

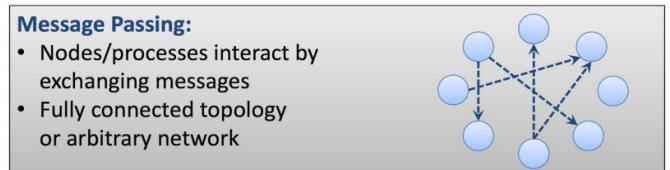
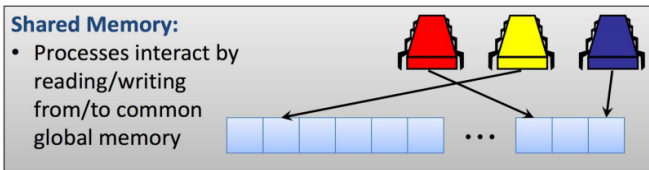
מערכות מבוזרות - Distributed System

אוסף של יחידות חישוב אשר יכולים לתקשר אחד עם השני
הערה: גם מחשב אחד הוא מבוזר, היות ויש כמה מעבדים וכמה תהליכים.

יתרונות של מערכות מבוזרות:

1. לשותף מידע בין קדקודים שונים
2. לטפל בכמות מידע גדולה יותר (במקום שכל המידע יהיה במקום כלשהו, אז ניתן להחזיק מידע בכל קודקוד)
3. עיבוד מקבילי (Parallelism)
4. Redundancy and Resiliency (אם מתוך 100 מכשירים אז 30 נפלו אז נבנה אותם ככה שהם יעבדו רק בעומס)
5. Scaling

מודלים למערכות מבוזרות



מערכות סינכרוניות: לכל המכשירים ישנו "קו שעון" משותף ולכן אפשר לדבר על סיבובים, שליחת הודעות מתרחשת בתחילת הסיבוב ולקראת סוף הסיבוב ההודעה מגיעה. זמן הסיבוב ידוע מראש

במודל "Message Passing"

בראונד r : בזמן $r - 1$, כל תהליך שולח הודעות. ההודעות מגיעות בזמן r

במודל "Shared Memory"

בכל ראונד, כל תהליך יכול לגשת לתא זיכרון בודד.

מערכות אסינכרוניות: אין "קו שעון", זמן הגעת ההודעה הוא מספר סופי כלשהו לא ידוע אשר יכול להשתנות בין הודעות.

במודל "Message Passing"

ההודעות תמיד מגיעות

זמן הגעת ההודעה הוא לא ידוע מראש ותמיד נסתכל על המקרה הגרוע ביותר

במודל "Shared Memory" (יש זיכרון אחד)

כל התהליכים בסופו של דבר יבצעו את הצעד הבא

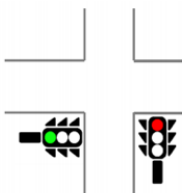
זמן העיבוד לא ידוע מראש ותמיד נסתכל על המקרה הגרוע ביותר

תקלות Failures:

- Crash Failure**
 - קודקוד אשר מפסיק לעבוד בזמן מסוים, במערכת סינכרונית זה יכול לקרות באמצע השעון.
- Byzantine Failure**
 - קודקודיים שרוצים לפגוע בחישוב, "האקרים".
- Omission Failure**
 - קודקוד אשר מפסיק לעבוד לזמן מסוים
- Resilience**
 - התמודדות עם מספר מסויים של תקלות (מודל שמתמודד לדוגמא עם 50 תקלות כן 51 לא)

תכונות נכונות של מערכת מבוזרת:

- Safety** - שום דבר רע לא קורה ויקרה
 - דוגמא: בכל נקודת זמן, לכל היותר רמזור אחד בצבע ירוק
 - על מנת להוכיח Safety נוכיח בעזרת Invariant, המצב ההתחלתי היה בטוח, וכל מצב משאיר את המערכת בטוחה
- Liveness** - משהו טוב יקרה למשהו
 - דוגמא: יש תמיד אור ירוק אחד דלוק ברמזור
 - דוגמא: במערכת אשר מספקת אוכל לאנשים, אז משהו מקבל אוכל.
 - דוגמא: במערכת החלפת הקשר, אז משהו יכול לגשת למקור מידע.
- Fairness** - משהו טוב יקרה לכולם
 - דוגמא: במערכת אשר מספקת אוכל לאנשים, אז כולם מקבלים מספיק אוכל.
 - דוגמא: במערכת החלפת הקשר, אז כל אחד יכול לגשת למקור מידע כל הזמן.
 - דוגמא: שני האורות ירוקים ברמזור



מערכת בסיסית מכילה:

1. n קודקודים אשר מיוצגים ב- v_1, \dots, v_n
באופן לא מפורש - Implicit יש n קודקודים ממוספרים מ-1 עד n כאשר n הוא מספר ידוע.
2. בכל זמן, לכל קודקוד v_i יש מצב פנימי Q_i

Schedule: רצף של שליחות וקבלות המהוות את האירועים במערכת.
Admissible Schedule: רצף הגיוני, כלומר אין קבלת הודעה לפני שהיא נשלחת.

קונפיגורציה C זה וקטור של n כניסות אשר מתאר את מצב המערכת.
כאשר לאחר סדרת אירועים נקבל את הריצה הבאה:
 $C_0, \Phi_1, C_1, \Phi_2, \dots$
כאשר Φ_i הוא אירוע כלשהו.

Local View – מה כל קודקוד רואה מה-Schedule מסויים.

בהינתן מתזמן S נגדיר $S|i$ להיות נקודת המבט של קודקוד v_i לאורך המערכת.
לדוגמא, בהינתן רצף $S = s_{13}, s_{23}, s_{31}, r_{13}, s_{32}, r_{31}, r_{23}, s_{13}, s_{21}, r_{31}, r_{12}, r_{32}$
 $S|1 = s_{13}, r_{13}, s_{13}, r_{12}$
 $S|2 = s_{23}, r_{23}, s_{21}$
 $S|3 = s_{31}, s_{32}, r_{31}, r_{31}, r_{32}$
אבחנה: $S|i$ רואה את כל מה ש- i בהתחלה

תרגיל לדוגמא

יהיו p_1, \dots, p_n קודקודים במערכת שליחת הודעות המריצים את האלגוריתם הבא:
אם $i = 1$:

שלח p_1 ל- p_2
המתן לקבל הודעה ממנו
אם קיבלת $5 \leq$ החזר 1
אחרת, החזר 0.

אחרת:

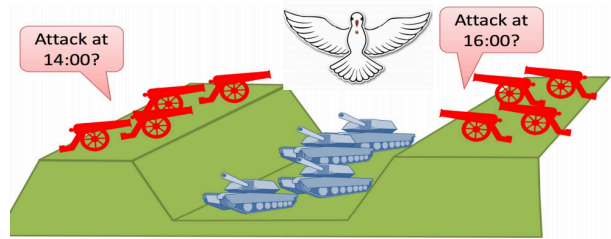
המתן לקבל הודעה מ- p_{i-1}
אם $i = n$ דלג על השלבים הבאים (תהליך אחרון)
שלח הודעה ל- p_{i+1}
המתן לקבל הודעה מ- p_{i+1}
שלח אותה ל- p_{i-1} בתוספת 1

אם המתנת ל- p_{i+1} יותר מ- $2n \cdot T$ יחידות זמן, החזר 0 ושלח ל- p_{i-1} את p_{i-1} , אחרת החזר p_{i-1} .
עבור השאלה הזו: (מערכת סינכרונית)
Safety – קודקוד p_1 מחזיר p_1
Liveness – משהו שאינו p_1 מחזיר p_1
Fairness – כולם מחזירים p_1
זמן הסיבוב – T

מודל הרצה לאלגוריתם:

1. מערכת סינכרונית, הודעה אחת הולכת לאיבוד.
Safety – אם הודעה אבדה בדרך מהקודקוד הראשון לקודקוד החמישי אז הקודקוד הראשון יקבל $5 >$ ולכן ואין **Safety**
Liveness – אם ההודעה מהקודקוד הראשון לשני אבדה אז כולם תקועים
הערה: אם **Liveness** לא מתקיים אז **Fairness** לא מתקיים

2. מערכת אסינכרונית – כל ההודעות מגיעות.
Safety – לא, כי אם הטיימר של p_2 פקע, אז הוא ישלח לקודקוד הראשון p_1 ו- $1 < 5$
Liveness – גם אם הטיימרים של כולם יפקעו אז p_n לא יחכה ולכן יחזיר 1.
Fairness – כמו **Safety**



מודל: שני יחידות חישוב דטרמיניסטיות אשר מדברות במערכת סינכרונית מעל מערכת שליחת הודעות לא אמינה (הודעות יכולות ללכת לאיבוד)

קלט: כל יחידת חישוב מתחילה עם קלט מהקבוצה $\{0,1\}$

פלט: כל אחד מיחידות החישוב צריך להוציא פלט $\{0,1\}$

הסכמה: שני יחידות החישוב צריכות להסכים על אות פלט

נכונות: אם הקלט זהה אז הפלט צריך להיות זהה לקלט במידה והודעות לא הולכות לאיבוד, אחרת, להחזיר 0 או 1 מוסכמים.

תקלות: הודעה יכולה ללכת לאיבוד

סיום: שני יחידות החישוב צריכים להסכים בזמן סופי

אבחנה: ללא נכונות, אפשר לבצע אלגוריתם פשוט אשר הינו 'החזר 0'



הבעיה אינה פתירה

שיטת הוכחה: להניח שקיים אלגוריתם ולהראות סדרת הרצה הזחות זו לזו שתביא לסתירה

סימונים:

הרצה E לא ניתנת להבדל מהרצה E' עבור קודקוד v אם v רואה את אותו הקלט ואת אותם ההודעות ב- E ו- E'

אם E לא ניתנת להבדל עבור קודקוד v , אזי v מבצע את אותה סדרה של פעולות (בגלל שזה דטרמיניסטי) בשני

ההרצאות, ולכן נסמן $E|v = E'|v$ או לחלופין $E \sim_v E'$

אבחנה: אם נשקול סדרה של הרצאות $E_0, E_1, E_2, \dots, E_k$ כך ש- $E_{i-1} \sim_v E_i$ עבור $i \in \{1, \dots, k\}$ עבור קודקוד v כלשהו אזי מתקיים כי כל הקודקודיים מוצאים אותו פלט בכל ההרצאות (בגלל שזה דטרמיניסטי)

הוכחה:

נניח בשלילה כי קיים אלגוריתם סופי דטרמיניסטי הפותר את הבעיה ב- T סיבובים

נשקול את רצף ההרצות $Executions$ הבא:

Nodes always decide after exactly T rounds

Execution E_0 : Both inputs are 0, no messages are lost.

Execution E_1 : One of the messages in round T is lost.

...

Execution E_{2i} : Both messages in round $T+1-i$ are lost.

Execution E_{2i+1} : One of the messages in round $T-i$ is lost.

...

Execution E_{2T} : Both inputs are 0, no messages are delivered. All outputs are 0 due to similarity.

Execution E_{2T} : Both inputs are 0, no messages are delivered. All outputs are 0 due to similarity.

Execution E_{2T+1} : Input of v_1 is 0 but input of v_2 is 1. No messages are delivered.

Execution E_{2T+2} : Both inputs are 1, no messages are delivered.

...

Execution $E_{2T+2i+1}$: Exactly one of the messages in round i is delivered.

Execution $E_{2T+2i+2}$: Both messages in round i are delivered.

...

Execution E_{4T+2} : Both messages in round T are delivered. Decision must be 1 - a **contradiction**.

- We start with an execution in which both nodes have input 0 and no messages are lost: by validity both nodes must decide 0.
- We remove messages one by one to obtain a sequence of executions such that consecutive executions are similar.
- From an execution with no messages delivered and both inputs 0, we can get to an execution with no messages delivered and both inputs 1 (in two steps).
- By adding back messages one by one, we obtain an execution in which both nodes have input 1 and no messages are lost: by validity both nodes must decide 1 \Rightarrow contradiction!
- Not hard to generalize to an arbitrary number $n > 1$ of nodes.

הרצאה 3 – בעיית 2 הגנרלים רנדומלי

תזכורת: בעיית 2 הגנרלים יכולה להיות פתירה אם:

- נאפשר לאחד מהצדדים להטיל מטבע
- נקבל גם פתרון עם הסתברות $1 - \epsilon$ להסכמה

אלגוריתם "הרמות" - Level Algorithm:

1. שני הרמות מאותחלות ל-0
2. בכל ראונד: שני הקודקודים שולחים את הרמה שלהם לשני
3. כאשר קודקוד u עם רמה l_u מקבל הודעה מקודקוד v אשר עם רמה l_v אזי u מעדכן את הרמה שלו ל- $l_u = \max\{l_u, l_v + 1\}$

אבחנה: הרמות בחיים לא יורדות

תכונות

טענה: בכל הזמנים, כמות הרמות שונה לכל היותר ב-1

הוכחה: באינדוקציה על מספר הסבבים

בהתחלה $l_u = l_v = 0$ ולכן הבסיס מתקיים.

נניח כי הטענה נכונה עבור סיבוב t ונשקול את הסיבוב $t + 1$

אם $l_u = l_v$ אזי הטענה מתקיימת כי הרמה של כל קודקוד עולה לכל היותר ב-1 ובחיים לא יורדת.

ולכן נניח בה"כ כי $l_u = l_v + 1$, לא משנה מה קורה בסיבוב $t + 1$ אזי l_u לא משתנה, אם v מקבל הודעה l_u מקודקוד u אזי הרמה עולה ל- $l_v = l_u + 1$

טענה: אם כל ההודעות מתקבלות אזי הרמה של שני הקודקודים זהה ושווה למספר הסיבובים

הוכחה: באינדוקציה על מספר הסיבובים

עבור שלב הבסיס זה מתקיים באופן טריוויאלי

נניח כי הטענה נכונה $l_u = l_v = t$ לאחר סיבוב t ונשקול את הסיבוב $t + 1$

יהי l'_u ו- l'_v להיות הרמה של הקודקודים u, v לאחר הסיבוב $t + 1$, לפי ההגדרה של האלגוריתם אזי $l'_u = l_u + 1 = t + 1$
 $l'_v = l_v + 1 = t + 1$

כנדרש

טענה: הרמה l_u של קודקוד u שווה ל-0 אם u לא מקבל אף הודעה.

הוכחה: אם u לא מקבל אף הודעה אזי הרמה לא משתנה ולכן זה נשאר 0, ולכן נניח כי בנקודה מסויימת הוא מקבל הודעה עם הרמה l_v מקודקוד v .

בבגלל שהרמה בחיים לא קטנה ומתחילה ב-0 אז היא לפחות 0, כלומר $l_v \geq 0$ ולכן מהגדרת האלגוריתם הוא מעדכן את

הרמה $l_u = \max\{l_u, l_v + 1\} > 0$ ובגלל שהרמה בחיים לא יורדת $l_u > 0$ לכל אורך האלגוריתם

לסיכום:

1. בסוף האלגוריתם הפרש רמות לכל היותר 1
2. אם כל ההודעות מגיעות, אזי כל הרמות שוות למספר הסיבובים r
3. הרמה של קודקוד u היא לפחות 1 אם u מקבל לפחות הודעה אחת

אלגוריתם רנדומי לפתרון בעיית הגנרלים:

- נניח כי u יכול להשתמש ברנדומיזציה
- נניח שהקלטים יכולים להיות רק 0 או 1

1. קודקוד u בוחר מספר $t \in \{1, \dots, r\}$ באופן אחיד
2. שני הקודקודים מריצים את האלגוריתם "הרמה" (Level Algorithm) עבור r צעדים בזמן ריצת האלגוריתם, בכל הודעה שני הצדדים מוסיפים גם כן את הקלט שלהם וקודקוד u מציין גם כן את t
3. כל קודקוד פולט 1 אם:
 1. הקודקוד יודע מה הערך של t ויש להם קלט זהה
 2. שני הקלטים זהים ל-1
 3. הרמה של הקודקוד הוא לפחות t
 4. הקודקוד פולט 0

טענה: אם אחד הקלטים הוא 0 אז שני הקודקודים פולטים 0
הוכחה: נובע ישירות מהאלגוריתם

טענה: נניח שאחד הקלטים הוא 1

1. אם אף הודעה לא הולכת לאיבוד אזי שני הקלטים יפלטו 1
 2. שני הקודקודים פולטים את אותו הערך אלא אם כן $\{l_u, l_v\} = \{t-1, t\}$
- הוכחה:** נניח תחילה כי $\{l_u, l_v\} = \{t-1, t\}$ בהתאמה, לפי ההגדרה של האלגוריתם, u מחליט 0. בגלל ש- $t > 0$ אזי v מכיר את t ואת 2 הקלטים אשר הם 1, ולכן הוא מחליט 1. כעת, נניח כי $\{l_u, l_v\} \neq \{t-1, t\}$. אם $l_u, l_v < t$ אזי שניהם פולטים 0, אחרת $l_u, l_v \geq t$, מכיוון ש- $t > 0$ אזי שני מכירים את t וראו את הקלטים, ולכן שניהם יפלטו 1. קבלנו כי אם אף הודעה לא הולכת לאיבוד אזי $l_u = l_v = r \geq t$ ולכן יפלטו 1.

טענה: האלגוריתם מגיע להסכמה עם הסתברות של לפחות $1 - \frac{1}{r}$

הוכחה:

תזכורת: ההפרש רמות הוא לכל היותר 1

בגלל שהרמה נקבעת רק לפי כמות ההודעות שמגיעות/הולכות לאיבוד, אזי ה"אויב" יכול לקבוע שהם יהיו שונות, ולכן

$$\{l_u, l_v\} = \{i-1, i\} \text{ עבור } i \in \{1, \dots, r\}$$

בגלל שה"אויב" לא מכיר את t אזי ההסתברות $\{l_u, l_v\} = \{t-1, t\}$ הינה $\frac{1}{r}$ בגלל התפלגות אחידה

לפי שני הטענות הקודמות, הקודקודים מגיעים להסכמה בכל שלב אחר ולכן נוכל להסיק כי ההסתברות לכשלון הינה $\frac{1}{r}$

סם תחתון:

להשלים