УДК 004.421.2

ДВУХУРОВНЕВАЯ СХЕМА ОРГАНИЗАЦИИ ТАБЛИЦ РАСПРЕДЕЛЕНИЯ ЗАПРОСОВ В КЛАСТЕРНОЙ СИСТЕМЕ DNS

С. Н. Шилов, С. Д. Кургалин, А. А. Крыловецкий

Воронежский государственный университет

Поступила в редакцию 02.10.2013 г.

Аннотация. Данная статья посвящена проблеме распределения нагрузки на узлы в DNS кластере. В рамках статьи рассматривается модернизированный метод построения системы распределения нагрузки, основанный на распределенной хеш-таблице (Distributed Hash Table, DHT) с использованием двухуровневой схемы таблиц распределения. Статья предоставляет детальное описание принципов построения данной системы, а также сравнение с прежней одноуровневой схемой.
Ключевые слова: кластер, кластеризация, распределенная хеш-таблица, распределение нагрузки, балансировка нагузки, система доменных имен.
Annotation. The problems, bound with synthesis of the control algorithms by working machines for the coordination of rough and precise references of a difference in phase, coordinates setting and development of a correction signal of driving of mobile technique in road building are considered.
Keywords: cluster, clustering, distributed hash table, load distribution, load balancing, domain name system.

ВВЕДЕНИЕ

При реализации практически любых кластеризованных систем возникает задача выбора метода балансировки нагрузки на узлы кластера. В DNS кластерах предъявляются особые требования к подсистемам, выполняющим распределение входящих запросов между серверами, составляющими единую кластеризованную систему. В настоящей работе будут рассмотрены рекурсивные кэширующие DNS серверы, для которых содержание кэша имеет очень большое значение.

Наиболее эффективные решения при реализации подобных подсистем основаны на использовании распределенной хеш-таблицы (distributed hash table, DHT).

Данная статья является продолжением работы [1], в которой была представлена первоначальная реализация системы распределения нагрузки с одноуровневой организацией DHT. Здесь мы опустим описание различных подходов к построению подобных систем и объяснение того, почему применение именно DHT является здесь наиболее эффективным решением. Также мы лишь кратко коснемся описания системы распределения нагрузки, построенной на основе одноуровневой организации таблиц распределения запросов. Все эти вопросы развернуто описаны в статье [1]. В настоящей работе будут рассмотрены недостатки предыдущей реализации и их устранение с помощью усовершенствованной схемы применения DHT – двухуровневой организации таблиц распределения запросов – и ее детальное описание.

АНАЛИЗ ПРОБЛЕМЫ

Кластер – группа компьютеров, объединенных высокоскоростной сетью, использующаяся как единый, унифицированный компьютерный ресурс. В рамках данной статьи будет рассматриваться исключительно кластер рекурсивных кэширующих DNS серверов.

Кластеризация имеет ряд несомненных преимуществ по отношению к использованию отдельных высокопроизводительных машин. Прежде все это масштабируемость, высокая доступность сервиса (за счет использования множества узлов) и крайне выгодное соотношение стоимости оборудования и суммарной производительности [3].

Кластер состоит из нескольких узлов, в связи с чем возникает вопрос распределения нагрузки для обеспечения эффективности его функционирования. При этом существуют дополнительные требования к системе балансировки нагрузки, обусловленные спецификой работы рекурсивных кэширующих DNS серверов. Как было сказано выше, для таких типов серверов кэш и его содержимое имеют большое значение. Если запрашиваемая запись содержится в кэше, сервер незамедлительно отдает ее клиенту, не производя дополнительных запросов к вышестоящим серверам. Если сервер будет получать только определенные домены, то он будет максимально эффективно использовать свой кэш, что положительно скажется на его производительности. Таким образом, система распределения нагрузки должна не просто равномерно распределять запросы среди узлов кластера, но, в то же время, ретранслировать запросы с определенными доменами определенным узлам кластера [1].

Крайне важным моментом здесь является обеспечение масштабируемости кластера. Кластер является динамической системой, в которой состав участников может со временем изменяться. При этом можно рассмотреть две основные ситуации и соответствующие требования к поведению системы распределения нагрузки:

1) выход узла из состава кластера: обслуживаемая им часть запросов должна равномерно перераспределиться между оставшимися участниками кластера. При этом запросы, обслуживаемые оставшимися узлами, не должны перераспределиться для сохранения актуальности кэшей;

2) вход узла в состав кластера: узел должен «вернуть» себе именно свою часть запросов, которую он обслуживал до выхода.

Наиболее эффективным решением для построения системы распределения нагрузки является система, построенная на инновационном принципе распределенной хеш-таблицы.

Распределенная хеш-таблица (distributed hash table, DHT) – это класс децентрализованных распределённых систем, которые обеспечивают поисковый сервис, похожий по принципу работы на таблицу хешей. Иначе говоря, DHT – система, которая выполняет отображение хеш-значения, полученного от определенного параметра (в нашем случае это запрашиваемый домен, т.к. речь идет о DNS кластере), в узел системы. Одна из наиболее известных распределенных систем хранения Amazon's Dynamo построена именно на этом принципе [2].

Первоначальная реализация, описанная в статье [1], представляла собой схему с одноуровневой организацией DHT. Она содержала одну таблицу распределения, в которой находились только «живые» узлы кластера. При этом данная схема обладала высокими показателями равномерности распределения нагрузки и полностью удовлетворяла требованиям, предъявляемым к обработке входа узла в кластер.

Однако в ходе дальнейших исследований и эксплуатации выявились некоторые недостатки в обработке выхода узла из состава кластера. Допустим, таблица содержит четыре «живых» узла, условно обозначенных как 0, 1, 2, 3. Представим графически отображение таблицы распределения на область значений используемой хеш-функции (рис. 1).

Стоит напомнить, что вся область значений хеш-функции (2^{64}) делится на 2^{32} чанков. В свою очередь каждый чанк в равных частях делится на количество «живых» узлов кластера.

Теперь детально рассмотрим какой-либо из чанков, например, 1-й, до и после выхода узла 2 из кластера (рис. 2). Размеры чанков всегда остаются неизменными, изменяются только размеры пространств ответственности узлов внутри чанков.

Как видно из рисунка, пространства ответственности узлов 0 и 1 только расширились (пересечение нового пространства с

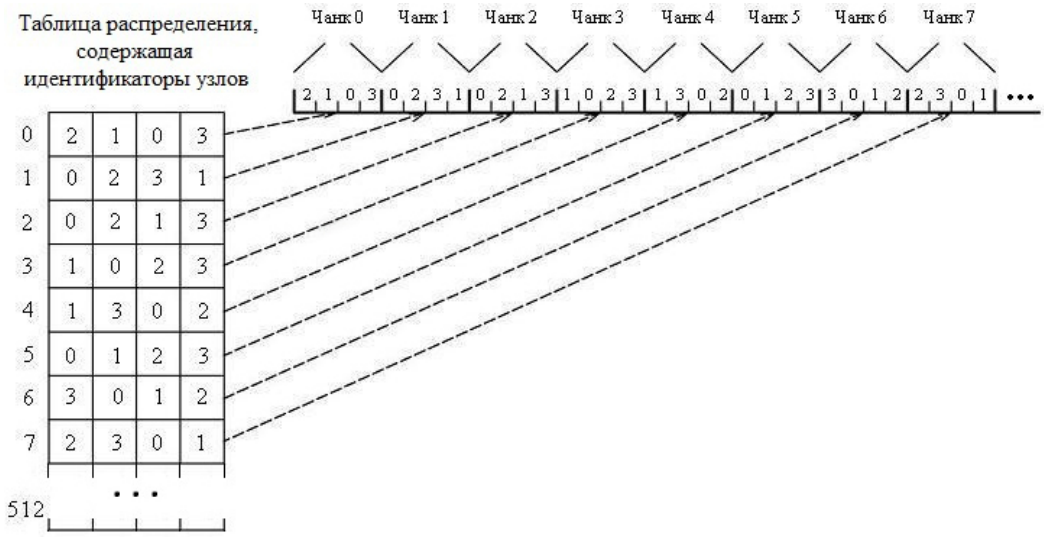

Рис. 1. Отображение таблицы распределения на область значений хеш-функции

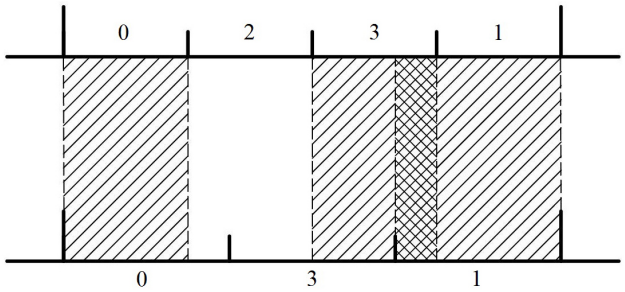

Рис. 2. Ситуация выхода узла 2 из кластера в пределах одного чанка в линейном представлении (верхняя и нижняя горизонтальные линии – хеш-пространство до и после выхода соответственно)

прежним для каждого узла показано обычной штриховкой). Для этих двух узлов прежнее пространство полностью включается в новое, а это значит, что домены, ключи которых попадают в эти пространства, не перешли в области ответственности других узлов. Следовательно, узлы 0 и 1 полностью сохранили актуальность своих кэшей.

Однако для узла 3 произошло смещение пространства ответственности. Область, помеченная перекрестной штриховкой, мигрировала из области ответственности узла 3 в область ответственности узла 1. Иначе говоря, кэш узла 3 стал частично неактуален из-за перехода к узлу 1 соответственной части доменов, обслуживаемых узлом 3 до выхода узла 2 из состава кластера. Т.е. требование отсутствия перераспределения запросов, обслуживаемых оставшимися узлами до и после изменения состава кластера, частично не выполняется.

На практике, т.к. область значений хеш-функции достаточно большая, такие смещения вызывают миграцию существенного количества ключей (~ 358000000 ключей в пределах одного чанка в данном конкретном случае, где кластер состоит из 4-х узлов), т.е. довольно большая часть доменов становится незакэшированной.

Для предотвращения описанной ситуации был реализован подход, основанный на применении двухуровневой схемы таблиц распределения.

РЕАЛИЗАЦИЯ

Суть двухуровневой схемы организации таблиц распределения состоит в применении двух таблиц распределения вместо одной. При этом одна из таблиц (2-й уровень) фактически представляет собой ту же самую таблицу, которая применялась в одноуровневой схеме, она содержит только «живые» узлы кластера.

Ключевым моментом здесь является введение еще одной таблицы распределения (1-й уровень), содержащей как «живые» узлы кластера, так и те, которые помечены как «мертвые».

Здесь стоит напомнить, что наряду с таблицами распределения в системе существует базовая таблица всех узлов кластера с по-

меткой «живой» или «мертвый», на ее основе строятся таблицы распределения. При этом существуют следующие сценарии изменения данной таблицы:

• появление нового узла в системе: узел добавляется в таблицу и помечается как «живой»;

• временный выход узла из кластера: соответствующий узел не удаляется из базовой таблицы, а лишь помечается как «мертвый»;

• вход узла обратно в кластер после временного выхода: соответствующий узел помечается в базовой таблице как «живой».

• окончательный выход узла: соответствующий узел удаляется из базовой таблицы.

При любом из этих событий сначала изменяется базовая таблица, после чего на ее основе строятся таблицы распределения.

Процедура построения таблиц распределения теперь выглядит следующим образом. Как и в прошлой реализации, обе таблицы содержат 512 вариантов распределения. При построении варианта распределения для каждого узла (независимо от того, является он в данный момент «живым» или «мертвым») рассчитывается хеш-значение на основе комбинации его IP-адреса и номера варианта, после чего полученные значения сортируются в порядке возрастания. В случае возникновения коллизий, т.е. совпадения хеш-значений (что маловероятно, однако такой вариант также необходимо предусмотреть), происходит сравнение значений IP-адресов узлов в численном представлении, при этом используется сетевой порядок байт (network byte order). Полученный после сортировки порядок следования хеш-значений определяет позиции соответствующих узлов в варианте распределения. При этом в таблицу распределения 1-го уровня записываются все узлы (включая «мертвые»), а в таблицу 2-го уровня – только «живые». Таким образом, обе таблицы строятся параллельно и имеют одинаковый порядок следования узлов в соответствующих вариантах распределения за исключением того, что в таблице 2-го уровня пропущены узлы, помеченные как «мертвые».

Графически отображение обеих таблиц распределения на одну и ту же область значений хеш-функции представлено на рис. 3 (где все узлы являются «живыми») и рис. 4 (где узел 2 становится «мертвым»). В обоих случаях отображение таблицы 1-го уровня располагается над отображением таблицы 2-го уровня.

Как видно на рис. 3, обе таблицы имеют абсолютно одинаковое отображение. Однако на рис. 4, где узел 2 стал «мертвым», произошло смещение пространств ответственности

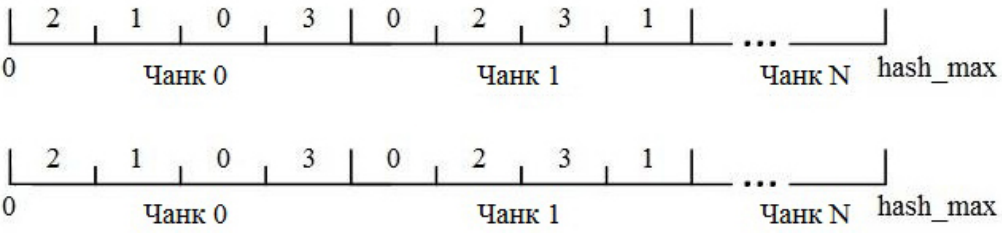

Рис. 3. Отображение таблиц распределения на область значений хеш-функции (отображение таблицы 1-го уровня располагается над отображением таблицы 2-го уровня) до временного выхода узла 2 из состава кластера

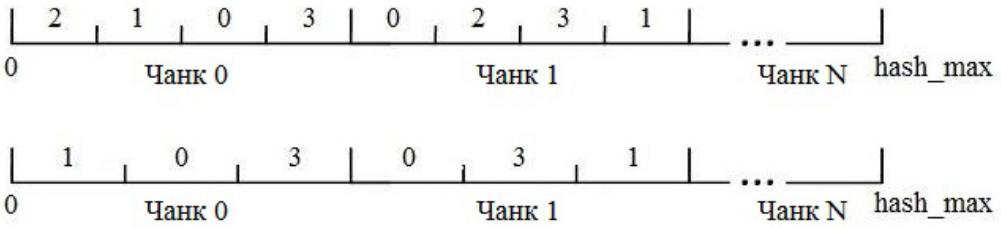

Рис. 4. Отображение таблиц распределения на область значений хеш-функции (отображение таблицы 1-го уровня располагается над отображением таблицы 2-го уровня), узел 2 временно вышел из состава кластера

узлов для таблицы 2-го уровня (аналогичное тому, которое мы рассматривали при использовании одноуровневой схемы), но для таблицы 1-го уровня смещение не произошло за счет содержания «мертвого» узла.

Теперь рассмотрим механизм принятия решения при использовании двухуровневой схемы.

Как и в случае одноуровневой схемы, при поступлении домена на обработку сначала от него рассчитывается хеш-значение. Далее на первом шаге система пытается определить ответственный узел, используя таблицу 1-го уровня, которая может содержать «мертвые» узлы. При попадании хеш-значения в область ответственности «живого» узла данный узел выбирается ответственным, и на этом алгоритм прекращает работу без использования таблицы 2-го уровня. Однако, если хеш-значение попало в область ответственности «мертвого» узла, система на втором шаге принимает решение, используя таблицу 2-го уровня, где хеш-значение гарантированно попадает в область ответственности «живого» узла.

Таким образом, алгоритм принятия решения состоит из одного или двух этапов:

1) попытка определения ответственного узла с использованием таблицы 1-го уровня;

2) определение ответственного узла с использованием таблицы 2-го уровня, если на первом этапе ответственный узел оказался «мертвым».

Представим рассмотренный алгоритм графически на примере чанка 1 до и после временного выхода узла 2 из состава кластера (рис. 5 и рис. 6 соответственно).

Предположим, что ключ 1 и ключ 2 получены от двух различных доменов. На рис. 5 все узлы являются «живыми» и в обоих случаях решение принимается только на основании таблицы 1-го уровня: для ключа 1 ответственным является узел 2, для ключа 2 соответственно узел 3. Отметим, что при полном составе участников кластера решение всегда принимается с помощью таблицы 1-го уровня, т.к. таблицы 1-го и 2-го уровней в этом случае полностью идентичны.

На рис. 6 узел 2 временно вышел из состава кластера (его область ответственности помечена штриховкой). В этом случае ключ 1 на первом этапе попадает в область ответственности вышедшего узла 2, после чего «проваливается» через отображение таблицы 1-го уровня на отображение таблицы 2-го уровня, где попадает в область ответственности «живого» узла 0. Таким образом, теперь ответственным узлом для ключа 1 стал узел 0, для ключа 2 ответственный узел не изменился, что видно на рис. 6.

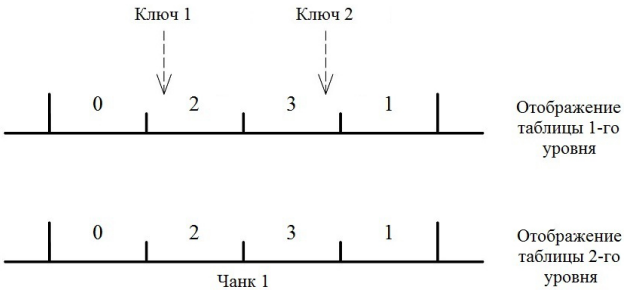

Рис. 5. Определение ответственных узлов для двух различных доменов до временного выхода узла 2 из состава кластера

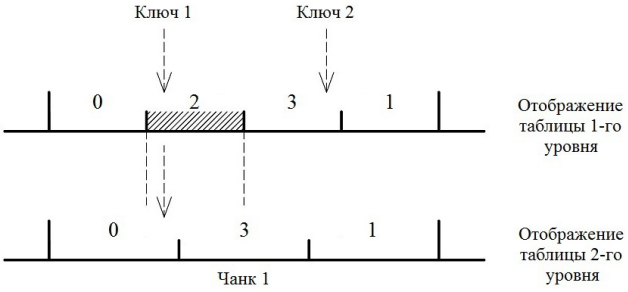

Рис. 6. Определение ответственных узлов для двух различных доменов после временного выхода узла 2 из состава кластера

Здесь именно ключ 2 достоин особого внимания. Рис. 6 показывает, что для таблицы 2-го уровня произошло смещение пространств ответственности узлов, и при отсутствии таблицы 1-го уровня ответственным узлом для ключа 2 после выхода узла 2 стал бы узел 1, несмотря на то, что узел 3 не покинул состав кластера. В этом состоит недостаток одноуровневой схемы, где таблица 1-го уровня отсутствует, и решение принимается только с помощью таблицы 2-го уровня. Применение таблицы 1-го уровня, содержащей и «мертвые», и «живые» узлы, позволяет полностью избежать миграцию ключей за счет отсутствия смещения пространств ответственности узлов. Именно в этом и есть суть применения двухуровневой схемы.

На рис. 6 видно, что пространство ответственности узла 2 после его выхода перераспределяется между узлами 0 и 3, причем в неравных пропорциях. Из этого можно предположить, что перераспределение неравномерное, узел 0 принял большую нагрузку, чем узел 3. Однако это происходит только внутри одного конкретного чанка. Мы имеем 2^{32} чанков и 512 вариантов распределения с различным порядком следования узлов, что нивелирует локальную неравномерность перераспределения пространств ответственности. Данных факт подтвержден на практике.

Стоит заметить, что критическим здесь является предположение о том, что выход узла из кластера всегда является временным (по причине технического обслуживания, обновления программного обеспечения, различного рода сетевых проблем, сбоя в работе сервера и т.д.). Существует возможность и окончательного выхода узла из кластера с его удалением из базовой таблицы, однако после этого системе необходимо некоторое время на стабилизацию, т.к. происходит неизбежное смещение пространств ответственности узлов, чего лишен временный выход узла. Практика показывает, что окончательный выход узлов из состава кластера является крайне редким событием, в подавляющем большинстве случаев состав кластера является стабильным.

ЗАКЛЮЧЕНИЕ

Применение двухуровневой схемы организации таблиц распределения позволило привести систему распределения нагрузки в полное соответствие предъявляемым к ней требованиям. Отсутствие эффекта миграции ключей при временных выходах узлов позволяет полностью сохранить актуальность кэшей остающихся участников кластера.

При этом система обладает всеми достоинствами одноуровневой схемы:

- высокие показатели равномерности распределения нагрузки;
- масштабируемость системы;
- быстродействие (непосредственно вычисление ответственного узла составляет несколько арифметических операций и условных переходов);
- надежность;
- гибкость поведения при изменении состава участников кластера.

Высокие показатели быстродействия достигнуты за счет того, что все накладные расходы перенесены в процесс подготовки необходимой инфраструктуры, который инициируется в случае изменения состава участников кластера, что является достаточно редким событием по отношению к частоте прихода запросов.

По поводу сравнения данного решения с существующими аналогами стоит сказать следующее. Ведущие мировые DNS провайдеры, такие как OpenDNS и Google DNS, заявляют об использовании кластеризации своих DNS сервисов, однако разработки в этой области составляют в данный момент «ноу-хау» крупных IT-корпораций. Поэтому, во избежание потери актуальности подобных решений, сведения о них отсутствуют в открытой печати, что делает невозможным сравнение с ними рассмотренной системы.

Система распределенного хранения Dynamo от Amazon построена на аналогичных принципах [2]. Во многом именно она повлияла на выбор базисных принципов, лежащих в основе рассмотренного решения. Однако эти две системы создавались с учетом особенностей их областей применения, вследствие чего они имеют различные требования, несмотря на схожие основы.

На данный момент мы можем оценивать эффективность нашего решения только относительно собственных требований и критериев качества, таких как показатели равномерности распределения, быстродействие и масштабируемость с учетом особенностей DNS. Лабораторное тестирование и функционирование в составе DNS кластеров, расположенных в Европе и США, показывает полное соответствие системы распределения нагрузки предъявляемым требованиям.

СПИСОК ЛИТЕРАТУРЫ

1. *Шилов С. Н.* Реализация инфраструктуры распределенной хеш-таблицы в рамках кластерной системы DNS / С.Н. Шилов, С.Д. Кургалин, А.А. Крыловецкий // Вестник ВГУ. Серия: Системный анализ и информационные технологии, № 2, 2012. – Воронеж : изд-во Воронежского гос. ун-та, 2012. – С. 74–79.

2. *DeCandia G.* Dynamo: Amazon's Highly Available Key-value Store / G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels // In Proceedings of the 21st ACM Symposium on Operating Systems Principles (Skamania Lodge Stevenson, WA, USA, October 14-17, 2007). ACM Press, New York. 205–218 p.

3. *Fox A.* Cluster-based scalable network services / A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, P. Gauthier // In Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles (Saint Malo, France, October 05 - 08, 1997). ACM Press, New York, 78–91 p.

Шилов С. Н. – аспирант, Воронежский государственный университет, E-mail: shilov_sn88@mail.ru

Кургалин Сергей Дмитриевич – д.ф.-м.н., заведующий кафедрой цифровых технологий факультета компьютерных наук, Воронежский государственный университет, e-mail: kurgalin@bk.ru Тел.: (473) 2-208-384

Крыловецкий А.А. – к.ф.-м.н., доцент, Воронежский государственный университет, aakryl@cs.vsu.ru

Shilov S. – postgraduate student, of the Digital Technologies Department of Computer Science Faculty, Voronezh State University. E-mail: shilov_sn88@mail.ru

Kurgalin S. D. – Doctor of Physical and Mathematical Science, Head of the Digital Technologies Department of Computer Science Faculty; tel.: (473) 2-208-384, e-mail: kurgalin@bk.ru

Krylovetsky A.A. – candidat of Physical and Mathematical Science of the Digital Technologies Department of Computer Science Faculty, Voronezh State University. E-mail: aakryl@cs.vsu.ru