Experimental Approaches in Computer Science

Dror Feitelson
Hebrew University

Lecture 8 – More Workloads

Previous lecture:

- Representativeness of workloads
- Workload data and sanitization
- Heavy tails

This lecture:

- Burstiness and self similarity
- Locality of sampling

Burstiness and Self-Similarity

Let's make the following assumptions about how new work (jobs, packets, requests) arrives at a computer system:

- Work items arrive independently of each other
- They can arrive at any instant with uniform probability
- We measure time at fine granularity, so at each instant at most one arrives

This defines a Poisson process

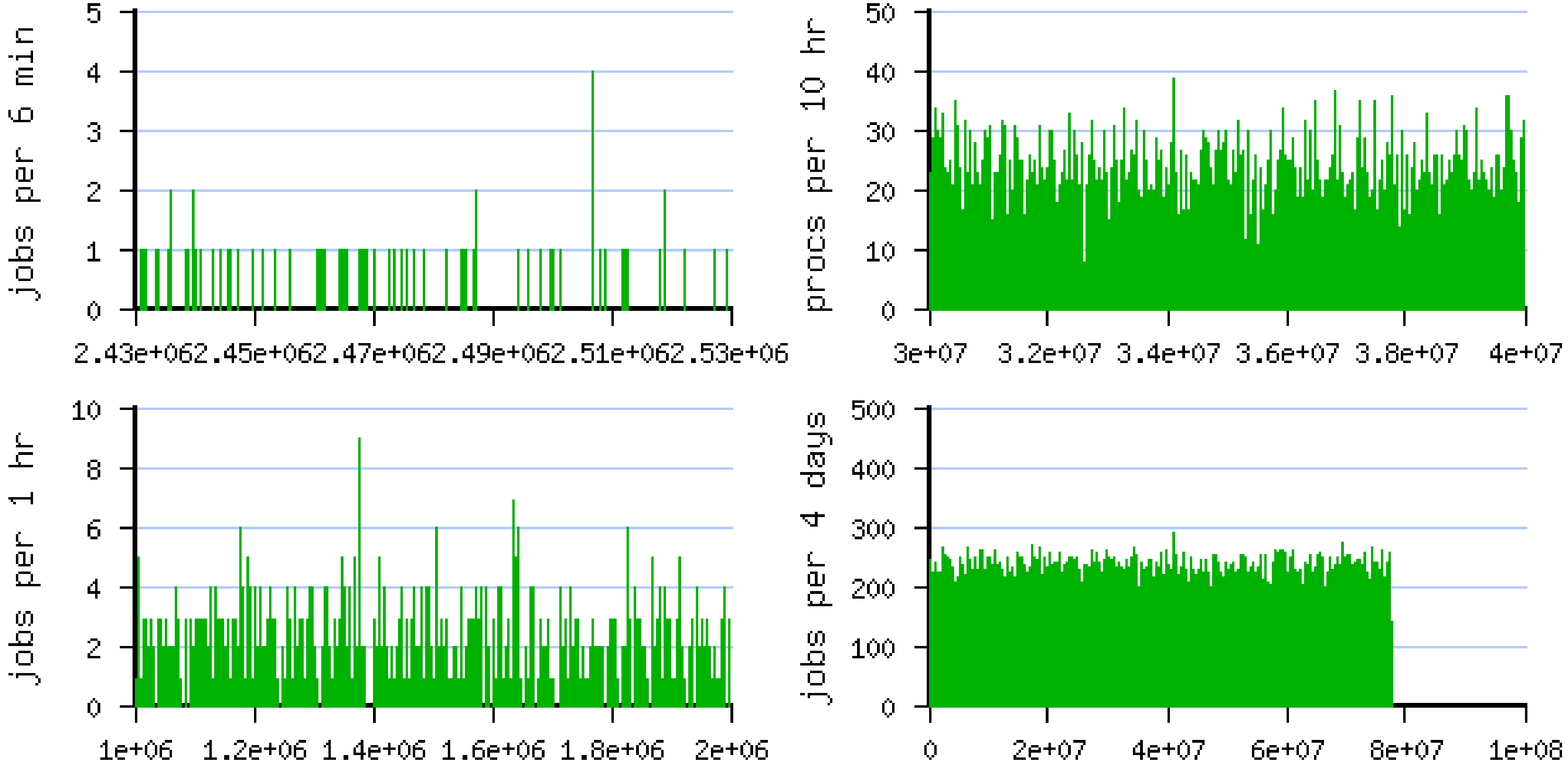
Implications of a Poisson process:

- Work arrives uniformly over time
  - No large bursts of sudden activity
  - No cycles of activity
- Inter-arrival times are exponentially distributed
  - Allows for easy simulation of arrivals without deciding in advance how many will arrive
- Merging multiple Poisson processes is also a Poisson process
- Variability is reduced with aggregation
  - If we look at a longer time, periods with more activity cancel out with periods with less activity

Checking experimentally that arrivals are Poisson:

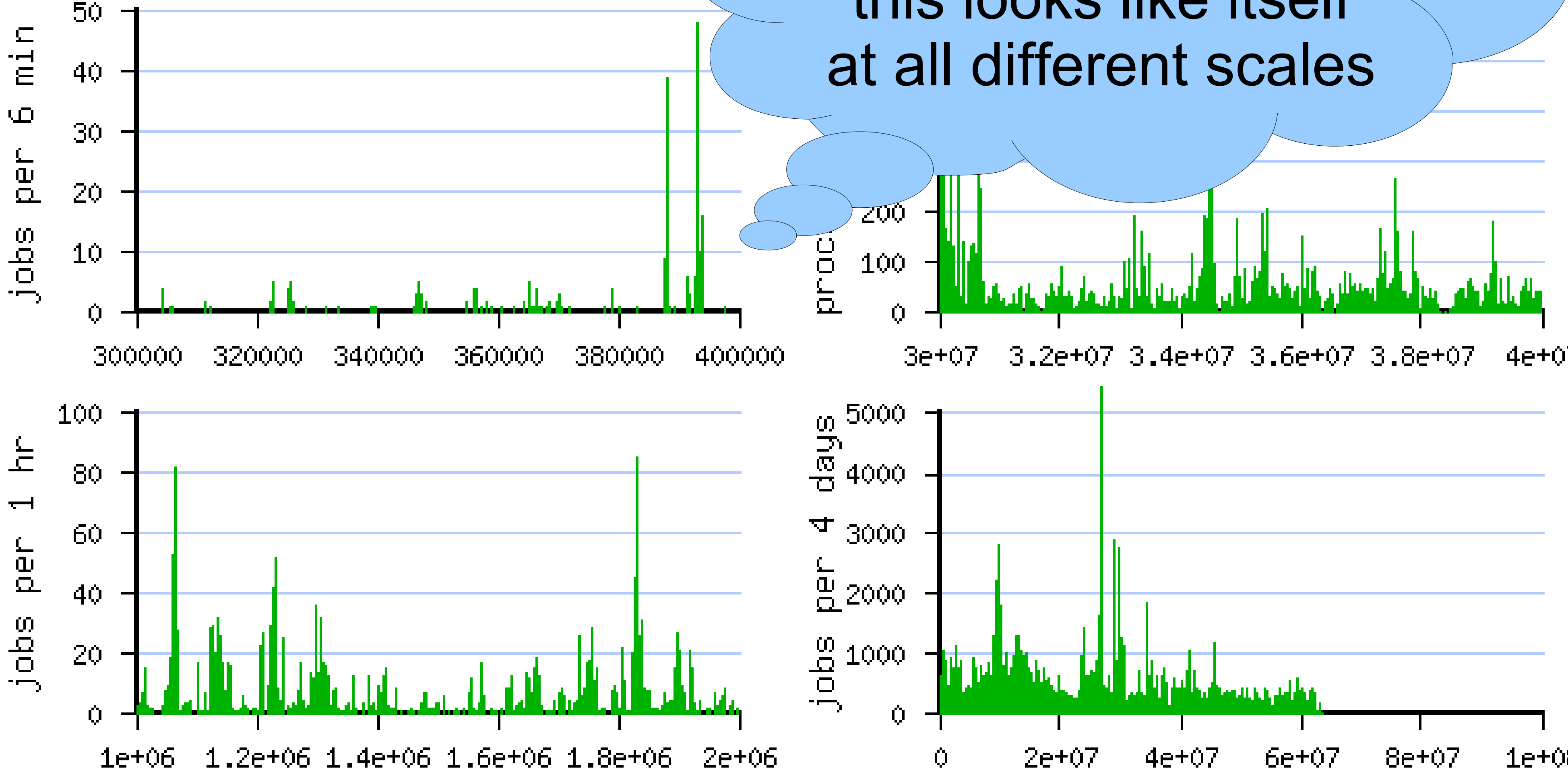
- Verify that distribution of inter-arrivals is indeed exponential
  - Compare to exponential distribution with same average arrival rate
- Verify that successive inter-arrivals are independent of each other
  - Look at correlation of successive inter-arrivals
- Verify that when aggregated the variance is reduced

Poisson arrivals aggregated

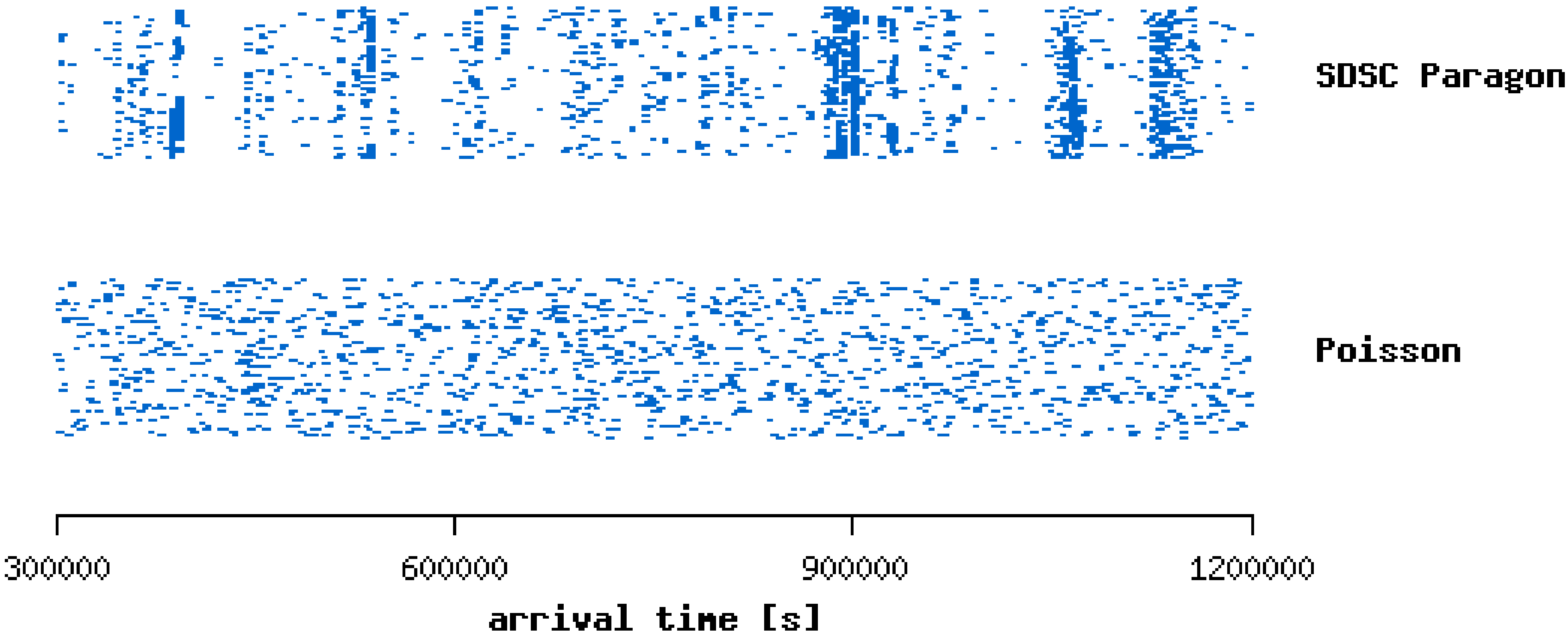


Real arrivals aggregated

The term "self-similar" derives from the fact that this looks like itself at all different scales



Another visualization using texture plots



This defines a time unit u, and plots each datum at $X = t / u$ and $Y = t \bmod u$

- Results: arrivals are often not Poisson
  - Packets in a communication network
  - Jobs to a parallel supercomputer
- But sometimes they are
  - New flows on a network
- This has implications for system capacity
  - Network buffers need to be large enough for bursts of activity
- Also need to consider other effects, e.g. the daily work cycle

The R/S metric and Hurst Parameter

How do you quantify self-similarity?

- Successive items are correlated
- So if you sum them up, you will get large deviations from the average
- Deviations larger than those of random independent items indicate self-similarity

- Start with a time series $X_1, X_2, X_3, \ldots$

  For example, X_i can be the number of packets that arrived in second i

- Center the data by subtracting its average, giving $Z_i = X_i - \bar{X}$

- Now create the sum of the first n items, for all n

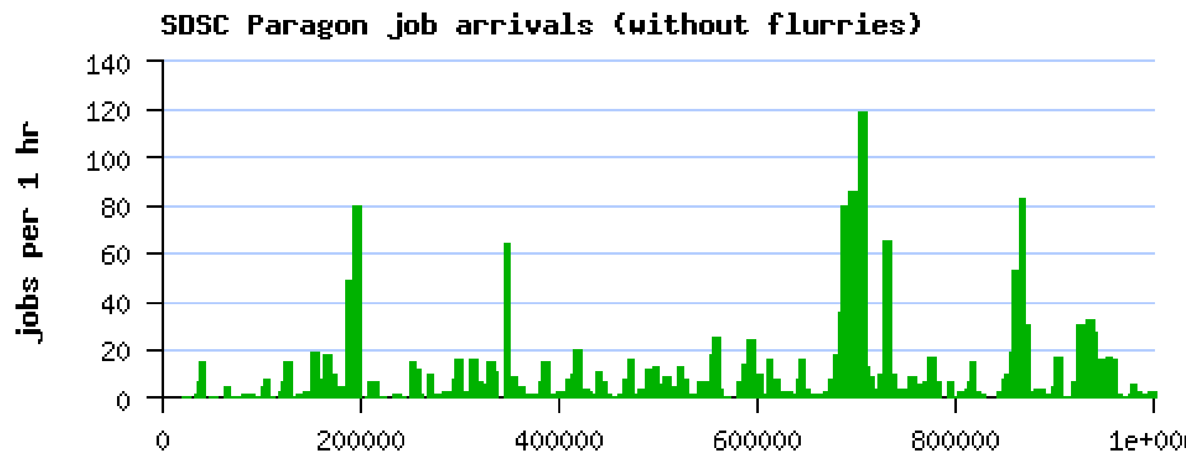
$$Y_j = \sum_{i=1}^{j} Z_i$$

  Note that $Y_n = 0$

- Finally, look at the range covered by these

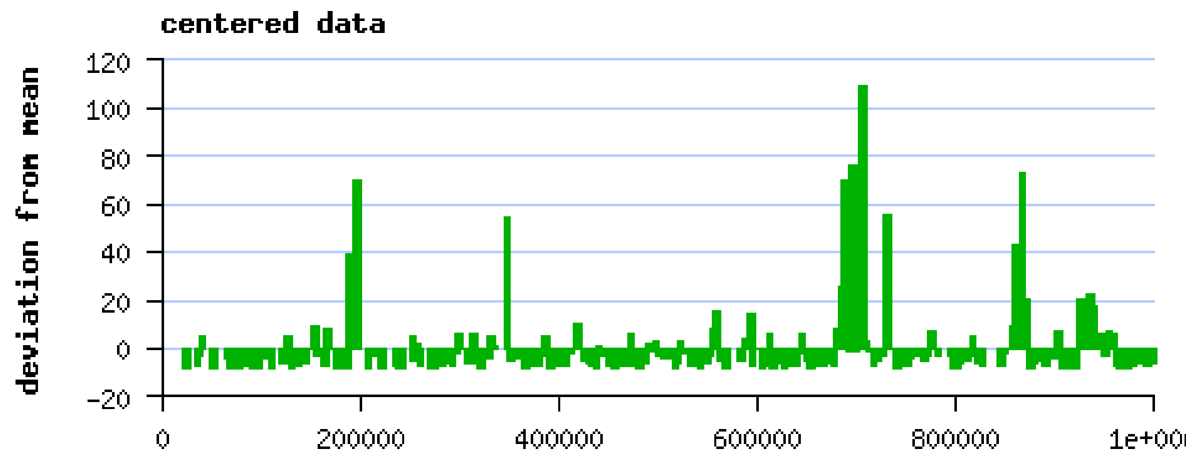
$$R_n = max_j Y_j - min_j Y_j$$

X_i



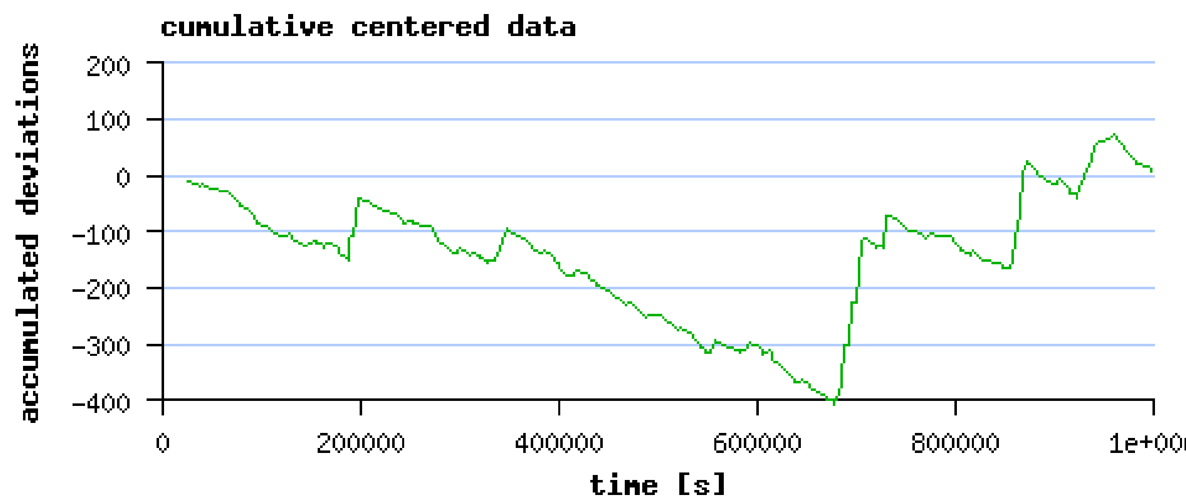
SDSC Paragon job arrivals (without flurries)
jobs per 1 hr
0
20
40
60
80
100
120
140
0
200000
400000
600000
800000
1e+006

Z_i



centered data
deviation from mean
-20
0
20
40
60
80
100
120
0
200000
400000
600000
800000
1e+006

Y_j



cumulative centered data
accumulated deviations
-400
-300
-200
-100
0
100
200
0
200000
400000
600000
800000
1e+006
time [s]

- The magnitude of R_n is related to
  - The number of consecutive steps in each direction
  - The size of each step
- To remove the second effect and focus on the first one, we divide by the standard deviation
- The model is that this grow as a power law

$$\left(\frac{R}{S}\right)_n = C\, n^H \qquad 0 \le H \le 1$$

- By taking the log, we get

$$\log\left(\frac{R}{S}\right)_n \propto H \log n$$

What happens for a random walk?

- Each step is $X_i = +1$ or $X_i = -1$
- The expected distance *squared* is

$$\begin{aligned} E[(Y_j)^2] &= E[(Y_{j-1} + X_j)^2] \\ &= E[Y_{j-1}^2] + 2E[Y_{j-1} X_j] + E[X_j^2] \\ &= E[Y_{j-1}^2] + 1 \\ &= j \end{aligned}$$

- So the root-mean-square distance is

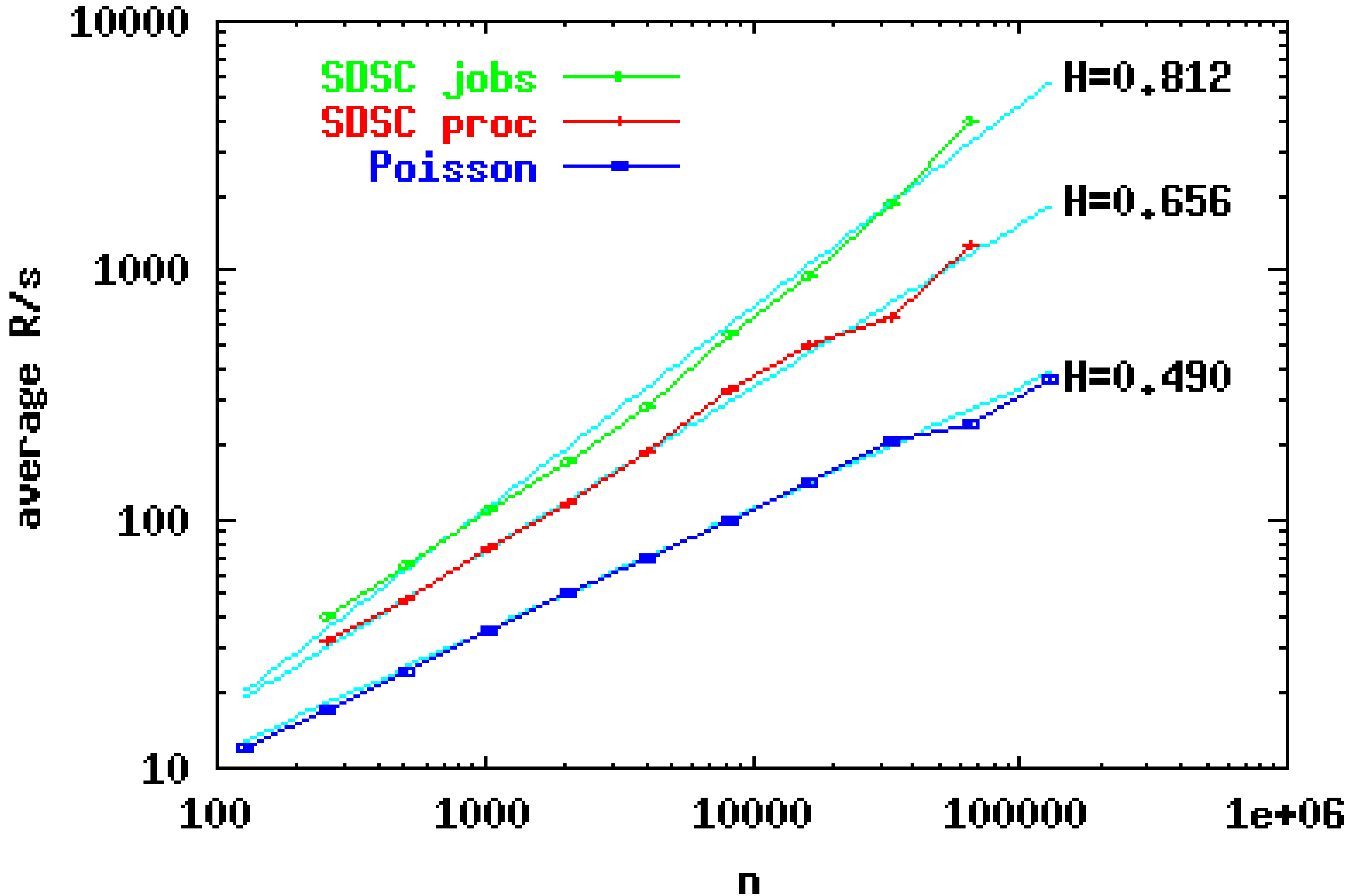
$$RMS(Y_n) = n^{0.5}$$

- And indeed we get H = 0.5

For self-similar data:

- Collect data for many different sizes *n*
- For each one, look at many different subsets of this length
- Calculate $(R/S)_n$ for each one
- Draw a pox-plot: the measured $(R/S)_n$ as a function of *n* on log-log axes
- Expect to get a straight line, with slope proportional to the Hurst parameter *H*

SDSC Paragon jobs
32768
8192
2048
512
128
32
8
2
R/s
H=0.79
16
64
256
1K
4K
16K
64K
256K
1M
n



SDSC jobs
SDSC proc
Poisson
H=0.812
H=0.656
H=0.490
10000
1000
100
10
average R/s
100
1000
10000
100000
1e+06
n

Locality of Sampling

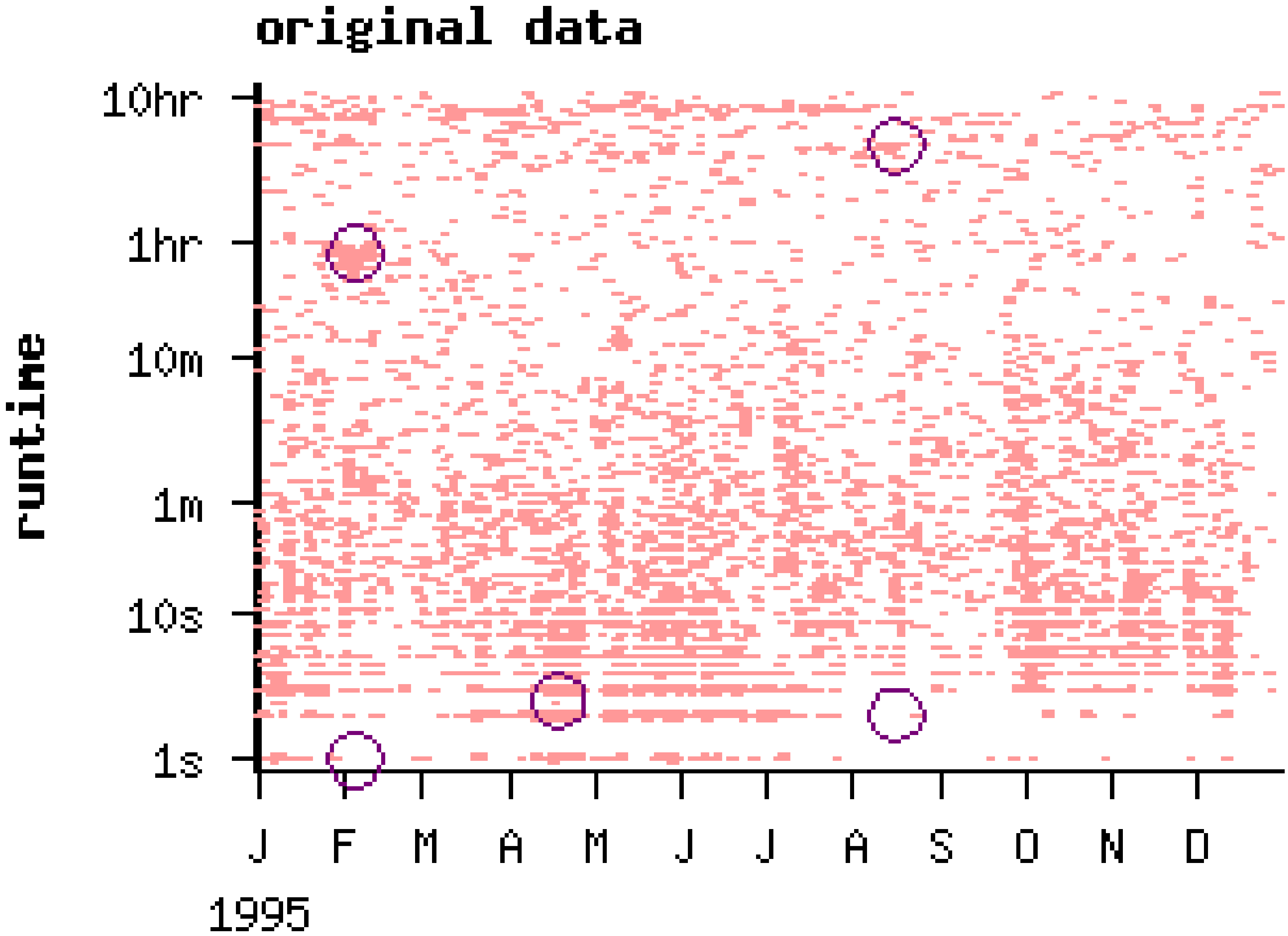
- Common model of workload generation is sampling from a distribution
  - Implied in fitting distributions to data and random variate generation in simulations
  - Implied in definition of arrival and service distributions in queueing analysis
- This requires a stationarity assumption
- But real workloads are non-stationary
  - Daily/weekly cycles
  - Workload evolution as usage changes
  - Locality in user behavior: repeated activity + shifting focus with time

Locality reduces randomness

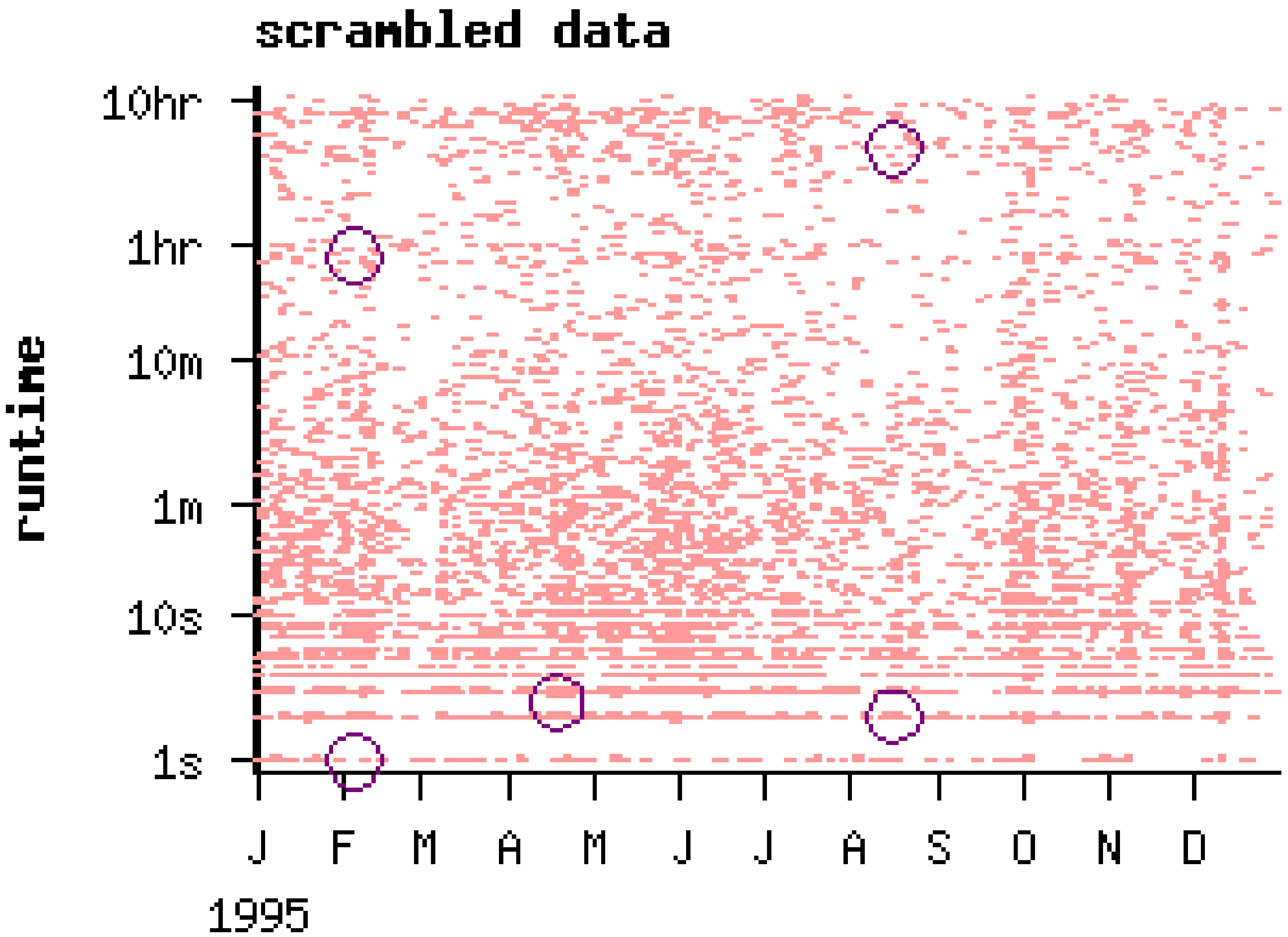
- Important for adaptive systems
  - Can learn about the workload
  - Can make predictions for the future
- Important for performance evaluations
  - Randomness is good because things tend to average out
  - Lack of randomness is harder to handle

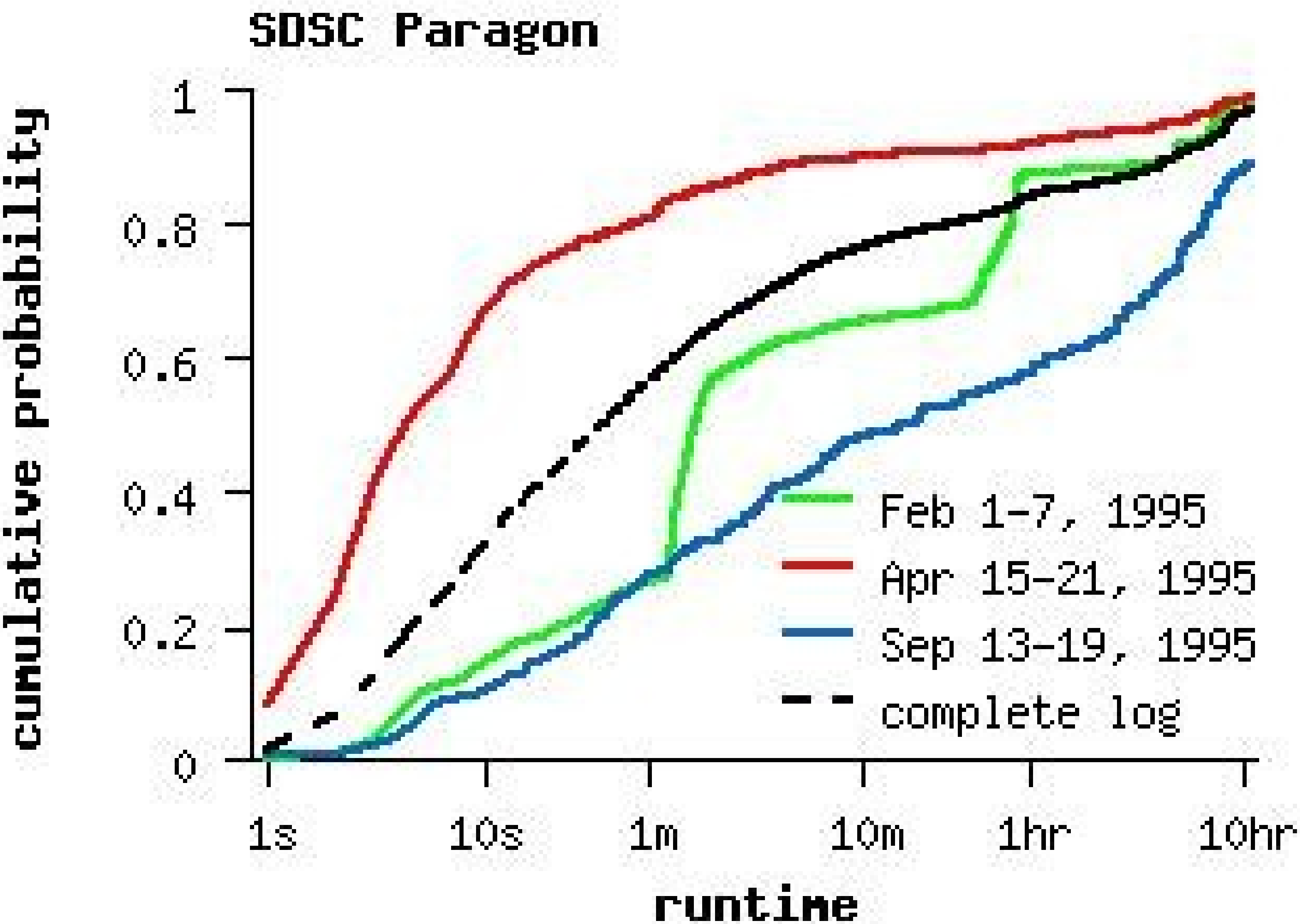
"Locality of sampling"

- Assume an underlying stationary distribution
  - e.g. empirical distribution from a long data log
- Workload is generated by a 2-level sampling process
  - Select a location within the distribution
  - Sample multiple items from this location
- Generative model of user behavior
  - At a given time, users focus on a certain project
  - While working on this project they repeatedly to the same thing



original data
runtime
10hr
1hr
10m
1m
10s
1s
J
F
M
A
M
J
J
A
S
O
N
D
1995


scrambled data
runtime
10hr
1hr
10m
1m
10s
1s
J
F
M
A
M
J
J
A
S
O
N
D
1995


SDSC Paragon
cumulative probability
1
0.8
0.6
0.4
0.2
0
1s
10s
1m
10m
1hr
10hr
runtime
Feb 1-7, 1995
Apr 15-21, 1995
Sep 13-19, 1995
complete log

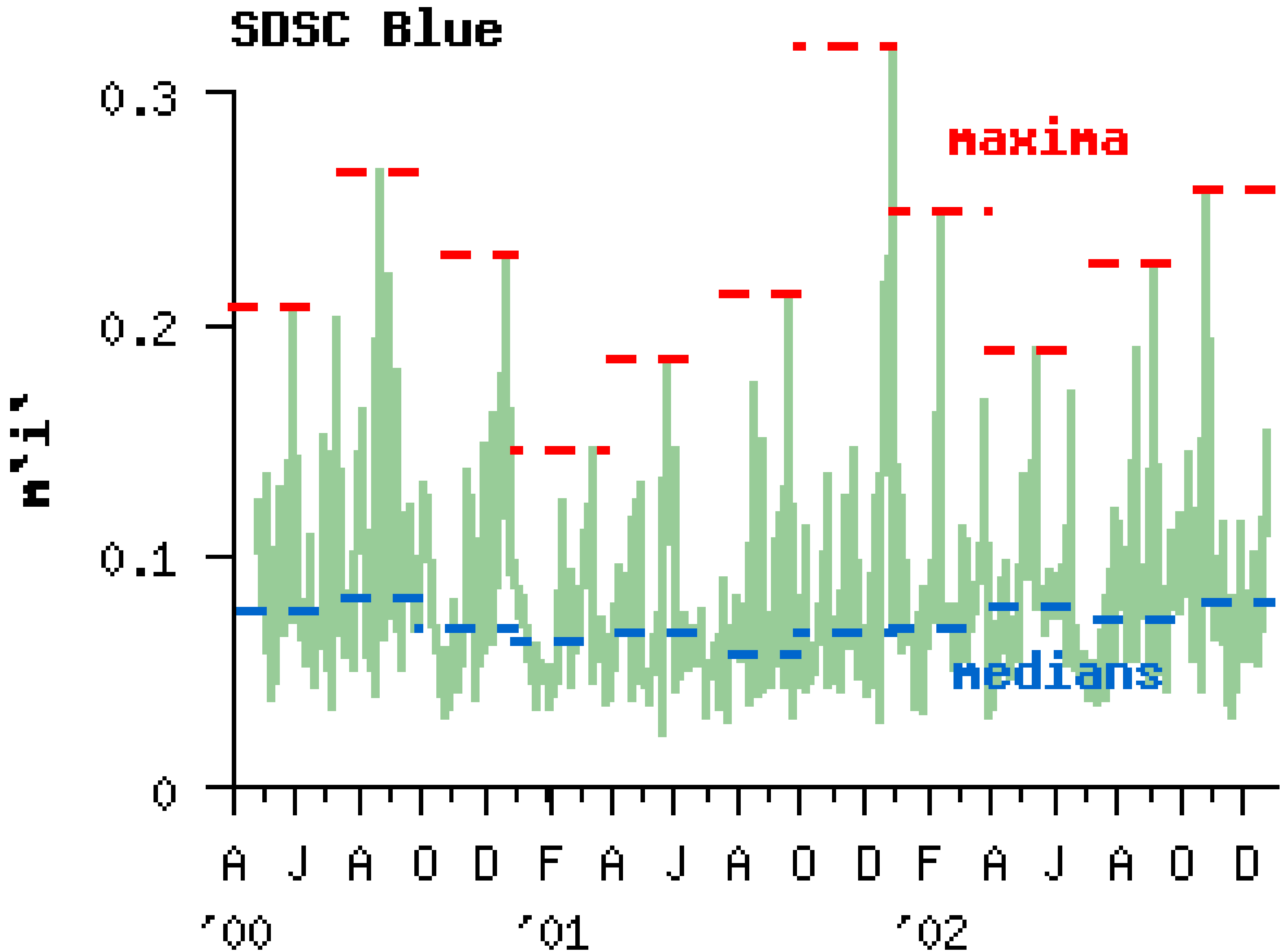
Quantifying locality of sampling:

1. Create histogram of global data, and partition into *r* equally likely ranges
2. Partition the log into slices that are long enough to contain sufficient data (>5*r* items)
3. For each slice *i* find number of items in each range o_j, and compute

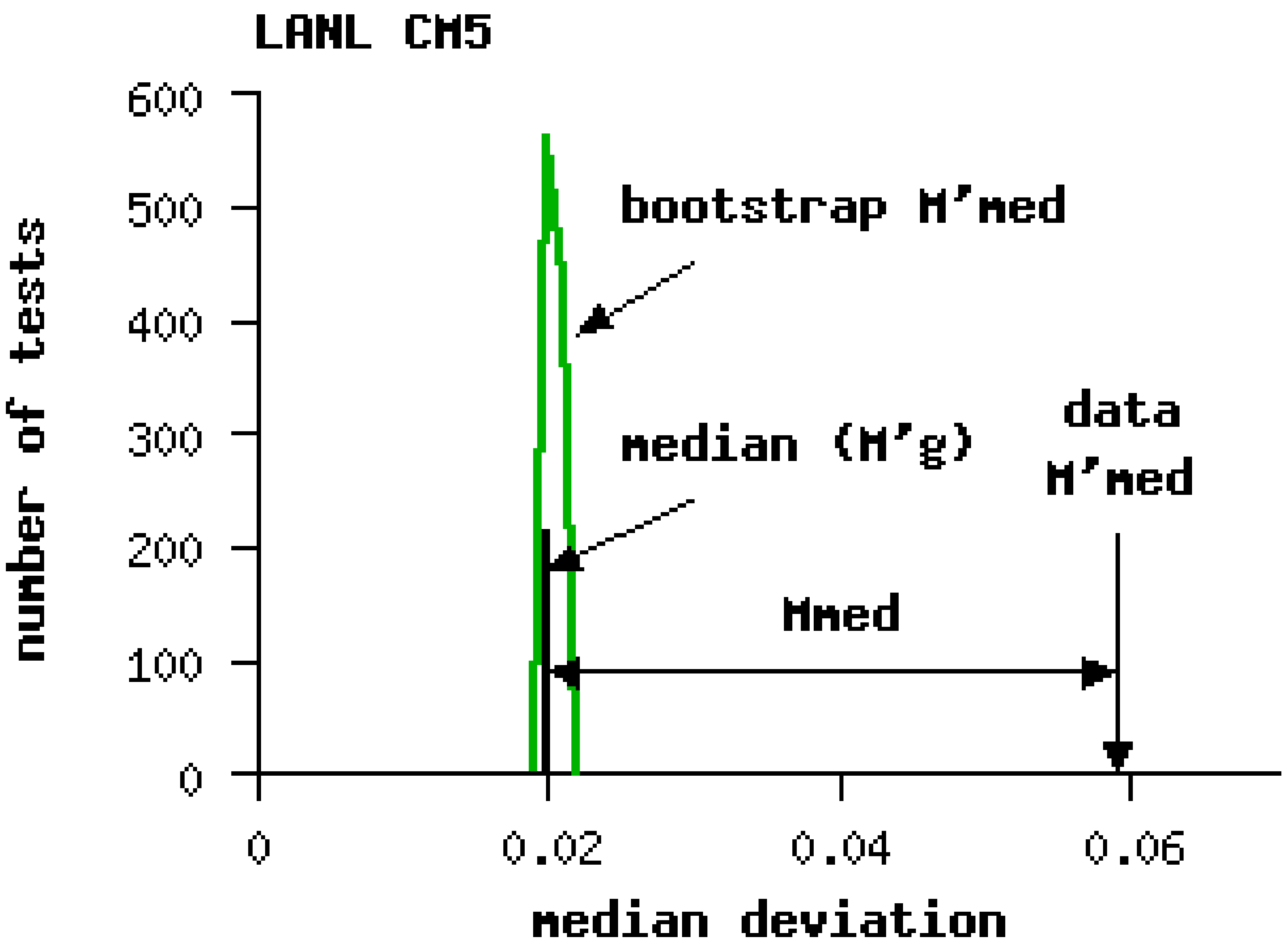
$$m_i = \frac{max_j \{ |o_j - e_i| \}}{N_i - e_i}$$

4. Find median of all the m_i

The idea: quantify concentration of values in one range of the global distribution



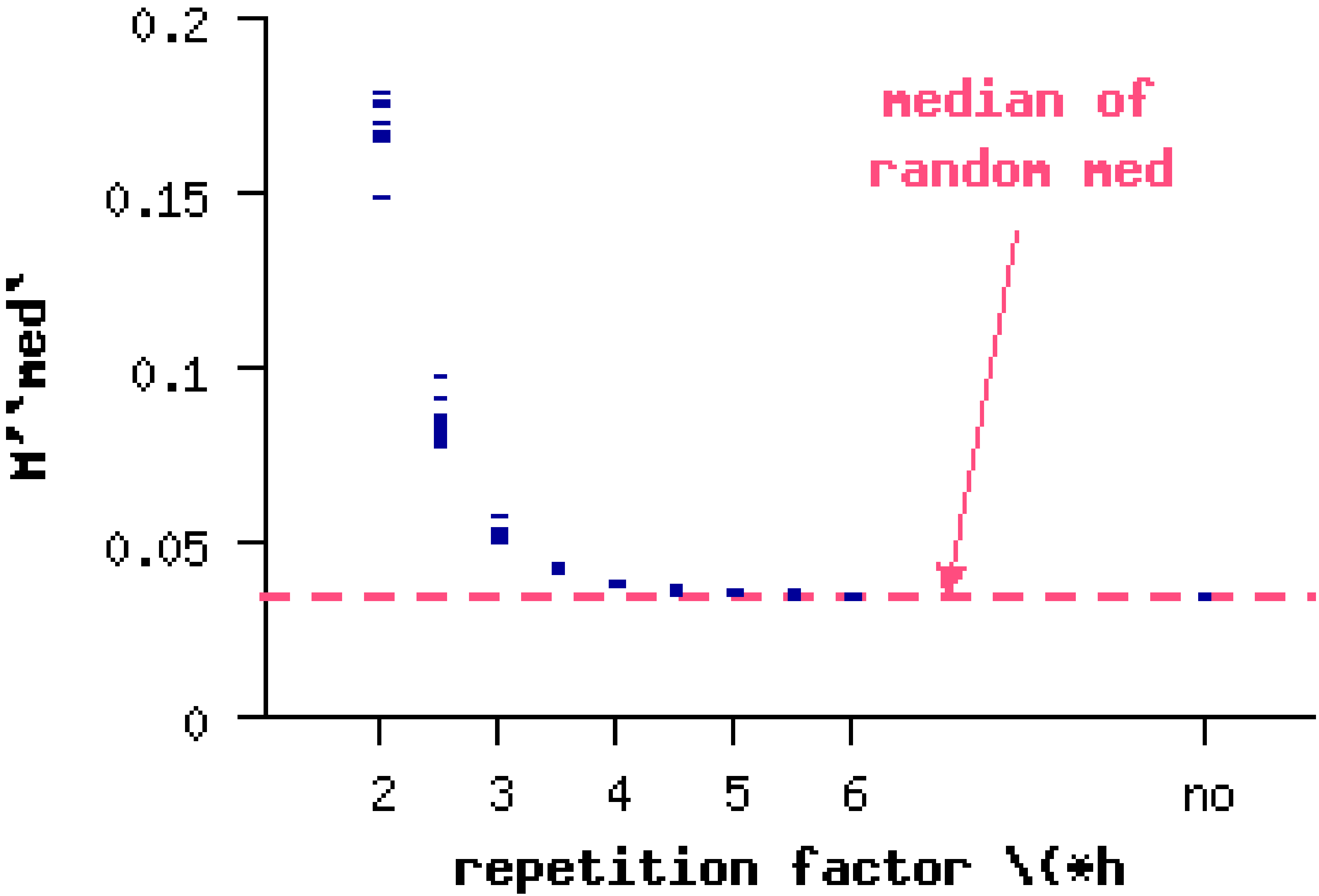
Example results



Significance of results

Modeling locality of sampling:

- Empirical data: job repetitions are heavy tailed
- Top level of model: choose a job
- Bottom level: repeat it according to Zipf distribution
- Tail parameter of distribution allows control over the level of locality


0.2
0.15
0.1
0.05
0
2
3
4
5
6
no
repetition factor \(*h
median of
random med