רשימת נושאים אפשריים לפרוייקט הסיום של הקורס - תכנות פונקציונלי במערכות מקבילות ומבוזרות

דרישות כלליות לביצוע הפרויקטור

1. עבודה בזוגות
2. שימוש במספר מחשבים (לפחות 4)
3. שימוש במאות/ אלפי תהליכונים ויותר
   - אם היישום עתיר חישובים, מספר התהליכים יהיה בהתאם למספר הליבות בכל מחשב ומספר המחשבים שבשימוש, תוך התחשבות בשיקולים שהוצגו בהרצאות
4. חובה לעבוד ב-OTP עם מודלים **gen_server** ו-**gen_statem** אלא אם התקבל אישור לכך מהמרצה עקב אופי ייחודי של הפרויקטון.
5. חובה לשלב עבודה עם ETS ו/או DETS. שימוש ב-mnesia במקרים שזה מתאים - ייתרון.
6. כל שילוב של מודול נוסף מאלו המוצעים ב-framework של Erlang ומצריך לימוד עצמי, מאחר ולא נדון בקורס מעלה את הדירוג של הפרויקט. דוגמא למודולים כאלה - MNesia, QLC, עצים, מערכים דינמיים וכו׳. הכל בהתאם לפרויקט.
7. פיתוח והעברת האפליקציה תוך שימוש ב-rebar3 (לימוד עצמי)
   7.1. בתוספת מסמך הוראות התקנה והפעלה
8. מסמך תכנון, מצגת סופית והגנה על הקוד
   8.1. ייתכן וההרצאה + ההגנה על המימוש תצולם בוידאו ע״י סגל הקורס

דרישות כלליות - 2

9. יש לצרף סרטון קצר שידגים את פעולת הקורס. הסרטון יועלה ליוטיוב

10. יש להראות בסרטון באנגלית ובצורה בולטת את

- שם הפרויקט
- שמות הסטודנטים
- שם הקורס - Functional Programming in Concurrent and distributed Systems
- שם המרצה - Dr. Yehuda Ben-Shimol
- שם המתרגל - Mr. Guy Peretz

שלבים הכרחיים לביצוע הפרויקטון

1. הכנת תכנון ראשוני הכולל

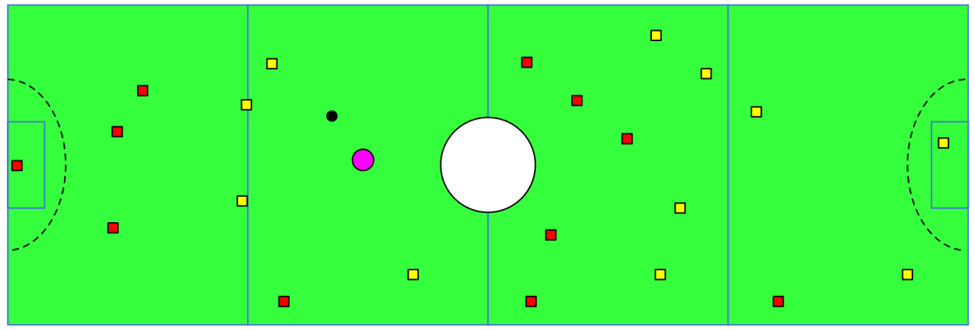
- שם ונושא הפרויקטון
- ארכיטקטורה תוכנתית - אילו ישויות יהיו בכל מחשב, מה האחריות של כל ישות כזו, איזה מידע עובר מישות לישות, תכיפות הערבת המידע וכו׳
- התייחסות לדרישה של שימוש במספר גדול של processes במהלך הפרויקטון
- במידת האפשר בשלב מקדמי זה, לציין האם יש מודולים נוספים בהם אתם מתכוונים להשתמש במסגרת הפרויקטון.
- תחום האחריות המתוכנן של כל סטודנט בצוות.

2. קבלת אישור ממרצה הקורס או לחלופין, על פי דרישת המרצה, קביעת פגישה בזום לדיון בנושא ורק לאחריו יתקבל אישור, על פי מצב ההצעה.
   - ללא אישור בכתב אין להתחיל עם המימוש
3. מימוש
4. הגשה, על פי המתואר בשני השקפים הקודמים

הנושאים המוצגים בשקפים באים מהווים רק רעיונות. אין, וגם מומלץ שלא לבחור מהם, אלא להציע רעיון שקרוב ללבכם

סימולציה של משחק כדורגל FIFA

- בפרויקט זה יבנה משחק כדורגל מבוזר. שתי קבוצות כדורגל, כ"א 11 שחקנים + שופט.
  - לכל קבוצה יש 2 שחקנים שניתן להחליף תוך כדי משחק. זהו פרויקט המשחק היחיד שיותר השנה.
  - כ"א מהשחקנים ממומש ע"י מכונת מצבים gen_statem.
  - יש לפחות 4 תפקידים שונים בכל קבוצה שוער (שחקן אחד) - תוקף, קשר ומגן (במערך שיבחר ע"י ה"מאמן" = מפעיל הפרויקט) .
- המגרש מחולק לארבעה רביעים. כל רביע מנוהל ע"י מחשב נפרד באמצעות gen_server.



- כל קבוצה תשתמש באסטרטגיית משחק שונה.
- הכדור מיוצג ע"י מכונת מצבים gen_statem.
- מחשב חמישי מנהל את הגרפיקה - כדור, שחקנים, שופט ושוער.
- יש לממש את חוקי משחק הכדורגל ברמה מספקת כדי לאפשר התנהלות עצמאית של המשחק.
- יש לאפשר למנהל המשחק להשתלט על שחקן אחד (2 מנהלים, אחד לכל קבוצה, כל אחד פועל ממחשב אחר(ולאפשר למנהל לקבוע לו את המהלכים בצורה ידנית תוך כדי שימוש במקלדת או עכבר).
- המחשב הגרפי יציג גם סטטיסטיקות שונות לגבי השחקנים (מהירות תנועה, מרחק מצטבר, מספר כיבושים וכו').
- תוצג גם סטטיסטיקה על מספר השחקים בכל אחד מ-4 האזורים כתלות בזמן לאורך המשחק, וכנ"ל סטטיסטיקות של ממוצע וסטיית תקן לכל רבע בנפרד

משחק רשת (השמדת אסטרואידים)

- שלושה מחשבים שולטים בשלושה חלקים של אזור המשחק. כל מחשב נשלט ע"י מקלדת/עכבר של משתמש שונה
  - הוספה / הסרה דינמית של מחשבים מהמשחק (עם מינימום של אחד, מקסימום יותר מ-3)
- אלמנטים
  - אסטרואידים מסוגים שונים (לפחות שלושה סוגים)
  - כלי נשק מסוגים שונים (לפחות 2)
  - כל אלמנט מעדכן תהליך שרת באשר למיקומו בפונקציה של הזמן - שרתי OTP
  - גרפיקה מטופלת ע"י מחשב ייעודי נוסף תוך שימוש ב- WxWidgets אשר יתפקד כשרת OTP

סימולציה של מערכת ההגנה "כיפת ברזל"

- הגדרות של מיקום יישובים ישראלים ויישובים "עוינים"
  - השטח מצוי בפיקוח של לפחות שלושה מחשבים, כ"א מנהל את השטח שלו בלבד ומכיל סוללת "כיפת ברזל" יחידה עם מכ"ם עצמאי משלה.
- אלמנטים
  - טילים (לפחות שני סוגים) מתעופפים לכוון ישובים ישראלים מיישובים "עוינים" עם הסתברות פגיעה מוגדרת מראש, אשר תקבע את המסלול. הטיל יעדכן את המכ"ם על אודות מיקומו
  - מכ"ם של כיפת ברזל (שרת OTP) ישערך האם יש או אין פגיעה ובהתאם תינתן הוראת יריה לשני טילי כיפת ברזל
  - המכ"ם "עוקב" אחרי הטילים - מעשית הטיל מדווח למכ"ם באשר למיקומו
- מחשב ייעודי לגרפיקה
  - שרת OTP
  - יקבל נתונים מהמכ"מים על מיקום טילי אויב, וטילי כיפת ברזל
  - שימוש ב-WxWidgets לצורכי ייצוג גרפי

מידול רשתות רכבים וחלוקה לאשכולות תקשורת

- מפת כבישים נתונה (2 מסלולים X 2 נתיבים)
- מכוניות נעות לאורך הכביש בשני כוונים
- מימוש האלגוריתם לחישוב אשכולות
- הקצאת חריצי זמן לתקשורת - TDM
- ארועים של התקרבות, עקיפה, פקק, תאונה
- העברת נתונים בין מכוניות
- מחשב ייעודי אחראי לגרפיקה (שרת OTP)
- המחשה גרפית של האשכולות (אליפסה...)

MapReduce

- הפעלת MapReduce על מאגר נתונים
- דוגמא - DBLP, IMDB
- חישוב גרף המרחקים מתחיל משם נתון של חוקר (או שחקן במקרה של IMDB)
- ייצוג תוצאות גרפים באמצעות GraphViz.
- קריאה לגרפיקה ישירות דרך Erlang
- שימוש ב-GraphViz
- שמירת תוצאות בקבצים מתאימים
- הפעלה מממשק גרפי וגם משורת הפעלה

חישוב PageRank מבוזר באמצעות BSP

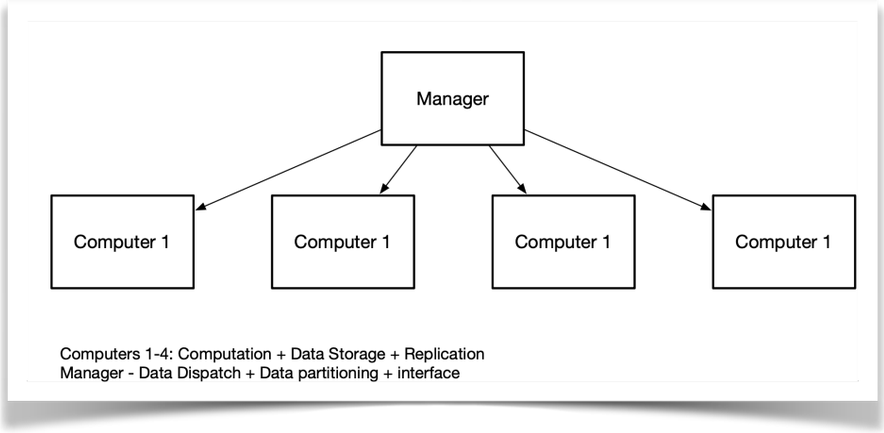
- זיהוי הרשת האוניברסיטאית באמצעות זחילה ליצירת בסיס נתונים מתאים
- שימוש בארבעה מחשבים לפחות
- סטטיסטיקות ביצועים לחלוקת מטלות שונה של אותו האלגוריתם
- שימוש בתכונות MultiCore
- מחשב ייעודי לייצוג גרפי של התהליך והתוצאות - שימוש ב-OTP
- הכנסת חיפוש ושליפה של דפי Web על פי הניקוד של האלגוריתם

עיבוד מקבילי ומבוזר של גרפים גדולים מתוך רשתות חברתיות באמצעות מודל BSP: חישוב PageRank מבוזר באמצעות BSP

- המידע לניתוח יהיה גרף מקושר גדול (large connected graph) שמכיל לפחות מספר מאות אלפים עד מליוני קדקודים. הקשתות ממושקלות.
- בשלב הראשון יש לייצר גרף אקראי ממושקל.
  - עליכם לחשוב על דרך יעילה לאחסן את הגרף הזה לאחר החלוקה שלו ב-4 מחשבים שונים לפחות.
- באמצעות אלגוריתם BSP אשר יופעל בצורה מקבילית עם כל הליבות של כל מחשב ועל 4 מחשבים לפחות. העבודה חייבת להתנהל במקביל על כל אחד מהמחשבים מחשב תוך שימוש בכל הליבות שלו בצורה מבוזרת.

יממושו האלגוריתמים הבאים (יישום מקבילי ומבוזר):

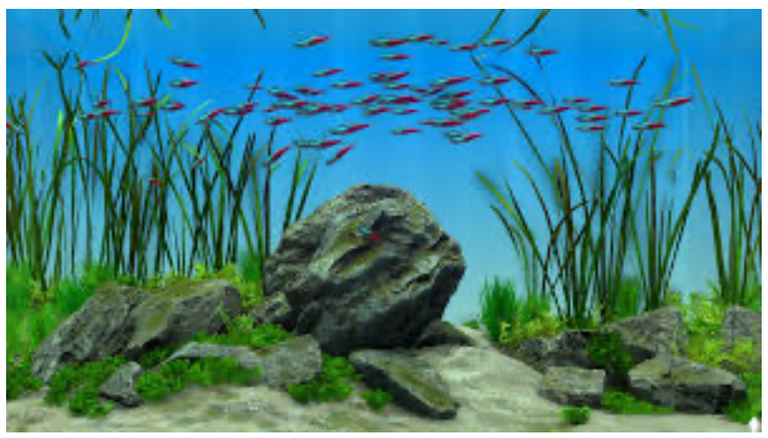
1. חישוב עץ המסלול הקצר ביותר מקדקוד "שורש" נתון לכל שאר הקדקודים
2. העץ הפורש בעל המשקל המינימלי
3. עץ פורש כאשר יש חסם על הדרגה של כל קדקוד.



- יש להריץ לפחות עם 20 קודקודי "שורש" שיבחרו אקראית, למדוד זמני הרצה של כל היישום, ואת זמני הריצה של כל אחד מהמחשבים (בחלק של ה-BSP).
  - יש למדוד כמה processes רצו בכל אחדמהמחשבים עבור כל אחד מקדקודי ה"שורש".
  - יש לספק נתונים סטטיסטיים לגבי זמן ריצה (ממוצע וסטיית תקן) לגבי כל אחד מהמחשבים, וגם לגבי היישום הכללי (מהרגע שהיישום התחיל לרוץ ועד לסוף הריצה של המחשב האחרון), כמות processes בכל אחד מהמחשבים ע"י אלגוריתם ה-BSP.

אקוריום

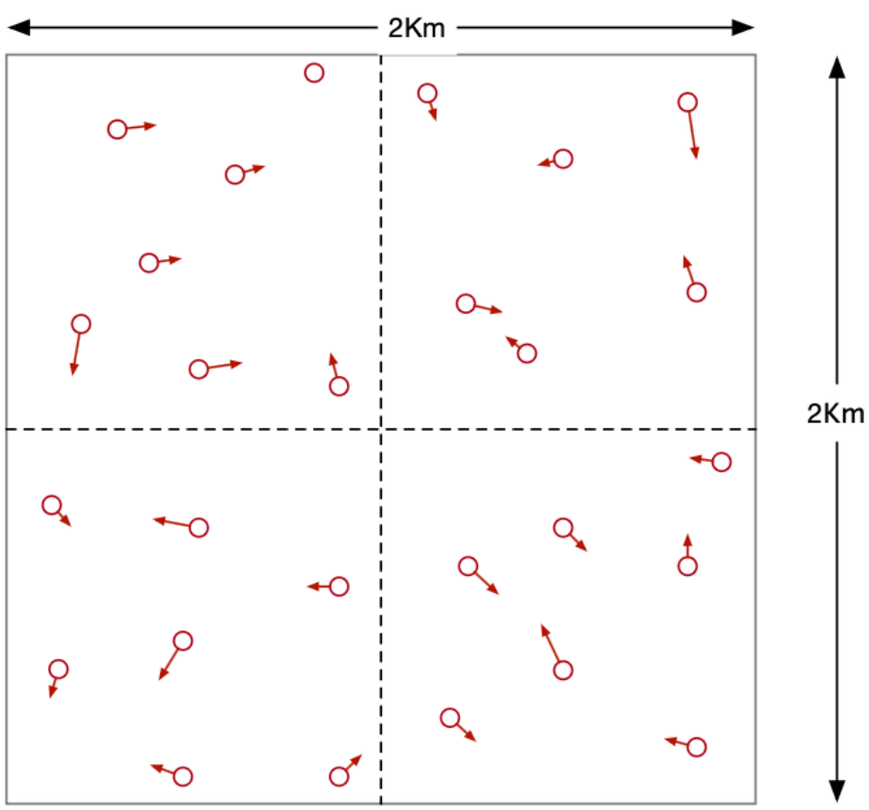
- דגים שוחים באקוריום - יש לקבוע מודל לשחיה
  - אזור שחיה מחולק ל-3, כ"א באחריות מחשב נפרד. כל אזור מבוקר ע"י שרת OTP מתאים
    - תנועה בתלת מימד (3D). דג שעובר אזור, הופך להיות תהליך בצומת המתאים ומפסיק להיות כזה בצומת אותו עזב. יש להגדיר פרוטוקול מעבר מתאים
- "ארועים חיצוניים" של



  - הפחדה (הקלקה עם העכבר) ותגובת הדגים בהתאם (עם רדיוס קרבה למוקד ההפחדה)
  - פיזור מזון מלמעלה (אוטומטי או בדיד באמצעות העכבר)
    - התכנסות של הדגים בהתאם לקרבה לפירור המזון
    - התמעטות המזון בקצב שמתאים למספר (חסום) של דגים שאוכלים ממנו
  - הוספה של דגים חדשים באמצעות הקלקות עכבר, או הגדרה כמותית באמצעים אחרים (עדיפות לאמצעים גרפיים וגם באמצעות תפריטים)
  - מחשב נוסף ישמש לשרת האחראי על הייצוג גרפי של הדגים, האוכל, מיקום העכבר וכו'
    - שימוש ב-WxWidgets

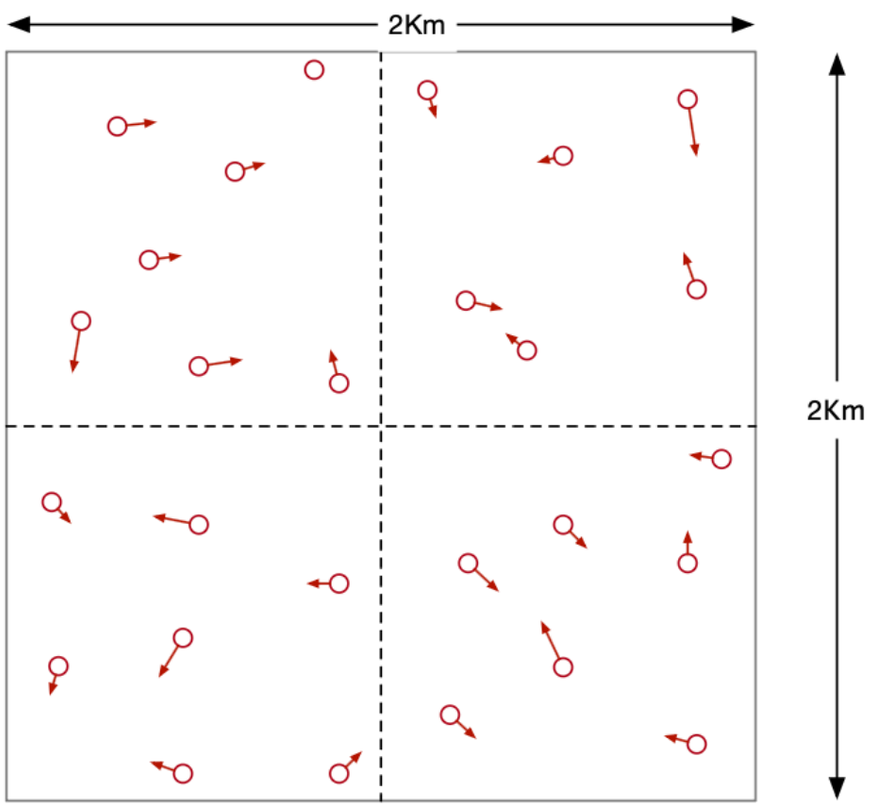
ניתוב ברשתות אד הוק אלחוטיות באמצעות פרוטקול TBRPF

- הרשת תכלול בין מאות לאלפי קדקודים
- מודל תנועה אקראי (בחירת כיוון והתנגשות בקירות על פי חוק סנל)
- כל מחשב (צומת Erlang) אחראי על התחנות שבאיזור שלו
  - תחנה שעוברת לאיזור שליטה אחר תופעל בצומת האחראי לאיזור
- שימוש ב-OTP
- גרפיקה באמצעות GraphViz ו/או-WxWidgets
- איסוף סטטיסטיקות
- העברת נתונים בין התחנות, על פי אלגוריתם הניתוב, והטבלאות המתוחזקות בו



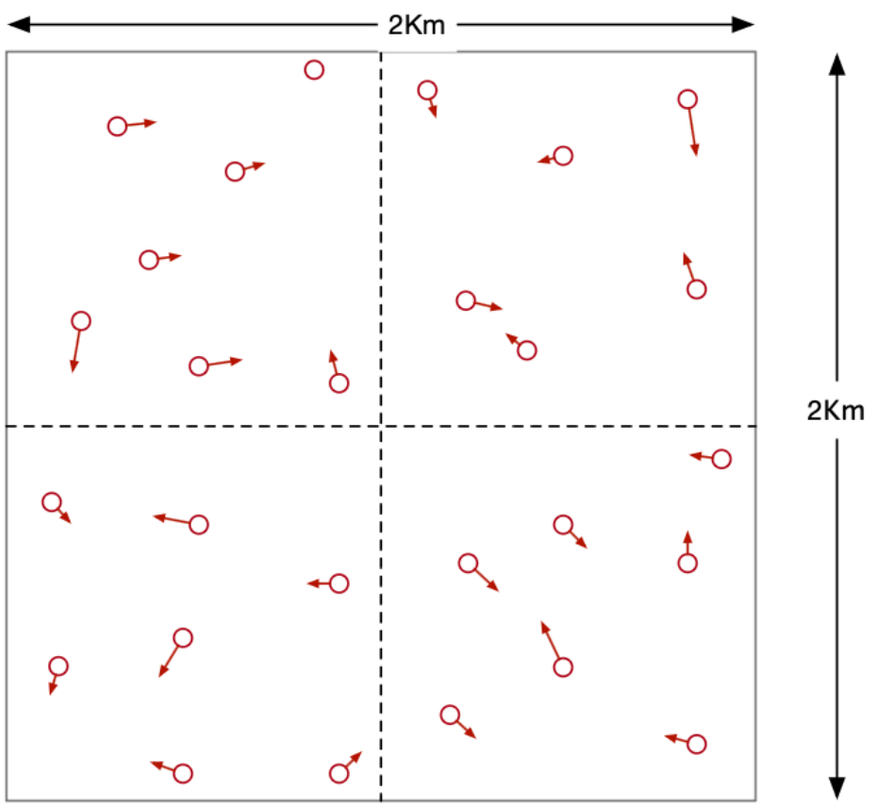
ניתוב ברשתות אד הוק אלחוטיות באמצעות פרוטקול RPL

- הרשת תכלול בין מאות לאלפי קדקודים
- מודל תנועה אקראי (בחירת כיוון והתנגשות בקירות על פי חוק סנל)
- כל מחשב (צומת Erlang) אחראי על התחנות שבאיזור שלו
  - תחנה שעוברת לאיזור שליטה אחר תופעל בצומת האחראי לאיזור
- שימוש ב-OTP
- גרפיקה באמצעות GraphViz ו/או-WxWidgets
- איסוף סטטיסטיקות
- העברת נתונים בין התחנות, על פי אלגוריתם הניתוב, והטבלאות המתוחזקות בו



ניתוב ברשתות אד הוק אלחוטיות באמצעות פרוטקול LoadNG

- הרשת תכלול בין מאות לאלפי קדקודים
- מודל תנועה אקראי (בחירת כיוון והתנגשות בקירות על פי חוק סנל)
- כל מחשב (צומת Erlang) אחראי על התחנות שבאיזור שלו
  - תחנה שעוברת לאיזור שליטה אחר תופעל בצומת האחראי לאיזור
- שימוש ב-OTP
- גרפיקה באמצעות GraphViz ו/או-WxWidgets
- איסוף סטטיסטיקות
- העברת נתונים בין התחנות, על פי אלגוריתם הניתוב, והטבלאות המתוחזקות בו



סימולציה של מערכת ביולוגית - שועלים, שפנים וגזרים

- תנועה אקראית של השועלים והשפנים. חיה שמתקרבת למקור מזון (גזר או שפן) מתכנסת אליו. צריכת מזון מאפשרת ריבוי באוכלוסיה. הרעבה גורמת למיתה (הקטנת אוכלוסיה)
  - יש מרחק מינימלי אשר רק ממנו שועל יכול לטרוף את השפן
- גרפיקה באמצעות WxWidgets - נשלט ע"י מחשב אחד
- שלושה מחשבים לפחות מכילים את "שדה המערכה"
  - כ"א אחראי על אזור גיאוגרפי מוגדר - שימוש ב-OTP מתאים לכ"א מהם
  - חיה שעוברת אזור - הופכת לתהליך חדש במחשב האחראי על האיזור ונמחקת מהאזור ממנו יצאה

ניתוב נמלים- Ant Colony Routing

- הרשת תכלול בין מאות לאלפי קדקודים
- מודל תנועה אקראי של "נמלה" בשלב החיפוש של מזון
- התכנסות למקור מזון על פי מרחק
- סימון מסלול בשלב החיפוש ולאחר המציאה
- נמלה שמתקרבת למסלול שמסמל מזון תשתמש בו
- כל מחשב (צומת Erlang) אחראי על הנמלים שבאיזור שלו
  - נמלה שעוברת לאיזור שליטה אחר תופעל בצומת האחראי לאיזור
- שימוש ב-OTP
- גרפיקה באמצעות WxWidgets
- איסוף סטטיסטיקות

מידול אלגוריתמי תנועה של להקת ציפורים

- אזור נשלט ע"י ארבעה מחשבים - שימוש ב-OTP
- גרפיקה נשלטת ע"י מחשב נוסף - שימוש ב-OTP
- כל "ציפור" מהווה תהליך במחשב האחראי על האיזור שלה
- אלגוריתם מקומי של כל ציפור לקביעת מיקום ביחס לשכנים
  - מבוצע דרך תקשורת בין ה"ציפורים"
- קביעת מסלול ומנהיג, הוספה לשליטה באמצעות העכבר כך שהלהקה תעקוב אחריו
- תנועה בתלת מימד (3D)

איזון עומסים במערך אחסון מבוזר

- חמש תחנות **לפחות** מתחזקות בסיסי נתונים בפורמט key → value (כולן עובדות עם OTP)
- מחשב שליטה אשר יבקר את העומס על כל אחת מהתחנות (שרת OTP)
- שתי תחנות אשר מייצרות עומס באמצעות מאות עד אלפי תהליכים אשר כ"א מדמה משתמש אשר יכול לאחסן מידע, למחוק מידע, לעדכן מידע, לשלוף מידע
- ממשק משתמש גרפי בסיסי אשר יציג את מצב המערכת
- הנחיות מדויקות יתקבלו מיחיאל

מידול של תנועה והקצאת ערוצים ברשתות תאיות עם (ובלי) חלוקה לסקטורים

- פרישה דו מימדית באמצעות משושים - על כל משושה אחראי שרת ערוצים מתאים
- צרכנים נעים על פי מודל תנועה דו ממדי (2D) מוגדר בין הערוצים. צרכן שעובר משושה, מקבל ערוץ מהשרת המתאים ומפנה את הערוץ בו השתמש
- מעקב על המרחק מהאנטנה המשרתת ויצירת קשר עם התחנות המשרתות הפוטנציאליות
- שרתי ערוצים יכולים להעביר ערוצים לאזורים עמוסים יותר (מעקב על עומס השיחות)
  - שימוש בפרוטוקול הודעות מתאים לצורך הניהול
- גרפיקה באמצעות WxWidgets נשלטת ע"י מחשב ייעודי
- איסוף סטטיסטיקות מתאים ומודל סטטיסטי של השיחות - ידון עם המרצה במקרה שהפרויקט ייבחר