**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Теплоенергетичний факультет**

**Кафедра автоматизації проектування  
енергетичних процесів і систем**

***Звіт***

***з лабораторної роботи №3***

***"*** ***Reservoir Sampling"***

***з дисципліни***

***"Обробка потокової інформації"***

**Виконав:**

Студент групи ТІ-01мп

Стеценко Д. О.

**Перевірив:**

Д.ф.-м.н.

Матичин І.І.

**м. Київ – 2021 р.**

**Завдання**:

Написати програмну реалізацію алгоритму резервуарної вибірки (Reservoir Sampling). Продемонструвати її роботу на прикладі потоку даних, що генерується продюсером, створеним в рамках Лабораторної роботи №1.

**Передумови**:

Для реалізації конс’юмера даних була використана платформа **Node.js**, бібліотека **Kafka.js**, бібліотека **Rx.js** для створення і генерації потоку даних, та створений у Лабораторній роботі №1 продюсер. Припустимо, що Zookeeper і брокер Kafka вже запущені.

**Хід виконання**:

Спочатку створюємо екземпляр об’єкту kafka з параметрами підключення, потім на основі нього ініціалізуємо consumer, як у лабораторній №2. З коду програми видно, що ми зберігаємо S перших повідомлень, потім замінюємо випадкове збережене повідомлення новим з ймовірністю S / N, де N – це кількість оброблених повідомлень.

Код конс’юмера для отримання даних координат (з частотою .5 секунди) наведений нижче та за посиланням:

<https://github.com/oddestdan/kafka-streams/tree/main/task-3>

const { Kafka } = require('kafkajs');

const { handleErrors } = require('./error-handling');

const { printSwapValues } = require('../shared/utils');

let processedCount = 0;

const S = 10;

const savedMessages = [];

const kafka = new Kafka({

  clientId: 'coords-app',

  brokers: ['localhost:9092'],

});

const consumer = kafka.consumer({ groupId: 'random-coords' });

const run = async () => {

  console.log('Consumer is running...');

  await consumer.connect();

  await consumer.subscribe({ topic: 'random-coords', fromBeginning: false });

  await consumer.run({

    eachMessage: async ({ *topic*, *partition*, *message* }) => {

      processedCount++;

      console.log('> ', message.value.toString());

      if (processedCount <= S) {

        savedMessages.push(message.value.toString());

      } else {

        const shouldReplace = Math.random() < S / processedCount;

        if (!shouldReplace) {

          return;

        }

        const i = Math.floor(Math.random() \* savedMessages.length);

        const toBeReplaced = JSON.parse(savedMessages[i].toString());

        const toReplaceWith = JSON.parse(message.value.toString());

        printSwapValues(toBeReplaced, toReplaceWith);

        savedMessages[i] = message.value.toString();

      }

    },

  });

};

run().catch((*error*) => {

  console.error(error);

  process.exit(1);

});

process.on('SIGINT', function () {

  consumer.stop();

  console.log('\nTotal processed messages: ', processedCount);

  console.log('Kept messages:', savedMessages.length);

  console.table(savedMessages.map((*m*) => JSON.parse(m)));

});

handleErrors();

Результат виконання програми:

$ node index

Consumer is running...

{"level":"INFO","timestamp":"2021-05-25T06:58:46.008Z","logger":"kafkajs","message":"[Consumer] Starting","groupId":"random-coords"}

{"level":"INFO","timestamp":"2021-05-25T06:58:46.036Z","logger":"kafkajs","message":"[ConsumerGroup] Consumer has joined the group","groupId":"random-coords","memberId":"coords-app-3b13ebc6-d9dc-4a1f-bff8-0d34738af530","leaderId":"coords-app-3b13ebc6-d9dc-4a1f-bff8-0d34738af530","isLeader":true,"memberAssignment":{"random-coords":[0]},"groupProtocol":"RoundRobinAssigner","duration":27}

> {"lat":"7.4607","long":"38.6991","id":"b8avvlakrmm","coordId":0}

> {"lat":"14.8615","long":"-5.4567","id":"alpbol10th","coordId":1}

> {"lat":"-77.9967","long":"-32.7770","id":"w4y0ayhvdx","coordId":2}

> {"lat":"84.3952","long":"-83.4077","id":"52za60vdmuh","coordId":3}

> {"lat":"-16.9410","long":"-171.3485","id":"l8ipj9fwx1","coordId":4}

> {"lat":"-40.7738","long":"-19.2683","id":"j7kua7lfopi","coordId":5}

> {"lat":"-44.2574","long":"-46.5093","id":"j4br7d8ogt","coordId":6}

> {"lat":"60.7112","long":"-75.9898","id":"xyohanooqu","coordId":7}

> {"lat":"-56.4736","long":"-22.6965","id":"gx3mduxgkvg","coordId":8}

> {"lat":"14.3327","long":"133.5315","id":"x7ibiksz4","coordId":9}

> {"lat":"31.6960","long":"53.2120","id":"mikd0cltpaj","coordId":10}

{ lat: '-40.7738', long: '-19.2683', id: 'j7kua7lfopi', coordId: 5 }

⬇ ⬇ ⬇

{ lat: '31.6960', long: '53.2120', id: 'mikd0cltpaj', coordId: 10 }

> {"lat":"-65.3016","long":"-69.3325","id":"gw4lfgwykpg","coordId":11}

{ lat: '-56.4736', long: '-22.6965', id: 'gx3mduxgkvg', coordId: 8 }

⬇ ⬇ ⬇

{ lat: '-65.3016', long: '-69.3325', id: 'gw4lfgwykpg', coordId: 11 }

> {"lat":"-58.1141","long":"-121.7228","id":"hw9l1iue9ui","coordId":12}

> {"lat":"-62.9687","long":"108.7397","id":"jhg13dvcyj","coordId":13}

{ lat: '84.3952', long: '-83.4077', id: '52za60vdmuh', coordId: 3 }

⬇ ⬇ ⬇

{ lat: '-62.9687', long: '108.7397', id: 'jhg13dvcyj', coordId: 13 }

> {"lat":"-61.1113","long":"110.0175","id":"cg97fzdl7tb","coordId":14}

Total processed messages: 15

Kept messages: 10

┌─────────┬────────────┬─────────────┬───────────────┬─────────┐

│ (index) │ lat │ long │ id │ coordId │

├─────────┼────────────┼─────────────┼───────────────┼─────────┤

│ 0 │ '7.4607' │ '38.6991' │ 'b8avvlakrmm' │ 0 │

│ 1 │ '14.8615' │ '-5.4567' │ 'alpbol10th' │ 1 │

│ 2 │ '-77.9967' │ '-32.7770' │ 'w4y0ayhvdx' │ 2 │

│ 3 │ '-62.9687' │ '108.7397' │ 'jhg13dvcyj' │ 13 │

│ 4 │ '-16.9410' │ '-171.3485' │ 'l8ipj9fwx1' │ 4 │

│ 5 │ '31.6960' │ '53.2120' │ 'mikd0cltpaj' │ 10 │

│ 6 │ '-44.2574' │ '-46.5093' │ 'j4br7d8ogt' │ 6 │

│ 7 │ '60.7112' │ '-75.9898' │ 'xyohanooqu' │ 7 │

│ 8 │ '-65.3016' │ '-69.3325' │ 'gw4lfgwykpg' │ 11 │

│ 9 │ '14.3327' │ '133.5315' │ 'x7ibiksz4' │ 9 │

└─────────┴────────────┴─────────────┴───────────────┴─────────┘