УДК 004.75

Акимов В. И., Богданова Н. С., Винницкая Я. А.

ДЕЦЕНТРАЛИЗОВАННОЕ ХРАНЕНИЕ ИНФОРМАЦИИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ: ОБЗОР ТЕХНОЛОГИИ DHT

В последнее время развитие и совершенствование технологий сетевого взаимодействия, как и лавинообразное увеличение количества устройств в мировой сети, привело к формулировке и актуализации ряда задач, требующих работы с огромными массивами данных [1]. Это, в свою очередь, привлекло внимание исследователей и разработчиков в области информационных технологий, таких как: Шон Фэннинг, Джастин Франкель [2], Том Пеппер, Брэм Коэн, к системам и средствам распределенного хранения и поиска информации [3]. Одно из основных требований к таким системам – надежность, в частности, устойчивость работы всей системы к нарушениям функционирования отдельных узлов. Анализ проблемы показывает, что для достижения удовлетворительного уровня стабильности необходима децентрализация не только хранения, но и механизмов поиска и индексации внутри массивов данных. Одним из наиболее перспективных подходов к решению этой проблемы являются технологии класса DHT [4].

Целью статьи является обзор истории проблемы распределенного хранения и поиска информации, а также решение этой проблемы и области применения распределенного хранения информации.

Необходимо рассмотреть особенности, принцип работы и применение распределенной хеш-таблицы, как одного из вариантов решения проблемы хранения большого объема информации. Определить нерешенные проблемы в рамках распределенной хеш-таблице.

Приблизительно к 1995 году в развивающемся интернете возникла проблема – каналам связи стало не хватать пропускной способности для передачи музыкальных файлов в формате MP3. Выход из этой ситуации нашел 18-летний студент Шон Фэннинг, который придумал систему под названием Napster. Пользователь выкладывал в сеть сведения о файлах, которые у него есть, это заносилось в базу данных вместе с адресом этого компьютера. Любой желающий может обратится в базу данных и узнать адрес компьютера, потом установить с ним связь и скачать все, что ему нужно. Получается, любой может возпользоваться сетью Napster, только перед этим вложив что-то в эту сеть. Вскоре пользователей стало 70 миллионов, и в 2001 по судебному решению сеть закрыли.

Интернет очень быстро развивался, и идею Napster продолжила сеть Gnutella (одна из первых пиринговых сетей), созданная в 2000 году [5]. Клиент получает от узла, с которым он соединился список из 5 активных узлов, посылает им запрос на поиск ресурса по ключевому слову. А те в свою очередь отсылают запрос своим активным узлам вверх по дереву. Ясно, что на верхних ветвях дерева возникнут проблемы. Недостатки сети Gnutella привели к созданию DHT (Distributed Hash Tables) [6].

Сеть Fast Track протокол обмена файлами, который был реализован в программе KaZaa. Сеть использует «суперузлы» – временные базы данных, содержащие списки доступных файлов. Пользователь, чтобы воспользоваться данными «суперузла», должен выложить в него определенное количество информации.

Количество пользователей возрастало, и появились проблемы выделения больших объемов памяти в рамках одного компьютера для хранения информации, блокировки серверов, длительности поиска информации и ее скачивания. Также не каждый пользователь выкладывал информацию в сеть.

Идея хранения информации не на одном, а на нескольких компьютерах решает проблемы с выделением больших объемов памяти и обеспечения нужной скорости скачивания. Для этого стали использовать распределенную хеш-таблицу (англ. Distributed Hash Table).

Хэш-таблица – это структура данных, реализующая интерфейс ассоциативного массива, а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции [7]:

– операцию добавления новой пары;

– операцию поиска;

– операцию удаления пары по ключу.

Хэширование – это преобразование входного массива данных определенного типа и произвольной длины в выходную битовую строку фиксированной длины.

Функция, которая трансформирует ключ в некоторый индекс в таблице, называется хеш-функцией.

Результат хэш-функции называют хэшем, хэш-кодом, хэш-таблицей или дайджестом сообщения (рис. 1).

Для получения распределенной хеш-таблицы берется ключ и к нему применяется хеш-функция или она же функция свертки ключа. Таким образом, получаем хэш. В большинстве случаев длина ключа больше длинны хеш-функции.

Хэш – это обычное целое число от 0 до n. Каждый хэш содержит ip узла. В памяти хранятся только основные значения, весь массив данных не хранится [8].

ХЕШ	ССЫЛКА
0	ip узла
1	ip узла
2	ip узла
…	
n	ip узла

Рис. 1. Хэширование

Из упомянутых выше 3-х операций к ассоциативному массиву, рассмотрим поиск по ключу.

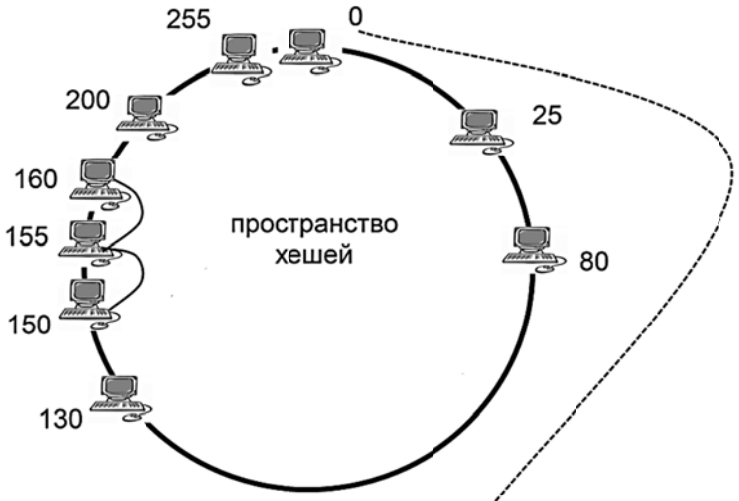
После того, как к ключу применена хеш-функция и получен хеш, становится актуальной проблема определения, на каком узле хранится информация по данному хешу. Поиск запускается на одном из узлов. В основе поиска лежит запрос.

Во всех запросах имеется ключ «id» и значение, содержащее ID запрашивающего узла. Во всех ответах имеется ключ «id» и значение, содержащее ID отвечающего узла.

Наиболее элементарным запросом является пинг. «q» = «ping». В пинг-запросе имеется один аргумент: «id» – 20-байтная строка (big-endian), содержащая ID узла отправителя. В правильном ответе на пинг содержится лишь один ключ «id», содержащий ID отвечающего узла.

Аргументы: {"id" : "<id запрашивающих узлов>"} ответ: {"id" : "<id ответивших узлов>"}

Каждый узел содержит соседей, в количестве около 5 узлов. Любой узел отвечает за определенный диапазон хешей [9]. Узел сравнивает длину своего хеша и искомого. В случае совпадения длины запрашиваемому узлу передается ссылка на узел, где расположен искомый файл [10, 11]. А если длина не совпадает, то данный узел обращается с запросом к диапазону своих соседних узлов и ожидает ответа (рис. 2).



ХЕШ	ССЫЛКА
0	ip узла
1	ip узла
2	ip узла
…	
n	ip узла

Рис. 2. Схематический вид структуры распределенной хеш-таблицы

DHT в БитТорренте используется для нахождения новых участников файлообмена. Реализация DHT в БТ основана на варианте DHT, называемом Kademlia, и использует UDP (быстрее, чем TCP, но без подтверждения ответа) протокол.

Kademlia – это безсерверная файлообменная сеть [12]. Она позволяет вам находить других пользователей напрямую без посредничества сервера, что значительно ускоряет поиск источников на файлы. Все пользователи являются серверами этой сети.

Каждый подключённый БТ клиент является в DHT сети отдельным узлом. У него есть свой уникальный ID (идентификатор), случайно выбираемый из того же 160-битного пространства, что и infohash'ы торрентов.

Когда узел хочет найти пиров ((peer) это клиент/сервер, прослушивающий TCP порт, к которому привязан bittorrent) для какой-то раздачи, он сравнивает infohash этой раздачи с ID известных ему узлов, и затем посылает запрос тому узлу, чей ID наиболее похож на этот infohash. Тот узел возвращает ему адрес узла, чей ID ещё ближе к infohash торрента [13].

DHT есть в различных клиентах BitTorrent (официальный), µTorrent, BitSpirit, BitComet, а также (несовместимо со всеми остальными) в Azureus [14].

DHT – Работает в независимости от трекера/ров и позволяет объединить все источники на раздаче. Например если раздача работает с DHT и на трекере А есть 30 сидов на раздаче, на трекере Б еще 40 сидов на той же раздаче и на трекере В еще 30 сидов, то с помощью DHT, вы будете скачивать с всех 100-та источников [15].

В более общем смысле DHT означает децентрализованную распределенную систему для объединения большого количества постоянно исчезающих и появляющихся узлов и эффективной передачи сообщений между ними [16]. На основе DHT структур можно строить более сложные системы, такие как P2P файлообмен, кооперативное веб кеширование, DNS сервисы и т. п.

Нерешенные проблемы DHT:

- Отсутствие унификации существующих протоколов.
- Проблемы безопасности, включающие сложность проверки целостности, безопасность маршрутизации, открытая публикация хешей.
- сильный паразитный трафик в имеющихся реализациях.
- Хэширование делает невозможным частичный поиск по ключу и содержанию.

ВЫВОДЫ

DHT – это перспективная, надежная, устойчивая, стабильная система хранения и поиска информации. Хранит в себе большой объем информации. Может использоваться как метод хранения данных в GRID системах и этот подход делает распределенную хеш-таблицу актуальной для исследования. Наличие нерешенных проблем гарантирует неослабевающий интерес научных сообществ к данному классу технологий.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Сюртуков И. Современные системы хранения данных. [Электронный ресурс] / И. Сюртуков. – Режим доступа : http://citforum.ru/hardware/data/overview/.

2. Альфа-гик из Аризоны: интервью с создателем WinAmp. 1. The Sellout. [Электронный ресурс] / Авторский техноблог – Режим доступа : http://blogerator.ru/page/winamp-frankel-interview.

3. Frederic P. Miller, Agnes F. Vandome, John McBrewster. Distributed Hash Table, Alphascript Publishing, 2010.

4. Интернет – Универмаг № 1 [Электронный ресурс] – Режим достапа : http://www.maria-online.com/electronics/article.php?q=Распределённая_хеш-таблица.

5. Электронный ресурс – Gnutella: www.gnutella.com.

6. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. / D. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, R. Panigrahy – In Proc. 29th Annual ACM Symposium on Theory of Computing (El Paso, TX, May 1997). – P. 654–663.

7. Алгоритмы. Построение и анализ / Кормен, Лейзерсон, Ривест, Штайн – Издание 2-е, Вильямс, 2007. – Глава 11. – С. 282.

8. Керниган. Практика программирования / Керниган, Пайк. – Издание 4-е, Вильямс, 2004. – Глава 2.9. – С. 72.

9. Левитин. Алгоритмы. Введение в разработку и анализ. / Левитин. – Вильямс, 2006 г. – Глава 7.3 –С. 323.

10. Словари и энциклопедии на Академике. – [Электронный ресурс] – Режим доступа :– http://academic.ru/dic.nsf/ruwiki/44718.

11. Bits of Mind. Заметки о программировании, и не только… – [Электронный ресурс] – Режим доступа : http://bitsofmind.wordpress.com/2008/07/28/introduction in hash tables/.

12. Peter Maymounkov, David Mazieres, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric", IPTPS 2002. [Электронный ресурс] – Режим доступа : http://www.cs.rice.edu/Conferences/ IPTPS02/109.pdf.

13. Ion Stoica, Robert Morris, David Karger,M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the ACM SIGCOMM '01 Conference, San Diego, California, August 2001.

14. Stefan Saroiu, P. Krishna Gummadi and Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001.

15. Электронный ресурс – Режим доступа : http://dht-tracker.org/forum/viewtopic.php?t=3313.

16. A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. Accepted for Middleware, 2001, 2001. [Электронный ресурс] – Режим доступа : http://research.microsoft.com/a˜ntr/pastry/.

Статья поступила в редакцию 10.09.2014 г.