Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

PROJECT ΕΑΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2023-24
Έκδοση εγγράφου 1.2 30/04/2024

ΥΠΕΥΘΥΝΟΙ ΚΑΘΗΓΗΤΕΣ: Μεγαλοοικονόμου Βασίλειος, Κομνηνός Ανδρέας

Big Data Applications

Σκοπός εργασίας

Η παρούσα εργασία αποσκοπεί στην εξοικείωση των φοιτητών με τις τρέχουσες τεχνολογίες αποθήκευσης, ανάκτησης και ανάλυσης των Big Data. Τα Big Data εμφανίζονται με μια πληθώρα από μορφές όπως τα web logs, τα internet clickstreams, τα αδόμητα ή ημιδομημένα δεδομένα. Η ανάλυση των Big Data χρησιμοποιεί πηγές δεδομένων οι οποίες παρέμεναν ανεκμετάλλευτες από τις συμβατικές λύσεις. Ζητήματα όπως η διαχείριση ετερογενών κι ανομοιόμορφων σημαντικά μεγάλων δεδομένων από σχεσιακές βάσεις δεδομένων, η ανάκτηση μεγάλων data sets που είναι διάσπαρτα σε ομογενή ή ετερογενή συστήματα, η επεξεργασία φυσικής γλώσσας, η μηχανική μάθηση και τεχνητή νοημοσύνη καθώς και η πρόβλεψη που στηρίζεται σε αδόμητα δεδομένα, ικανοποιούνται πλέον από τα συστήματα ανάλυσης Big Data. Ειδικότερα στην παρούσα εργασία θα ασχοληθούμε και θα δούμε στην πράξη (hands-on) την διαχείριση δεδομένων με την χρήση εργαλείων ανοικτού κώδικα.

Ομάδες

Η εργασία μπορεί να εκπονηθεί από μεμονωμένα άτομα ή ομάδες το πολύ δύο ατόμων.

Προετοιμασία & Documentation

Προαπαιτούμενο SW

Τα ερωτήματα μπορούν να υλοποιηθούν τοπικά στον υπολογιστή σας. Για την υλοποίηση του project στον τοπικό υπολογιστή θα χρειαστείτε τα παρακάτω βοηθητικά εργαλεία:

- Δημιουργία μιας εικονικής μηχανής με Linux (π.χ. ubuntu) – προαιρετικό αλλά συστήνεται ανεπιφύλακτα.
- Εγκατάσταση Java 8 (11 με μικρές αλλαγές στο config)
- Εγκατάσταση Python 3.8+
- Εγκατάσταση Jupyter Notebooks (προαιρετικό αλλά συνιστάται)
- Εγκατάσταση Kafka, Spark, MongoDB και συναφών εργαλείων.

Περιγραφή datasets και εργασιών

Δεδομένα

Για την εργασία θα αξιοποιηθεί ένας εξομοιωτής παραγωγής δεδομένων από διαδράσεις χρηστών με μια online υπηρεσία κράτησης ξενοδοχείων. Μπορείτε να φανταστείτε ότι τα δεδομένα παράγονται από κώδικα για analytics, που είναι ενσωματωμένος στον ιστότοπο της υπηρεσίας. Ο εξομοιωτής είναι γραμμένος σε Python και ο κώδικάς του επισυνάπτεται με την εκφώνηση.

Στόχοι εργασίας

Για την εργασία θα υλοποιήσετε ένα τυπικό αγωγό (pipeline) εισόδου, επεξεργασίας σε πραγματικό χρόνο και αποθήκευσης σε NoSQL βάση δεδομένων (MongoDB).

1. Παραγωγή δεδομένων. Με βάση τα αποτελέσματα του εξομοιωτή θα γράψετε ένα python script που θα στέλνει δεδομένα σε ένα Kafka broker σε τακτά χρονικά διαστήματα.
2. Επεξεργασία σε πραγματικό χρόνο. Τα εισρέοντα δεδομένα από τον Kafka broker θα διαβάζονται από μια υλοποίηση Apache Spark, η οποία εκτελεί real-time επεξεργασία στα δεδομένα.
3. Αποθήκευση σε NoSQL βάση δεδομένων. Τα ωμά δεδομένα και η επεξεργασμένη (από το Spark) μορφή τους αποθηκεύονται σε μια υλοποίηση MongoDB.

Ερώτημα 1: Παραγωγή δεδομένων

1. Εγκαταστήστε και τρέξτε τον εξομοιωτή ώστε να παράγετε δεδομένα για το project. Τα δεδομένα μπορούν να παραχθούν σε μορφή CSV ή JSON (επιλέξτε ότι σας βολεύει). Ο εξομοιωτής παράγει δεδομένα διάδρασης για κάθε χρήστη, τα οποία συνοδεύονται από κάποιο timestamp. Τα πιθανά πεδία που περιέχει κάθε εγγραφή είναι:

    - user_id: μοναδικό αναγνωριστικό για κάθε χρήστη
    - session_id: μοναδικό αναγνωριστικό session
    - timestamp: χρόνος καταγραφής διαδράσης του χρήστη
    - event_type: είδος διάδρασης του χρήστη
    - booking_id: αν έγινε κράτηση, το id της κράτησης
    - check_in_date: αν έγινε αναζήτηση ή κράτηση, η ημερομηνία άφιξης
    - check_out_date: αν έγινε αναζήτηση ή κράτηση, η ημερομηνία αναχώρησης
    - hotel_id: μοναδικό αναγνωριστικό για κάθε ξενοδοχείο,
    - item_id: μοναδικό αναγνωριστικό για κάθε δωμάτιο που προστίθεται σε μια κράτηση
    - location: η πόλη αναζήτησης / κράτησης
    - num_guests: πλήθος ατόμων
    - page_url: το url στο οποίο βρίσκεται ο χρήστης
    - payment_method: ο τρόπος πληρωμής
    - price: η τιμή ανά δωμάτιο
    - room_type: το είδος δωματίου
    - total_price: το συνολικό κόστος κράτησης

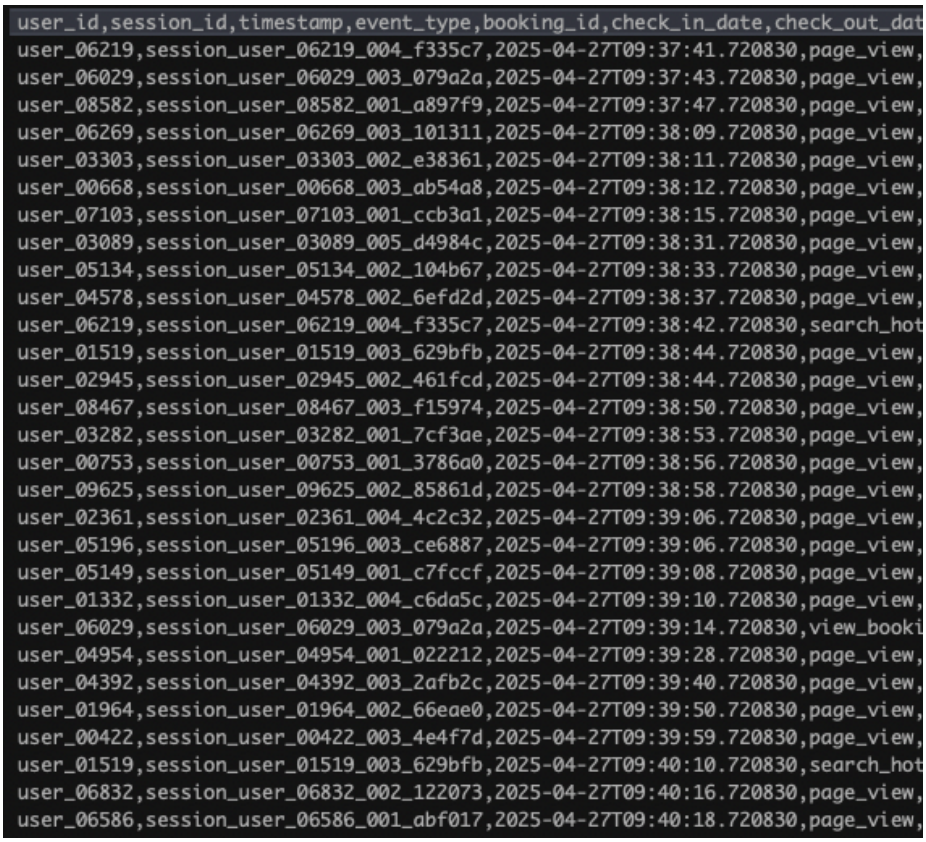
    Προσέξτε ότι κάθε εγγραφή περιέχει υποχρεωτικά τα πεδία ["user_id", "session_id", "timestamp", "event_type"], ενώ τα υπόλοιπα εξαρτώνται από το είδος του event που καταγράφεται.

2. Υλοποιήστε μια υπηρεσία kafka broker ακολουθώντας τις σχετικές οδηγίες εγκατάστασης.
3. Δημιουργήστε ένα νέο topic με όνομα “`website_actions`” μέσω command line (step 3 του οδηγού) ή μέσω της βιβλιοθήκης kafka-python. Στο topic αυτό θα στέλνονται τα δεδομένα που δημιουργήθηκαν από τον εξομοιωτή.

4. Γράψτε ένα script σε python ώστε κάθε *N* δευτερόλεπτα να αποστέλλει τα δεδομένα όλων των διαδράσεων που έχουν γίνει, προς τον kafka broker του ερωτήματος 2 (δείτε παρακάτω). Για το σκοπό αυτό χρησιμοποιήστε τη βιβλιοθήκη kafka-python.

   Για την αποστολή, δημιουργήστε ένα αντικείμενο της κλάσης `KafkaProducer` και στέλνετε περιοδικά (κάθε *N* δευτερόλεπτα) όλα τα σχετικά δεδομένα. Κάθε διάδραση θα πρέπει να αποστέλλεται ως ξεχωριστό JSON αντικείμενο που στέλνεται αυτόνομα και ανεξάρτητα από όλες τις άλλες διαδράσεις. Προσέξτε επίσης ότι θα πρέπει να τροποποιήσετε τη χρονική σφραγίδα των δεδομένων με βάση την ημερομηνία και ώρα που τρέξατε την προσομοίωση. Συνεπώς, για παράδειγμα ας θεωρήσουμε ότι φτιάξατε κάποια δεδομένα και το πρώτο από αυτά έχει ημερομηνία T1 = 27/4/2025 09:37:41. Λίγο αργότερα (π.χ. στη χρονική στιγμή T2 = 27/4/2025 09:42:39) εκκινείτε το script αποστολής των δεδομένων. Η ημερομηνία της 1ης εγγραφής θα πρέπει να τροποποιηθεί με βάση τη χρονική διαφορά ΔT = T2-T1, και προφανώς κάθε επόμενη εγγραφή πρέπει να τροποποιηθεί κατά τον ίδιο τρόπο.

   Κατά συνέπεια, το script αποστολής εκκινεί τη στιγμή T2, ανακτά όλες τις εγγραφές που είναι εντός του χρόνου T2+N (προφανώς αφού κάθε εγγραφή έχει τροποποιηθεί κατά ΔT), και τις αποστέλλει στο kafka topic. Θα παρατηρήσετε ότι ο εξομοιωτής παράγει τα δεδομένα με αύξουσα χρονική σειρά, άρα δε χρειάζεται να κάνετε κάποια ταξινόμηση των δεδομένων (ειδικά αν εργαστείτε με csv αρχείο).



```
user_id,session_id,timestamp,event_type,booking_id,check_in_date,check_out_dat
user_06219,session_user_06219_004_f335c7,2025-04-27T09:37:41.720830,page_view,
user_06029,session_user_06029_003_079a2a,2025-04-27T09:37:43.720830,page_view,
user_08582,session_user_08582_001_a897f9,2025-04-27T09:37:47.720830,page_view,
user_06269,session_user_06269_003_101311,2025-04-27T09:38:09.720830,page_view,
user_03303,session_user_03303_002_e38361,2025-04-27T09:38:11.720830,page_view,
user_00668,session_user_00668_003_ab54a8,2025-04-27T09:38:12.720830,page_view,
user_07103,session_user_07103_001_ccb3a1,2025-04-27T09:38:15.720830,page_view,
user_03089,session_user_03089_005_d4984c,2025-04-27T09:38:31.720830,page_view,
user_05134,session_user_05134_002_104b67,2025-04-27T09:38:33.720830,page_view,
user_04578,session_user_04578_002_6efd2d,2025-04-27T09:38:37.720830,page_view,
user_06219,session_user_06219_004_f335c7,2025-04-27T09:38:42.720830,search_hot
user_01519,session_user_01519_003_629bfb,2025-04-27T09:38:44.720830,page_view,
user_02945,session_user_02945_002_461fcd,2025-04-27T09:38:44.720830,page_view,
user_08467,session_user_08467_003_f15974,2025-04-27T09:38:50.720830,page_view,
user_03282,session_user_03282_001_7cf3ae,2025-04-27T09:38:53.720830,page_view,
user_00753,session_user_00753_001_3786a0,2025-04-27T09:38:56.720830,page_view,
user_09625,session_user_09625_002_85861d,2025-04-27T09:38:58.720830,page_view,
user_02361,session_user_02361_004_4c2c32,2025-04-27T09:39:06.720830,page_view,
user_05196,session_user_05196_003_ce6887,2025-04-27T09:39:06.720830,page_view,
user_05149,session_user_05149_001_c7fccf,2025-04-27T09:39:08.720830,page_view,
user_01332,session_user_01332_004_c6da5c,2025-04-27T09:39:10.720830,page_view,
user_06029,session_user_06029_003_079a2a,2025-04-27T09:39:14.720830,view_booki
user_04954,session_user_04954_001_022212,2025-04-27T09:39:28.720830,page_view,
user_04392,session_user_04392_003_2afb2c,2025-04-27T09:39:40.720830,page_view,
user_01964,session_user_01964_002_66eae0,2025-04-27T09:39:50.720830,page_view,
user_00422,session_user_00422_003_4e4f7d,2025-04-27T09:39:59.720830,page_view,
user_01519,session_user_01519_003_629bfb,2025-04-27T09:40:10.720830,search_hot
user_06832,session_user_06832_002_122073,2025-04-27T09:40:16.720830,page_view,
user_06586,session_user_06586_001_abf017,2025-04-27T09:40:18.720830,page_view,
```

5. Υλοποιήστε ένα απλό καταναλωτή (`KafkaConsumer`) για την υποδοχή των μηνυμάτων μέσα από τον Kafka broker ώστε να ελέγξετε ότι η διαδικασία αποστολής και κατανάλωσης των δεδομένων από το Ερώτημα 1 γίνεται σωστά.

Χρήσιμοι σύνδεσμοι:

- Βιβλιοθήκη KafkaPython
- Οδηγίες εγκατάστασης Kafka
- Απλή υλοποίηση Kafka producers & consumers σε Python

Ερώτημα 2: Κατανάλωση και επεξεργασία με Spark

1. Εγκαταστήστε το Spark και την απαραίτητη βιβλιοθήκη PySpark σε Python.
2. Στη συνέχεια υλοποιήστε μια διεργασία σε Spark η οποία επεξεργάζεται τα εισερχόμενα δεδομένα και παράγει ένα Spark Dataframe το οποίο θα ανανεώνεται κάθε *N* λεπτά (π.χ. N=10) με τα ακόλουθα πεδία:
    - `time`: η χρονική στιγμή από την αρχή της εξομοίωσης
    - `destination_name`: το όνομα μιας πόλης
    - `search_volume`: το πλήθος αναζητήσεων για αυτή την πόλη
    - `bookings_volume`: το πλήθος κρατήσεων για αυτή την πόλη
    - `sales_volume`: ο συνολικός τζίρος κρατήσεων για αυτή την πόλη

Χρήσιμοι σύνδεσμοι

- Εγκατάσταση PySpark
- Κατανάλωση δεδομένων από Kafka και επεξεργασία JSON σε Spark
- Επεξεργασία JSON σε Spark

Ερώτημα 3: Αποθήκευση σε MongoDB

1. Εγκαταστήστε την MongoDB.
2. Κατεβάστε τον MongoDB Spark driver και ξεκινήστε το PySpark shell ώστε να τον ενσωματώνει.
3. Σε κατάλληλες συλλογές στη MongoDB, αποθηκεύστε τόσο τα ωμά δεδομένα όπως έρχονται στο Spark από τον Kafka, όσο και τα αποτελέσματα της επεξεργασίας. Για το σκοπό αυτό χρησιμοποιήστε τα foreach / foreachBatch output sinks του Spark.

Υλοποιήστε στη συνέχεια ένα ξεχωριστό python script με το οποίο κάνουμε query την MongoDB και απαντούμε στα ακόλουθα ερωτήματα:

1. Ποια πόλη είχε το μεγαλύτερο πλήθος κρατήσεων μεταξύ μιας προκαθορισμένης χρονικής περιόδου;
2. Ποια πόλη είχε το μεγαλύτερο πλήθος αναζητήσεων μεταξύ προκαθορισμένης χρονικής περιόδου;
3. Πόση ήταν η μέση διάρκεια παραμονής (από τις κρατήσεις) σε κάθε πόλη για μια προκαθορισμένη χρονική περίοδο;

Χρήσιμοι σύνδεσμοι

- Εγκατάσταση MongoDB (τοπικά ή σε δωρεάν hosted account – μέχρι 512Mb χώρος)
- Εγκατάσταση Mongo Spark Driver (τελευταία έκδοση) [και για Python]
- Ανάγνωση και εξαγωγή δεδομένων μεταξύ Spark και MongoDB
- Έτοιμο docker container με Spark, Mongo & Jupyter [κι εδώ]

Παραδοτέα

Για την εργασία θα πρέπει να παραδώσετε:

1) Τον κώδικα που γράψατε για την εκτέλεση όλων των ερωτημάτων **αποκλειστικά σε γλώσσα Python (αρχεία .py)**.

2) Αναφορά στην οποία περιλαμβάνονται
    a. Περιγραφή του data generator script
    b. Περιγραφή της διαδικασίας εγκατάστασης όλου του pipeline
    c. Ο σχεδιασμός της ΒΔ.
    d. Ερωτήματα DDL και εισαγωγής δεδομένων, με screenshots από την επιτυχή εκτέλεση των τελευταίων.
    e. Ερωτήματα DML για την ανάκτηση δεδομένων και παραγόμενα αποτελέσματα από την εκτέλεση ερωτημάτων.

Να χρησιμοποιηθεί αποκλειστικά το template υποβολής που επισυνάπτεται στην εκφώνηση.

Καταληκτική ημερομηνία υποβολής: 13/6/2025

Επικοινωνία

Για την επιτυχία σας στο project θα χρειαστείτε καθοδήγηση καθώς κι απαντήσεις σε ερωτήματα που ίσως δεν έχουν καλυφθεί στο παρόν κείμενο. Για απορίες μπορείτε να αποστέλλετε στο akomninos@ceid.upatras.gr

Διαθέσιμη υποδομή

Για τους φοιτητές που αντιμετωπίζουν δυσκολία εγκατάστασης του απαιτούμενου λογισμικού σε δικό τους υπολογιστή, προσφέρεται η ακόλουθη υποδομή σε εξοπλισμό του εργαστηρίου μας. Η διαθέσιμη υποδομή μπορεί να τροποποιηθεί κατά τη διάρκεια του εξαμήνου (θα ενημερώνεστε με σχετική ανακοίνωση στο eclass).

Kafka broker

- Μπορείτε να χρησιμοποιήσετε διαθέσιμο Kafka broker στη διεύθυνση **150.140.142.67:9094** (προσοχή στη θύρα).
- Το μηχάνημα είναι φυσικός υπολογιστής (όχι VM) Intel XEON 8-core, 16GB RAM, 256GB SSD.
- Για την αποφυγή προβλημάτων παρακαλώ χρησιμοποιείτε ως topic name το ΑΜ σας με το πρόθεμα SDMD (π.χ. **SDMD-10433393**).

Σε περίπτωση τεχνικού προβλήματος (αδυναμία πρόσβασης, μη ανταπόκριση, timeouts) επικοινωνήστε με τον κο. Κομνηνό μέσω email για την επίλυσή του.

Παράρτημα 1 – Πρότυπο αναφοράς άσκησης

Συστήματα Διαχείρισης Δεδομένων Μεγάλου Όγκου

Εργαστηριακή Άσκηση 2023/24

Ονομα	**Επώνυμο**	**ΑΜ**

Βεβαιώνω ότι είμαι συγγραφέας της παρούσας εργασίας και ότι έχω αναφέρει ή παραπέμψει σε αυτήν, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για το συγκεκριμένο μάθημα/σεμινάριο/πρόγραμμα σπουδών.

Έχω ενημερωθεί ότι σύμφωνα με τον εσωτερικό κανονισμό λειτουργίας του Πανεπιστημίου Πατρών άρθρο 50§6, τυχόν προσπάθεια αντιγραφής ή εν γένει φαλκίδευσης της εξεταστικής και εκπαιδευτικής διαδικασίας από οιονδήποτε εξεταζόμενο, πέραν του μηδενισμού, συνιστά βαρύ πειθαρχικό παράπτωμα.

Υπογραφή

___ / ___ / 2024

Υπογραφή

___ / ___ / 2024

Συνημμένα αρχεία κώδικα

Μαζί με την παρούσα αναφορά υποβάλλουμε τα παρακάτω αρχεία κώδικα

Αρχείο	Αφορά το ερώτημα	Περιγραφή/Σχόλιο
Erotima1.py	*1*	*Περιέχει όλα τα ερωτήματα για το ερ. 1*

Τεχνικά χαρακτηριστικά περιβάλλοντος λειτουργίας

[Τεχνικά χαρακτηριστικά φυσικού Η/Υ που χρησιμοποιήθηκε για την εργασία, αν χρησιμοποιήθηκε hosted υπηρεσία μπορείτε απλά να αναφέρετε αυτό αντί για τον πίνακα]

Χαρακτηριστικό	**Τιμή**
CPU model	*Intel i5-10400F*
CPU clock speed	*2.9GHz*
Physical CPU cores	*6*
Logical CPU cores	*12*
RAM	*16*
Secondary Storage Type	*~~HDD~~/SSD*

Ερώτημα 1: Παραγωγή δεδομένων

[περιγράψτε τη δημιουργία δεδομένων εξομοίωσης και την λογική του script παραγωγής δεδομένων ως kafka producer. Δώστε screenshots από την επιτυχή εγκατάσταση του Kafka, και τη δοκιμή της λειτουργίας του μοντέλου producer-consumer]

Ερώτημα 2: Κατανάλωση και επεξεργασία με Spark

[περιγράψτε τη λειτουργία του script κατανάλωσης και επεξεργασίας δεδομένων από το Spark, με κατάλληλα screenshots που δείχνουν την ορθή λειτουργία της διαδικασίας]

Ερώτημα 3: Αποθήκευση σε MongoDB

[παραθέστε: 1. Το σχεδιασμό της βάσης και τα statements δημιουργίας των συλλογών, 2. Τη λειτουργία του script αποθήκευσης των ωμών δεδομένων και των αποτελεσμάτων επεξεργασίας από το Spark στη MongoDB με screenshots των αποτελεσμάτων, 3. Την εκτέλεση του script επερωτημάτων προς τη MongoDB μαζί με τα queries που σχεδιάσατε, και τα αποτελέσματα της εκτέλεσης αυτών με κατάλληλα screenshots].

Σχολιασμός αποτελεσμάτων

[Συνοψίστε τα αποτελέσματα της εμπειρίας σας από το project.]

Βιβλιογραφία

[πηγές που χρησιμοποιήσατε για την εργασία]