Project 2

Corso di Sistemi e Architetture per Big Data

A.A. 2023/24

Valeria Cardellini, Matteo Nardelli

Laurea Magistrale in Ingegneria Informatica

Project delivery

- Submission deadline
  - July 9, 2024
- Your presentation
  - July 10, 2024 (to be confirmed)
- What to deliver
  - Link to cloud storage or repository containing project code
  - Project report composed by 3-6 pages in ACM or IEEE proceedings format
  - Presentation slides (max. **15 minutes** per group), to be delivered after your presentation
- Team
  - Target: 2 students per team
  - Alternatives: 1 student or 3 students per team

Dataset

- Already known from Project 1 ☺
- Hard disk drives: the most frequently replaced hw components of data centers and the main reason behind server failures
- You will analyze a dataset of real-world telemetry data for hard drive failures
  - Data provided by [Backblaze]() and used for DEBS 2024 GC
  - See link in project description or Teams
  - Reduced dataset in CSV format (23 days), ~600MB, ~3M events

A type of disk failure:
a head crash

Dataset

- Each tuple includes fields that provide:
  - SMART (Self-Monitoring Analysis and Reporting Technology) telemetry data
    en.wikipedia.org/wiki/Self-Monitoring,_Analysis_and_Reporting_Technology
  - Plus additional attributes added by Backblaze
- The most relevant fields for this project

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)	✓

Dataset

- Some lines of the dataset

header

```
date,serial_number,model,failure,vault_id,s1_read_error_rate,s2_throughput_performance,
s3_spin_up_time,s4_start_stop_count,s5_reallocated_sector_count,s7_seek_error_rate,
s8_seek_time_performance,s9_power_on_hours,s10_spin_retry_count,s12_power_cycle_count,
s173_wear_leveling_count,s174_unexpected_power_loss_count,s183_sata_downshift_count,
s187_reported_uncorrectable_errors,s188_command_timeout,s189_high_fly_writes,
s190_airflow_temperature_cel,s191_g_sense_error_rate,s192_power_off_retract_count,
s193_load_unload_cycle_count,s194_temperature_celsius,s195_hardware_ecc_recovered,
s196_reallocated_event_count,s197_current_pending_sector,s198_offline_uncorrectable,
s199_udma_crc_error_count,s200_multi_zone_error_rate,s220_disk_shift,s222_loaded_hours,
s223_load_retry_count,s226_load_in_time,s240_head_flying_hours,s241_total_lbas_written,
s242_total_lbas_read
```

tuple 1

```
2023-04-01T00:00:00.000000,8HK2SSMH,HGST HUH721212ALN604,0,1113,0.0,96.0,396.0,24.0,
0.0,0.0,18.0,38445.0,0.0,24.0,,,,,,,,,1613.0,1613.0,31.0,,0.0,0.0,0.0,0.0,,,,,,,,
```

tuple 2

```
2023-04-01T00:00:00.000000,10B0A01UF97G,TOSHIBA MG07ACA14TA,0,1067,0.0,0.0,7889.0,7.0,
0.0,0.0,0.0,27425.0,0.0,7.0,,,,,,,,1.0,1.0,64.0,32.0,,0.0,0.0,0.0,0.0,,17956865.0,
27325.0,0.0,592.0,0.0,,
```

tuple 3

```
2023-04-01T00:00:00.000000,5080A117F97G,TOSHIBA MG07ACA14TA,0,1095,0.0,0.0,7872.0,8.0,
0.0,0.0,0.0,18029.0,0.0,8.0,,,,,,,,234.0,5.0,14.0,36.0,,0.0,0.0,0.0,0.0,,34996225.0,
17990.0,0.0,590.0,0.0,,
```

. . .

```
117542,2023-04-01T00:00:00.000000,70K0A08ZF97G,TOSHIBA MG07ACA14TA,1,1106,0.0,0.0,
7923.0,3.0,0.0,0.0,0.0,15604.0,0.0,3.0,,,,,,,,0.0,2.0,9.0,21.0,,0.0,1.0,0.0,0.0,,
34734083.0,15573.0,0.0,591.0,0.0,,
```

temperature

A failure occurred: `failure` set to 1

Queries with Flink

- Use Flink framework to answer some queries on the dataset
  - Programming language? Your choice
- Include in your report/slides queries' latency time and throughput on your reference platform

Queries with Flink

Query 1

- For vaults (`vault_id` field) having an identifier between 1000 and 1020, compute number of events, average value and standard deviation of temperature measured on its hard disks (`s194_temperature_celsius` field)
  - Warning: temperature is missing in some tuples
  - For standard deviation, use an online algorithm, (e.g., Welford algorithm)
- Time windows:
  - 1 day (event time)
  - 3 days (event time)
  - from the beginning of the dataset
- Output example

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


Queries with Flink

Query 2

- Compute the ranking updated in real time of the 10 vaults that record the highest number of failures on the same day. For each vault, report the number of failures and the model and serial number of the failed hard disks
- Time windows:
  - 1 day (event time)
  - 3 days (event time)
  - from the beginning of the dataset
- Output example

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

Queries with Flink

Query 3

- Compute minimum, 25th, 50th, 75th percentile and maximum of the operating hours (field s9_power_on_hours) of hard disks for vaults having 1090 <= `vault_id` <= 1120
  - Warning: `s9_power_on_hours` reports a cumulative value, therefore statistics should refer to the last useful value for each specific hard disk (use the serial number)
  - Compute percentiles in real time by means of estimators (i.e., *approximation*), without sorting all the values and storing them in such a way to reduce the memory footprint
    - Trade-off between accuracy and performance
    - For percentile approximation algorithms (e.g., t-digest, KLL Sketch), see references in the project description: most algorithms involve some sort of modified histogram to represent the overall shape of the data more compactly, while still capturing much of the shape of the distribution

Queries with Flink

Query 3

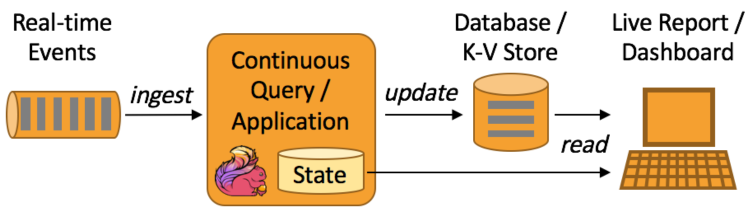
- Time windows:
  - 30 minutes (event time)
  - 1 hour (event time)
  - 1 day (event time)
- Output example

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

Data ingestion

- We are now considering streaming analytics: <u>do not</u> read the entire dataset, but ingest real-time event streams so that Flink continuously produces and updates results as events are consumed

Streaming analytics



Data ingestion

- You need to simulate the arrival of data streams in real time
  - `sleep()` between consecutive tuples
- To speed up the simulation, you can divide the time interval between consecutive tuples by a constant speeding factor
- Frameworks: Kafka, Pulsar, RabbitMQ, NiFi, …

Platform and performance evaluation

- Evaluate experimentally the latency time and throughput on your reference platform
- Platform can be a standalone node
  - Recommended: use Docker Compose to orchestrate the containers running on the same machine
- Alternatively, you can use a Cloud service for Big Data processing (i.e., Amazon Managed Service for Apache Flink) using AWS Academy grant

Team of 3 students

- Use either Spark Streaming or Kafka Streams to solve Queries 1 and 2
- Evaluate the performance of Queries 1 and 2 on your reference platform and compare with that achieved when using Flink

Optional

- Use Kafka Streams or Spark Streaming to answer one query of your choice
  – Query 3 if you are in team of 3 students

Team composition and tasks

- 1 student in the team:
  – Queries 1 and 2
- 2 students in the team:
  – Queries 1, 2 and 3
- 3 students in the team:
  – Queries 1, 2 and 3
  – Queries 1 and 2 using Kafka Streams or Spark Streaming