

לכוונן או לקנות – הנה השאלה!

ארגונים רבים שבחרו לשדרג את החומרה במקום לכוונן את מערכת ה-CICS עשויים למצוא עניין רב ב-ICPU/CICS - תוכנת דגימה חדשה אותה ניתן להריץ בתקורה נמוכה וקבועה בסביבת הייצור, תוכנה המאפשרת לזהות תוכניות בזבזניות לפני שהן מגיעות לייצור

ורנר צולי, חברת לוג-און

מה הייתם עונים לו היו שואלים אתכם האם מערכת ה-CICS שלכם מכווננת? חישבו על כך לרגע. לא, לא מדובר כאן על זמן תגובה. האם יש סיבה בכלל לכוונן את המערכת כאשר זמן התגובה מצוין ו-90% מהטרנזקציות מסתיימות בפחות משניה. בכל פעם שהמעבד במחשב שלכם מגיע לרף העליון – האם לא כלכלי יותר לקנות חומרה נוספת, במיוחד כעת, שמחירי החומרה כה נמוכים? אכן, בפועל ארגונים רבים בוחרים לשדרג את החומרה במקום לכוונן את מערכת ה-CICS – הקנייה, הם אומרים, פשוטה יותר מאשר כוונן שהוא תהליך מחזורי הדורש ידע טכני רחב, זמן, עצבים, בדיקות ועוד. אך האם זו הגישה הנכונה?

משתני מערכת ה-CICS

הבה ונתבונן בכוונן ברמת ה-CICS, לדוגמה הפרמטר ICV. ערך ברירת המחדל של הפרמטר הוא 1,000 מילישניות, המורה ל-CICS לבדוק כל שנייה האם ממתנה עבודה לביצוע. אבל CICS יתעורר בין כה וכה באשר ממתנה עבודה. לכן, אם נניח שמדובר ב-CICS non-swappable, אזי על ידי שינוי הפרמטר ICV ל-20,000 מילישניות, נשחרר משאבי מעבד ללא פגיעה בזמן התגובה. אין ספק שכוונן מסוג זה הוא משתלם, הזמן הדרוש לבצע את השינוי הוא מזערי ויתר על כן, זהו תהליך חד פעמי שחוסך זמן מעבד מרגע זה ואילך. דוגמה זו מוכיחה שהגישה המעדיפה שידרוג החומרה אינה אופטימלית בהכרח.

כוונן אפליקציות

עד כאן כוונן ברמת המערכת, אך אנו רוצים להתרכז בכוונן ברמת האפליקציה - כוונן שהוא קשה יותר לביצוע אבל בעל מאפיינים דומים לדוגמה שהובאה כאן. כוונן אפליקציה עשוי לגרום לשינוי דרמטי בהתנהגות מערכת ה-CICS, וכפי שהניסיון מלמד התועלות בטווח הקצר ובטווח הארוך, מאפליקציות מכווננות הן בעלות ערך משמעותי והעלות והמאמץ שהושקעו על מנת להגיע ליעד זה מצדיקים את עצמם. אולם, הנושא אינו טריוויאלי וכלי הניתור הקיימים מתרכזים במתן מידע ברמת הטרנסקציה, משום שזמן התגובה הוא המאפיין העיקרי של מערכות online. מידע זה מאפשר זיהוי בעיות זמן תגובה בחומרה, ובתוכנת המערכת אותן ניתן לפתור על ידי שידרוג ושינויים תשתיתיים. אך, מה ניתן לעשות במקרים בהם ישנו חשד כי התוכנית גורמת לבעיית ביצועים, או כאשר רוצים לוודא שתוכנית כתובה בצורה לקויה לא תועבר לסביבת הייצור?

הסטטיסטיקות של CICS

ה-CICS אכן מספק מידע אודות התוכניות והמשימות שרצות, וניתן לעשות שימוש במידע זה, לנתח אותו ולהגיע לתוכניות הבעייתיות. נוכל גם לקבל מה-CICS סטטיסטיקות כל שעה ולמצוא את התוכניות שרצות במשך הזמן הארוך ביותר, לבצע להן סקר קוד, לחפש פקודות שגורמות לניצול גבוה של המעבד. זוהי שיטה אפשרית, אך בה בעת איטית ומסורבלת וברוב המקרים לא נדע לזהות את קטע הקוד הכתוב בצורה לקויה ואשר החלפתו מהווה פוטנציאל לשיפור בביצועים.

איתור פקודות בעייתיות

ה-CICS מספק מידע גם באמצעות ה-Auxiliary Trace. אך על מנת להפיק את התועלת המרבית יש לדעת אילו נקודות Trace להפעיל – כלומר להריץ את הטרנזקציות שהתוכנות החשודות מעורבות בהן, להדפיס את הדיווח של ה-Auxiliary Trace באמצעות תוכנת שירות של CICS ואז לעבור על הפקודות שהורצו בטרנזקציה בכדי לאתר את הפקודות הבעייתיות. גישה זו משתלמת כאשר יש לנגד עינינו תוכנית חשודה ונדרש חיפוש מעמיק יותר לתוכה. אבל מה עלינו לעשות אם אנו שרויים לגמרי בעלטה. זה הרי לא מעשי להריץ Auxtrace על מערכת CICS שלמה.

יצוין כי ישנם גם כלי ניפוי זמינים שניתן להשתמש בהם בכדי לעקוב צעד אחר צעד על מהלך תוכנית ולמצוא לולאות מיותרות אך גם זו גישה טובה רק במידה ויש לנו תוכנית מועמדת המצדיקה מבט מקרוב. אין זה פרקטי אם אנו מבקשים לקבל מבט-על על כל סט התוכניות.

הקלטה והרצה חוזרת

יש ארגונים המשקיעים זמן ומאמץ בהגדרת תמהיל טרנזקציות טיפוסי למערכת CICS ספציפית. גישה זו מאפשרת להריץ את התסריט יותר מפעם אחת ולעשות שימוש בתוכנה למדידת ביצועים כדי לבדוק האם הגרסה תספק את הביצועים הנדרשים. על מנת להכין סביבה כזו של הקלטה והרצה חוזרת יש צורך במאמץ ראשוני גדול ובתחזוקה מתמדת. כמו כן, קיימת בעיית דיוק בתמהיל טרנזקציות טיפוסי ובמידת ההצלחה שלו לשקף סביבת ייצור אמיתית. מגבלות נוספות הן הצורך להריץ סביבה כזו באופן מבודד, לוודא שהתוכניות זהות לסביבת הייצור, להגדיר את פרמטרי ה-CICS בדיוק כמו בייצור ועוד.

דגימה מסורתית מול גישה חדשה

תוכנות דגימה מסורתיות מאפשרות הרצתן בסביבת הייצור לפרקי זמן קצרים מאוד ובעלות גבוהה מאוד של משאבי המערכת ושל מערכת ה-CICS עצמה. מסיבה זו לא נהוג להריץ תוכנות דגימה בשעות השיא - אף אחד לא ממש יכול להרשות לעצמו את התקורה המתלווה לכך. עתה דמיינו שבידכם מצויה תוכנת דגימה אותה ניתן להריץ בתקורה נמוכה וקבועה בסביבת ייצור. תוכנה המצביעה על התוכניות הכי תובעניות מבחינת זמן מעבד ועל הנקודות המדויקות בתוך התוכנית שבו זמן המעבד נצרך. יעדים אלו הם שהובילו לפיתוח המוצר ICPU/CICS. עם מוצר זה יחדית ה-IT בארגון יכולה לקבל את תמהיל הטרנזקציות האמיתי, אפילו בזמנים העמוסים ביותר ביממה. עם ICPU/CICS ניתן לקבל את שמות התוכניות שצורכות את זמן המעבד הגבוה ביותר, ללא חשיבות איזו טרנזקציה משתמשת בהן. ניתן אף לזהות את הפרוצדורות בתוכניות הצורכות הכי הרבה זמן מעבד, ואת שרותי ה-CICS המופעלים על ידי התוכניות הללו. כל המידע הזה זמין מיידית או ב-Batch לאנליזה נוספת (for catching off-time sampling) – זאת ועוד ב"תשלום" זעום של 5% תקורה לכל היותר.