# Sistemi e Architetture per Big Data - A.A. 2023/24
# Progetto 2: Analisi in tempo reale di eventi di monitoraggio di dischi con Apache Flink

Docenti: Valeria Cardellini, Matteo Nardelli
Dipartimento di Ingegneria Civile e Ingegneria Informatica
Università degli Studi di Roma Tor Vergata

## Requisiti del progetto

Lo scopo del progetto è rispondere ad alcune query su dati di telemetria di circa 200k hard disk nei data center gestiti da Backblaze [1], utilizzando l'approccio di processamento a *stream* con Apache Flink. Per gli scopi di questo progetto, viene fornita una versione ridotta del dataset indicato nel Grand Challenge della conferenza ACM DEBS 2024 [2], che è disponibile all'URL[1]:

http://www.ce.uniroma2.it/courses/sabd2324/project/hdd-smart-data_medium-utv.tar.gz.

Il dataset riporta i dati di monitoraggio S.M.A.R.T.[2], esteso con alcuni attributi catturati da Backblaze. Il dataset contiene eventi riguardanti circa 200k hard disk, dove ogni evento riporta lo stato S.M.A.R.T. di un particolare hard disk in uno specifico giorno. Il dataset ridotto contiene circa 3 milioni di eventi (a fronte dei 5 milioni del dataset originario). La Tabella 1 descrive i campi di ogni evento. I campi rilevanti per questo progetto sono evidenziati nella tabella.

Il progetto è dimensionato per un gruppo composto da **2 studenti**; per gruppi composti da 1 oppure 3 studenti, si vedano le indicazioni specifiche.

Per lo svolgimento del progetto, si chiede di effettuare il replay del dataset, simulando il flusso di dati e accelerando opportunamente la scala temporale. La riproduzione accelerata deve preservare la coerenza tra intervalli temporali (ad esempio, accelerando con un fattore 3600, 1 ora in event time viene riprodotta in 1 secondo).

Le query a cui rispondere in modalità *streaming* sono:

**Q1** 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`). Si faccia attenzione alla possibile presenza di eventi che non hanno assegnato un valore per il campo relativo alla temperatura. Per il calcolo della deviazione standard, si utilizzi un algoritmo online, come ad esempio l'algoritmo di Welford[3].

Calcolare la query sulle finestre temporali:

- 1 giorno (event time)

[1] sha1sum di raw_data_medium-utv_sorted.csv: f5667bf30be58fbf83016f83924b29e65e6246f3; sha1sum del file compresso in TAR GZ: b7d026fdf9b2d14f57400fe206dee9c6f87c7e59

[2] `https://en.wikipedia.org/wiki/Self-Monitoring,_Analysis_and_Reporting_Technology`

[3] `https://en.m.wikipedia.org/wiki/Algorithms_for_calculating_variance#Welford's_online_algorithm`

Tabella 1: Formato dei dati forniti da Backblaze

| Campo | Informazioni | Rilevante |
|---|---|---|
| date | format: 2023-04-01T00:00:00.000000 | ✓ |
| serial_number | string | ✓ |
| model | string | ✓ |
| failure | (bool) | ✓ |
| vault_id | group of storage servers (int64) | ✓ |
| s1_read_error_rate | (int64) | |
| s2_throughput_performance | (int64) | |
| s3_spin_up_time | (int64) | |
| s4_start_stop_count | (int64) | |
| s5_reallocated_sector_count | (int64) | |
| s7_seek_error_rate | (int64) | |
| s8_seek_time_performance | (int64) | |
| s9_power_on_hours | (int64) | ✓ |
| s10_spin_retry_count | (int64) | |
| s12_power_cycle_count | (int64) | |
| s173_wear_leveling_count | (int64) | |
| s174_unexpected_power_loss_count | (int64) | |
| s183_sata_downshift_count | (int64) | |
| s187_reported_uncorrectable_errors | (int64) | |
| s188_command_timeout | (int64) | |
| s189_high_fly_writes | (int64) | |
| s190_airflow_temperature_cel | (int64) | |
| s191_g_sense_error_rate | (int64) | |
| s192_power_off_retract_count | (int64) | |
| s193_load_unload_cycle_count | (int64) | |
| s194_temperature_celsius | (int64) | ✓ |
| s195_hardware_ecc_recovered | (int64) | |
| s196_reallocated_event_count | (int64) | |
| s197_current_pending_sector | (int64) | |
| s198_offline_uncorrectable | (int64) | |
| s199_udma_crc_error_count | (int64) | |
| s200_multi_zone_error_rate | (int64) | |
| s220_disk_shift | (int64) | |
| s222_loaded_hours | (int64) | |
| s223_load_retry_count | (int64) | |
| s226_load_in_time | (int64) | |
| s240_head_flying_hours | (int64) | |
| s241_total_lbas_written | (int64) | |
| s242_total_lbas_read | (int64) | |

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

L'output della query ha il seguente schema:

```
ts, vault_id, count, mean_s194, stddev_s194
```

dove:

- `ts`: timestamp relativo all'inizio della finestra su cui è stata calcolata la statistica;

- `vault_id`: identificativo del vault;
- `count`: numero di misurazioni;
- `mean_s194`: valor medio della temperatura nella finestra;
- `stddev_s194`: (stimatore della) deviazione standard della temperatura nella finestra.

**Q2** 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.

L'output della query ha il seguente schema:

```
ts, vault_id1, failures1 ([modelA, serialA, ...]), ..., vault_id10,
failures10 ([modelZ, serialZ, ...])
```

dove:

- `ts`: timestamp relativo all'inizio della finestra su cui è stata calcolata la classifica;
- `vault_id[1-10]`: identificativo del vault in posizione [1-10] nella classifica top-10;
- `failures[1-10]`: numero di fallimenti registrati per il vault con `vault_id[1-10]` nella finestra considerata;
- `[modelA, serialA, ...]`: lista di modelli e numeri seriali degli hard disk guasti per il vault di riferimento.

**Q3** 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). Si presti attenzione, il campo `s9_power_on_hours` riporta un valore cumulativo, pertanto le statistiche richieste dalla query devono far riferimento all'ultimo valore utile di rilevazione per ogni specifico hard disk (si consideri l'uso del campo `serial_number`).

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, e.g., [4, 6, 7, 3, 5].

Calcolare la query sulle finestre temporali:

- 30 minuti (event time);
- 1 ora (event time);
- 1 giorno (event time).

L'output della query ha il seguente schema:

```
ts, vault_id, min, 25perc, 50perc, 75perc, max, count
```

dove:

- `ts`: timestamp relativo all'inizio della finestra su cui sono calcolate le statistiche;
- `vault_id`: identificativo del vault;
- `min`: valore minimo delle ore di funzionamento degli hard disk appartenenti al vault di riferimento;
- `25perc`: 25-esimo percentile delle ore di funzionamento degli hard disk appartenenti al vault di riferimento;
- `50perc`: 50-esimo percentile delle ore di funzionamento degli hard disk aappartenenti al vault di riferimento;
- `75perc`: 75-esimo percentile delle ore di funzionamento degli hard disk appartenenti al vault di riferimento;
- `max`: valore massimo delle ore di funzionamento degli hard disk appartenenti al vault di riferimento;
- `count`: numero di hard disk del vault di riferimento considerati per il calcolo delle statistiche.

Il risultato di ciascuna query deve essere consegnato in formato CSV (Flink supporta la scrittura di file in questo formato). Si chiede inoltre di valutare sperimentalmente i tempi di latenza ed il throughput delle query durante il processamento sulla piattaforma usata per la realizzazione del progetto. Riportare l'analisi del confronto nella relazione e nella presentazione del progetto.

**Per gruppi composti da 1 studente**: si richiede di rispondere alle query 1 e 2.

**Per gruppi composti da 3 studenti**: in aggiunta ai requisiti sopra elencati, si richiede di utilizzare *Kafka Streams* oppure *Spark Streaming* per rispondere alle query 1 e 2 e di confrontare, sulla stessa piattaforma di riferimento, le prestazioni in termini di tempo di latenza e throughput con quelle ottenute usando Apache Flink. Riportare l'analisi del confronto nella relazione e nella presentazione del progetto.

**Opzionale**: Rispondere ad una query a scelta tra le tre sopra descritte usando *Kafka Streams* oppure *Spark Streaming* e confrontare, sulla stessa piattaforma di riferimento, le prestazioni in termini di tempo di latenza e throughput con quelle ottenute usando Apache Flink.

## Svolgimento e consegna del progetto

Comunicare la composizione del gruppo ai docenti entro **venerdì 21 giugno 2024** (sole se diversa rispetto al progetto 1).

Per ogni comunicazione via email è necessario specificare *[SABD]* nell'oggetto dell'email. Il progetto è valido **solo** per l'A.A. 2023/24 ed il codice deve essere consegnato **entro martedì 9 luglio 2024**.

La consegna del progetto consiste in:

1. link a spazio di Cloud storage o repository contenente il codice del progetto da comunicare via email ai docenti **entro martedì 9 luglio 2024**; inserire i risultati delle query in formato CSV in una cartella denominata `Results`.

2. relazione di lunghezza compresa tra le 3 e le 6 pagine, da inserire all'interno della cartella denominata `Report`; per la relazione si consiglia di usare il formato ACM proceedings (https://www.acm.org) oppure il formato IEEE proceedings (https://www.ieee.org);

3. slide della presentazione orale, da inviare via email ai docenti **dopo** lo svolgimento della presentazione.

La presentazione si terrà **mercoledì 10 luglio 2024**; ciascun gruppo avrà a disposizione **massimo 15 minuti** per presentare la propria soluzione.

## Valutazione del progetto

I principali criteri di valutazione del progetto saranno:

1. rispondenza ai requisiti;
2. originalità;
3. architettura del sistema e deployment;
4. organizzazione del codice;
5. efficienza;
6. organizzazione, chiarezza e rispetto dei tempi della presentazione orale.

## Riferimenti bibliografici


[1] Backblaze. `https://www.backblaze.com/`, 2024.

[2] DEBS 2024. Call for Grand Challenge Solutions. `https://2024.debs.org/call-for-grand-challenge-solutions/`, 2024.

[3] T. Dunning. The t-digest: Efficient estimates of distributions. *Software Impacts*, 7:100049, 2021. `https://www.softwareimpacts.com/article/S2665-9638(20)30040-3/fulltext`.

[4] S. Engelhardt. Calculating Percentiles on Streaming Data Part 1: Introduction. `https://www.stevenengelhardt.com/2018/03/06/calculating-percentiles-on-streaming-data-part-1-introduction/`, 2018.

[5] I. Epicoco, C. Melle, M. Cafaro, M. Pulimeno, and G. Morleo. UDDSketch: Accurate tracking of quantiles in data streams. *IEEE Access*, 8:147604–147617, 2020. `https://ieeexplore.ieee.org/iel7/6287639/8948470/09163358.pdf`.

[6] L. Fernando, H. Bindra, and K. Daudjee. An experimental analysis of quantile sketches over data streams. In *Proc. of 26th Int'l Conf. on Extending Database Technology, EDBT '23*, 2023. `https://cs.uwaterloo.ca/~kdaudjee/Daudjee_Sketches.pdf`.

[7] R. Jain and I. Chlamtac. The $p^2$ algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM*, 28(10):1076–1085, 1985. `https://www.cse.wustl.edu/~jain/papers/ftp/psqr.pdf`.