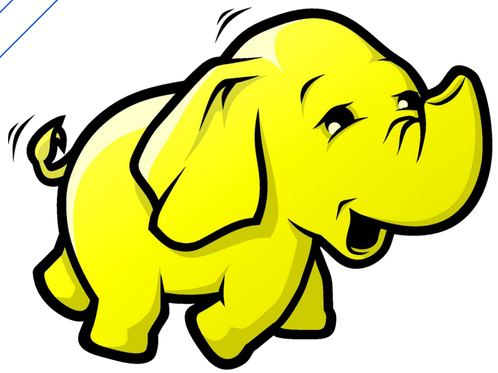
JASMINE

Batch Processing

Simone Mancini Francesco Ottaviano Andrea Silvi



JASMINE

Design Choices

Frameworks

Programming Languages

Store

APIs

JASMINE

Data Ingestion

- Handles files and puts them into *HDFS* as clean as possible.
- Guarantees simple batch processing.
- Uses *Redis* as separated and distributed cache (***DistributedCacheService***).
- Custom processors

JASMINE

ReformatCSVProcessor

- Converts the original "*datetime-based*" csv files into "*city-based*" csv.
- PROs: Simplifies forward steps; fixed header.
- CONs: Introduces data redundancy and increases files size.

datetime,Portland,San Francisco,Seattle
2012-10-01 13:00:00,282.96,,300009

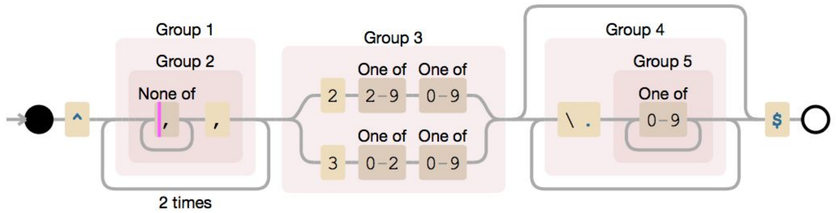
datetime,city,value
2012-10-01 13:00:00,Portland,282.96
2012-10-01 13:00:00,San Francisco,
2012-10-01 13:00:00,Seattle,300009

JASMINE

FileFilterProcessor - 1

- Filters csv rows according to the given commands list property.

Temperatures *regex* flow



- Command format:

  'filename<-regex<-action-value(-type)<-[position,position,...]'

- Example:

  "temperature.csv<-^(([^,]+,)){2}(2[2-9][0-9]|3[0-2][0-9])(\\.([0-9]+))$<-numeric-1000.0-divide<-[2]"*

datetime,city,value
2012-10-01 13:00:00,Portland,282.96
2012-10-01 13:00:00,San Francisco,
2012-10-01 13:00:00,Seattle,300009

datetime,city,value
2012-10-01 13:00:00,Portland,282.96
2012-10-01 13:00:00,Seattle,300.009

JASMINE

FileFilterProcessor - 2

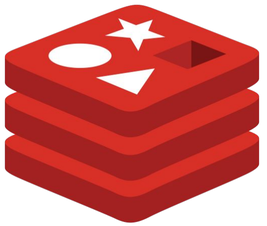
```
city_attribute.csv<-^(([^,]+,)){2}[^,]+$<-continue-0<-[]
-and-
weather_description.csv<-^(([^,]+,)){2}[^,]+$<-continue-0<-[]
-and-
pressure.csv<-^(([^,]+,)){2}(7[7-9][0-9]|[8-9][0-9]{2}|1[0-2][0-9]{2})(\.([0-9]+))$<-continue-0<-[]
-and-
humidity.csv<-^(([^,]+,)){2}((100(\.)0+)|((|[0-9]|[1-9][0-9])(\.([0-9]+))))$<-continue-0<-[]
-and-
temperature.csv<-^(([^,]+,)){2}(2[2-9][0-9]|3[0-2][0-9])(\.([0-9]+))*$<-numeric-1000.0-divide<-[2]
```

☐ Set empty string

CANCEL OK

ReverseGeocodingProcessor

- Retrieves countries and timezones information.
- Different providers (*GeoNames* and *GoogleAPI* already available).
- *Redis* as cache.



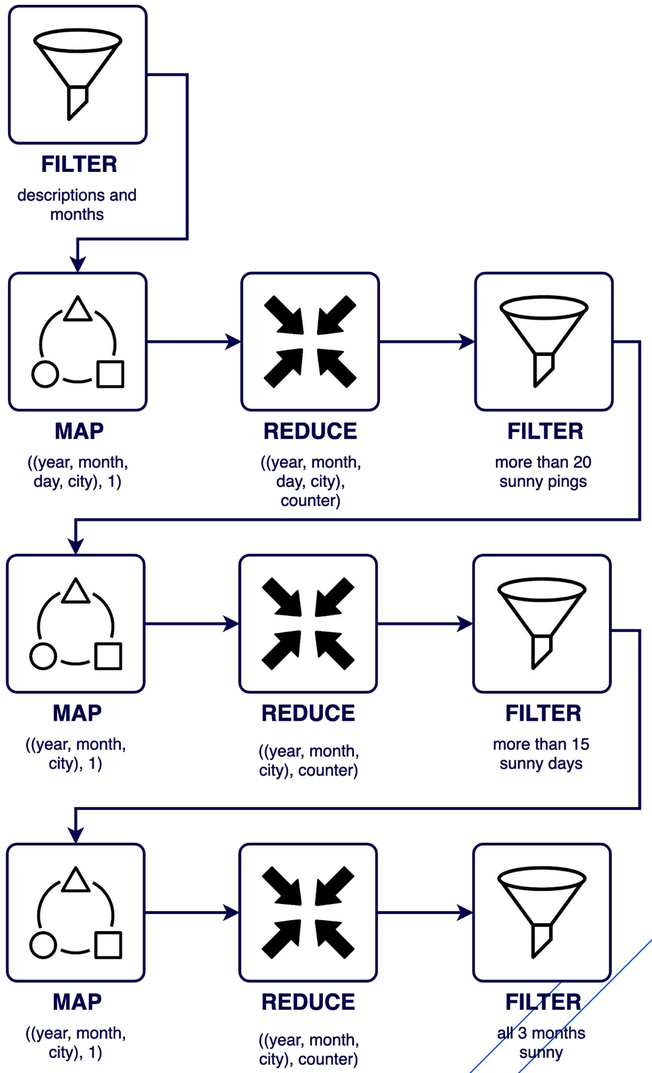
city,latitude,longitude
Los Angeles,34.052231,-118.243683
Jerusalem,31.769039,35.216331

city,latitude,longitude,country,timezone
Los Angeles,34.052231,-118.243683,United States,America/Los_Angeles
Jerusalem,31.769039,35.216331,Israel,Asia/Jerusalem

Queries

JASMINE

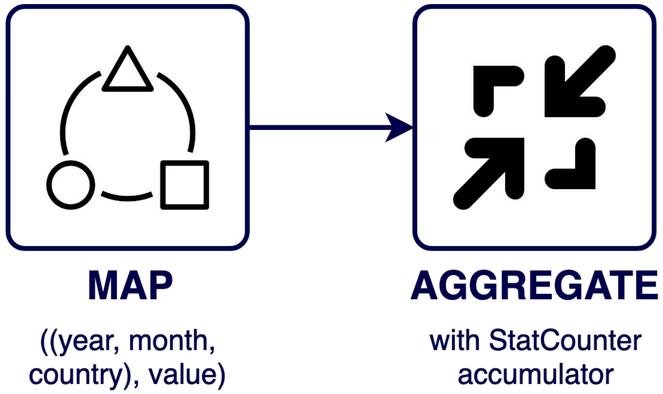
Query 1



- Identify cities with clean sky, for at least 15 days a month in March, April and May.
- It has been established a day to be clear if it is “*clear*” for at least 20 pings a day.
- 3 incremental MAP - REDUCE - FILTER sequences.

JASMINE

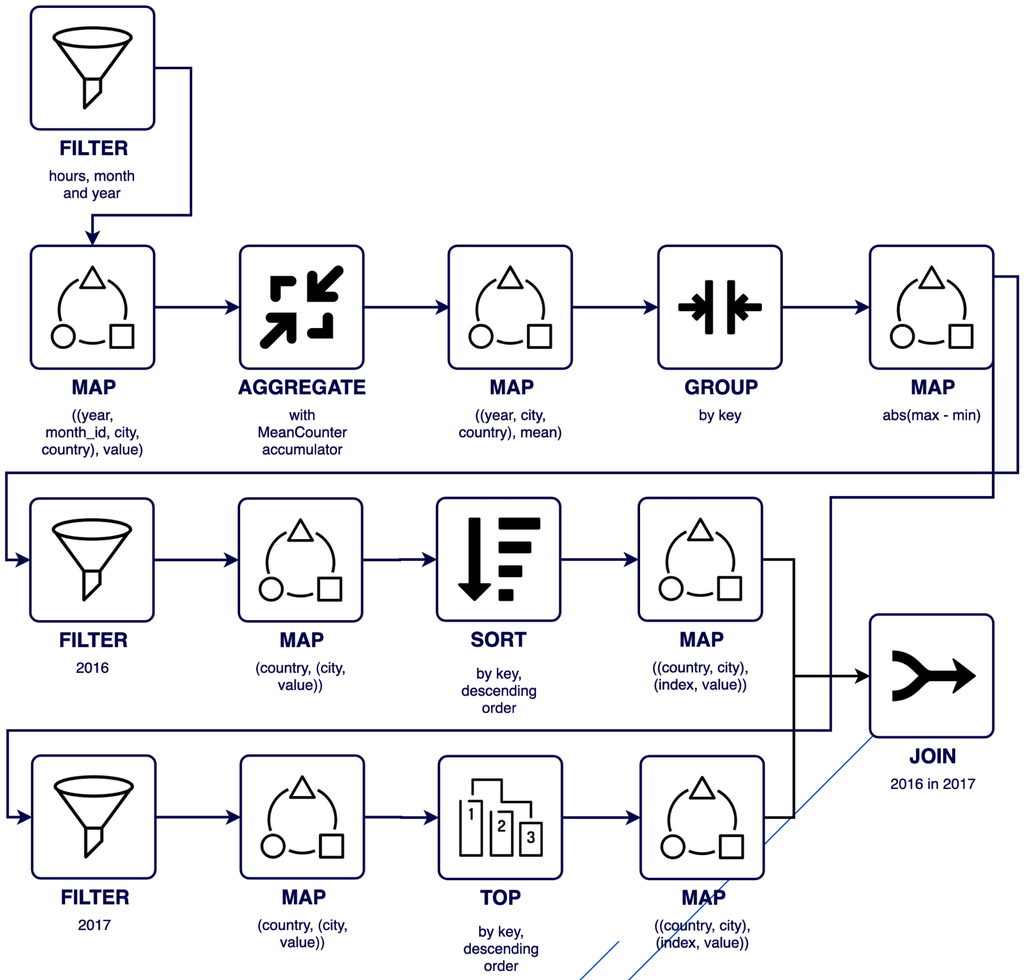
Query 2



- Statistical calculations on measurements: mean, standard deviation, maximum, minimum.
- Overflow safe and optimized computation for mean and std.
- Use of aggregation operator for smart *StatCounter* usage.

JASMINE

Query 3



- Show the three different cities that have experienced the higher temperature difference between 12:00 and 15:00 in two months groups.
- Computation of each months groups averages differences.
- Split and join for smart top 3 leaderboard computation.

JASMINE

SQL

Aligned logic between Spark Core and Spark SQL

Plain text SQL queries

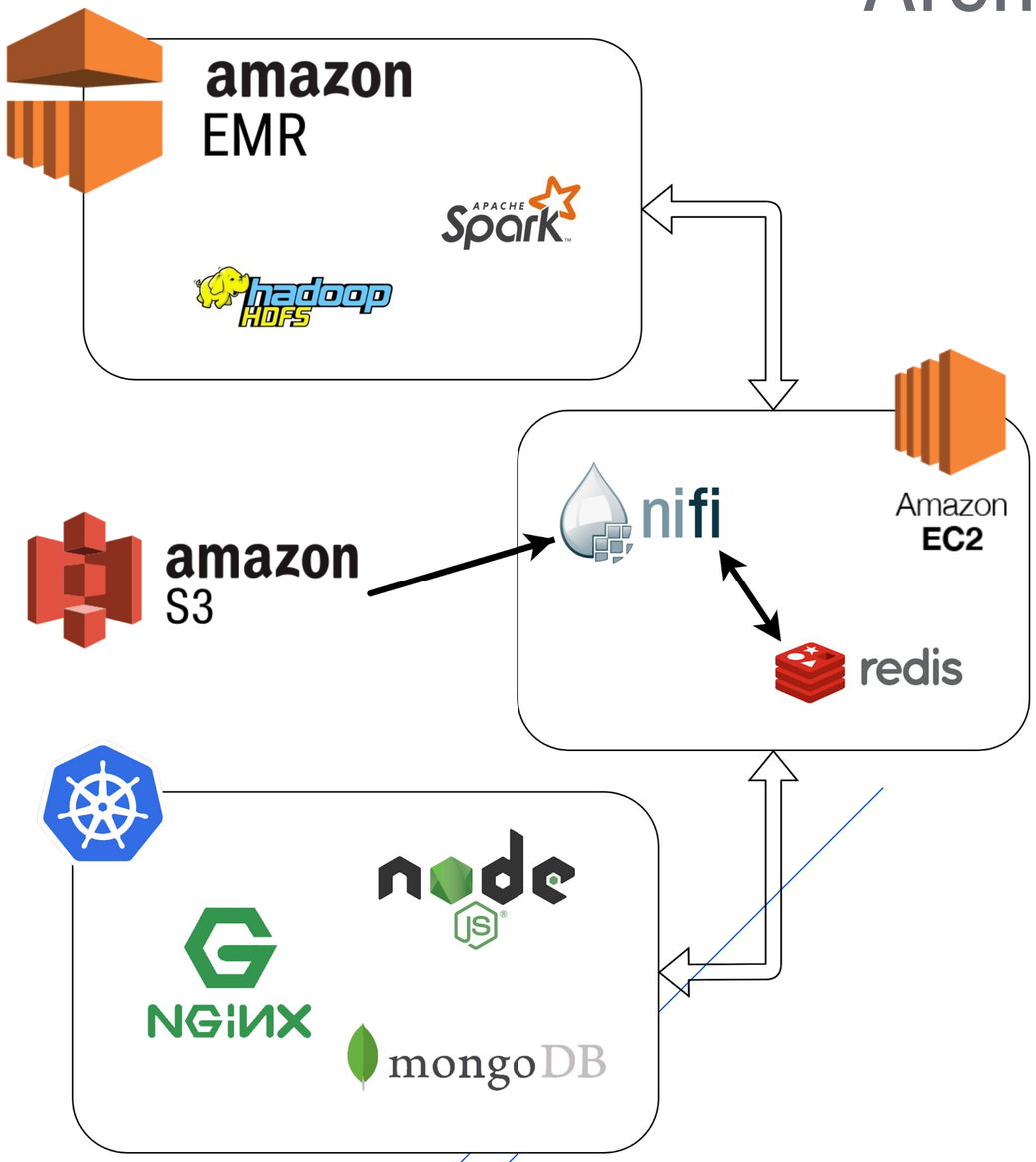
"*Builder*" code design pattern

Easier updates and maintenance

Architecture

JASMINE

Architecture

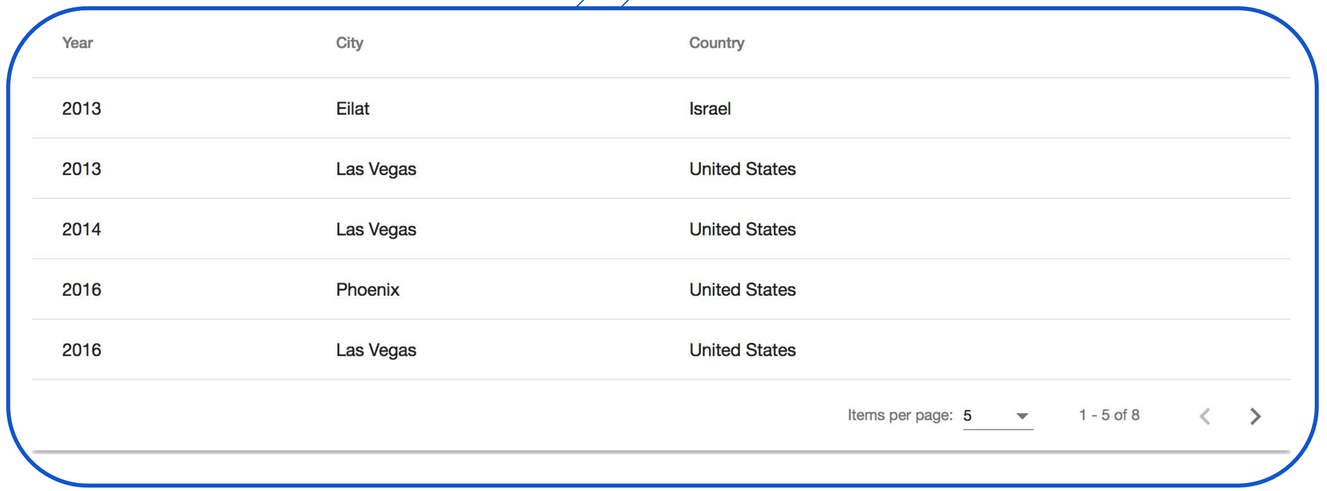


- *Cluster EMR*
  - 1 Master, 2 Nodes (m3.xlarge, 8 vCore, 15 GiB memory, 80 SSD GB storage).
  - *Spark* and *Hadoop*.
- *EC2 instance*
  - m4.large, 2 CPUs, 8 GiB memory, 2,4 GHz Intel Xeon E5-2676 v3.
  - *NiFi* and *Redis*.
- *AWS S3*
  - Configuration files and dataset
- *Kubernetes*
  - 1 Master, 2 Nodes (t2.micro).
  - *MongoDB*, *NodeJS*, *Nginx*.

JASMINE

Outputs

- Results are stored in *HDFS* (*JSON* format).
- *NiFi* fetches files (regex check: *^part-([0-9]+)$*) and routes outputs to *MongoDB*'s appropriate collection.
- Web UI available at www.simonemancini.eu:32080.



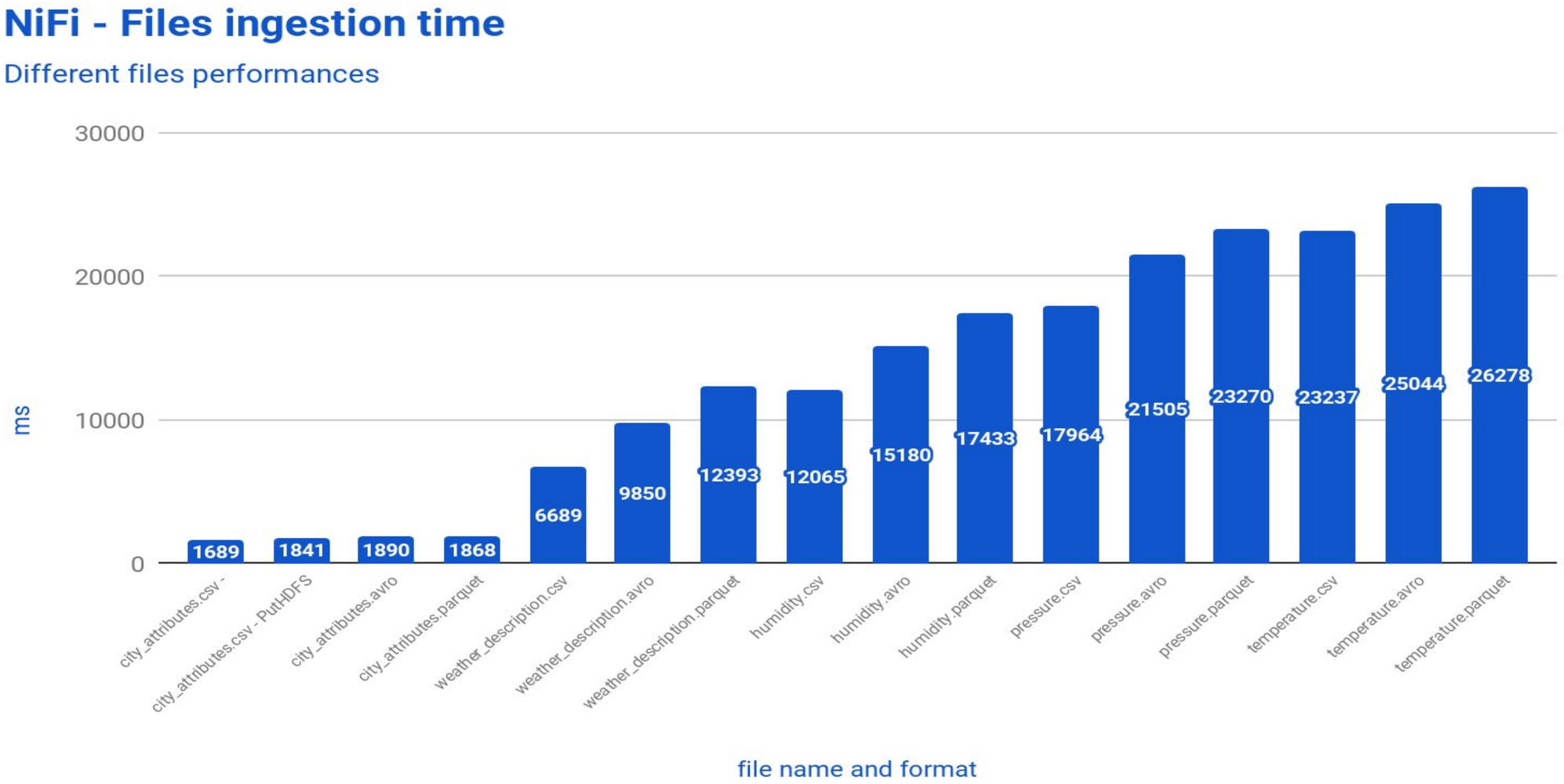
Year	City	Country
2013	Eilat	Israel
2013	Las Vegas	United States
2014	Las Vegas	United States
2016	Phoenix	United States
2016	Las Vegas	United States

Items per page: 5 1 - 5 of 8

Performances

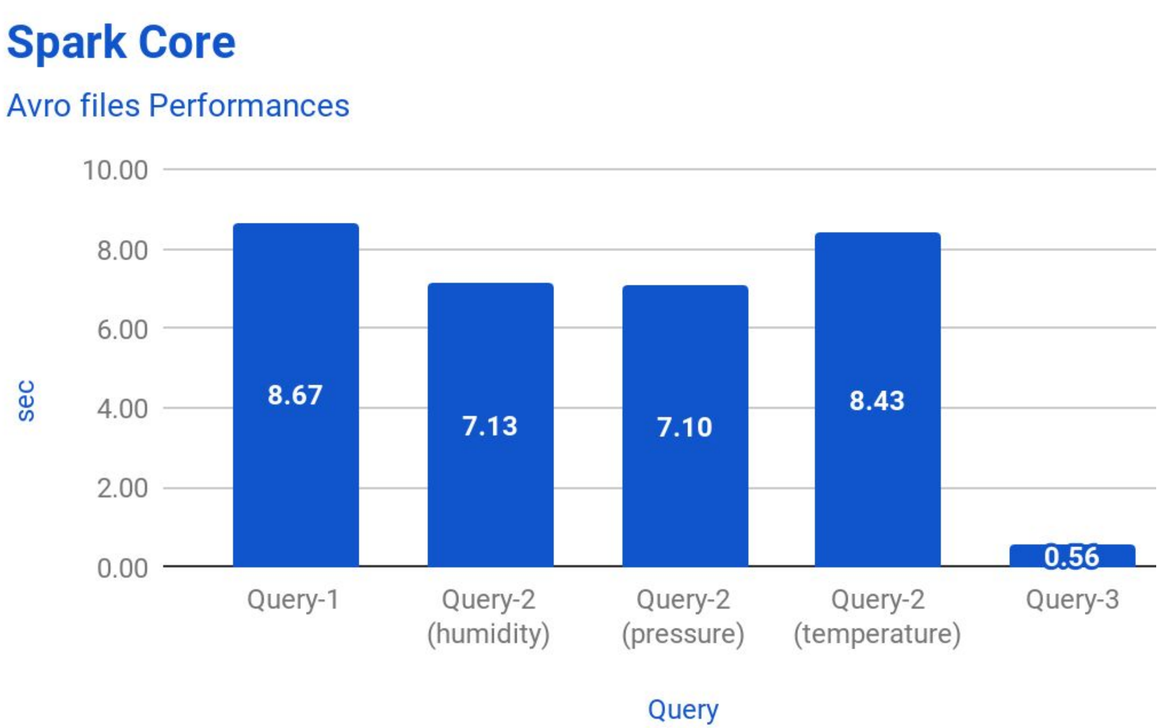
JASMINE

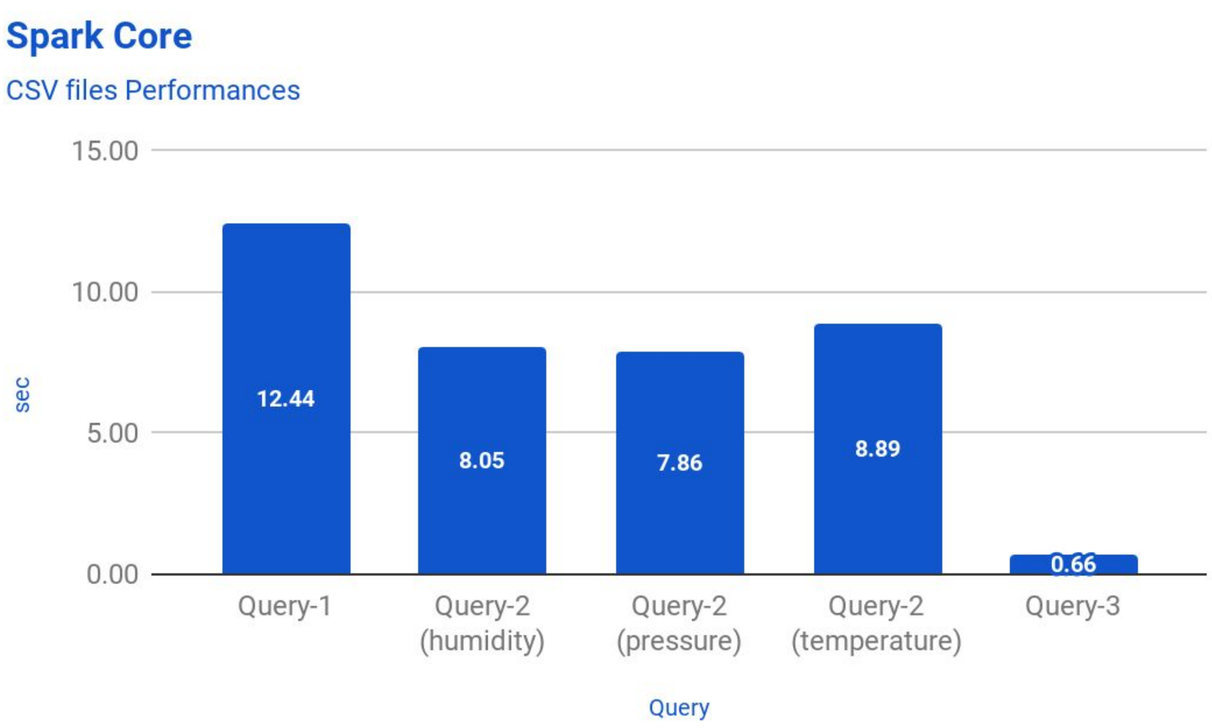
NiFi Evaluation



JASMINE

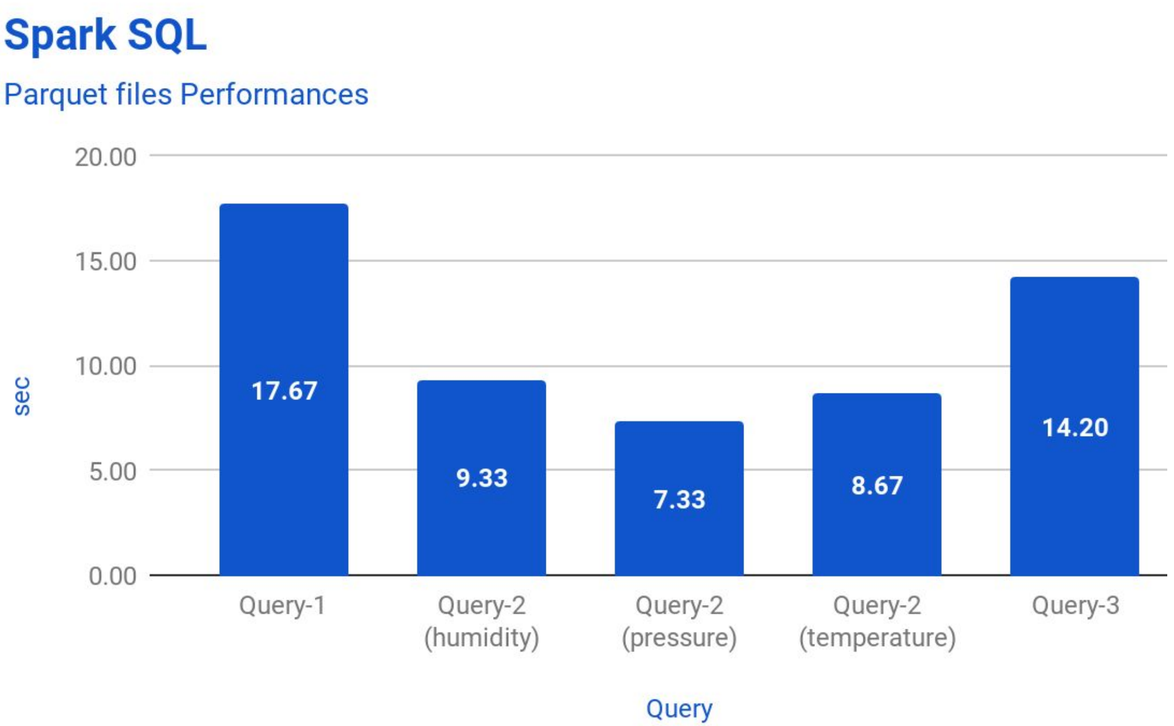
Spark-Core Evaluation

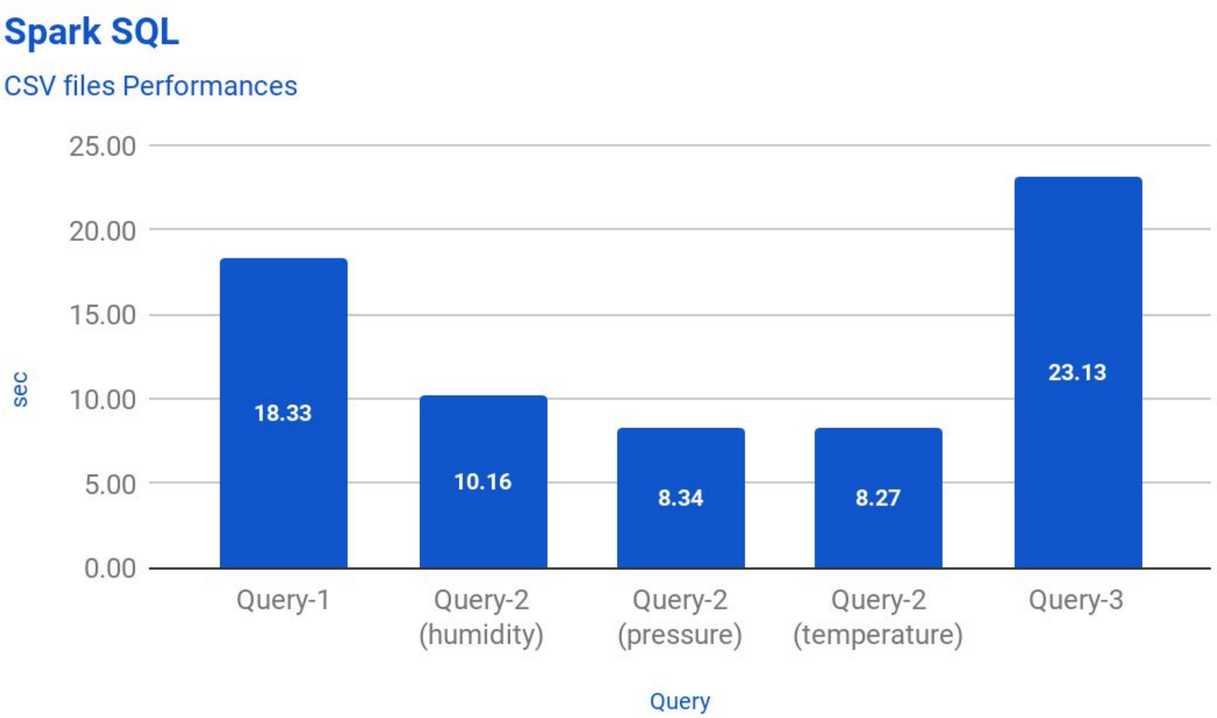




JASMINE

Spark-SQL Evaluation



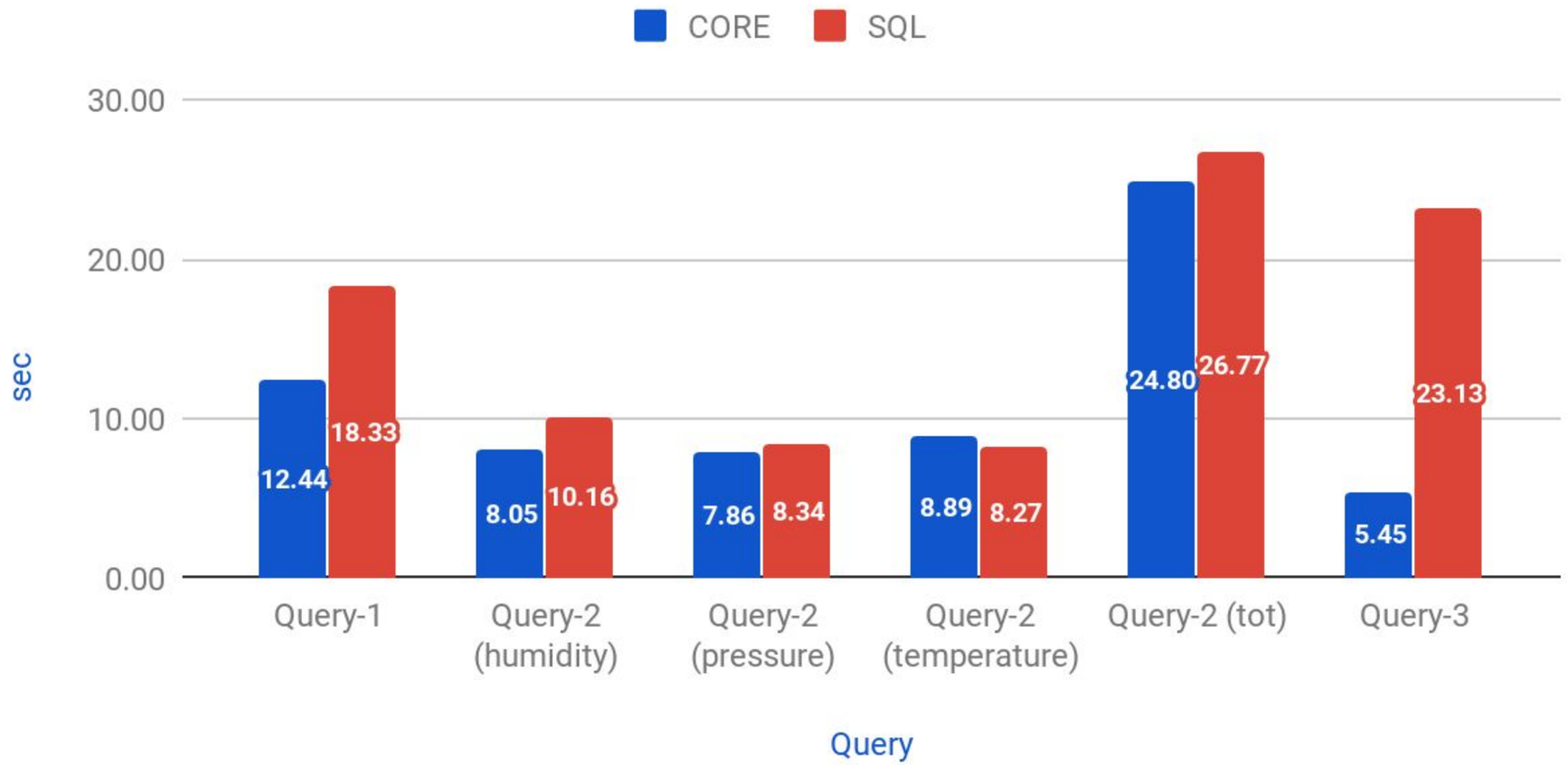


JASMINE

Core-SQL Comparison

Comparison without cache

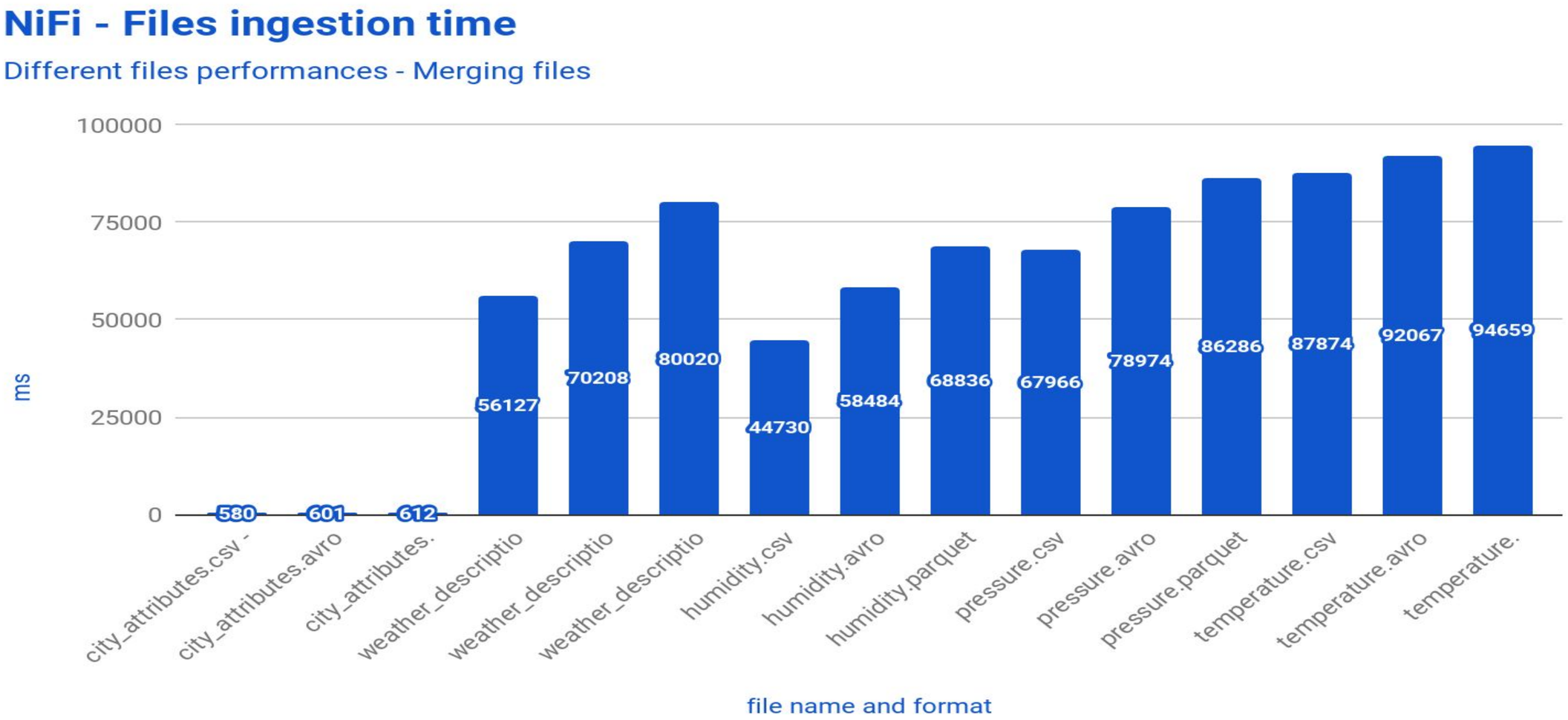
CSV format



JASMINE

NiFi Evaluation

Alternative scenario



Improvements

- *NiFi* and *Redis* splitting and clusterization (Cluster Manager usage).
- *MongoDB* as second level cities cache.
- Weather description strings classification methods.
- Improve system security introducing (e.g. users and authentication in RestAPI component).

JASMINE

Thank You!