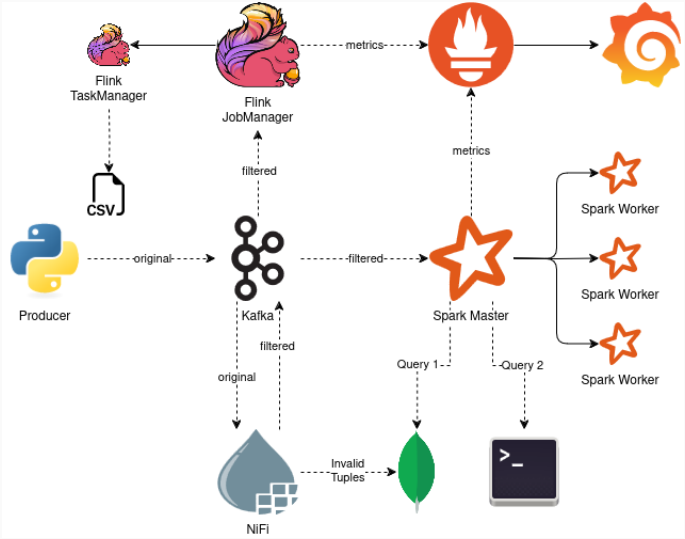
Analisi di dati di monitoraggio con Apache Spark

Sistemi e Architetture per Big Data - Progetto 2

Alessandro Lioi, 0333693

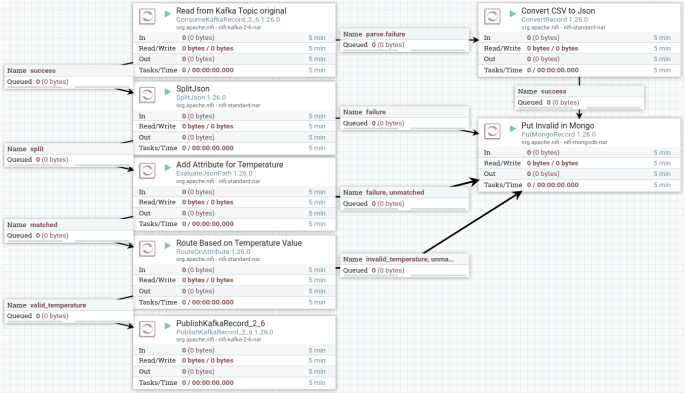
Architettura



Componenti:

- Producer: replay del dataset
- NiFi: pre-processing
- Kafka: data ingestion
- Flink/Spark: stream processing
- MongoDB: data store NoSQL
- Prometheus: aggregatore metriche
- Grafana: visualizzazione

Producer e Apache NiFi



Producer

- inizializza NiFi, tramite template
- esegue replay del dataset, scrivendo su Kafka

NiFi

- legge da Kafka
- scrive su Mongo tuple non valide
- scrive su Kafka tuple valide

Query 1 - Flink

Tupla Originale:

date_ts	**serial_number**	**model**	**failure**	**vault_id**	**temperature**

Tuple Risultante:

vault_id in $[1000, 1020]$	**temperature**

- filtro sui `vault_id` in $[1000, 1020]$
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`: implementazione di Welford
- `map` nella stringa di output

Query 1 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

key	value
vault_id in [1000, 1020]	temperature

- filtro sui vault_id in [1000, 1020]
- keyBy su vault_id
- window di 1, 3 e 23 giorni
- aggregate: implementazione di Welford
- map nella stringa di output

Query 1 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

key	aggregate		
`vault_id` in $[1000, 1020]$	`count`	`mean`	`stddev`

- filtro sui `vault_id` in $[1000, 1020]$
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`: implementazione di Welford
- `map` nella stringa di output

Query 1 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

`window.start,vault_id,count_s194,mean_s194,stddev_s194`

- filtro sui `vault_id` in $[1000, 1020]$
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`: implementazione di Welford
- `map` nella stringa di output

Query 2 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

vault_id	failure > 0	serial_number	model

- filtro su `failure` > 0
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`:
  - si sommano le `failure`
  - si crea la lista di HDD
- `windowAll` di 1, 3 e 23 giorni
- `process`:
  - si identifica la classifica
  - si converte nella stringa in output

Query 2 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

key	value		
vault_id	failure > 0	serial_number	model

- filtro su `failure` > 0
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- aggregate:
  - si sommano le `failure`
  - si crea la lista di HDD
- `windowAll` di 1, 3 e 23 giorni
- process:
  - si identifica la classifica
  - si converte nella stringa in output

Query 2 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

key	value	
vault_id	failures_count	set(serial_number, model)

- filtro su `failure` > 0
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`:
  - si sommano le `failure`
  - si crea la lista di HDD
- `windowAll` di 1, 3 e 23 giorni
- `process`:
  - si identifica la classifica
  - si converte nella stringa in output

Query 2 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

key	value	
vault_id	failures_count ↓	set(serial_number, model)

- filtro su `failure` > 0
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`:
  - si sommano le `failure`
  - si crea la lista di HDD
- `windowAll` di 1, 3 e 23 giorni
- `process`:
  - si identifica la classifica
  - si converte nella stringa in output

Query 2 - Flink

Tupla Originale:

date_ts	serial_number	model	failure	vault_id	temperature

Tuple Risultante:

`ts,vault_id,set(serial_number, model),...`

- filtro su `failure` > 0
- `keyBy` su `vault_id`
- `window` di 1, 3 e 23 giorni
- `aggregate`:
  - si sommano le `failure`
  - si crea la lista di HDD
- `windowAll` di 1, 3 e 23 giorni
- `process`:
  - si identifica la classifica
  - si converte nella stringa in output

Query 1

- si filtrano i vault con ID in $[1000, 1020]$
- si selezionano le colonne di interesse: `date_ts`, `vault_id`, `temperature`
- si raggruppa per:
  - finestra di 1, 3 e 23 giorni
  - `vault_id`
- si aggrega calcolando `count`, `mean`, `stddev`

Query 2

- si filtrano i vault con failure > 0
- si selezionano le colonne di interesse: `date_ts`, `vault_id`, `failure`, `serial_number` e `model`
- si raggruppa per:
  - finestra di 1, 3 e 23 giorni
  - `vault_id`
- si aggrega con `count` e `collect_set`
- si ordina per `failures` decrescenti e si limitano i valori a 10
- si raggruppa per timestamp
- si aggrega con `collect_list` di `vault_id` `failures` e `hdds`

Metriche - Flink

Calcolate tramite metriche custom

Query 1

Query 2

Metriche - Spark Structured Streaming

Si utilizzano la latenza e il processing rate fornito da Spark

Query 1

Query 2

Grazie Per l'Attenzione!

Il codice è presente su GitHub al seguente link:
https://github.com/lioia/sabd_project_2