Сергеев Антон Валерьевич
аспирант кафедры интеллектуальных и информационных систем
Киевского национального университета имени Тараса Шевченко
г. Киев, Украина

РОЛЬ РАСПРЕДЕЛЁННЫХ ХЕШ-ТАБЛИЦ В ПОСТРОЕНИИ ЭФФЕКТИВНЫХ ДЕЦЕНТРАЛИЗОВАННЫХ СЕТЕЙ

Тенденция последнего десятилетия к росту объемов сетевого трафика привела к возникновению проблем в сетях, построенных на основе классической клиент-серверной архитектуры. Вследствие увеличения количества участников сетей, их географической распределённости, а так же объёма передаваемого контента между ними, в таких сетях проявляются критические недостатки связанные с масштабируемостью и отказоустойчивостью. Главным источником этих проблем является узкое место клиент-серверной топологии — единственный сервер, мощностей которого зачастую не хватает для обработки всех запросов, направленных к нему. Также, при выходе сервера из строя, из строя так же выходит вся сеть, что является особо критичным, особенно для коммерческих сетей.

Одним из возможных решений данных проблем является использование децентрализованных (одноранговых — peer-to-peer) сетей. Как следует из названия, у таких сетей отсутствует единый центральный сервер и каждый элемент сети исполняет роль как сервера, так и клиента. Среди приложений, построенных на основе P2P стоит выделить такие сервисы как SopCast, Bitcoin, Edonkey и другие.

В следствии распределённости данных по всей сети, для такой системы необходимы эффективные алгоритмы поиска, с помощью которых, элемент сети сможет найти необходимую ему информацию за приемлемое время и с использованием рационального объёма ресурсов.

Для реализации таких алгоритмов [1, с. 232–234], в одноранговых сетях зачастую используют оверлейную структуру — распределённую хеш-таблицу (Distributed Hash Table — DHT). Данная структура работает по принципу ассоциативного массива, то есть каждый участник сети может искать значение, ассоциированное с данным ключом. Помимо поиска, такая система позволяет эффективно добавлять, а так же удалять элементы сети. Преимуществом таких сетей так же является то, что каждый участник сети должен контактировать лишь с $O(\log n)$ элементов сети, где n — общее количество элементов сети.

Каждому элементу, участвующему в DHT, назначается уникальный идентификатор (ID), выбранный случайным образом в определённом ключевом пространстве (обычно, 160- или 128-битном). Ключом каждого файла является хеш-функция от его названия. Для определения, к какому элементу в сети будут привязаны данные, определяется метрика, называемая *расстоянием*. Ссылка на файл будет хранится у того узла, у которого расстояние между ID и ключом минимально.

Вопрос построения оптимального алгоритма для работы DHT является открытым [1 с. 232; 2]. Впрочем, эффективный должен отталкиваться от метрики расстояния, которая и является его базой. В отличии от большинства теоретически разработанных DHT сетей [3; 4; 5], в Kademlia [6 с. 4] в качестве метрики используется результат операции XOR (исключающее «или») в виде целого десятичного числа, а не количества отличающихся битов, т.е., например, $d(3,7) = 0011 XOR 0111 = 0100 = 4$. XOR арифметика формирует абелеву группу, что позволяет теоретически строго проанализировать протокол, тогда как для других же систем необходима сложная формализация с целью спрогнозировать надёжность и поведение сети. Целью алгоритма является нахождения узла, который хранит искомый файл. Алгоритм является рекурсивным и на каждом шагу рассчитывает вышеописанную метрику между данным ключом файла и идентификаторами элементов, с которыми элемент на данном шаге контактирует. Исполнение алгоритма происходит до тех пор, пока не будет найден файл, или не закончатся элементы сети. Так как зависимость количества элементов, с которыми контактирует узел зависит логарифмически от общего количества элементов сети, то при увлечении сети в два раза количество необходимых шагов для нахождения файла увеличится всего на один.

К недостаткам данной системы следует отнести недостаточное внимание к вопросам безопасности, что вылилось в исследованиях по атаке сетей, которые базируются на Kademlia [7; 8; 9].

В отличии от большинства DHT, разработанных исключительно теоретически, Kademlia имеет и практические реализации, такие как BitTorrent, Kad Network, Gnutella и другие.

В целом, не смотря на то, что децентрализованные сети, построенные на основе распределённых помогают решить проблемы, возникающие у сетей, построенных на основе клиент-серверной архитектуры, остаются недостатки, которые могут быть критичны в определённых сферах деятельности.

Литература

1. Г. В. Порєв. Методи та засоби побудови інформаційних технологій на основі територіально розосереджених сервіс-орієнтованих однорангових мереж. Київ — 2013.

2. H. Balakrishnan, M. Kaashoek, D. Karger, and I. Stoica, 2003. «Looking up data in p2p systems.» Comm. ACM 46, 2(Feb.)

3. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan, «Chord: A scalable peer-to-peer lookup protocol for internet applications», IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17–32, 2003.

4. S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. and Shenker, «A scalable content-addressable network.» In Proceedings of ACM SIGCOMM, San Diego, CA (August 2001).

5. A. Rowstron, and P. Druschel, «Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.» In Proceedings of the 18th IFIP/ACM Int'l Conf. on Distributed Systems Platforms (Nov. 2001);

6. Maymounkov P. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric / Maymounkov P., Mazieres D. // IPTPS2002, 7–8 March 2002 p.

7. P. Wang, J. Tyra, E. Chan-Tin, T. Malchow, D. F. Kune, N. Hopper, and Y. Kim, «Attacking the kad network,» in Proc. of the 4th Int. Conf. on Security and Privacy in Communication Netowrks (SecureComm '08), August 2008.

8. Thomas Locher, David Mysicka, Stefan Schmid, and Roger Wattenhofer. Poisoning the Kad Network. In 11th International Conference on Distributed Computing and Networking, Kolkata, India, 01 2010.

9. Thomas Locher, Stefan Schmid, and Roger Wattenhofer. eDonkey & eMule's Kad: Measurements & Attacks. Journal Fundamenta Informaticae, Volume 110, Number 3, 2011. (Journal version of a DYNAS2009 paper.)