Data ~~Management~~ Compression for Data Lakes

Jonathan WINANDY
primatice.com

About ME

- I am (Not Only) a Data Engineer

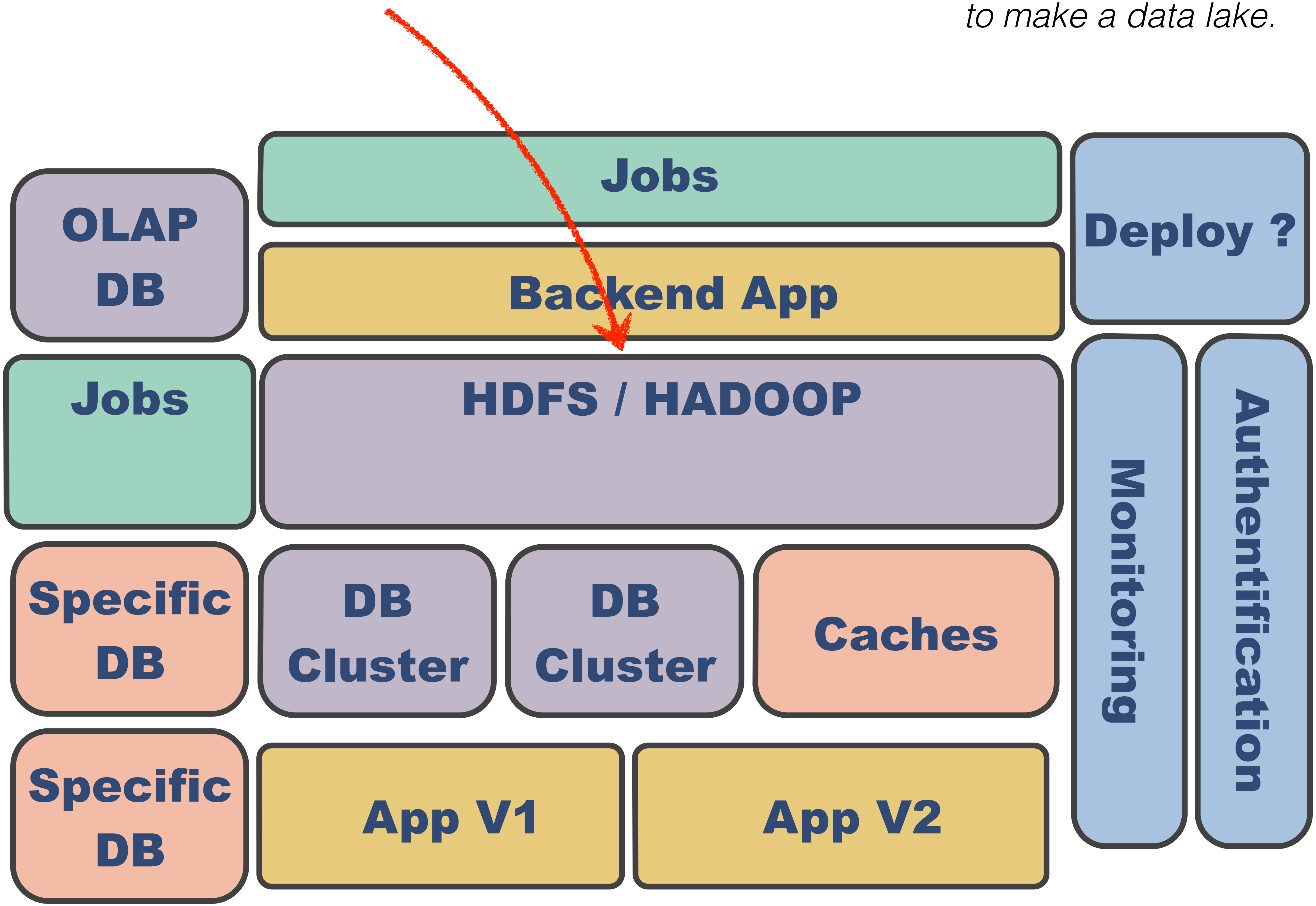
Contents

- Data Lake : What is a Data Lake ?
- Compute : What can we do with the gathered data ?
- Conclusion !

Data Lake

Infrastructure > The Data Lake*

(*) you actually need more than just Hadoop to make a data lake.



Jobs

OLAP DB

Backend App

Deploy ?

Jobs

HDFS / HADOOP

Monitoring

Authentification

Specific DB

DB Cluster

DB Cluster

Caches

Specific DB

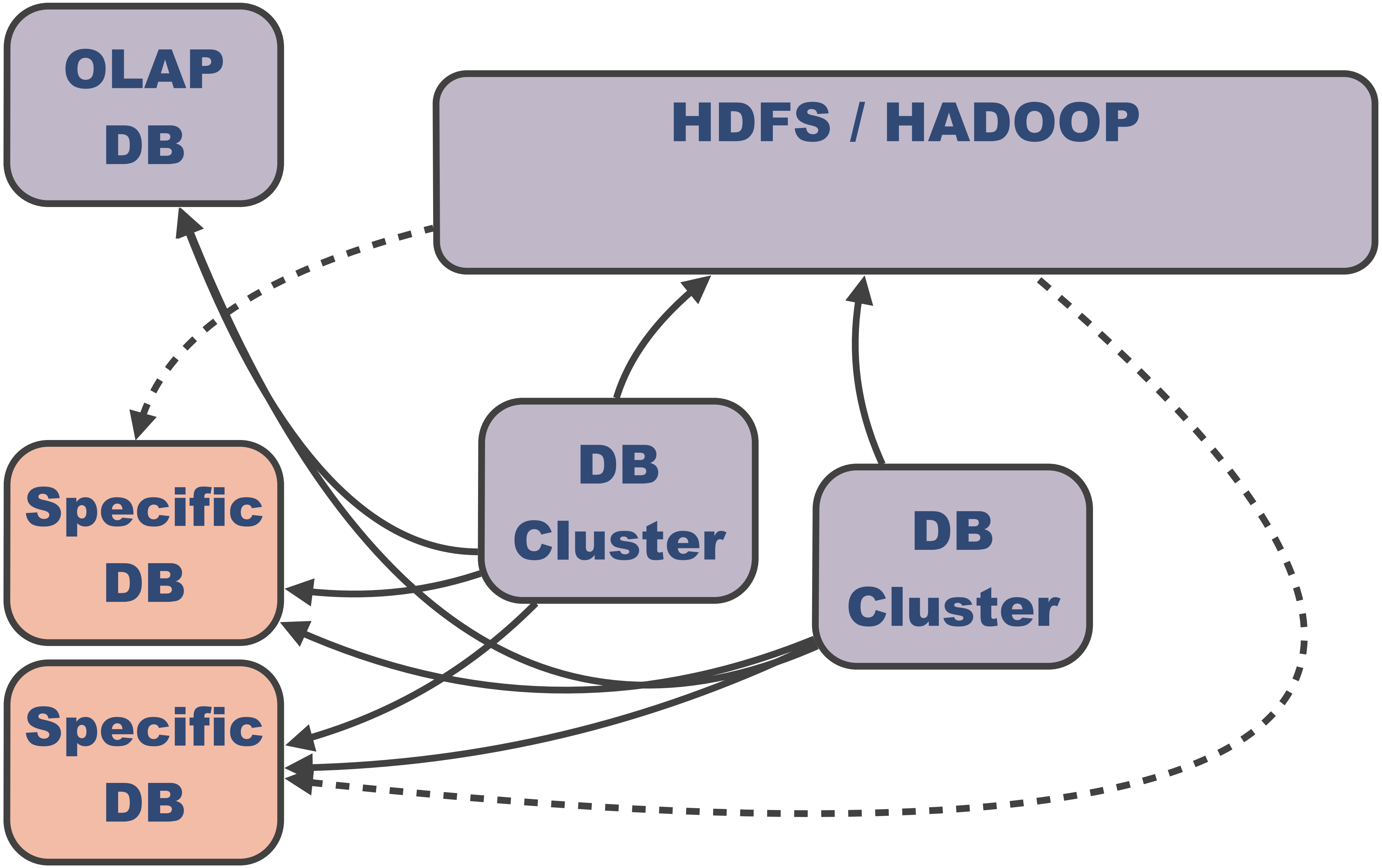
App V1

App V2

Data Lake

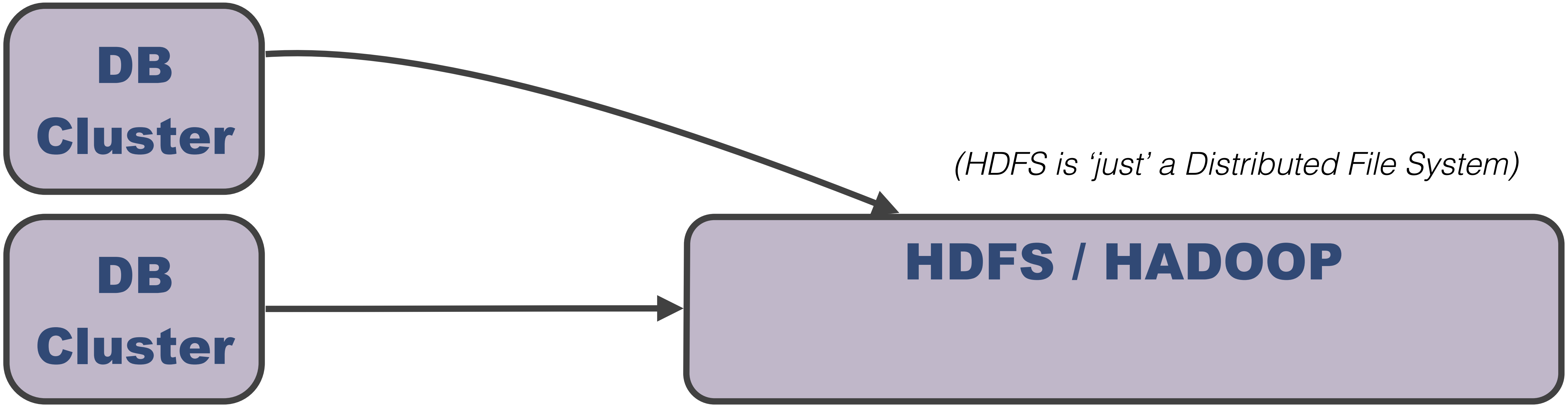
We use HDFS to store a copy of all Data.

And we use HADOOP to create views on this Data.



Offloading to HDFS

Offloading data is just making a copy of databases data onto HDFS as a directory with a couple of files (parts).



(HDFS is 'just' a Distributed File System)

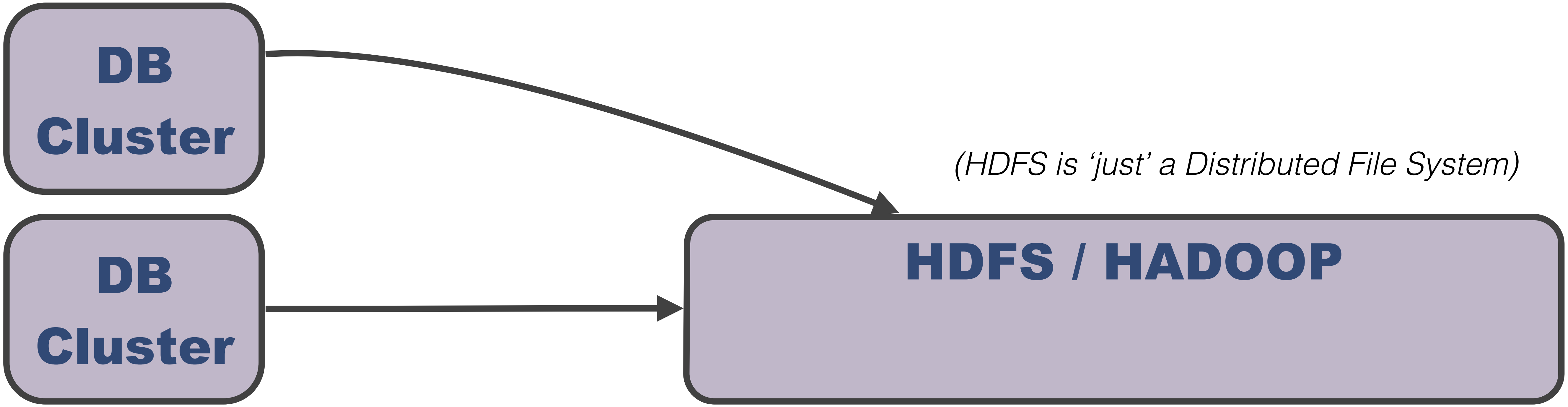
Data Lake

For relational databases, we can use apache sqoop to copy for example the table ***payments*** from schema ***sales***

into ***hdfs://staging/sales/payments/2014-09-22T18_01_10***

into ***hdfs://staging/sales/payments/2014-09-21T17_50_25***

into ***hdfs://staging/sales/payments/2014-09-20T18_32_43***



(HDFS is 'just' a Distributed File System)

Data Lake

A light capacity planning for your potential cluster :

- If you have 100 GB of binary compressed data*,
- With a cost of storage around 0,03$/per GB/per month,
- An offload every day would cost 90$/per month of kept Data,
- In six month, this offloading would cost 1900$ and weight 18TB (<- this is ***Big Data***).

* obviously not plain text/JSON

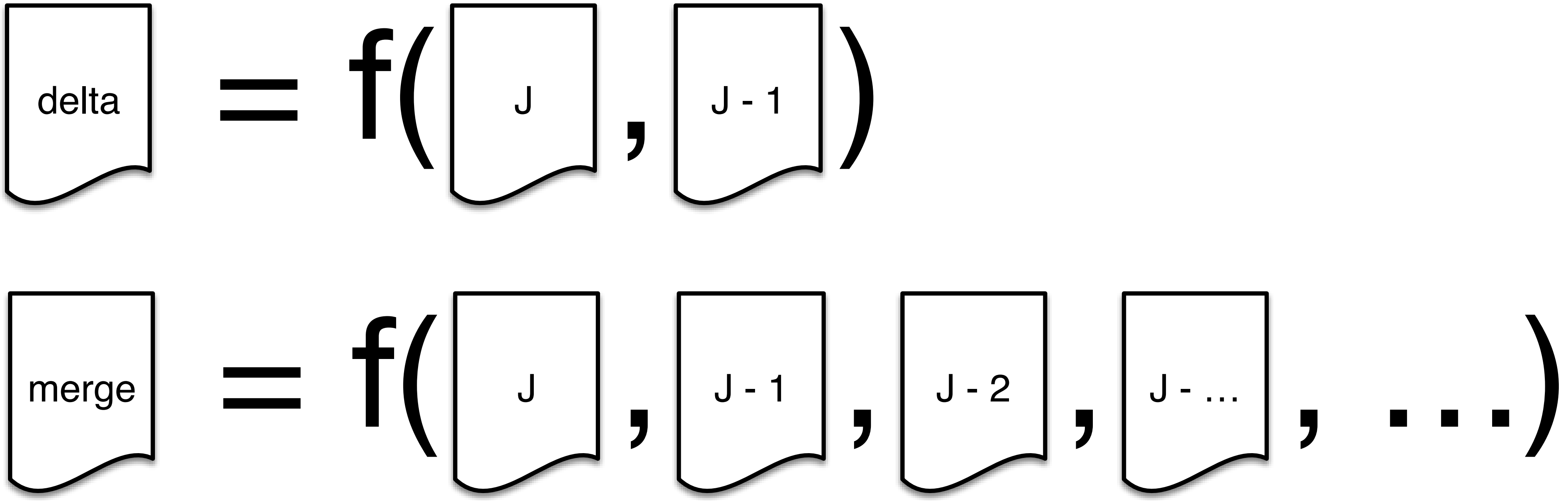
Data Lake

But for this 18TB, you have quite a lot of features :

- You can track and understand bugs and data corruption,
- Analysing the business without harming your production DBs,
- Bootstrap new products based on your existing Data,
- And also have now a real excuse to learn Hadoop or make some use of your existing Hadoop bare metal clusters !

Compute

Having a couple of snapshots in HDFS, we can use the tremendous power of MapReduce to join over the snapshots, and compute what changed in the database during a day.



delta = f(J , J - 1)

merge = f(J , J - 1 , J - 2 , J - … , …)

Compute

Δ function takes 2 (or +) snapshots and output a merge view of those snapshots with an extra column ‘dm’.

Inputs :

Δ(

id	name
1	Jon
2	Bob
3	James

,

id	name
1	John
3	James
4	Jim

) =

Output :

id	name	dm
1	John	01
1	Jon	10
3	James	11
4	Jim	01
2	Bob	10

https://github.com/viadeo/viadeo-avro-utils

Compute

Δ function takes 2 (or +) snapshots and output a merge view of those snapshots with an extra column ‘dm’.

Inputs (generalised) :

Δ(

id
1
2
3

,

id
1
3
4

,

id
1
4
5

,

id
1
3
5

) =

Output :

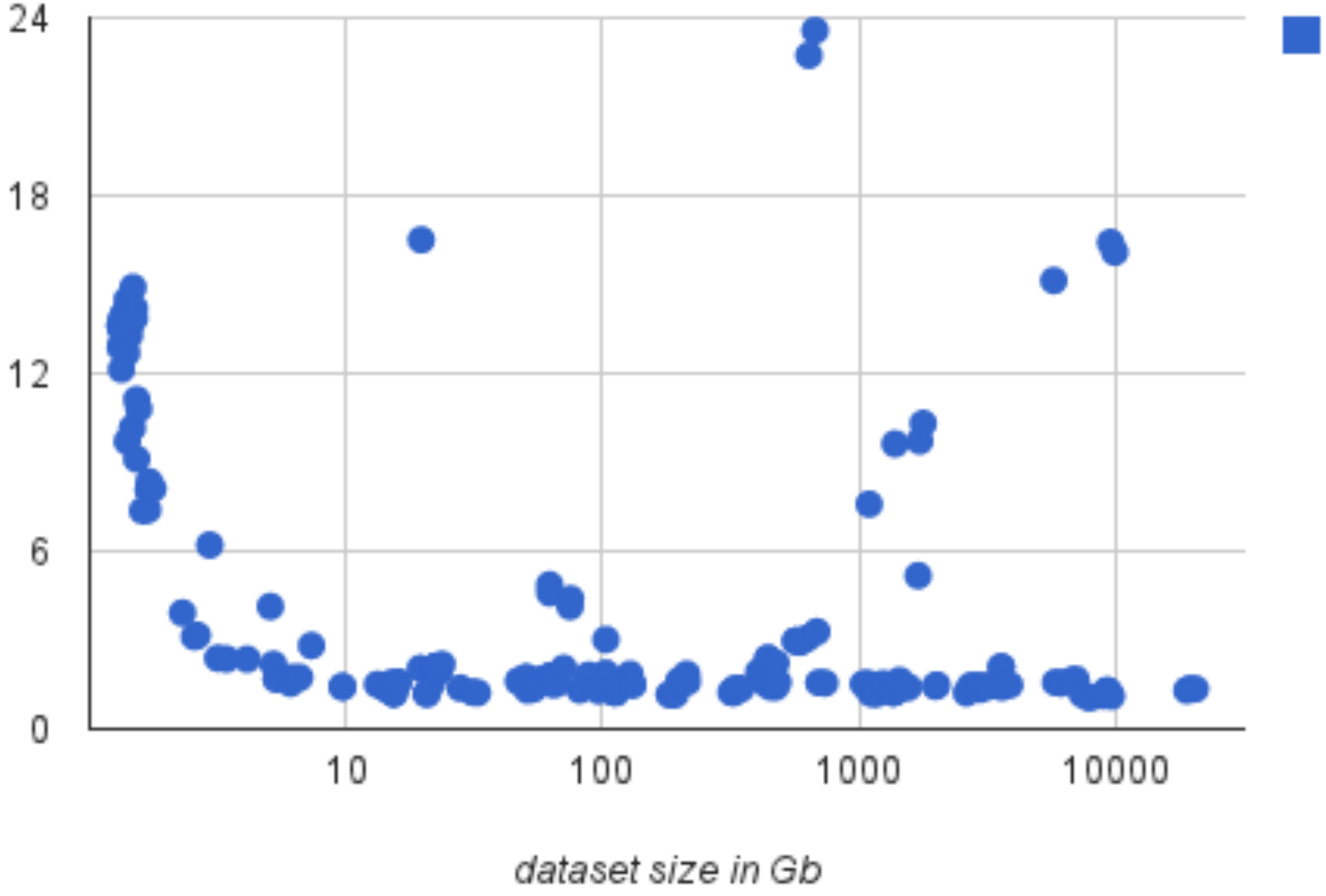
id	dm
1	1111
2	1000
3	1101
4	0110
5	0011

https://github.com/viadeo/viadeo-avro-utils

Compute

Analysing the compression : merging 50 snapshots

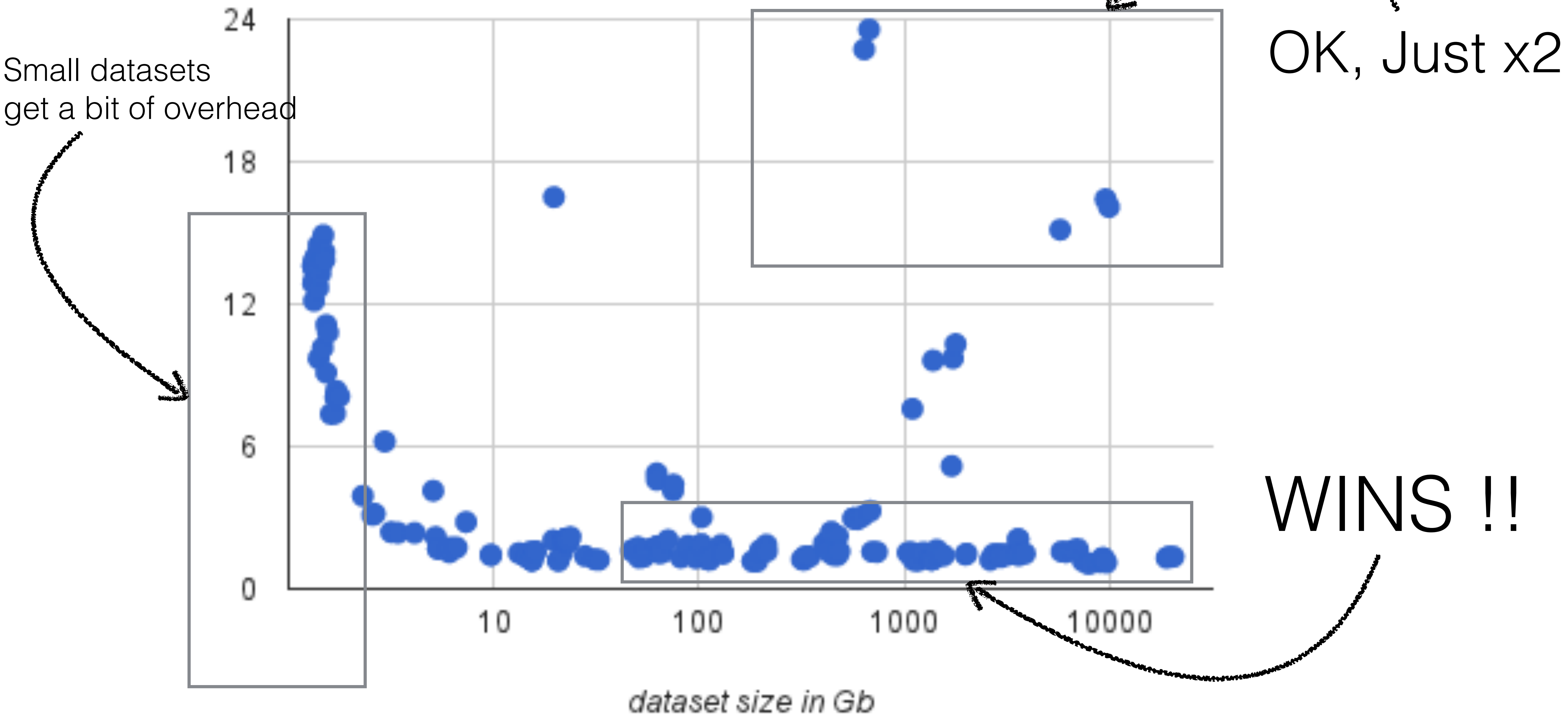
Ratio : Size of merge output / Size of largest input
Less is better, ≤ 1 is perfect



Compute

Analysing the compression : merging 50 snapshots

Ratio : Size of merge output / Size of largest input
Less is better, ≤ 1 is perfect



Small datasets get a bit of overhead

OK, Just x2

WINS !!

Conclusion

- So now the offloading might cost just 36$ of storage for 6 month (6$ per month).
- Welcome back at small data !
- But what does this mean for production Data ?
  Did you really need to store that in a mutable database in the first place ?

Thank you,
Questions ?