ИСПОЛЬЗОВАНИЕ ВЕРОЯТНОСТНЫХ СТРУКТУР ДАННЫХ ПРИ РАБОТЕ С БОЛЬШИМИ ОБЪЁМАМИ ДАННЫХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Закревский И.Е.
Гурский А.Л. – д-р физ.-мат. наук., профессор

Операции над большими массивами данных являются неотъемлемой частью ежедневной работы во многих областях информатики. Так как каждая операция над конкретным элементом применяется много раз, даже незначительное сокращение ресурсов в пересчете на одно действие может дать значительную экономию. Также, в виду бурного развития рынка интернета вещей, поднимается вопрос об энергопотреблении устройств и, как следствие, оптимизации их исходного кода. Растут объемы баз данных, что сказывается на скорости работы средств защиты информации, таких как средства аутентификации, DLP системы, анализаторы трафика и т.д. Одним из путей решения таких проблем является использование вероятностных структур данных, в частности — фильтр Блума.

Фильтр Блума — это вероятностная структура данных, позволяющая хранить и проверять принадлежность элемента к множеству[1]. В фильтре Блума возможны ложноположительные срабатывания, то есть структура данных может положительно ответить о наличии элемента, когда на самом деле его нет, в то время как ложноотрицательных срабатываний быть не может. Фильтр Блума не хранит элементы, а только предоставляет информацию об их наличии во множестве.

Фильтр Блума представляет собой битовый массив из m бит, которые по умолчанию обнулены. Далее, пользователю необходимо определить k независимых хеш-функций, которые будут преобразовывать массив входных данных произвольной длины в битовую строку фиксированной длины m достаточно равномерным способом. Процент ложноположительных срабатываний может быть уменьшен увеличением размера массива m и/или числа хеш-функций k [2].

Чтобы добавить новый элемент в фильтр Блума, необходимо пропустить его через k хеш-функций, которые вернут k номеров позиций в массиве, подлежащих установлению в 1.

Для проверки наличия элемента во множестве надо пропустить его через k хеш-функций, которые вернут k позиций массива. Если любой из битов в этих позициях равен 0 – данный элемент точно отсутствует в множестве, т.к. все биты должны были быть установлены в 1 во время вставки. Если все биты равны 1 – то либо элемент находится во множестве, либо значение 1 было установлено в результате коллизии хеш-функций и это приведёт к ложноположительному срабатыванию.

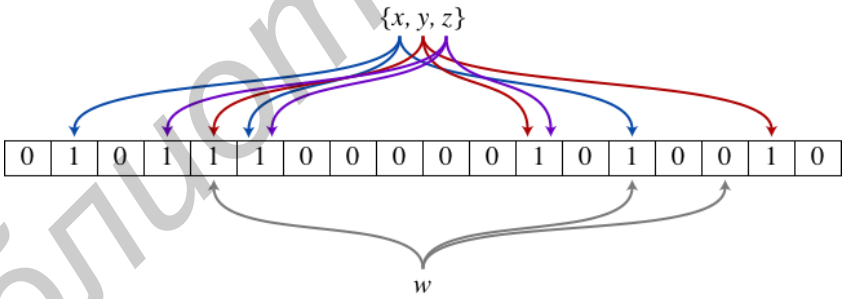

Рис. 1 - Пример фильтра Блума. Элементы *x, y, z* являются членами множества, *w* — нет

Если количество потенциальных значений невелико и многие из них могут быть уже во множестве, фильтр Блума легко превосходит детерминированный битовый массив, который требует только один бит для каждого потенциального элемента.

Дополнительным преимуществом фильтра Блума является тот факт, что время добавления и проверки наличия элемента в множестве постоянно и является $\theta(k)$ и не зависит от количества элементов в множестве. В виду того, что запросы к структуре независимы, они могут быть легко распараллелены.

В данной работе был реализован фильтр Блума (k = 3, m = 10^9) и использован для определения необходимости вызова удалённой БД. Использование фильтра Блума в качестве структуры данных для хранения информации о наличии элемента в БД позволило сократить на 77% используемую память по сравнению с хеш-таблицами, также незначительно уменьшило время на добавление состояния новых элементов в множество (>5%).

Список использованных источников:
1. Bloom, Burton H., Space/time trade-offs in hash coding with allowable errors,
2. Dillinger, Peter C.; Manolios, Panagiotis, "Fast and Accurate Bitstate Verification for SPIN"