ПОСТРОЕНИЕ РЕЙТИНГА УНИВЕРСИТЕТОВ В СОЦИАЛЬНЫХ СЕТЯХ С ПРИМЕНЕНИЕМ ИНСТРУМЕНТОВ BIG DATA

А.И. Парамонов
Доцент кафедры программное обеспечение информационных технологий БГУИР, кандидат технических наук

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: anton_paramonov@tut.by

Abstract. The growth of social services gives rise to new tasks for data analysis and processing. These data are unsystematic and heterogeneous. The paper proposes an approach for building analytics university presence in social networks. The solution is based on the modern tools: database Neo4j, Spring Batch framework, the language of Big Data processing - R. As an example, demonstrate the relation of the two university communities in the social network. Construction ranking of universities by the criterion of the centrality of their community nodes.

В настоящее время анализ социальных сетей является мощным инструментом социологии, который нашел применение в различных областях науки. С появлением социальных сервисов, присутствие людей в интерактивных сетях многократно увеличилось, что дало социологам возможность анализировать большие объемы информации для выявления общих свойств коммуникации или определять связи тысяч пользователей, чтобы изучать тренды и культурные феномены Подобная глубокая аналитика стала возможной благодаря принятию сетевой концепции. Социальные сервисы сегодня связывают всё вокруг нас: людей, информацию и события. Логичным представлением структуры этих сервисов является сеть, в которой узлы-сущности связаны друг с другом. Подобную структуру можно и нужно анализировать для получения самой разнообразной информации: поиска наиболее важных узлов, определения отдельных сообществ, отслеживания потоков распространения информации и т.д. Результаты таких исследований находят применение в экономике, социологии, математике, биологии и многих других направлениях.

Основной проблемой в задаче анализа реальных социальных сетей на основе существующих социальных сервисов является крайне большой объём доступных, разрозненных и несистематизированных данных. Для обозначения

таких массивов данных, настолько больших, что анализировать их традиционным программным обеспечением не представляется возможным, был введён термин «Big Data» [1]. Подходы к обработке Big Data включают средства массово-параллельной обработки неопределённо структурированных данных, прежде всего, решениями категории NoSQL, алгоритмами MapReduce, программными каркасами и библиотеками проекта Hadoop. Одним из базовых инструментов для работы с большими данными является программная среда R[2], а также одноименный язык. Развитие направления аналитики больших данных формирует новые запросы и требования, а как следствие и новый инструментарий.

Многие социальные сервисы, в числе которых Facebook и Twitter, предоставляют собственные официальные средства для аналитики и статистики. Однако, наиболее популярный сервис подобного рода на постсоветском пространстве – сеть «ВКонтакте», пока не предоставил свои официальные инструменты для анализа и статистики данных о пользователях. Поэтому в работе предлагается решение для проведения анализа данных в сети «ВКонтакте», на основе предоставленного для разработчиков API социальной сети и современного инструментария обработки больших данных.

Для анализа информации следует определиться, какая информация о пользователях вообще представляет для нас интерес. При анализе социального графа можно выделить метрики, на основе которых будут производиться расчёты и оценки. В теории социальных графов выделают ряд метрик [3], из которых выделена и взята за основу в проекте характеристика центральности. Центральность относится к группе метрик, целью которых является определение «значительности» или «влияния» (в различных значениях) определённого узла (или группы) в сети. Центральность собственного вектора опирается на принцип, что связи с высокоранговыми вершинами увеличивают собственный рейтинг. Этот алгоритм является наиболее популярным для решения задачи ранжирования узлов по их важности. Математически центральность собственного вектора выражается таким образом. Пусть для графа $G = (V, E),$ где $|V|$ – число вершин, $A = (a_{v,t})$ – матрица смежности, где $a_{v,t} = 1,$ если узел υ связан с узлом t, и $a_{v,t} = 0$ иначе.

Тогда центральность узла υ это:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t \qquad (1)$$

где *M(*υ*)* – множество соседей υ, и λ – константа [4].

Информация, предоставляемая открытым API социальной сети, разрознена и ограничена возможностями, поэтому первоочередной задачей является консолидация этой информации в единое хранилище данных. Следует обратить

внимание на то, что развитие популярности обработки «больших данных» спровоцировало интерес к альтернативным способам хранения данных – так называемым Not-only-SQL (NoSQL) технологиям. Поскольку традиционные реляционные базы данных (SQL) сталкиваются с трудностями в масштабировании и процессах параллелизма. В последние годы появилось множество продуктов, направленных непосредственно на решение проблем хранения данных, которые возникают в связке с Big Data-аналитикой. Среди популярных сегодня можно выделить такие разработки как MongoDB, Redis, Neo4j [5].

Не меньшую важность представляет и вопрос агрегации информации из социальной сети, так как прежде чем сохранять данные их нужно откуда-то получить. Агрегация информации из социальной сети сопряжена с определёнными рисками. Если приложение должно собирать и анализировать данные в течение долгого периода времени, потеря соединения или ошибка в запросе могут привести к значительным потерям. Именно поэтому традиционные методы обработки данных уже не справляются с подобными задачами. Для решения этой проблемы предлагаются специальные фреймворки для длительной автономной обработки информации с высокой степенью отказоустойчивости и надёжности. В проекте предлагается использовать фреймворк Spring Batch [6]. Поскольку приложения Spring Batch легко масштабировать и конфигурировать для быстрой и отказоустойчивой обработки значительных объёмов данных. Дополнительным преимуществом является возможность интеграции с Neo4j – оба продукта написаны на Java и тесно взаимодействуют, основываясь на стандартные Java API.

Принимая во внимание все аспекты работы с большими данными, было разработано клиент-серверное приложение: серверная часть представлена VK API, который возвращает данные в ответ на запросы клиентской части – приложения-агрегатора. На рисунке 1 изображена диаграмма компонентов, описывающая архитектуру анализатора.

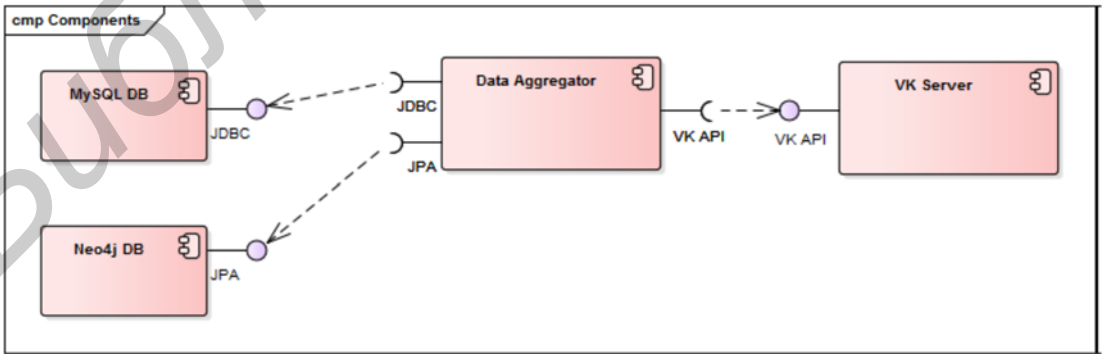

Рис. 1. Диаграмма компонентов приложения анализатора

В качестве контрольного примера исходными параметрами для сбора данных были определены идентификаторы страниц университетов (данные о подписчиках групп университетов) и глубина обхода их связей. Для сбора

данных используется открытое API сети «ВКонтакте» [7], доступ к которому осуществляется через GET-метод HTTP. На основе собранных данных формируется объектная модель для последующего импорта в базу данных. Каждой сущности устанавливается в соответствие узел графа. Таким образом программа формирует структуру графа. А этот граф анализируется и полученные по формуле 1 оценки сохраняются как параметры соответствующих вершин графа. В результате работы нашего приложения возвращаются значения оценок центральности узлов групп университетов в проанализированной сети. Для наглядности представления полученной модели строится визуальная репрезентация графа. Размер и цвет отдельных узлов можно задать пропорциональным их оценкам центральности. На рисунке 2 представлена визуализация подмножества графа, который описывает связи в сообществах двух университетов (узлы отмеченные желтой и голубой метками). Общее количество узлов в графе достигло трех миллионов. В данном случае приближенное значение центральности «желтого» узла равно 0.8766, в то время как приближенное значение другого сообщества – 0.0088 пунктов. Как видно из примера значение центральности «желтого» сообщества во сто крат превосходит конкурента.

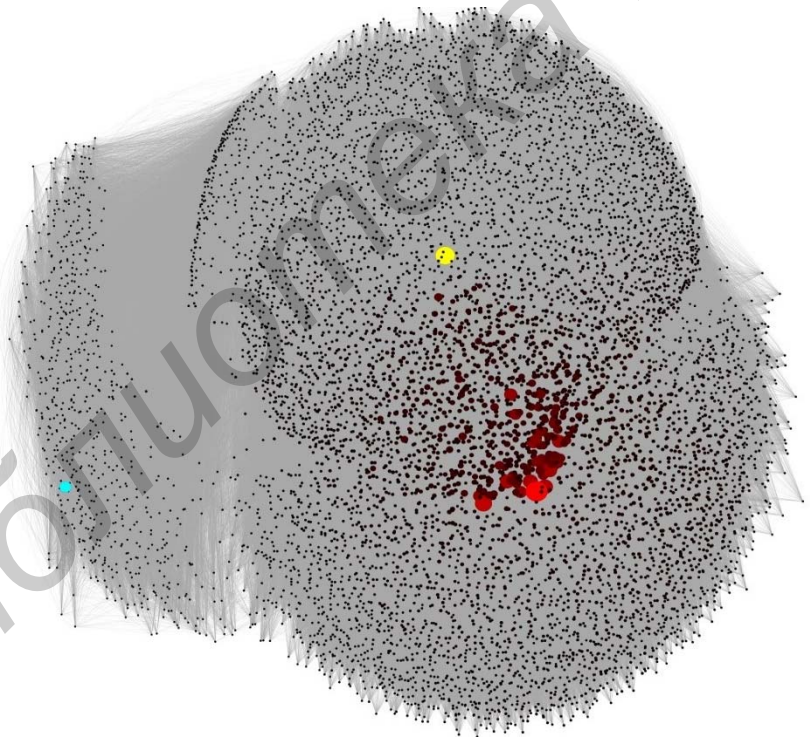

Рис. 2. Визуализация графа центральности сообществ двух университетов

Для обработки графа использован стандартный веб-интерфейс сервера Neo4j. Запросы к БД Neo4j осуществляются с помощью особого языка Cypher, спроектированного для эффективной работы с графовыми структурами данных. Для визуализации модели всего графа, а также результатов отдельных выборок, можно использовать интерактивный веб-интерфейс Neo4j. Так, например, на

рисунке 3 представлена интерактивная визуализация подмножества пользователей, состоящих в обеих сообществах университетов. Используя данный инструментарий можно получать различные репрезентации сущностей в социальной сети и связей между ними.

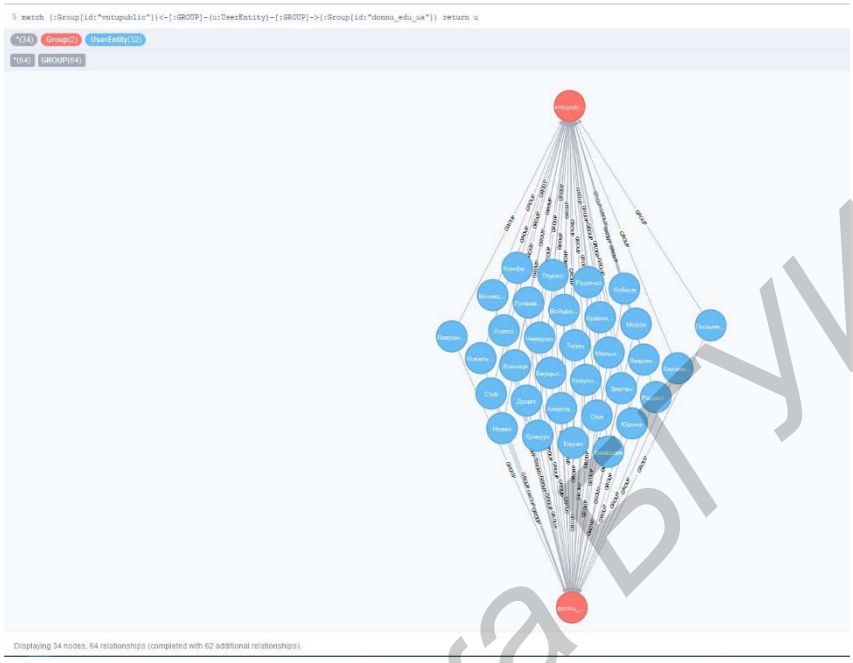

Рис. 3. Пример визуализации модели графа в веб-интерфейсе Neo4j

Литература

[1]. What is Big Data: [Электронный ресурс] // IBM. – 2015. URL: https://www.ibm.com/big-data/us/en/. (Дата обращения: 08.05.2016).

[2]. What is R: [Электронный ресурс] // Revolution Analytics. – 2015. URL: http://www.revolutionanalytics.com/what-r. (Дата обращения: 08.05.2016).

[3]. Scott J. Social Network Analysis / John Scott., 2012. – 216 с. – (Third edition).

[4]. Franceschet M. Eigenvector Centrality: [Электронный ресурс] // ‘Gabriele d’Annunzio’ University. – 2014. URL: http://www.sci.unich.it/~francesc/teaching/network/eigenvector.html. (Дата обращения: 08.05.2016).

[5]. Neo4j: The World's Leading Graph Database [Електронний ресурс] / Neo Technology, Inc. – 2016. URL: http://neo4j.com/. (Дата обращения: 08.05.2016).

[6]. Pivotal Software Inc. Spring Batch: [Электронный ресурс] // Spring. – 2016. URL: http://projects.spring.io/spring-batch/. (Дата обращения: 08.05.2016).

[7]. Выполнение запросов к API: [Электронный ресурс] // ВКонтакте. – 2015. URL: https://vk.com/dev/api_requests. (Дата обращения: 08.05.2016).