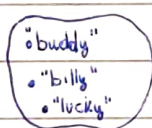
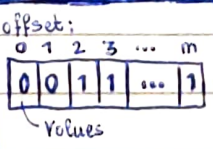
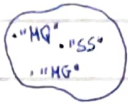


NB: Redis è single-thread, non ci possiamo essere race-condition.
Tutte le operazioni sono atomiche, 1 comando alla volta.

TIPI DI DATO

TIPO	Key (es)	Value (es)	note:
STRINGA	user:1 counter	"MAURO" "1"	Posso anche usarle come contatori. Per esempio > INCR counter → "2" > INCRBYFLOAT counter 2.4 → "3.4"
LIST	users:top	0) "MQ" 1) "SS" 2) "MG"	Posso usarle per implementare uno STACK o uno QUEUE.
HASH	movie	"title" ⇒ "Halo" "year" ⇒ "1972" "rating" ⇒ "4"	Key e Value sono STRINGHE (non ordinate) Anche qui posso usare INCRBY e compagne belle > INCRBY movie "rating" 2 → 6
SET	user:max:dogs		<ul style="list-style-type: none"> Collezione non ordinata di stringhe NB: Quando si fa lo sharding dei dati e si usano operazioni come UNION bisogna fare in modo (key tagging) che gli insiemi siano sullo stesso MASTER!
SORTEDSET	leaders	"MQ": 700 "SS": 101 "MG": 102	È una collezione di stringhe con uno score associato. Ordinato per score e a parità di score è ordinato lessicograficamente.
BITMAPS (*)	mails:2015-04-01	offset: 	Viene rappresentato come stringa. È un array di bit. È il dominio che sa cosa rappresenta un valore di 0/1 alla posizione (offset) n dell'array. ES: offset: ID utente Value: mail inviata 0=F, 1=V

TIPO	Key (es)	Value (es)	note:
HYPERLOGLOG	visits: 2015-01-01	COUNT OF A SET: 	Non è un vero tipo di dato ma un algoritmo super-efficiente per contare gli elementi di un insieme. NB: È accurato al 99,19%

(*) BITMAPS

Benchmark:

È sempre meglio farlo nonostante "pensiamo" che una soluzione sia migliore di un'altra!

Non è sempre vero che BITMAPS è sempre migliore di SET, dipende (per esempio) dalla distribuzione degli ID (offset), se sono tutti valori molto grandi meglio non usare BITMAPS.

SET size: 100 ~~1000000~~

• decId1
• decId2

BITMAP size: 100

