

Clickhouse@Infovista

Thierry Hanot - Architect

infovista

KNOW YOUR NETWORK™

Who We Are

The leader in modern network performance

We give you complete visibility and unprecedented control to deliver brilliant experiences and maximum value with your network and applications

Global Presence

Strategic regional headquarters in **Washington**, **Paris** and **Dubai** to offer in-region services and support



More than **700** employees in **12** regional offices



Partnerships and alliances with major industry leaders



More than **1,500** customers in **150** countries



Our complete portfolio covers enterprise and service providers from **SD-WAN** to **5G** and **beyond**

Who am I ?

- An architect with a geek dream job
- My role :
 - Discover, play and test all the open source or commercial technologies relevant in our domain
 - Define the proper technologies for the different R&D use cases
 - Then follow R&D during implementation to give them the right training and support to use those technologies.



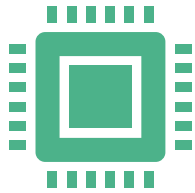
What we do :Solutions for Complete Network Lifecycle



Network Company = Data Company

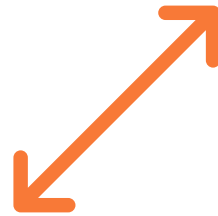
DOMAIN	Role	TYPES OF DATA	VOLUME	LEGACY Technology
RAN Engineering	Plan Optimise	Coverage Maps Call Trace Counters	Large to Very Large	Files/SQL Server Oracle
Network Testing	Assess Benchmark Troubleshooting	Geo-Localized TimeSeries Logs Coverage Maps	Large	SQL Server SOLR
Service Assurance	Supervise Monitor	Time Series CDR	Large	ORACLE MYSQL
SDWAN	Control Optimise	CDR	Large to Very Large	ORACLE POSTGRES

Data Problematics (1)



Heterogenous technology

Difficulties to correlate data together



5G : Amount of data will increase exponentially

Coverage map a more precise and will include one additional dimension (3D)

Number of Counters/Metrics increase with reduced granularity



Software Defined Networks/WAN

More analytics have to be computed to drive automation

- Sum/Min/Max/Avg are not enough
- More real time

Data Problematics (2)



Multi Tenancy

Ability to filter data for particular user in a simple and secure way
Integration with third party user management



Integration with BI

Our customer already have corporate BI and they want to use it on our data



Sensitive Data

GPDR : Ability to delete (alter)
Ability to encrypt

Data Problematics (3)



Operational simplicity

Many customers with all different topologies of data. We cannot afford to spend weeks to tune the system per customer.



Large panel of analytics capability

To handle a very large diversity of use case

Which should be extended of time



Jdbc/odbc interface, Rest API, Standard SQL language

To integrate in a larger ecosystem



Ability to handle text in a flexible way

To correlate and search efficiently

What we have tried

Technology	Temporal Series	Geographic Data	Data records (CDR/Call Trace)	Pseudo Real time	batch	Computing capabilities	Operational Simplicity
Hive	-	+	+	--	++	+	--
Impala with Kudu	-	+	++	Unstable	++	++	+
Impala with Parquet	+	+	++	+	++	++	+
Impala with ORC	+	+	++	+	++	++	+
Druid	+	-	+++	+++	++	-	-
Drill	+	+	+	-	+	++	+
Clickhouse	++	+	+++	+++	+++	+++	+++
Pinot	+	-	++	++	++	-	-
Citus/Postgre	+	+++	++	++	++	+++	++
MariaDB	+	++	++	++	++	+++	++
MySQL	+	++	+	+	+	++	++
Oracle	+	++	+	++	++	+++	+
SQL Server	+	++	+	++	++	+++	++
Vertica	++	+++	+++	+++	++	+++	+++

And the winner is



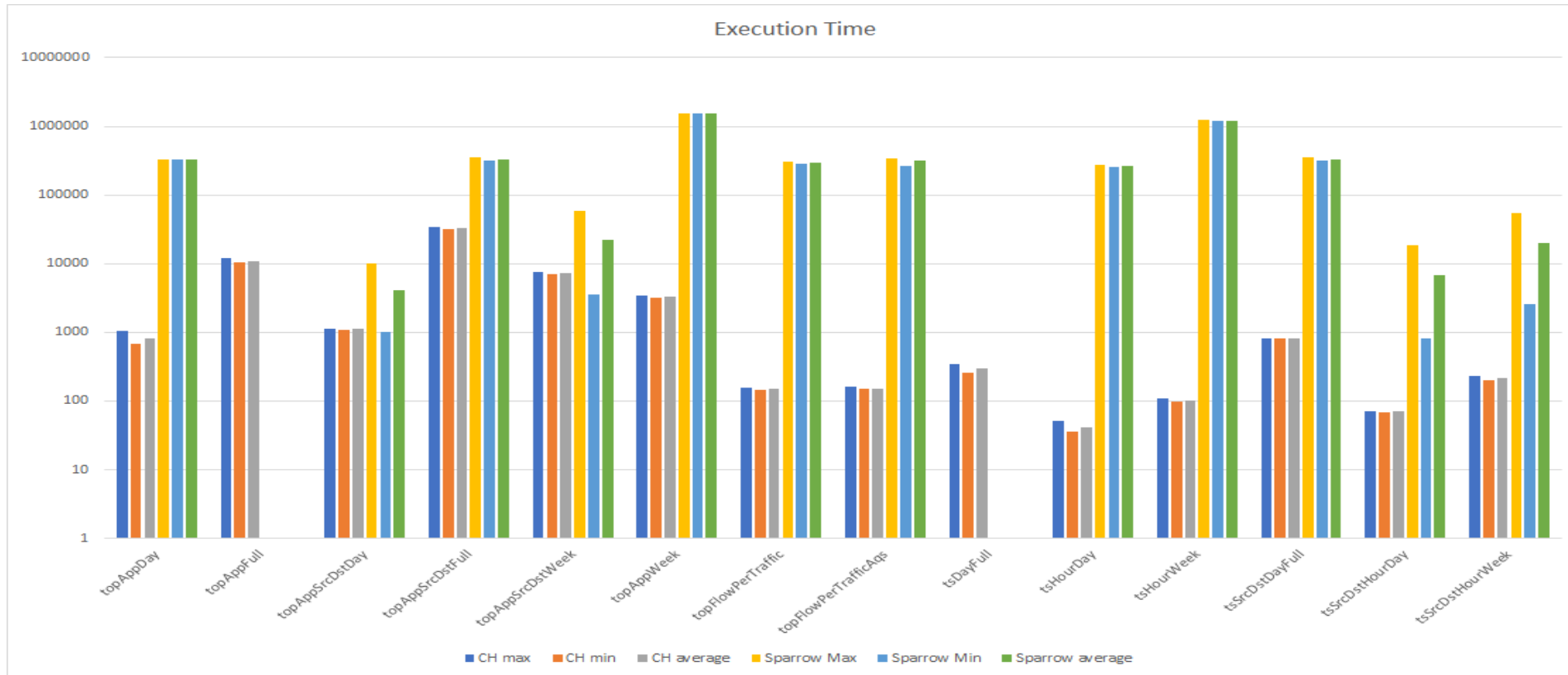
- The only product with the ability to compete with clickhouse was Vertica
 - But cost is really to high so forget it.

First impression with Clickhouse

- Dataset : one billion rows , 68 columns
- Test on impala :
 - require 6 nodes to run aggregation queries in an acceptable time
 - Lot of tuning / tweak (Hadoop, resource manager)
 - Test duration : 3 weeks
- Test on clickhouse :
 - One node
 - Working straight away
 - Some tuning on the model
 - Test duration : 2 days
 - Up to 15 time faster than impala on certain queries

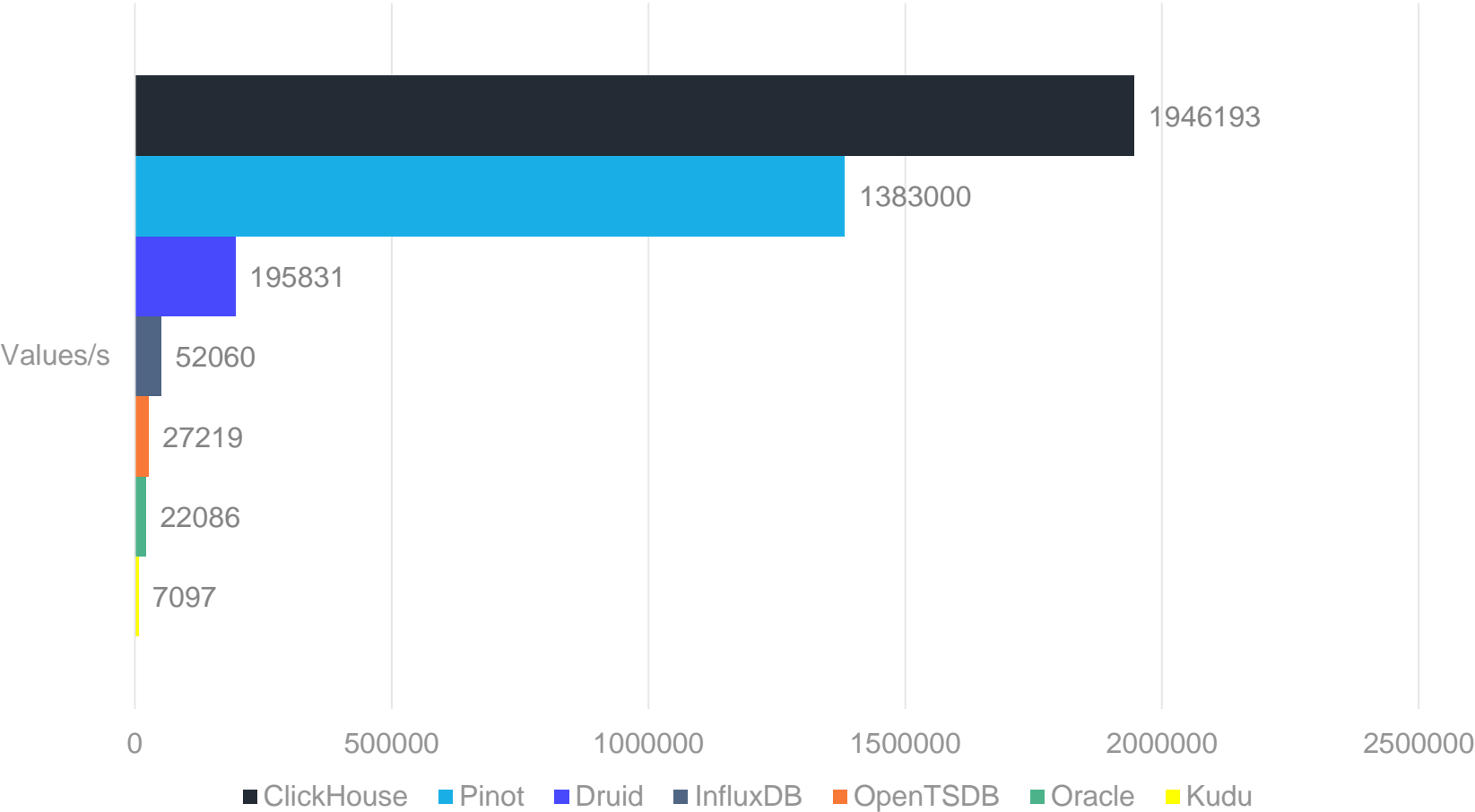


Clickhouse vs MySql (with an optimized storage engine)



Be careful : Y axis
with log scale

Insertion Speed



Features we love in Clickhouse



The speed



The materialized views

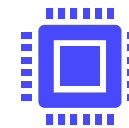
Automated and flexible incremental aggregation.

Clickhouse is more than a query/storage backend , it's a powerfull processing engine



Operational simplicity

30 min to deploy and run the first analysis.



The Sql and set of functions supported

Array management is key if correctly used
What was done with text
: NGRAM to correlate text in a very easy way

....



Design as simple building blocks to let us build what we need

Features we would like to have in clickhouse

Better BI compliance (SQL92)

- Full join support
- Exposition of state
 - Either a user A should not see the state , so when a query is done by this user , the state are automatically merged
 - Either view with full support of push down predicate (enable_optimize_predicate_expression)
 - Preferred solution as it may also allow to expose efficiently arrays

Better null support

- Remove the additional cost
- If a column has away null values in a partition, why storing it

Full GIS support

- Start with a **PointInPolygons** instead of **PointInPolygon** and allow the polygon list to be the result of a query

A better documentation

- Use the community to put more examples (we are still relying on the code and the test of clickhouse to understand how to write some queries)?

Automated compression adjustments

- Chose the best combination of algorithm based on historical data for the new data

A lot more but a lot is already in the roadmap 😊

- **RBAC/LDAP** integration/Encryption/Hot-Cold storage/ML/Query Profiling.....



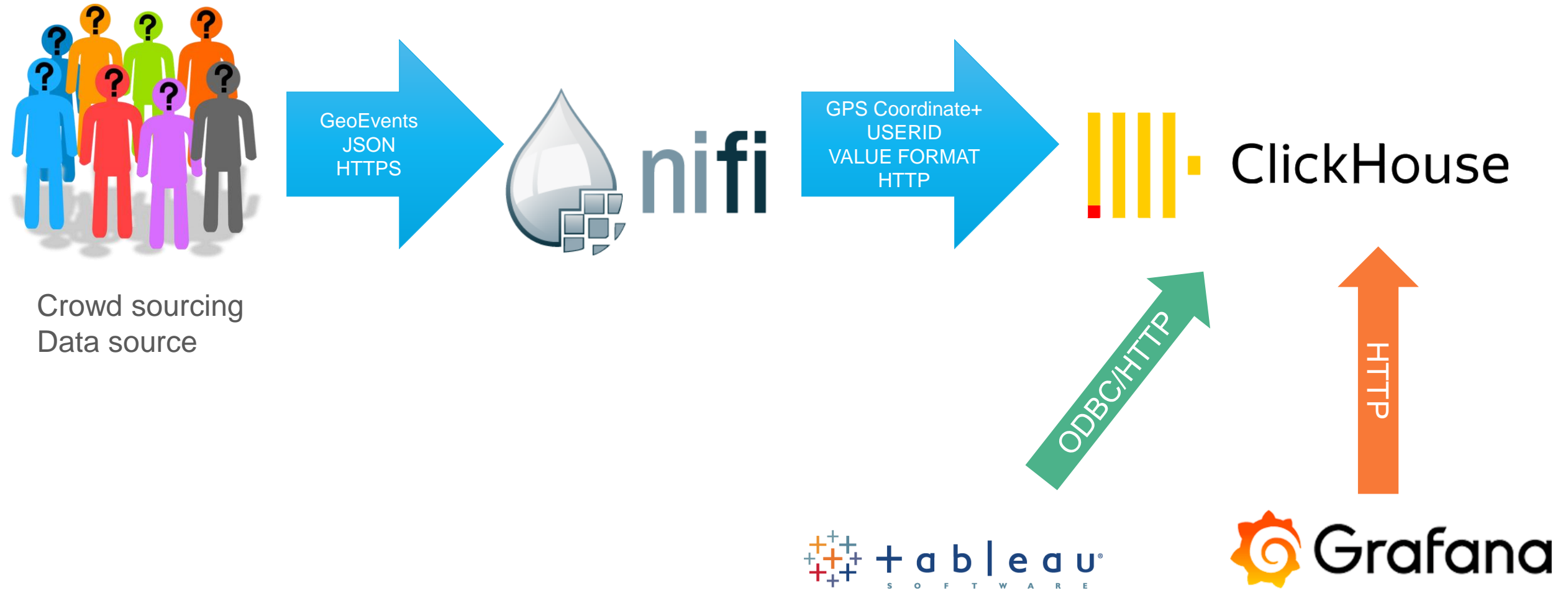
Common mistakes

- Clickhouse is simple but you still must learn it to use it efficiently
 - SQL is not standard , Materialized View/State concept are not Natural if you know other DB
 - Arrays bring a lot of added value , but you need to spend a little time on them
 - Primary key is “key” for select but also for compression
 - To keep a statistically relevant result , sampling should be done in a uniform way over the distribution of data. It’s more complex that what you may think
- The first bottleneck we did face was the number of files (and the inability of certain FS to handle millions of nodes)
 - Be careful with the partitioning and number of columns to avoid reaching FS limitation
- There are still a few bugs 😊
 - So Test and Try
 - Example : jdbc driver was not supporting request of size > 16KB (now fixed of course 😊) .

Quick example of clickhouse usage

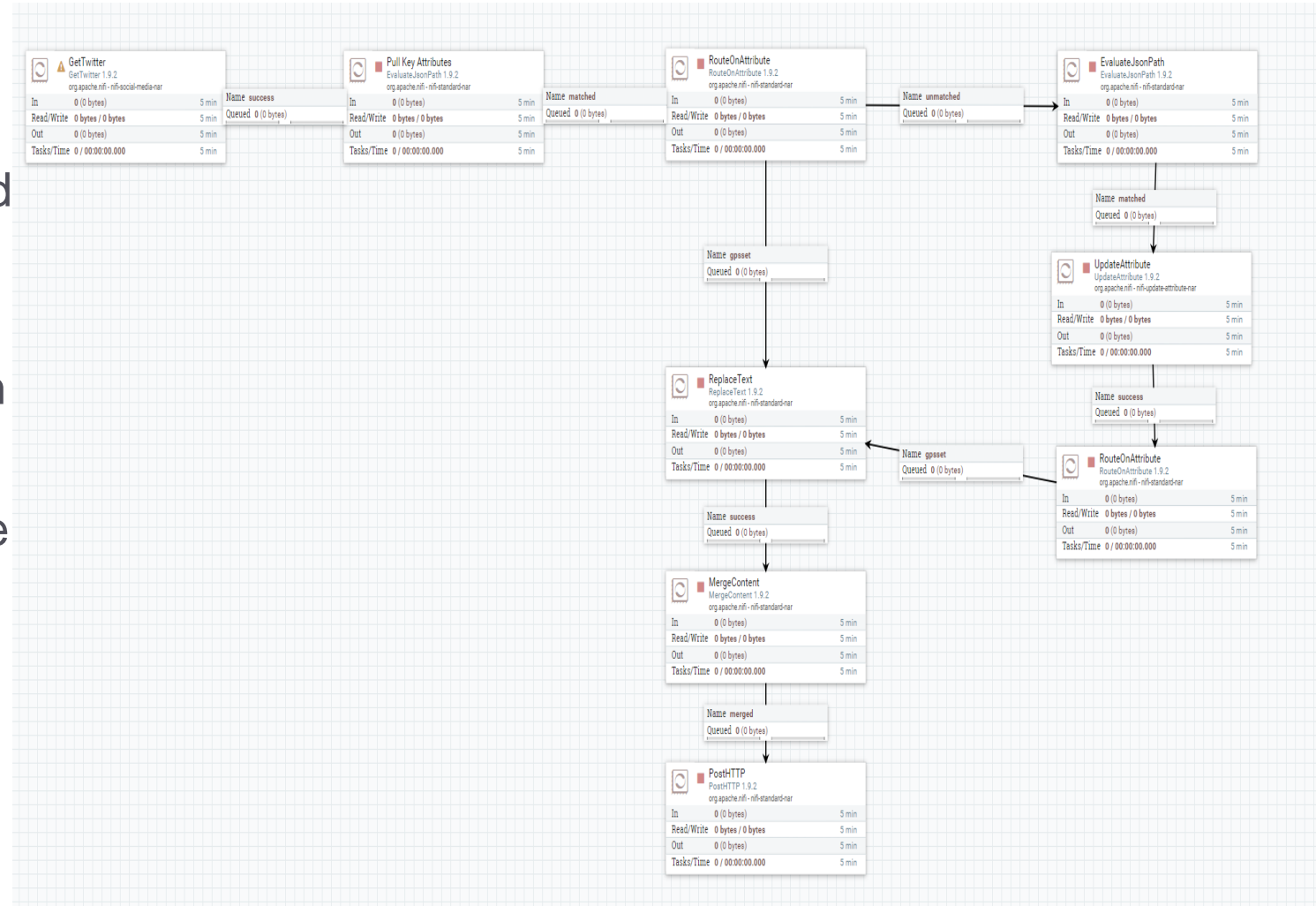
- We want to collect crowd sourcing data to build population density maps
 - We will use twitter to demonstrate the concept
 - The tweets you have in an area are generally representative of density of population using their mobile phone in this area
 - We need to
 - Collect geolocated events
 - Extract location
 - Store
 - Then display those data using a heatmap

Solution



Collect and extract GPS coordinate

- Use Nifi to collect tweets and extract location
 - Nifi is a toolbox allowing to read and write from hundred of data sources
 - It allow to extract and transform the data in the proper format
 - It scale horizontally and provide the right mechanism in term of security and lineage
 - Additionally it's visual development : no code



Pushing data to clickhouse from NiFi

First Clickhouse is better with large batch

- Use buffering capability of Nifi to group write based on size and time. Use the MergeContent Processor

Then push the data

- Simple postHTTP
 - Allow to insert type difficult to implement in jdbc (array , nested value) by using format supported by clickhouse (CSV/TSV....)
 - The only complexity here is the url (to have the right escaping)
- Use the jdbc driver (but require a bit more knowledge – Record processing in Nifi)

Nifi : Visual Development

Configure Processor

SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field

Property	Value
Merge Strategy	Bin-Packing Algorithm
Merge Format	Binary Concatenation
Attribute Strategy	Keep Only Common Attributes
Correlation Attribute Name	No value set
Metadata Strategy	Do Not Merge Uncommon Metadata
Minimum Number of Entries	20000
Maximum Number of Entries	100000
Minimum Group Size	0 B
Maximum Group Size	No value set
Max Bin Age	200 secs
Maximum number of Bins	5
Delimiter Strategy	Text
Header	No value set
Footer	No value set

CANCEL APPLY

Configure Processor

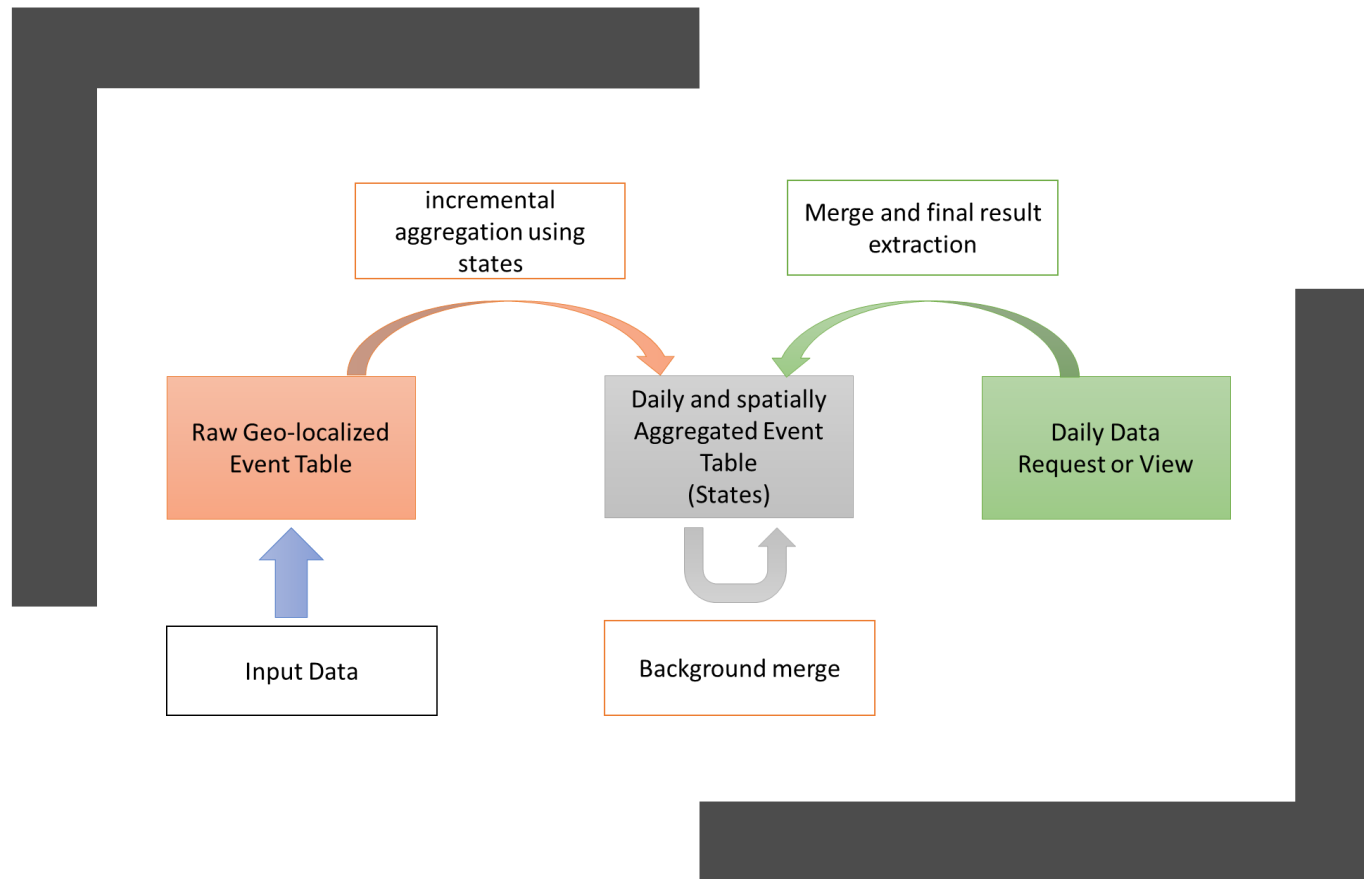
SETTINGS SCHEDULING **PROPERTIES** COMMENTS

Required field

Property	Value
URL	http://10.148.27.22:8123/?query=INSERT%20INTO%20de...
Max Batch Size	100 MB
Max Data to Post per Second	No value set
SSL Context Service	No value set
Username	No value set
Password	No value set
Send as FlowFile	false
Use Chunked Encoding	No value set
Compression Level	4
Connection Timeout	30 sec
Data Timeout	30 sec
Attributes to Send as HTTP Headers (Regex)	No value set
User Agent	Apache-HttpClient/4.5.5 (Java/1.8.0_171)
Proxy Configuration Service	No value set

CANCEL APPLY

Store the data in clickhouse



- Store the raw data into a dedicated table
 - Use a Materialized view to aggregate the number of tweet per coordinate
 - Use the twitter id to count the number of unique user
 - Ideally use a bin map to aggregate on coordinate (but not done yet)

Visualise the data using Tableau or Grafana

- Tableau :
 - Draw yourself the polygon in the map
 - Use arrays to compute the polygon and its path to be drawn by tableau
 - Use the new heatmap visualization
 - On raw data directly , but this will suffer of scalability issues
 - On aggregated data , but as heatmaps are based on the number of point , you will need to convert the count in a number of line

```
select
  arrayJoin(
    [(0,0,1),(0.01,0,2),(0.01,0.01,3),(0,0.01,4)] as pt
  ) as adj,
  longitude + tupleElement(adj, 1) as long,
  latitude + tupleElement(adj, 2) as lat,
  tupleElement(adj, 3) as path,
  uniquenessr,
  concat(toString(longitude), ',', toString(latitude)) as key
from
  demo.daily_res100
```

```
select
  arrayJoin(arrayResize(emptyArrayUInt8(), uniquenessr)) as adj,
  longitude,
  latitude,
  id
from
  demo.daily_res100
```

Et voilà

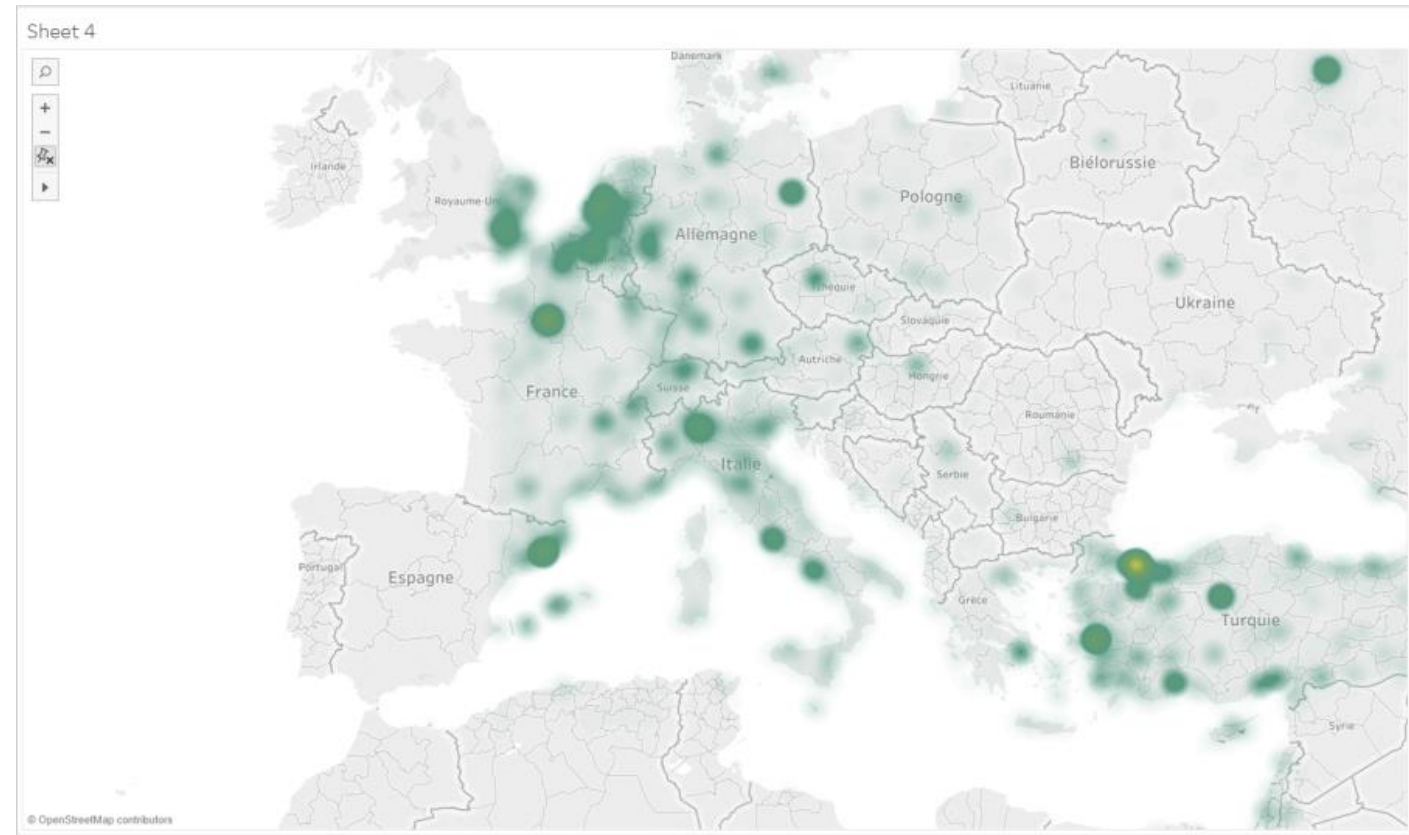
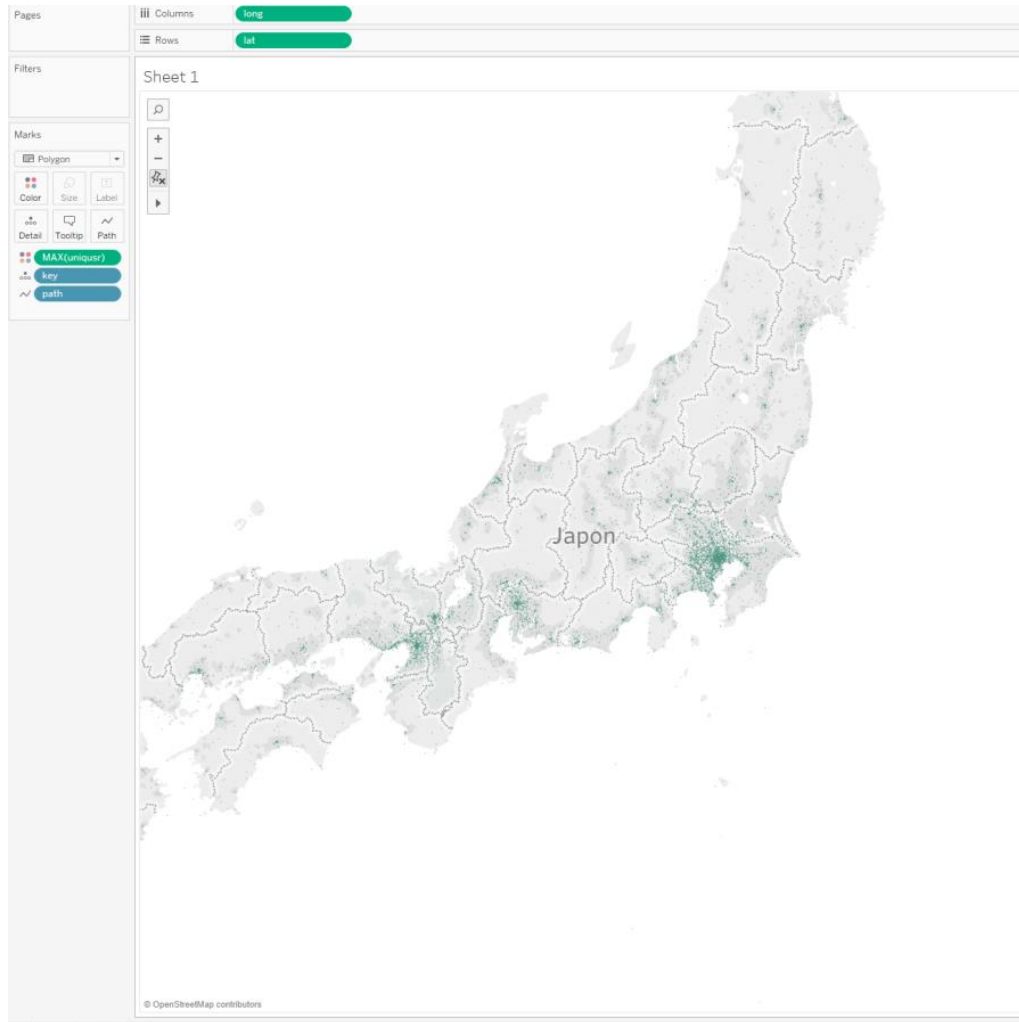
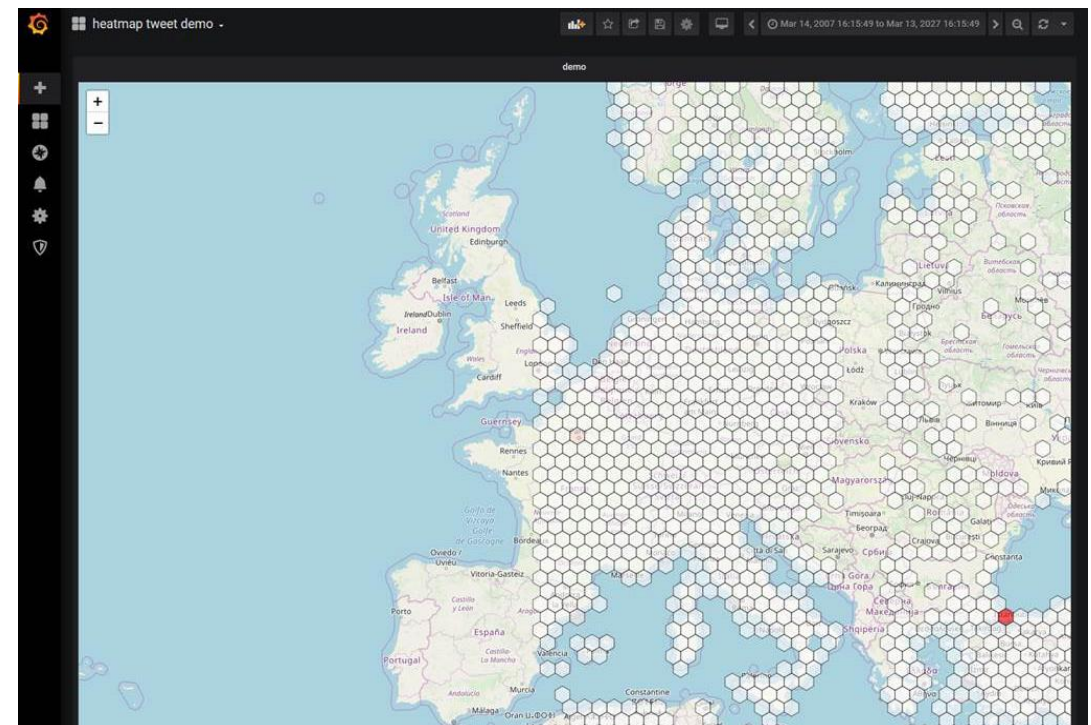
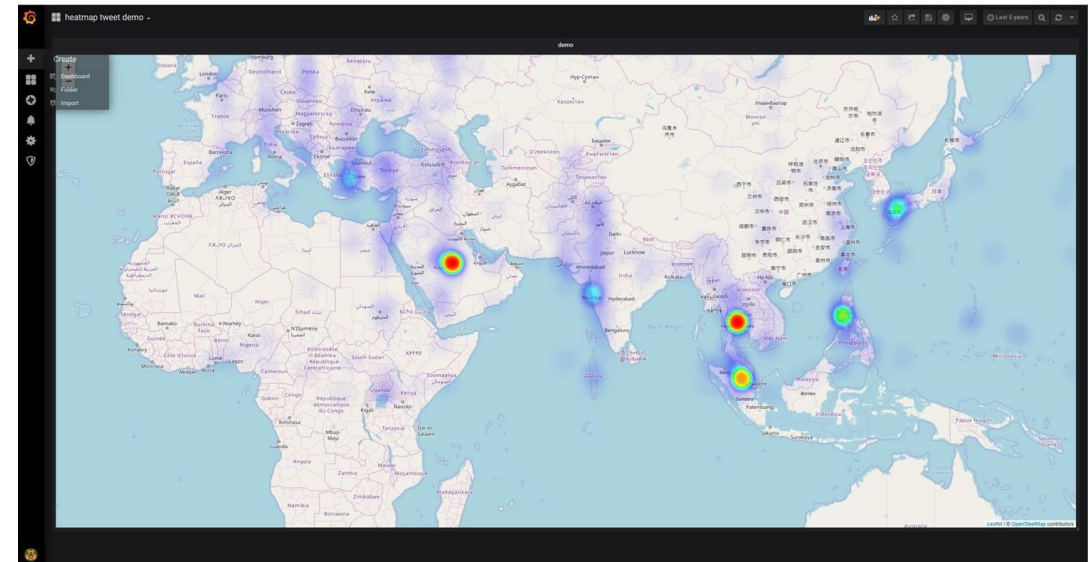


Tableau Free Alternative : Grafana

- There is a very nice plugin call TrackMap (by Alexandra)
 - Badly named because you can also display heatmap / hexbin
 - Works fine with clickhouse
 - Apply the same mechanism as the one for tableau to generate all points on aggregated values or use the raw data
 - It even has an advantage compare to tableau : in tableau you cannot filter on visible map bounding box, in this plugin it was added a week ago



Conclusion



We need a multi domain analytical platform which is fast and scale horizontally



This platform need to be integrated in a larger ecosystem



Clickhouse is the right answer for us

Speed is incredible

Features are rich

Integration is easy

Operations are not a burden (And this will be improved on cluster with the clickhouse operator thanks to Altinity)

Roadmap is aligned with our needs and we can contribute

Just one word for the Clickhouse TEAM

T H A N K Y O U

Thank you!

www.infovista.com

