Moshav Bnei Zion P.O.B 151, Israel 60910, Tel: 972-9-7907000 Fax: 972-9-7442444

Operational Data Store- ODS עדכונים בתחום

פיני כהן

STKI

סיכום מנהלים

- תפיסת ה- ODS רווחת בארגוני enterprise רבים.
- רובם המכריע של ה- ODS מתבססים על טכנולוגיה של מסדי נתונים רלציונים.
- כעת ניתן לשקול בניה\העברה של ODS למסדי נתונים noSQL.

נייר עמדה

המונח ODS (operational data store) משמש ארגונים בסיטואציות שונות אולם הכוונה הרווחת בשימוש במונח זה הנה "גישה לנתונים (בהקשרים תפעוליים, בדרך כלל read) שמתבצעת לתוך אזור נתונים ספציפי (ולא למערכות המקוריות ששם נוצר ונמצא המידע)".

המידע מהמערכות התפעוליות מועבר ל- ODS באופן מהיר[1] (real time או near real time). בדרך כלל על ידי טכנולוגיות CDC (change data cupture) או dbms replication.

STKI מעריכה של-20% מארגוני ה- enterprise ישנו אזור נתונים המוגדר כ- ODS ואולם בפועל כמעט בכל ארגון enterprise יש אזור נתונים, אפילו אם קטן, אשר ממש קונספט זה. לדוגמה, בארגונים רבים חלק מהנתונים בשכבת הדיגיטל (web או מובייל ללקוחות הארגון) אינה מציגה נתונים ישירות מהמערכות התפעוליות אלא ממקום אשר מהווה העתק של מידע זה, וזהו מימוש (אפילו אם קטן) של ODS.

במובן מסוים מהוה ה- ODS מימוש רוחבי של דפוס ארכיטקטוני Command Query Responsibility Segregation[2] שמשמעותו הפרדת הקריאה והכתיבה.

ישנם מספר יתרונות בקונספט ארכיטקטוני זה:

1. לא מעמיסים על מערכות המקור כאשר נגשים לנתונים.
2. העמסת נתונים על מערכות המקור עלולה לגרום לעלויות כבדות (לדוגמה גידול בעלויות MIPS). ה- ODS חוסך גידול זה.
3. ניתן להמשיך ולקיים תהליכים עסקיים כאשר מערכות המקור אינן פועלות. כתוצאה מכך לא חייבים לספק את רמת השרידות והיתירות בכל המערכות גם יחד.
4. ה- ODS הנה תשתית מהירה ויציבה ומכילה את הנתונים בפורמט המתאים למערכת היעד ובכך מקבלים שיפור בביצועים של מערכת היעד.
5. תשתית אחידה לנתונים תפעוליים ובכך יש מניעה של פערי נתונים.
6. שיפור ההגנה על המידע התפעולי במיוחד כאשר יש גישה לנתונים מחוץ לארגון.

---

[1] הערה: real time לעיתים גורם לאתגרים עסקיים. לדוגמה אם רוצים להציג ללקוח יתרה באופן מידי אך חישוב הטרנזאקציה הסופי מתבצע רק בלילה (כי מסתמך על batch), חייבים להוסיף ש"החישוב אינו סופי".

[2] ראה https://martinfowler.com/bliki/CQRS.html וגם https://docs.microsoft.com/en-us/azure/architecture/patterns/cqrs

7. הפחתת התלות בין המערכות – שיפור הגמישות.

לעיתים ה- ODS מקבל את הנתונים מה- DW3 (ובכך מוריד את העומס על ה- DW). לעיתים להיפך, ה-ODS משמש ש- staging area לנתונים לתוך ה-DW.

בדרך כלל ה- ODS יכיל נתונים לתקופת זמן ספציפית ולא את כל הנתונים. זאת בכדי שלא לגרום לכך שה- ODS "יתנפח" ויהיה איטי.

מכיוון שהמידע ב- ODS אמור להתאים למערכת המקבלת אותו (מבחינת פורמט ותוכן) לעיתים תכופות מבצעים מניפולציות על המידע שמגיע ל- ODS לפי הצורך המתאים במערכות היעד ולא טוענים אותו as is. במידה וישנם פורמטים שונים הנדרשים על ידי כמה מערכות יעד, ניתן לראות מצבים שיהיה שכפול של המידע למספר פורמטים שונים. כל זאת בכדי להגיע למהירות הגבוהה ביותר.

מסורתית, ארגונים בנו את ה- ODS על בסיס מסדי נתונים רלציונים כאשר הנתונים מעודכנים (כאמור) על ידי CDC או dbms replication (פחות על ידי ETL שמתבצע בתדירות פחות גבוהה).

הגישה ל- ODS, כלומר צריכת הנתונים מתוך ה- ODS התבצע על ידי תשתית הקישוריות הארגונית, ה-esb, שיותר ויותר מתבססת על rest.

עד כאן הגישה המסורתית. לאחרונה ישנה הרחבה של ה-ODS אשר יכול כעת להתבסס על מסד נתונים noSQL (ולא רלציוני). ה-noSQL הנו key-value store אשר מאפשר שליפה מהירה ב-rest (כי הנתונים נמצאים במסד הנתונים ב- xml\json ולכן חוסכים המרה שמתבצעת כאשר מאחסנים את המידע במסד נתונים רלציוני) ולא פחות חשוב, מאפשרים מבנה רשומה (מבנה collection) משתנה – ללא שינוי סכמה. לדוגמה לאחר ש-ODS שמכיל נתוני לקוח פעיל, מחליטים להוסיף תוכן נוסף "תקציר chat אחרון שהתבצע", וה-noSQL מאפשר הוספה זו ללא שינוי סכמה (מה שבלתי אפשרי במסד נתונים רלציוני).

ישנם סוגים רבים של מסדי נתונים noSQL ולכל אחד יתרונות וחסרונות. חלק מהדברים הרשומים בהמשך לא נכונים לגבי כל משפחת ה- noSQL.

אולם, "אין ארוחות חינם", ול-noSQL ישנם גם חסרונות, לדוגמא אם רוצים לשנות את אופן השליפה, יש לבנות את הטבלה מחדש, דבר שבמסד נתונים רלציוני אפשר לעיתים לשנות באמצעות כתיבת sql חדש ושימוש ב- joins שלא אפשריים ב- noSQL. כלומר, התכנון מראש חשוב לא פחות ב- noSQL מהאופציה הרלציונית .

ה-noSQL גם לא מאפשר ביצוע סיכומים או פעולות מתקדמות (כגון "כל הלקוחות ללא מספר טלפון נייד" – left\right join) שמתבצעים on line ולכן אם המערכות התפעוליות נדרשות בהצגת מידע באופן זה, ה- noSQL לא יתאים.

---

[3] אולם במקרה זה, כאשר ה- ODS ניזון מה- DW , המשמעות היא שהנתונים ב-ODS אינן עדכניים-זהים לנתונים במערכות התפעוליות

במידה ובוחרים ב- noSQL יש לשים לב שמסד הנתונים יתמוך בשפות התכנות, יכיל את ה- drivers המתאימים ויתאים למערכות ההפעלה הנדרשות.

מבחינת הארכיטקטונית ישנם דרכים רבות לבנות את ה- ODS. להלן דוגמה יושמה על ידי אחד מהלקוחות. ברמה הארכיטקטונית נבנה ה- ODS משתי סביבות של שרתים העובדים כ- Active Active. שרת אחד הוא ה- ODS המשרת את צרכני המידע. השרת השני הוא Pre-ODS אשר מבצע עיבודים כגון טרנספורמציות של קודים (כל הקודים הופכים להיות אחידים) ומבצע את הלוגיקות המוכנות שבהן ה- CRM ומערכות היעד האחרות ישתמשו. ה- Pre ODS – משמש גם ל- BATCH, כלומר מערכות ה- BATCH ניגשות ל- preODS או ל- DW. ה- preODS עובד כאשר ה- ODS למטה.

חלק מהארכיטקטורה מדברת על החלפת השרתים בין ה- ODS ו- Pre-ODS . כלומר מבצעים filp-flop בין השרתים. ארכיטקטורה זו בודקת כל הזמן את נושא היתירות של המערכת.