Bloom Filter

wisely memorized

ABC Inc.

startup phase
from 0 to 1
no user

Product Manager:

I want our website to have a registration feature that everyone has an unique username.

“a piece of cake”

ABC Inc.

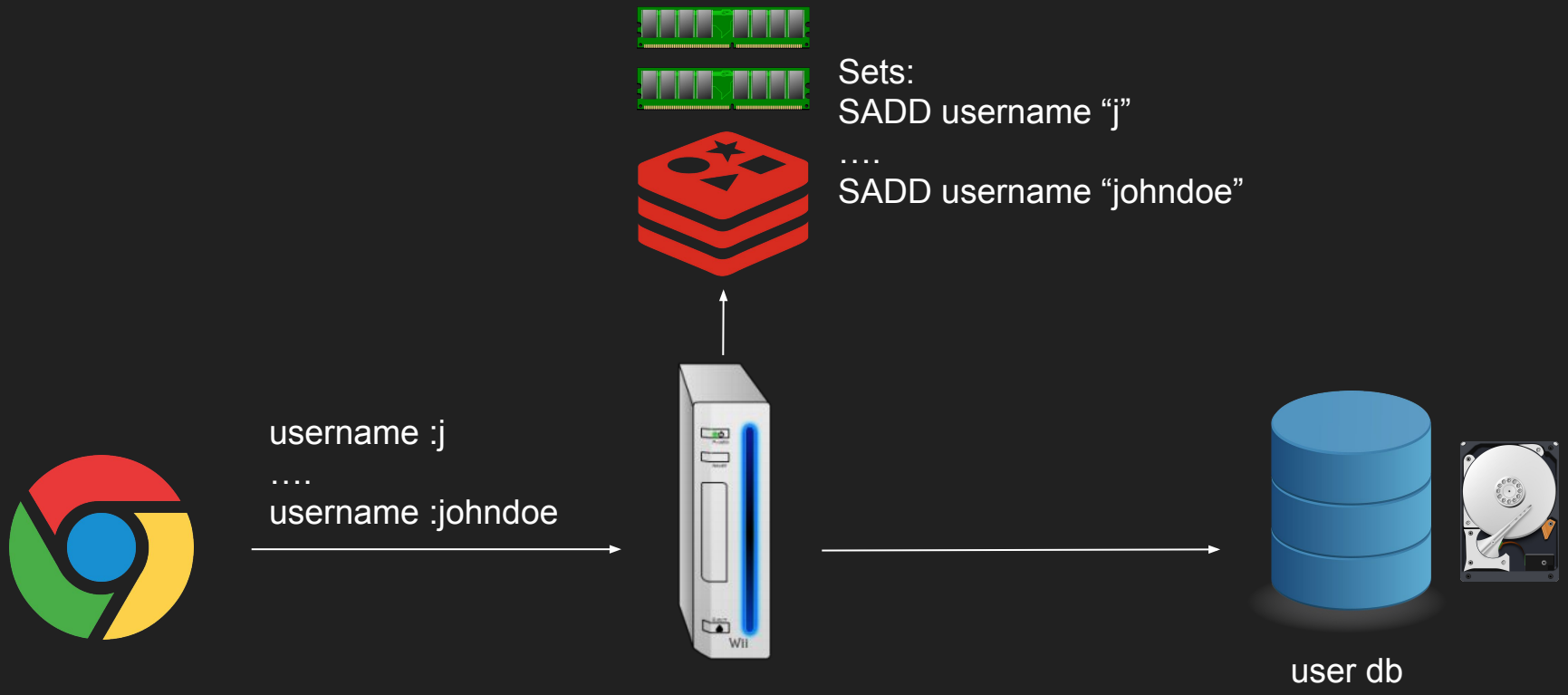
start to gain popularity
more users

Product Manager:

Users report that the registration is a bit slow, but we have very limited budget for scaling up our DB

Set as a middleware?

“no problem, dude”



ABC Inc.

lot of hype
tons of users

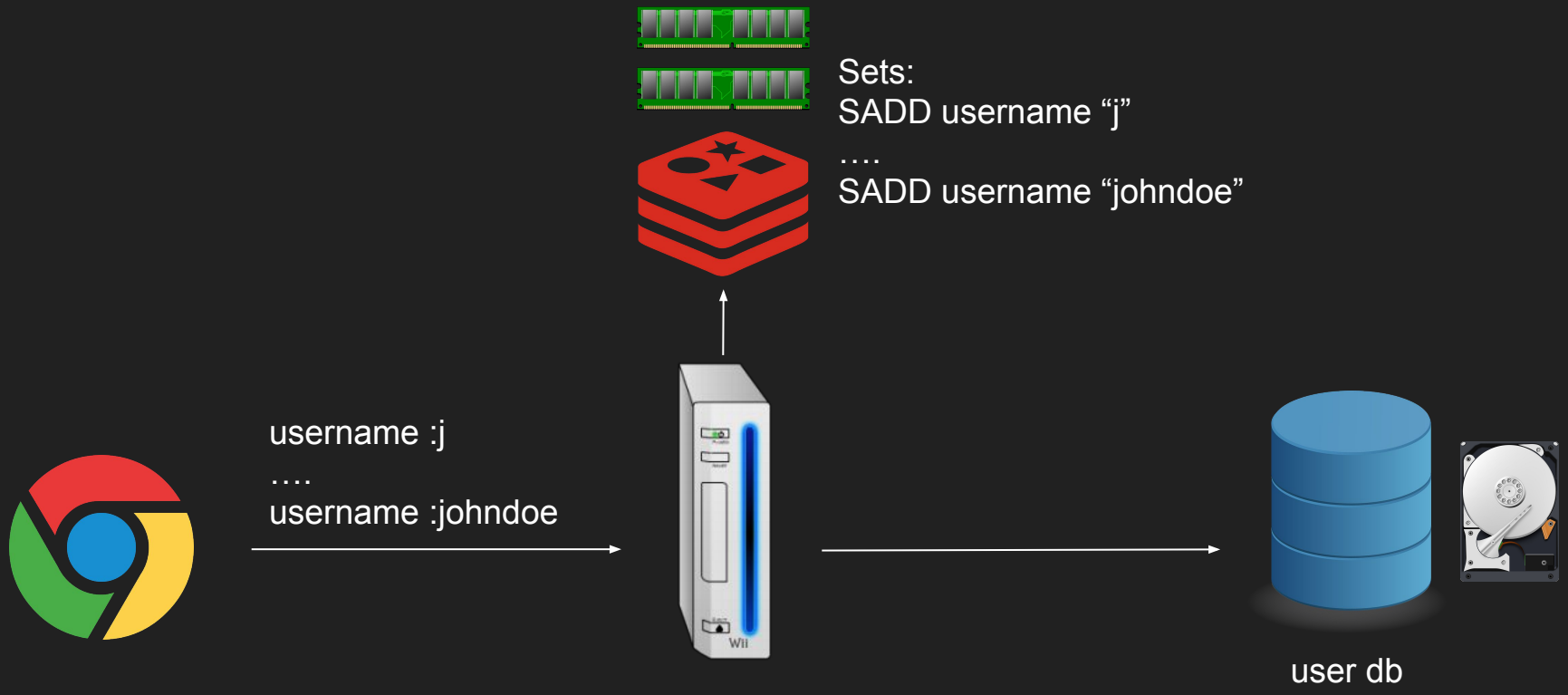
Product Manager:

As username collision happen quite often, we want to increase responsiveness.

By typing in the input box (without clicking the submit button), a user can quickly know if the username has already been taken.

Again, we have very limited budget for this.

It works, but ...


Sets:
SADD username “j”
….
SADD username “johndoe”
username :j
….
username :johndoe
Wii
user db

- require a lot of memory

- in our example,
  33,916,470 records require 1,890,633,554 bytes
  (~1.89 GB)

We want a solution that:

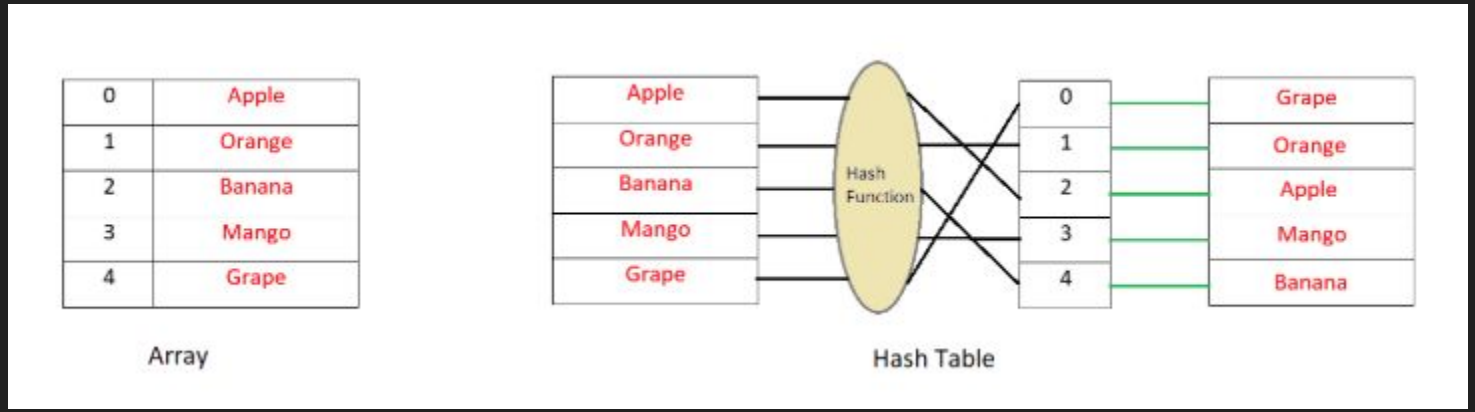
1. significantly reduce the memory consumption
2. while maintaining the speed (or even faster)

Bloom Filter comes to the rescue

Bloom Filter

- Ask : Does “johndoe” exist ?
- Bloom Filter :
  1.) probably in a set, or
  2.) definitely not in a set

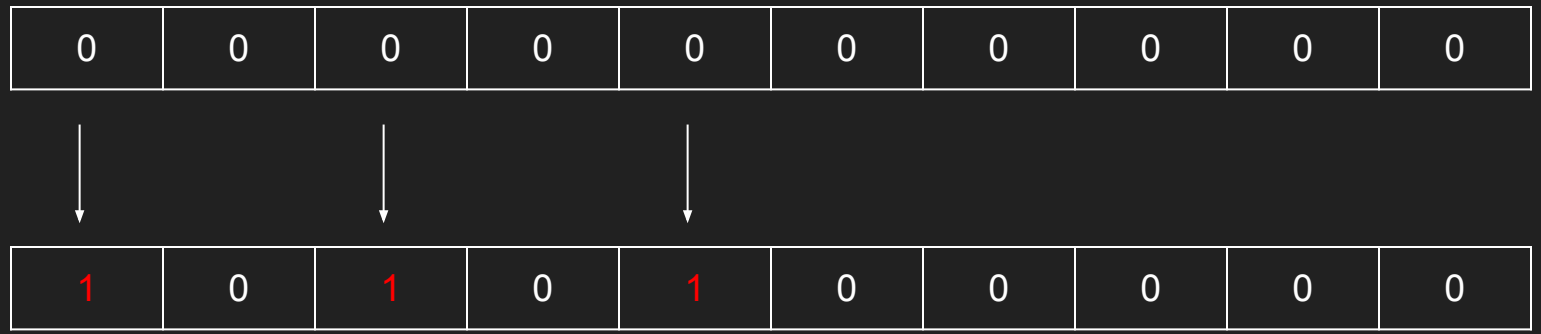
Hash Table


0
Apple
1
Orange
2
Banana
3
Mango
4
Grape
Array
Apple
Orange
Banana
Mango
Grape
Hash Function
0
1
2
3
4
Grape
Orange
Apple
Mango
Banana
Hash Table

Bloom Filter init state

Attempt: “johndoe”

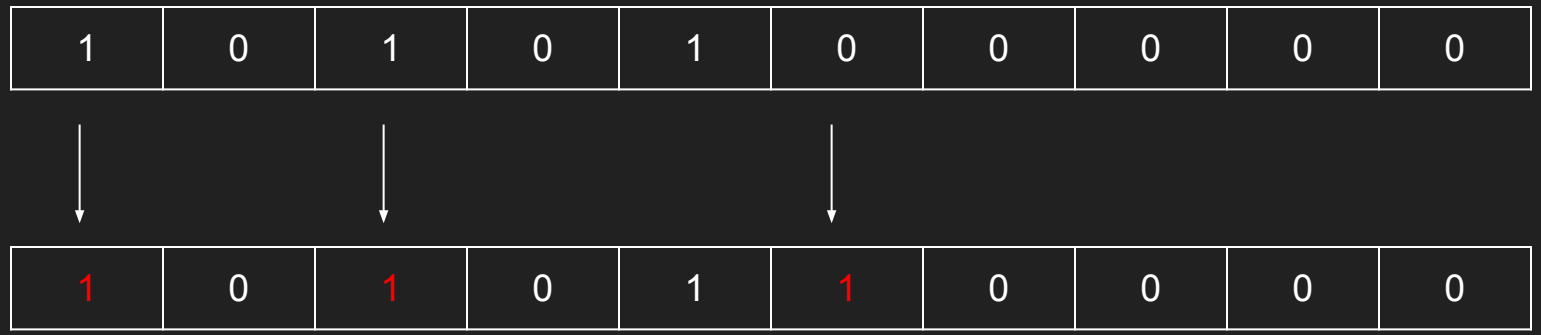
hashFuncA(“johndoe”) = 0
hashFuncB(“johndoe”) = 2
hashFuncC(“johndoe”) = 4



definitely not in the set

Attempt: “franz”

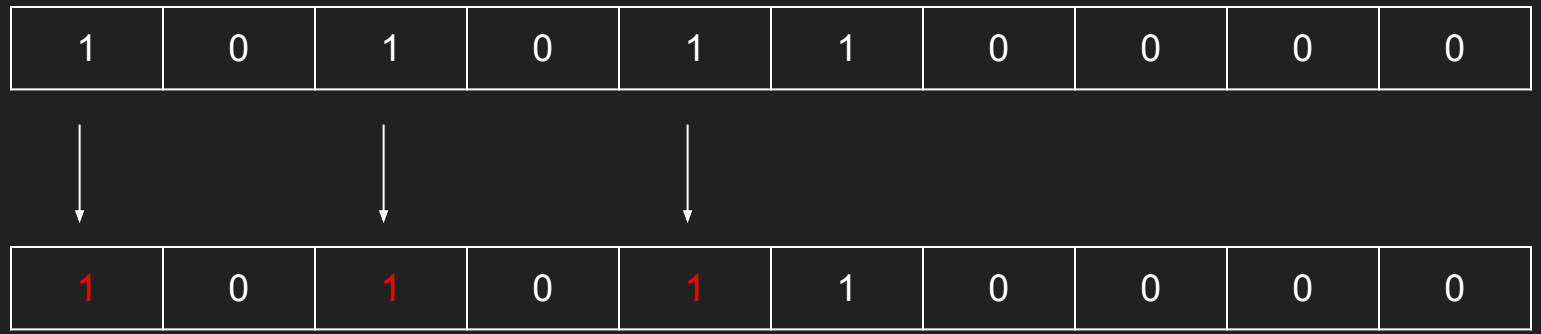
hashFuncA(“franz”) = 0
hashFuncB(“franz”) = 2
hashFuncC(“franz”) = 5



definitely not in the set

Attempt: “pa1980”

hashFuncA(“pa1980”) = 0
hashFuncB(“pa1980”) = 2
hashFuncC(“pa1980”) = 4



probably in a set

Interactive Bloom Filter

https://www.jasondavies.com/bloomfilter/

In a nutshell

- If any of the hashed indexes for a value is ‘0’ then, the value is definitely not in the set.
- If all of the hashed indexes is ‘1’, then ‘maybe’ the value is in the set. (probabilistic nature)

Bloom Filter size

if size is too small → quicker to be filled up → more ‘false positive’

Number of Hash Functions

the more the hash functions → the slower the bloom filter,
and the quicker to fill it up
However, if we have too few → more false positives

Error Rate

$$p \approx (1 - e^{\frac{-kn}{m}})^k$$

p = error rate
m = filter size
k = the number of hash functions
n = the number of elements inserted

WTF ?

Bloom Filter Calculator

https://hur.st/bloomfilter/?n=33916470&p=0.05&m=&k=

Real World Bloom Filter Application

- Cassandra: avoid expense disk read
  https://docs.datastax.com/en/archived/cassandra/3.0/cassandra/dml/dmlHowDataWritten.html
  https://docs.datastax.com/en/archived/cassandra/3.0/cassandra/dml/dmlAboutReads.html

- Medium : determine if a user already read post
  https://blog.medium.com/what-are-bloom-filters-1ec2a50c68ff

Question

What if we want to remove username from bloom filter ?

Reminder: Payme

Tim & Temp