УДК 004.421.2 DOI: 10.26102/2310-6018/2019.26.3.017

В.А. Музыченко

ОРГАНИЗАЦИЯ ИНДЕКСА РАСПРЕДЕЛЕННОЙ ПОИСКОВОЙ СИСТЕМЫ, РАБОТАЮЩЕЙ ПО АЛГОРИТМУ КОНСЕНСУСА *BFT

Воронежский государственный университет,
Воронеж, Россия

*В рамках данной статьи рассматривается алгоритм построения поискового индекса распределенной поисковой системы, применяющей алгоритм консенсуса семейства *BFT (Byzantine Fault Tolerance), а также его реорганизация, вызванной добавлением или выходом узлов из состава поискового кластера. В статье детально описывается структура кластера, добавление данных в индекс, алгоритм реорганизации, а также рассматриваются возможные сопутствующие проблемы, описываются их решения. Вместе с тем рассматриваются ограничения, накладываемые как самой природой распределенных вычислений, так и необходимостью соответствию требованиям BFT. Актуальность задачи обуславливается возрастающей необходимостью применения распределенных систем для решения разнородных практических задач, в том числе и организации поиска, а также необходимостью адаптации существующих решений к условиям распределенных систем и учета накладываемых ими ограничений, что подтверждается как существованием активных исследований в данной области, так и сложившейся на рынке ситуацией. Методы и подходы, применяемые в данной работе, с некоторой модификацией, могут быть перенесены на схожие задачи, возникающие в распределенных системах других типов. Результаты и материалы статьи могут быть использованы для решения прикладных задач при реализации распределенных вычислительных систем, распределенных поисковых систем, распределенных систем хранения данных.*

Ключевые слова: поиск, распределенные системы, алгоритмы консенсуса, BFT, балансировка нагрузки

1. Введение

Поисковые системы стали неотъемлемой частью World Wide Web практически с момента его появления, сформировав Web таким, каким мы его знаем сейчас. Однако, даже будучи крайне важной составляющей, они не лишены недостатков, которые ограничивают их применение, и, как следствие, развитие Web.

Существует множество исследований, посвященных совершенствованию как поисковых систем, так и Web. Но несмотря на то, что разрабатываемые системы, способные заменить его, такие как IPFS [1], являются достаточно перспективными, в обозримом будущем вытеснить они его не смогут. Поэтому задача поиска остается актуальной.

С другой стороны, развитие получили подходы к организации распределенных систем, упрощающие их создание, за счет использования

менее сложных к реализации протоколов распределенного консенсуса [2], а также исключающие необходимость доверия между участниками [3]. Такие системы обладают важной особенностью – ориентацией на работу в условиях отсутствия гарантий корректности и непротиворечивости действий между отдельными узлами системы. В них информация, полученная лишь от одного участника, не может считаться достоверной [3]. При выполнении каких-либо операций узлы проводят проверку её правильности и затем голосование за её принятие или отклонение. Ярким примером такого алгоритма консенсуса является алгоритм PBFT [3].

О реализации поискового индекса в поисковых системах такого рода и идет речь в данной статье.

2. Материалы и методы

2.1 Устройство индекса

Для начала необходимо определиться с тем, что такое поисковый индекс. Поисковый индекс – некоторая структура данных, организованно хранящая информацию о данных и позволяющая эффективно выполнять по ним поиск. Стоит отметить, что помимо непосредственно поиска такие системы также выполняют ранжирование результатов, чтобы представить их в порядке, соответствующем их релевантности.

Для решения этих задач о каждой странице необходимо знать как минимум две вещи:

1. Ключевые слова и другие признаки, которые наиболее точно описывают данную страницу.
2. Связи между страницами. Страницы в Web связаны друг с другом гиперссылками, и информация об этих связях является крайне полезной.

В то время как наборы признаков могут использоваться как для поиска, так и для ранжирования результатов, качество ранжирования, основанного лишь на них, как правило, невысокое. Основная причина этого заключается в том, что страницы, имеющие большее число совпадений, по ключевым словам, будут подниматься выше, даже если фактически не несут никакой полезной для человека информации. Приемы, ориентированные на это, часто применяются с целью продвижения сайтов путем повышения их “привлекательности” с точки зрения поисковых систем [4].

По этой причине для повышения качества оценки релевантности применяются алгоритмы, которые учитывают также количество прямых или косвенных ссылок между статьями. Логика заключается в следующем – чем больше страниц, имеющих высокую оценку, ссылаются на данную, тем выше оценивается, и она сама. Наиболее известным и часто применяемым алгоритмом ранжирования, учитывающим связи между страницами, является алгоритм PageRank. Алгоритм PageRank был впервые описан

компанией Google и после публикации стал активно применяться во многих проектах, в том числе не связанных с Web.

Релевантность статьи в PageRank рассчитывается следующим образом:

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u), \quad (1)$$

где

R'(u) – ранг страницы u;

R(v) – ранг некоторой страницы v, ссылающейся на страницу A;

Nv – количество ссылок страницы v на другие страницы;

E(u) – вектор-параметр, предназначенный для статей и групп статей, не имеющих общих связей с другими

c – нормализационный параметр, ограничивающий рост ранга.

Как можно заметить, некоторые страницы могут прямо или косвенно ссылаться друг на друга. В результате вычисление ранга конкретной страницы зависит от вычисления ранга другой и, в конце концов, от самой себя (рисунок 1). По этой причине вычисление, как правило, производят итеративно, на каждом этапе получая приближенное значение все более высокой точности.

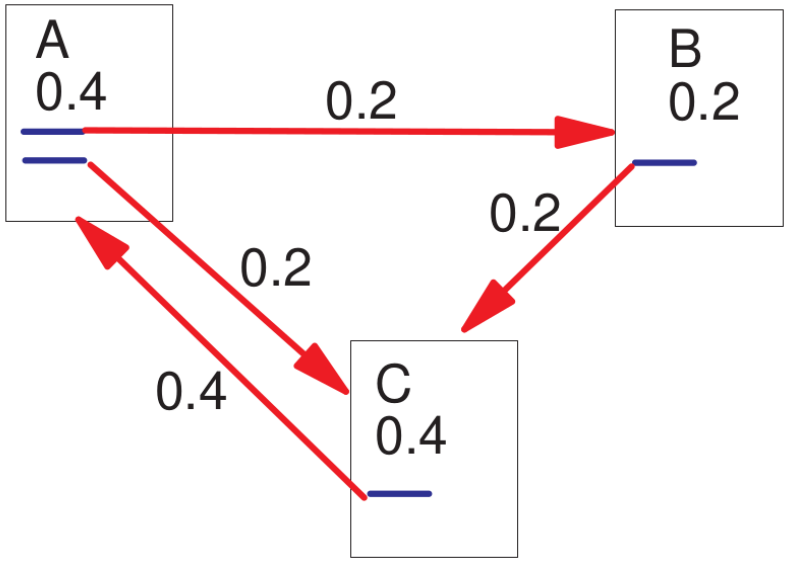

Рисунок 1 – Пример упрощенного вычисления PageRank [4]

В случае необходимости применения данного алгоритма в распределенной поисковой системе, работающей по *BFT протоколу, возникают некоторые проблемы, накладывающие ограничения на используемые методы и подходы:

1. Разные узлы могут иметь разный набор данных.
2. Из-за особенностей алгоритма консенсуса, исключающего доверие между узлами, для принятия результатов каждого действия,

выполненного одним узлом, необходимо предоставить, в том или ином виде, доказательство другим узлам или клиентам.

3. Сам факт наличия консенсуса ограничивает производительность.

Для адаптации алгоритма PageRank к BFT консенсусу в данной системе реализуется разделение ответственности между узлами.

Нагрузка разделяется между разными подкластерами (рисунок 2), отвечающими за разные наборы страницы, в зависимости от идентификатора, выданного странице. Эти подкластеры связаны друг с другом через узлы-лидеры, выступающие связующими звеньями. Такая структура позволяет при необходимости масштабировать систему, увеличивая количество подкластеров и сужая их область ответственности.

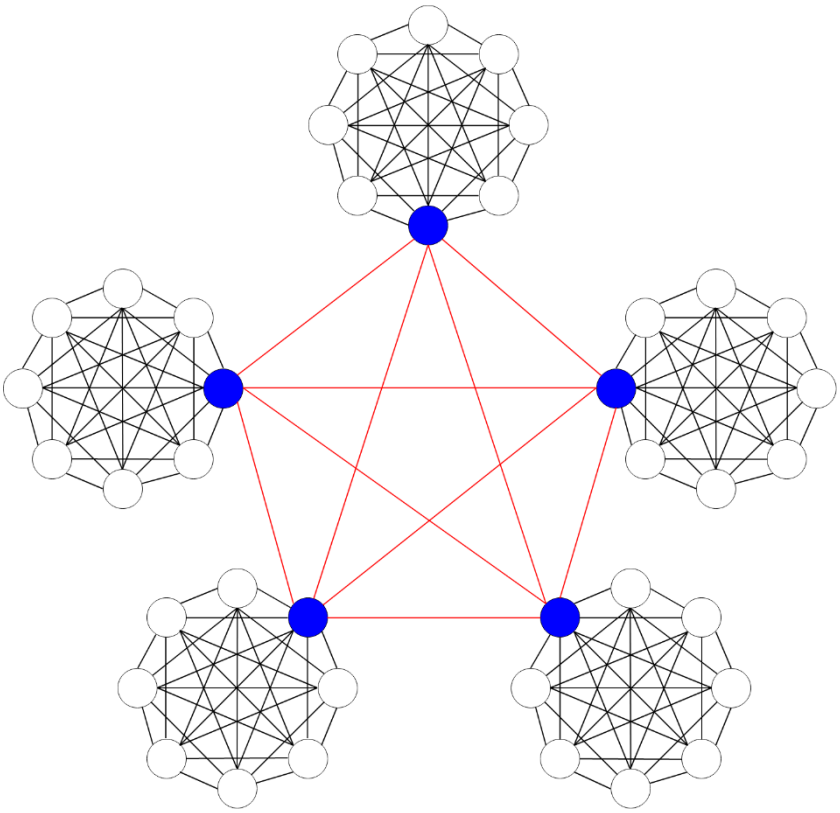

Рисунок 2 – Пример структуры поисковой системы

Рассмотрим алгоритм добавления статьи в индекс и расчета её оценки PageRank:

1. Группа узлов-лидеров берет ссылку на страницу в Web и вычисляет её идентификатор.
2. В зависимости от идентификатора ссылка передается на дальнейшее рассмотрение в соответствующий подкластер. Данный этап может повторяться и далее на более низких уровнях иерархии, пока не остановится на подкластере, отвечающем за группу статей, в которую попадает данная.

3. Узлы конечного подкластера, каждый в отдельности, обращаются по ссылке и получают контент страницы.
4. Узлы извлекают необходимые признаки (например, ключевые слова) и ссылки на другие web-страницы.
5. Затем узлы выполняют вычисление ранга статьи по PageRank. Для этого они выполняют запрос вверх по иерархии, через узлы-лидеры, в другие подкластеры с целью найти все страницы, ссылающиеся на данную.
6. Каждый узел-лидер, получив запрос, перенаправляет его на все известные ему подкластеры, а также вверх по иерархии.
7. Каждый конечный подкластер, получив запрос, делает проверку в своей части индекса на наличие статей ссылающихся на целевую и возвращает информацию о них.
8. Получив ответы, узлы проверяют необходимые подписи и выполняют расчет оценки PageRank.

Так как расчет PageRank происходит итеративно, то этапы 5-8 могут выполняться несколько раз с целью повышения точности расчета приближенного значения ранга.

2.2 Реорганизация индекса при изменении состава кластера

При реализации распределенных систем, имеющих состояние, так или иначе возникает необходимость сохранения этого состояния в случае изменения структуры кластера, например, выхода какого-либо узла из строя или же добавления новых узлов. Это относится и к распределенным поисковым системам, которые для выполнения своих задач вынуждены поддерживать поисковый индекс.

Существует два варианта хранения индекса в таких системах.

1. Репликация – каждый узел хранит всю информацию.
2. Шардирование – каждый из узлов хранит только часть информации.

В случае репликации, так как каждый из узлов имеет все необходимые данные, то при добавлении новых узлов достаточно получить копию от одного из рабочих узлов или, если это поможет ускорить процесс, нескольких, а в случае ухода никаких действий принимать и вовсе не нужно. В случае же шардирования список узлов, от которых можно получить данные, сужается, что усложняет как добавление, так и удаление узлов кластера. Рассмотрим отдельно добавление и удаление узлов.

Если какой-либо узел покидает кластер, необходимо убедиться, что данные, хранящиеся на нем, будут перенесены на другие узлы, и функциональность системы не пострадает. Стоит учитывать, что есть два возможных варианта выхода узла:

1. Запланированный выход – выход узла происходит в штатном режиме, и он может способствовать реорганизации, уведомить другие узлы о своем выходе и распределить данные среди них.
2. Экстренный выход – выход узла в результате наступления каких-то событий, препятствующих его дальнейшей нормальной работе.

В реальных системах вероятность наступления второго случая довольно высокая и может быть вызвана:

- проблемами в сети передачи данных;
- перебоями питания;
- проблемами с оборудованием на узле;
- проблемами с программным обеспечением на узле;
- атакой на узел, приводящей к его отказу.

Таким образом, реализация обработки нештатного выхода узла из строя имеет большую важность, так как это может привести к полному отказу системы и потере данных.

Для решения задачи реорганизации индекса при изменении состава кластера данная система опирается на свою структуру (рисунок 2), в которой данные равномерно распределяются по узлам.

Стоит также отметить, что коэффициент репликации всегда должен поддерживаться на высоком уровне, то есть данные никогда не должны храниться на небольшом числе узлов. Ниже будет показано, что необходимое количество узлов должно быть выбрано в соответствии с особенностями алгоритма консенсуса.

Рассмотрим, как происходит добавление узла:

1. Новый узел связывается с кластером и производит обмен ключами.
2. Узел добавляется в кластер как “новый” и получает идентификатор, необходимый для определения части поискового индекса, за которую он отвечает.
3. Узел связывается с другими узлами, отвечающими за те же данные, и делает запрос на синхронизацию.
4. Узел получает необходимые данные и помечается как “готовый”.
5. Узел входит в работу и становится “активным”.

Также рассмотрим процесс реорганизации в случае выхода узла из кластера:

1. Узел по какой-либо причине теряет соединение с кластером.
2. Оставшиеся узлы замечают это и ждут некоторое заданное время его возвращения. Такая задержка необходима, чтобы избежать выполнения реорганизации в случае кратковременных ошибок сети.

3. По истечению времени на возвращение активные узлы кластера запускают процесс реорганизации и вычисляют части индекса, имеющие недостаточное количество узлов-носителей.
4. Если такие части не найдены, процесс завершается, в противном случае части индекса перераспределяют, чтобы восполнить пониженный уровень репликации.

В случае, если выход узла из состава кластера происходит штатно, этап 2 не требуется, так как узел имеет возможность сообщить остальным о своих намерениях, тем самым экономится время и ресурсы.

Стоит отметить, что из-за особенностей применяемого протокола консенсуса минимальное количество узлов в составе кластера не может быть ниже, чем 2f + 1, где f – расчетное количество злонамеренных узлов, входящих в состав системы, с которыми она должна по-прежнему нормально функционировать [2].

3. Результаты

В данной статье рассмотрена задача организации поискового индекса в распределенной поисковой системе, использующей протокол консенсуса семейства *BFT.

Были проведены следующие действия:

1. Предложена структура системы, позволяющей строить поисковый индекс большого размера в распределенной системе на основе PBFT-подобного алгоритма консенсуса.
2. Описано применение алгоритма PageRank в распределенной среде.
3. Рассмотрены возможные проблемы при изменении состава кластера.
4. Описан алгоритм поведения кластера при выходе узлов из состава кластера и добавлении новых узлов.

4. Обсуждение

В контексте данной статьи рассматривается возможность использования распределенных систем, основанных на применении алгоритмов консенсуса семейства *BFT для целей создания поисковых систем. Однако алгоритмы семейства *BFT не являются единственным вариантом для обеспечения устойчивости к BFT ошибкам. Другим вариантом являются системы с распределенным консенсусом на основе алгоритма доказательства проделанной работы (Proof-of-Work) и им подобные [5, 6].

Главный плюс таких систем – возможность включения произвольно большого количества участников, так как для достижения консенсуса не требуется выполнять коммуникацию между всеми узлами сети. Основными недостатками, в свою очередь, является, с одной стороны, высокое потребление вычислительных ресурсов для выполнения сравнительно

небольшой работы, а с другой – низкая пропускная способность сети. И попытки реализации поисковых систем на их основе тоже существуют [7].

Системы же, построенные на основе BFT, имеют значительно меньший размер, чем системы на основе алгоритма Proof-of-work и подобных ему, а также испытывают трудности с добавлением новых узлов, однако их пропускная способность, обычно, значительно выше, а затраты ресурсов не такие высокие, что обеспечивает их конкурентоспособность.

5. Заключение

Создание распределенной поисковой системы, работающей по *BFT протоколу консенсуса, является сложной задачей и связано с необходимостью решения множества проблем, тем не менее, как было показано выше, для многих из них существуют решения.

Растущий спрос со стороны бизнеса и большое количество исследовательских групп не оставляют сомнения в том, что данная область остается актуальной в ближайшие годы, и будущие исследования могут значительно повлиять на развитие поисковых систем и Web в целом.

ЛИТЕРАТУРА

1. Benet J. IPFS – Content Addressed, Versioned, P2P File System / J. Benet. – Версия DRAFT 3 – 2014. – 11с. https://arxiv.org/pdf/1407.3561.pdf
2. Ongaro D., Ousterhout J. In Search of an Understandable Consensus Algorithm / D. Ongaro, J. Ousterhout – Stanford University – 2013. – 18 с.
3. Castro M., Liskov B. Practical Byzantine Fault Tolerance / Miguel Castro, Barbara Liskov – Laboratory for Computer Science, Massachusetts Institute of Technology – 1999. – 14 с.
4. Page L., Brin S., Motwani R., Winograd T. The PageRank Citation Ranking: Bringing Order to the Web / Lawrence Page, Sergey Brin, Rajeev Motwani, Terry – Winograd Stanford InfoLab – 1999. – 17 с. http://ilpubs.stanford.edu:8090/422/
5. Back A. Hashcash - A Denial of Service Counter-Measure / A. Back – 2002. – 10с. http://www.hashcash.org/papers/hashcash.pdf
6. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System – 2008г. – 9 с. https://bitcoin.org/bitcoin.pdf
7. The Community-Powered Search Engine. – 2017г. – 39 с. https://www.presearch.io/uploads/WhitePaper.pdf

V.A. Muzychenko

BUILDING AND MANAGING INDEX OF DISTRIBUTED SEARCH ENGINE BASED ON *BFT CONSENSUS ALGORITHM

Voronezh State University,
Voronezh, Russia

*In this article, we consider an algorithm for building a search index of a distributed search engine that uses the consensus algorithm of the *BFT (Byzantine Fault Tolerance) family, as well as its reorganization caused by the addition or removal of nodes from the search cluster. The article describes the structure of the cluster, addition of data to the index, cluster reorganization algorithm, and also discusses possible related problems and their solutions. At the same time, the limitations imposed by the very nature of distributed computing and the need to comply with BFT requirements are considered. The value of the task is due to the increasing need to use distributed systems to solve different practical tasks, including the organization of search, as well as the need to adapt existing solutions to the conditions of distributed systems and to take into account the limitations imposed by them, as evidenced by the existence of active research in this area and the current in the market situation. The methods and approaches used in this work, with some modifications, can be transferred to similar tasks arising in distributed systems of other types. The results and materials of the article can be used to solve applied problems in the implementation of distributed computing systems, distributed search systems, distributed data storage systems.*

Keywords: search, distributed systems, BFT consensus algorithm, load balancing

REFERENCES

1. Benet J. IPFS - Content Addressed, Versioned, P2P File System / J. Benet – RAFT 3 – 2014. – 11 p.
2. https://arxiv.org/pdf/1407.3561.pdf
3. Ongaro D., Ousterhout J. In Search of an Understandable Consensus Algorithm / D. Ongaro, J. Ousterhout – Stanford University – 2013. – 18 p.
4. Castro M., Liskov B. Practical Byzantine Fault Tolerance / Miguel Castro, Barbara Liskov – Laboratory for Computer Science, Massachusetts Institute of Technology – 1999. – 14 p.
5. Page L., Brin S., Motwani R., Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. / Lawrence Page, Sergey Brin, Rajeev Motwani, Terry – Winograd Stanford InfoLab – 1999. – 17 p. http://ilpubs.stanford.edu:8090/422/
6. Back A. Hashcash - A Denial of Service Counter-Measure / A. Back – 2002. – 10 p. http://www.hashcash.org/papers/hashcash.pdf
7. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. – 2008. – 9 p. https://bitcoin.org/bitcoin.pdf
8. The Community-Powered Search Engine. – 2017. – 39 p. https://www.presearch.io/uploads/WhitePaper.pdf