

# 1. 存储性能估算

1. 存储量：假设每门学科每年2次考试，每个学生平均一学期20门课，考试采取机考的方式，每门考试的答案20判断题、20选择题、4道大题(答案200字以内)，考试结果永久保存，在校学生能够看到自己曾经的考试结果,则考试结果记录的存储量为：
  1. 在校学生:  $1000\text{万} \times 20(\text{课}) \times 2(\text{考试次数}) \times 1000(\text{答案}) \times 2(\text{学期}) \times 3(\text{只有前三年考试}) = 2.4\text{T}$ 。
  2. 离校学生: 每年250万，存储量为 0.6T。
2. 请求量：假设学校的考试都安排在某一个月內，考试的时候请求试卷，提交答案，中间答题过程浏览器本地完成，由于考试集中在上午4小时和下午4小时，且请求试卷集中在考试开始的前1分钟，提交答案集中在考试结束前的30分钟，因此：
  1. 考试开始前1分钟，请求试卷:  $1000\text{万} \times 20(\text{课}) / 20(\text{周末不考试}) / 4(\text{每天4堂考试}) / 1\text{分钟} = 250\text{万请求/分钟} \approx 5\text{万/每秒}$ 。
  2. 考试结束前30分钟，提交试卷:  $1000\text{万} \times 20(\text{课}) / 20(\text{周末不考试}) / 4(\text{每天4堂考试}) / 30\text{分钟} = 1700/\text{每秒}$ 。

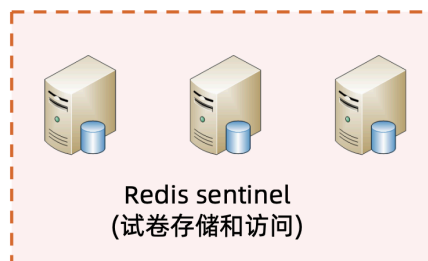
## 【总结】

1. 在校学生考试结果存储: 2.4T
2. 离校学生考试结果存储: 每年增长0.6T
3. 试卷请求 QPS: 5万/s
4. 提交试卷 TPS: 1700/s。

# 2. 存储系统选择

由于通常能买到的服务器的存储量是1T到2T，这里每一届学生的考试结果就有2.4T，所以不能单机存储所有数据，不需要分区部署，所以使用分片架构。

按照不同的场景，分别使用 Redis Sentinel、MySQL 和 HBase 来存。



# 3. 存储方案设计

## 3.1 Redis 存储方案设计

参加考试的同学，获取的试卷都是一样的，Redis sentinel 支持主从，所以从从机上读试卷也是一样的，这样的话可以实现高性能（5万 QPS）。

#### 【数据结构设计】

- Key: 学校ID + 考试ID
- Value: String

#### 【读写分析】

可以预先在后台把试卷加载到 Redis，这样学生在访问的时候就直接通过 学校ID + 考试ID 去访问Redis，从而获取试卷。提交的话是往HBase里提交（课上有讲）。

#### 【具体部署】

这里 Redis 主要要承担的是大批量的 GET 操作，根据 benchmark 结果，最搓的GET并发数也有3.7万（Linode 2048 instance (without pipelining)），所以，可以使用一个 Master Node + 一个 Slave Node，2 Node都能负责应对读请求。