

讲师介绍



Hash QQ: 805921455

从事Java软件研发十年。
前新浪支付核心成员、
咪咕视讯(中国移动)项目经理、
对分布式架构、高性能编程有深入的研究。

明天，你一定会感谢今天奋力拼搏的你

手把手搭建主从集群， 让你的Redis更高效

分布式高并发—缓存技术

目录

课程安排



01

搭建高可用集群

主从集群架构分析、搭建主从复制的集群



02

客户端及监控

如何动手接入集群、集群的健康状态监控



03

实现Redis读写分离

数据库的读写分离我们知道，Redis的呢？



04

总结

会学习的人都会总结



搭建高可用集群

Redis主从集群架构分析

什么是主从复制？



为什么要使用主从复制？

- redis-server 单点故障
- 单节点QPS有限
- 持久化，从处理持久化，避免对主性能影响

主从复制应用场景分析

- 读写分离场景，规避redis单机瓶颈
- 故障切换，master出问题后还有slave节点可以使用

搭建主从复制

主Redis Server以普通模式启动，主要是启动从服务器的方式

1、 第一种方式：命令行

```
# 连接需要实现从节点的redis，执行下面的命令  
slaveof [ip] [port]
```

2、 第二种方式：redis.conf 配置文件

```
# 配置文件中增加  
slaveof [ip] [port]  
# 从服务器是否只读(默认yes)  
slave-read-only yes
```

退出主从集群的方式

```
Slaveof no one
```

查看主从复制信息

master

```
127.0.0.1:6379> info replication
# Replication
# 角色 主
role:master
# 当前从服务器数量1
connected_slaves:1
# 从服务器信息
slave0: ip=127.0.0.1, port=6380, state=online, offset
=15, lag=1
# 计数器，主节点复制偏移量(复制的字节数)
master_repl_offset:351
# 主从同步缓存区
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:14
```

Slave

```
127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:up
master_last_io_seconds_ago:7
master_sync_in_progress:0
slave_repl_offset:351
# 从节点优先级
slave_priority:100
# 是否为只读节点
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
```

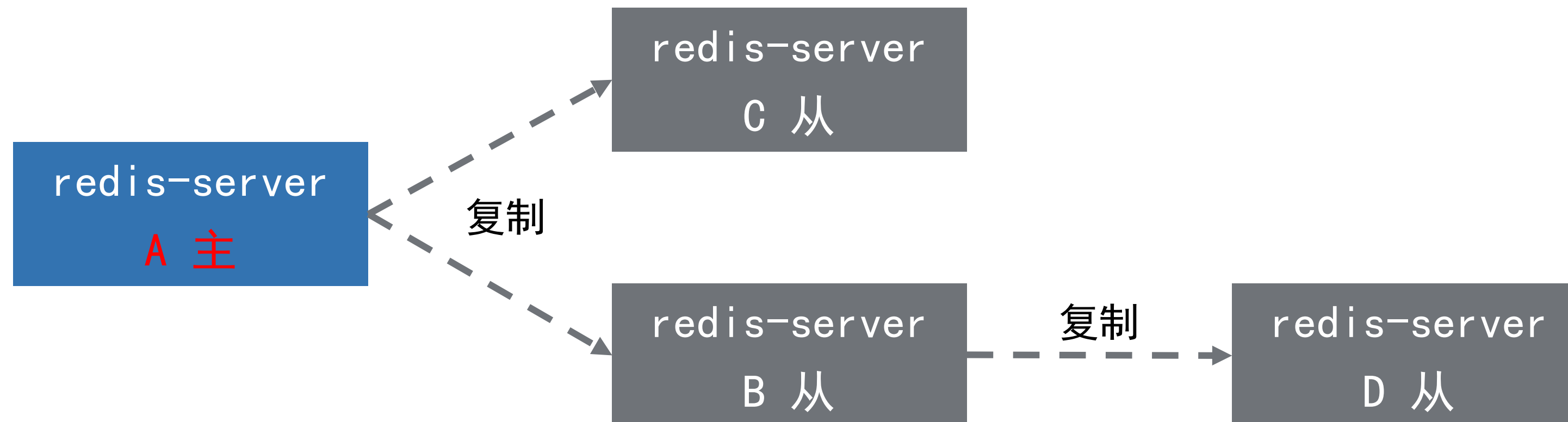
主从复制流程



1. 从服务器通过psync命令发送服务器已有的同步进度(同步源ID、同步进度offset)
2. master收到请求, 同步源为当前master, 则根据偏移量增量同步
3. 同步源非当前master, 则进入全量同步: master生成rdb, 传输到slave, 加载到slave内存

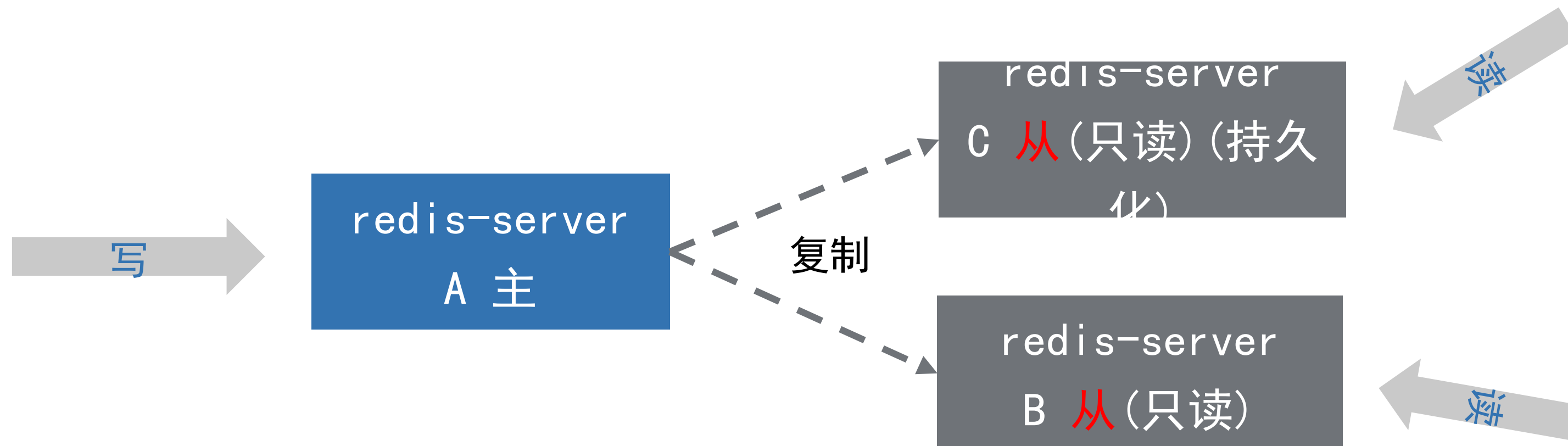
主从复制核心知识

- Redis 默认使用异步复制，slave 和 master 之间异步地确认处理的数据量
- 一个 master 可以拥有多个 slave
- slave 可以接受其他 slave 的连接。slave 可以有下级 sub slave
- 主从同步过程在 master 侧是非阻塞的
- slave 初次同步需要删除旧数据，加载新数据，会阻塞到来的连接请求



主从复制应用场景

- 主从复制可以用来支持读写分离
- slave服务器设定为只读，可以用在数据安全的场景下。
- 可以使用主从复制来避免 master 持久化造成的开销。master 关闭持久化，slave 配置为不定期保存或是启用 AOF。（**注意：**重新启动的 master 程序将从一个空数据集开始，如果一个 slave 试图与它同步，那么这个 slave 也会被清空。）



主从复制的注意事项

读写分离场景：

- 数据复制延时导致读到过期数据或者读不到数据（网络原因、slave阻塞）
- 从节点故障（多个client如何迁移）

全量复制情况下：

- 第一次建立主从关系或者runid不匹配会导致全量复制
- 故障转移的时候也会出现全量复制

复制风暴：

- master故障重启，如果slave节点较多，所有slave都要复制，对服务器的性能，网络的压力都有很大影响。
- 如果一个机器部署了多个master

主从复制的注意事项

写能力有限

- 主从复制还是只有一台master，提供的写服务能力有限

master故障情况下：

- 如果是master无持久化，slave开启持久化来保留数据的场景，建议不要配置redis自动重启。
- 启动redis自动重启，master启动后，无备份数据，可能导致集群数据丢失的情况。

带有效期的key：

- slave不会让key过期，而是等待 master 让 key 过期
- 在Lua脚本执行期间，不执行任何 key 过期操作

02

客户端及监控

Java集群客户端

使用Lettuce客户端进行连接

见代码示例

Monitor命令

monitor 是一个调试命令，返回服务器处理的每个命令。对于发现程序的错误非常有用。

出于安全考虑，某些特殊管理命令CONFIG 不会记录到MONITOR输出

```
$ redis-cli monitor
1339518083.107412 [0 127.0.0.1:60866] "keys" "*"
1339518087.877697 [0 127.0.0.1:60866] "dbsize"
1339518090.420270 [0 127.0.0.1:60866] "set" "x" "6"
1339518096.506257 [0 127.0.0.1:60866] "get" "x"
1339518099.363765 [0 127.0.0.1:60866] "del" "x"
1339518100.544926 [0 127.0.0.1:60866] "get" "x"
```



运行一个MONITOR命令能够降低50%的吞吐量，运行多个MONITOR命令 降低的吞吐量更多。

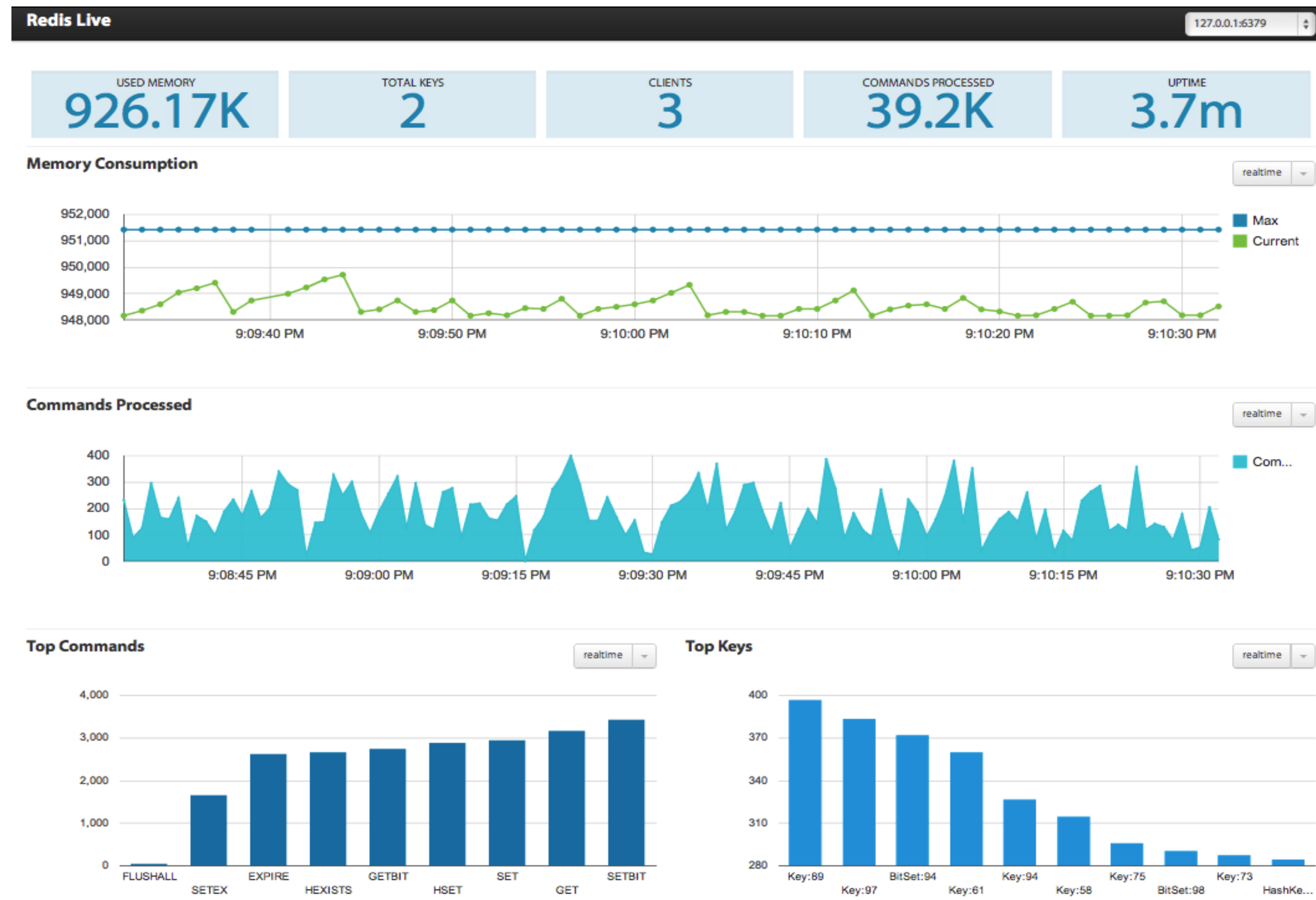
Info 命令

INFO命令以一种易于理解和阅读的格式，返回关于Redis服务器的各种信息和统计数值。

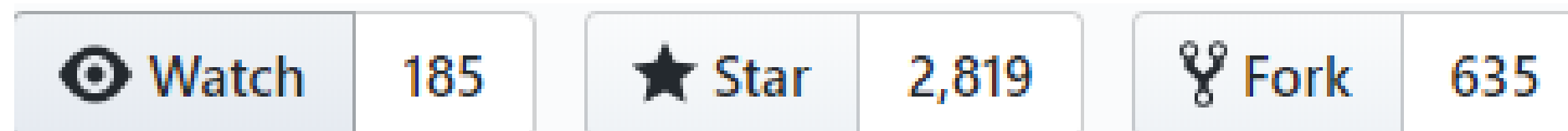
Info命令	返回信息
server	Redis服务器的一般信息
clients	客户端的连接部分
memory	内存消耗相关信息
persistence	持久化相关信息
stats	一般统计
replication	主/从复制信息
cpu	统计CPU的消耗
commandstats	Redis命令统计
cluster	Redis集群信息
keyspace	数据库的相关统计

可以通过section 返回部分信息，如果没有使用任何参数时，默认为default。

图形化监控工具 - RedisLive



<https://github.com/nkrode/RedisLive>



03

实现Redis读写分离

实现Redis读写分离

使用Lettuce客户端进行开发

见代码示例

04

总结

总结

谢谢观看