Filtros Bloom

Son una estructura de datos porbabilística y optimizada. Se usan para encontrar si un objeto pertenece o no a un dataset. Optimiza este tipo de peticiones usando funciones hash en los elementos a procesar. Cuando el resultado de una petición es positivo, entonces el objeto posiblemente pertenezca al dataset en cuestión, de todas formas pueden ocurrir falsos positivos. Cuando el resultado es negativo, entonces el objeto no pertenece al dataset, no hay falsos negativos. Esta pensado para volúmenes de datos a gran escala.

Un filtro bloom puede ser definido como una tabla o array compuesta por m bits. Inicialmente todos los bits están inicializados a 0. Para añadir un elemento x a la tabla, se usan funciones hash k para encontrar su posición en la tabla y se establecen dichos bits a 1. En un filtro bloom clásico no se pueden eliminar items.

Parametrización de los filtros de bloom

La probabilidad de falsos positivos para un elemento que no pertenece al set es:

$$ε = (1-(1-\frac{1}{m})^{nk})^k ≈ (1 - e^{-kn/m})^k$$

Por lo tanto, el numero de funciones hash óptimo es:

$$k = \frac{m}{n}ln2$$

Y el tamaño del flitro de bloom puede ser determinado como:

$$m = - \frac{n ln ε}{(ln 2)^2}$$

n es el número de objetos almacenados dentro del filtro de bloom.

Propiedades de los Filtros de Bloom

Podemos estimar el número de elementos en un filtro de bloom F como:

$$|F| ≈ - \frac{m}{l}ln(1-\frac{∑^m_{i=1}F_i}{m})$$

La unión de 2 filtros de bloom A y B puede ser computada aplicando una operación OR:

$$|A ∪ B| ≈ -\frac{m}{k}ln(1 - \frac{∑^m_{i=1}(A ∪ B)_i}{m})$$

La intersección de 2 filtros de bloom A y B puede ser computada aplicando una operación AND:

$$|A ∩ B| = |A| + |B| - |A ∪ B|$$

Consideraciones sobre los filtros de Bloom

- No son una estructura que almacena datos por sí misma, pero puede ser usada como un mecanismo de optimización para mejorar el rendimiento de muchas aplicaciones.
- La tasa de falsos positivos debe ser medida y monitorizada. El rendimiento de los filtros de bloom se puede desplomar si hay demasiados elementos insertados.

Funciones Hash

En teoría, se deben seleccionar k funciones hash diferentes para implementar en los filtros de bloom. En la práctica las funciones hash son generadas por un esquema de doble hasing:

$$h_i(x) = h_1(x) + i*h_2(x)$$

En este caso, dos funciones hash diferentes son requeridas. Tambiñen es común suar una función hash con valores de entrada divididos en dos partes.

Diccionarios y Descomposición

Muchas palabras pueden ser descompuestas usando múltiples reglas, pero algunas requieren buscar su separación en una tabla de descomposición. Un filtro bloom es usado para almacenar la representación de dicha tabla. Si la palabra no está almacenada en el bloom filter, su descomposición puede ser resuelta usando reglas simples. Si la palabra fue almacenada en el filtro, entonces se ha realizado una búsqueda de tabla. Los falsos positivos significan que la tabla fue buscada cuando no era necesario. Extensiones de esta aplicación pueden ser usadas para sistemas de revisión de pronunciación y diccionarios de contraseñas no válidas y nombres de usuario ya en uso.

Bases de datos

Optimiza operaciones JOIN en sistemas distribuidos. Un nodo A envía un Filtro de Bloom que representa un subset de datos del nodo B. El nodo B devuelve al nodo A pares que han coincidido con el bloom filter. Finalmente, los falsos positivos son eliminados de A. Usando esta aproximación se reducen el consumo de recursos y las comunicaciones.

Proxies de Cacheo distribuidos

Cuando hay un fallo de cache, entonces el proxy intenta adivinar si hay otro proxy que tenga la página web en cache. Intercambiar todo el contenido en cache entre todos los proxies es muy caro. Un filtro de Bloom que representa los contenidos de la cache puede ser usado. Cuando un proxy tiene que encontrar otro que tenga la web en caché, simplemente revisa su Filtro de Bloom. Un falso positivo significa que la petición será hecha al proxy para descubrir que no tiene la página en cache. En esquema, los falsos negativos son posibles ya que la cache pudo haber sido actualizada durante el proceso.

Problemas de seguridad

Los filtros de bloom no son una estructura de datos segura. No pueden ser considerados una medida de seguridad activa. Pueden ser utilizados para intentar preservar la privacidad de los datos que representan hasta cierto punto. Son vulnerables a ciertos tipos de ataques.

- Es fácil manipular y modificar sus contenidos para incrementar los falsos positivos (Ataque de polución)
- El análisis probabilístico usando frecuencias puede resultar en un enlace de registros con bases de datos externas para revelar datos.

El uso de funciones hash criptográficas no reduce el número de falsos positivos. Pueden ser usadas si se desea, pero pueden afectar considerablemente el rendimiento del filtro. Normalmente se usan funciones hash no criptográficas. Si se usa un esquema de doble hasheo basado en slicing, las funciones criptogáficas pueden ser una buena elección.

From:
https://knoppia.net/ - **Knoppia**

Permanent link:
https://knoppia.net/doku.php?id=pan:filtros_bloom_v2&rev=1767874538

Last update: **2026/01/08 12:15**