

УДК 004.043

## ДОСЛІДЖЕННЯ ФРЕЙМВОРКІВ ДЛЯ ОБРОБКИ ГРАФІВ У ЗАВДАННЯХ АНАЛІЗУ ВЕЛИКИХ ДАНИХ

Чан С.Т.

Науковий керівник – к.т.н., доц. Кравець Н.С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

тел.: +38(093) 612-53-68, e-mail: suan.chan@nure.ua

The purpose of this work is to investigate the field of graph processing, namely database management systems, graph frameworks, their classification, specifics of use and characteristics. Graph-based structures are not relatively widespread in systems usage, despite their characteristics that are naturally suitable for processing in many areas, such as the World Wide Web, social networks, distributed system structures, knowledge bases, etc. Knowledge about them is not as widely known and popular as traditional relational solutions. The work analyzed the main classifications/characteristics related to graph frameworks, architectural properties, problems of their usage, and gave examples of the most famous of them. These results will provide essential insights into the world of big graph data processing.

Використання графів як структур даних, формат візуалізацій чи мережевих структур у програмних системах є достатньо поширеним явищем у різноманітних сферах використання програмного забезпечення. Прикладами такого використання є побудова та обслуговування мережі Інтернет, збереження даних та робота багатьох алгоритмів у соціальних мережах, побудова «баз знань» з подальшим використанням для пошуку даних та створенні аналітичних звітів, використання для побудови мережі нейронів у задачах машинного навчання та штучного інтелекту, тощо. Не дивлячись на те, що графові структури даних використовуються у різноманітних системах які грають важливу роль у просторі комп'ютерних технологій, вони не є достатньо освітленими у інформаційному просторі на відміну від використання таблиць чи об'єктів у реляційних та об'єктно-орієнтованих БД. Тому метою і результатами роботи є висвітлення ключових відмінностей, проблем та переваг графових фреймворків, що використовуються у задачах аналізу великих даних.

В дискретній математиці граф є представленням структури, що складається з множини об'єктів які можуть бути пов'язані між собою певними зв'язками. Математичне представлення простого неорієнтованого графу є  $G = (V, E)$ , де  $V$  – набір вершин,  $E \subseteq \{\{x, y\} | x, y \in V \text{ та } x \neq y\}$  – набір ребер [1]. Таким чином у графі акцентується увага не тільки на самих даних, але і на їх зв'язках між собою, що дозволяють використовувати їх у специфічних задачах, наприклад пошуку найкоротшого шляху чи пов'язаних даних (використовуючи «Теорію графів»). Такі властивості

створили бажання використовувати їх у програмних системах, що породило низку різноманітних інструментів для роботи з ними, але зі збільшенням об'єму даних для обробки виникли проблеми, основними з яких є: обчислення на основі даних (кроки алгоритмів обробки графів прямо залежать від структури самого графу), нерегулярність/гнучкість графів (через те, що структура графу досить гнучка, складно її ділити для рівномірного/оптимального збереження та для паралельної обробки), високе співвідношення процесів читання/запису до обробки (читання та запис даних займає велику частину усього процесу, через що він може займати більшу частину роботи алгоритму і є його вузьким місцем) [2]. Для зменшення впливу цих недоліків використовуються різні прийоми у фреймворках, такі як використання різноманітних архітектурних моделей обробки, моделей координації задач, оптимізацій на рівні кроків обробки графа, тощо. Загалом процес обробки графа можна поділити на наступні етапи: читання даних, попередня обробка, розбивання на частини, обчислення, запис даних. Алгоритми, що використовуються у обробці можна розділити за наступними класифікаціями: алгоритми обходу, аналітичні, розділення компонент, розділення спільнот, вимір мір центральності, анонімізація графу, інші специфічні алгоритми. За архітектурними моделями графові фреймворки можна поділити на такі типи: розподілена архітектура (система з множини обробляючих вузлів/комп'ютерів), архітектура основана на одному вузлі/комп'ютері, гетерогенна архітектура (система з множини різних за потужністю/характеристиками вузлів чи один комп'ютер з множиною додаткових прискорювачів, наприклад графічних процесорів). Самі графові фреймворки можливо поділити на категорії, що описують підходи до розділення та обробки даних: вершиноцентричні (які ділять дані за вершинами, прикладами таких є Pregel, GPS, Apache Giraph, GraphX), реброцентричні (які ділять дані за ребрами, прикладами таких є Chaos, X-Stream), графоцентричні (які ділять дані за компонентами графа, прикладами є Apache Giraph++, GoFFish) та інші (які використовують інші алгоритми для розділення даних, прикладами є PathGraph, MapReduce, Blogel). Таким чином результатом роботи є встановлення основних проблем графових фреймворків, висвітлення їх різновидів та класифікацій за певними параметрами, приведення їх прикладів, що можуть допомогти у наступних дослідженнях пов'язаних з обробкою графів або отримати основні уявлення про сферу графових фреймворків.

Список використаних джерел:

1. Bollobas, B. (2013). *Modern Graph Theory*. Springer New York.
2. Eisenman, A., Cherkasova, L., Magalhaes, G., Qiong, C., Farabochi, P., Katti, S. (2016). *Parallel Graph Processing: Prejudice and State of the Art*. ICPE '16, 85-90. <https://doi.org/10.1145/2851553.2851572>.