הטכניון - מכון טכנולוגי לישראל **הפקולטה למדעי המחשב**

28 לפברואר 2013

פרופ"ח ארז פטרנק
שחר תמנת

אלגוריתמים לניהול זיכרון - 236780
חורף תשע"ג – מועד א'

הנחיות:

- הבחינה עם חומר סגור.
- אסור להחזיק פלאפונים ברשות הנבחנים בזמן הבחינה.
- כתבו בצורה מסודרת ונקייה ובכתב ברור.
- בבחינה 4 שאלות, משך הבחינה שלוש שעות.
- נמקו את כל תשובותיכם.
- כאשר מבקשים להציע פיתרון אלגוריתמי לאיזושהי בעיה, יש לנסות לתת פיתרון יעיל ככל היותר במקום ובזמן.

בהצלחה !

שאלה 1: (28 נקודות) שאלה זו מתייחסת לאלגוריתם ה- Mostly Concurrent.

א. (4 נק) תארו כל פאזה באלגוריתם: scanning, card cleaning, ו- stop-the-world.

ב. (4 נק) הסבירו את השיפור של האלגוריתם שבו אין סורקים דרך כרטיס מלוכלך: מה החיסכון מהאלגוריתם המקורי, ומדוע נשמרת הנכונות?

ג. (4 נק) הסבירו את השיפור של האלגוריתם שבו מנקים כרטיסים שטרם נסרקו: מה הפעולה הנוספת שמבצעים? מדוע היא מועילה? מדוע נשמרת הנכונות של האיסוף?

ד. (4 נק) תנו דוגמא (מוזרה וקיצונית כאוות נפשכם) של מבנה heap, מצב סריקה, ותצורה של כרטיסים מלוכלכים שעבורם לא יתבצע אף ניקוי בפעולה הנוספת שמבצעים.

ה. (4 נק) תנו דוגמא (מוזרה וקיצונית כאוות נפשכם) של מבנה heap ותצורה של כרטיסים מלוכלכים שעבורם יתבצעו ניקויים רבים בפעולה הנוספת אך בסופו של דבר הפעולה לא תועיל לקיצור זמן העצירה.

ו. (4 נק) תנו דוגמא (מוזרה וקיצונית כאוות נפשכם) של מבנה heap ותצורה של כרטיסים מלוכלכים שעבורם אי סריקה דרך כרטיס מלוכלך תגרום להארכת זמן העצירה באופן משמעותי. הניחו שמנקים כרטיסים רק בעת העצירה.

ז. (4 נק) האם הדוגמא שלכם תגדיל את זמן העצירה באופן משמעותי גם אם נריץ את שלב ניקוי הכרטיסים פעם אחת במקביל לריצת התוכנית לפני העצירה? אם כן – הסבירו. אם לא, עדכנו את הדוגמא כך שגם במקרה זה זמן העצירה יגדל באופן משמעותי. (הניחו שבזמן ניקוי כרטיסים מקבילי מסתכלים על בנים של אובייקטים מסומנים ומסמנים אותם אם צריך, אבל אין סורקים את הבנים אם הם נמצאים על כרטיס מלוכלך.)

שאלה 2: (18 נקודות) שאלה זו מתייחסת ל-Parallel Garbage Collection

א. (3 נק) מהו parallel garbage collection?

ב. (4 נק) תאר מצב של מערכת שבו parallel garbage collection יותר טוב מ- concurrent garbage collection .

ג. (3 נק) בהנחה שה-bitmap מכיל ביט עבור כל 8 מילים ב-heap' למה tracing parallel collector צריך לסנכרן את הכתיבה ל-bitmap? הסבירו מדוע תיווצר בעיה אם כל חוט ירשום ל-bitmap ע"י כתיבה פשוטה ללא שום סינכרון. (הניחו חומרה סטנדרטית שבכל כתיבה משנה ערך של מילה שלמה ולא רק ביט יחיד.)

ד. (4 נק) תאר את מנגנון ה-stealable mark queues. איזו בעיה הוא פותר?

ה. (4 נק) נציע את השינוי הבא לאלגוריתם. במקום להעביר ½ מהעבודה אל ה-stealable mark queue כאשר הוא ריק, יעביר אליו ה- thread 100% מהעבודה הלוקלית ואז ינסה להחזיר רבע ממנה בפעולת סינכרון עם החוטים האחרים. כמו כן כל thread שלוקח עבודה מ-stealable mark queue של מישהו אחר, ינסה לקחת רבע מכמות העבודה שם. איך ישפיע השינוי על האלגוריתם?

שאלה 3: (29 נקודות) שאלה זו מתייחסת ליעילות cache.

ננסה לתכנן אלגוריתם איסוף שהתנהגות ה-cache שלו טובה. לשם כך נחלק את הזיכרון לאיזורים ונשתדל לסרוק כל איזור לחוד, כך שפעילות האיסוף תתרכז כל פעם באיזור אחר.

א. (6 נק) הציעו אלגוריתם מסגנון sweep-mark שמנסה לעבוד כל פעם על חלק אחד של ה-heap בלבד, וכשנגמרת שם העבודה, הוא עובר לרכז את מאמצי האיסוף באיזור אחר. ניתן להתחיל מאיזשהו איזור המוצבע

על-ידי השורשים. הסבירו מה יעשה האלגוריתם שלכם כאשר מתגלה מחוון לאובייקט שאינו נמצא באיזור האיסוף הנוכחי.

ב. (4 נק) איך מזהים את סיום שלב ה-mark?

ג. (2 נק) הסבירו מדוע אתם מצפים לראות יתרון בביצועים.

ד. (3 נק) מהי התקורה במקום ובמורכבות האלגוריתם יחסית לאלגוריתם sweep-mark רגיל?

ה. (4 נק) תארו מבנה heap שעבורו האלגוריתם יעבוד מצוין. נמקו.

ו. (4 נק) תארו מבנה heap שעבורו האלגוריתם יעבוד לא טוב. נמקו.

ז. (6 נק) תארו מיקבול של האלגוריתם שהצעתם כך שכל חוט יעבוד על איזור אחר. תארו מה עושה כל חוט ואילו חלקים בביצוע דורשים סינכרון בין החוטים השונים.

שאלה 4: (27 נקודות) שאלה זו מתייחסת ל-On-the-fly GC של דייקסטרה.

1. (2 נק) הסבר בקצרה מהי משמעות הצבעים שחור/אפור/לבן באלגוריתם.
2. (4 נק) האלגוריתם של דייקסטרה עושה שימוש ב-write barrier. הבא דוגמא לבעיה שתיווצר אם האלגוריתם יורץ ללא ה-write barrier.
3. (4 נק) ב-write barrier באלגוריתם נעשה שימוש בשיטת change & shade. הסבירו מהי שיטה זו. מהי הבעיה הנגרמת אם פועלים במקום זאת בשיטת shade & change ?

חיסרון עיקרי של האלגוריתם של דייקסטרה הוא שהאלגוריתם אינו מתאים לתוכניות מקביליות שיש בהן מספר חוטים (של התוכנית) הרצים במקביל. הוצע לבצע שינוי באלגוריתם של דייקסטרה כך שיתאים למספר חוטי תוכנית. לצורך השינוי, ה-write barrier עדיין יעבוד בשיטת change & shade, שבו יש חשש שחוט הנמצא באמצע write-barrier לא יספיק לדווח על האפרה של צומת, ובאותו זמן החוט האוסף יסיים את הסריקה. כדי לפתור את הבעיה, מוסיפים שלב handshake שיבוצע לפני שה-collector עובר לשלב ה-sweep. לאחר סיום ה-handshake עם כל חוטי התוכנית, ה-collector יוכל להחליט האם לעבור לשלב ה-sweep, או שיש לו עבודה נוספת בשלב ה-mark. אם יש להמשיך לעבוד על הסריקה, ה-collector ימשיך בסריקה ובסופה ינסה שוב handshake לנסות לסיים שוב וחוזר חלילה.

4. (6 נק) הסבר את האלגוריתם החדש. מה יבוצע ב-handshake? כלומר, איזו בדיקה צריך ה-collector לעשות על-מנת להחליט אם לעבור לשלב ה-sweep? אם לא יעבור לשלב ה-sweep, מה עוד עליו לבצע בשלב ה-mark?
5. (3 נק) האם ניתן להראות ש-handshake אחד תמיד יספיק לסיום הסריקה? או שעלול באמת להיווצר צורך בחזרה לשלב הסריקה וביצוע של יותר מ-handshake אחד? נמקו.
6. (4 נק) האם באלגוריתם החדש עלולה להיווצר בעיה של termination? כלומר, שהאלגוריתם לא יסיים לעולם? נמקו.
7. (4 נק) האם באלגוריתם החדש ניתן לוותר על ה-write barrier ולהשתמש רק ב-handshake שהצעתם בסעיף ד? נמקו.