All you need to know on Cache troubleshooting - Redis

SRE Tech-Tonic Session.

- ArunSanthoshKumar and Karthik.

What is Cache

- Cache Memory is a special very high-speed memory.
- Smaller and faster memory that stores copies of the data from frequently used main memory locations.
- Delivers fast (sub-millisecond response) and consistent data to micro services or Applications.
- Benefiting Industries like gaming, Ed-tech, financial services, healthcare, and IoT.

.

Why is Cache

- Performance is important
- Dealing the large Database
- Least Recently used.
- HardDisk (HDD/SSD) Vs InMemory(RAM)
- Computation Cost is High
- Customer Experience
- Application Usability

Types of Caching

- Browser or Client Side
- DNS Cache
- Proxy
- CDN Cache for Static pages (Location based)

- Load Balanced or Reverse Proxy or API gateway
- Server Side Cache (Application Cache)
- Database Cache

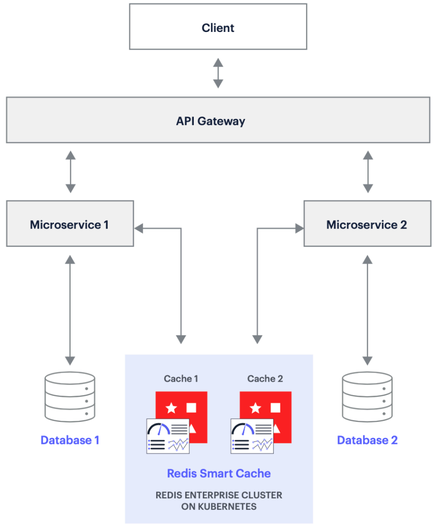
Things to Cache

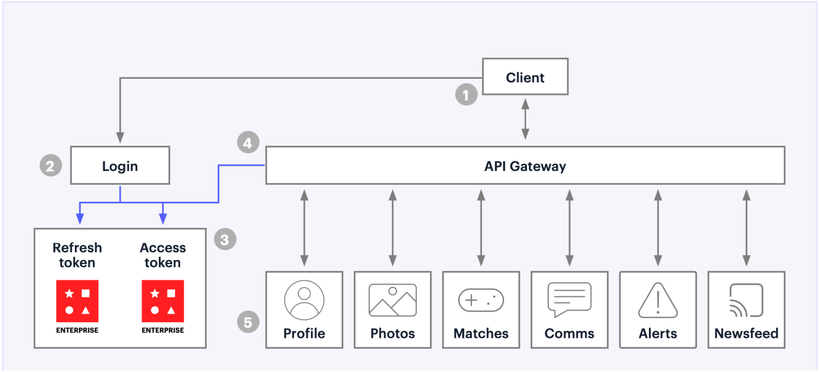
- Machine Learning Interfaces.
- Full Text Searches
- Complex relational database queries
- User session records
- Result of query from Microservices
- Third Party API Results

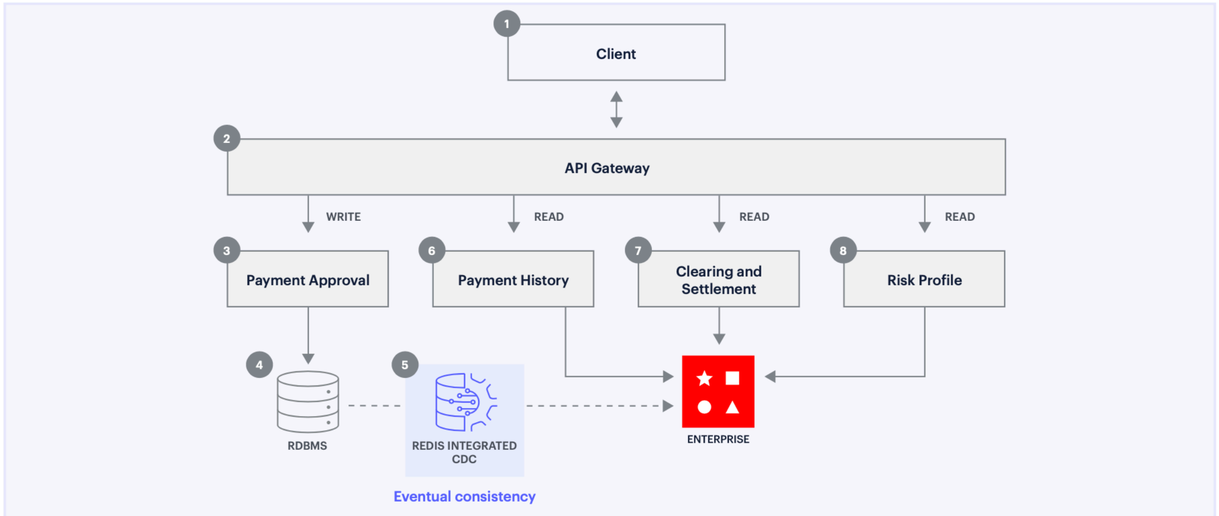
Caching Softwares.

Application Architecture - Redis

- **API gateway caching** for globally shared data that must be accessed by all micro services (session data, authentication tokens, etc.)
- **Command Query Responsibility Segregation (CQRS)** for data shared by multiple micro-services, but not needed by all at the global level (cross- domain data)
- **Query caching** for data within a single micro service (domain-specific)





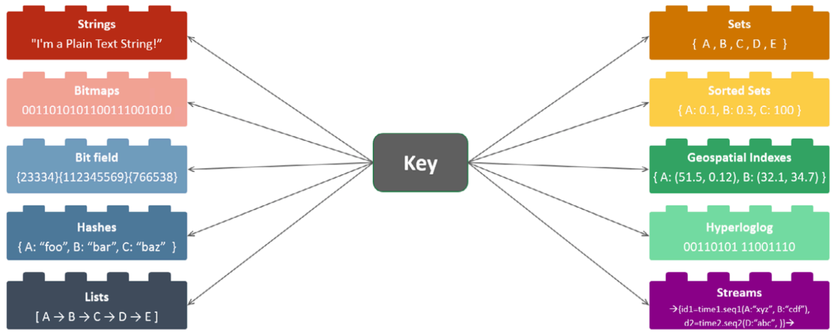


How Data Can be cached - Redis

Data Structure

- String
- Hashes
- Lists
- Sets
- Sorted Set
- Streams

- Bitmaps
- BitField
- Geospatial Indexes
- Hyperloglog.



Redis Stack

- Json
- Time Series
- Blob
- Search
- Bloom

SET title:1 “CahceDemo"

GET title:1

HSET user:1 name arun title sse

HGETALL user:1

LPUSH tutorials redis mongo

LRANGE tutorials 0 10

SADD lang redis mongo

SMEMBERS lang

ZADD prog 10 redis 20 mongo

ZRANGE prog 0 10 withscores

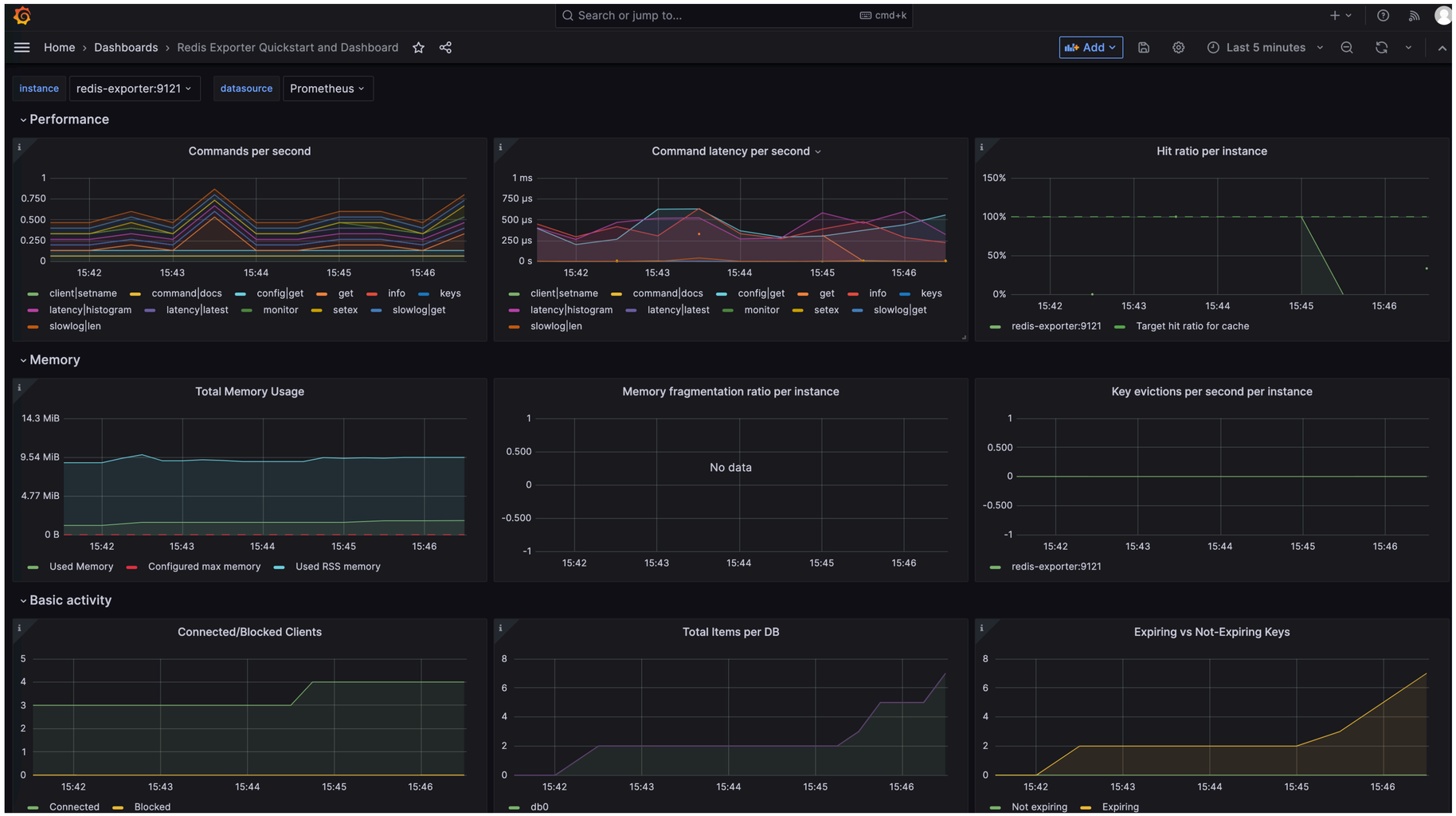
How to Gauge Performance - Redis

- Commands per Second (no of commands per second)
- Cache Hit Ratio(H) = hit / (hit + miss) = no. of hits/total accesses (how effective)
- Cache Miss Ratio = miss / (hit + miss) = no. of miss/total accesses = 1 - hit ratio(H)
- Slow Log
- Latency(in Ms) - Client request and Server response

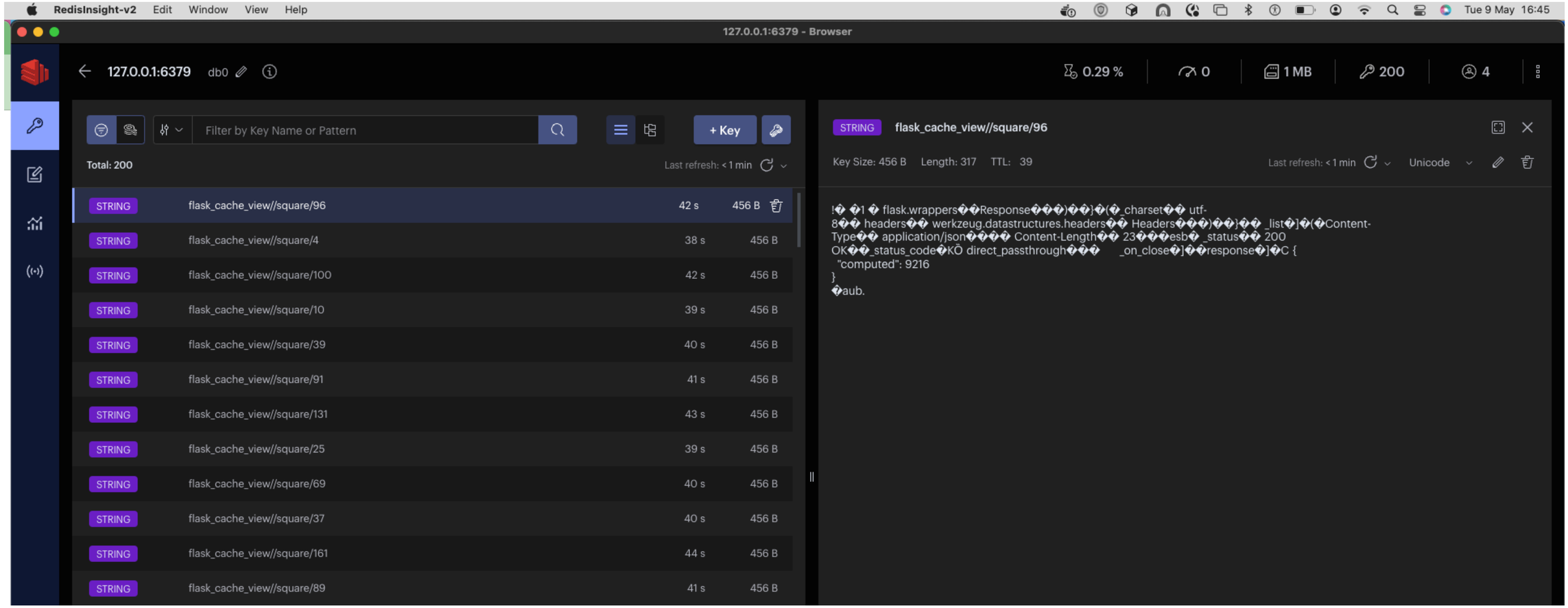
Redis - TroubleShooting with Cli

```
127.0.0.1:6379>
# redis-cli
127.0.0.1:6379> info commandstats
# Commandstats
cmdstat_info:calls=334,usec=91659,usec_per_call=274.43,rejected_calls=0,failed_calls=0
cmdstat_client|setname:calls=332,usec=1288,usec_per_call=3.88,rejected_calls=0,failed_calls=0
cmdstat_client|list:calls=1,usec=76,usec_per_call=76.00,rejected_calls=0,failed_calls=0
cmdstat_monitor:calls=2,usec=4,usec_per_call=2.00,rejected_calls=0,failed_calls=0
cmdstat_command|docs:calls=3,usec=6309,usec_per_call=2103.00,rejected_calls=0,failed_calls=0
cmdstat_slowlog|len:calls=332,usec=969,usec_per_call=2.92,rejected_calls=0,failed_calls=0
cmdstat_slowlog|get:calls=332,usec=1085,usec_per_call=3.27,rejected_calls=0,failed_calls=0
cmdstat_keys:calls=11,usec=570,usec_per_call=51.82,rejected_calls=0,failed_calls=0
cmdstat_setex:calls=870,usec=15803,usec_per_call=18.16,rejected_calls=0,failed_calls=0
cmdstat_config|get:calls=332,usec=93308,usec_per_call=281.05,rejected_calls=0,failed_calls=0
cmdstat_get:calls=1100,usec=9779,usec_per_call=8.89,rejected_calls=0,failed_calls=0
cmdstat_latency|latest:calls=332,usec=959,usec_per_call=2.89,rejected_calls=0,failed_calls=0
cmdstat_latency|histogram:calls=332,usec=108699,usec_per_call=327.41,rejected_calls=0,failed_calls=0
127.0.0.1:6379> info keyspace
# Keyspace
db0:keys=200,expires=200,avg_ttl=19735
127.0.0.1:6379> client list
id=30 addr=172.20.0.5:60268 laddr=172.20.0.2:6379 fd=8 name= age=4576 idle=33 flags=N db=0 sub=0 psub=0 ssub=0 multi=-1 qbuf=0 qbuf-free=0 argv-mem=0 multi-mem=0 rbs=10
4 rbp=0 obl=0 oll=0 omem=0 tot-mem=1800 events=r cmd=setex user=default redir=-1 resp=2
id=339 addr=127.0.0.1:39266 laddr=127.0.0.1:6379 fd=9 name= age=27 idle=0 flags=N db=0 sub=0 psub=0 ssub=0 multi=-1 qbuf=26 qbuf-free=20448 argv-mem=10 multi-mem=0 rbs=
024 rbp=0 obl=0 oll=0 omem=0 tot-mem=22298 events=r cmd=client|list user=default redir=-1 resp=2
127.0.0.1:6379> keys 8
(empty array)
127.0.0.1:6379> keys *
  1) "flask_cache_view//square/117"
  2) "flask_cache_view//square/87"
  3) "flask_cache_view//square/165"
  4) "flask_cache_view//square/154"
  5) "flask_cache_view//square/195"
  6) "flask_cache_view//square/156"
  7) "flask_cache_view//square/186"
```

Redis - Troubleshooting With Grafana



Redis - Troubleshooting with Insight.



How to improve cache performance. - Redis

- Deciding when to set and get from Cache.
- Usable Key Combination to set in cache.
- Setting a TTL for Cache Entries
- Selecting the right data eviction policy.
- Limiting the response size to store in the keys.
- Using the Different data structure to efficiently store and respond.
- High availability - Replica to serve the consistent data.
- Cluster the instance for handling the load by sharding.
- Proper Capacity planning based on the requirement.
- Use of the right library based on the use case.

Live Demo and Walkthrough

Simple Micro-service with Cache and Without Cache. (Using Docker)
Redis Cli Navigation
Redis exporter Monitoring
Prometheus and Grafana metric.
Client Use case (Kubernetes Redis Monitoring)

Thanks and Questions.

Connect LinkedIN. - https://www.linkedin.com/in/arun7pulse/
Reach me @Teams. - aannamalai@altimetrik.com