234322

פרק 2

RAID

RAID

Redundant Array of Inexpensive Disks

מערכות מרובות דיסקים

Chen, Lee, Gibson, Katz, Patterson,

***RAID: High-performance reliable secondary storage,* ACM Computing Surveys 26(2), 145-186, June 1994.**

מערכות מרובות דיסקים

- שיפור בביצועים של מערכות מחשב:
  - מעבדים 50% – לשנה,
  - זמן תנועת זרוע 10% – לשנה,
  - קצב העברה 10% – לשנה.
- אפליקציות חדשות (וידאו, hypertext, מולטימדיה) דורשות שטח אחסון רב, מהיר וזול.
- אפליקציות ישנות נעשות יותר שאפתניות ודורשות יותר מהדיסקים

הפתרון: שימוש במספר רב של דיסקים זולים.

- אפליקציות שונות עובדות במקביל
- באותה אפליקציה, ההעברה נעשית במקביל

אמינות

אם ההסתברות שדיסק ייפול במשך שעה היא p

MTTF = Mean Time To Failure = 1/p

לדיסק בודד

MTTF = 20,000 hours = 2.3 years

עבור 100 דיסקים,

P(לפחות דיסק אחד נופל) = 1 – (1 – p) ^{100} ≅ 100 p

$$MTTF\ (100) = \frac{MTTF\ (1)}{100} = 200\ H \cong 8\frac{1}{3}\ days$$

- ההנחה שנפילות של דיסקים שונים הן בלתי-תלויות אינה מציאותית.
  - הפרעות ברשת החשמל משפיעות על כל הדיסקים
  - הסיכוי ליפול תלוי בזמן (מכסימלי בהתחלת פעולת המערכת, וכשהמערכת מזדקנת)

יתירות (redundancy)

אינפורמציה נוספת בדיסקים, כך שאם דיסק אחד נופל אפשר לשחזרו.

איפה לשים את האינפורמציה הנוספת?

- **ראי mirroring**: תוכן כל דיסק משוכפל.
  - יקר אך זמין.
- **קודים לתיקון שגיאות:**
  - לדוגמא, פיזור בייטים: כל בייט נכתב ב 9 דיסקים,
    - כל ביט, כולל ביט הזוגיות, נכתב על דיסק אחר.
  - לא עמיד בפני יותר מנפילה אחת (מתוך תשעה דיסקים).
  - יותר זול, אך לפעמים מאיט את הקריאות.

הסתברות נפילת מערך דיסקים

יהי p_i ההסתברות **שבדיוק** i דיסקים מתוך G יתקלקלו.

$$p_0 = (1-p)^G = 1 - Gp + \binom{G}{2} p^2 + O(p^3)$$

$$p_1 = \binom{G}{1} p(1-p)^{G-1} = Gp\left[1-(G-1)p+O(p^2)\right] =$$

$$= Gp - G(G-1)p^2 + O(p^3)$$

$$p_2 = \binom{G}{2} p^2(1-p)^{G-2} = \frac{G(G-1)}{2} p^2\left[1+O(p)\right] =$$

$$= \frac{G(G-1)}{2} p^2 + O(p^3)$$

הסתברות נפילת מערך דיסקים

יהי $p_{\geq i}$ ההסתברות **שלפחות** i דיסקים מתוך G יתקלקלו.

$$p_{\geq 2} = 1 - p_0 - p_1 =$$

$$= 1 - \left(1 - Gp + \frac{G(G-1)}{2} p^2\right) - \left(Gp - G(G-1)p^2\right) + O(p^3)$$

$$= \frac{G(G-1)}{2} p^2 + O(p^3) \cong p_2$$

ההסתברות שלפחות שני דיסקים מתוך תשעה יפולו:

$$p_{\geq 2}(9) = \binom{9}{2} p^2 + O(p^3) \cong 36\, p^2$$

הסתברות נפילת המערכת

- נניח שבמערכת N דיסקים המחולקים לקבוצות של G

$$P_{Global} \cong \frac{N}{G} p_{\geq 2:G} \cong \frac{N}{G} \frac{G(G-1)}{2} p^2 = \frac{N(G-1)}{2} p^2$$

- מערך של 1000 דיסקי נתונים ו-8/1000=125 דיסקי זוגיות (1125 דיסקים סה"כ) יפול, אם שני דיסקים מתוך אחת מ-125 הקבוצות יפלו:

$$P_{Global} \cong 125 \cdot p_{\geq 2:9} \cong 125 \times 36\, p^2 = 4{,}500\, p^2$$

$$p = \frac{1}{20{,}000} \Rightarrow P_{Global} = \frac{4{,}500}{20{,}000^2} = \frac{1}{88{,}888}$$

כלומר,

MTTF ≈ 3,700 days ≈ 124 mos = 10 years 4 mos

איפיונים של שיטות RAID

רמת הגרעיניות: Striping Unit

- אם נפזר כל בייט בין מספר דיסקים
  - בכל קריאה נצטרך לקרוא 8 דיסקים
  - בכל כתיבה נצטרך לקרוא 9 דיסקים.
- אם נשמור את כל המסילה על אותו דיסק ודיסק אחר יחזיק את ביטי הזוגיות של 8 דיסקים:

$$track_9 = track_1 \oplus track_2 \oplus \ldots \oplus track_8$$

  - קריאה בגישה לדיסק אחד,
  - כתיבה בגישה לשני דיסקים.

האם כדאי לשמור את כל המסילות של הקובץ באותו דיסק, או לפזר אותם בין דיסקים שונים?

איפיונים נוספים לשיטות RAID

<u>השיטה לפיזור היתירות</u>

- אם אחד הדיסקים יכיל את הזוגיות של 8 דיסקים אחרים, כל כתיבה תחייב גישה לדיסק זה (hotspot)
- אפשר לפזר את הזוגיות בין דיסקים שונים.

<u>איזו אינפורמציה יתירה להחזיק:</u>

- שכפול כל דיסק
- זוגיות
- קוד אחר Hamming, Reed-Solomon

מדריך: רמות RAID

- רמה 0: ללא-יתירות (nonredundant)
- רמה 1: שכפול (mirroring)
- רמה 2: שימוש בקודים מקובלים לתיקון שגיאות בזכרון
- רמה 3: זוגיות ברמת הביט, bit parity
- רמה 4: זוגיות ברמת הבלוק, bit interleaved parity
- רמה 5: זוגיות מפוזרת,
  block interleaved distributed-parity
- רמה 6: P + Q Redundancy

רמה 0: ללא יתירות (nonredundant)

- הנתונים נכתבים לדיסקים ללא עיבוד נוסף.
  כתיבה מהירה ביותר,
  אך נראה שיטות בהן הקריאה מהירה יותר.
- ניצול זכרון אופטמלי.
- לא עמיד בפני נפילות.

רמה 1: שכפול (mirroring)

- לכל דיסק יש העתק מדוייק: אותו נתון מופיע בכתובת A הן בדיסק א והן בדיסק ב.
  - קריאה מהירה, כי נשתמש בזרוע הקרובה יותר לכתובת היעד.
  - הכתיבה איטית, כי נצטרך להזיז זרועות שני הדיסקים, וצריך לחכות עד שהכתיבה השניה תסתיים.
- התאוששות מהירה מתקלה בודדת
- ניצול זכרון נמוך.

חישוב השהיית הסיבוב בכתיבה

- גם אם שתי הזרועות נעות באופן מסונכרן, הדיסקים עצמם אינם מסונכרנים (כל אחד מהם נמצא בגזרה אחרת).
- הכתיבה תיגמר כאשר אחרון הדיסקים יגמור את הכתיבה.
- כיון שהשהיית הסיבוב של כל דיסק בנפרד הוא משתנה מקרי המפולג אחיד, יש לחשב את התוחלת של מכסימום של שני משתנים אקראיים מפולגים אחיד

$i = 1,2 \qquad X_i \sim U[0, 1]$

השהיית הסיבוב היא $X = \max\{X_1, X_2\}$,

ועלינו לחשב את $E(X)$

- נכליל ונחשב את התוחלת של $X = \max\{X_1, X_2, \ldots, X_n\}$

שתי דרכים לחישוב תוחלת

עבור משתנים דיסקרטיים (בדידים):

$$E(X) = \sum_{i>0} ip_i = \sum_{i>0} \sum_{j \geq i} p_j$$

עבור משתנים רציפים:

$$E(X) = \int_0^1 t \frac{d}{dt} P(X \leq t) dt = \int_0^1 P(X \geq t) dt$$

חישוב השהיית סיבוב (המשך)

$$X = \max\{X_1, \ldots X_n\}$$

$$P(X \geq t) = 1 - P(X \leq t) = 1 - P(X_1 \leq t) \cdot \ldots \cdot P(X_n \leq t) = 1 - t^n$$

$$E(X) = \int_0^1 (1 - t^n) dt = \int_0^1 dt - \int_0^1 t^n dt = t \Big|_0^1 - \frac{t^{n+1}}{n+1}\Big|_0^1 = 1 - \frac{1}{n+1} = \frac{n}{n+1}$$

מסקנה: השהיית הסיבוב ב-MIRROR היא 2/3 סיבוב.

רמה 2: שימוש בקודים לתיקון שגיאות

- שימוש בקודים ידועים לתיקון שגיאות בזכרון (memory style error correcting codes)

- במקום זוגיות, משתמשים בקודים יעילים יותר לתיקון שגיאות.
  - חוסך מקום.

רמה 3: זוגיות ברמת הביט (bit interleaved parity)

- דיסקים מחולקים לקבוצות של G דיסקים (קבוצת תיקון שגיאות), ולכל קבוצה מתוסף דיסק זוגיות.
  - יחידת האינפורמציה שכתובה על אותו הדיסק היא ביט.
- לכן, בקריאה ניגש לכל $G+1$ הדיסקים, אך הקריאה תיגמר כאשר ייקראו G הראשונים. ע"כ השהיית הסיבוב היא

$$\frac{G}{G+2}$$ סיבוב.

- כל כתיבה צריכה לעדכן גם את דיסק הזוגיות, ולכן השהיית הסיבוב תהיה

$$\frac{G+1}{G+2}$$ סיבוב.

- קצב העברה גבוה, אך זמן גישה ארוך יחסית.
- אלגוריתם פשוט.

רמה 3: מה צריך לקרוא?

- יחידת האינפורמציה הקטנה ביותר שניתן לקרוא מדיסק בודד היא גזרה.
- אבל ב-RAID 3 כדי לקרוא גזרה, חייבים לקרוא G דיסקים.
  - לכן, ביחידה הקטנה ביותר יש G גזרות.
- כדי לכתוב גזרה אחת, חייבים לעדכן G גזרות, כלומר, לקרוא G ולכתוב את הגירסה המעודכנת.
- כדי לכתוב G גזרות, אין צורך לקרוא את האינפורמציה הישנה.

רמה 4: זוגיות ברמת הבלוק (block interleaved parity)

- יחידת האינפורמציה שכתובה על דיסק היא בלוק שנקרא striping unit (יכולה להיות גזרה או מסילה).
  - קריאה ליחידה שמוכלת בבלוק, תיגש רק לדיסק אחד
  - כתיבה תיגש גם לדיסק הזוגיות
- איך נחשב מה לכתוב לדיסק הזוגיות?

READ old_data
READ old_parity
new_parity = old_parity ⊕ old_data ⊕ new_data
WRITE new_data
WRITE new_parity

- כל כתיבה דורשת 4 פעולות I/O
  - הכתיבה איטית ודיסק הזוגיות הוא hotspot
- הקריאות יכולות להתבצע במקביל.

רמה 5: זוגיות מפוזרת (block interleaved distributed-parity)

- דומה לרמה 4, אך דיסק הזוגיות מפוזר בין הדיסקים

- אם נעבור על הקובץ סדרתית, נעבור על כל הדיסקים פעם אחת לפני שנחזור לאותו דיסק פעם נוספת.

הערה: רמה 5 תמיד עדיפה על רמה 4.

d0	d1	d2	d3	d4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

רמה 6: P + Q Redundancy

- משתמש בקוד Reed-Solomon לתיקון שגיאות.
  - עמיד בפני שתי תקלות בו זמניות.
  - תיקון t שגיאות ע"י הוספת 2t ביטים.
  - ראה הקורס: "מבוא לתורת הצפינה".

- כל כתיבה מחייבת 6 פעולות I/O
  - קריאת הנתונים,
  - קריאת זוגיות מדיסק ראשון
  - קריאת זוגיות מדיסק שני.
  - כתיבת הנתונים,
  - כתיבת זוגיות לדיסק זוגיות ראשון,
  - כתיבת זוגיות לדיסק זוגיות שני.

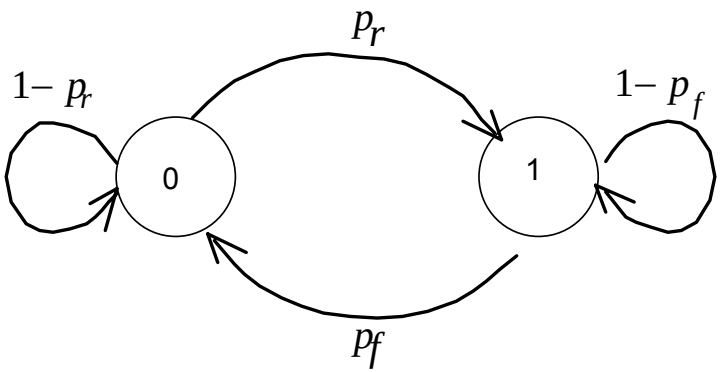
זמן השחזור (הצפוי) לאחר תקלות

נניח זמן דיסקרטי

תהי p_f ההסתברות שדיסק יפול ביחידת זמן,

ו- p_r ההסתברות שמערכת שנפלה תתוקן ביחידת הזמן.

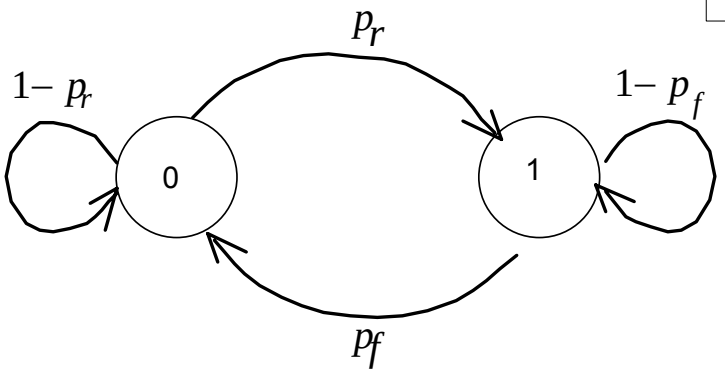
למערכת שני מצבים: תקין = 1, ומקולקל = 0.



זמן השחזור (הצפוי) לאחר תקלות (המשך)

$$\begin{cases} p_1 = (1-p_f)p_1 + p_r p_0 \\ p_0 + p_1 = 1 \end{cases} \Rightarrow p_0 = \frac{p_f}{p_f + p_r} \;;\; p_1 = \frac{p_r}{p_f + p_r}$$

כיון ש- p_f קטן ו-p_r גדול
אזי $p_0 \cong p_f / p_r$



סוגי תקלות

<u>תקלת מערכת</u>: כל תקלה שאינה נובעת ממערכת הדיסקים:

הפסקת חשמל, טעות תכנה,...

<u>תקלת דיסק</u>: תקלה שמקורה במערכת הדיסקים:

סקטור פגום, תקלת בקר, מחיקת דיסק,...

שני מדדי אמינות

MTTF (Mean time to failure)

MTTDL (Mean time to data loss) זמן צפוי לאבדן נתונים.

MTTF << MTTDL

הסתברות לאבדן נתונים

<u>התרחיש העיקרי:</u>

אם יש נפילת מערכת + נפילת דיסק, עלולים לאבד נתונים (בהנחה שהמערכת שומרת יומן של הפעולות, כך שניתן לשחזר את הנתונים אם רק מערך הדיסקים נופל).

<u>ההסתברות לכך:</u>

$$P_{\text{Data Loss}} = P_{\text{System Failure}} \cdot P_{\text{Global}}$$
$$= \tfrac{1}{2} P_{\text{System Failure}} \cdot N(G-1)p^2$$

$$\text{MTTDL (disk + system)} = 1/\ P_{\text{Data Loss}}$$

שחזור אוטומטי

במערכת יש מספר דיסקים עודפים, כדי להחליף דיסקים שהתקלקלו.
ההחלפה נעשית באופן אוטומטי, ללא התערבות המפעיל.
תקופתית, טכנאי השרות מחליף את הדיסקים שהתקלקלו
מבלי להשבית את המערכת.

כל עוד אף דיסק לא התקלקל,
אפשר לנצל אותם כדי להגדיל את היתירות,
ולמנוע אובדן נתונים.

זכרון מטמון cache

- מערכות RAID מצויידות בזכרון אלקטרוני גדול (עד 4GB)
  (מהיר פי ≈ 10,000 מהדיסק) שמגובה בבטריות.
  - עמיד לתקלות מערכת וחומרה.
  - כתיבה בזכרון זה היא אטומית.

- זכרון המטמון מהווה חוצץ לכתיבה ב- RAID :
  - כדי לכתוב ערך לדיסק, נכתוב אותו לזכרון המטמון.
  - כאשר זכרון המטמון מתמלא, נוריד גזרות לדיסקים.

- בקריאה: נבדוק אם הגזרה הדרושה נמצאת בזכרון המטמון,
  אחרת נטען אותה.

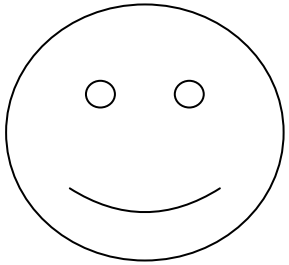
יתרונות

<u>שיפור זמן הכתיבה</u>:

מבחינת המשתמש הכתיבה מסתיימת ברגע שנכתב בזכרון המטמון.

<u>ניצול לוקליות</u>:

אם נקרא ערך שכתבנו/קראנו לא מזמן,
הערך המבוקש יהיה בזכרון המטמון.



<u>אטומיות</u>:

הופך את הכתיבה בדיסק לאטומית
(בעזרת האלגוריתמים שנלמד בפרק על שחזור).

חסכונות נוספים של cache

הקטנת מספר הכתיבות:

נחסוך כתיבות אם נכתוב שוב לגזרות שטרם העברנו לדיסק.

תזמון כתיבה:

אפשר להשתמש באלגוריתמים מתוחכמים לכתיבה:

Piggy-backing:

אם ניגשים לגליל לצורך קריאה,
אפשר באותה ההזדמנות להעביר גזרות שמיועדות לגליל זה מזכרון המטמון.

באופן זה חוסכים תנועות זרוע.

סיכום

ביצועי מערכות RAID טובים באופן ניכר ממערכות קונבנציונליות,
והן המערכות המקובלות של דיסקים על main frames .