How to Benchmark Redis

Or Shwartz
Product Manager @Redis Labs

Filipe Oliveira
Performance Engineer @Redis Labs

redisconf2021

Or Shwartz

Product Manager @ Redis Labs

- Core-team Product Manager
- Develope Redis Labs’ core platform

redisconf2021

Filipe Oliveira
Performance Engineer @Redis Labs

- Focused on Redis Modules and OSS performance
- Improve/develop open source performance/observability
    - Some by necessity, some for the fun of it

redisconf2021

Agenda

- General performance testing 101
- Redis benchmark tools overview
- Benchmark demo
- Takeaways and next steps for Redis benchmarks

WHY Should You Care About Redis Performance Benchmarks?

redisconf2021

WHY Should You Care - as a user

- Test the performance of your application
- Idialy size your infrastructure
- Compare Redis to other technologies

WHY Should You Care - as a Redis developer

- Understand the baseline performance of new features
- Improve Redis performance
- Intercept regressions

General Performance Testing 101

redisconf2021

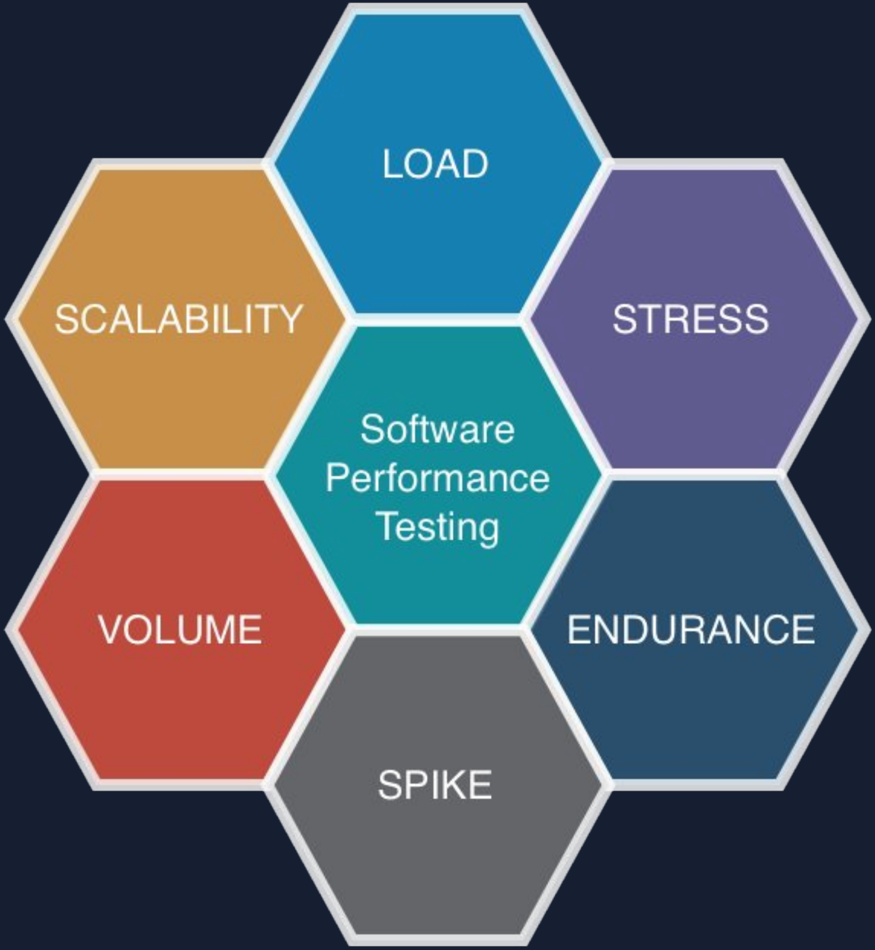
Requirements For Valid Performance Tests

1. Stable testing environment
2. Well defined key performance indicators
3. Deterministic testing tools
4. Deterministic outcomes
5. [**] Reduced testing/probing overhead
6. [**] Reduce tested changes to the minimal

[**] Redis developer related

Understand What You Are Measuring

1. Commands patterns and ratios
2. Ops per-second vs latency
3. Cluster vs a single node
4. Cost & sizing benchmarking



1) Stable testing environment

- Cloud vs on-prem:
    - production graded cloud VMs
    - **No pet VMs:** deployed/teared-down on the fly
- If running comparisons reduce tested changes to the minimal
- Deep understanding of underlying infrastructure performance
    - network(netperf, iperf3,...)
    - compute performance
    - measuring sustained memory bandwidth

2) Well defined key performance indicators

- Understand your application requirements
- Your product defines the key performance indicators and their thresholds
    - Sustainable ops/sec
    - Sustainable latency
- Performance per $. Efficiency matters as much as performance.

3) Deterministic testing tools

- Deterministic randomness
- The tool should not be the bottleneck
- Large pre-processing steps done once and pre-testing...

more about tooling next....

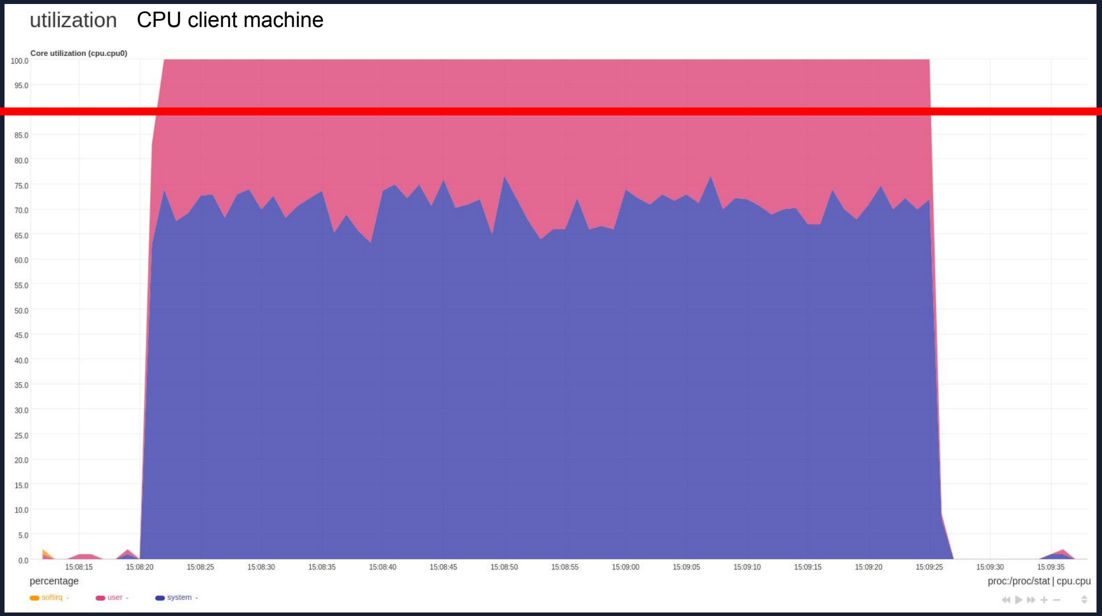
4) Deterministic outcomes

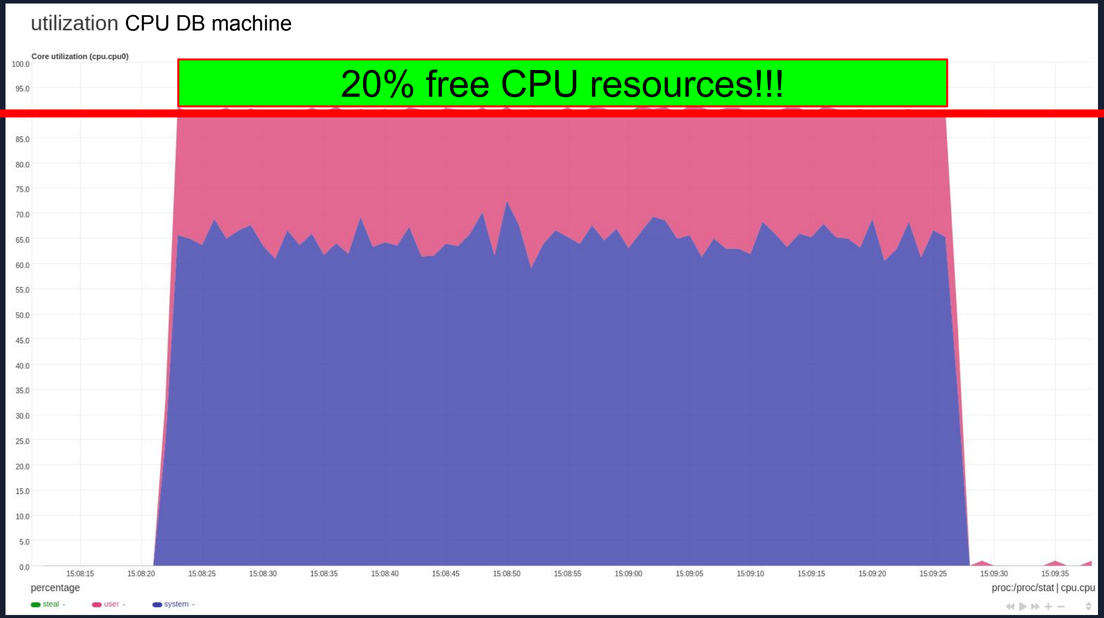
- Include warm-up on low-starting systems
- K-best testing methodology
- Prefer metrics that represent a set of queries (percentiles over averages) [1]

[1] ApacheCon Why averages lie and how to truely monitor your systems

Common pitfalls: client bottleneck

1. Performance bottleneck is on client side, and not server-side



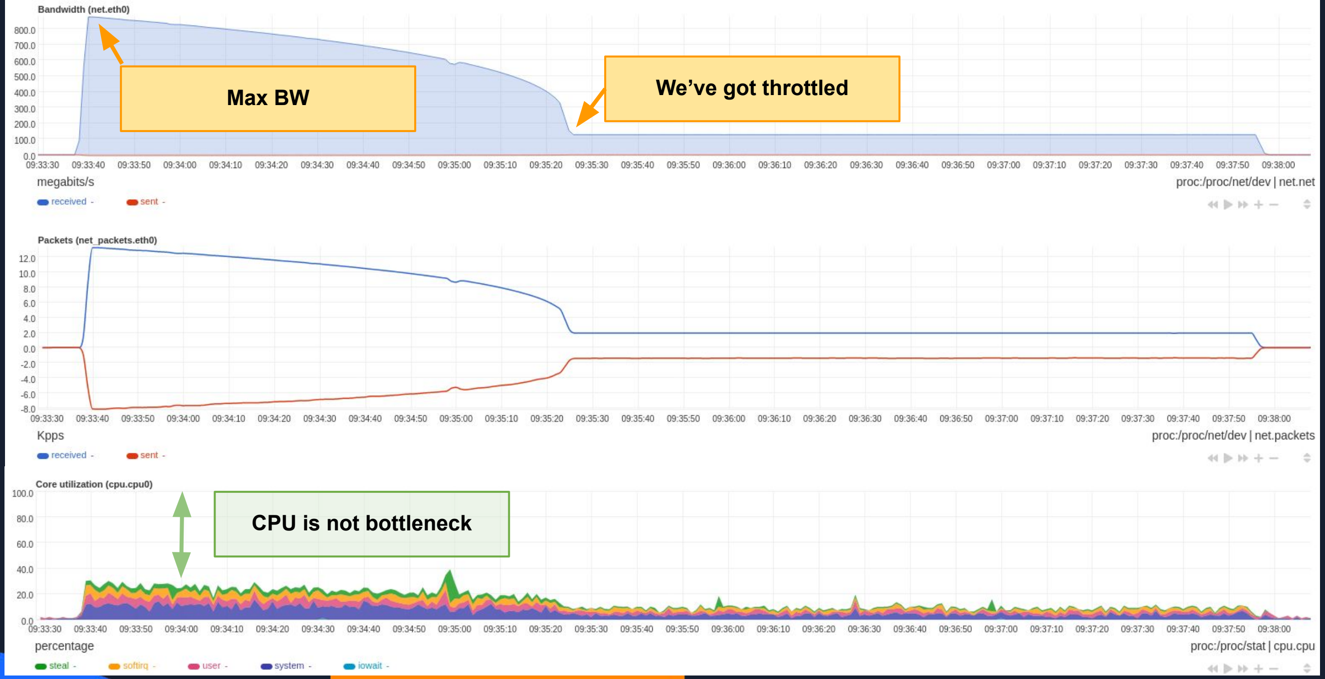


20% free CPU resources!!!

Common pitfalls: infra bottleneck

1. Redis throughput is limited by the network well before being limited by the CPU

WHILE BENCHMARKING REDIS



Max BW

We've got throttled

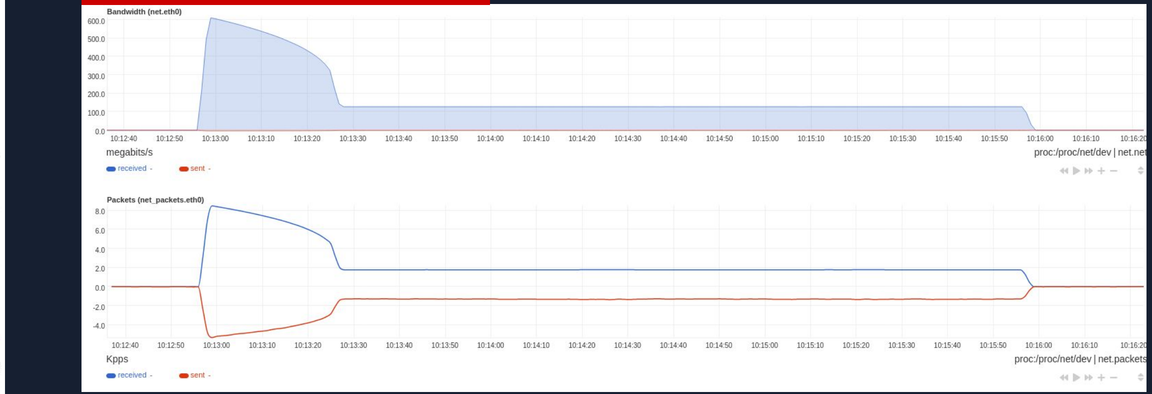
CPU is not bottleneck

2. **Prove it:** validate with iperf BW/PPS

```
~# iperf3 -t 180 -P 50 -c <HOST>
(...)
[SUM]   0.00-180.00 sec  3.94 GBytes   188 Mbits/sec  47545             sender
[SUM]   0.00-180.00 sec  3.92 GBytes   187 Mbits/sec                    receiver

iperf Done.
```

WHILE BENCHMARKING NETWORK



redisconf 2021

Common pitfalls

1. More misconceptions: https://redis.io/topics/benchmarks

redisconf2021

Redis Benchmarks Tooling Overview

redisconf2021

redis-benchmark

> https://github.com/redis/redis

redis-benchmark

PROS

- Quick and useful way to get some figures
- Built-in tool - if you have redis-server, you have redis-benchmark!
- Support simple command patterns

CONS

- Always print the errors!!
- Lack of multi-command benchmarks
- By default the benchmark runs against a single key
- It’s easy to reach client’s bottlenecks

memtier_benchmark

> https://github.com/RedisLabs/memtier_benchmark

memtier_benchmark

PROS

- Multi-command benchmarks
- Pseudo-random data, data access patterns, range manipulation
- Advanced metric collection and exporting
- Full latency spectrum

CONS

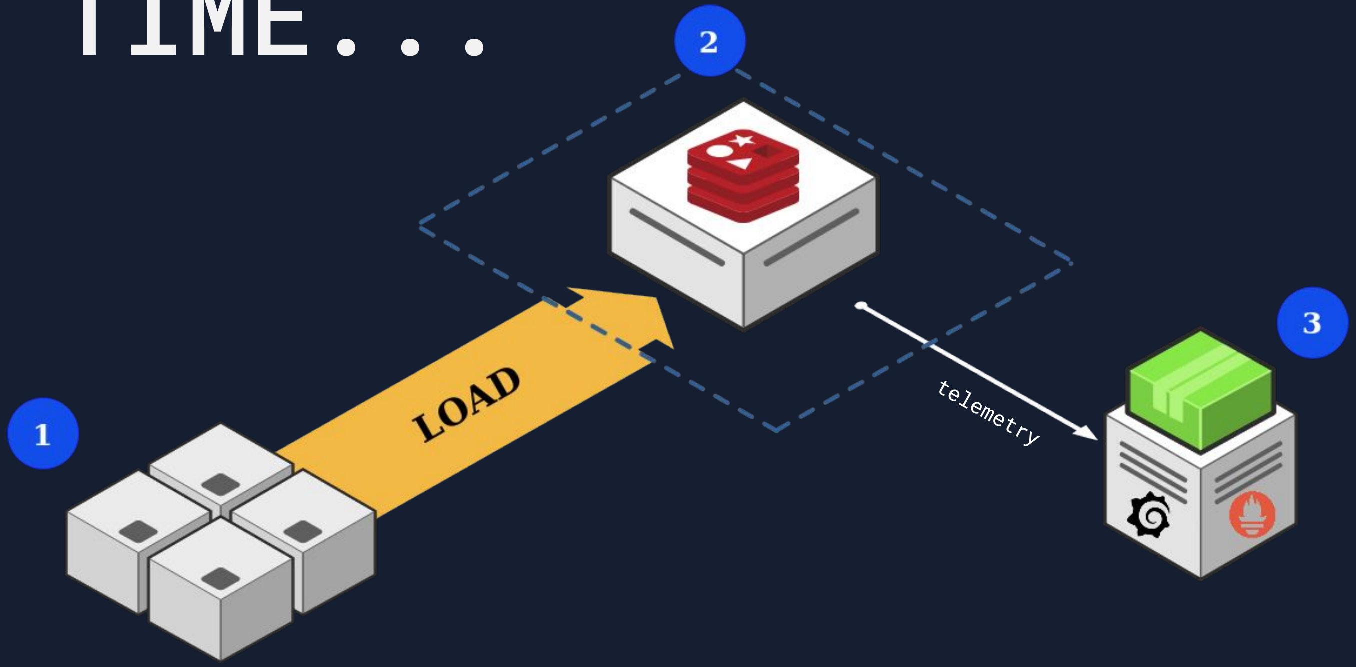
- Sustainable Throughput SLAs / Os testing (Github issue)

Demo Application Workloads Simulation

Requirements - set the goal, infrastructure, use case, and KPIs

- Maximum achievable throughput for our data / use-case constraints
- Infrastructure requirements: m5d.8xlarge, Ubuntu 20.04
- Hashes datatype, 1M keys, 1KB values, 95% reads, 5% writes
- SLA: 99% requests to be served below 10ms (max of 50 concurrent APP connections to the DB)

DEMO TIME...



https://github.com/filipecosta90/redisconf21-how-to-benchmark-redis-demo

redisconf2021

Key Takeaways

- Battle tested tools
- Stable environments
- Well defined key performance indicators
- Know your application
- Be critical and share you thoughts with community

More Resources

- https://redislabs.com/benchmarks
- https://redis.io/topics/benchmarks

redisconf2021

Thank you.

redisconf2021