

SABD - Progetto 2

Progetto 2 - Stream Processing

Di Totto Luca 0333084 - Di Marco Luca 0333083

Introduzione

L'obiettivo è progettare ed implementare un'applicazione in grado di effettuare **l'elaborazione streaming** di un grande volume di dati.

Nel caso specifico, lo scopo è quello di analizzare ed eseguire query su un dataset riguardante i **dati** di **monitoraggio** S.M.A.R.T.

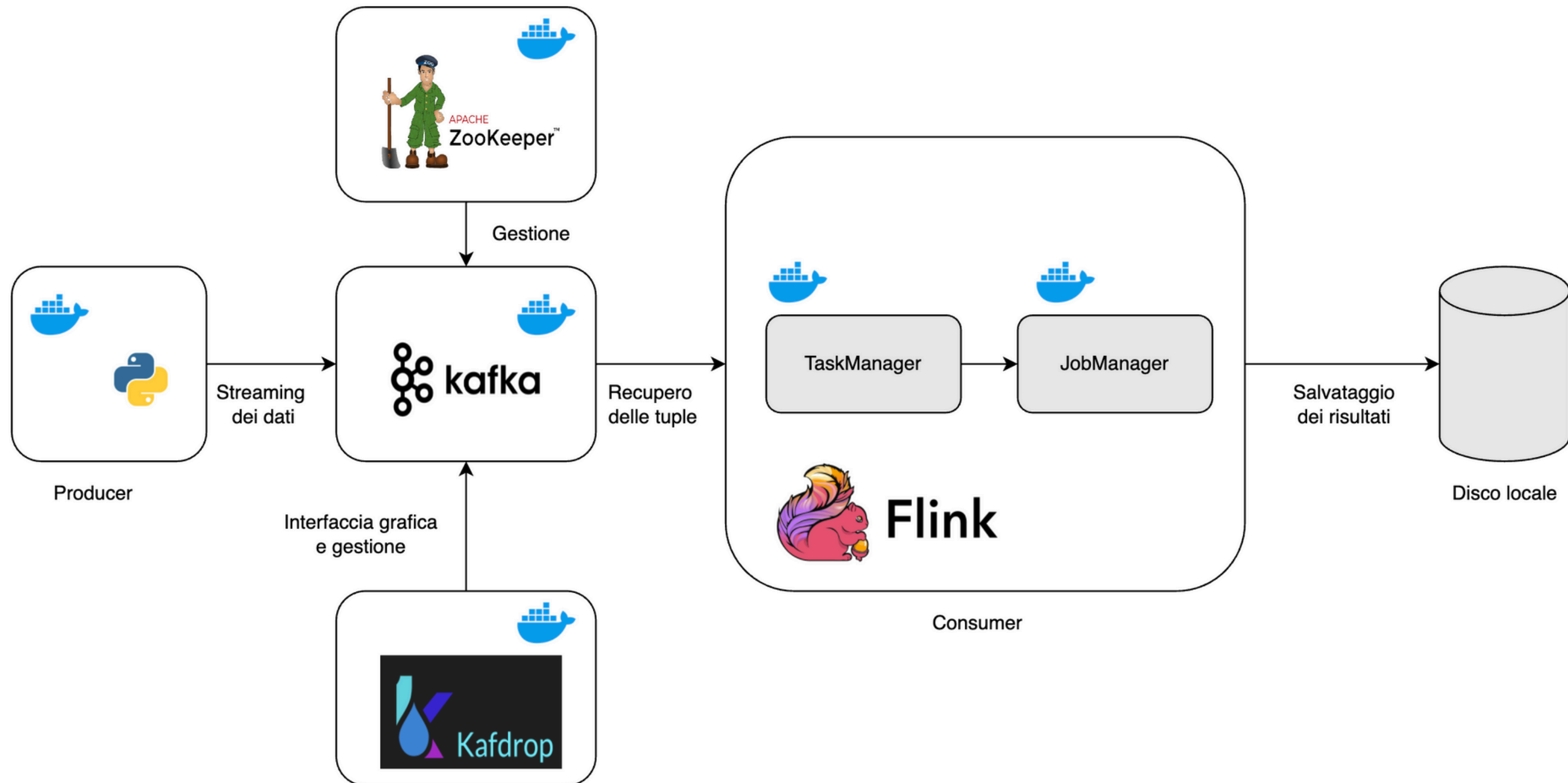
Quest'ultimo contiene **eventi** riguardanti circa duecento mila hard disk in un intervallo temporale di 23 giorni, per un totale di circa **tre milioni di eventi**.

Architettura

Il progetto prevede l'uso di diversi **framework** di messaggistica e di processamento streaming dei dati, eseguiti in più container docker. I componenti coinvolti sono:

- **Python:** Programma Python utilizzato per la simulazione del flusso streaming (data ingestion)
- **Kafka:** Layer di messaggistica per disaccoppiare layer di processamento e layer di ingestion
- **Kafdrop:** Framework per la gestione grafica di Kafka
- **Zookeeper:** Framework per la gestione di Kafka
- **Flink:** Framework per il processamento streaming

Architettura





SABD - Progetto 2

Python

Mediante Python avviene **l'ingestion** dei dati verso Kafka: prende il ruolo di Producer.

I dati vengono divisi in **23 dataframes** ed inoltrati in maniera distanziata per simulare il flusso streaming.



Kafka

Framework a livello di layer di messaggistica distribuita, utilizza un pattern di comunicazione di tipo **publisher-subscriber**.

Disaccoppia il programma Python da Flink.

- Kafka funge da **broker di messaggi**
- **Python** funge da **produttore**
- **Flink** funge da **consumer**

Necessaria la presenza di **Zookeeper**, che funge da coordinatore.

SABD - Progetto 2

Flink

Framework di **processamento streaming**.

Viene utilizzata l'immagine più recente di Flink per creare i due **servizi** principali: il **JobManager** e il **TaskManager**.

JobManager come responsabile della gestione delle risorse e del coordinamento dei lavori di streaming.

TaskManager come responsabile dell'esecuzione effettiva delle operazioni di calcolo

SABD - Progetto 2

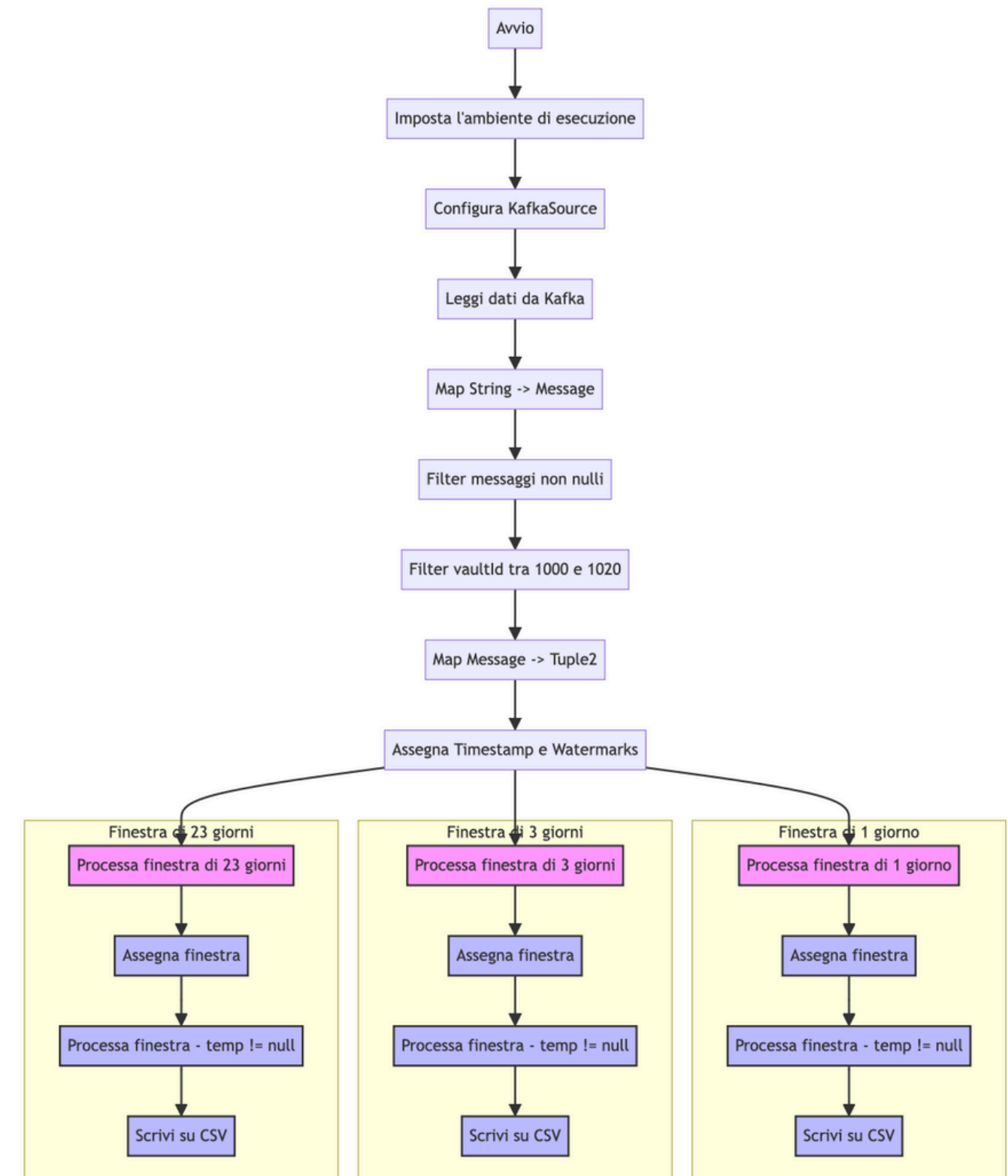
Query 1

Per i vault (campo **vault id**) con **identificativo** compreso tra **1000 e 1020**, calcolare il **numero di eventi**, il **valor medio e la deviazione standard della temperatura misurata** sui suoi hard disk (campo s194 temperature celsius). Per il calcolo della deviazione standard, si utilizzi un algoritmo online, come ad esempio l'algoritmo di Welford.

Calcolare la query sulle **finestre temporali**:

- 1 giorno (event time);
- 3 giorni (event time);
- dall'inizio del dataset.

```
window start,window end,vault_id,count,mean_s194,stddev_s194
2023-04-01, 2023-04-02, 1011, 526, 26.26, 6.62
2023-04-01, 2023-04-02, 1000, 564, 22.15, 2.38
2023-04-01, 2023-04-02, 1010, 521, 22.05, 2.79
2023-04-01, 2023-04-02, 1020, 573, 26.52, 5.86
2023-04-01, 2023-04-02, 1019, 565, 27.76, 5.09
2023-04-01, 2023-04-02, 1008, 577, 29.59, 4.14
2023-04-01, 2023-04-02, 1007, 554, 25.74, 4.63
2023-04-01, 2023-04-02, 1018, 544, 27.33, 3.82
2023-04-01, 2023-04-02, 1006, 556, 22.79, 3.94
2023-04-01, 2023-04-02, 1017, 556, 28.20, 5.38
2023-04-01, 2023-04-02, 1005, 547, 21.26, 2.74
2023-04-01, 2023-04-02, 1016, 537, 26.39, 4.63
2023-04-01, 2023-04-02, 1015, 538, 26.17, 4.08
2023-04-01, 2023-04-02, 1004, 545, 19.70, 2.68
2023-04-01, 2023-04-02, 1003, 560, 18.36, 2.30
2023-04-01, 2023-04-02, 1014, 548, 26.08, 5.50
```



SABD - Progetto 2

Query 2

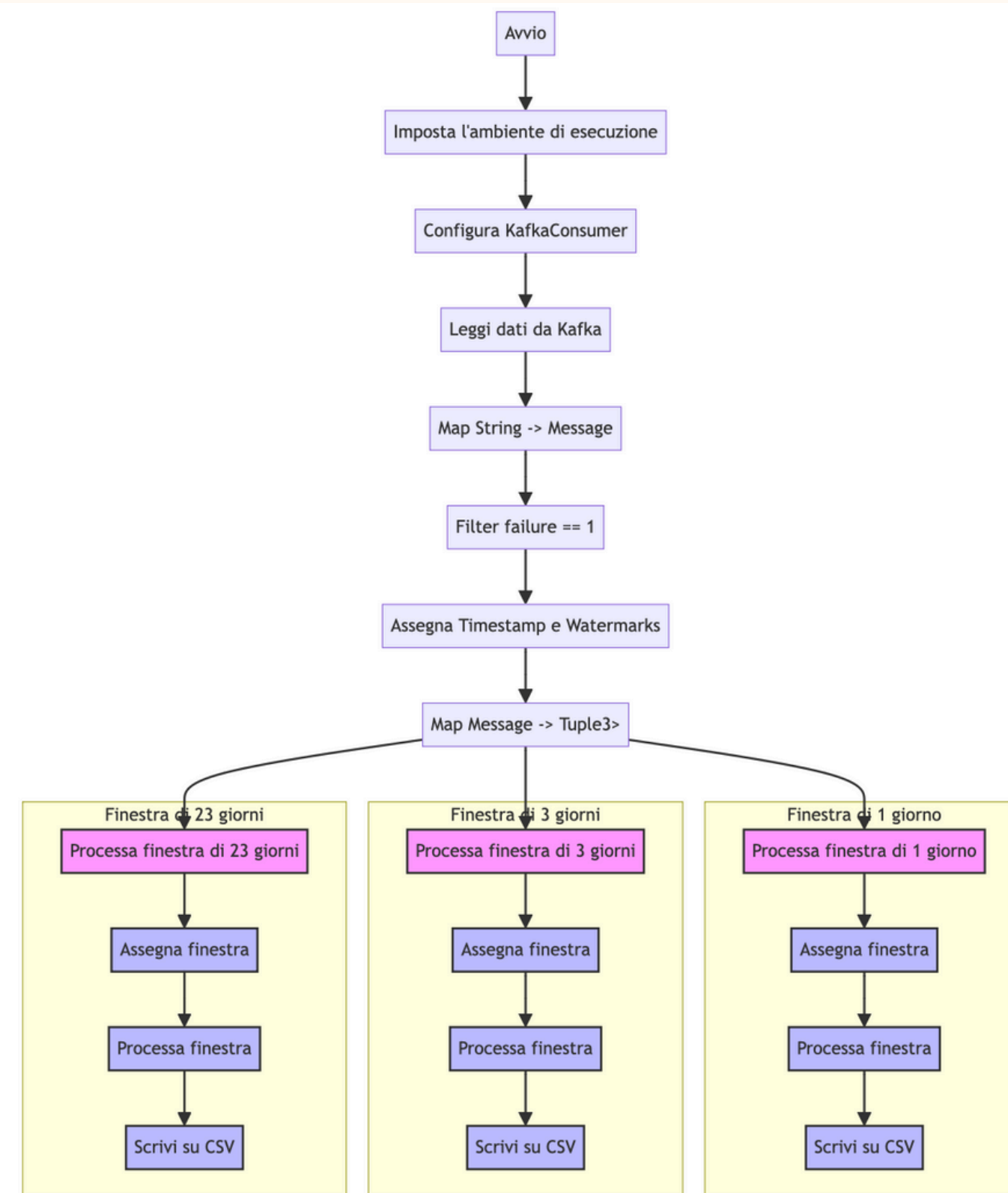
Calcolare la **classifica aggiornata in tempo reale dei 10 vault che registrano il più alto numero di fallimenti nella stessa giornata.**

Per ogni vault, riportare il numero di fallimenti ed il modello e numero seriale degli hard disk guasti.

Calcolare la query sulle **finestre temporali**:

- 1 giorno (event time);
- 3 giorni (event time);
- dall'inizio del dataset.

```
Window Start,Window End,Vault ID,Failures,Disk Details
2023-04-01 00:00:00,2023-04-04 00:00:00,1055,2,ST8000NM0055-ZA1814BV;ST8000NM0055-ZA16K94X
2023-04-01 00:00:00,2023-04-04 00:00:00,1091,1,ST8000NM0055-ZA155872
2023-04-01 00:00:00,2023-04-04 00:00:00,1099,1,ST12000NM001G-ZTN0AGRW
2023-04-01 00:00:00,2023-04-04 00:00:00,1131,1,ST12000NM0008-ZHZ2JJMP
2023-04-01 00:00:00,2023-04-04 00:00:00,1142,1,ST12000NM001G-ZLW0EFT7
2023-04-01 00:00:00,2023-04-04 00:00:00,1042,1,ST8000NM0055-ZA132DWE
2023-04-01 00:00:00,2023-04-04 00:00:00,1152,1,WDC WUH721414ALE6L4-X1G8LB2L
2023-04-01 00:00:00,2023-04-04 00:00:00,1085,1,ST14000NM001G-ZL2CWPWQ
2023-04-01 00:00:00,2023-04-04 00:00:00,1041,1,ST8000NM0055-ZA171RWG
2023-04-01 00:00:00,2023-04-04 00:00:00,1040,1,ST8000NM0055-ZA16DEVN
```



SABD - Progetto 2

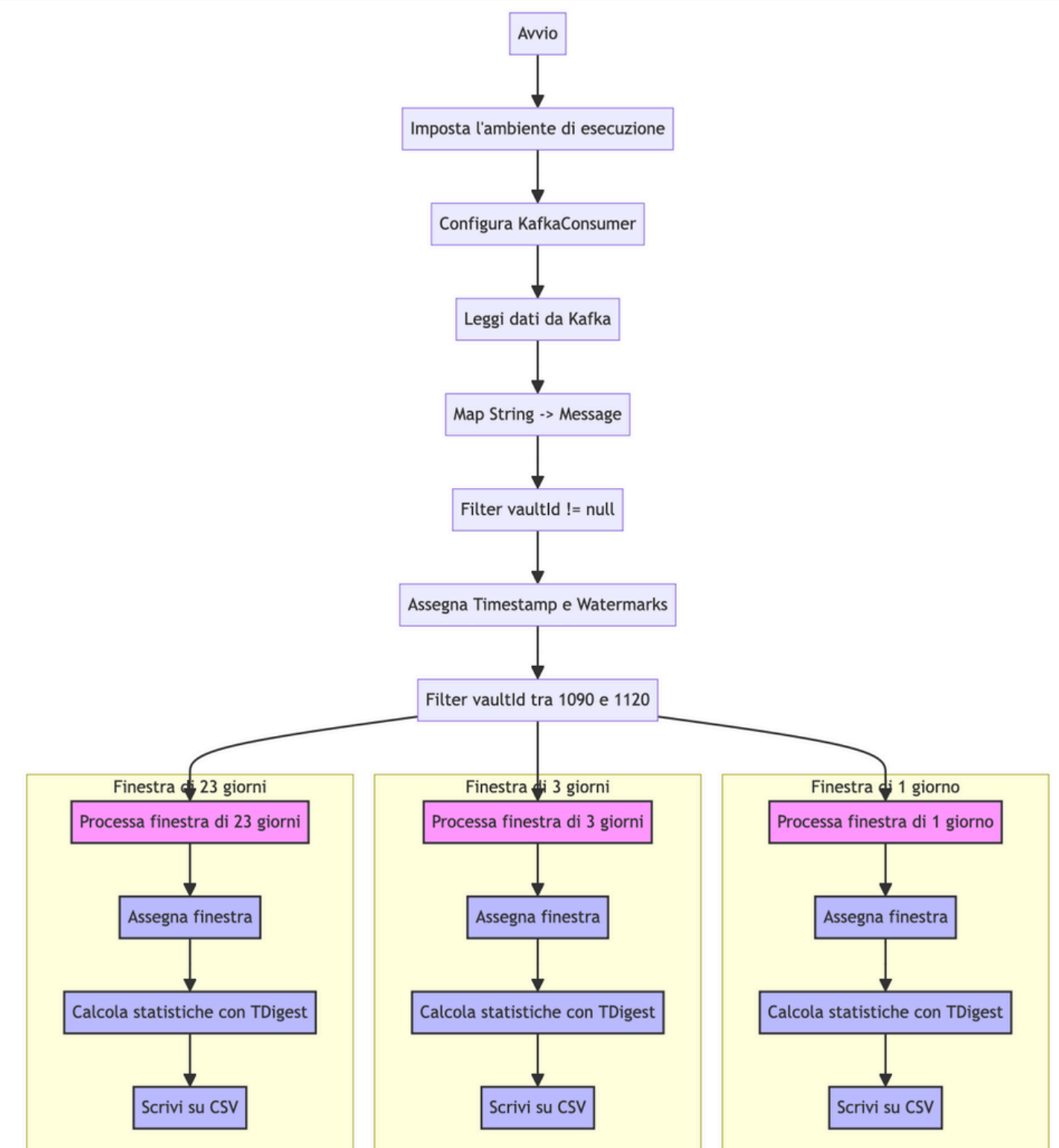
Query 3

Calcolare il **minimo, 25-esimo, 50-esimo, 75-esimo percentile e massimo delle ore di funzionamento** (campo s9 power on hours) degli hark disk per i vault con **identificativo tra 1090 (compreso) e 1120 (compreso)**. I percentili devono essere calcolati in tempo reale, senza ordinare tutti i valori e possibilmente senza accumularli; si utilizzi pertanto un **algoritmo approssimato** che consente di calcolare i percentili riducendo la quantità di memoria occupata al prezzo di una minore accuratezza.

Calcolare la query sulle **finestre temporali**:

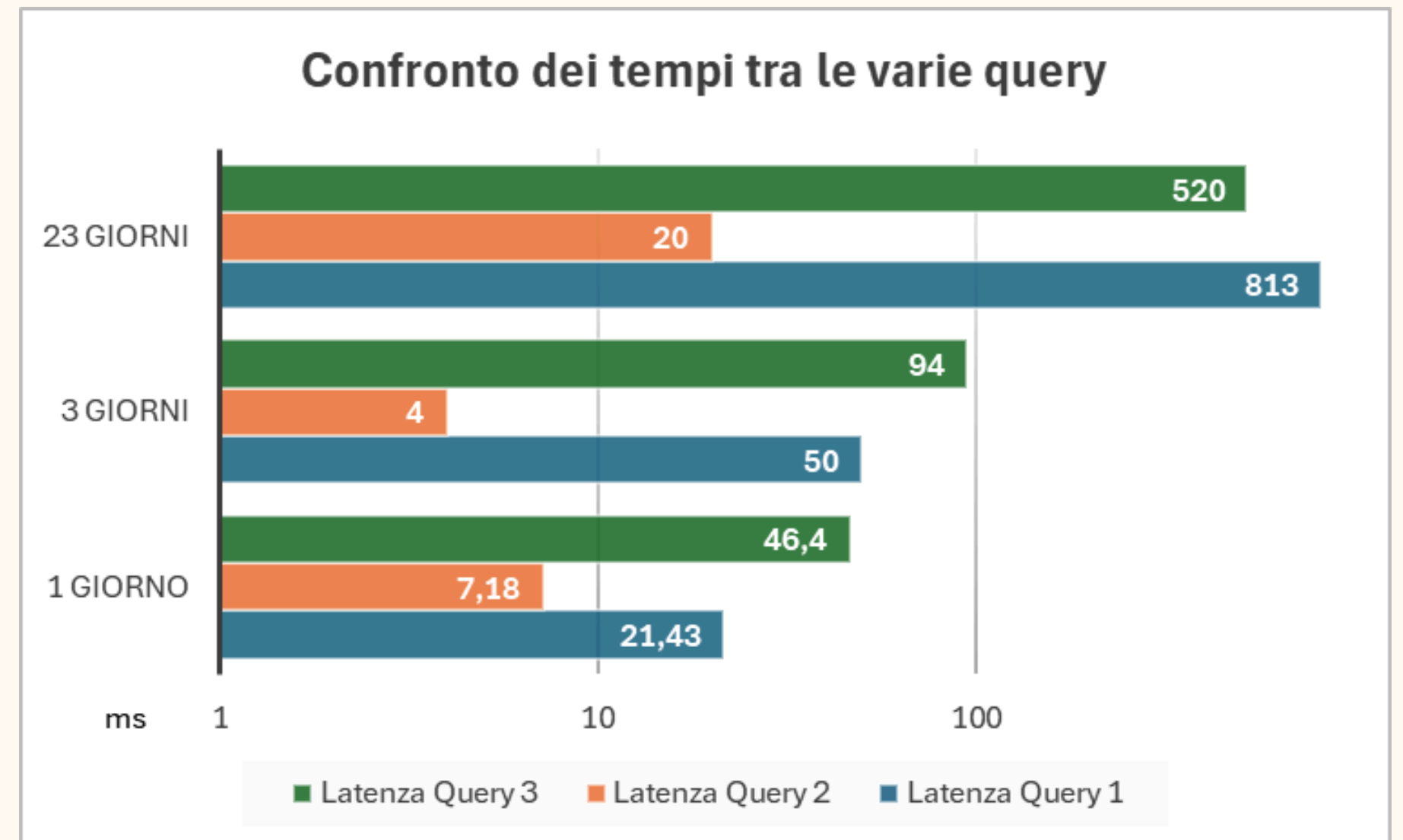
- 1 giorno (event time);
- 3 giorni (event time);
- dall'inizio del dataset.

```
window start>window end>vault_id,min,25perc,50perc,75perc,max,count
2023-04-01 00:00:00,2023-04-02 00:00:00,1120,94.00,35993.00,35997.00,36002.00,42532.00,373
2023-04-01 00:00:00,2023-04-02 00:00:00,1118,741.00,37502.00,37506.16,37511.00,37525.00,381
2023-04-01 00:00:00,2023-04-02 00:00:00,1117,125.00,18063.16,20442.44,22293.48,37633.00,414
2023-04-01 00:00:00,2023-04-02 00:00:00,1116,380.00,15713.07,17811.96,20780.38,85188.00,393
2023-04-01 00:00:00,2023-04-02 00:00:00,1115,1879.00,14785.26,16962.86,21350.76,38024.00,380
2023-04-01 00:00:00,2023-04-02 00:00:00,1114,1660.00,17546.40,18903.25,21593.47,38852.00,389
2023-04-01 00:00:00,2023-04-02 00:00:00,1113,126.00,38441.00,38448.08,38458.58,67990.00,404
2023-04-01 00:00:00,2023-04-02 00:00:00,1112,14367.00,16942.49,18524.48,21597.88,51007.00,368
2023-04-01 00:00:00,2023-04-02 00:00:00,1111,747.00,39381.78,39398.20,39406.38,39416.00,374
2023-04-01 00:00:00,2023-04-02 00:00:00,1119,1981.00,17495.92,18647.80,22187.89,36794.00,367
2023-04-01 00:00:00,2023-04-02 00:00:00,1091,139.00,49141.10,49146.56,49151.00,53076.00,392
2023-04-01 00:00:00,2023-04-02 00:00:00,1090,111.00,49426.60,49447.17,49455.00,75989.00,402
2023-04-01 00:00:00,2023-04-02 00:00:00,1110,1698.00,17136.80,18416.00,22250.28,40385.00,390
2023-04-01 00:00:00,2023-04-02 00:00:00,1099,428.00,14955.07,17926.12,22364.99,76005.00,377
2023-04-01 00:00:00,2023-04-02 00:00:00,1098,5451.00,17696.76,21185.44,23012.28,30953.00,398
```



Prestazioni

- L'**elaborazione** della **prima finestra temporale** riporta dei tempi ben superiori la media, per i seguenti motivi:
 - Il **primo** è il **delay** introdotto dalla fase di setup/avvio di Flink
 - Il **secondo** è legato al fatto che l'elaborazione del primo messaggio nella finestra giornaliera include il tempo di **creazione del file CSV** per i risultati di output
- Il tempo di **processamento** della **query n.2** è nettamente **minore** rispetto agli altri, a causa del filtro a monte
- In ogni query, i **tempi** medi delle finestre da uno e tre giorni sono decisamente **inferiori rispetto** ai tempi ottenuti considerando la finestra da **23 giorni**



SABD - Progetto 2

Prestazioni

Nella query n.2 è possibile osservare che i valori di **throughput** sono estremamente **bassi** e inferiori alle altre due query.

Questo comportamento è dovuto alla presenza di **operazioni di computazione all'interno della finestra molto onerose**, come: aggregazione complessa dei messaggi e ordinamento dei risultati.

