Euro 2020 Lambda Architecture

Per un pugno di dati

Alexandru Pavel, Antonio Turco

Introduzione e Obiettivi

- Ambito Euro 2020
- Integrare diverse fonti di dati
- Definire una pipeline per l’elaborazione dei dati
- Gestire il tutto in maniera scalabile e fault tolerant

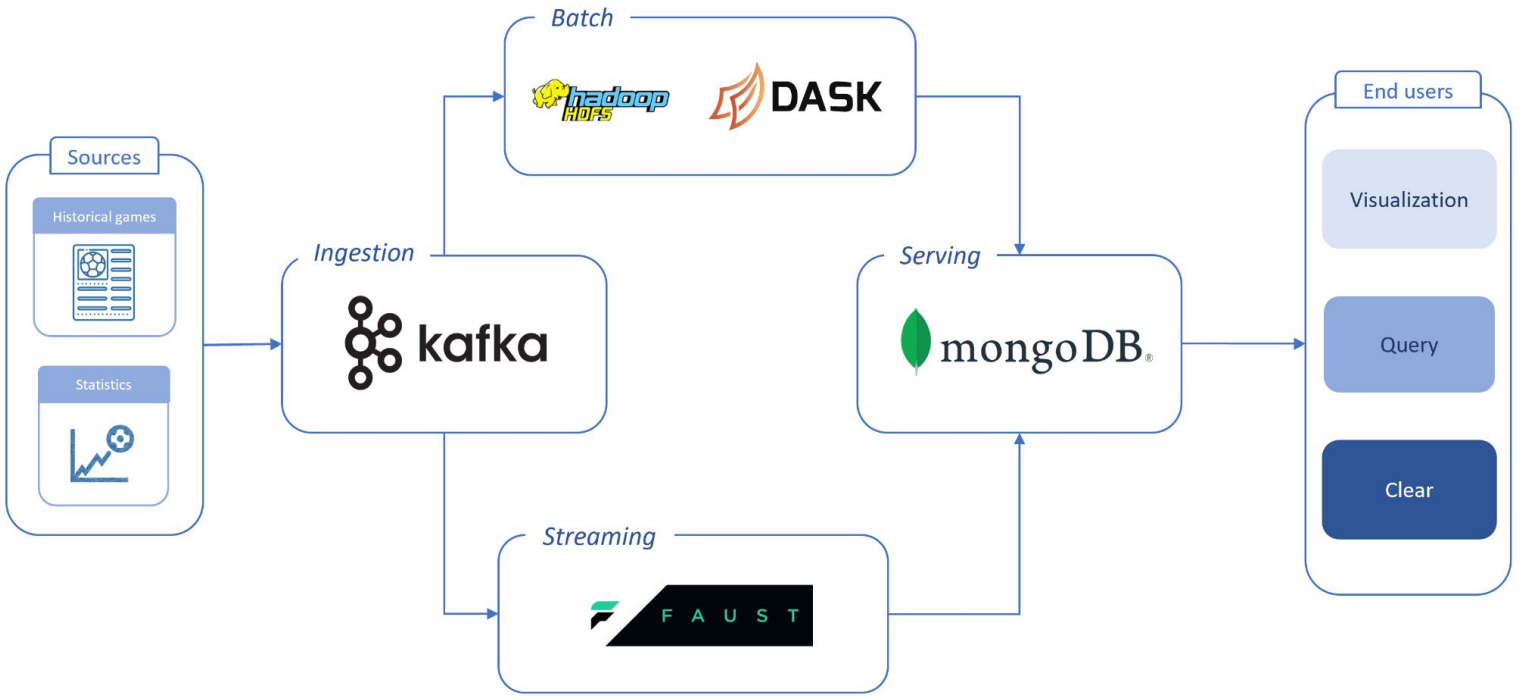
Dataset

- Api-Football
  - Limite di 90 chiamate al giorno
  - Estratte in totale 5 partite complete
  - Trasformate in statistiche istantanee tramite preprocessing

Minuto	1	2	3	4	5
Originale	0	10	50	100	120
Preprocessed	0	10	40	50	20

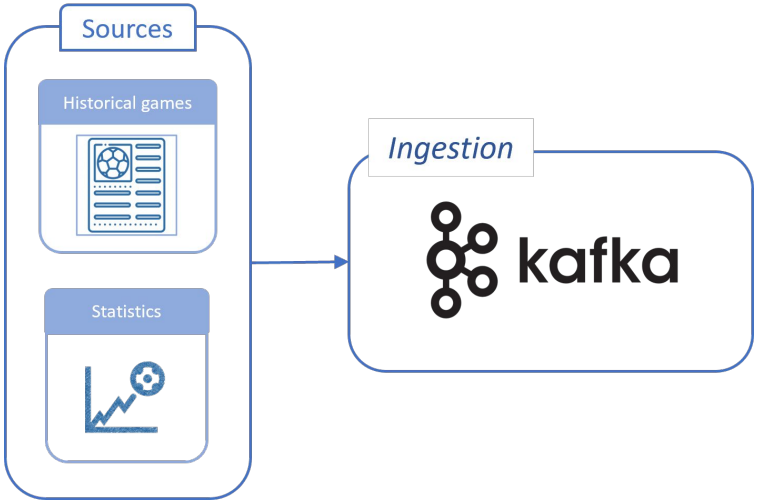
- Historical Dataset
  - Dataset scaricato da Kaggle
  - Risultati partite internazionali dal 1870 fino ad oggi

Architettura



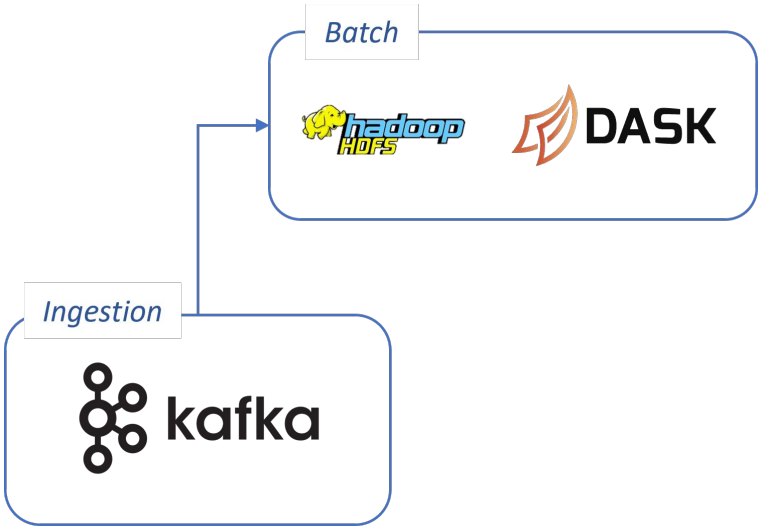
Ingestion

- Topic creati
  - Un topic per lo stream dello storico delle partite
  - Cinque topic con le statistiche in tempo reale delle partite
    - Uno per ogni partita
- Formato dati
  - JSON per le statistiche
    - Dizionario con chiave il tipo di statistica
  - CSV per le partite storiche
    - Ogni record rappresenta una partita



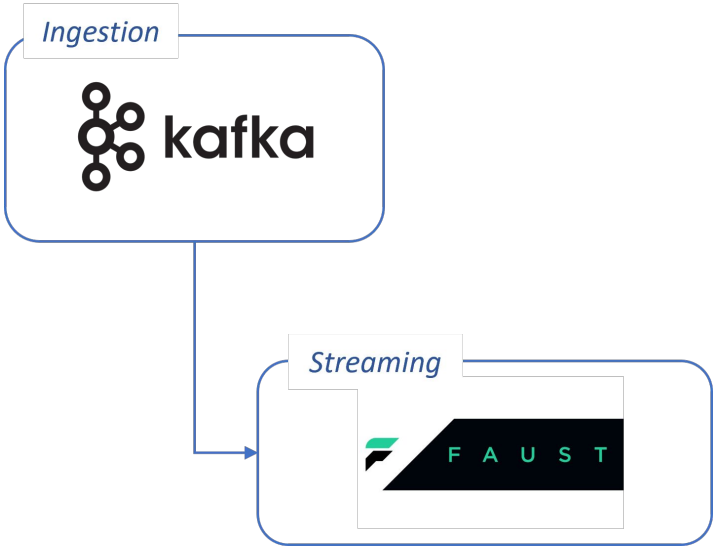
Batch Layer

- Master dataset **HDFS**
  - Dati ricevuti in streaming vengono inseriti in modalità append
  - In questa maniera è facile fare rollback in caso di dati corrotti/errati
- Batch Engine in **Dask**
  - Nativo in Python
  - Possibilità di usare librerie come Pandas, Numpy, etc.. in maniera distribuita
  - Dashboard per il profiling e il debugging
    - misura risorse assegnate ad ogni task del job in esecuzione



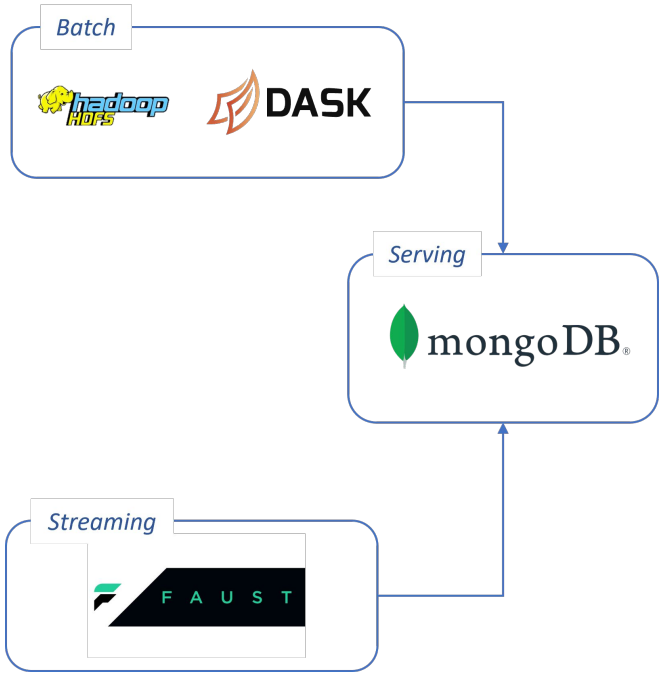
Streaming Layer

- Streaming Engine **Faust**
  - Nativo Python
  - Integrazione nativa con Kafka
- Table
  - Dizionario chiave/valore
  - Memorizza dati in maniera scalabile e distribuita
  - Basato su RocksDB
- Finestre
  - Finestra di un minuto
  - Finestra su tutta la partita

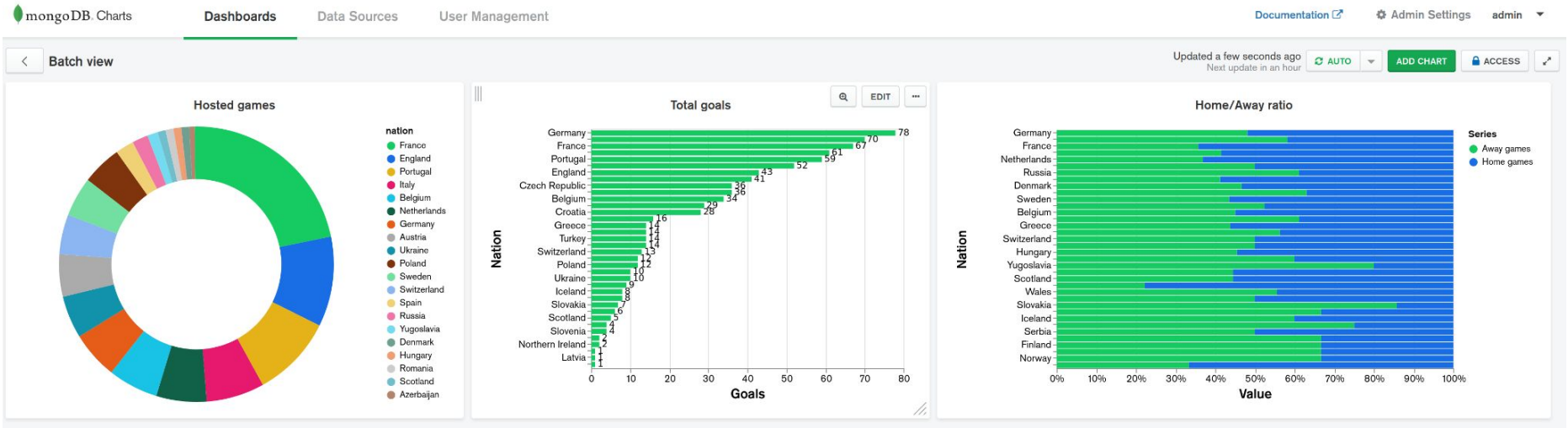


Serving Layer

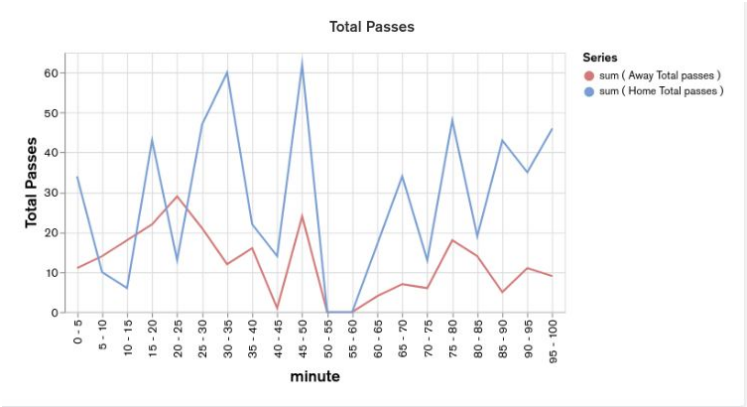
- Memorizzazione avviene su **MongoDB**
- Un database per ogni ramo in input
  - Batch_view
  - Streaming_view
- Dati letti automaticamente dalla Dashboard grazie a Mongo Charts
- Grazie a Mongo Charts non servono alte competenze tecniche per l'accesso ai report

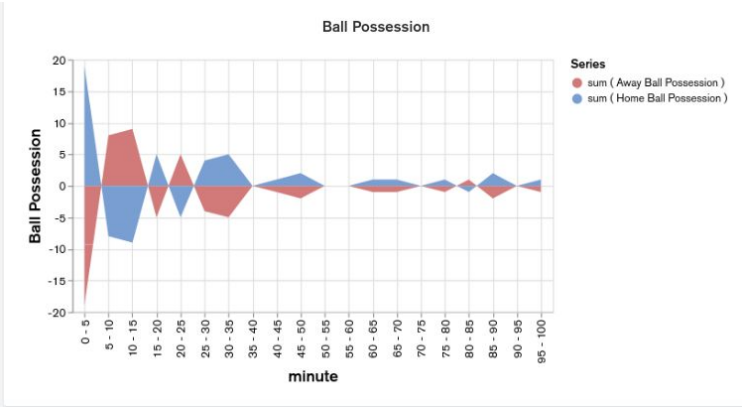


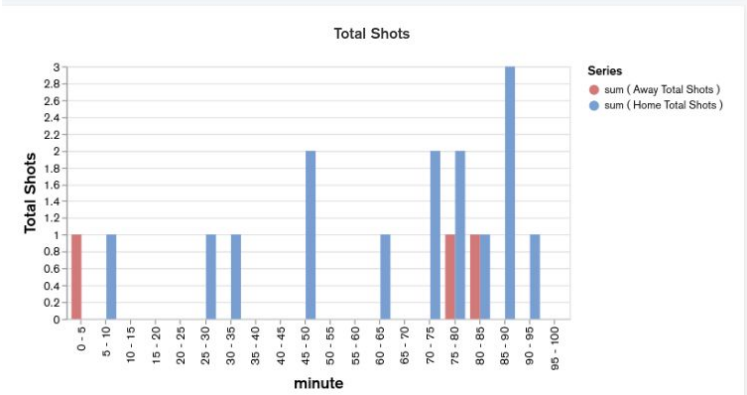
Dashboard Batch

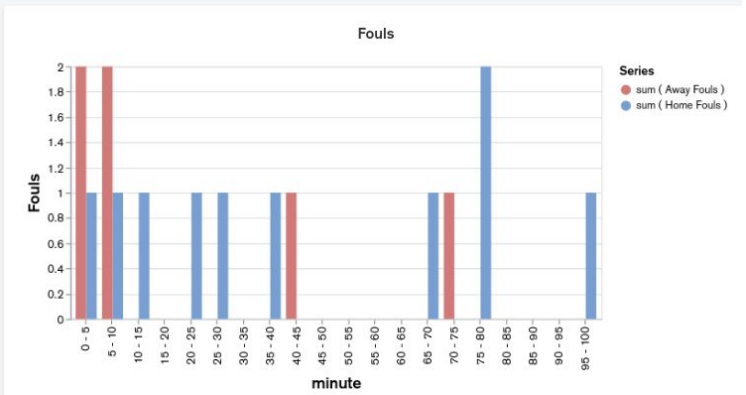


Dashboard Streaming





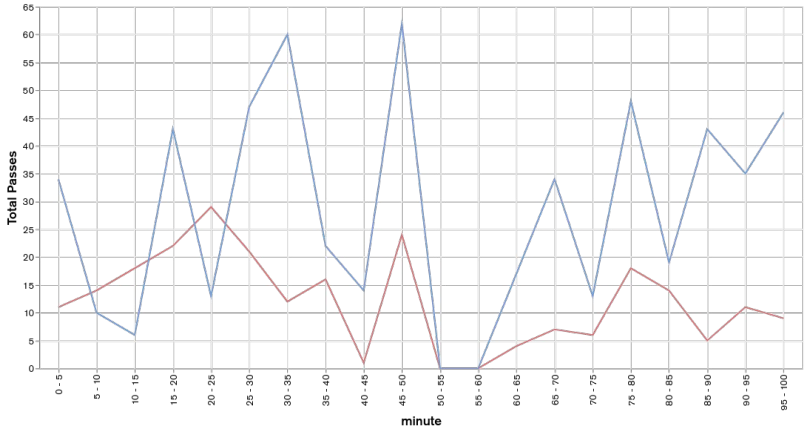


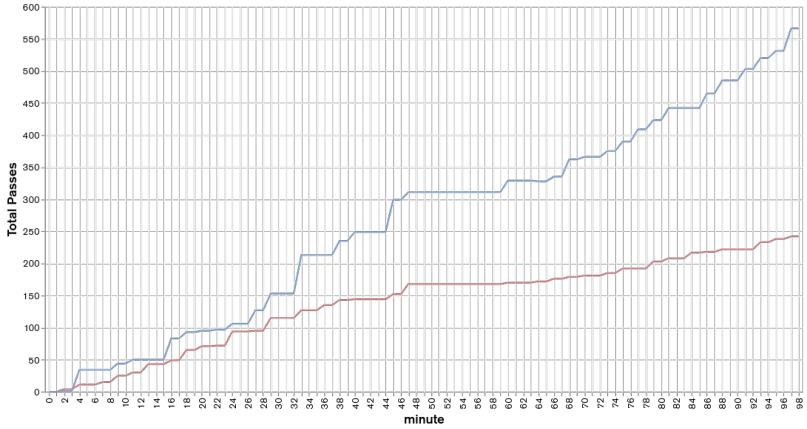


Tipologie di finestre

- Finestra di 1 minuto
  - Statistiche istantanee in ogni momento della partita
  - Aggregazione in gruppi da 5 minuti su Mongo Charts

- Finestra sull'intera partita
  - Statistiche aggregate su tutta la partita
  - Visione più generale dei trend





Conclusioni

- Definita architettura scalabile e fault tolerant
  - Sperimentando tecnologie alternative agli standard dell'industria
- Integrati dati da diverse fonti
- Esecuzione dei vari servizi avviene in maniera automatica con Docker

Sviluppi futuri

- Rendere l'architettura *"plug and play"*
  - Automatizzando ulteriormente la pipeline di esecuzione
- Gestire dati calcistici generici e non solo di una competizione
- Provare le tecnologie Python in un ambiente più sfidante