Cardinality Estimation: An Experimental Survey

Hazar Harmouch and Felix Naumann

Information System Group

Hasso Plattner Institut

University of Potsdam

Germany

VLDB 2018
Estimation and Approximation Session
Rio de Janeiro-Brazil
29th August 2018

Finding Number of Distinct Values Problem

A Polyonymous Problem

How many distinct voice actors are there in our series?

- Statistics: number of species in a population.
- DB: „COUNT DISTINCT"
- Streaming: The zeroth-frequency moment of a multiset [Alon96]

Why cardinality is an important statistic?

How Many Distinct …

… queries did I get?

…pairs (sourceIP,destinationIP) have I seen?

…distinct messages have I seen?

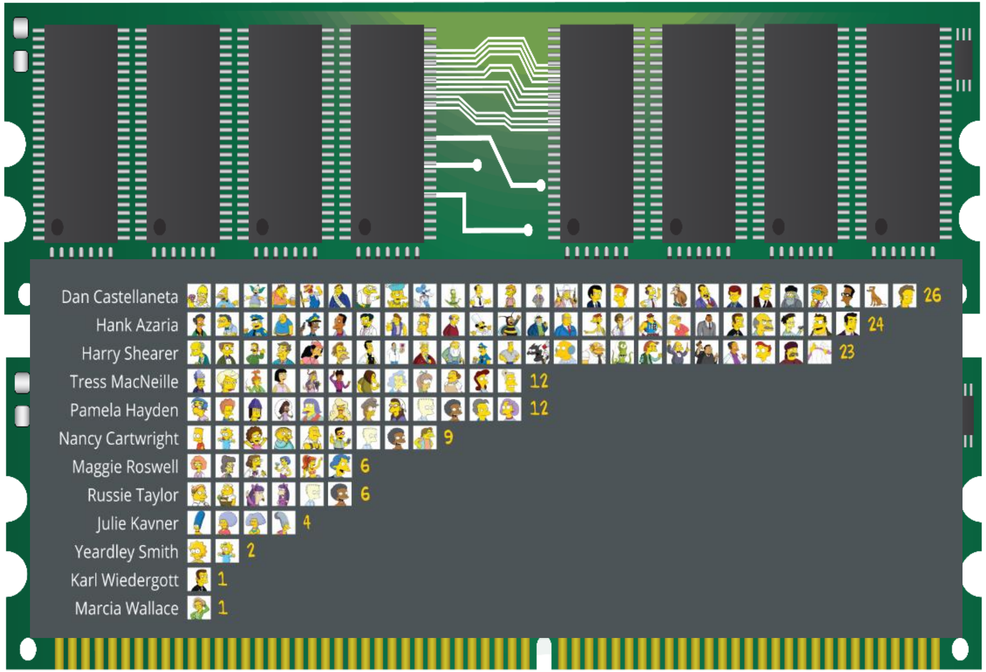
…values have I seen for this attribute x?

… connections have been established from same source?

… visitors to this website in order to advertise in it?

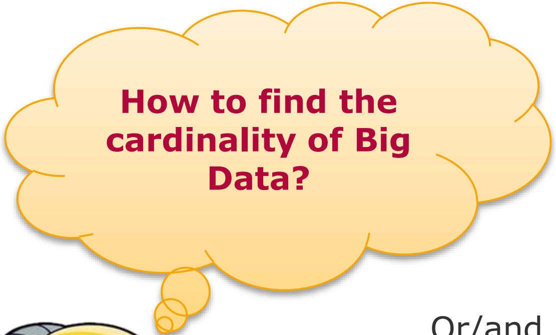
Is exact cardinality sufficient?

Big Data: Exact Counting is Not Easy!

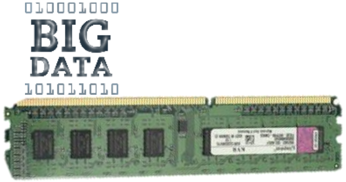


Exact cardinality of multiset determined with storage proportional to dataset size

Is exact cardinality sufficient?
Big Data-Scale! Estimate!

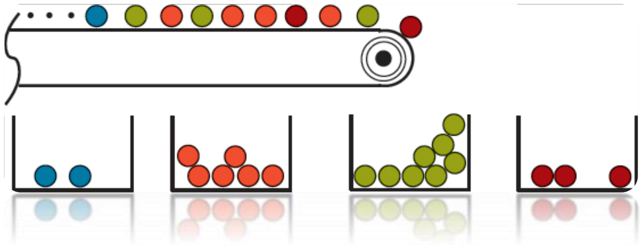


Or/and



- Scale-up the computation
  - Expensive (hardware, equipment, energy).
  - Not always fast.
- Scale-down the data
  - Create synopsis: data structure maintained by the estimation algorithm in main memory.
    - *Temporary*: static scenarios.
    - *Compact representation*: streaming applications
  - Need to fit the problem.

Cardinality Estimation Approaches (1-6)
Exact cardinality: Sorting



- Sorting eliminates duplicates.

- **Problem**:
  - Expensive operation.
  - Synopsis size is at least as large as the dataset.
  - Impractical for current big datasets.

Cardinality Estimation Approaches (2-6)

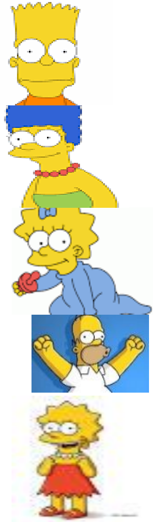
Exact Cardinality: Bitmap

- **Synopsis**: is a bitmap of size equals to universe size and initialized to 0s.
  - Scan dataset once and set the bit i to 1 whenever an item with the i –th value of the universe is observed.
  - Cardinality= Number of 1s.

- **Problem**: The synopsis size is a function of the universe size N, which is potentially much larger than the size of the dataset itself.

! Still used in another approached.

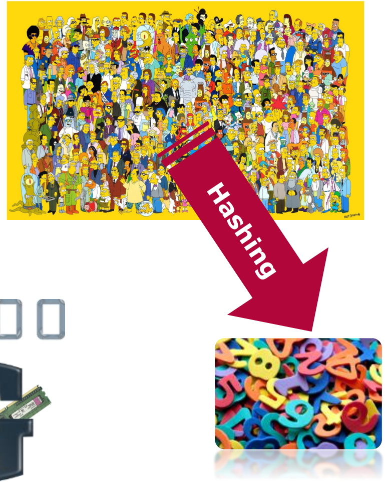
Dan Castellaneta	0
Hank Azaria	0
Harry Shearer	0
Tress MacNeille	0
Pamela Hayden	0
Nancy Cartwright	1
Maggie Roswell	0
Russie Taylor	0
Julie Kavner	0
Yeardley Smith	0
Karl Wiedergott	0
Marcia Wallace	0

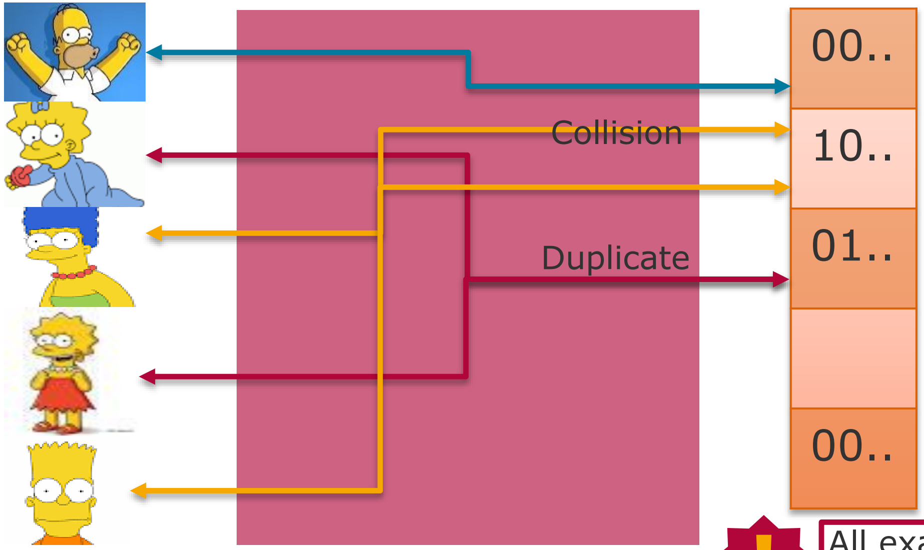


Cardinality Estimation Approaches (3-6)

Exact Cardinality: Hashing

- Hashing eliminates duplicates without sorting, scale-down synopsis size and requires one pass.
- Simple application of hashing can be worse than sorting in terms of memory consumption.



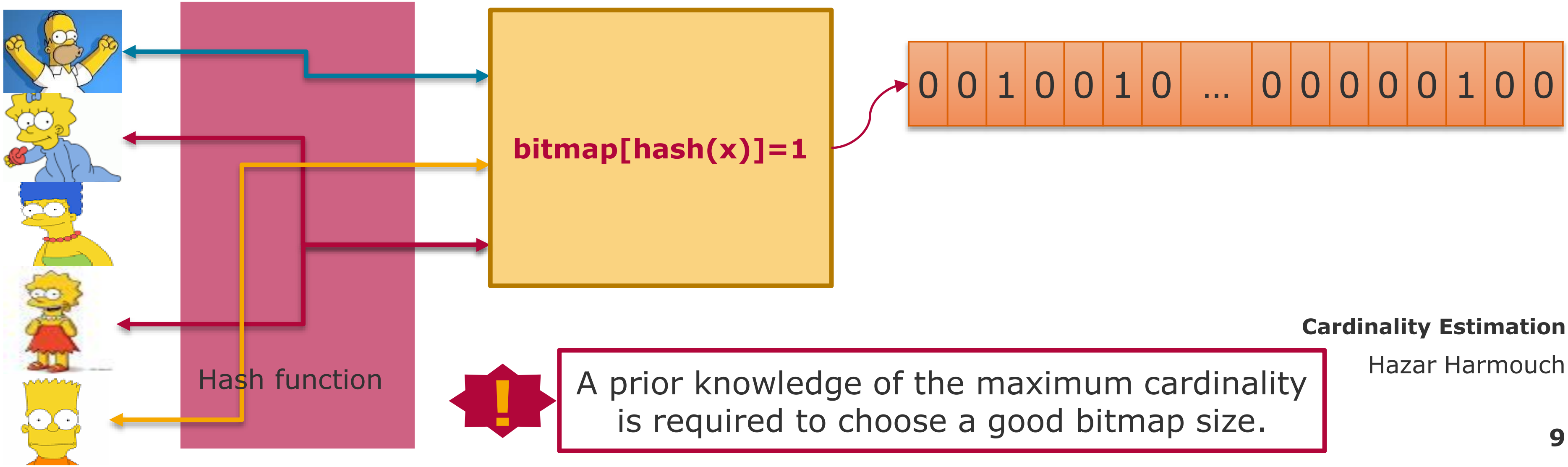


All exact approaches are expensive in both size and runtime.

Cardinality Estimation Approaches (4-6)

Estimation: Bitmap of hash values

- Scales down the synopsis size by don’t store the hash values.
- **Synopsis**: a bitmap keeps track of the hashed values.
  - The hash function maps each item to a bit in the bitmap.
  - Like Bloom filters



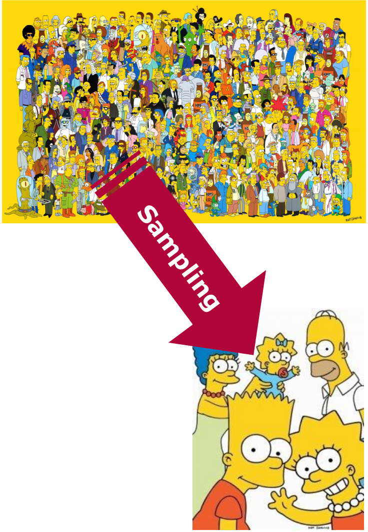
A prior knowledge of the maximum cardinality is required to choose a good bitmap size.

Cardinality Estimation approaches (5-6)

Estimation: Sampling

- Reduces the synopsis size

- Several negative results.
  - For every estimate based on a small-sample, there is a dataset where the ratio error can be made arbitrarily large [Charikar00].
  - Almost all the dataset needs to be sampled to bound the estimation error within a small constant [Haas95, Haas98].

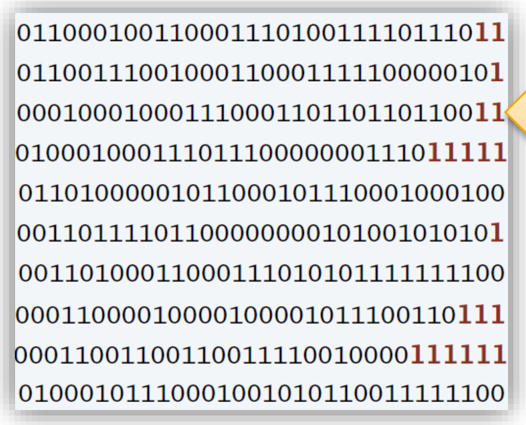


Cardinality Estimation approaches (6-6)

Estimation: Observations in hash values

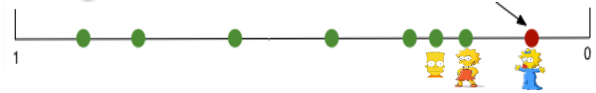
Hash values can be seen as:

- **Bit strings**
- **Range of real numbers**


01100010011000111010011110111011
01100111001000110001111100000101
00010001000111000110110110110011
01000100011101110000000111011111
01101000001011000101110001000100
00110111101100000000101001010101
00110100011000111010101111111100
00011000010000100001011100110111
00011001100110011110010000111111
01000101110001001010110011111100

- Bit pattern observables depends on the occurrence of particular bit patterns at the binary string representation.

Smallest value seen≈ $1/F_0$



- Order statistic observables consider the hash values as real numbers.
- The order statistic of rank k is the k-th smallest value in the dataset.

Cardinality Estimation
Classification of 12 Algorithms

[Durand-Flajolet03] [Metwally08]

Algorithm	Observables	Intuition	Core method
FM	Bit-pattern	Logarithmic hashing	Count trailing 1s
PCSA	Bit-pattern	Logarithmic hashing	Count trailing 1s
AMS	Bit-pattern	Logarithmic hashing	Count leading 0s
BJKST	Order statistics	Bucket-based	Count leading 0s
LogLog	Bit-pattern	Logarithmic hashing	Count leading 0s
SuperLogLog	Bit-pattern	Logarithmic hashing	Count leading 0s
HyperLogLog	Bit-pattern (order statistics)	Logarithmic hashing	Count leading 0s
HyperLogLog++	Bit-pattern	Logarithmic hashing	Count leading 0s
MinCount	Order statistics	Interval-based	k-th minimum value
AKMV	Order statistics	Interval-based	k-th minimum value
LC	No observable	Bucket-based	Linear synopses
BF	No observable	Bucket-based	Linear synopses

Counting trailing 1`s Algorithm (1-2)

Flajolet-Martin (FM) [Flajolet-Martin85]

As I said over the phone, I started working on your algorithm when Kyu-Young Whang considered implementing it and wanted explanations/estimations. I find it simple, eleg and amazingly powerful.

Without analysis (original algorithm)

```
After all the values have been processed, then
if M(MAP)=000,  then RESULT=LO(MAP)-1
if M(MAP)=111,  then RESULT=LO(MAP)+1
otherwise RESULT=LO(MAP).

For example,
    if MAP was 0000000000000000000000000001111111
    LO(MAP) is 8 and M(MAP) is 000: RESULT=7
    if MAP was 0000000000000000000000011101111111
    LO(MAP) is 8 and M(MAP) is 111: RESULT=9
    if MAP was 0000000000000000000000001001111111
    LO(MAP) is 8 and M(MAP) is 010: RESULT=8
```

With analysis (Philippe)

Philippe determines that

$$\mathbb{E}[2^P] \approx \phi n$$

where $\phi \approx 0.77351\ldots$ is defined by

$$\phi = \frac{e^\gamma \sqrt{2}}{3} \prod_{p=1}^{\infty} \left[\frac{(4p+1)(4p+2)}{(4p)(4p+3)}\right]^{(-1)^{\nu(p)}}$$

such that we can apply a simple correction and have unbiased estimator,

$$Z := \frac{1}{\phi} 2^P \qquad \mathbb{E}[Z] = n$$

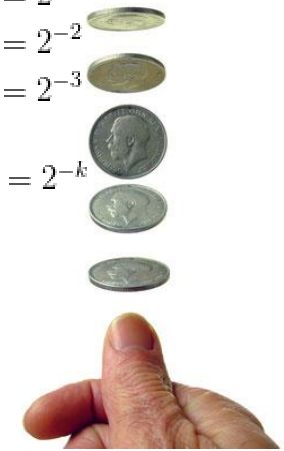
Photos:https://speakerdeck.com/timonk/philippe-flajolets-contribution-to-streaming-algorithms

Counting trailing 1`s Algorithm (2-2)

Flajolet Martin (FM) [Flajolet-Martin85]

- Intuition:
  - Seeing $\rho = k$ means there are at least 2^{k+1} different bit strings.
  - Find the largest ρ and estimate the cardinality by 2^{ρ}.

$$P(\ldots..1) = 2^{-1}$$
$$P(\ldots.10) = 2^{-2}$$
$$P(\ldots100) = 2^{-3}$$
$$\vdots$$
$$P(\ldots10^{k-1}) = 2^{-k}$$



- **Bitmap[rho(hash(x))]=1**
- **rho(y)=position of the LSB=1 in y.**

0	0	0	0	0	0	1	…	0	1	0	1	1	1	1	1
L-1										Z					0

Z:Number of trailing 1s in the bitmap

L: length of the hash bit string (e.g. 32 bit)

Estimate $F_0 = \lfloor 2^Z / 0.77351 \rfloor$

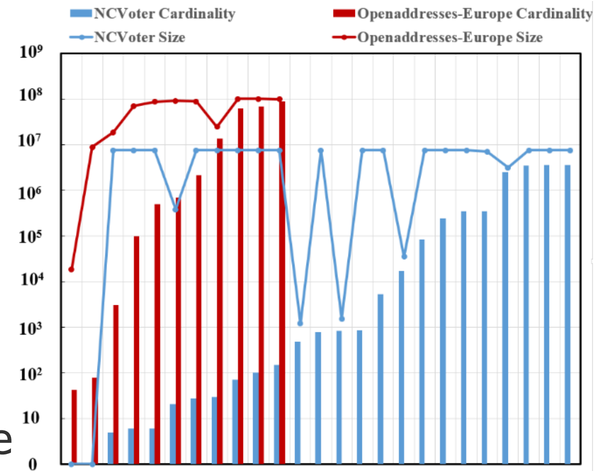
Why comparative experiments is needed?

- Some applications require a very accurate estimation. However, others accept a less accurate estimation.
  - The number of distinct visitors of a website = money.
  - The number of distinct connections ≈ Denial of service.

- Why re-evaluation is good?
  - Is theoretical error analysis matches real-world?
  - What is hidden in the Big-O notation in space bound?
  - Different hash function assumptions
  - Different error metric

Comparative experiments
Experimental setup

- **Implementations** (Unified test environment):
  - Implemented for Metanome
    https://hpi.de//en/naumann/projects/repeatability/data-profiling/cardinality-estimation.html
  - MurmurHash 64-bit. (32-bit for AKMV and MinCount)
  - All algorithms were configured to produce theoretical (standard/relative) errors of 1%.

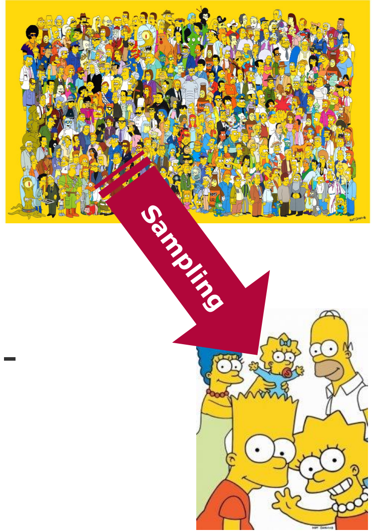


- **Datasets**: 90 synthetic datasets. The exact cardinalitie made to be the powers of 10, starting with 10 up to 10^9.

Dataset	# Attributes	# Tuples
NCVoter	25 (of 71)	7,560,886
Openadresses-Europe	11	93,849,474

Comparative experiments
Sampling-based experiments

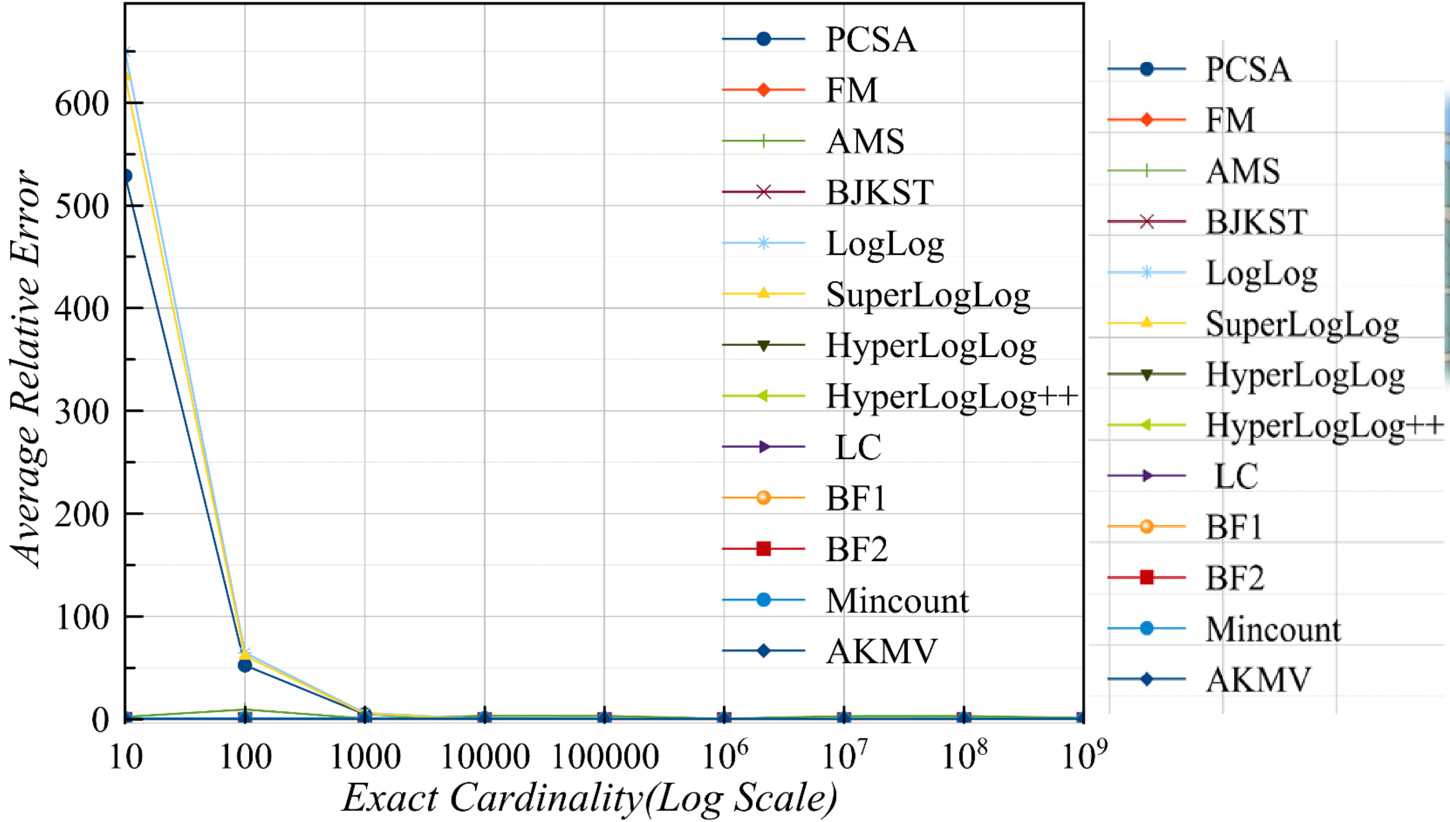
- Guaranteed Error Estimator (GEE) [Charikar00] uses frequency of the values within the sampled data.
- We used Reservoir sampling without replacement.
  - 1% relative error requires sampling more than 90% of the dataset.
  - Minimum heap size of at least 13 GByte and 35 GByte is needed to guarantee an estimation error below 1% on NCVoter and Openadress-Europe, respectively.
  - Runtime noticeably increases with the size of the dataset, but only slightly with the sampling rate.



Dataset	Sampling rate				
	20%	40%	60%	80%	100%
Synthetic	0.54	0.43	0.4	0.2	0
NCVoter	0.26	0.19	0.17	0.07	0.00002
Openadresses	0.28	0.2	0.19	0.09	0.00001

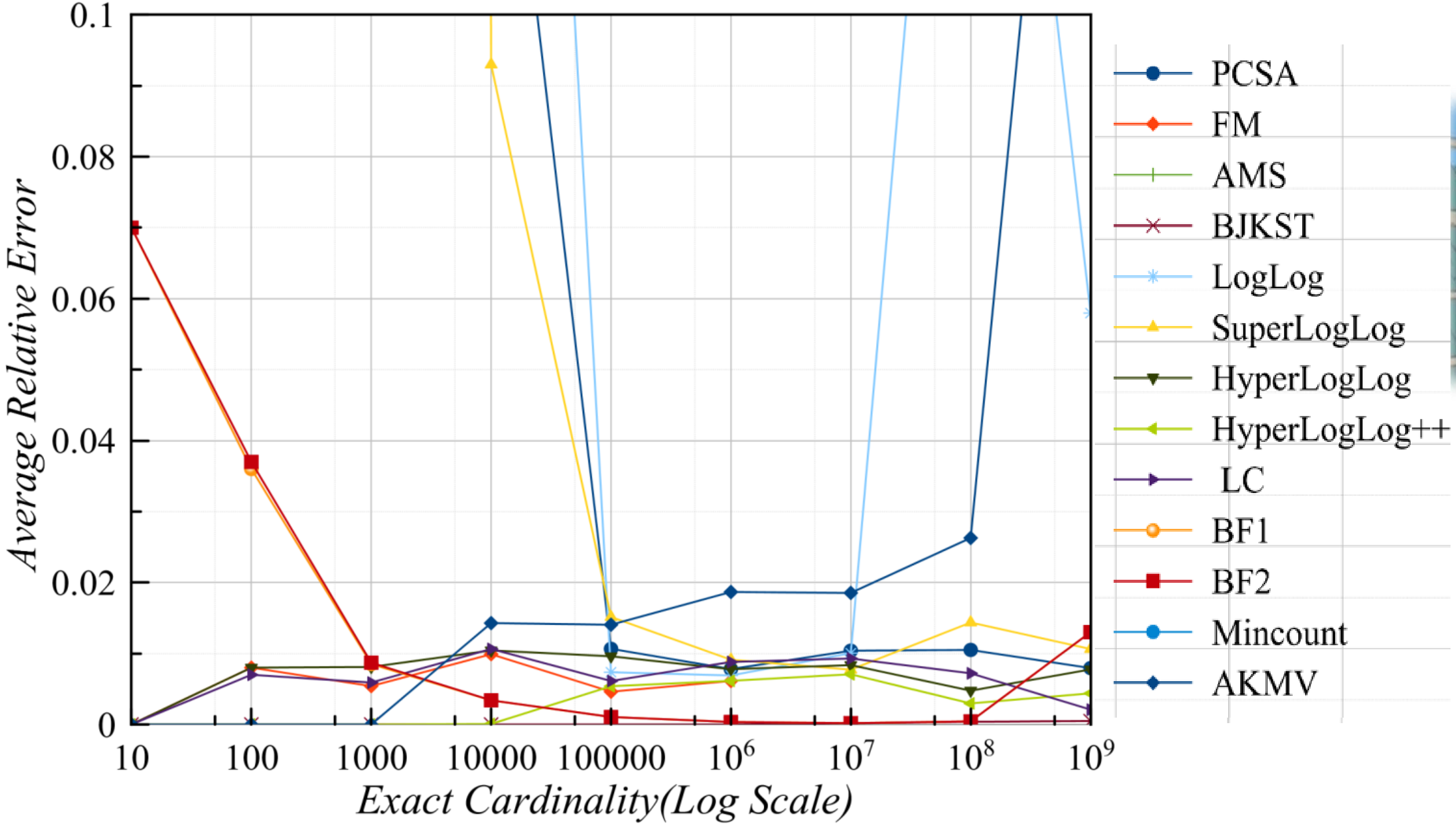
Comparative experiments

Accuracy experiments- synthetic datasets

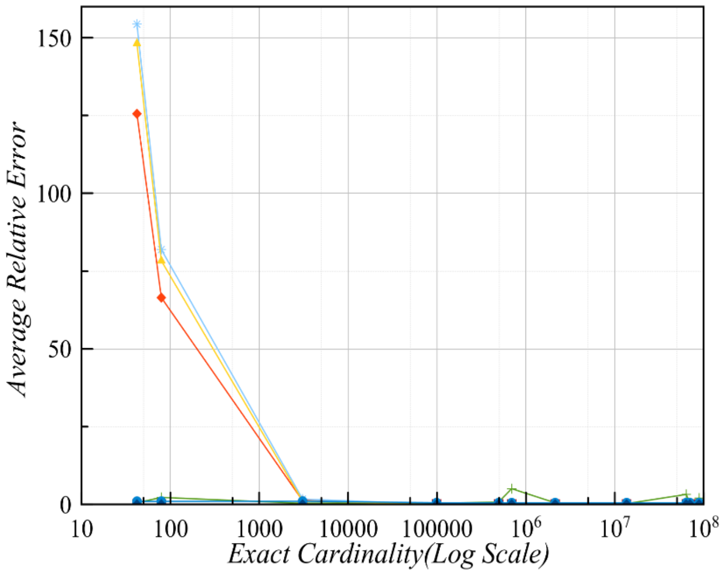


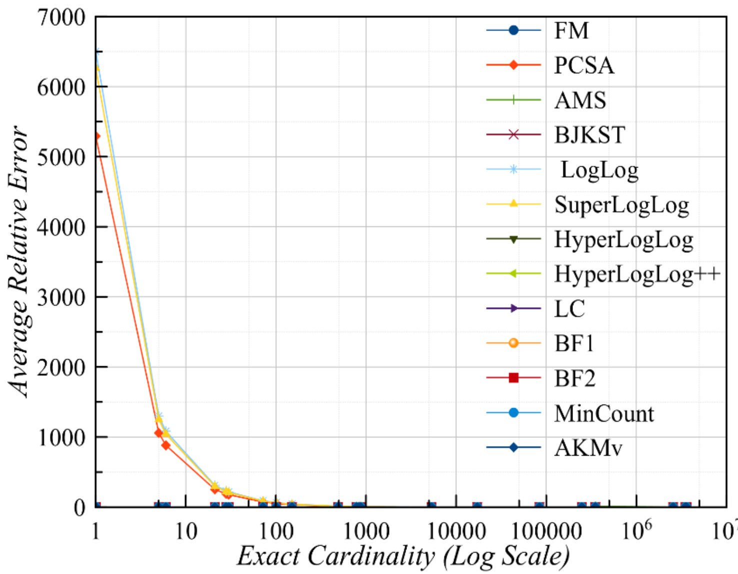
Comparative experiments

Accuracy experiments- synthetic datasets



Comparative experiments
Accuracy experiments-real-world datasets

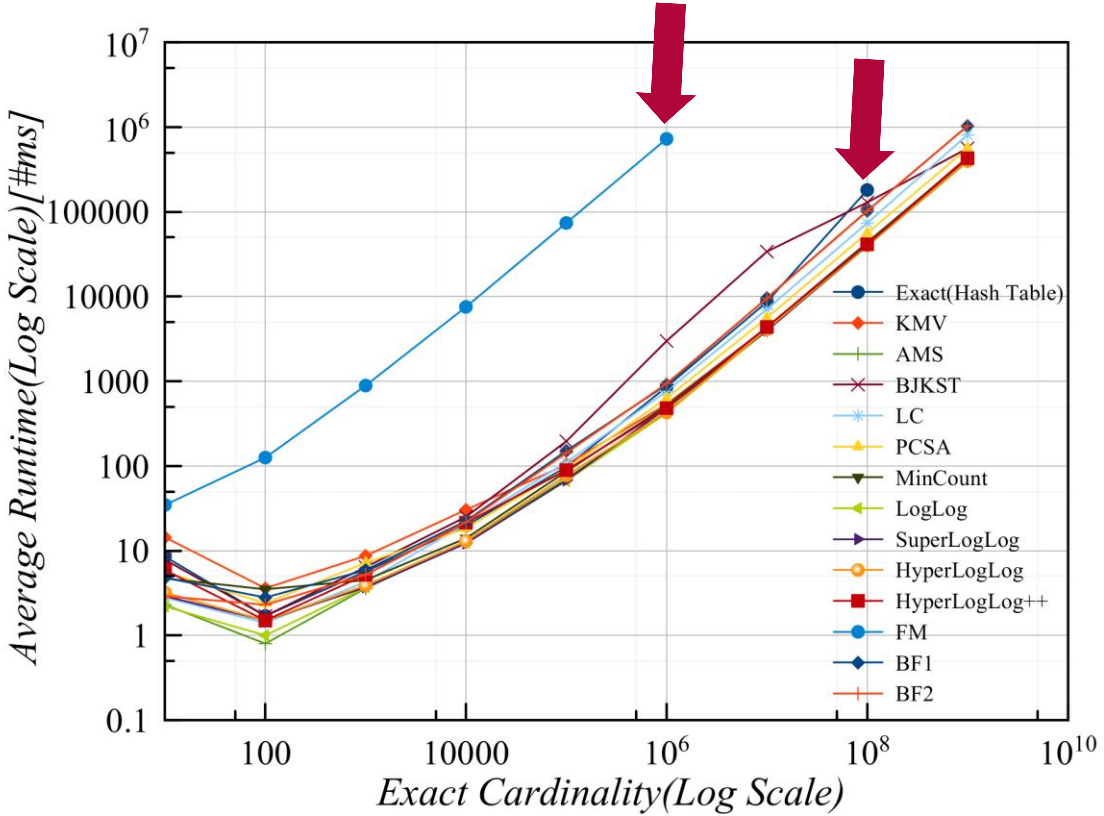




- Openaddress-Europe
- NCVoter

Comparative experiments

Runtime behavior experiments-synthetic datasets

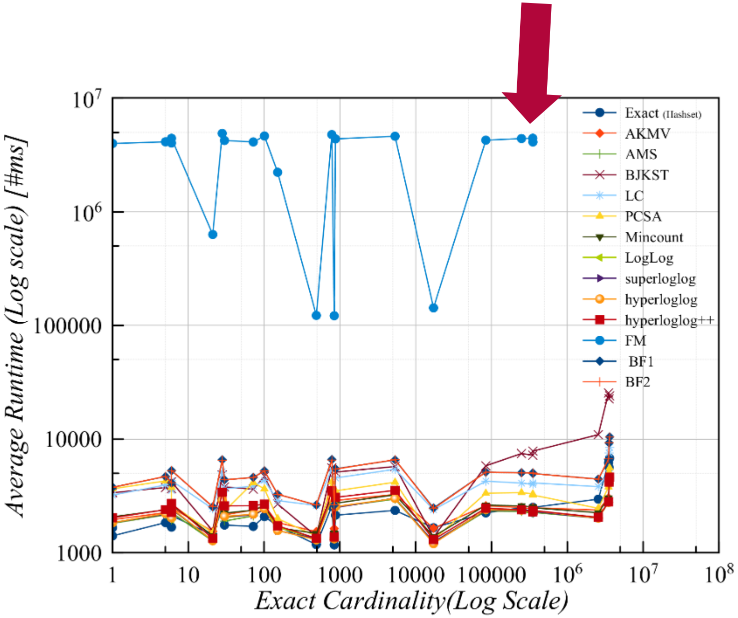


Main factors:

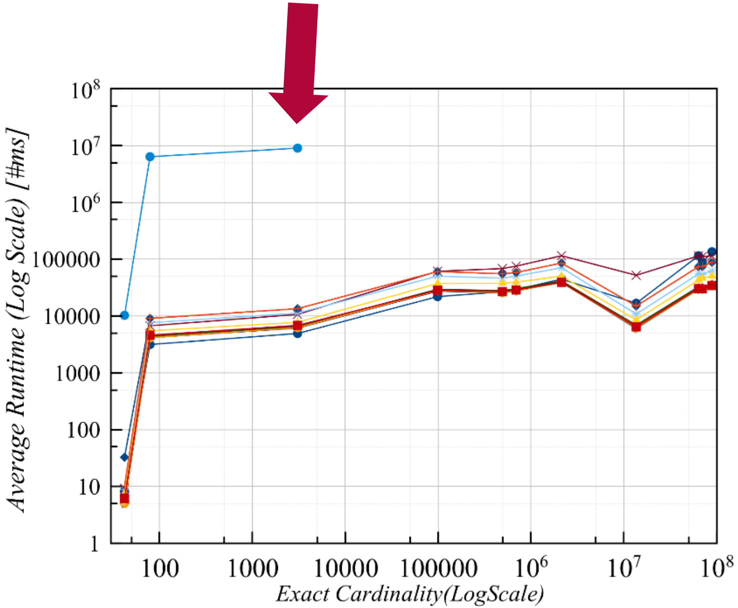
- Dataset size
- Nb. of hash functions
- Synopsis type

Comparative experiments

Runtime behavior experiments-real-world datasets



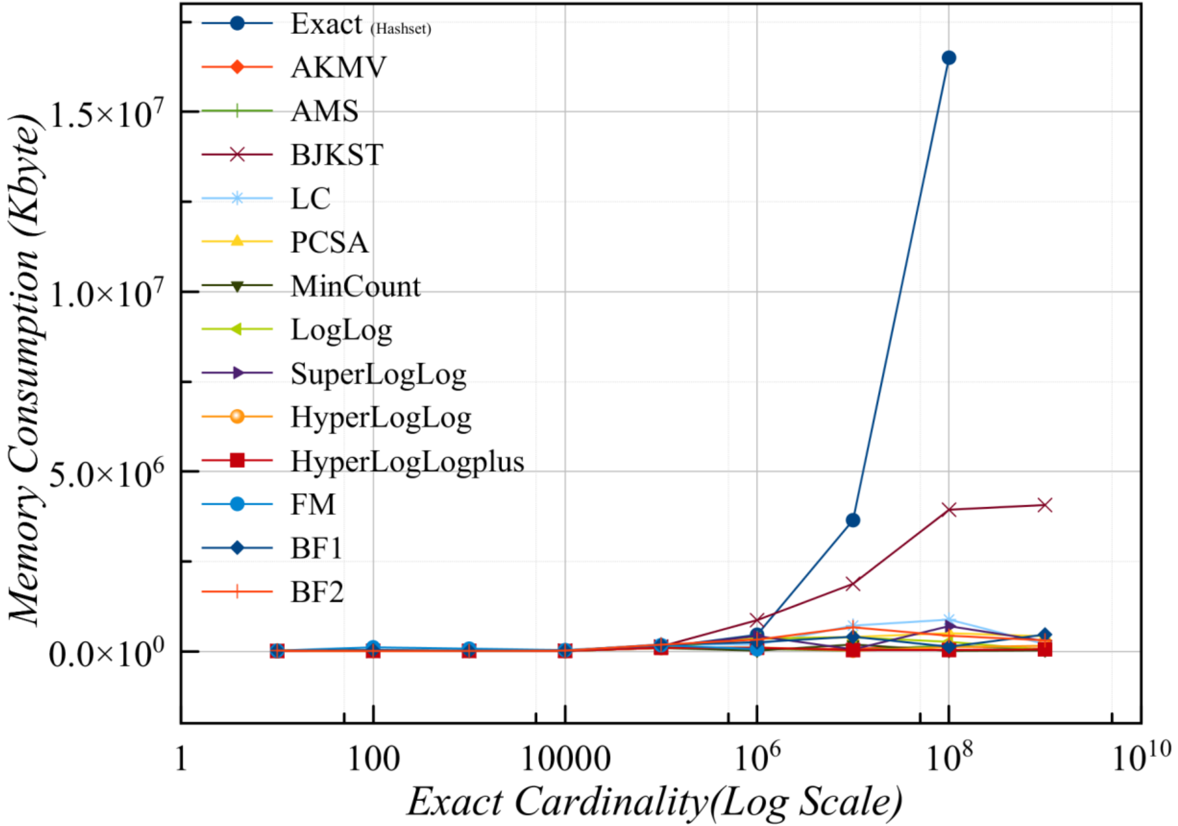
- NCVoter



- Openaddress-Europe

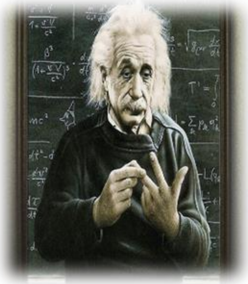
Comparative experiments

Memory Consumption experiments-synthetic datasets



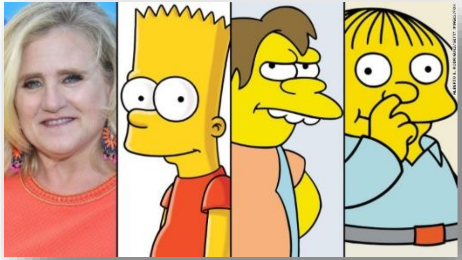
Summary
Counting like Einstein

- Cardinality estimation is a widely studied problem
- Some preliminary solutions, such as sampling and hash tables, are valid only when one can scale up the available computational resources
- For a given accuracy, dataset size is obviously the main factor, affecting all the algorithms' runtime and memory consumption.
- FM: extremely high runtime
- BJKST and Bloom filter have a high memory consumption.
- PCSA, LogLog, SuperLogLog: overestimation problem for datasets with expected small cardinalities.
- HyperLogLog, AKMV, and LC are efficient over all cardinality ranges by all means.



How many distinct voice actors?

The Answer



https://en.wikipedia.org/wiki/List_of_The_Simpsons_cast_members
http://edition.cnn.com/2016/10/16/entertainment/simpsons-600th-episode/index.html

References

- [Charikar00] M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards Estimation Error Guarantees for Distinct Values. In Proceedings of the 19th ACM PODS Symposium on Principles of Database Systems, pages 268–279, 2000.
- [Haas98] P. Haas and L. Stokes. Estimating the Number of Classes in a Finite Population. Journal of the American Statistical Association, 93:1475–1487, 1998.
- [Haas95] Haas, P. J., Naughton, J. F., Seshadri, S., & Stokes, L. (1995, September). Sampling-based estimation of the number of distinct values of an attribute. In *VLDB* (Vol. 95, pp. 311-322).
- [Bar02] Bar-Yossef, Ziv, et al. Counting distinct elements in a data stream.Randomization and Approximation Techniques in Computer Science. Springer Berlin Heidelberg, 2002. 1-10.
- [Heule13] Heule, S., Nunkesser, M., & Hall, A. (2013). HyperLogLog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm. In Proceedings of the 16th International Conference on Extending Database Technology.

References

- [Wang90] Whang, K. Y., Vander-Zanden, B. T., & Taylor, H. M. (1990). A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems (TODS)*, *15*(2), 208-229.
- [Flajolet-Martin85] Flajolet, P., & Martin, G. N. (1985). Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, *31*(2), 182-209.
- [Alon96] Alon, N., Matias, Y., & Szegedy, M. (1996, July). The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 20-29). ACM
- [Durand-Flajolet03] Durand, M., & Flajolet, P. (2003). Loglog counting of large cardinalities. In*Algorithms-ESA 2003* (pp. 605-617). Springer Berlin Heidelberg.
- [Flajolet08] Flajolet, P., Fusy, É., Gandouet, O., & Meunier, F. (2008). Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. *DMTCS Proceedings*, (1).

References

- [Giroire09] *F. Giroire. Order statistics and estimating cardinalities of massive data sets. Discrete Applied Mathematics, 157(2):406{427, 2009.*
- [Papapetrou10] *O. Papapetrou, W. Siberski, and W. Nejdl. Cardinality estimation and dynamic length adaptation for Bloom filters. Distributed and Parallel Databases, 28(2):119-156, 2010.*
- [Beyer07] *K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla. On synopses for distinct-value estimation under multiset operations. In Proceedings of the International Conference on Management of Data (SIGMOD), pages 199-210, 2007.*
- [Metwally08] *A. Metwally, D. Agrawal, and A. E. Abbadi. Why go logarithmic if we can go linear? Towards effective distinct counting of search traffic. In Proceedings of the International Conference on Extending Database Technology (EDBT), pages 618–629, 2008.*