

ניהול תנועות
חלק 4

*Transaction
Management
Part 4*

בקרת מקביליות על ידי חותמות זמן

Timestamp-Based
Concurrency Control

בקרת מקביליות המבוססת על חותמות זמן Timestamp-Based Concurrency Control

במקום לבצע את הבדיקה לקראת סיום העיסקה,
מבצעים בדיקה בכל פעם שהעיסקה קוראת או
כותבת

עיסקה מקבלת חותמת זמן כאשר היא מתחילה

מטרת הבדיקות לוודא שהתזמון הנוצר בפועל
הינו שקול לתזמון הסדרתי שבו העסקאות
מתבצעות לפי הסדר של חותמות הזמן

▶ "כאילו" כל עיסקה מתבצעת בצורה אטומית מיד
כשהיא מתחילה

זמן קטן יותר פירושו זמן מוקדם יותר

גם לכל פריט יש חותמות זמן

לכל פריט A במסד יש שתי חותמות זמן 

▶ $RTS(A)$ הנה חותמת זמן המציינת מתי הפריט נקרא לאחרונה (ליתר דיוק, הזמן של העיסקה הצעירה ביותר שקראה את A)

▶ $WTS(A)$ הנה חותמת זמן המציינת מתי הפריט נכתב לאחרונה (ליתר דיוק, הזמן של העיסקה הצעירה ביותר שכתבה את A)

בעקרון, עיסקה T שקוראת פריט A משנה את $RTS(A)$ ל- $TS(T)$ אם $RTS(A)$ קטן מ- $TS(T)$ 
באופן דומה, עיסקה T שכותבת פריט A משנה את $WTS(A)$ ל- $TS(T)$ אם $WTS(A)$ קטן מ- $TS(T)$ 

מה צריך לבדוק?

עיסקה T לא יכולה לקרוא פריט שכבר נכתב על ידי עיסקה אחרת, שאמורה להתבצע אחרי T במילים אחרות, כשעיסקה T רוצה לקרוא פריט A

אם $TS(T) < WTS(A)$, אז צריך לבטל את T ולהתחיל אותה מחדש (עם חותמת זמן חדשה) ← מדוע?

אחרת T קוראת את A ו- ▶

$$RTS(A) := \max\{ TS(T), RTS(A) \}$$

מה עוד צריך לבדוק?

עיסקה T לא יכולה לכתוב פריט שכבר נקרא
על ידי עיסקה אחרת, שאמורה להתבצע אחרי
 T

במילים אחרות, כאשר T רוצה לכתוב פריט A

אם $TS(T) < RTS(A)$, אז צריך לבטל את T
ולהתחיל אותה מחדש (עם חותמת זמן חדשה)

← מדוע?

אם $TS(T) \geq RTS(A)$ וגם $TS(T) \geq WTS(A)$, אז
 T כותבת את A ו- $WTS(A) := TS(T)$

ומה קורה אם $TS(T) \geq RTS(A)$
וגם $TS(T) < WTS(A)$?

❖ לא עושים כלום והעיסקה יכולה להמשיך, מבלי לכתוב
ערך חדש עבור A או לשנות את $WTS(A)$

❖ אפשר להמשיך בלי לעשות כלום כי השינוי ש- T
הייתה אמורה לבצע בערכו של הפריט A הינו חסר
משמעות בתזמון הסדרתי השקול, בגלל סיבות אלה:

▶ עיסקה T' שהיא מאוחרת מ- T (כלומר, מופיעה אחרי T
בתזמון הסדרתי השקול) כבר כתבה את A (ולמעשה
עשתה זאת בצורה עיוורת)

▶ והעיסקה הכי מאוחרת T'' שכבר קראה את A מופיעה לפני
 T בתזמון הסדרתי השקול (או שזו T בעצמה)

❖ כלל זה ידוע בשם Thomas Write Rule

איך יודעים שהכתיבה של A על ידי T' היא כתיבה עיוורת?

$TS(T')$ שווה ל- $WTS(A)$, כי T' היא הכי מאוחרת שכבר כתבה את A 🦠

אילו T' גם קראה את A , אז היה מתקיים
 $TS(T) < RTS(A)$, כי $TS(T) < WTS(A)$ 🦠

אבל נתון ש- $TS(T) \geq RTS(A)$ 🦠

הנקודה החשובה

הנקודה החשובה אינה שהייתה כתיבה עיוורת ע"י T' ,
אלא שהכתיבה ש- T רוצה לבצע עכשיו הינה מיותרת
(קרי, חסרת השפעה) בתזמון הסדרתי השקול

היא מיותרת כי $RTS(A) \leq TS(T) < WTS(A)$, כלומר כל
העסקאות שקראו את A עד כה הן קודמות ל- T (או T
בעצמה) ובנוסף הערך הנוכחי של A נכתב ע"י עיסקה
מאוחרת מ- T

אם בעתיד עיסקה תרצה לקרוא את A ולגביה הכתיבה
של T אינה מיותרת, אז עיסקה זו לא תעבור את
הבדיקה (ולכן תתבטל ולא תהיה חלק מהתזמון
הסדרתי השקול)

האם זה אומר שלמעשה פרוטוקול זה מבטיח בר
סדרתיות קונפליקטית?

צריך למנוע קריאה מלוכלכת

עיסקה T שכותבת פריט A צריכה לחסום עסקאות אחרות שרוצות לקרוא את A עד ש-T מתחייבת אם הדבר לא נעשה, אז יכולות להיות קריאות מלוכלכות, שעלולות לגרום למפל הפלות ולתיזמון שאינו מאפשר התאוששות

בעיה דומה נוצרת ע"י Thomas Write Rule
נניח שהחוק מופעל, כלומר הכתיבה של פריט A ע"י עיסקה נתונה הינה מיותרת, כי עיסקה מאוחרת יותר כבר כתבה את A בצורה עיוורת

אבל מה יקרה אם העיסקה המאוחרת יותר תתבטל?
מסקנה: עיסקה T שכותבת פריט A צריכה לחסום עסקאות אחרות שרוצות לכתוב או לקרוא את A החסימה עד ש-T מתחייבת

החסימה פירושה מנעול בלעדי

המשמעות המעשית של החסימה המתוארת בשקף 
הקודם היא שעיסקה T מחזיקה מנעול בלעדי על
פריט A מהרגע ש-T כותבת את A ועד שהיא
מתחייבת

המנעול הבלעדי מונע לא רק קריאה או כתיבת ערך 
חדש, אלא גם בדיקה של חותמות הזמן של הפריט
(כדי לדעת אם בכלל צריך לכתוב ערך חדש)
כך מונעים בעיה כתוצאה מ- Thomas Write Rule ►

דרך אחרת למימוש החסימה

להלן דרך אחרת לתאר כיצד מתבצעת החסימה

לכל פריט A יש ביט בודד C , שערכו ההתחלתי הוא $true$, ומתקיימים הכללים הבאים

▶ כשעיסקה כותבת את A היא משנה את C ל- $false$

▶ כשעיסקה מתחייבת, היא משנה ל- $true$ את הביט C של כל הפריטים שהיא כתבה

▶ עיסקה שרוצה לבצע פעולה כלשהי על פריט A מתעכבת עד שהביט C של הפריט A הוא $true$

השימוש בביט עדיף על נעילות

מנגנון נעילות צריך לנהל באופן מרכזי, בעוד
שהשימוש בביט מאפשר לכל עיסקה לפעול
באופן עצמאי

הבדל זה משמעותי במיוחד במערכות מבוזרות

עיסקה יכולה לקרוא את חותמות הזמן של
פריט גם כאשר הביט הוא false

כך העיסקה יודעת מיד אם היא צריכה להתבטל
(בהנחה שהעיסקה שכרגע מעכבת אותה תתחייב)
או צריכה לחכות עד שהביט ישונה ל-true, כדי
לבצע קריאה, כתיבה או הפעלת הכלל של תומאס

הערה

כאשר עיסקה מתבטלת, צריך לגלגל לאחור לא רק את ערכי הפריטים שכתבה, אלא גם את חותמות הזמן שלהם, כדי שעסקה שעוכבה בגלל הכלל של תומס תוכל לבדוק האם (בעקבות הביטול) עליה לבצע את פעולת הכתיבה