第八周作业

1.测试不同value大小下的redis get set性能：

10字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 10
====== SET ======
  100000 requests completed in 0.58 seconds
  50 parallel clients
  10 bytes payload
  keep alive: 1

99.88% <= 1 milliseconds
100.00% <= 1 milliseconds
172711.58 requests per second

====== GET ======
  100000 requests completed in 0.60 seconds
  50 parallel clients
  10 bytes payload
  keep alive: 1

99.82% <= 1 milliseconds
100.00% <= 1 milliseconds
166112.95 requests per second
```

20字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 20
====== SET ======
  100000 requests completed in 0.56 seconds
  50 parallel clients
  20 bytes payload
  keep alive: 1

99.93% <= 1 milliseconds
100.00% <= 1 milliseconds
179211.45 requests per second

====== GET ======
  100000 requests completed in 0.57 seconds
  50 parallel clients
  20 bytes payload
  keep alive: 1

99.86% <= 1 milliseconds
100.00% <= 1 milliseconds
176366.86 requests per second
```

50字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 50
====== SET ======
  100000 requests completed in 0.57 seconds
  50 parallel clients
  50 bytes payload
  keep alive: 1

99.96% <= 1 milliseconds
100.00% <= 1 milliseconds
175746.92 requests per second

====== GET ======
  100000 requests completed in 0.57 seconds
  50 parallel clients
  50 bytes payload
  keep alive: 1

99.88% <= 1 milliseconds
100.00% <= 1 milliseconds
173913.05 requests per second
```

100字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 100
====== SET ======
  100000 requests completed in 0.55 seconds
  50 parallel clients
  100 bytes payload
  keep alive: 1

99.90% <= 1 milliseconds
100.00% <= 1 milliseconds
182815.36 requests per second

====== GET ======
  100000 requests completed in 0.58 seconds
  50 parallel clients
  100 bytes payload
  keep alive: 1

99.87% <= 1 milliseconds
100.00% <= 1 milliseconds
172117.05 requests per second
```

200字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 200
====== SET ======
  100000 requests completed in 0.56 seconds
  50 parallel clients
  200 bytes payload
  keep alive: 1

99.91% <= 1 milliseconds
100.00% <= 1 milliseconds
179856.11 requests per second

====== GET ======
  100000 requests completed in 0.52 seconds
  50 parallel clients
  200 bytes payload
  keep alive: 1

99.85% <= 1 milliseconds
100.00% <= 1 milliseconds
191204.59 requests per second
```

1k字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 1000
====== SET ======
  100000 requests completed in 0.58 seconds
  50 parallel clients
  1000 bytes payload
  keep alive: 1

99.87% <= 1 milliseconds
99.98% <= 2 milliseconds
100.00% <= 2 milliseconds
171232.88 requests per second

====== GET ======
  100000 requests completed in 0.50 seconds
  50 parallel clients
  1000 bytes payload
  keep alive: 1

99.88% <= 1 milliseconds
100.00% <= 1 milliseconds
198807.16 requests per second
```

5k字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t get,set -n 100000 -d 5000
====== SET ======
  100000 requests completed in 0.60 seconds
  50 parallel clients
  5000 bytes payload
  keep alive: 1

99.88% <= 1 milliseconds
100.00% <= 1 milliseconds
167785.23 requests per second

====== GET ======
  100000 requests completed in 0.50 seconds
  50 parallel clients
  5000 bytes payload
  keep alive: 1

99.90% <= 1 milliseconds
100.00% <= 1 milliseconds
199600.80 requests per second
```

2.分析不同value大小下，平均每个key的占用内存空间：

每次写入数据前先清空数据库，清空后的memory信息如下：

```
127.0.0.1:6379> flushdb
OK
127.0.0.1:6379> info memory
# Memory
used_memory:858560
used_memory_human:838.44K
```

10字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 10 -r 100000
====== SET ======
  100000 requests completed in 0.57 seconds
  50 parallel clients
  10 bytes payload
  keep alive: 1

99.86% <= 1 milliseconds
99.98% <= 2 milliseconds
100.00% <= 2 milliseconds
176678.45 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:7449184
used_memory_human:7.10M
```

平均每个key占用10字节。

20字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 20 -r 100000
====== SET ======
  100000 requests completed in 0.55 seconds
  50 parallel clients
  20 bytes payload
  keep alive: 1

99.88% <= 1 milliseconds
100.00% <= 1 milliseconds
182149.36 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:8442720
used_memory_human:8.05M
```

平均每个key占用20字节。

50字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 50 -r 100000
====== SET ======
  100000 requests completed in 0.53 seconds
  50 parallel clients
  50 bytes payload
  keep alive: 1

99.85% <= 1 milliseconds
100.00% <= 1 milliseconds
187969.92 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:10481984
used_memory_human:10.00M
```

平均每个key占用50字节。

100字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 100 -r 100000
====== SET ======
  100000 requests completed in 0.53 seconds
  50 parallel clients
  100 bytes payload
  keep alive: 1

99.72% <= 1 milliseconds
100.00% <= 1 milliseconds
189393.94 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:13522576
used_memory_human:12.90M
```

平均每个key占用100字节。

200字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 200 -r 100000
====== SET ======
  100000 requests completed in 0.55 seconds
  50 parallel clients
  200 bytes payload
  keep alive: 1

99.85% <= 1 milliseconds
100.00% <= 1 milliseconds
182149.36 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:20662320
used_memory_human:19.71M
```

平均每个key占用200字节。

1k字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 1000 -r 100000
====== SET ======
  100000 requests completed in 0.56 seconds
  50 parallel clients
  1000 bytes payload
  keep alive: 1

99.82% <= 1 milliseconds
100.00% <= 1 milliseconds
176991.16 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:71186336
used_memory_human:67.89M
```

平均每个key占用1k字节。

5k字节：

```
zhi@zhi-Yoga-14s:~$ redis-benchmark -t set -n 100000 -d 5000 -r 100000
====== SET ======
  100000 requests completed in 0.58 seconds
  50 parallel clients
  5000 bytes payload
  keep alive: 1

99.77% <= 1 milliseconds
99.94% <= 2 milliseconds
99.95% <= 8 milliseconds
99.97% <= 9 milliseconds
100.00% <= 9 milliseconds
171232.88 requests per second
```

```
127.0.0.1:6379> info memory
# Memory
used_memory:329654704
used_memory_human:314.38M
```

平均每个key占用5k字节。