

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

Redis高性能底层探秘

- 如何支撑50ws请求

分布式高并发—缓存技术

目录

课程安排



01

限制性能的核心因素

IO、内存、CPU



02

从协议看高性能原因

简单高效



03

pipeline机制是如何支撑50w每秒的

彻底明白pipeline



04

redis中的多线程异步处理机制

Redis真的是单线程?

01

限制性能的核心因素

Redis的速度有多快？

```
$ ./redis-benchmark -r 1000000 -n 2000000 -t  
get, set, lpush, lpop -q  
SET: 122556.53 requests per second  
GET: 123601.76 requests per second  
LPUSH: 136752.14 requests per second  
LPOP: 132424.03 requests per second
```

英特尔 (R) Xeon (R) CPU E5520 @ 2.27GHz

```
$ ./redis-benchmark -r 1000000 -n 2000000 -t  
get, set, lpush, lpop -q  
SET: 35001.75 requests per second  
GET: 37481.26 requests per second  
LPUSH: 36968.58 requests per second  
LPOP: 35186.49 requests per second
```

云服务供应商 Linode 2048实例

性能瓶颈

□ 性能瓶颈 - 网络

1000Mb网卡的传输速度理论上是125MB/s

100Mb网卡的传输速度理论上是12.5MB/s

单位不一样，8Mb=1MB 100/8=

□ CPU型号

Intel、AMD、龙芯、申威、飞腾...

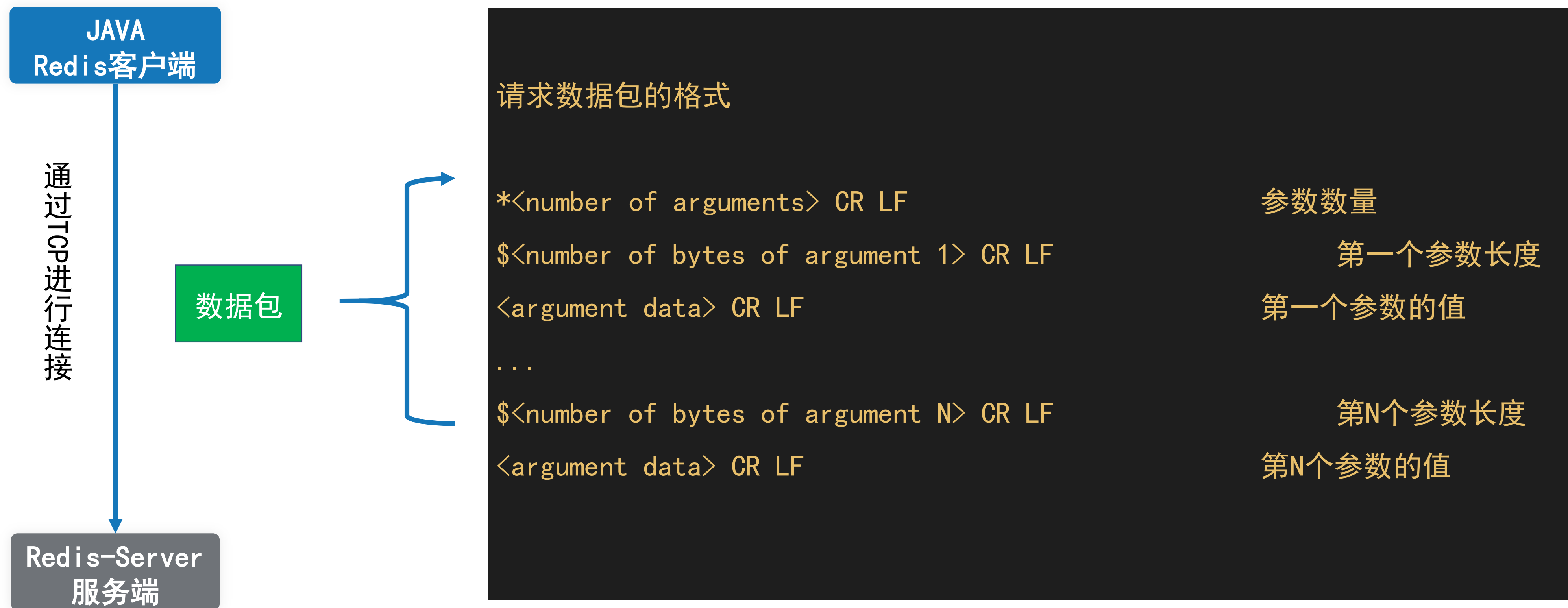
□ 性能瓶颈 - 内存操作

内存操作速度在纳秒级别，一般情况下，说的瓶颈，是指**存储瓶颈**

02

从协议看高性能原因

Redis通信协议 - RESP





协议格式

类型	前缀标志	场合	示例
简单字符串	+	最小的开销传输非二进制安全字符串	" +0K\r\n"
错误	-	在发生错误时发送，内容跟简单字符串相同	" -Error message\r\n"
整数	:	简单的返回一个整数的结果，也用于回复真或假	" :0\r\n"
批量字符串	\$	长度最大为512 MB的单个二进制安全字符串	" \$6\r\nfoobar\r\n"
数组	*	多个RESP批量字符串	" *2\r\n\$3\r\nfoo\r\n\$3\r\nbar\r\n"

命令每个部分都以” \r\n”（CRLF）结束

03

pipeline机制是如何支撑50w每秒的

管道机制 - pipeline

Client: INCR X

Server: 1

Client: INCR X

Server: 2

Client: INCR X

Server: 3

Client: INCR X

Server: 4

pipeline方式

Client: INCR X

Client: INCR X

Client: INCR X

Client: INCR X

Server: 1

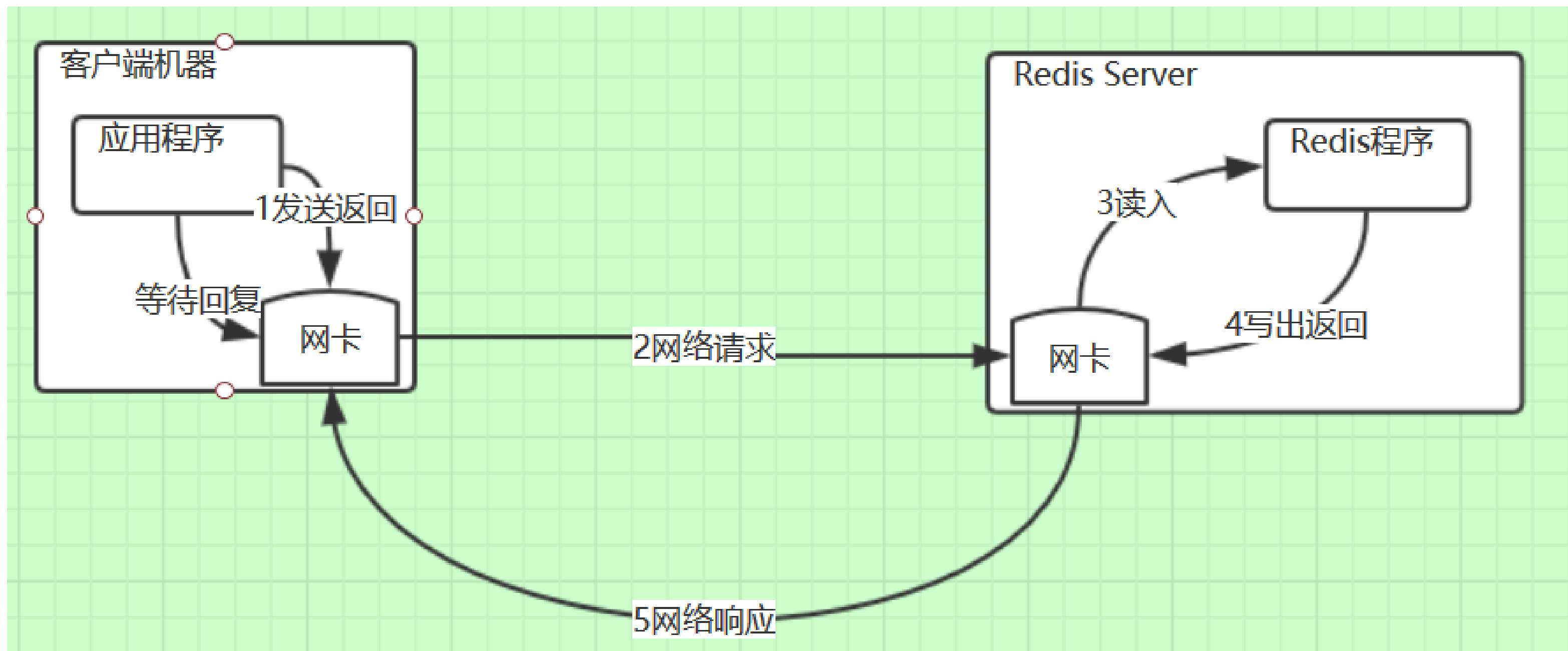
Server: 2

Server: 3

Server: 4

多个命令发送到服务器而无需等待回复，即使客户端尚未读取旧响应，服务端也能够处理新请求。

请求流程



04

redis中的多线程异步处理机制

Redis中的其他线程

- ❑ 持久化机制 AOF （RDB的方式属于fork子进程, 非线程）
- ❑ 主从之间数据异步同步 - PSYNC
- ❑ 关闭时将内存数据刷盘到文件

❑ lazyfree机制（4.0+特性）：

UNLINK：异步删除key的命令。

FLUSHDB ASYNC：异步清除当前的DB命令。

FLUSHALL ASYNC：异步清除所有数据库的命令。

配置以下内存释放为非阻塞的方式

lazyfree-lazy-eviction no 内存不够用，redis自动回收时

lazyfree-lazy-expire no 过期key清除时

lazyfree-lazy-server-del no 用户主动删除时

replica-lazy-flush no 主从同步中，副本接收完数据，清空数据时

线上环境谨慎操作

- 命令的复杂度；
- 数据清理操作；
- 持久化机制的选择；

谢谢观看