M מספר השורות, P מילים בשורה

קריאת כל הקובץ לזיכרון כאשר פותחים אותו הוא O(M)

מעבר על כל הקובץ בלולאת while, O(M/10,000)=O(M)  
מעבר על כל שורות של כל חלק – זאת אומרת 10,000 שורות, ומיד לאחר מכן מעבר על כל המילים בכל שורה בפונקציה frequency\_error - .O(M\*P) מכיוון P זניח לעומת M סיבוכיות הזמן תהיה O(M) + חיבור למילון הסופי בכל איטרציה O(1)

מיון של רשימת השכיחויות במקרה הגרוע יקח O(MlogM)  
   
הכנסה לתוך רשימה את N השגיאות השכיחות ביותר – O(M)

מכאן שסיבוכיות הזמן תהיה O(MlogM)  
  
כל חלק נשמר ברשימה lines- O(M)  
מילון total\_error\_counts שמכיל עד M קודי שגיאה O(M)

מילון error\_codes שמכיל עד M קודי שגיאה O(M)

רשימה result שמחזירה N שכיחויות O(N)

סיבוכיות המקום תהיה O(M)

כאשר הנתונים מגיעים בזרימה, לא נוכל להפעיל את חישוב הממוצעים בפעם אחת על כל הקובץ, אלא נצטרך לדאוג שתוך כדי הזרימה של הנתונים, הנתונים ישמרו במבנה נתונים מסויים כך שבכל פעם שנרצה להוציא ממוצע בזמן אמת, נוכל להפעיל את הפונקציה על מבנה הנתונים ונקבל את הפלט הרצוי.  
מבנה נתונים יעיל לפתרון הוא מילון כך שכאשר יגיע נתון בזמן אמת, נחשב את השעה העגולה של התאריך ולאחר מכן נשמור במילון את הvalue שלו לפי המפתח timestamp ששמור כבר כשעה עגולה, נחשב את הממוצע כאשר נכניס את הvalue למילון, כך שבכל מפתח timestamp יהיה הממוצע לשעה שנוכל לשלוף בכל עת.

היתרונות בפורמט parquet הן שהנתונים בקובץ נשמרים לפי עמודות ולא לפי שורות כמו שאר הפורמטים, ולכן אפשר לעבור על העמודות בדרך מהירה ויעילה יותר, בלי צורך לעבור על כל אורך השורות. בנוסף מכיוון שבקובץ Parquet הנתונים בעמודות הם מאותו הסוג, קל יותר לדחוס את הקובץ הזה לפי הטיפוסים השווים, כלומר יהיה חיסכון בזיכרון ובזמן הקריאה והכתיבה של הקובץ. עוד יתרון הוא שנשמרים טיפוסי הנתונים של כל עמודה כבר בקובץ עצמו. הפורמט גם תומך בבדיקות שעוזרות למצוא שגיאות ולהתאושש מהם בקלות. בנוסף הפורמט עוזר גם בסינון המידע על ידי שאילתות כך שאפשר למצוא נתון ספציפי במהירות וביעילות.