Adobe Identity Graph

XLDB Conference ’18

What I am going to cover ?

- What is Identity Graph & Why do we need it?
- How is Identity Graph built ?
- How is it shared, stored and served ?
- Challenges & Performance Strategy

What is Identity Graph ?

Cookies

+

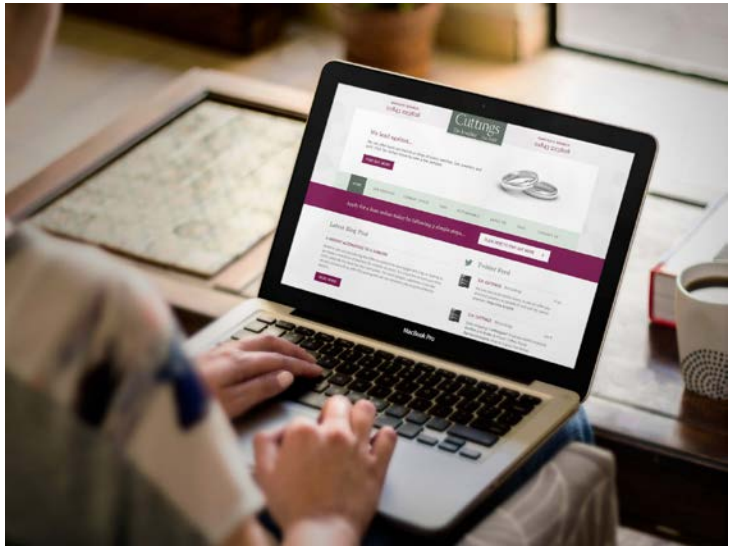
More cookies getting baked

Identity Graph

- An *Identity graph* connects all the known identifiers that correlate with individual consumer.
- Provides a holistic view of a consumer across different devices. .
- It is build up using *anonymous identities* (cookie data).

Why do we need Identity Graph ?

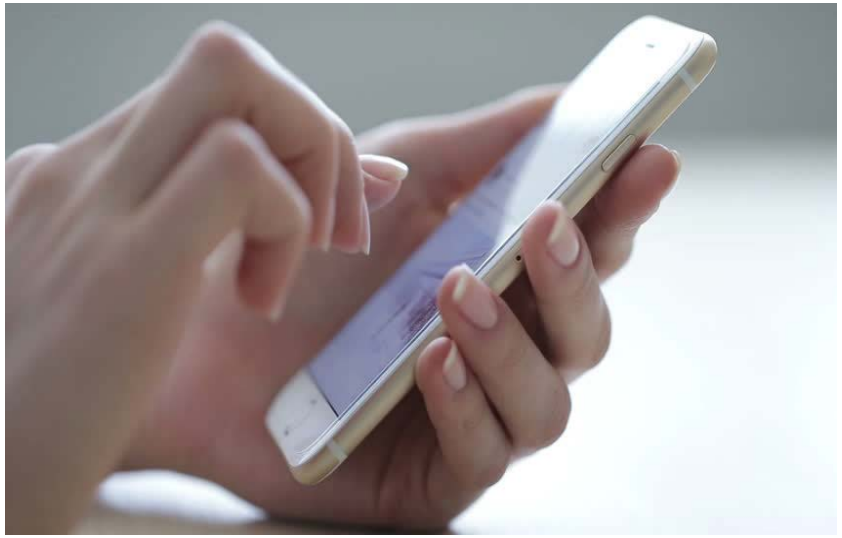
Imagine a following scenario:



Consumer visits a website on his laptop

Sign up for updates on the website. They fill up a form.

What happens next:



The next day the consumer picks up their mobile and re-visits the website.

Oops— we know nothing about them. It looks like a new surfer on the website.

If the same customer goes to retail store to purchase the product. It again seems to be a new customer.

What is missing here ?

Identities across these devices (online + offline) are not unified.

Solution

Identity graph helps in connecting this data to solve the fundamental problem in Digital Marketing domain.

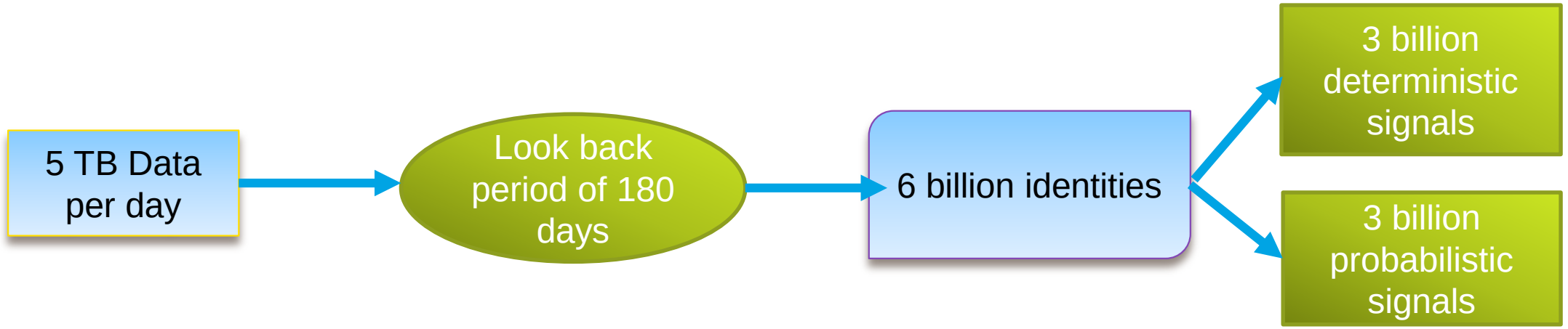
Links fragmented profile of a consumer on different devices to provide a unified view.

How do we build Identity Graph?

Built by discovering identity relationships in online consumer event logs.

- Data Ingestion

We ingest 5 TB of web logs per day and process it to extract relevant information.



Data Processing

- Data processing is done in a scalable and distributed fashion using Spark clusters
- Sparks clusters helps in parallel data processing and reduces CPU cycles by a large number.
- Scans & processes 180 days of data in approx. 5 hrs.

Sharing & Storing Identity Graph

- Graph data gets shared & stored in form of cluster data.
- Cluster contains all the related identities and their association times.
- This makes it easier to store data in form of key, value pairs into two data collections.
- No SQL DB – AWS Dynamo DB & Azure Cosmos DB
- **Data set size : 16 TB with 28 billion records**

Serving Identity Graph

We serve Identity Graph using API's. in cloud agnostic manner and is extendable to any cloud our customers operates on.

- One of the important API is "Members API" - Given an identity, give me all the connected identities.
- These API's are read intensive.
- It *can support up to 3 million lookups/second* which caters to different use cases.

Goal

High Throughput, Less Latency

Schema Design plays an important role !

Two way collection look up ?

Record 1

Record 2

Record 3

Cluster:

cluster ID: 123
Identity: xx66yx
Time:1423456679
....

Key : xx66yx

Cluster:

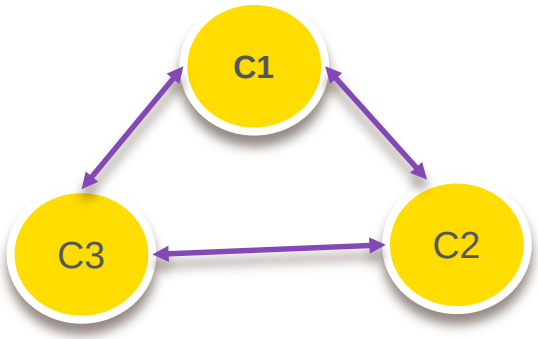
cluster ID: 123
Identity: xx66yz
Time:1423456688
.....

Key : xx66yz

Cluster:

cluster ID: 123
Identity: xx66yw
Time:14234566887
.....

Key : xx66yw

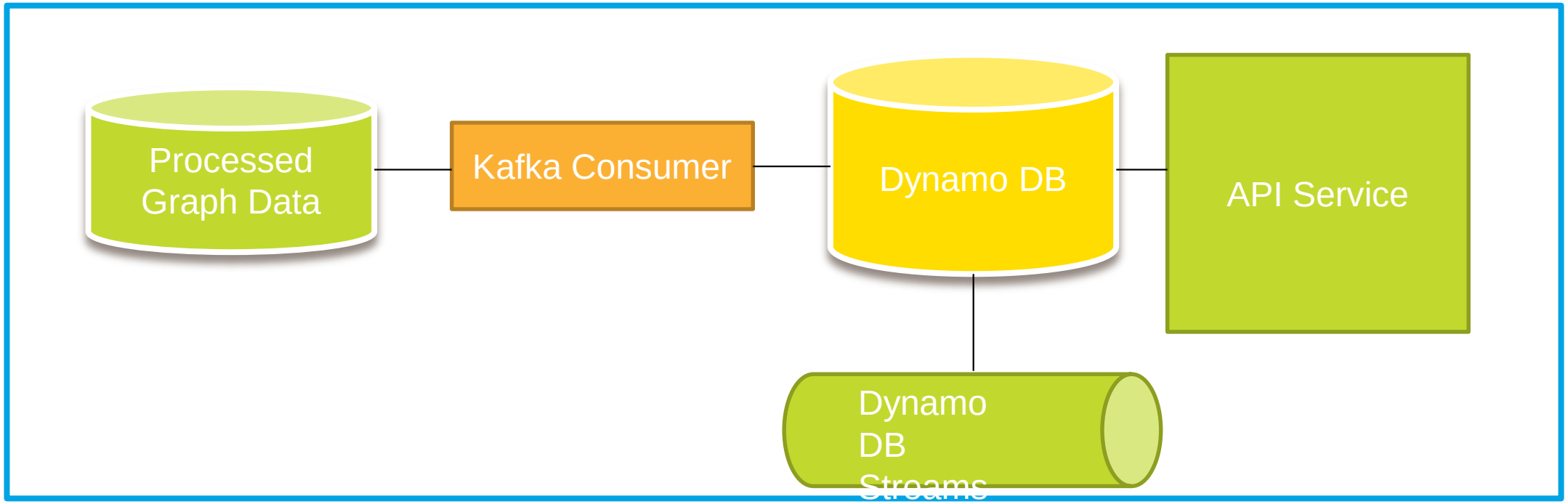


Identities: [xx66yx
Xx66yz.
xx66yw.
]
Cluster ID: 123

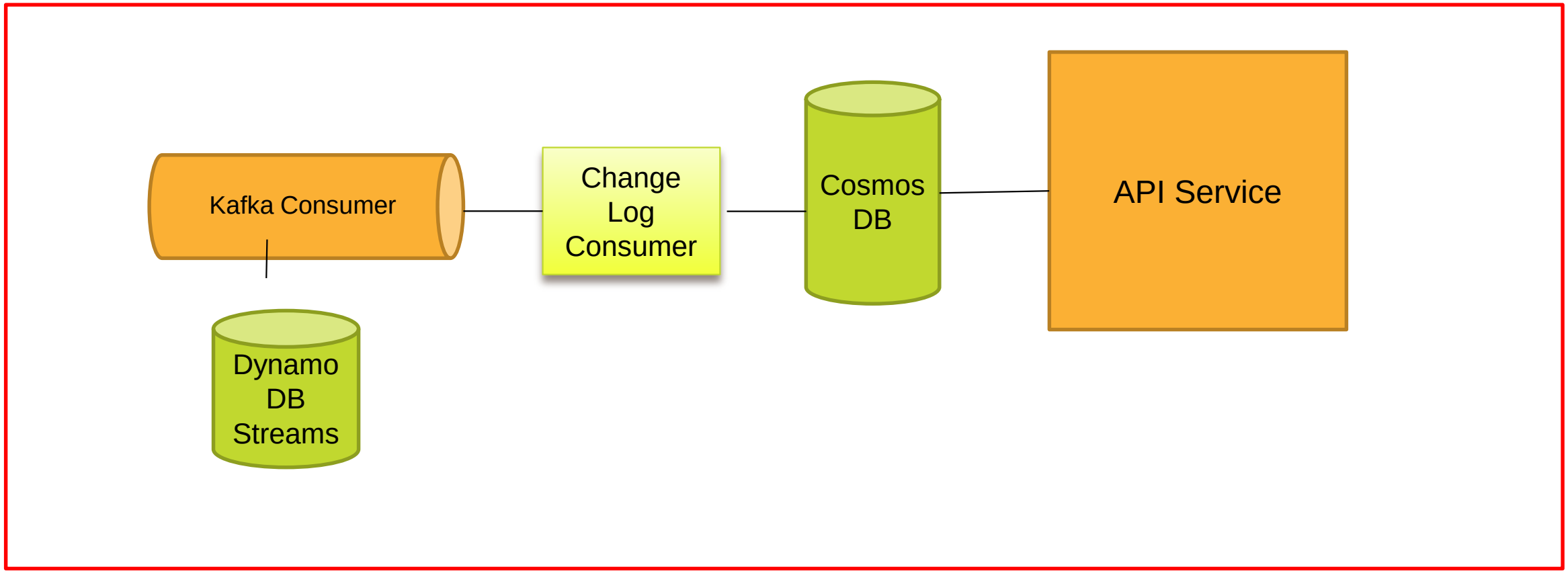
Key : 123

Identity Graph Architecture

It is a cross data center, multi cloud , globally distributed system of storing and serving graph data.



Identity Graph on Azure



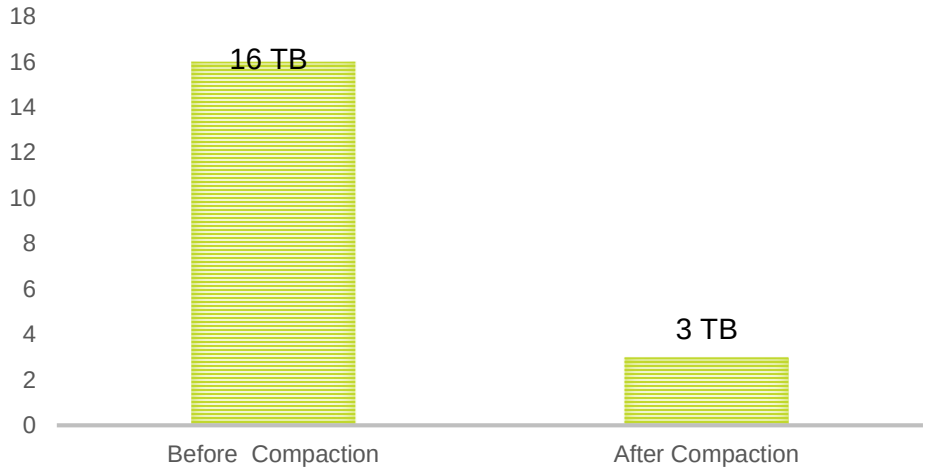
Performance Strategies for building Identity Graph

Data Compaction

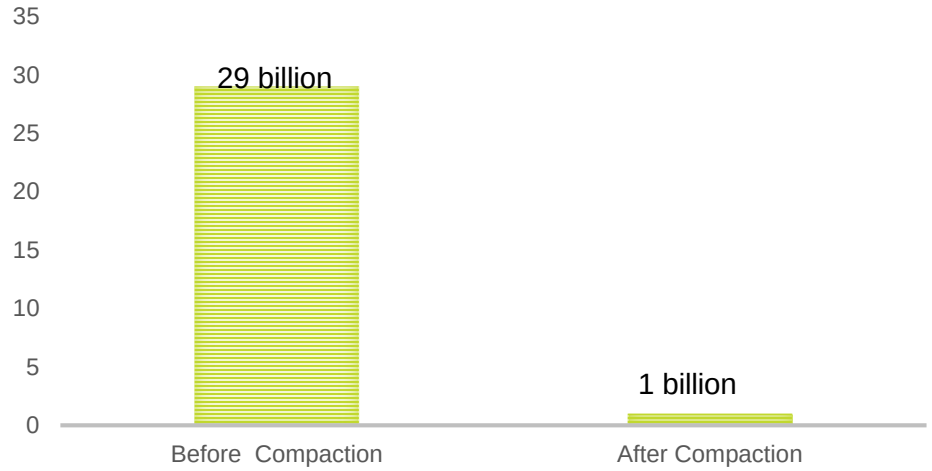
- Data compaction is packing up the similar documents into one pack.
- We compute hash of the cluster ID (32 bits hash) using *Murmurhash3* and pack all the documents which result in same hashed id in to one pack id.
- It results is data distribution over less number of partitions and helped us achieve better performance.

Data Compaction Results

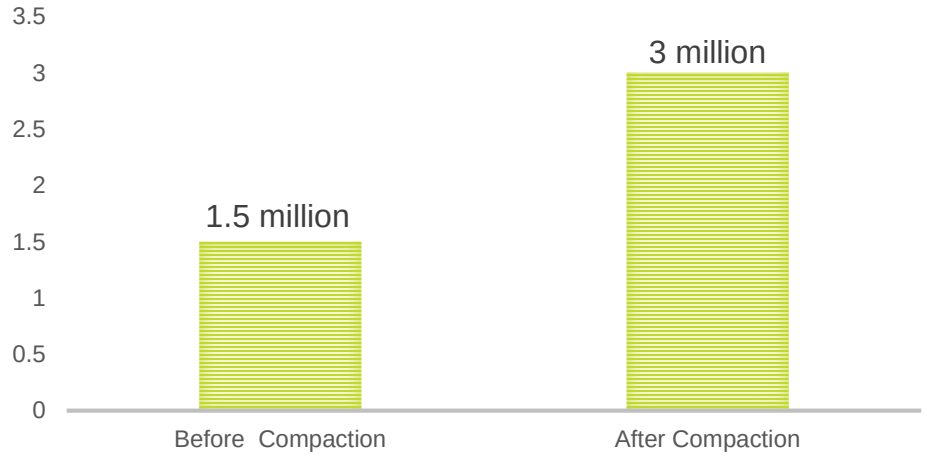
DATA SET SIZE



RECORD COUNT



THROUGHPUT - LOOK UPS/SEC



Thank you !

Akanksha Nagpal
Software Engineer, Adobe


Adobe