# **Визначення відстані Хеммінга в розподілених інфраструктурах для пошуку подібності.**

**Аннотація**. Сьогодні обсяги даних доступних в Інтернеті перевищують зетабайти (ZB). Ця умова визначає явище яке відоме в літературі як Big Data. Хоча традиційні бази даних дуже ефективні для пошуку та отримання потрібного контенту, вони не є ефективними в умовах роботи з великими обсягами даних, оскільки велика частина цих даних є неструктурованою та розподіленою в Інтернет. Таким чином для забезпечення пошуку подібності необхідні нові бази даних. Для того аби справлятися з даною ситуацією в даному розділі пропонується досліджувати схожість за Хеммінгом між ідентифікаторами контенту, що згенеровані з використанням функції випадкової гіперплощини хешування. Такі ідентифікатори забезпечують основу для побудови розподілених інфраструктур, що спрощує пошук подібності. В цьому розділі м представимо два різні підходи: P2P (Procure to Pay?) (Hamming DHT (Distributed Hash Table)) та Data Center (HCube). Представлені оцінки що вказують на те, що обидва підходи здатні покращити результати пошуку подібності.

## **1. Вступ**

Згідно сучасного розуміння явища Big Data, користувачі стали джерелами даних, компанії зберігають незліченні обсяги інформації про клієнтів, мільйони сенсорів отримують інформацію з середовища, створюючи та обмінюючись даними в Інтернеті речей. Згідно дослідження International Data Corporation (IDC), опублікованого в травні 2010 [1], кількість даних доступних в Інтернеті перевищує 2 ZB в 2010, подвоююись кожні 2 роки і може перевищити 8 ZB в 2015. Дослідження також показало що близько 90% цих даних складаються з неструктурованих, гетерогенних та різних за природою даних як тексти, зображення та відео.

Нові технології такі як Hadoop [2] та MapReduce [3] є прикладами рішень створених щоб вирішувати проблеми які накладає Big Data – обсяг, різноманітність та швидкість. Використовуючи пераленьні обчислення в поєднанні з грід технологіями, чи, як нещодавно, використовуючи інфраструктуру хмарних обчислень, IT організації пропонують засоби для роботи з широкомасштабними, розподіленими та інформаційномісткими (такими що використовують великі об’єми даних) задачами. Зазвичай такі технології пропонують розподілену файлову систему та автоматизовані інструменти для налаштування в процесі роботи (“на льоту”) кількості серверів залучених в обробку задач. В цьому випадку великі об’єми даних пропускаються через мережу і підключаючись до серверів передають пари <ключ, значення> з маперів до ред'юсорів щоб отримати необхідні результати. В цьому випадку бажано мінімізувати потреби в передачі даних по мережі щоб пришвидшити роботу всієї обчислювальної задачі.

В той час як дані рішення є, беззаперечно, ефективними для роботи традиційних застосувань, як, наприклад, пакетної обробки великих об’ємів даних, вони не забезпечують достатню підтримку пошуку подібності [4], ціллю якої є вилучення наборів подібних даних за даним рівнем подібності. Наприклад, схожість даних може бути використана в системі рекомендацій заснованій на інформації соціальних профілів користувачів. В даному прикладі профіль користувача може бути визначених як набір характеристик що унікально впливають на те як користувачі приймають рішення. Користувачі зі схожими характеристиками, скоріш за все, матимуть схоже інтереси та вподобання.

Таким чином, щоб зробити систему пошуку подібності засновану на даних профілів користувачів, характеристики користувачів в соціальній мережі можуть бути поміщені в вектор і, використовуючи Vector Space Model (VSM), подібність між користувачами може бути визначена використовуючи метрики відстаней між векторами як, наприклад, Евклідова відстань, косинус між векторами та Хеммінгова відстань. Але, за виключенням Хеммінгової відстані, інші метрики піддаються прокляттю розмірності (значний ріст необхідних обчислень при рості розмірності задачі) [4]. Великі обчислювальні витрати внаслідок розмірності задачі – це виклик що постає перед системами поуку подібності в випадку Big Data.

В цьому розділі описано як підтримувати пошук подібності використовуючи Хеммінгову відстань як метрику подібності. Щоб це досягти, дані індексуються в базі даних використовуючи LSH (Locality Sensitive Hashing) функцію що називається функцією RHH (Random Hyperplane Hashing) [5]. RHH – це сімейство LSH функцій, що використовують косинус між векторами та Хеммінгову відстань як метрику відстані між згенерованими двійковими рядками, тобто чим більший косинус між парою векторів контенту, тим менша Хеммінгова відстань між двійковими рядками. Ці двійкові рядки представляють ідентифікатори даних чия подібність можу бути виміряна Хеммінговою відстанню. Кожен запит в цій базі даних вираховується використовуючи Хеммінгову відстань між ідентифікатором запиту та кожним ідентифікатором даних.

При пошуку подібності запит що обробляється складається з того ж набору характеристик контенту індексованого в базі даних. Кожен користувач системи пошуку подібності вводить бажані характеристики та рівень подібності згідно очікуваного об’єму відповіді на запит. Чим більший рівень подібності тим менша кількість даних отримується внаслідок більшої специфічності (націленості) запиту. Запит індексується в базі даних з використанням RHH функції та подібності за Хеммінгом між запитом та всіма обчисленими ідентифікаторами даних в базі даних. Всі профілі чия подібність за Хеммінгом задовольняє бажаному рівню подібності повертаються у відповідь на запит.