

חלק א סעיף ב 3:

כאשר הנתונים זורמים למערכת, נשתמש בשני מבני נתונים מסוג מילון:

- **timestamp_map** – משמש לבדוק ולמנוע כפילויות של נתונים.
המפתח – ערך ה timestamp של הזוג שהתקבל בזרימה.
הערך – יהיה מילון שמכיל שני שדות –
1. value – הערכים שהתקבלו עבור אותו ה timestamp.
2. count – מספר הפעמים שאותו timestamp הופיע.
- **average_hour_map** – משמש לשמירה ועדכון הממוצע עבור כל שעה עגולה.
המפתח – מכיל שעה עגולה יחד עם התאריך עבור כל מפתח timestamp תקין המופיע ב timestamp_map.
הערך – יהיה מילון שמכיל שלוש שדות –
1. count – מספר הזוגות (value, timestamp) שקיימים בשעה זו.
2. sum – סכום כל ערכי ה value התקינים שקיימים בשעה זו.
3. average – הממוצע של כל ערכי ה value עבור שעה זו.

נשמור ונעדכן את הנתונים כך:

מכיוון שהנתונים עשויים להגיע במקביל (למשל ממספר חיבורים או תהליכים), נשתמש במנעול (lock) כדי להגן על הגישה למבנים timestamp_map ו- average_hour_map ולוודא שהנתונים לא ישתבשו כאשר ניגשים אליהם ממקביליות.

כל פעם שמגיע זוג חדש בזרימה נבדוק אם הוא תקין:

אם לא – לא מבצעים שום דבר וממשיכים הלאה.

אם כן – נבצע את הפעולות הבאות תחת מנעול (with lock):

- נבדוק אם הזמן שלו (timestamp) כבר הופיע קודם:
 - אם כן, כלומר אם מפתח זה כבר קיים במילון timestamp_map וזה כפילות, ולכן:
 - נגדיל במילון timestamp_map את ערך ה count ב-1.
 - אם לא – נוסיף אותו למילון timestamp_map ונשמור שם את הערך של הזוג הזה (value) ונעדכן את count ל 1 . count = 1
 - לאחר מכן, נבדוק:
 - אם במילון timestamp_map ערך ה- count=1 עבור אותו timestamp, כלומר קיבלנו בזמן הזה ערך יחיד ללא כפילויות ותקין ולכן:
 - נחשב את השעה העגולה מתוך ה- timestamp (כלומר, התאריך והשעה בלי דקות ושניות), וזה יהיה המפתח במילון average_hour_map.
 - אם זו הפעם הראשונה שהשעה הזו מופיעה – ניצור רשומה חדשה במילון:
 - count = 1
 - sum = value
 - average = sum/value
 - אם השעה כבר קיימת במילון – נעדכן את הנתונים:
 - count+=1
 - sum+=value
 - average = sum / count
- כך נעבור עבור כל שניה ונעדכן במקרה הצורך את timestamp ואת הערכים שלו. בצורה זו נוכל נעדכן את הממוצעים השעתיים בזמן אמת.