Hello!

I am Esmaeil Kazemi

I'm interested in learning how are you?

You can find me at @eskazemi

NOSQL

vs

SQL

Redis

Redis stands for Remote Dictionary Server

eskazemi

Map

Redis

1- Features

2- use

3- data type

4- Message Queue

5- Transactions

6- Pipelining

7- Lua Scripts

8- Persistence

9- Benchmarks

10- configuration

11- ACLs

12- Redis Cluster

13- Redis vs Memcached

14- Redis vs Hazelcast

ویژگی ها

- از نوع دیتابیس های key - value
- اطلاعات را ذخیره می کند داخل رم (in-memory) که باعث می شود سرعت خواندن و نوشتن اطلاعات افزایش پیدا کند.
- با زبان C نوشته شده است
- از Lua Scripting پشتیبانی می کند
- به صورت پیش فرض replication دارد
- دارای partitioning از طریق Redis Cluster هست
- پورت پیش فرض اون 6379 هست
- به صورت open source و تحت License BSD نگه داری می شود.
- فرآیند نصب ساده ای دارد و مدیریت آن آسان است.
- قابل حمل هست و می توان از آن ها در سیستم های مختلف استفاده کرد.

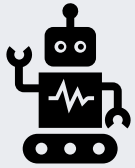
ویژگی ها

- ✓ Redis به سرعت مشهور است چون توانایی اجرای query 100000 در ثانیه را دارد. سرعت خود را با حفظ داده ها در حافظه و همچنین ذخیره آنها بر روی دیسک به دست می آورد.

- ✓ انواع زبان های مختلف چون python , c , c++, Ruby, java و ... می توانند به پایگاه داده redis متصل شوند و از آن استفاده کنند. در صورتی که زبان مورد نظر شما پشتیبانی نمی شود، می توانید کتابخانه client خود را ایجاد کنید زیرا پروتکل Redis ساده است.

- ✓ عملیات Redis بر روی انواع داده های مختلف Atomicity است و اجرای ایمن وظایفی مانند ایجاد کلیدها، اضافه کردن / حذف عناصر مجموعه ، افزایش شمارنده ها و... را تضمین می کند.

موارد استفاده از آن



دیتابیس اصلی

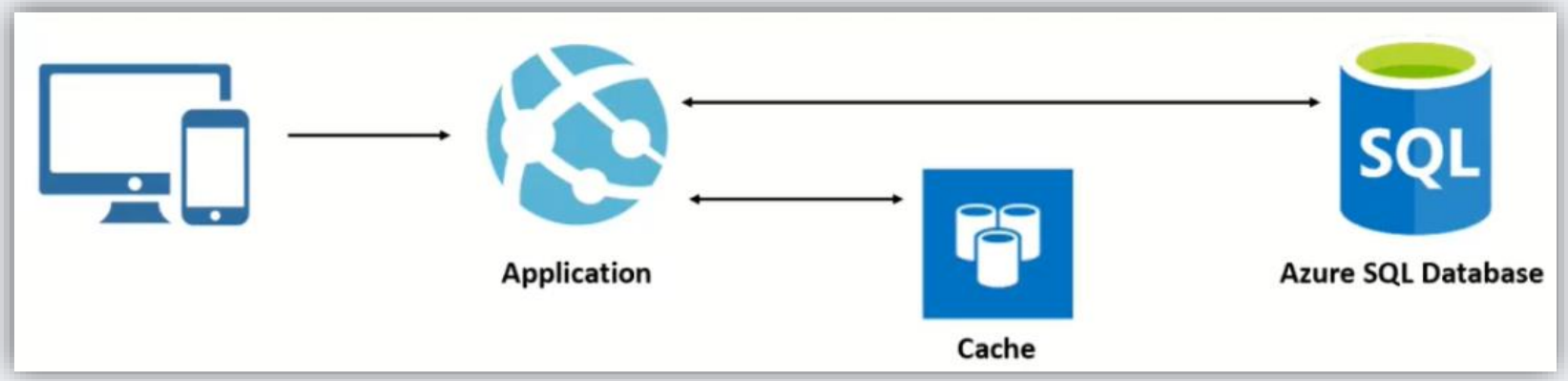
حافظه نهان (cache)

message broker

What is Caching?

حافظه نهان یک نوع حافظه بسیار سریع است که به عنوان یک بافر بین RAM و CPU عمل می کند.

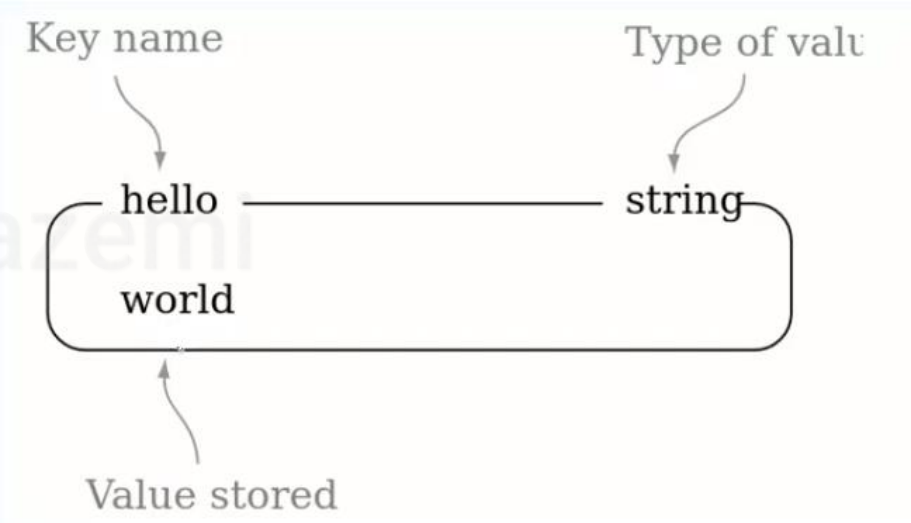
Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed. Cache memory is used to reduce the average time to access data from the Main memory. So data can be served faster by the cache.



توصیه های مختلف به هنگام استفاده از Redis

یکی از ایرادات این نوع پایگاه داده ها عدم وجود ویژگی سازگاری در آنها است. اگر داده‌هایی که می‌خواهیم ذخیره کنیم دارای ارتباطات متفاوتی باشند، پایگاه‌های داده کلید-مقدار عملکرد خود را از دست می‌دهند. این پایگاه های داده معمولا زمانی پیشنهاد می شوند که حجم داده های نوشته شده در پایگاه داده کم و خوانده شده زیاد باشد.

مدل ذخیره سازی اطلاعات در Redis



SET and KEY

SET

برای ایجاد کردن کلید و تعیین مقدار برای آن از دستور SET استفاده می کینم

GET

برای خواندن اطلاعات یک کلید از دستور GET استفاده می کنیم.

DEL

برای حذف یک کلید

EXISTS

برای چک کردن آیا یک کلید وجود دارد یا نه؟

KEYS

برگرداندن کلید ها براساس یک الگو (استفاده از regex)

TTL

مقدار زمان عمر یک کلید بر می گرداند

EXPRIE

برگرداندن کلید ها براساس یک الگو (استفاده از regex)

PERSIST

از بین بردن مقدار زمانی که برای عمر یک کلید در نظر گرفته شده است و ثبات بخشیدن به آن

RENAME

تغییر نام یک کلید

TYPE

برگاندن نوع ساختار داده یک کلید

Features KEYS

- کلیدهای Redis (binary safe)هستند، به این معنی که شما می توانید از هر دنباله باینری به عنوان کلید استفاده کنید، از رشته ای مانند " "foo تا محتوای یک فایل JPEG رشته خالی نیز یک کلید معتبر است.کلید ها بهتره طولانی نباشند.
- اگر کلیدی موجود باشد و دوباره برای کلید مقداری الصاق کنیم کلید بازنویسی می شود و مقدار جدید را می گیرد.

Features KEYS

دو نوع کلید در Redis

Volatile

a key with an expire set

Persistent

a key that will never expire as no timeout is associated

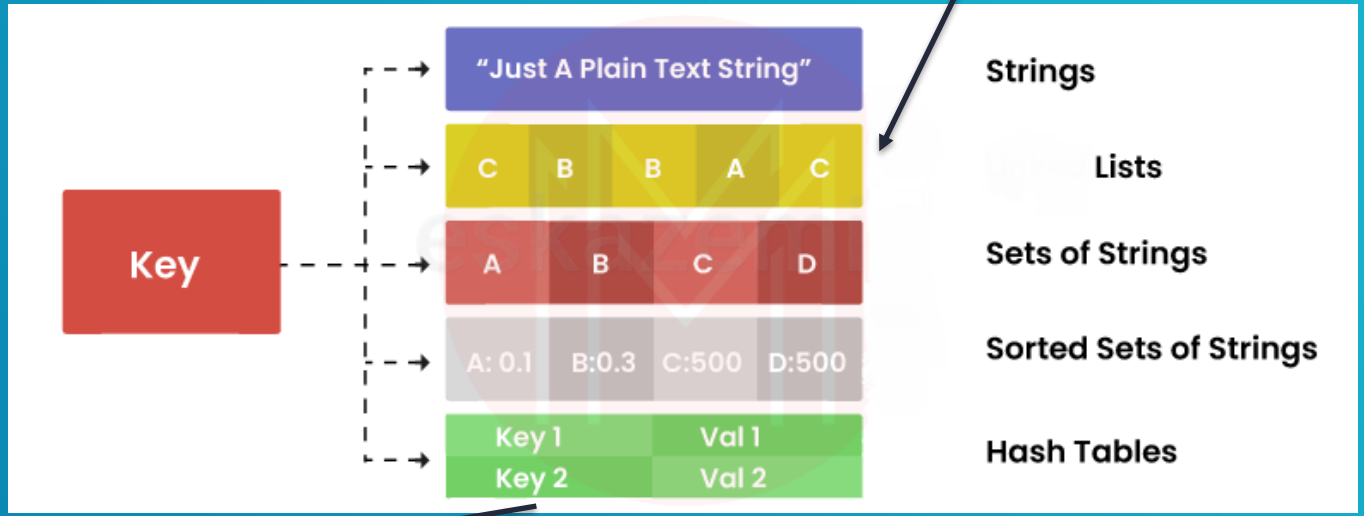
Data type

انواع ساختمان داده که Redis پشتیبانی می کند

1. String
2. hashes
3. Sets
4. Sorted sets
5. Lists
6. Bitmap
7. Stream
8. hyperloglogs

value

Valueمی تواند یه لیست باشد.



می تواند key-value باشد.

string

- حداکثر حجم هر string می تواند 512 مگابایت باشد.
- تمامی اعداد در redis به عنوان string ذخیره می شوند.
- Redis تمام مقادیر به صورت default به صورت string می بینه .

string commands

	command	description
1	SET key value	This command sets the value at the specified key.
2	GET key	Gets the value of a key.
3	GETRANGE key start end	Gets a substring of the string stored at a key.
4	GETSET key value	Sets the string value of a key and return its old value.
5	SETEX	Sets the value with the expiry of a key
6	SETNX	Sets the value of a key, only if the key does not exist

string commands

	command	description
7	STRLEN key	Gets the length of the value stored in a key
8	MSET key value [key value ...]	Sets multiple keys to multiple values
9	MSETNX key value [key value ...]	Sets multiple keys to multiple values, only if none of the keys exist
10	INCR key	Increments the integer value of a key by one
11	INCRBYFLOAT key increment	Increments the float value of a key by the given amount
12	DECR key	Decrements the integer value of a key by one

List

- با استفاده از لیست ها می توانند چندین مقدار داخل یک کلید ذخیره بکنید . المان های لیست به ترتیب ورود طبقه بندی می شوند یعنی ترتیب ورود اطلاعات مهم است.
- هر لیست می تواند بیشتر از 4 میلیارد المان در خود ذخیره کند.

List commands

	command	description
1	LPUSH key value1 [value2]	Prepends one or multiple values to a list
2	RPUSH key value1 [value2]	Appends one or multiple values to a list
3	LPUSHX key value	Prepends a value to a list, only if the list exists
4	RPUSHX key value	Appends a value to a list, only if the list exists
5	LINDEX key index	Gets an element from a list by its index

List commands

	command	description
6	LLEN key	Gets the length of a list
7	LPOP key	Removes and gets the first element in a list
8	LRANGE key start stop	Gets a range of elements from a list
9	LSET key index value	Sets the value of an element in a list by its index
10	BRPOP key1 [key2] timeout	Removes and gets the last element in a list, or blocks until one is available

List commands

دستورات مهم توضیحات بیشتر :

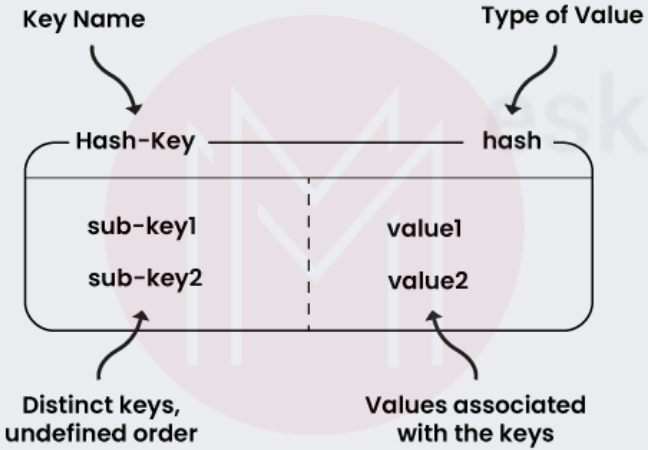
LPUSH RPUSH

هر دو ایجاد لیست با این تفاوت که LPUSH آخرین مقداری
که بهش داده میشه رو اول لیست قرار میده

BLPOP

برای حذف المان از لیست به کار می رود نکته ای که حائز اهیمت این است که زمان رو به عنوان ورودی میگیرد که اگر زمان صفر ست شود المان رو حذف می کنه اگر وجود داشته باشد و اگر نباشه اینقدر صبر می کنه تا داخل لیست المان قرار گیرد و اگر غیر صفر ست شود بعد از تایم مشخص شده غیر فعال می شود(بی خیال حذف می شود)

Hash



- نوعی از داده هستند که می توانید در یک کلید چندین مقدار به شکل کلید/مقدار ذخیره کنید.
- بیشترین استفاده از hash ها در Redis برای ذخیره آبجکت هاست.
- Hash ها توانایی ذخیره بیشتر از 4 میلیارد المان در خود را دارند.

Hash commands

	دستور	توضیح
1	HSET key field value	Sets the string value of a hash field
2	HMSET key field1 value1 [field2 value2]	Sets multiple hash fields to multiple values
3	HGET key field	Gets the value of a hash field stored at the specified key.
4	HGETALL key	Gets all the fields and values stored in a hash at the specified key
5	HDEL key field2 [field2]	Deletes one or more hash fields.

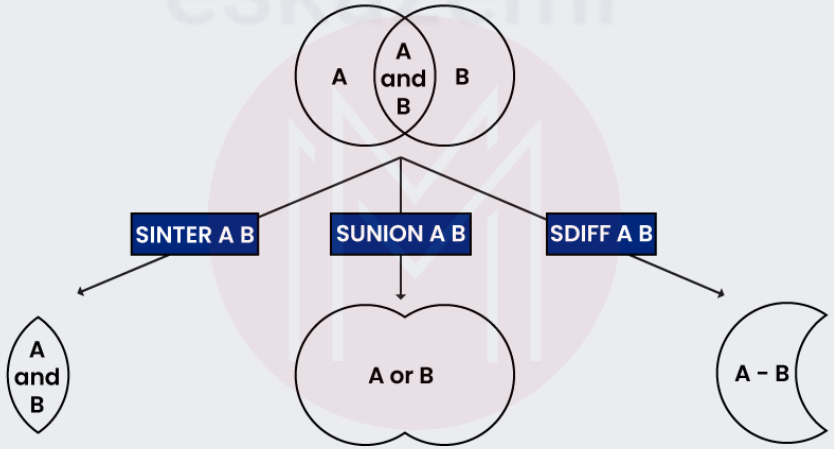
SET

- مجموعه ای از المان هایی است که به شکل نامنظم در یک کلید ذخیره می شوند .
- Set ها در Redis اجازه ذخیره المان های تکراری نمی دهندو اضافه کردن چندین بار المان های تکراری تاثیری در تعداد نخواهد داشت.
- هر set توانایی ذخیره بیشتر از 4 میلیارد المان در خود را دارند.

SUNION و SINTER به چه معناست؟

SUNION مجموعه ای برگردانده می شود که از اجتماع تمامی مجموعه هایی که داده می شود حاصل می شود.

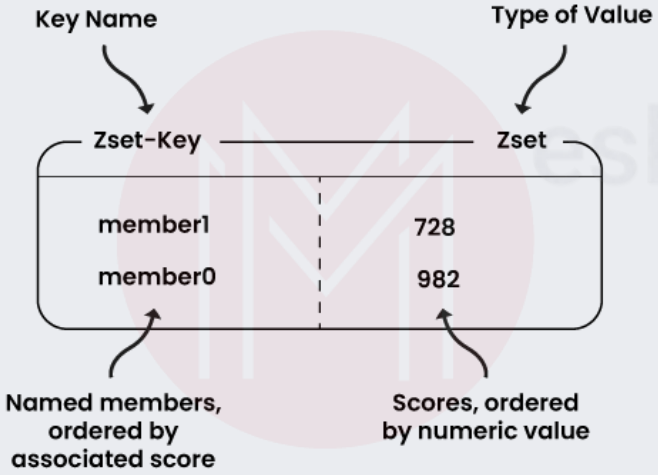
SINTER مجموعه ای برگردانده می شود که از اشتراک همه مجموعه هایی که داده می شود حاصل می شود.



Set commands

	دستور	توضیح
1	SADD key member1 [member2]	Adds one or more members to a set
2	SCARD key	Gets the number of members in a set
3	SPOP key	Removes and returns a random member from a set
4	SREM key member1 [member2]	Removes one or more members from a set
5	SRANDMEMBER key [count]	Gets one or multiple random members from a set.
6	SMEMBERS key	Gets all the members in a set
7	SISMEMBER key member	Determines if a given value is a member of a set

ZSET



- این نوع داده دقیقا شبیه setهاست و نمی توان المان های تکراری در آن ذخیره کرد . تنها تفاوتی که این دو باهم دارند این است که المان ها sorted_set یک score یا امتیاز دارند
- (از نوع float می باشد) که Redis المان ها را به ترتیب این score ها ذخیره می کند. در این نوع داده المان ها نمی توانند موارد تکراری داشته باشند اما امتیاز ها می توانند تکراری شوند.
- همچنین، می‌توانیم به ترتیب مرتب‌سازی شده و همچنین مقادیر امتیازات به موارد مجموعه دسترسی داشته باشیم.
- در ZSET ها اعمال خواندن ، اضافه کردن ، حذف کردن و آبدیت کردن المان ها بسیار سریع است.

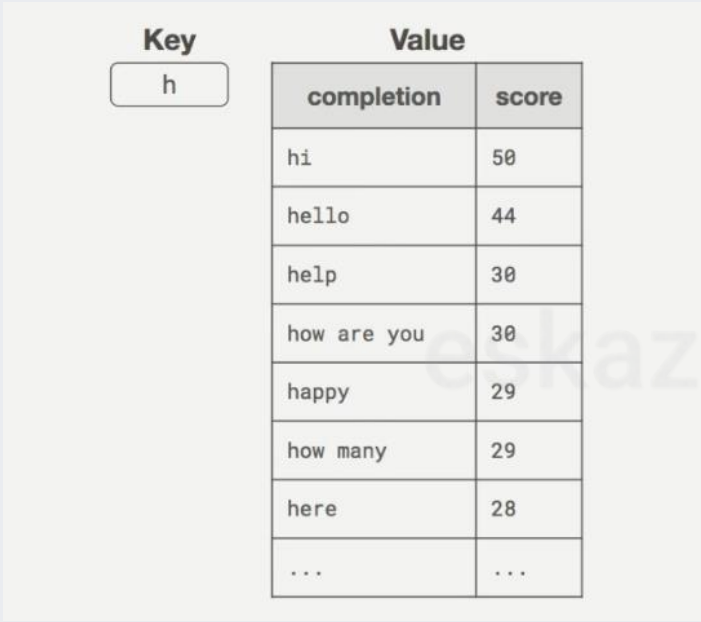
دستورات مهم:

ZADD: تمام اعداد مشخص شده با امتیازات مشخص شده به مجموعه مرتب شده ای که در کلید ذخیره شده است , اضافه می شوند.

ZREM : اعداد مشخص شده از مجموعه مرتب شده که در کلید ذخیره شده اند حذف می شوند . اعضای غیر موجود را نادیده می گیرد .

ZRANGE: طیف مشخصی از عناصر در مجموعه مرتب شده که در کلید ذخیره شده اند بازگردانده می شوند .

ZRANGEBTSCORE: تمام عناصر با امتیاز بین حداقل و حداکثر در مجموعه مرتب شده در کلید بازگردانده می شوند .



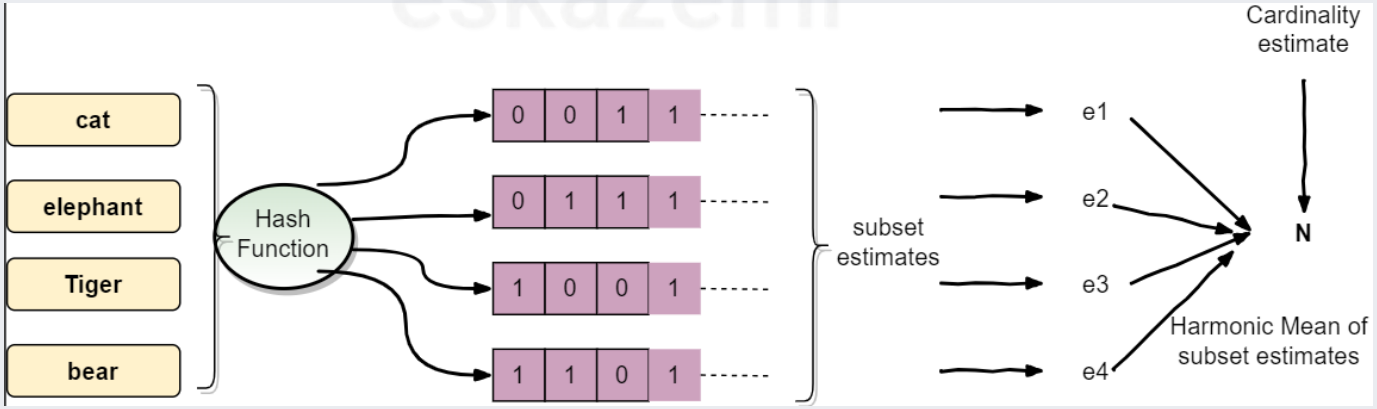
اگر امتیاز ها برابر باشند ، completion براساس حروف الفبا مرتب می شوند

hyperloglog

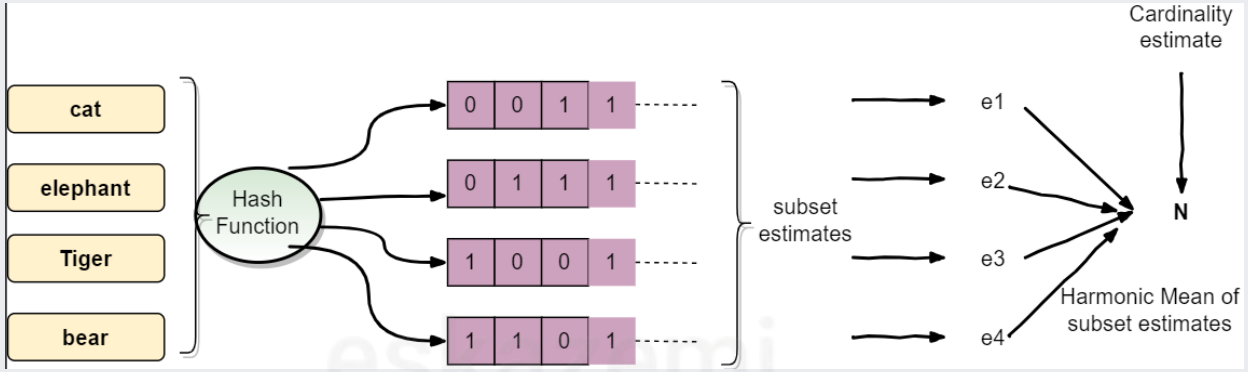
- مثل set ها هستند . یعنی امکان ذخیره موارد تکراری وجود ندارد و اینکه ترتیب ورود اطلاعات مهم نیست.
- این ساختار برای زمانی مناسب است که خود اطلاعات اهمیتی ندارند و فقط تعداد آنها مهم است .
- مثلا برنامه ای که به صورت real-time نیاز به ذخیره تعداد کاربر های Uniqe دارد یا اگر شما نیاز باشه تعداد ایمیل های uniqe در یک مجموعه 10 میلیون آدرس ایمیل بدست بیارید. جایی که نیمی از ایمیل ها unique باشند مقدار حافظه کافی برای ذخیره این اطلاعات نیازه که می توان از این نوع داده استفاده کنید که حل کرده این قضیه را با حداقل حافظه با استفاده از الگوریتم randomization algorithm (مخترع این الگوریتم ,Philippe Flajolet به همین خاطر که در ابتدای دستورات مربوط به این نوع داده از PF استفاده می کنیم.)
- بسیار بهینه تر از setها هستند. استفاده از منابع نسبت به set ها کمتر .

Algorithm Hyporloglog

این الگوریتم بر روی اعداد تصادفی توزیع شده یکنواخت کار می کند و برای تخمین تعداد آیتم های منحصر به فرد در یک مجموعه استفاده می شود. مفهوم کلیدی پشت این الگوریتم این است که اگر حداکثر تعداد صفرهای پیشرو در آیتم های مجموعه را بشماریم (که به اعداد تصادفی یکنواخت تبدیل شده اند و در باینری نمایش داده می شوند)، می توانیم تعداد آیتم های منحصر به فرد را با یک محاسبه ساده تخمین بزنیم.



If the maximum number of leading zeros is *n*, the estimated cardinality of the set is 2^n



فرض بگیرید می خواهیم تعداد بازدید های منحصر به فرد از یک سایت را بشماریم و ما داریم مجموعه ای بزرگ از IP های پشت سرهم که بازدید کردن از سایت ما:

ابتدا آدرس های IP ورودی را با استفاده از تابع هش به مجموعه ای از اعداد تصادفی توزیع شده یکنواخت تبدیل می کنیم cardinality .در اینجا تغییر نمی کند زیرا تنها آدرس های IP به اعداد تصادفی توزیع شده یکنواخت تبدیل می شوند.

این اعداد تصادفی با استفاده از چند بیت اولیه به زیرمجموعه های مختلفی تقسیم می شوند .تعداد حداکثر صفرهای پیشرو، در مقادیر آن، در حافظه ذخیره می شوند.

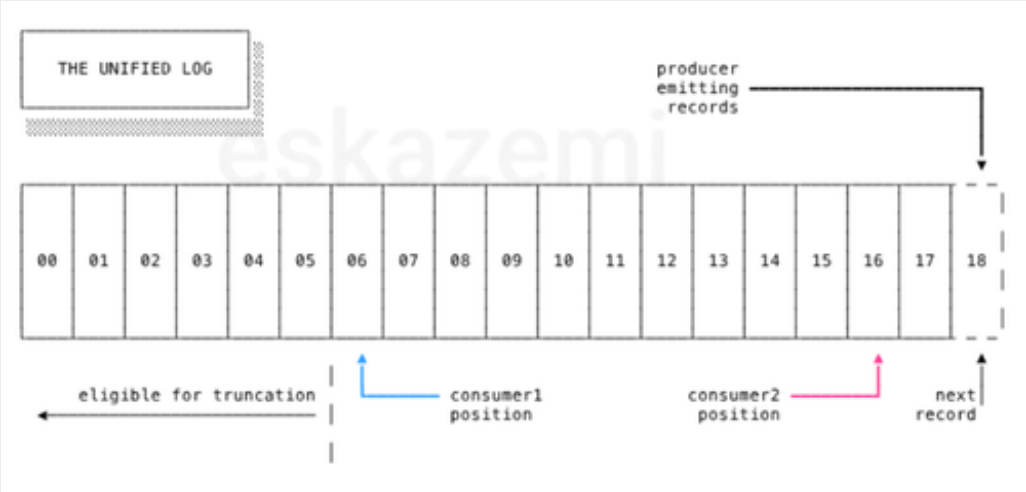
ما میانگین هارمونیک برآوردها را برای تمام زیرمجموعه های محاسبه شده قبلی محاسبه می کنیم.

میانگین هارمونیک محاسبه شده در بالا تخمینی از تعداد بازدید کننده منحصر به فرد در صفحه وب است.

Hyperloglog commands

	دستور	توضیح
1	PFADD key element [element ...]	Adds the specified elements to the specified HyperLogLog.
2	PFCOUNT key [key ...]	Returns the approximated cardinality of the set(s) observed by the HyperLogLog at key(s)
3	PFMERGE destkey sourcekey [sourcekey ...]	Merges N different HyperLogLogs into a single one.

Streams



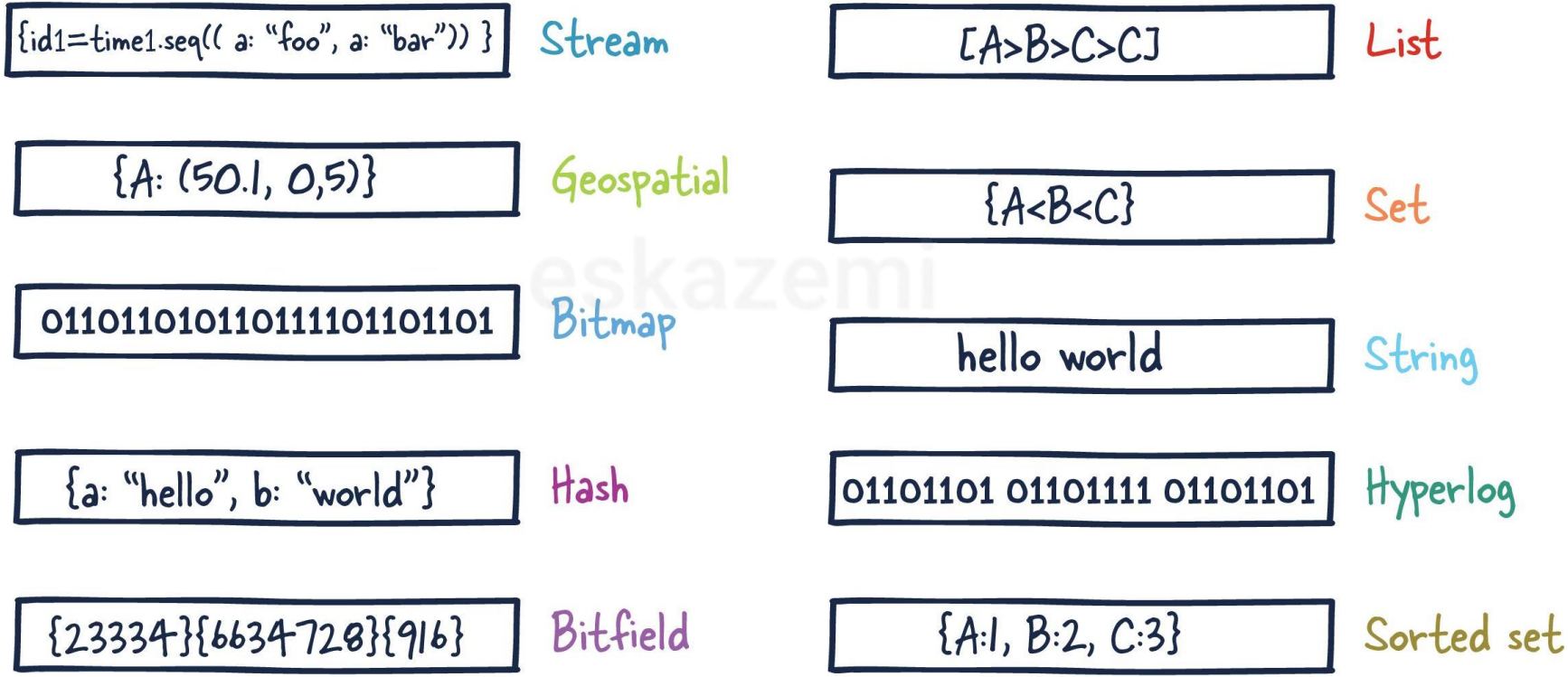
Streams

- ✓ Redis Streams یک ساختار داده فوق العاده قدرتمند برای مدیریت جریان های داده با سرعت بالا (مانند صف پیام)است.
- ✓ با پارتیشن بندی (out-of-the-box)، تکرار و پایداری، می تواند میلیون ها نقطه داده در ثانیه را با تاخیر زیر میلی ثانیه دریافت و پردازش کند.
- ✓ مبتنی بر پیاده سازی کارآمد radix-tree است (الگوریتمی که در آن هر گره که تنها فرزند است با والد ادغام می شود)، که دامنه و جستجوی پرس و جوها را بسیار سریع می کند.
- ✓ تولیدکنندگان(producers) و مصرف کنندگان (consumers) را با تماس های ناهمزمان متصل می کند و از گروه های مصرف کننده(consumer) پشتیبانی می کند.

Bitfields

Bitfields روشی فشرده و کارآمد برای پیاده سازی شمارنده های چندگانه در یک آرایه واحد ارائه می دهند. این کار امکان افزایش و کاهش شمارنده ها در یک موقعیت مشخص را فراهم می کند، و هنگامی که شمارنده به حد بالایی خود می رسد، flags ها سرریز می شوند.

offset 0	offset 1	offset 2	offset 3	offset 4
1	0	0	1	1


{id1=time1.seq((a: "foo", a: "bar")) }
Stream
{A: (50.1, 0,5)}
Geospatial
01101101011011110110110 1
Bitmap
{a: "hello", b: "world"}
Hash
{23334}{6634728}{916}
Bitfield
[A>B>C>C]
List
{A<B<C}
Set
hello world
String
01101101 01101111 01101101
Hyperlog
{A:1, B:2, C:3}
Sorted set

Redis چگونه اخراج داده ها را مدیریت می کند؟

Redis از مکانیسم های مختلفی برای حذف داده ها پشتیبانی می کند مانند LRU که اخیرا استفاده شده است و TTL مقدار انقضا

حداکثر تعداد کلید های که می توان در Redis ذخیره کرد به مقدار حافظه موجود در سیستم محدود می شود.

Thanks!

Any questions?

You can find me at:

- @eskazemi
- m.esmaeilkazemi@gmail.com