

EP47: Common Load-balancing Algorithms



ALEX XU
FEB 18, 2023



126



8



2

Share



This week's system design refresher:

- Top 5 Uses of Redis (Youtube video)
- Common load-balancing algorithms
- Types of VPNs
- Possible experiment platform architecture

Top 5 Uses of Redis

Most people think Redis is just for caching.

But Redis can do so much more than that. It is good for:

- Session store
- Distributed lock
- Counter
- Rate limiter
- Ranking/leaderboard
- etc.

In this video, we get insights into how Redis solves interesting scalability challenges and learn why it is a great tool to know well in our system design toolset.

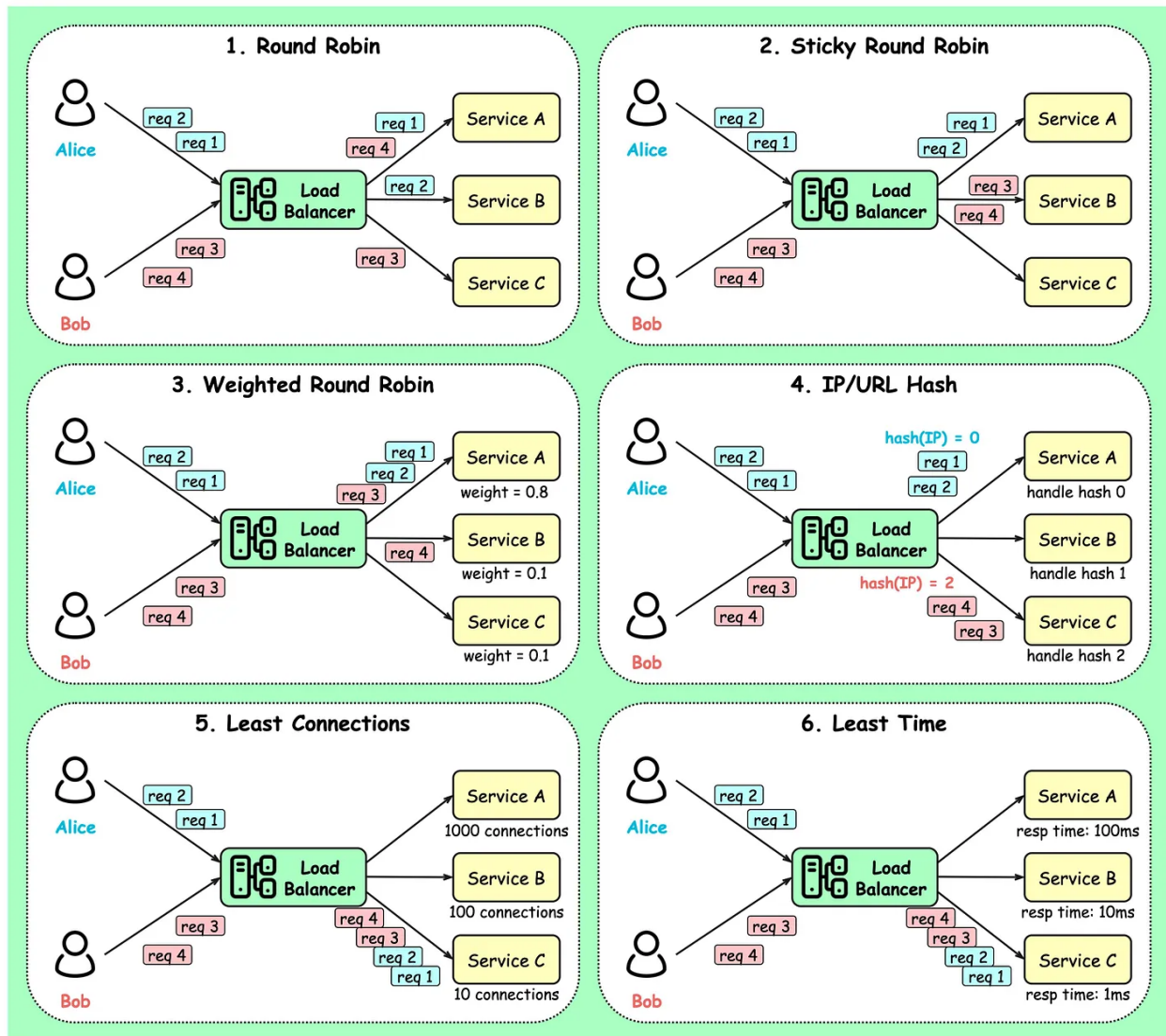
Top 5 Redis Use Cases



What are the common load-balancing algorithms?

The diagram below shows 6 common algorithms.

Load Balancing Algorithms



- **Static Algorithms**

1. Round robin

The client requests are sent to different service instances in sequential order. The services are usually required to be stateless.

2. Sticky round-robin

This is an improvement of the round-robin algorithm. If Alice's first request goes to service A, the following requests go to service A as well.

3. Weighted round-robin

The admin can specify the weight for each service. The ones with a higher weight handle more requests than others.

4. Hash

This algorithm applies a hash function on the incoming requests' IP or URL. The requests are routed to relevant instances based on the hash function result.

- **Dynamic Algorithms**

5. Least connections

A new request is sent to the service instance with the least concurrent connections.

6. Least response time

A new request is sent to the service instance with the fastest response time.

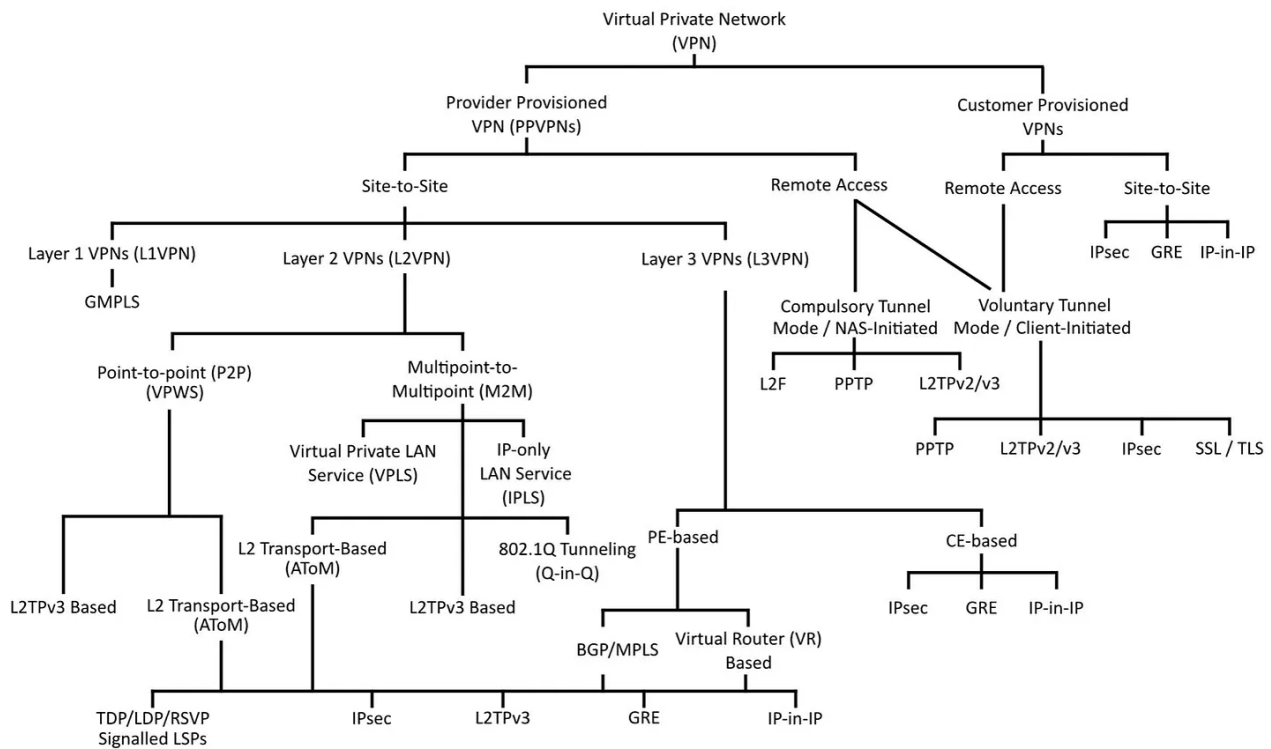
👉 Over to you:

1. Which algorithm is most popular?
2. We can use other attributes for hashing algorithms. For example, HTTP header, request type, client type, etc. What attributes have you used?

Types of VPNs

Think you know how VPNs work? Think again! 🤖 It's so complex.

Types of VPNs

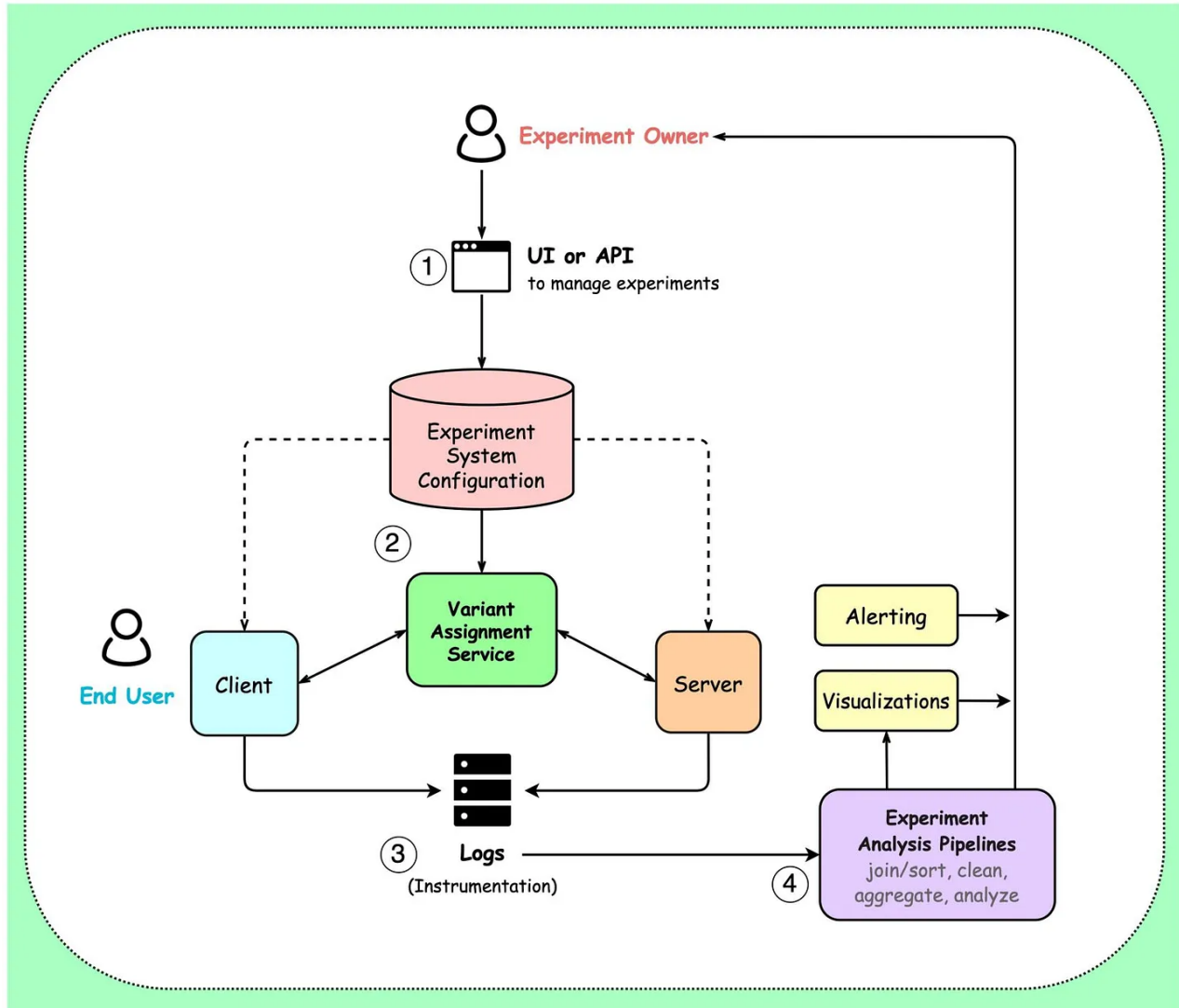


Source: Wikipedia Virtual private network

Possible experiment platform architecture

The architecture of a potential experiment platform is depicted in the diagram below. This content of the visual is from the book: "Trustworthy Online Controlled Experiments" (redrawn by me). The platform contains 4 high-level components.

Possible experiment platform architecture



- Experiment definition, setup, and management via a UI. They are stored in the experiment system configuration.
- Experiment deployment to both the server and client-side (covers variant assignment and parameterization as well).
- Experiment instrumentation.
- Experiment analysis.

The book's author [Ronny Kohavi](https://www.linkedin.com/in/ronnykohavi/) also teaches a live Zoom class on Accelerating Innovation with A/B Testing. The class focuses on concepts, culture, trust, limitations, and build vs. buy. You can learn more about it here:

<https://lnkd.in/eFHVuAKq>

126 Likes · 2 Restacks



8 Comments



Write a comment...



Yuan Writes Yuan's Substack Mar 12 · *edited Mar 12*

Kind of curious about the questions in LB section. Here is my thought

- Which algorithm is most popular?

I think maybe round robin or hash.

Round robin seems very simple to implement and maintain. If the overload of requests are normal distribution, round robin should get pretty even server loading across clusters.

Hash is quite useful if people want some requests go to the specific servers. It could achieve sticky session like functionality (same client go to same server) but with more flexibility (choose different attribute). The problem maybe is the hot key issue and distribute overload unevenly.

- We can use other attributes for hashing algorithms. For example, HTTP header, request type, client type, etc. What attributes have you used?

I think i would choose IP or specified header.

Choosing IP is because same IP means same client most of time. Then same client go to same server could benefit from local cache hit.

Choosing specified header is quite the same idea. I have set specific header and using AWS ELB target group rules to let some users go to the special target group. I can debug or control the behaviors.

♡ LIKE (1) 💬 REPLY ↗ SHARE

...



NK Writes System Design Newsletter Jun 16

- Which algorithm is most popular?

Round robin is probably the most common algorithm to load balance stateless services.

The consistent hashing algorithm is used by popular services such as Discord for sticky sessions without running into scaling issues. The simple hash algorithm cannot handle dynamic changes in server instances.

- We can use other attributes for hashing algorithms. For example, HTTP header, request type, client type, etc. What attributes have you used?

Client type (HTTP Header User-Agent) can be used to segregate mobile and desktop client traffic.

The HTTP request payload (user-id, session-id, or thing-id) can be used for routing traffic to the same service instance.

♡ LIKE 💬 REPLY ↗ SHARE

...

6 more comments...

© 2023 ByteByteGo · [Privacy](#) · [Terms](#) · [Collection notice](#)
[Substack](#) is the home for great writing