

מבוא לבדיקות עומסים

מורכבותם של יישומי אינטרנט, אפליקציות מובייל מודרניים, עבודה בענן והמספר הגדל של המשתמשים, מעלה מדרגה את נושא הבדיקות בכלל ובדיקות ביצועיים בפרט. עם הגידול במהירות הגישה, משתמשים אינם מוכנים עוד להשלים עם מהירות תגובה איטית או חוסר יציבות של היישום. כתוצאה מכך אנו עדים לסטנדרטים גבוהים ולדרישה מוגברת לפיתוח יישומים ואתרים מהירים ויעילים יותר, וכן ניצול מרבי של החומרה ודחיפת הביצועים עד גבול היכולת.

לכן, לא ניתן עוד להסתפק בבדיקות פונקציונאליות בלבד ויש צורך בבדיקות ביצועיים בה יבדקו זמני התגובה והיציבות של המערכת תחת העומס של המשתמשים. בדיקות ביצועיים הוא נושא רחב, שתחתיו נכנסים סוגי בדיקות שונים, כגון בדיקות עומסים, בדיקות יציבות, בדיקות השוואתיות, בדיקות ביצועיים של קוד (profiling) וכו'. במאמר זה נדבר על בדיקות עומסים שהם חלק חשוב מבדיקות ביצועיים של תוכנה.

מה זה בדיקת עומסים?

בדיקת עומסים היא שיטה למידול השימוש הצפוי בתוכנה, באמצעות סימולציה של גישה בו-זמנית של משתמשים רבים אל שרותי התוכנה. סימולציה זו נעשית על ידי כלים אוטומטיים המסוגלים לדמות מספר רב של משתמשים. בדיקות עומסים עוזרות להבין זמני תגובה של המערכת ומסייעות באיתור צווארי בקבוק ונקודות כשל.

למה חייבים לבצע בדיקות עומסים?

לא מעט חברות וארגונים מוותרים על בדיקות עומסים, מתוך מחשבה שאם המערכת תפקדה כראוי בזמן בדיקות ידניות, היא תתפקד בהתאם גם בסביבת הייצור. כמו כן, אנשי ה-IT אוהבים לטעון שאם המערכת תהיה איטית בייצור אז הם יוסיפו עוד שרתים וזה יפתור את כל הבעיות.

כמובן שכל זה לא נכון - ללא ביצוע בדיקות עומסים על מערכות מרובות משתמשים, לא ניתן לגלות בעיות ביצועיים מהותיות ולתקן מבעוד מועד. המערכת עלולה לקרוס, לא לתפקד או להיות איטית - כל זה יכול לגרום נזק ממשי למשתמשים ולארגון.

אחד מהדוגמאות הבולטות משנים האחרונות היא נפילה של אתר Obamacare - אתר שמאפשר לציבור אמריקאי להירשם לביטוח בריאות בארה"ב אחרי הרפורמה של אובמה. אתר זה לא נבדק כראוי וקרוס ביום הראשון להפעלתו. בניית האתר עלתה אז כ-400 מיליון דולר ובדיקות האתר, כולל בדיקות עומסים לא בוצעו כראוי. לקח חודשים להפעיל את האתר מחדש, אבל כמובן הפגיע שמשמעותית ביותר הייתה בתדמיתו של נשיא ארה"ב ומפעיליו של האתר.

כמו במקרים האחרים, גם במקרה זה הוכח שהוספת חומרה לרוב לא תעזור למנוע או לפתור בעיות ביצועיים מהותיות ורק תגרום לבזבז נוסף של זמן וכסף. לכן חייבים לבצע בדיקות עומסים כחלק אינטגרלי מבדיקות איכות של המערכת!

מתי מבצעים בדיקות עומסים?

בדיקת עומסים מצריכה מערכת יציבה ובדוקה היטב ולכן לרוב מבצעים בדיקות עומסים בשלב טרום ייצור או בשלבים מאוחרים של תהליך הפיתוח. בנוסף, מומלץ לבצע בדיקת עומסים בסביבה דומה ככל שניתן לסביבת הייצור - דבר שבדרך כלל זמין לקראת עליה לאוויר של המערכת.

כמובן שביצוע בדיקות עומסים זמן קצר לפני עליה לאוויר מעלה סיכון ולעיתים אף עלול לגרום לעיכובים משמעותיים. לכן מומלץ להשתדל לבצע בדיקות עומסים בשלבים מוקדמים יותר של הפיתוח. מומלץ להתחיל את בדיקות עומסים על חלקים יציבים יותר של המערכת ואז לקראת הסוף לבצע בדיקה כוללת של כל החלקים על המערכת השלמה. אפשר גם לשלב תהליך בדיקות עומסים עם Continues Integration ואף Unit Testing. ככל שמבצעים בדיקות עומסים בשלבים מוקדמים יותר, כך יכולת להבין

מהות הבעיה ולתקן אותה היא קלה ומהירה יותר. כמו כן, מחיר תיקון התקלה בשלבים מוקדמים יהיה נמוך משמעותית ממחיר התיקון רגע לפני העליה לאויר, או גרוע מכך בסביבת הייצור.

איך לתכנן בדיקת עומסים?

תכנון של בדיקת עומסים נגזר מדרישות הביצועים של המערכת. לכן כדאי להתחיל לתכנן בדיקת עומסים יחד עם הגדרת דרישות הביצועים של המערכת. בלא מעט מקרים, דרישות הביצועים לא מוגדרות מראש בשלב התכנון ולא מופיעות בדרישות המערכת. זה עלול לגרום לוויכוחים אין סופיים וגלגול אחריות בשלבים מאוחרים יותר בין גורמים שונים בארגון, בהם אנשי פיתוח, QA, IT, מנהלי מוצר, לקוח סופי ועוד רבים אחרים. כדאי למנוע מצב זה ע"י חשיבה משותפת של כל הגורמים הרלוונטיים והגדרה מסודרת של הדרישות עוד בשלב הגדרת הדרישות לכל המערכת הנבדקת.

דרישות ביצועיים בדרך כלל כוללים דברים כמו כמות משתמשים מקסימלית שהמערכת אמורה לתמוך, הגדרה של זמני תגובה מקובלים, עמידה בפיקים, הגדרת יכולת גדילה של המערכת (scale) וכו'. לאחר שדרישות אלו הוגדרו, אפשר להתחיל בתכנון של בדיקות העומסים. לרוב התכנון יכול את הדברים הבאים:

- **תסריטי בדיקה** – פירוט של תסריטי הבדיקה (הרחבה להלן).
- **סוגי הבדיקות** – סוגי הבדיקה הרלוונטיים לדרישות (הרחבה להלן).
- **מודל העמסה** – כמות משתמשים מרבית, זמן עליית המשתמשים ומשך הבדיקה בהתאם לסוגי הבדיקות. בניית מודל העמסה המייצג את המציאות הוא לעיטים משימה הכי קשה בבדיקה. בבסיסו של המודל עומד הגדרת כמות משתמשים המקסימלית שהמערכת צריכה לתמוך. אך צריך גם לקחת בחשבון נתונים כמו כמה זמן משתמש מבלה עושה שימוש רציף במערכת (session time), בכמה שעות ביום יש פעילות מוגברת ובכמה שעות אין כמעט כל פעילות. במקרה של מערכת קיים, כדאי להסתכל על נתונים של בגוגל אנליטיקס או כלים דומים ולחשב את מודל העמסה בהתאם למספר משתמשים בשעת השיא (מדד sessions) ואורך הכי קצר של הששן (session time).
- **סביבת הבדיקה** – הגדרת סביבת הבדיקות על כל מרכיביה. חשוב לבחור סביבת הבדיקות מקבילה או קרובה ככל שניתן לסביבת הייצור.
- **נתוני הבדיקה** – הגדרת סוג וכמות הנתונים הנדרשים לביצוע הבדיקה. לדוגמא, אם מדובר **במנוע חיפוש** – כדאי לתת רשימה רחבה של מילות מפתח שמכסות מקרי קצה שונים מבחינת המנוע (משפטים קצרים, ארוכים וכו'). אם מדובר ברישום של המשתמש, צריך להקצות מספיק שמות משתמשים ייחודיים כך שהבדיקה לא תעצר על שם משתמש שכבר קיים. לא פעם צריך להגדיר החרגות מיוחדות לטובת הבדיקה, כגון מספר כרטיס אשראי לא אמיתי שיתקבל ע"י המערכת וכו'.
- **לוח זמנים** – לוח זמנים מפורט של הבדיקה, כולל זמני פיתוח התסריטים, הרצות וזמן תיקונים משוער.
- **גורמים המעורבים וחלוקת אחריות** – רשימה של הגורמים המעורבים בבדיקה כגון DBA, אנשי מערכות מידע, מנהל הפרויקט וכו'.

בחירת תסריטי בדיקה

בניגוד לתסריטי בדיקה ידניים שמטרתם לחסות מרבית הפונקציונליות, בבדיקות עומסים מטרה העיקרית היא לבדוק תהליכים שהשפעתם ישירה או מקיפה היא קריטית לביצועי המערכת.

יש לבחור את תסריטי הבדיקה לפי כללים הבאים:

המסלול הקריטי – בחירת תהליכים שהשפעתם היא קריטית מבחינה עסקית לארגון או מהותית מבחינת המשתמש

אפקט הדומינו – בחירת תהליכים שאולי השפעתם הישירה היא לא קריטית לארגון או מערכת, אך עצם הפעלתם יוצרת סדרת פעולות שעלולה לגרום לעומס יתר של המערכת. דוגמא טובה לסוג זה של תהליך היא הפקת דוחות.

סוגי הבדיקות

ישנם 5 סוגי העמסה בסיסיים שכדאי להכיר. המטרה ואופן הביצוע של כל סוג הם שונים.

- **בדיקת עומס רגיל (workload)** - הדמיה של עומס יום-יומי של המערכת. בדרך כלל נגזר מפעילות ממוצעת של משתמשים לאורך היום.
- **בדיקת עומס קיצוני (stress)** - הדמיה של עומס קיצוני על המערכת. נגזר מפעילות או מצפי לפעילות המערכת בשעות השיא עם מכפיל ביטחון (כ-50%-30%). בדרך כלל עומס קיצוני הוא כפול או משולש מעומס היום-יומי.
- **בדיקת יציבות (stability)** - הדמיה של פעילות ממושכת של המערכת ברמת עומס יום-יומי. בדיקת יציבות בדרך כלל נמשכת בין 12-48 שעות. מטרת הבדיקה היא לגלות זליגת משאבי השרתים, כגון זיכרון, וגילוי בעיות יציבות נוספות.
- **בדיקת הרחבה (scale)** - מטרת הבדיקה היא לגלות את יכולת ההרחבה של המערכת – כלומר מה תהיה רמת הביצועים של המערכת במקרה של הרחבת החומרה והגדלה של העומס. בדיקה זו מאוד חשובה במערכות שנדרשות להתרחב אוטומטית (auto scale) או שיש צפי לגדילה משמעותית של כמות המשתמשים ותווק הקצר לאחר עליה לאוויר.
- **בדיקת קיבולת (capacity)** – מטרת הבדיקה היא למצוא קיבולת של המערכת, כאשר ביצועי המערכת נשמרים ברמה המקובלת ובהתאם לדרישות. כלומר, אם המערכת אמור לעמוד בזמן תגובה של 5 שניות וצריכת המשאבים בשרתים לא אמורה לעבור רף של 80%, אז בדיקת קיבולת תגלה מה היא כמות משתמשים מקסימלית שבה ניתן לעמוד ביעד זה.

הבדיקות הכי נפוצות הן workload ו-stress אך כמובן כדאי לבצע לפחות את שלושת סוגי הבדיקות הראשונים, כולל בדיקת יציבות. בדיקת קיבולת נגזרת בלא מעט מהמקרים מבדיקת stress מכיוון שלרוב בדיקת stress תביא את המערכת לקצה היכולת. בדיקת הרחבה (scale) תתאים יותר למערכות שיש בהם מנגנון auto-scale, כלומר הוספה אוטומטית של שרתים או משאבים בהתאם לעומס או ניצול המשאבים של המערכת. אך כמובן שכדאי לבצע בדיקת scale גם במקרים בהם צוות התשתיות רוצה להבין מה תהיה ההשפעה של הוספת חומרה נוספת על ביצועי המערכת.

סיכום

כתבה זו היא ראשונה בסדרת הכתבות בנושא בדיקות עומסים. בכתבה זו הבאתי במפניכם את מבוא והצד התאורטי יותר של הבדיקות עומסים. בכתבה הבא נדבר על הצד הפרקטי יותר של הבדיקה – נבחן את הכלים לבדיקות עומסים ונדבר על ביצוע הבדיקה בפועל. לכל שאלה, הערה או המלצה - אתם מוזמנים לכתוב לי ל: ndimer@test4load.com. להתראות בגיליון הבא!