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.