FECHA: / /

Consistent Hashing

Supongo que deseo tener un índice distribuido sobre n máquinas/servidores $S_1, \ldots, S_n$

Posible solución: Tener una función de hash:

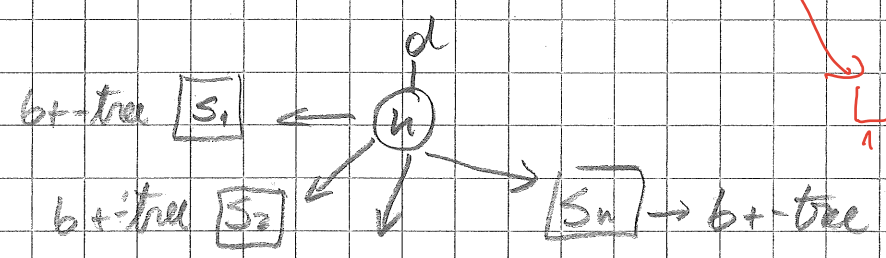
$$h: D \to \{1, \ldots, n\}$$

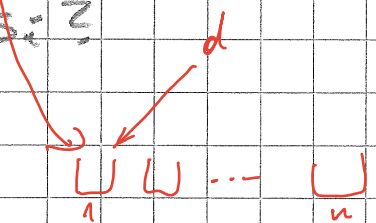
tal que cada servidor S_i reciba el conjunto:

$$\{ d \mid h(d) = i \}$$

Pregunta: - ¿cuál es la carga de cada máquina?

- ¿cómo buscamos/insertamos un dato?
- ¿cómo almacenamos los datos en cada servidor S_i?





$d_1 \ldots d_n$

$\frac{1}{n}$ $X_i :=$ # de datos en serv. i

$X_i \sim B(n, \frac{1}{n})$

$\Rightarrow E(X_i) = \frac{n}{n}$

* ¿es esta solución más eficiente que un b+-tree en un solo servidor?

FECHA: / /

¿Cuál es el problema con esta solución?

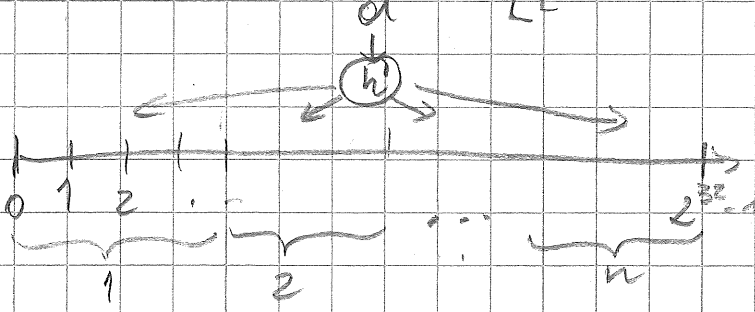
↳ inserción de una máquina $n+1$ requiere hacer re-hashing de todos los datos.

↳ ¿Por qué quisiéramos insertar otra máquina?

Hacer consistent hashing (otra solución)

Supongo que en vez de tener una función de hash $h: D \to \{1, \dots, n\}$, tenemos una función $h': D \to \{0, \dots, 2^{32}-1\}$ (palabras de 32-bits) y asignamos a cada servidor s_i el conjunto

$$\left\{ d \mid h(d) \in \left[\left\lfloor \frac{(2^{32}-1)\cdot(i-1)}{n} \right\rfloor, \left\lfloor \frac{(2^{32}-1)\cdot i}{n} \right\rfloor \right] \right\}$$



FECHA: / /

Preguntas: - ¿Cómo buscar/insertar un dato?

- ¿Sigue siendo "uniforme" la distribución de rango 2_i

- ¿Es posible insertar una máquina sin hacer re-hashing de todos los datos

Consistent Hashing

Solución: Hacer hashing de datos y máquinas

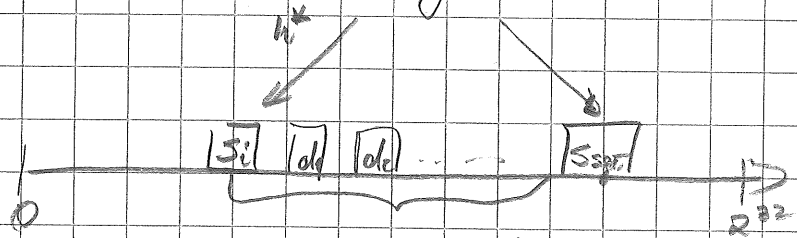
Supongo que se tiene una función de hash:

$$h^*: D \cup N \to \{0, \ldots 2^{32}-1\}$$

y asignamos a cada servidor s_i el conjunto

$$\{d \mid h^*(s_i) \leq h^*(d) < h^*(s_{sgt}) \bmod 2^{32}\}$$

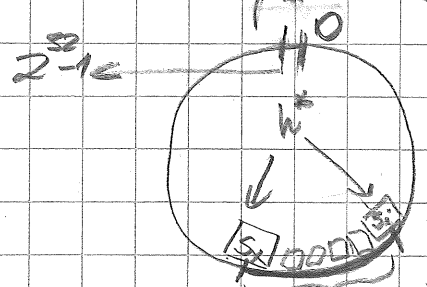
donde $s_{sgt} = \arg\min_s \{(h^*(s) - h^*(s_i)) \bmod (2^{32})\}$



todos estos datos pertenecen a la máquina s_i

FECHA: / /

Esto también se puede ver como un círculo



→ todos estos datos son asignados a s_i

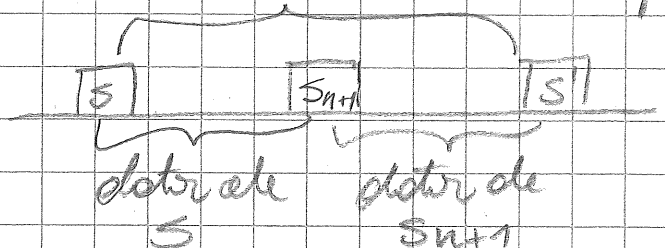
Preguntas: - ¿Cómo busco / inserto un dato?

- Suposición: máquina maestro (o todas las maq.) cuentan con un b+-tree de conjuntos

$$\{(h^*(s_i), s_i)\}$$

ordenado por $h^*(s_i)$.

- ¿Cómo hacemos para insertar una nueva máquina s_{n+1}?



∴ - Sólo s debe mover sus datos de lugar

- Todos los otros servidores siguen igual.

FECHA: / /

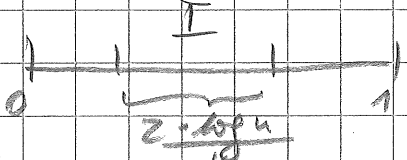
Pregunta: ¿Es uniforme la carga entre cliente mayor?

Teorema: Con prob. $1-\frac{1}{n}$, cada máquina es responsable de a lo más $O(\log n / n)$ fracción de buckets del intervalo $[0, 2^{32}-1]$.

Dem: SPDG supongo que $[0, 2^{32}-1)$ es el intervalo $[0,1)$ y sea:

$$S = \{h(s_i) \mid i \in \{1, \dots, n\}\}$$

Considere el subintervalo I de tam. $\frac{2 \cdot \log n}{n}$



$$P(I \cap S = \emptyset) = \left(1 - \frac{2 \cdot \log n}{n}\right)^n \geq \left(\left(1 - \frac{2 \cdot \log n}{n}\right)^{\frac{n}{2 \log n}}\right)^{2 \cdot \log n}$$

$$\approx e^{-2 \cdot \log n} = \frac{1}{n^2}$$

Subdivido $[0,1)$ $I_1, \dots, I_k$ intervalos iguales de tam $\frac{2 \cdot \log n}{n}$.

$$P\left(\text{alguna maq recibe más de } \frac{4 \cdot \log n}{n} \text{ fracc.}\right) \leq P\left(\bigcup_{i=1}^{k} I_i \cap S = \emptyset\right) \quad \text{y } k = \left\lceil \frac{n}{2 \cdot \log n} \right\rceil$$

$$\leq \sum_{i=1}^{k} P(I_i \cap S = \emptyset) = \sum_{i=1}^{k} \frac{1}{n^2} \leq \frac{1}{n}$$

RHEIN