Distributed Probabilistic Counting on ClickStream data

Michael Vogiatzis
Software Engineer
@mvogiatzis

About me

- Software Engineer in London
- Interested in
  - All things scaling
  - Real-time processing
  - Open Source
  - Good whiskey
- Msc in CS, University of Edinburgh

What is ClickStream data?

- Recording of user clicks on a web page or software app
- User actions are logged for further analysis
- USA Gov gives (some of) them for free!

ClickStream use cases

- User behavior analysis
  - Join with CRM, identify users, target ads
- Market research
- Software testing
- UX improvement
- Fraud detection

Problem

Find the most visited pages and/or most active users in a set of websites

Problem

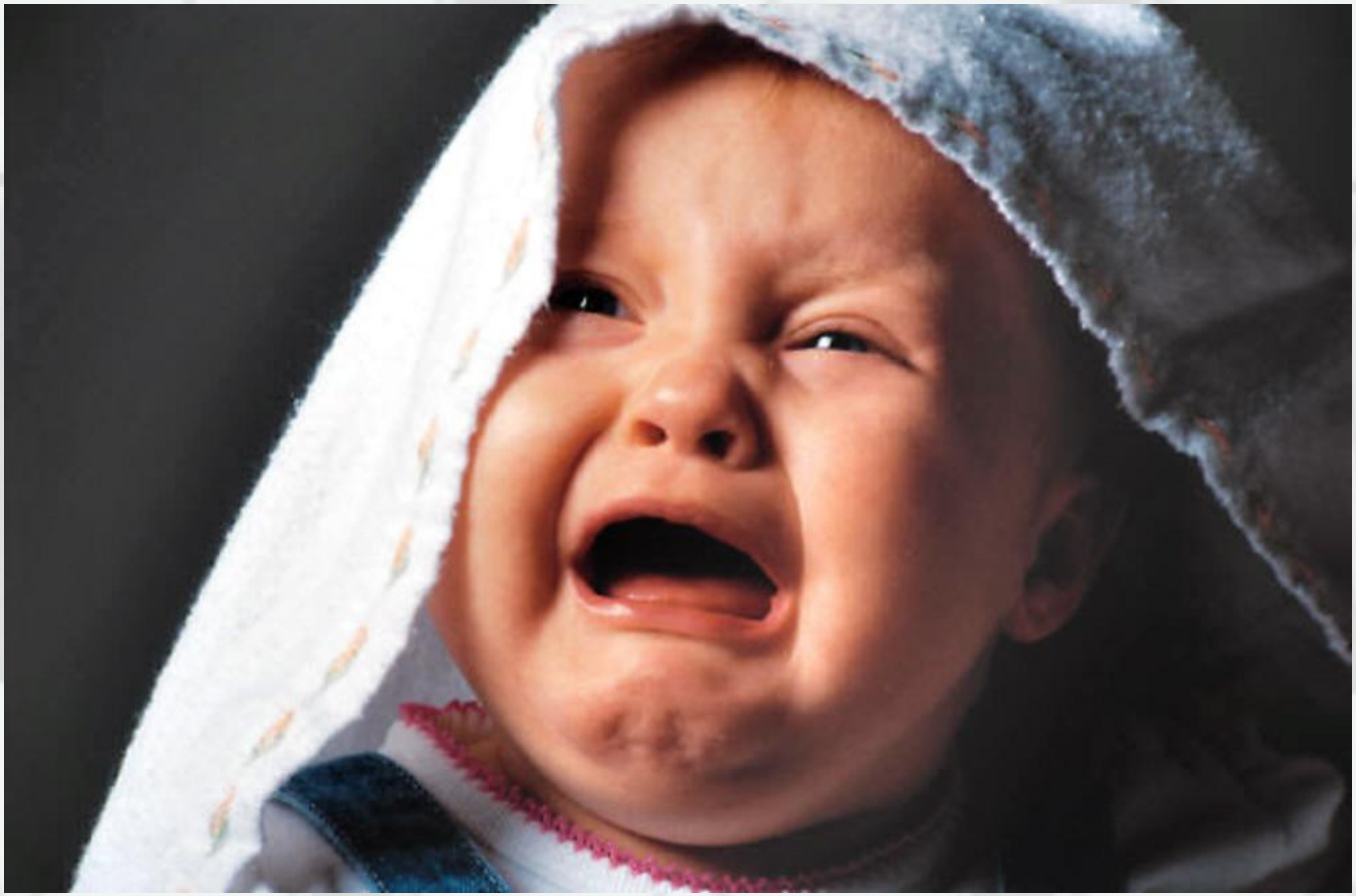
Find the most visited pages and/or most active users in a set of websites

count++

Problem Solved !

Solved?

Now count clicks on the ***Amazon*** website in **real-time** including **spikes**.



@mvogiatzis

Randomised Algorithms
Motivation

What if storage / processing times grows exponentially?

Randomised Algorithms Motivation

What if storage / processing times grows exponentially?

- Computational & space efficiency

Randomised Algorithms Motivation

What if storage / processing times grows exponentially?

- Computational & space efficiency
- Scaling

Randomised Algorithms Motivation

What if storage / processing times grows exponentially?

- Computational & space efficiency
- Scaling
- No need for *exact* counts
  - Perhaps top-N most frequent ones

Randomised Algorithms Motivation

- Randomised approaches are often the most efficient approach to many classes of demanding problems
- Trade-off between error rate and performance
- Bloom filters, Locality sensitive hashing, probabilistic counting, etc.

Probabilistic Counting

- Often referred to as “Approximate counting”
- “The approximate counting algorithm allows the counting of a large number of events using a small amount of memory.”
  - Wikipedia
- A randomised counting approach

How It Works

- Every time a new event comes update the counter with probability 2^{-f}
  - where f = current_frequency
- Meaning: Update the counter only if a random generated number (sampled uniformly between 0 and 1) is less than 2^{-f}

How It Works

- Every time a new event comes update the counter with probability 2^{-f}
  - where f = current_frequency
- Meaning: Update the counter only if a random generated number (sampled uniformly between 0 and 1) is less than 2^{-f}
- We now need log(log(f)) bits per counter instead of log(f)

  This counts in log-space.

Example

If Random_Num < 2^{-prev_count}
Count++;

Stream	Prev Count	Random Num	Curr Count
“dummy”	0	0.6	1
“dummy”	1	0.7	1
“dummy”	1	0.3	2
“dummy”	2	0.2	3
“dummy”	3	0.5	3

Exact results

- We may end up over-counting or under-counting
- $E[2^f-1]$ gives the estimated real count
- Decrease the error rate by using a smaller base than 2
  - Updates more frequently

Scale with Storm

- Single machine counting may not be enough

Scale with Storm

- Single machine counting may not be enough
- Use multiple machines for *load balancing* and *fault-tolerance*

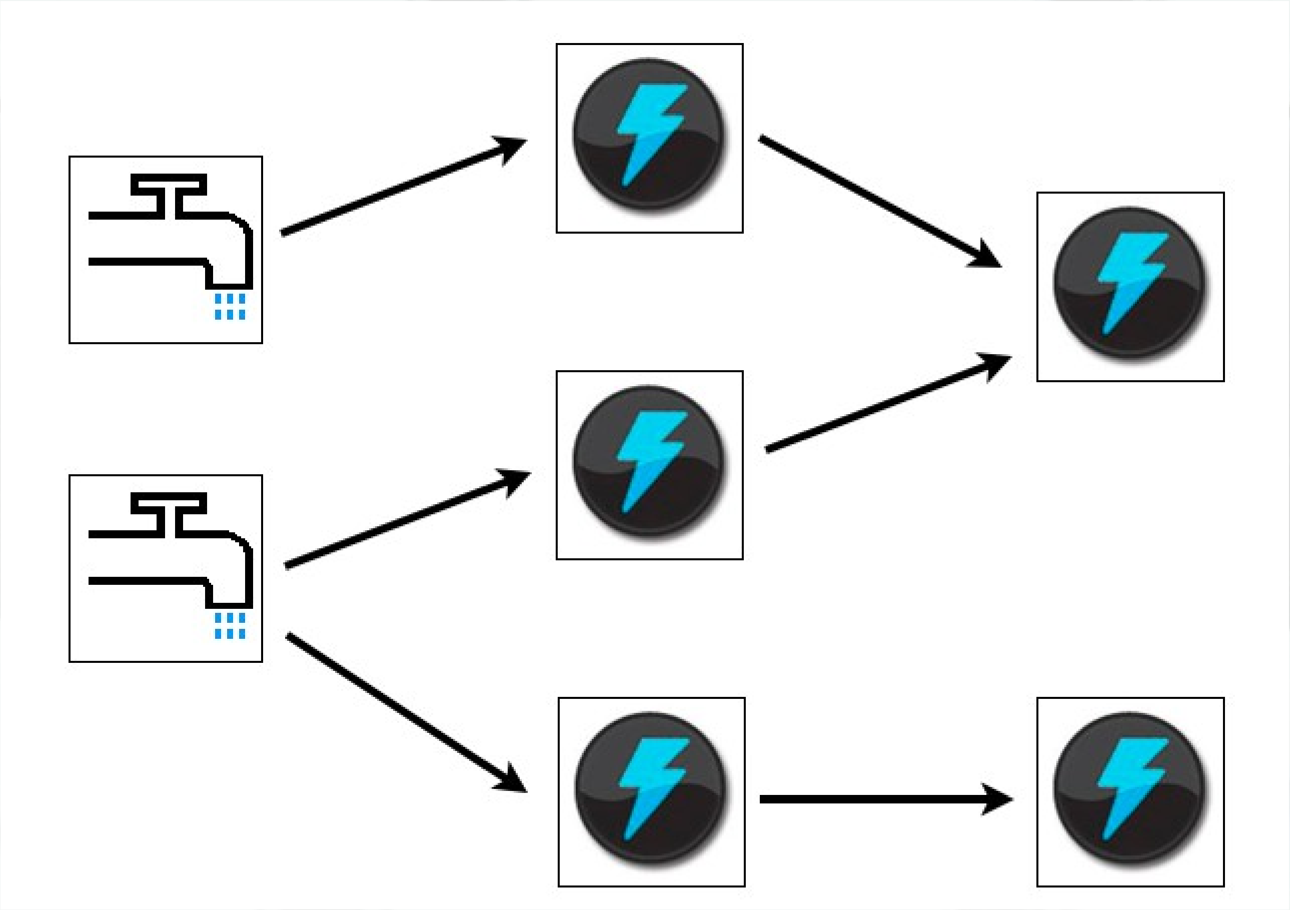
Storm

- Distributed real-time computation system
- Fault-tolerant
- Fast
- Scalable
- Guaranteed message processing
- Open source
- Multi-language capabilities

Elements

- Streams
  - Set of tuples
  - Unbounded sequence of data
- Spout
  - Source of streams
- Bolt
  - Application logic
  - Streaming aggregations, joins, DB ops

Topology

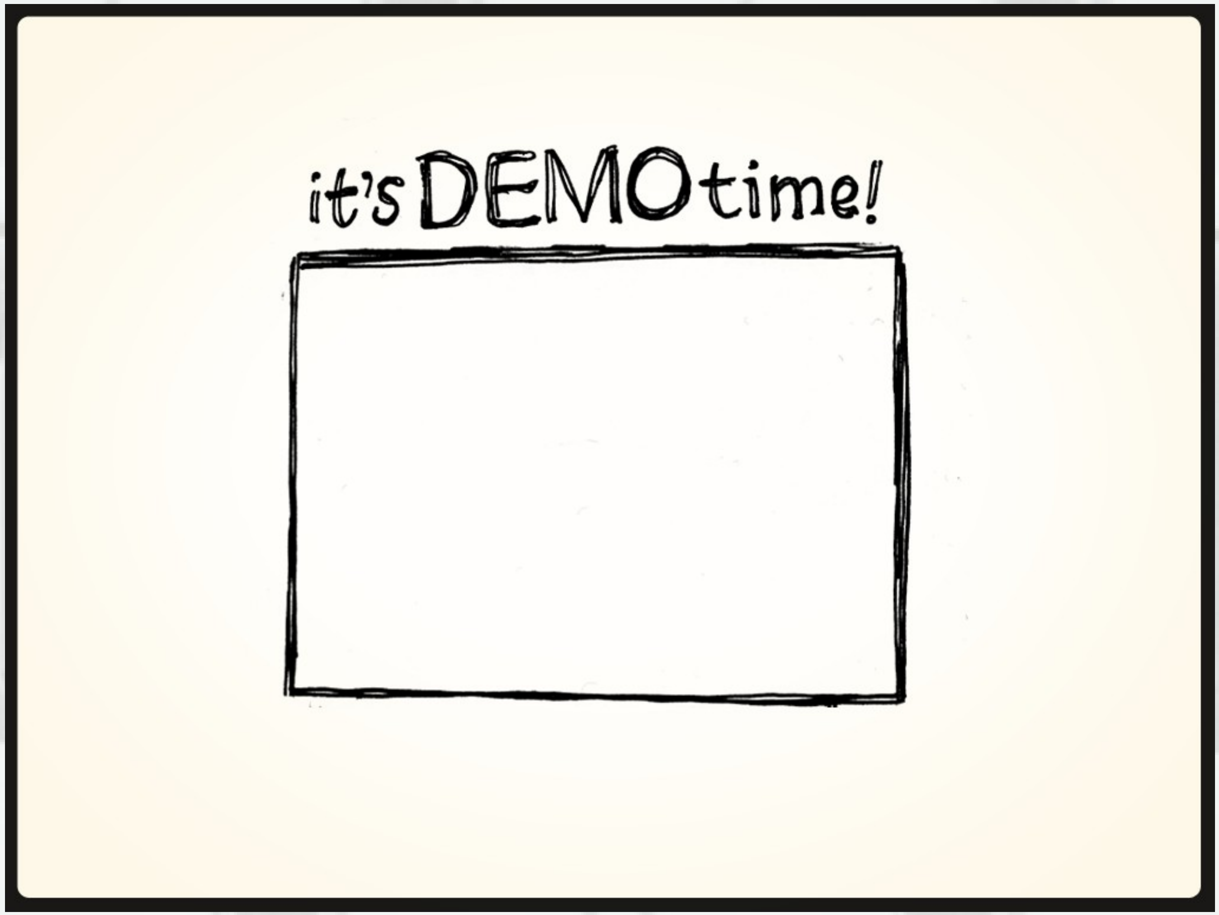


Data source

- 1.USA.gov URLs are created whenever anyone shortens a .gov or .mil URL using bitly.
- http://developer.usa.gov/1usagov

Data format

```
{
    "a": USER_AGENT,
    "c": COUNTRY_CODE, # 2-character iso code
    "nk": KNOWN_USER,  # 1 or 0. 0=first time we've seen this browser
    "g": GLOBAL_BITLY_HASH,
    "h": ENCODING_USER_BITLY_HASH,
    "l": ENCODING_USER_LOGIN,
    "hh": SHORT_URL_CNAME,
    "r": REFERRING_URL,
    "u": LONG_URL,
    "t": TIMESTAMP,
    "gr": GEO_REGION,
    "ll": [LATITUDE, LONGITUDE],
    "cy": GEO_CITY_NAME,
    "tz": TIMEZONE
    "hc": TIMESTAMP OF TIME HASH WAS CREATED,
    "al": ACCEPT_LANGUAGE
}
```


it's DEMO time!

The End

- Code on GitHub
  - .github.com/mvogiatzis/probabilistic-counting
- Follow me: @mvogiatzis
- E-mail me: michael@micvog.com
- Read my blog: http://micvog.com/

Q & A