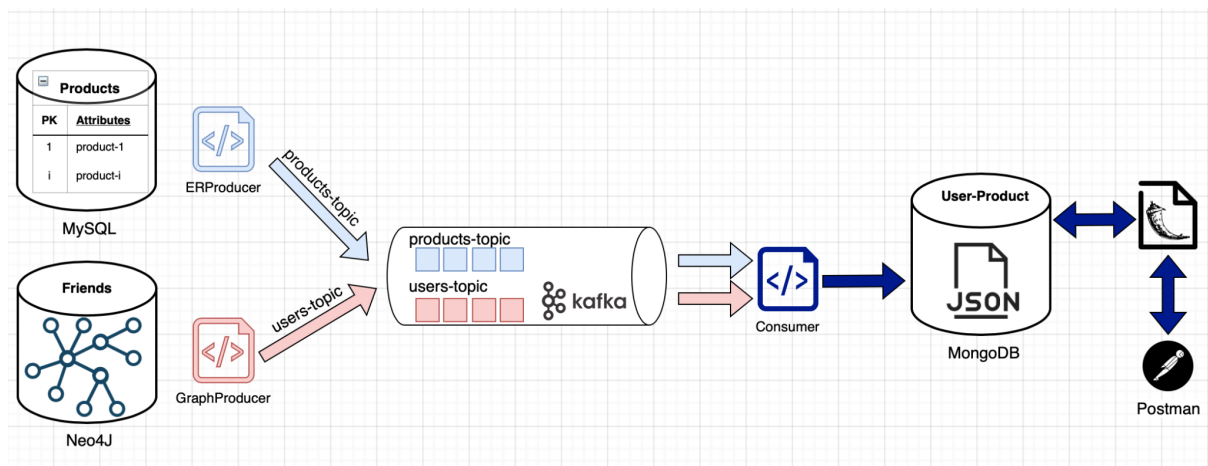




Δεύτερη εξαμηνιαία εργασία στον Προγραμματισμό Πλήρους Στοιβάς στον Παγκόσμιο Ιστό
(Deadline 30 January)

Extract Transform Load procedure

Στην παρούσα εργασία θα αναπτύξουμε έναν μηχανισμό Extract Transform Load (ETL) [1]. Ο ETL μηχανισμός θα ανακτά δεδομένα από δύο Βάσεις Δεδομένων (ΒΔ) διαφορετικού τύπου, θα τα ενοποιεί, επεξεργάζεται και τέλος θα τα αποθηκεύει σε μια τρίτη ΒΔ. Μια Rest υπηρεσία θα προβάλλει δεδομένα μέσω get request από την τρίτη ΒΔ. Η εργασία θα υλοποιηθεί σε Python 3. Οι δύο ΒΔ από όπου θα γίνονται extract τα δεδομένα θα είναι MySQL και Neo4J και η ΒΔ όπου θα γίνονται load τα δεδομένα θα είναι MongoDB. Το rest service θα υλοποιηθεί με το Flask Framework. Το data pipeline και integration θα υλοποιηθεί με τον Apache Kafka.



Use case:

MySQL

Θα δημιουργήσετε μια SQL database σε MySQL [2] η οποία θα περιέχει σε έναν πίνακα εμπορικά προϊόντα. Κάθε προϊόν έχει ως primary key ένα productID και μια σειρά από attributes που τα χαρακτηρίζουν.

Neo4J

Θα δημιουργήσετε μια graph database σε Neo4j [3] που θα περιέχει users ως nodes που συνδέονται αναμεταξύ τους με σχέσεις φιλίας. Οι σχέσεις φιλίας αναπαρίστανται με ακμές. Κάθε user έχει ως property τα products (productID) που έχει αγοράσει.



ERProducer

Θα δημιουργήσετε έναν kafka producer [4] που θα διαβάζει από την MySQL πλειάδες (products) [5] και θα τα κάνει publish στον kafka message bus στο topic “products-topic”. Ο ρυθμός που θα γίνεται το publish θα είναι δέκα πλειάδες ανα 20 δευτερόλεπτα.

GraphProducer

Θα δημιουργήσετε έναν kafka producer που θα διασχίσει τον γράφο που βρίσκεται στην Neo4J [6] και θα κάνει publish στο kafka message bus στο topic “users-topic” τους users (nodes). Ο ρυθμός που θα γίνεται το publish θα είναι πέντε nodes ανα 20 δευτερόλεπτα.

Consumer

Θα δημιουργήσετε έναν kafka consumer που θα κάνει subscribe τα topics “products-topic” και “users-topic” από το message bus και θα ενοποιεί (data fusion) [7] τα data των products και των user σύμφωνα με το productID. Τα καινούρια ενοποιημένα δεδομένα θα αναπαρίστανται μέσω json files και θα αποθηκεύονται σε μία MongoDB [8].

Flask

Θα δημιουργήσετε ένα Restful API μέσω Flask [9] που θα επικοινωνεί με την MongoDB. Θα δέχεται το αναγνωριστικό ενός χρήστη και θα επιστρέφει τα attributes από όλα τα products που έχει αγοράσει.

[1] https://en.wikipedia.org/wiki/Extract,_transform,_load

[2] <https://www.mysql.com/downloads/>

[3] <https://neo4j.com/download/>

[4] <https://kafka-python.readthedocs.io/en/master/>

[5] <https://pynative.com/python-mysql-database-connection/>

[6] <https://neo4j.com/developer/python/>

[7] https://en.wikipedia.org/wiki/Data_fusion

[8] <https://www.mongodb.com/languages/python>

[9] <https://flask.palletsprojects.com/en/2.0.x/>