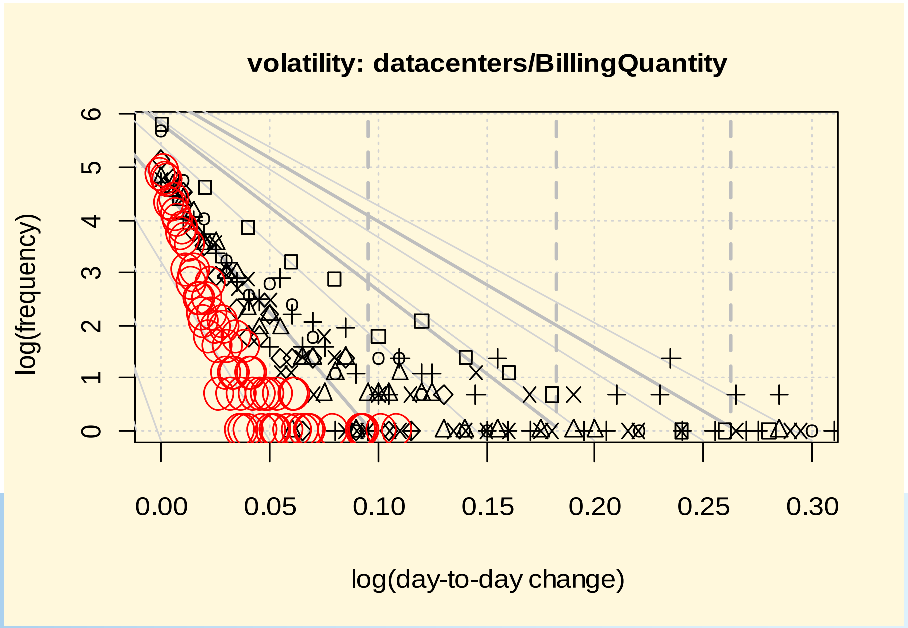
- so no law of large numbers, no 'averaging out' – we have big sudden swings

Much higher volatility than stock indexes

- so no predictability in the traditional sense (we have random walk with non-normal increments)

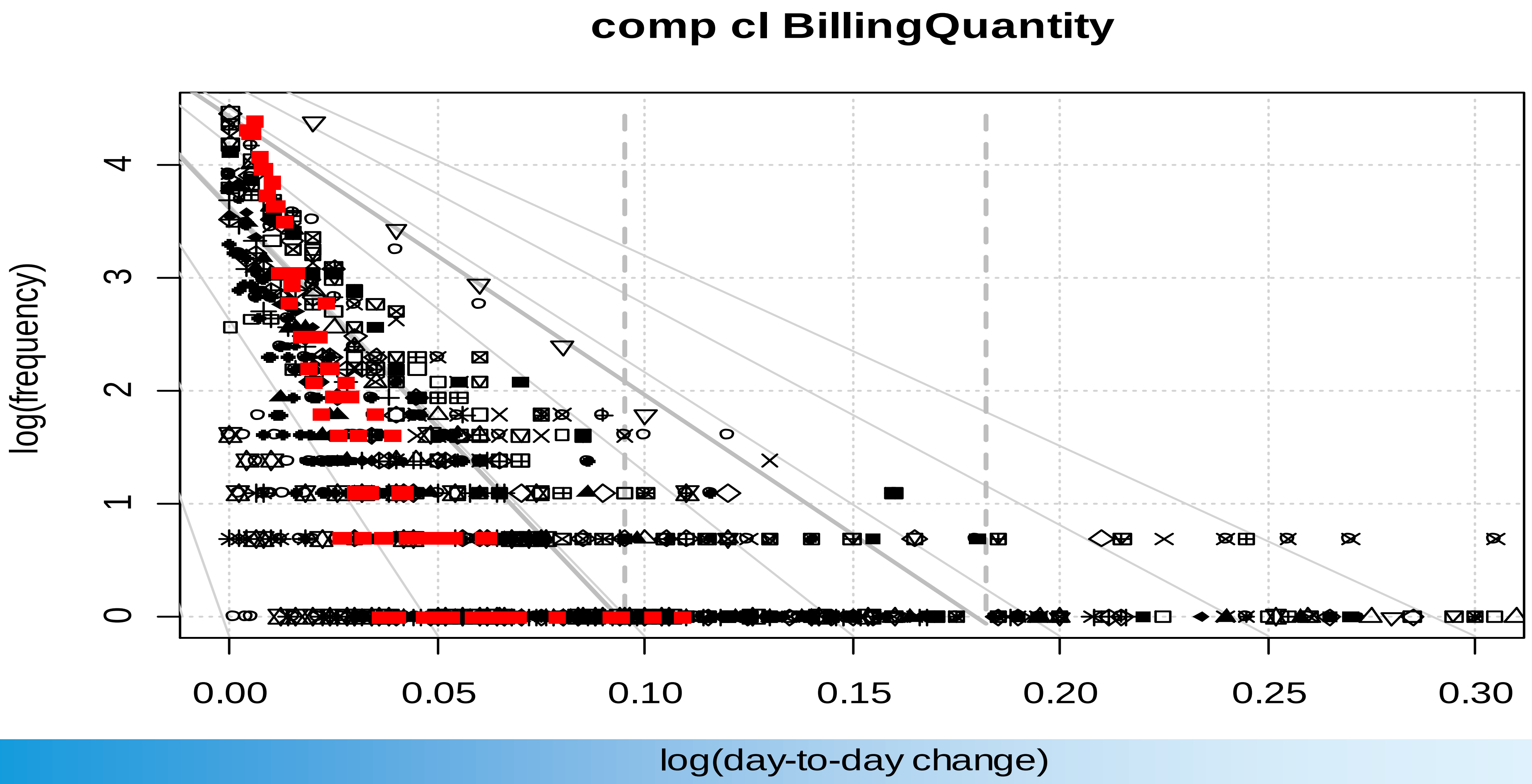
Traditional capacity planning and performance analysis are based on queuing systems and normal distributions – we need to update that - new concepts, new methods, new understandings





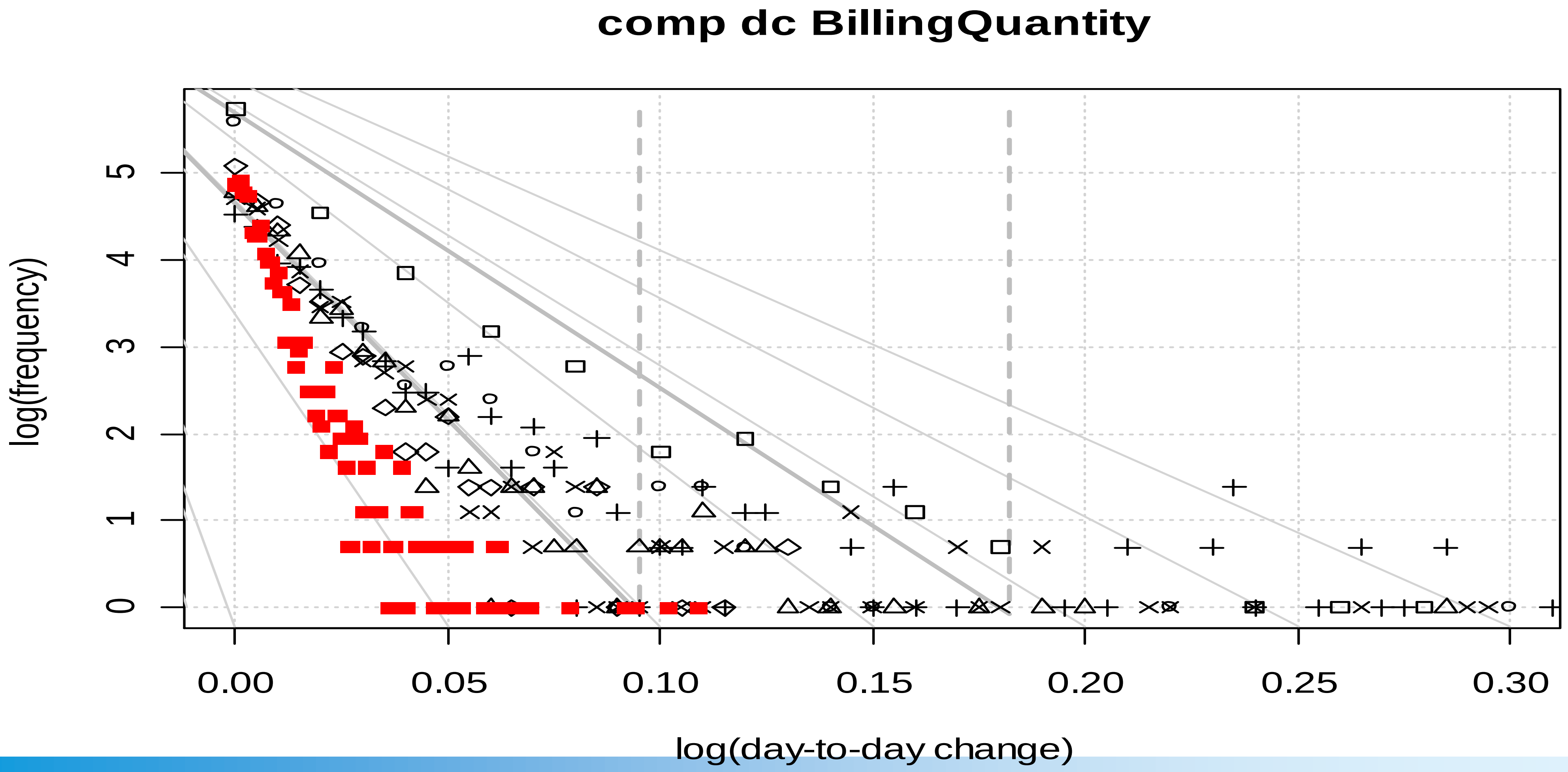
Forecasting – volatility – clusters

CPU usage distribution is less heavy-tailed than distribution for other resources (check the earlier bar graphs)



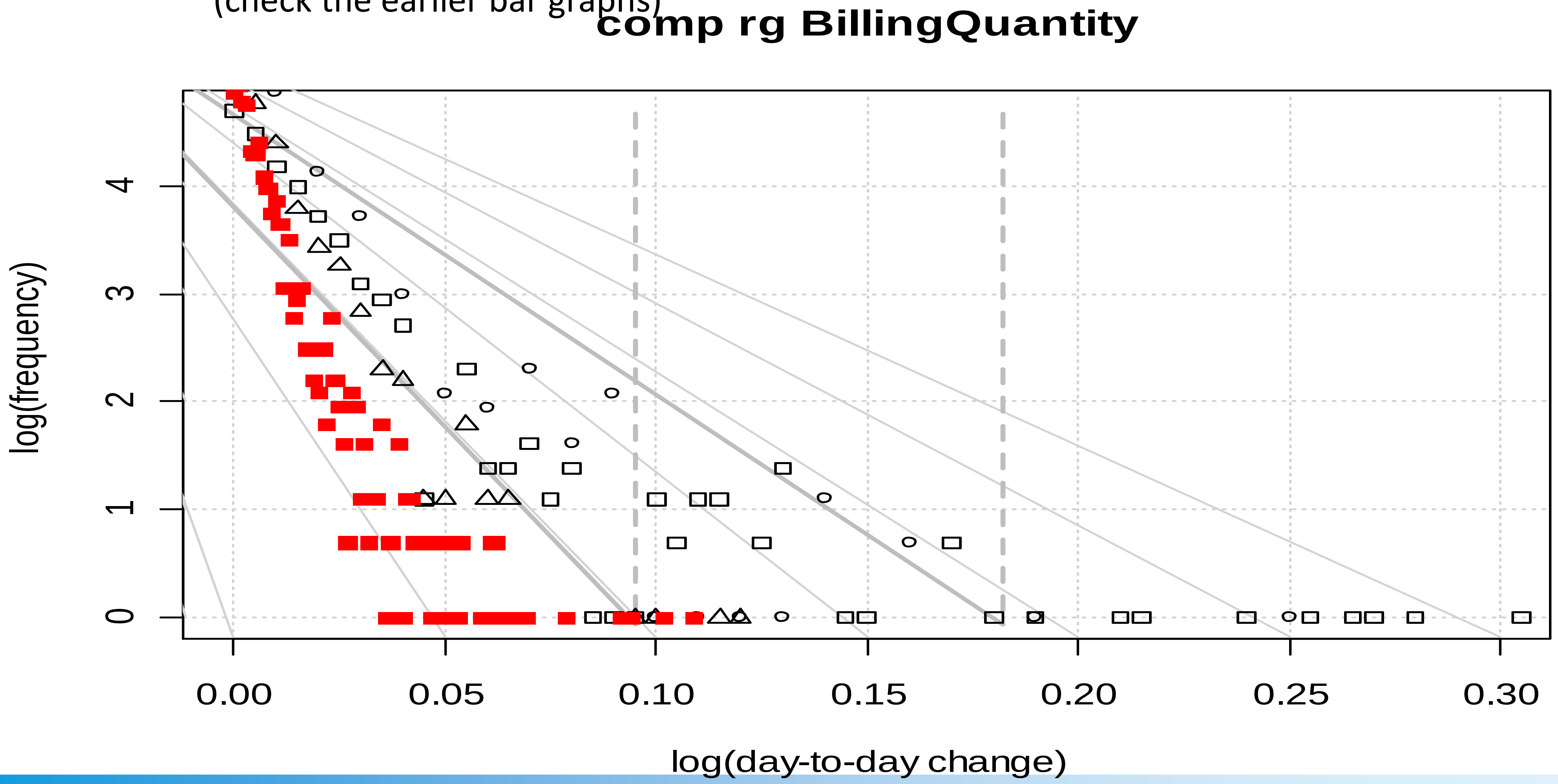
Forecasting – volatility – data centers

CPU usage distribution is less heavy-tailed than for other resources
(check the earlier bar graphs)



Forecasting – volatility – regions

CPU usage distribution is less heavy-tailed than for other resources (check the earlier bar graphs)



Cluster-level volatility for other resources– compared with S&P500

Spectrum of log-returns;

1-*StorageUsed*,

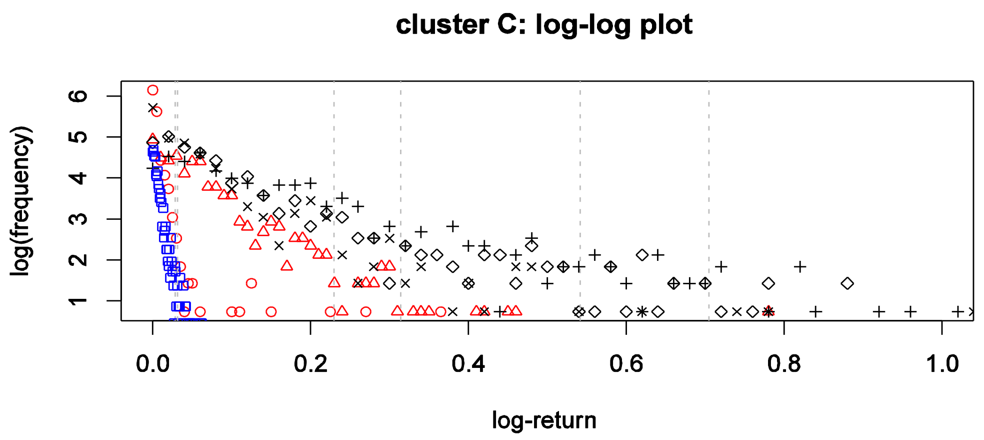
2-*Tx*,

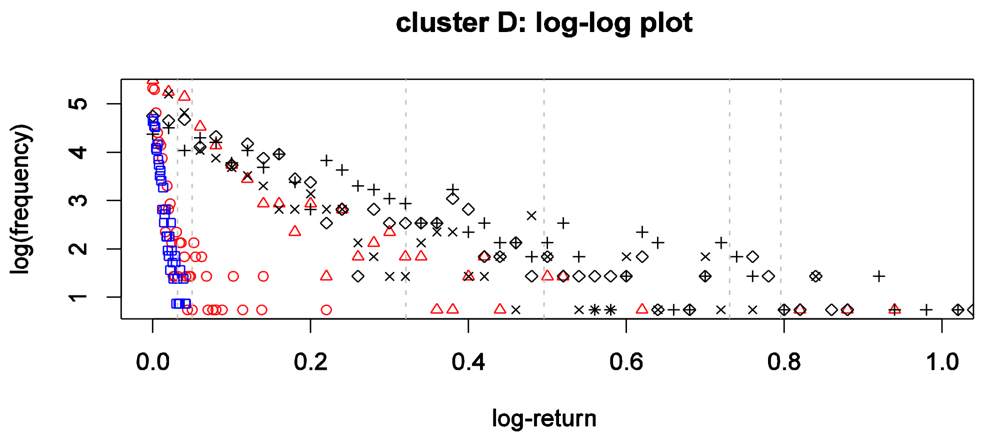
3-*EgresTransfer*,

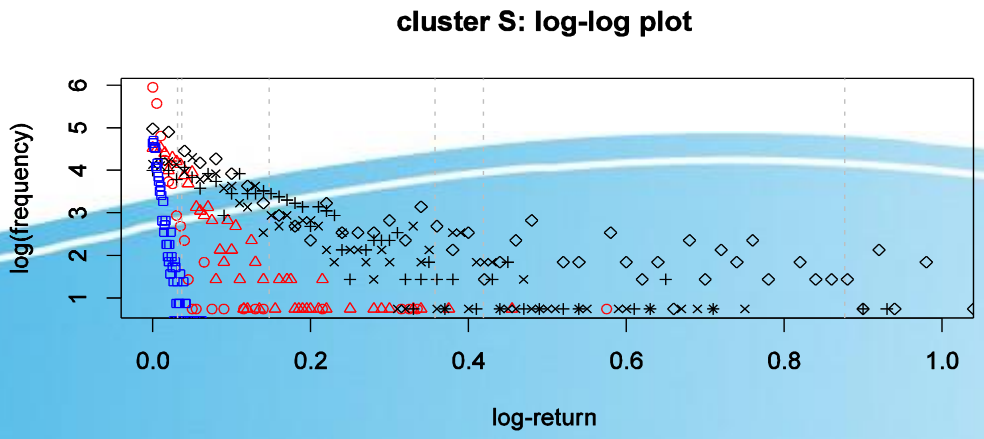
4-*InternalTransfer*,

5-*InterDCTransfer*;

thin vertical lines are 95th percentiles for resources, thick vertical lines denote 95th percentile and maximum for the S&P500 distribution; blue-square-S&P500..







What lies beneath

1. buzzwords: chaos theory, nonlinear dynamic systems, heavy-tailed distributions, Pareto distributions, power-law distributions
2. 80/20 rule (that's from Pareto)
3. [Though we seem to have 95/5 rule… we are extreme Pareto ☺]

Theory

Power-law probability distribution: $p(x) \propto x^{-\alpha}$, where α is the ‘scaling parameter’.

In practice power law applies only to a subset of data, values above threshold.

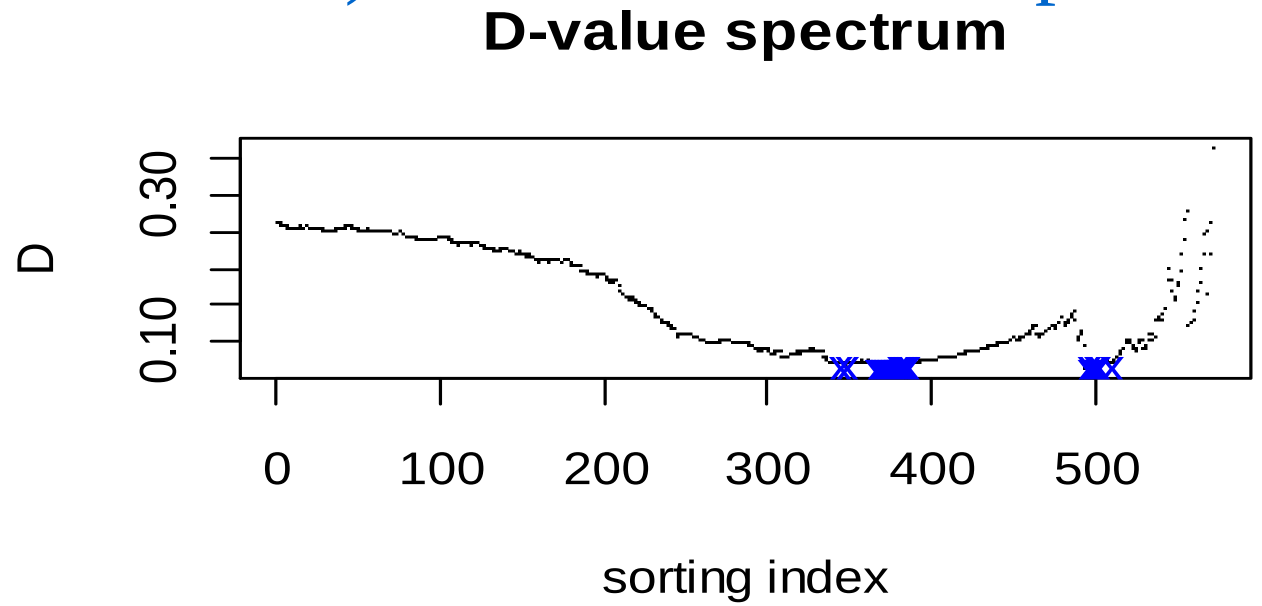
Shalizi: “*In practice, we can rarely, if ever, be certain that an observed quantity is drawn from a power-law distribution. The most we can say is that our observations are consistent with the hypothesis that x is drawn from a distribution of the form (…). In some cases we may also be able to rule out some other competing hypotheses.*”

Estimating power-law distribution

1. Log-log plots – bad, inaccurate - but practical
2. Finding the heavy-tail threshold the right way
   1. Probability distribution is $p(x) = \frac{\alpha-1}{x_{min}} \left(\frac{x}{x_{min}}\right)^{-\alpha}$, where x_{min} is the smallest value for which power-law holds
   2. to estimate x_{min} and α we use MLE with
      $\alpha = 1 + n \left(\sum_k \ln\left(\frac{x_i}{x_{min}}\right)\right)^{-1}$ giving
      $$MLE_{power} = n\,(\ln(\alpha-1) - \ln(\mathrm{x}_{xmin})) - \alpha \sum_k \ln(\frac{x_i}{x_{min}})$$
   3. (an R function for that attached in the Appendix)

Finding the power-law tail

We have 64 data points in the power-law tail (right-most blue cross) – and few other hopefuls.



11% of accounts in the tail – but 98.4% of all resource usage – almost everything is in the tail!

Some consequences

1. Volatilities in resource usage much higher than for stock indexes – problems with forecasting, confidence bounds.
2. Volatility of the mean is unknown and standard deviation mostly does not exist.
3. Standard capacity calculations implicitly assume tractable distributions – which are rare in practice.
4. So what can be done ? Aspects:
    1. reduce volatility – throttling traffic, usage caps as a part of an SLA, managing most volatile accounts.
    2. the volatility reduction techniques have to be incorporated early in the design *and* administration.
    3. Smoothing of time series should not be used blindly.
    4. The operational cost and customer problem – volatility may require 50% of capacity buffer instead of 10% - that would cost almost 40% as much money**.**
    5. If you persist in using traditional capacity formulas – be prepared for failure

Some meta-comments

1. We do *not* think ‘heavy-tails’ – we tend to fall back on ‘normal distribution’ thinking.
2. It is an extremely difficult habit to overcome. ‘It will average itself out soon’ – it will not – learn to live with it.
3. Most life is non-linear and fractal… normal distributions are the exception, *not* the rule
4. And *our subject matter* is more fractal than most others

Some background reading

- Raj Jain *The Art of Computer Systems Performance Analysis ;* classic summary, useful compendium, basic statistical info, queueing theory
- Clauset, A., Shalizi, C. R. and Newman, M. E. J. (2009). "Power-law distributions in empirical data". SIAM Review 51: 661–703.
- Newman, M. E. J., 2006. Power laws, Pareto distributions and Zipf's law. *Cont. Phys.*, 46, 323–351.
- Shalizi, C Power Law Distributions, 1/f Noise, Long-Memory Time Series http://cscs.umich.edu/~crshalizi/notabene/power-laws.html
- Loboz, C., *Cloud Resource Usage – Heavy Tailed Distributions Invalidating Traditional Capacity Planning Models*, Special Edition of Journal of Grid Computing, J. Grid Comput. 10(1): 85-108 (2012)
- James, A. and Plank, M. J. On fitting power laws to ecological data arxiv:0712.0613
- Nassim Nicholas Taleb *Black Swan* [deep background]
- James Gleick *Chaos* [even deeper background]

Code by Shalizi (I think)

```
plfit <-function(x=rpareto(1000,10,2.5),method="limit",value=c(),finite=TRUE,nowarn=TRUE){
   #init method value to NULL
   vec <- c() ; sampl <- c() ; limit <- c()
#  test and trap for bad input
   switch(method,
     range = vec <- value,
     sample = sampl <- value,
     limit = limit <- value,
     argok <- 0)

   if(exists("argok")){stop("(plfit) Unrecognized method")}

   if( !is.null(vec) && (!is.vector(vec) || min(vec)<=1 || length(vec)<=1) ){
     print(paste("(plfit) Error: "range" argument must contain a vector > 1; using default."))
     vec <- c()
   }
   if( !is.null(sampl) && ( !(sampl==floor(sampl)) ||  length(sampl)>1 || sampl<2 ) ){
     print(paste("(plfit) Error: "sample" argument must be a positive integer > 2; using default."))
     sample <- c()
   }
```

Sample data

```
v = c(188 ,188 ,196 ,283 ,304 ,325 ,341,424 ,441 ,443 ,476 ,480 ,484 ,529,541 ,544 ,583 ,604 ,626 ,659 ,822,824 ,858 ,914 ,953 ,1004 ,1056 ,1075

      ,1077 ,1295 ,1543 ,1582 ,1587 ,1592 ,1594,1606 ,1630 ,1639 ,1655 ,1658 ,1666 ,1677,1679 ,1696 ,1712 ,1732 ,2012 ,2085 ,2172

      ,2220 ,2691 ,2906 ,2910 ,3035 ,3096 ,3156,3388 ,3576 ,3655 ,3741 ,4254 ,4369 ,4436,4592 ,4625 ,4628 ,4695 ,4971 ,5142 ,5334

      ,5513 ,6054 ,6337 ,6563 ,6633 ,6653 ,6669,7418 ,7826 ,8780 ,8884 ,9397 ,9783,10266,10378,10908,12186,13062,13082,14167,15110

      ,15289,15928,16733,17874,17905,19900,20054,20850,21086,21351,21593,22700,26007,27840,34570,37050,41589,42210,45957,55685,56976

      ,58350,59712,61945,62061,68480,70524,72307,76440,77863,80891,83254,84753,90892,101112,103673,113586,117171,130616,133419,142528,147014

      ,149385,161956,162259,165056,166947,167375,169320,171395,183200,199417,205187,205294,211752,228929,238721,239193,249623,260805,263377,279410,292641

      ,294107,309121,321433,326674,352603,376704,387356,390392,392464,402628,402967,423925,433166,440169,444651,446759,452651,457631,463437,485926,507800

      ,518333,519151,519164,529573,555638,604183,628063,630479,634717,636041,668387,732610,740794,773755,779418,814182,835845,842910,853300,866992,899083

      ,938743,939946,947816,964552,965179 ,1014971 ,1107057,1108742 ,1169645 ,1176419 ,1178216 ,1204998 ,1221241 ,1316501,1456291 ,1517163 ,1541788 ,1598074
,1632295 ,1637677 ,1710648

      ,1712867 ,1750058 ,1778146 ,1888581 ,1903817 ,2062790 ,2123088,2129209 ,2136790 ,2170535 ,2312321 ,2357405 ,2403536 ,2534857,2610294 ,2756608 ,2862520
,2893348 ,3000182 ,3192640 ,3220880

      ,3294378 ,3315556 ,3517828 ,3567835 ,3789428 ,3827286 ,3920232,4195811 ,4403574 ,4430743 ,4453488 ,4507512 ,4525905 ,4577606,4590208 ,4753184 ,4909445
,4999500 ,5027940 ,5146166 ,5302474

      ,5325075 ,5528844 ,5584281 ,5760840 ,5769281 ,5958012 ,5958385,6014067 ,6027073 ,6104304 ,6158539 ,6237618 ,6277032 ,6363781,6419826 ,6490008 ,6582066
,6671276 ,6808893 ,7001008 ,7111015

      ,7223519 ,7304495 ,7364244 ,7447713 ,7641262 ,7668925 ,7783074,7784824 ,8113263 ,8150515 ,8312051 ,8654176 ,8980038 ,9086761,9126250 ,9441554 ,9517220
,9577492 ,9642090 ,9647820 ,9952659

      ,9979308 ,10573264 ,10753338 ,10776849 ,10784664 ,10886581 ,10982208,11087826 ,11998224 ,12224230 ,12256801 ,12372216 ,12513205 ,12553027,12603149
,12651007 ,12701768 ,12760634 ,12791800 ,13091895 ,13130881
```