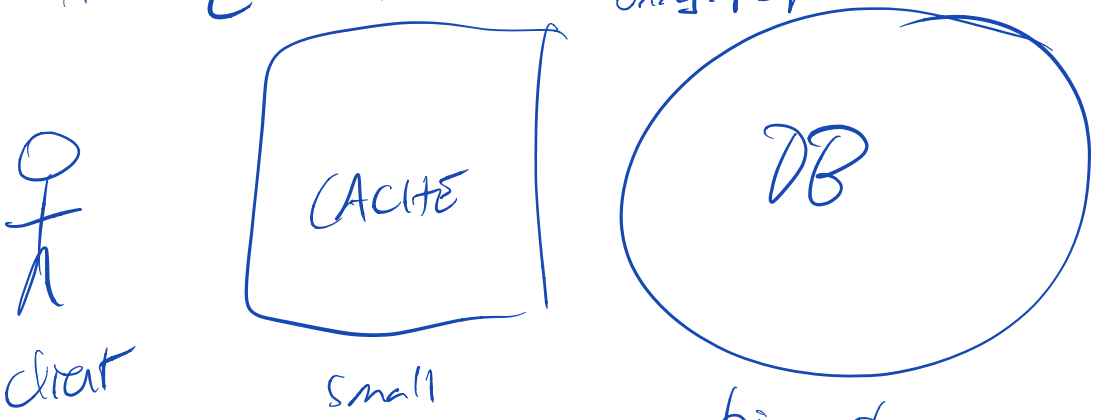
xavim@ac.upc.edu ← new unreg. people — send him a rose



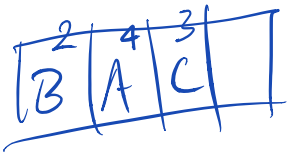
100%

LRU

replacement policies

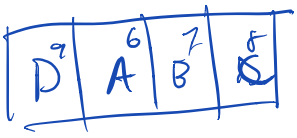
Past behavior is a good predictor of future behavior

full → evict least-recently used



A B C A B C A B C

1 2 3 4 5 6 7 8 9



A B C D E A B C D E
5 6 7 8 9

MRU LFU Least Frequently Used] Frequency

OPT { LRU
recency

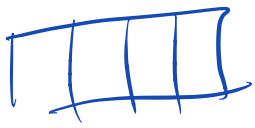
"optimal" worst-case size of cache

k ≤ OPT

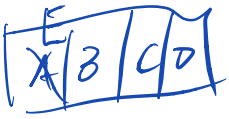
best you can do

A B C D E A B C D E A B C D E

RAND



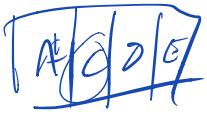
A B C A B C . . .



A B C D E A B C D E A B C D E

$\frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}\ \ldots\ \left(\frac{1}{4}\right)^n$

A A C D A E B C

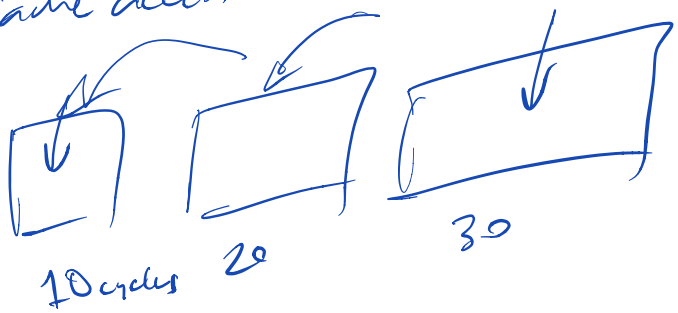


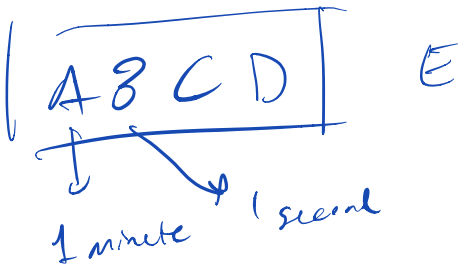
A B C D E F G H

1,000,000

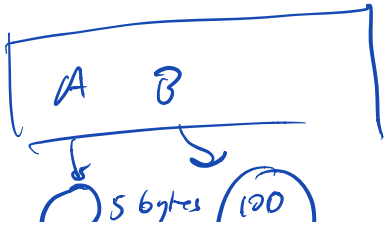
big cache = RAND ~ LRU
in good case
while avoiding
pathological case

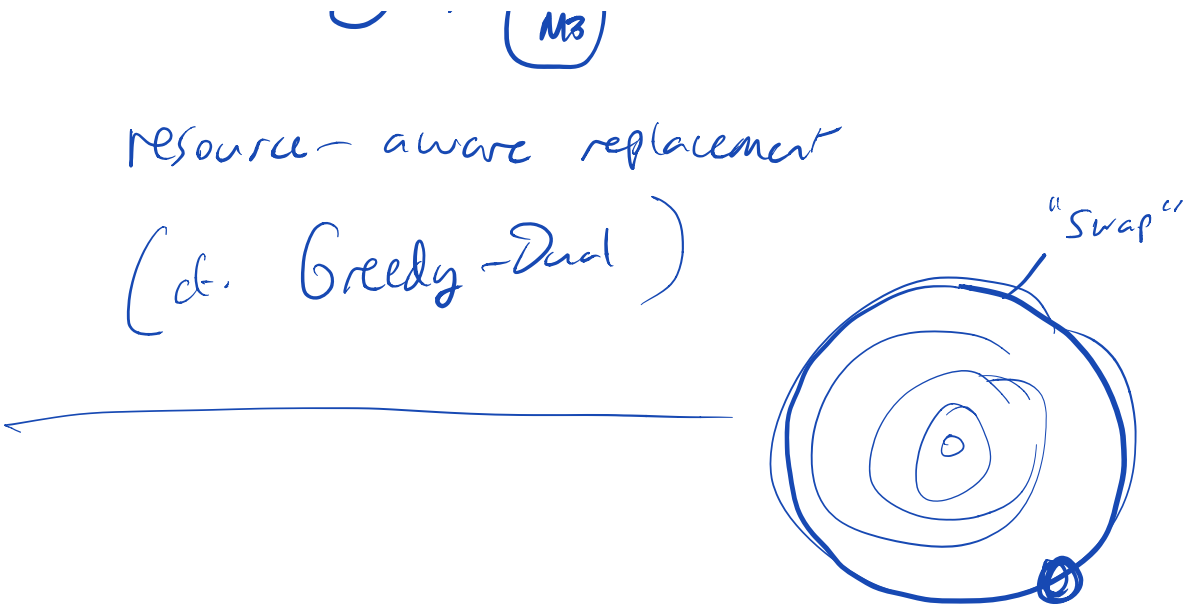
NUCA
non-uniform
cache access

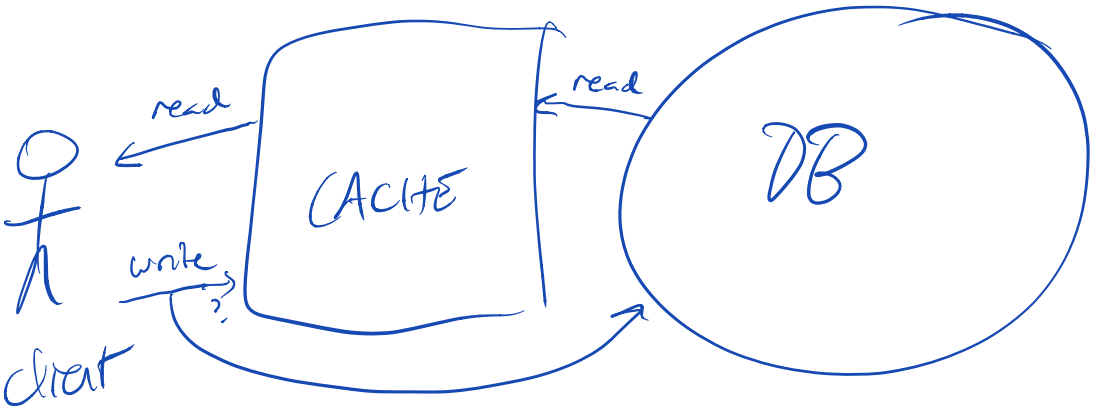




assumes
uniform
cost
& uniform
size

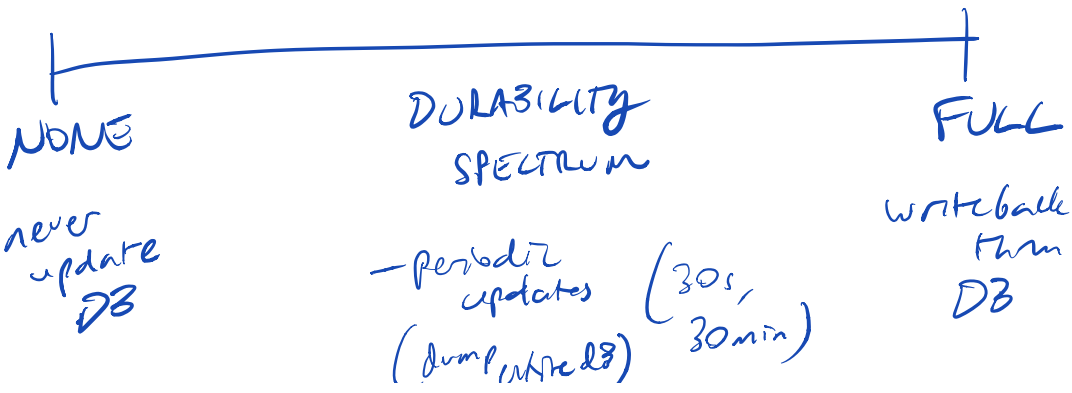


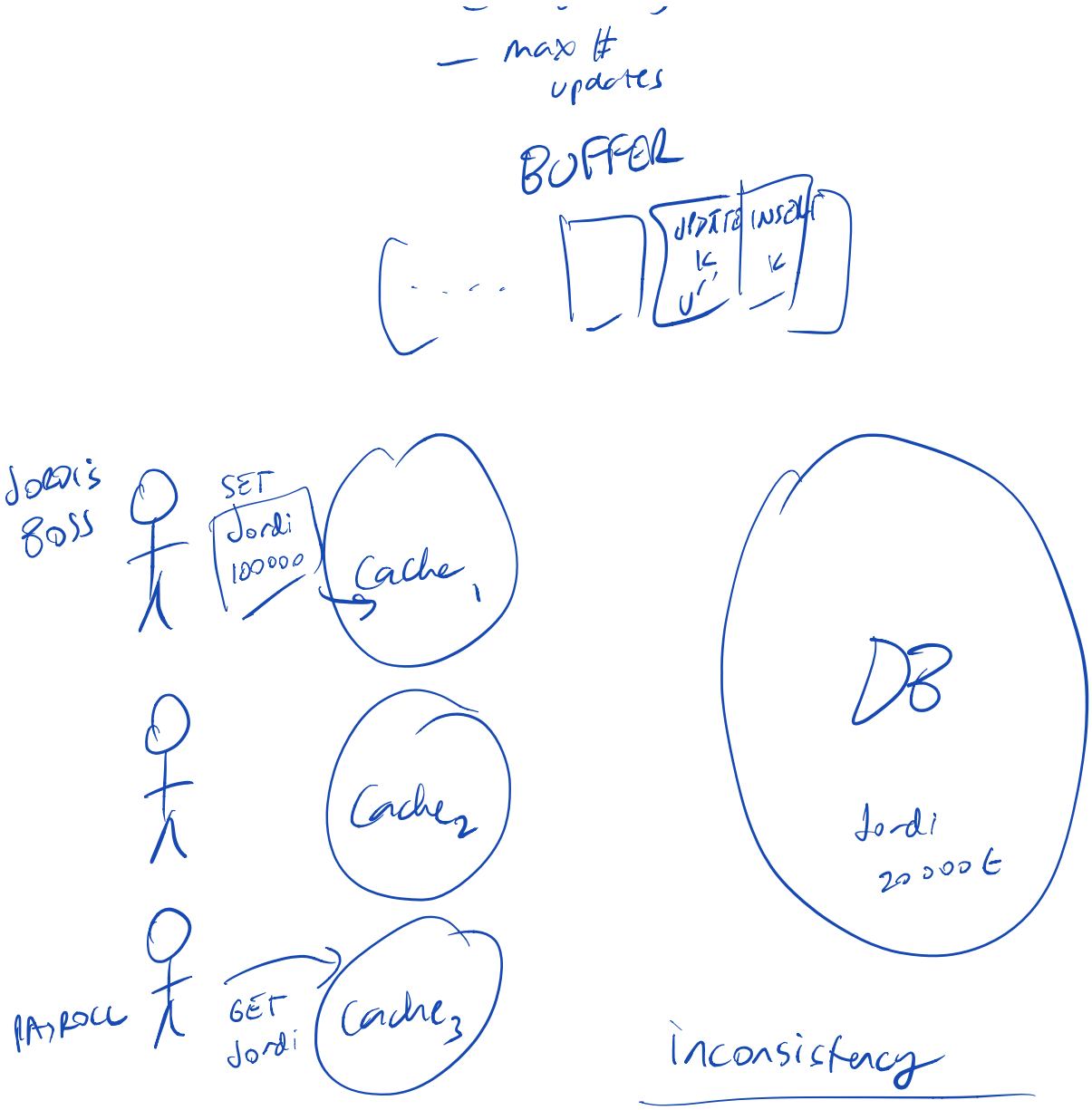




write = { insertion / update

write-through (DB only)
write-back (DB + cache)





Inconsistency

Cache coherence

"MESI"

modified

exclusive (only one copy)

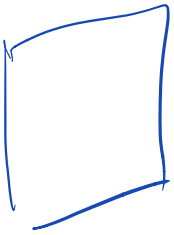
shared (# copies, all read)

invalid

"lease" (30s)

how long something can be kept in cache.

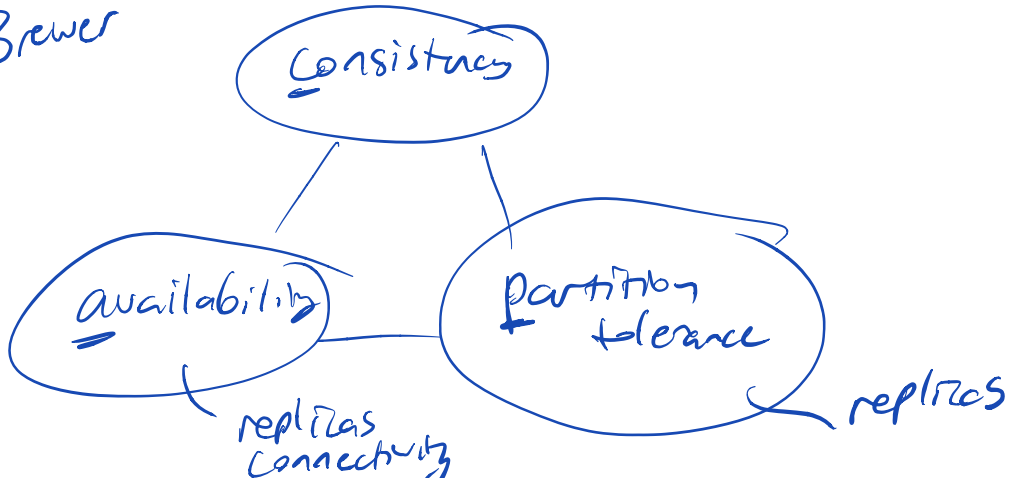
GET(SALARY, "Jordi", 1 week)



"eventual consistency"

"CAP theorem" — pick two

Eric Brewer



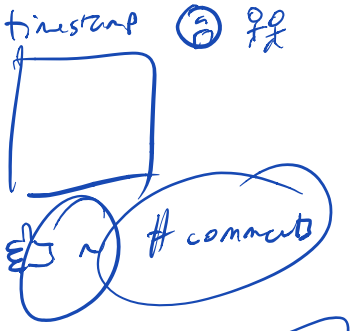
Amazon

Stale
views
of
in stock

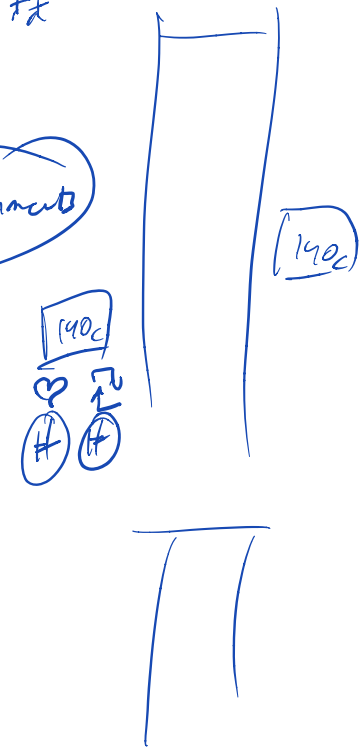
eBay

stale
# of items/
sold/not sold

Facebook

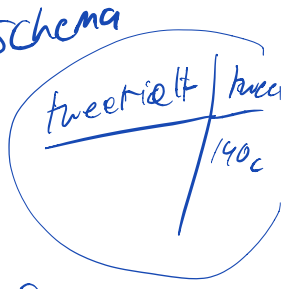


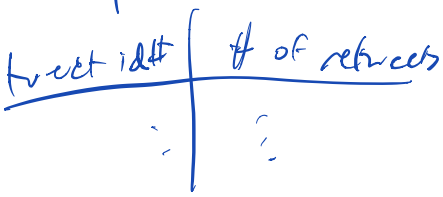
Twitter



NoSQL

- cost (DBs $$$)
- eventual consistency OK
- Flexible schema
  NO schema
  Schema rebuilding??





— low latency (disk — out)

simple queries
low latency needs (~ 3 orders of magn. faster)

— high scalability

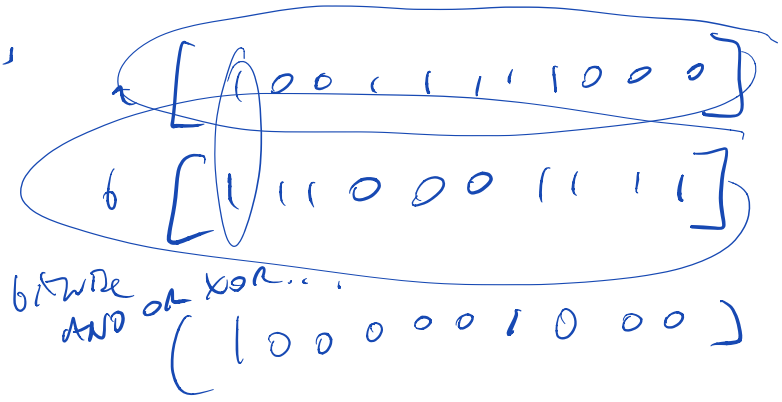
Redis API

in-memory
data structure
store

MongoDB

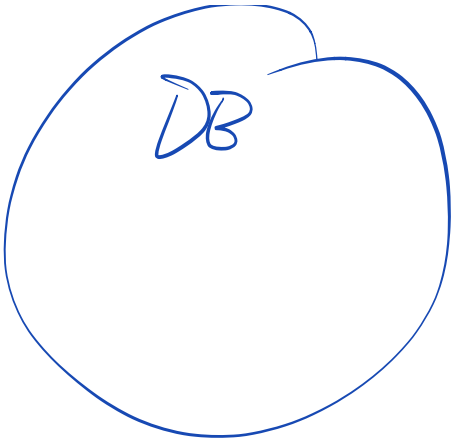
document
store
JSON

dicts (hash tables)
strings
lists
sets
sorted set — range queries
bitmaps



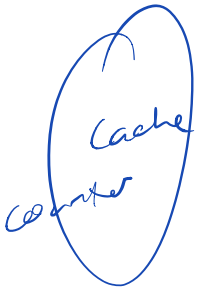
HyperLogLog

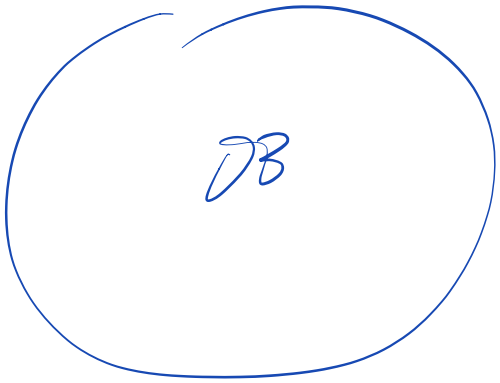
geospatial index (radius queries)



ID	Count-
k	

c = get(k)
set(k, c+1)

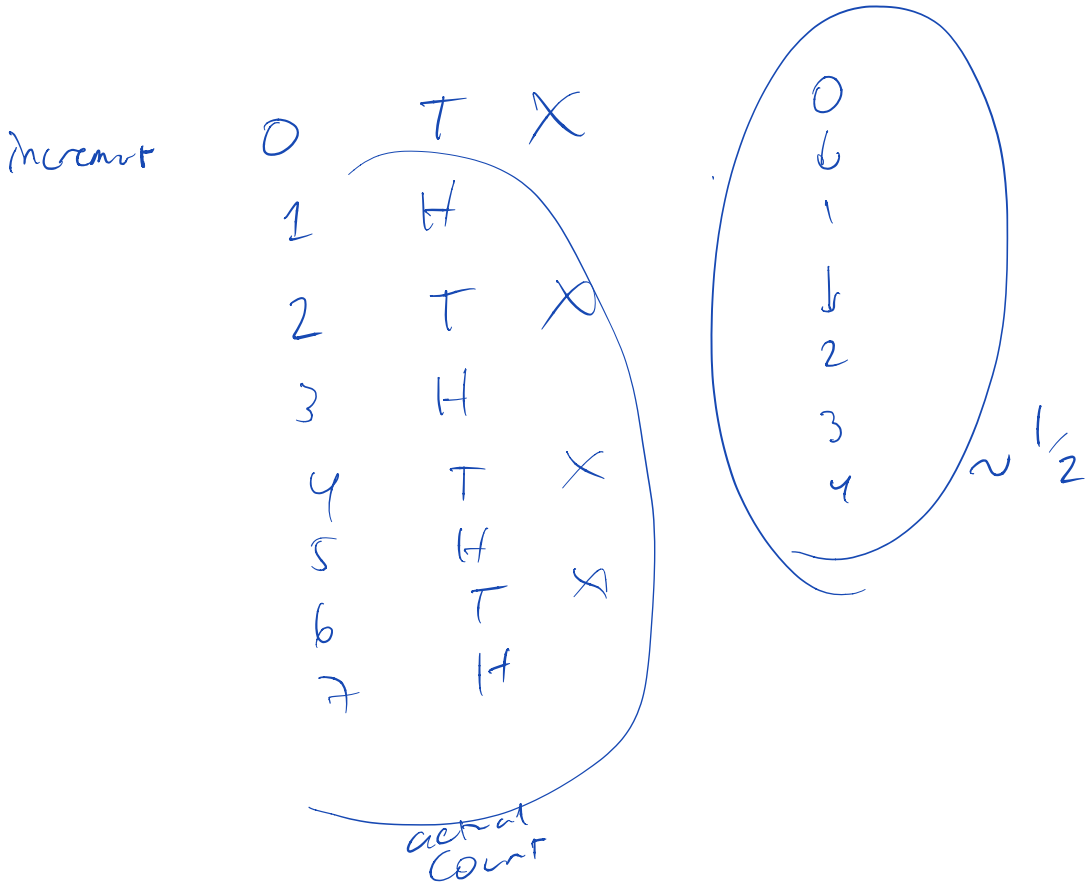




Counter	
1	→ Update
2	→ Update
3	→ Update
4	→ Update
5	→ Update
6	→ Update
⋮	

Robert Morris (97

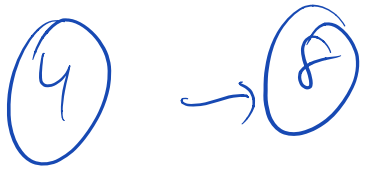
dad of "inventor" of
(Morris Worm - 1988)

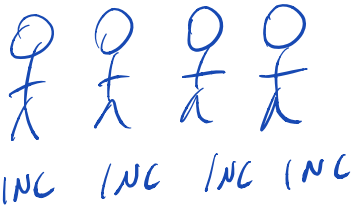


1

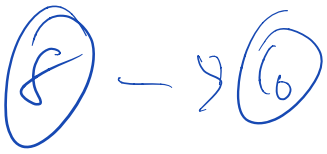
2 1/2

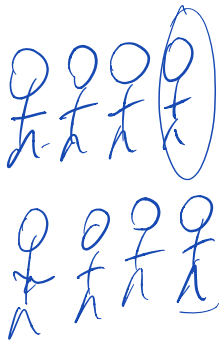
4 1/4



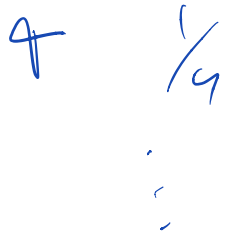


$P(\text{success}_{\text{inc}}) = 1/4$





131072 262144



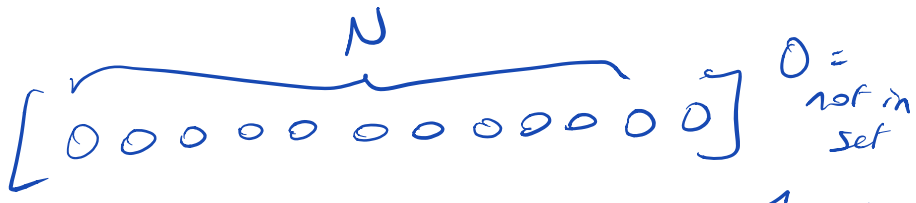
"approximate counting"

probabilistic data structures

trade accuracy for performance

add (item, set)
check (item, set)

} set membership

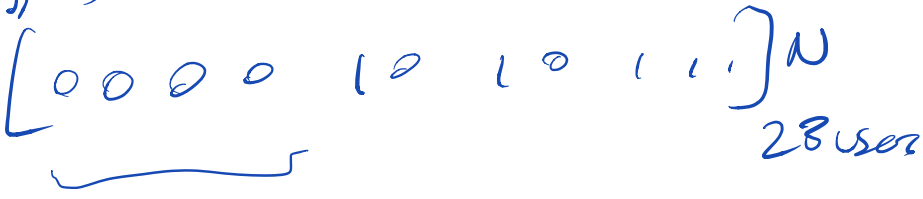


add (k, set)

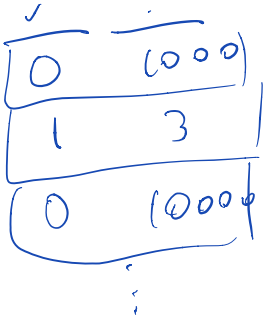
check (k, set)

0 if 1, return TRUE
else FALSE

isEmpty(Emus, x)

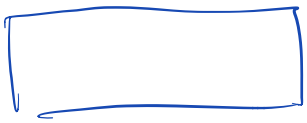


RLE
run-length encoding



"TTFF"

Bloom filter



"FB friend"

little bitmap ✗ exact small ✓

NOT IN SET — exact

IN SET — ? maybe

no friends

≥ 1 friend

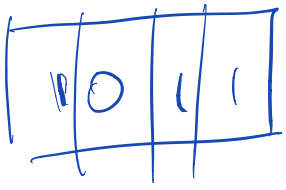
string → hash

"Jordi" → 1010 ?

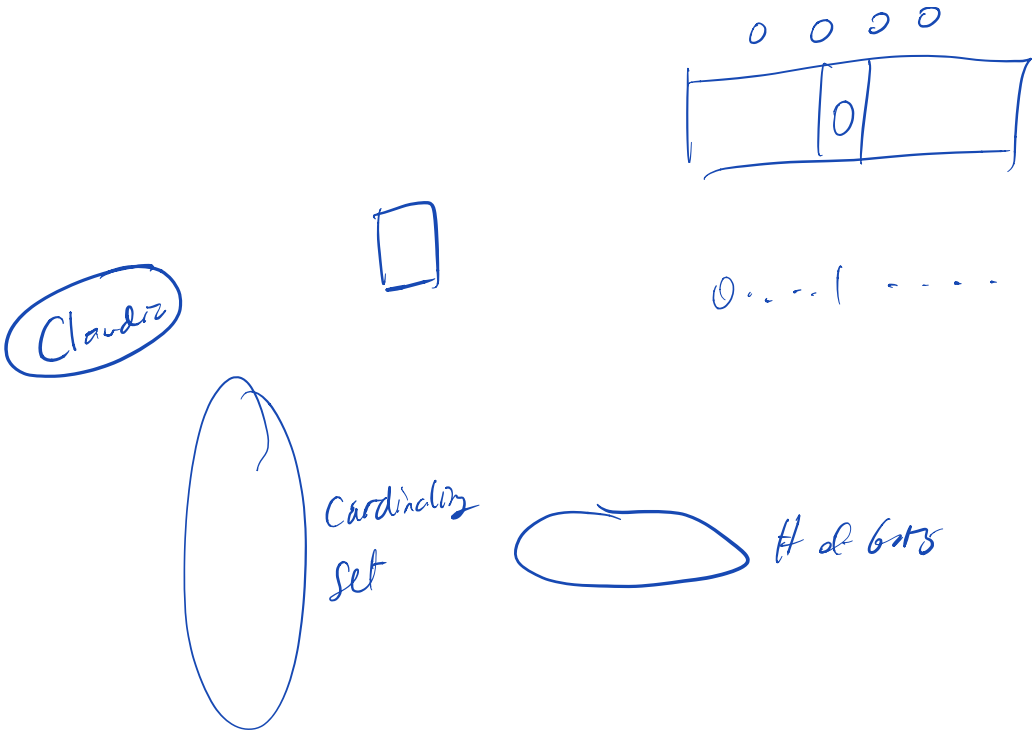
"Paula" → 0011 ?

"Alp" → 0100 ✗

"Alex" → 1011 ?



0000
OR
1010
OR
AND 0100 0011



Cardinality

# of distinct things