TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA
Macroarea di Ingegneria
Dipartimento di Ingegneria Civile e Ingegneria Informatica

Project 1

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
  - June 10, 2024
- Your presentation
  - June 13, 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

- 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)	

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,,
```

```
. . .
```

Failure: `failure` set to 1
```
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,,
```


Queries with Spark

- Use Spark framework to answer some queries on the dataset
  - RDDs/Dataframes without using SQL
  - Programming language? Your choice
- Include in your report/slides queries' response time on your reference platform

Queries with Spark

Query 1

- For each day and each vault (`vault id` field), compute the total number of failures. Determine the vaults for which occurred exactly 4, 3 and 2 disk failures.
  - Output example

 # DD-MM-YYYY, vault_id, count
11-04-2023, 1090, 4
...
19-04-2023, 1120, 3
...
01-04-2023, 1055, 2
...

Queries with Spark

Query 2

- Compute the ranking of the 10 hard disk models for which the highest number of failures occurred. The ranking shows the hard disk model and the total number of failures occurred to hard disks of that specific model.
  - Output example

 # model, failures_count
HGST HUH721212ALN604, 47
ST8000NM0055, 45
ST4000DM000, 28
...

Queries with Spark

Query 2

- Then, compute a second ranking of the 10 vaults that recorded the highest number of failures. For each vault, report the number of failures and the list (without repetitions) of hard disk models in that vault subject to at least one failure.
  - Output example

 # vault_id, failures_count, list_of_models
1113, 16, HGST HUH721212ALN604
1093, 9, ST10000NM0086
1090, 9, ST8000NM0055, TOSHIBA MQ01ABF050, WDC WD5000LPVX
...

Queries with Spark

Query 3

- Compute minimum, 25, 50 (median), 75 percentile and maximum of operating hours (`s9 power on hours` field) for failed hard disks and not failed hard disks; provide also the total number of events you used to compute the statistics
  - Note: since `s9 power on hours` is a cumulative value, determine the last useful day of detection for each specific hard disk (use `serial number` field)
  - Output example

 # failure, min, 25th percentile, 50th percentile, 75th percentile, max, count
0, 0, 15082, 22490, 47118, 87726, 243336
1, 2306, 24505, 38603, 53079, 71491, 267

Platform and performance evaluation

- Evaluate experimentally query processing times 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 EMR) using the AWS Academy grant

Optional part A

- Compulsory for team composed of 3 students

- Use Spark SQL to solve Queries 1, 2 and 3
- Evaluate the performance of all the queries on your reference platform for both cases

Data acquisition and ingestion

- Which framework to ingest data into HDFS?
  - Flume, NiFi, Kafka, Pulsar, …
- Which format to store data?
  - csv, columnar format (Parquet, ORC), row format (Avro), …
- Where to export your results?
  - HBase, Redis, Kafka, …

Optional part B

- Use a visualization framework (e.g., Grafana) to graphically present the query results

Team composition and tasks

- 1 student in the team:
  - Queries 1 and 2
  - Data acquisition and ingestion are optional, HDFS is mandatory
- 2 students in the team:
  - Queries 1, 2 and 3
  - Plus data acquisition and ingestion
- 3 students in the team:
  - Queries 1, 2 and 3
  - Plus data acquisition and ingestion
  - Plus optional part A using Spark SQL