

规模控制

如何开发知晓工具榜第一的小程序



黎博



罗盘
2012



微趋势 \ CircoQ \ OT \ Tracker
2014



华尔兹
2013



问卷
2015 - Now

校招加入CDC
2012

排序

[86, 64, 36, 52, 99]

[36, 52, 64, 86, 99]

排序

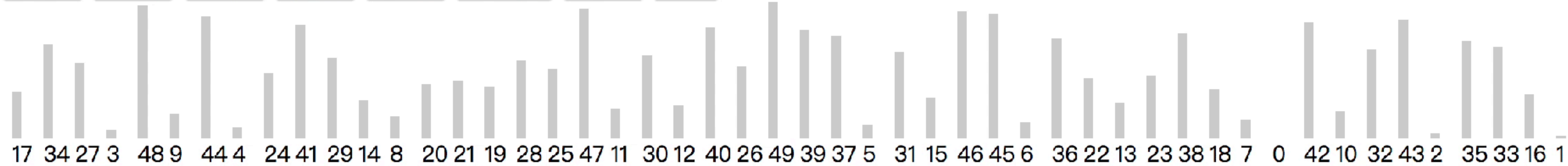
排序算法动画演示

动画速度：

数组长度： 随即数组

待排序数组：

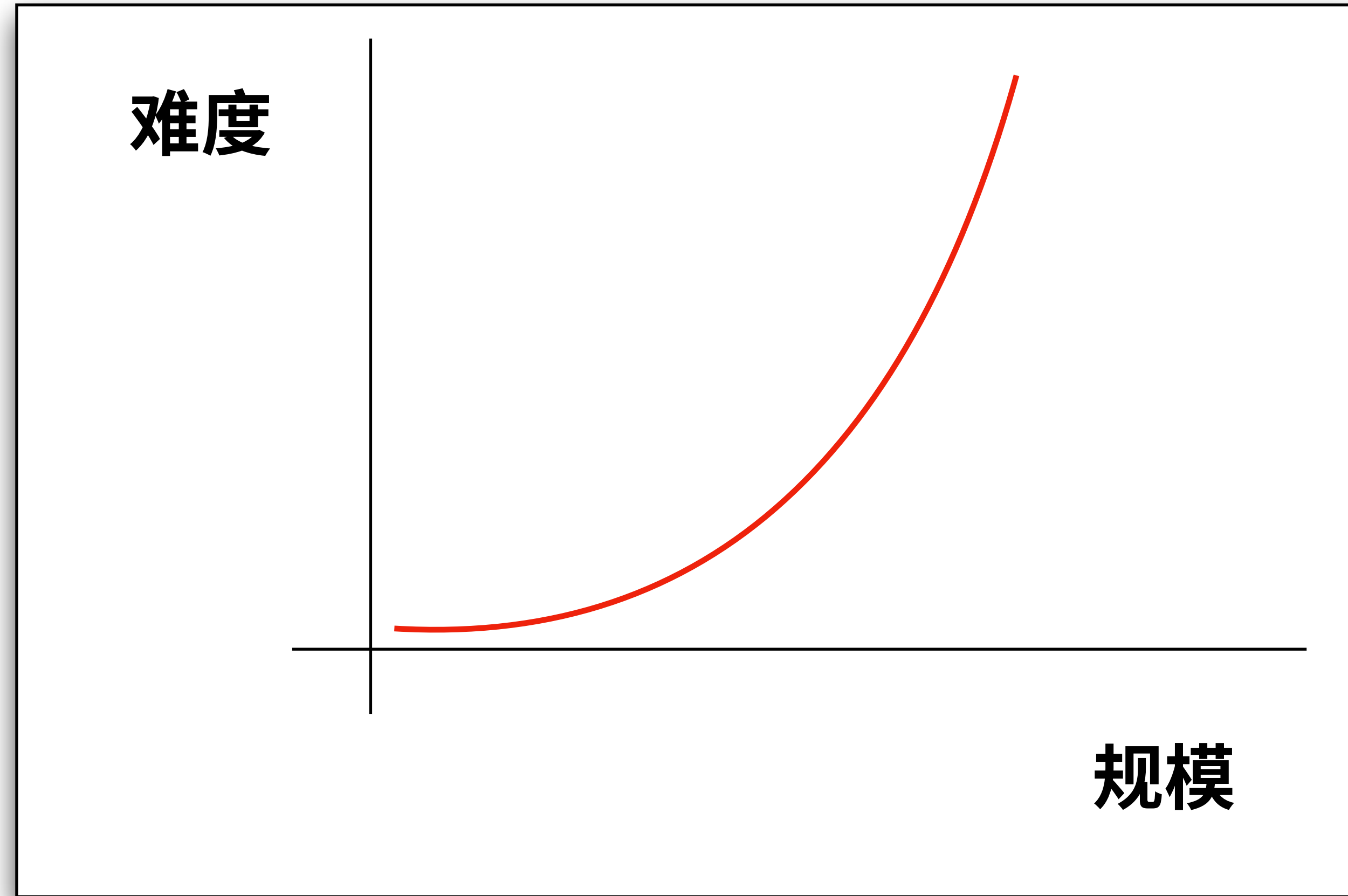
冒泡排序 快速排序 插入排序 希尔排序 选择排序 鸡尾酒排序 奇偶排序 梳排序



排序

Top Results

	Daytona	Indy
Gray	<p>2016, 44.8 TB/min</p> <p>Tencent Sort</p> <p>100 TB in 134 Seconds</p> <p>512 nodes x (2 OpenPOWER 10-core POWER8 2.926 GHz, 512 GB memory, 4x Huawei ES3600P V3 1.2TB NVMe SSD, 100Gb Mellanox ConnectX4-EN)</p> <p>Jie Jiang, Lixiong Zheng, Junfeng Pu, Xiong Cheng, Chongqing Zhao</p> <p>Tencent Corporation</p> <p>Mark R. Nutter, Jeremy D. Schaub</p>	<p>2016, 60.7 TB/min</p> <p>Tencent Sort</p> <p>100 TB in 98.8 Seconds</p> <p>512 nodes x (2 OpenPOWER 10-core POWER8 2.926 GHz, 512 GB memory, 4x Huawei ES3600P V3 1.2TB NVMe SSD, 100Gb Mellanox ConnectX4-EN)</p> <p>Jie Jiang, Lixiong Zheng, Junfeng Pu, Xiong Cheng, Chongqing Zhao</p> <p>Tencent Corporation</p> <p>Mark R. Nutter, Jeremy D. Schaub</p>
Cloud	<p>2016, \$1.44 / TB</p> <p>NADSort</p> <p>100 TB for \$144</p> <p>394 Alibaba Cloud ECS ecs.n1.large nodes x (Haswell E5-2680 v3, 8 GB memory, 40GB Ultra Cloud Disk, 4x 135GB SSD Cloud Disk)</p> <p>Qian Wang, Rong Gu, Yihua Huang</p> <p>Nanjing University</p> <p>Reynold Xin</p> <p>Databricks Inc.</p> <p>Wei Wu, Jun Song, Junluan Xia</p> <p>Alibaba Group Inc.</p>	<p>2016, \$1.44 / TB</p> <p>NADSort</p> <p>100 TB for \$144</p> <p>394 Alibaba Cloud ECS ecs.n1.large nodes x (Haswell E5-2680 v3, 8 GB memory, 40GB Ultra Cloud Disk, 4x 135GB SSD Cloud Disk)</p> <p>Qian Wang, Rong Gu, Yihua Huang</p> <p>Nanjing University</p> <p>Reynold Xin</p> <p>Databricks Inc.</p> <p>Wei Wu, Jun Song, Junluan Xia</p> <p>Alibaba Group Inc.</p>
Minute	<p>2016, 37 TB</p> <p>Tencent Sort</p> <p>512 nodes x (2 OpenPOWER 10-core POWER8 2.926 GHz, 512 GB memory, 4x Huawei ES3600P V3 1.2TB NVMe SSD, 100Gb Mellanox ConnectX4-EN)</p> <p>Jie Jiang, Lixiong Zheng, Junfeng Pu, Xiong Cheng, Chongqing Zhao</p> <p>Tencent Corporation</p> <p>Mark R. Nutter, Jeremy D. Schaub</p>	<p>2016, 55 TB</p> <p>Tencent Sort</p> <p>512 nodes x (2 OpenPOWER 10-core POWER8 2.926 GHz, 512 GB memory, 4x Huawei ES3600P V3 1.2TB NVMe SSD, 100Gb Mellanox ConnectX4-EN)</p> <p>Jie Jiang, Lixiong Zheng, Junfeng Pu, Xiong Cheng, Chongqing Zhao</p> <p>Tencent Corporation</p> <p>Mark R. Nutter, Jeremy D. Schaub</p>



问题的难度，随着规模增长而增长

投票

「举手」

「写在纸上」

「电话投票」

「Web端的勾选提交」

「陶片驱逐法」

小程序里的投票

- 16年1月11日，小程序公布
- WebSocket - 实时投票 - 提升体验



实时化方案的演变

第一代 HTTP Poll



- 客户端：「拿起电话」有新消息吗？

- 服务端：没有新消息。「挂电话」

请求

响应

等待

- 服务端：有新消息了，电话怎么还没响？

第二代 HTTP Long-Poll

- 客户端：「拿起电话」有新消息吗？

- 服务端：「鸦雀无声」

请求

响应

等待

- 服务端：有新消息了，bababa。「挂电话」

- 客户端：「拿起电话」有新消息吗？

第三代 HTTP Stream

- 客户端：「拿起电话」有新消息吗？

- 服务端：「鸦雀无声」

请求

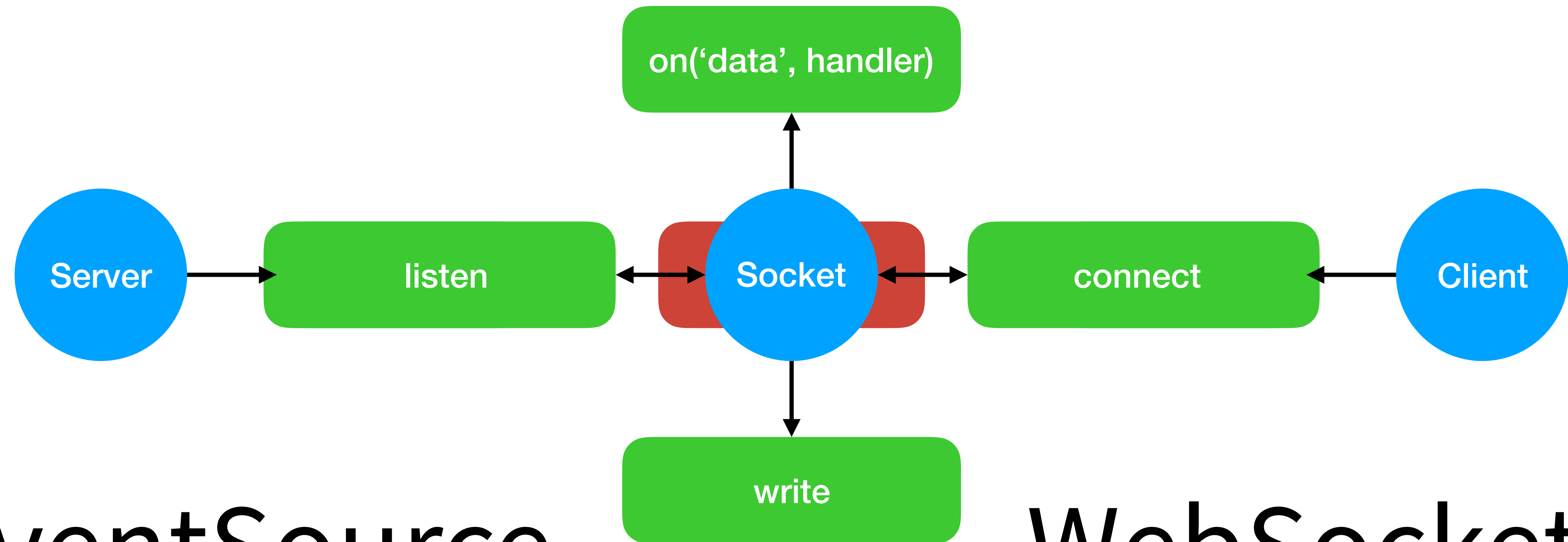
响应

等待

- 服务端：有新消息了，bababa。

- 服务端：有新消息了，bababa。

现代实时方案



EventSource

WebSocket

小程序的 WebSocket

- 启停
 - wx.connectSocket
 - wx.closeSocket
- 收发消息
 - wx.sendSocketMessage
 - wx.onSocketMessage
- 异步事件
 - wx.onSocketOpen
 - wx.onSocketClose
 - wx.onSocketError

再也不用微信举手了



用户实际使用场景

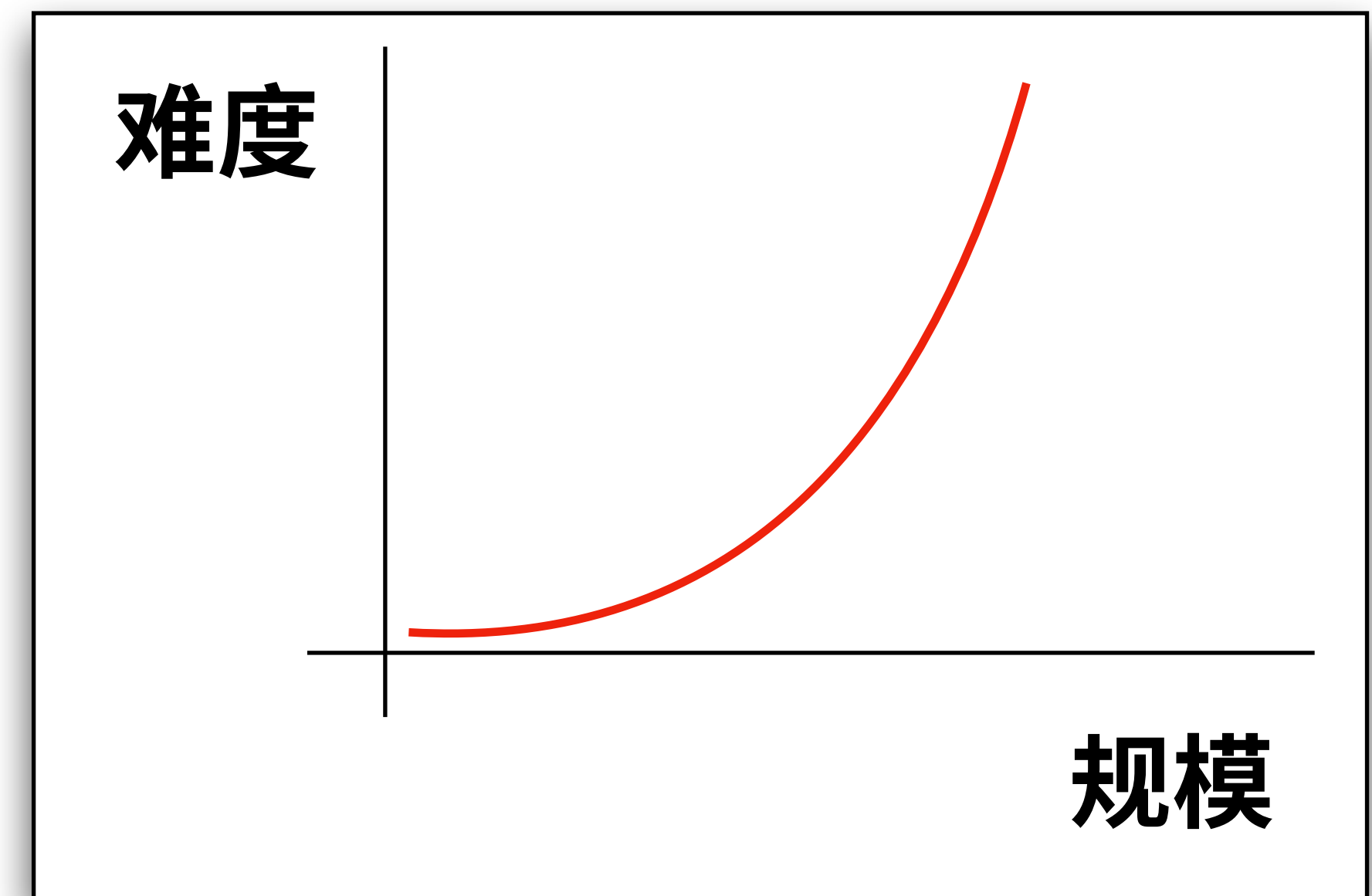
- 几个人投票吃什么
- 几十人投票去哪团建
- 全校优秀班级评选



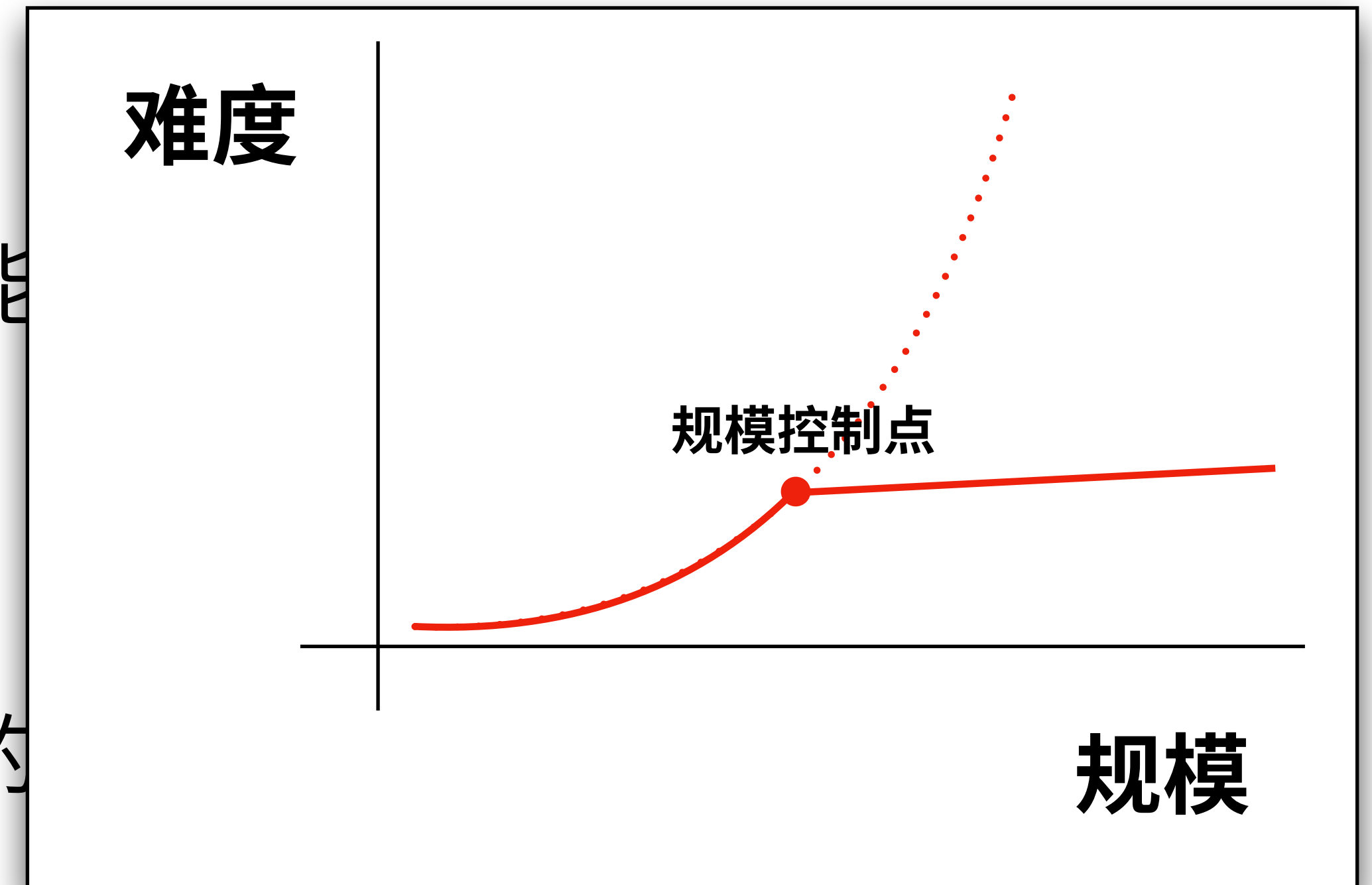
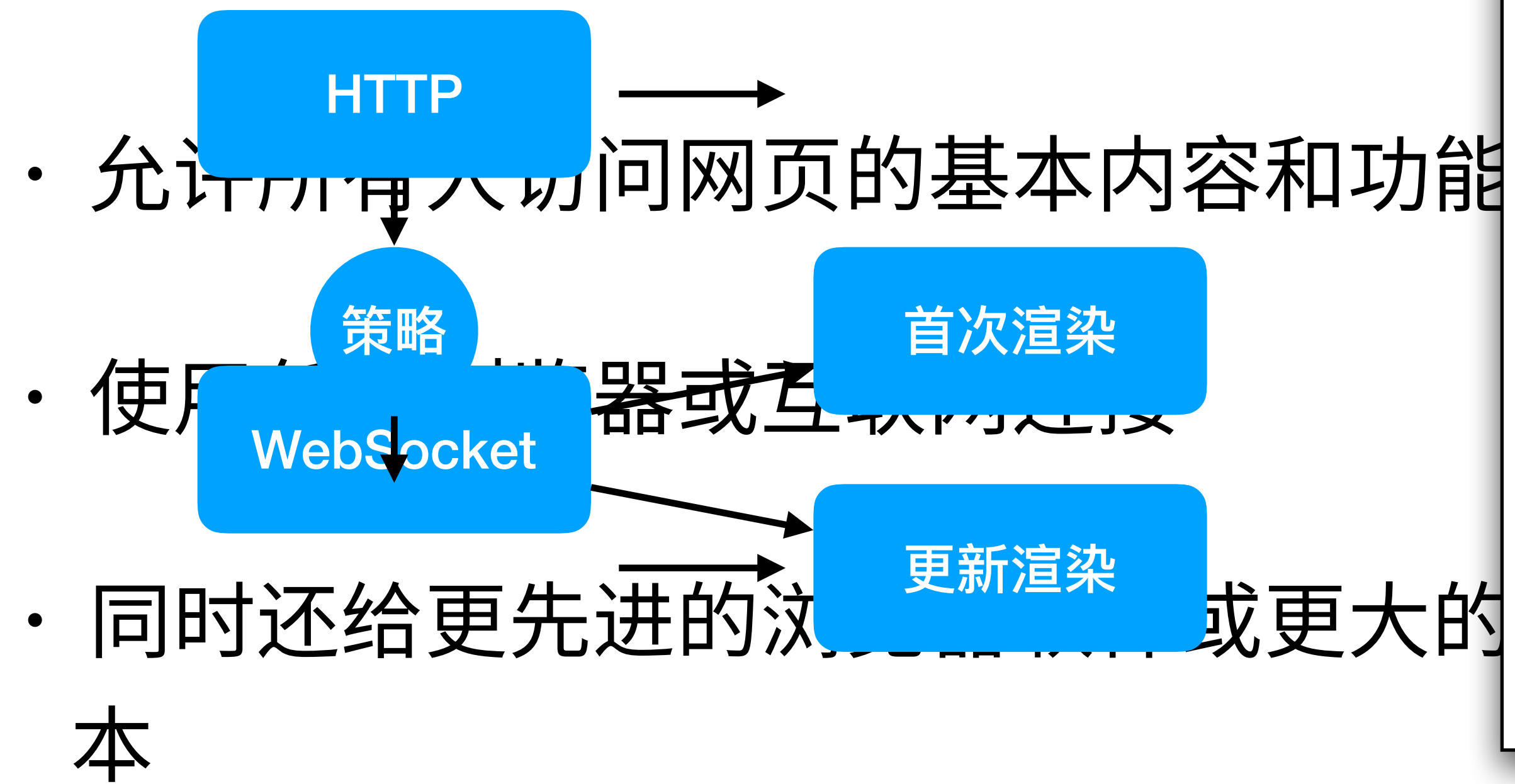
居然还允许家长拉票 🤔

实时多人在线系统的难点

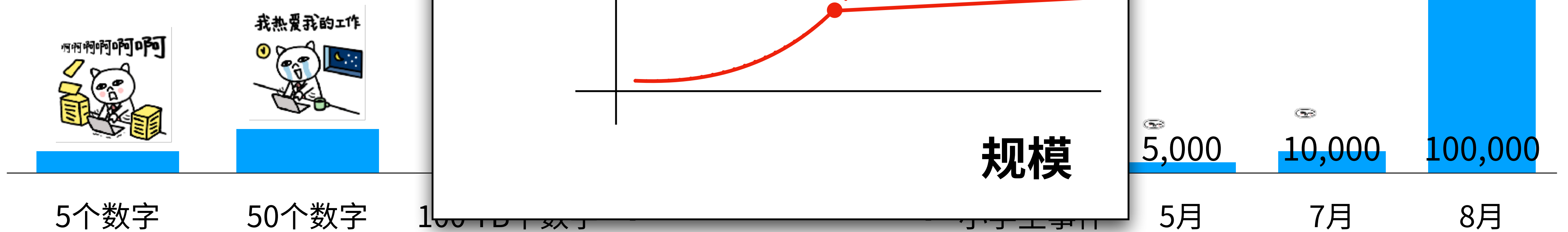
- 用户的每个操作，都要同步给其他用户
- PUBG 最大100人
- 微信群最大500人
- 投票，没有限制🤔



渐进增强



排序 投票



规模控制

小程序规模控制 实践

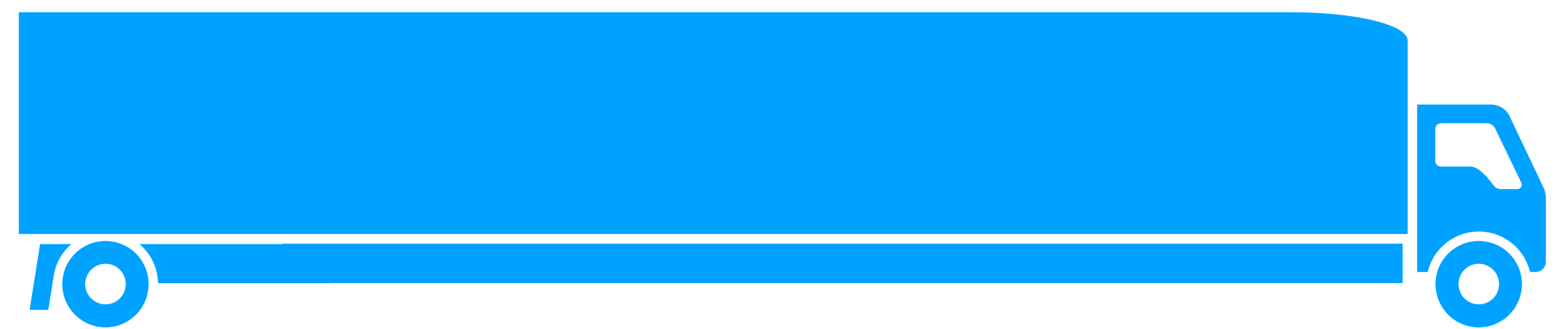
多人实时

- 首屏优化，渐进增强
- 关注 `wx.onSocketMessage` 的调用频率



渲染巨型对象

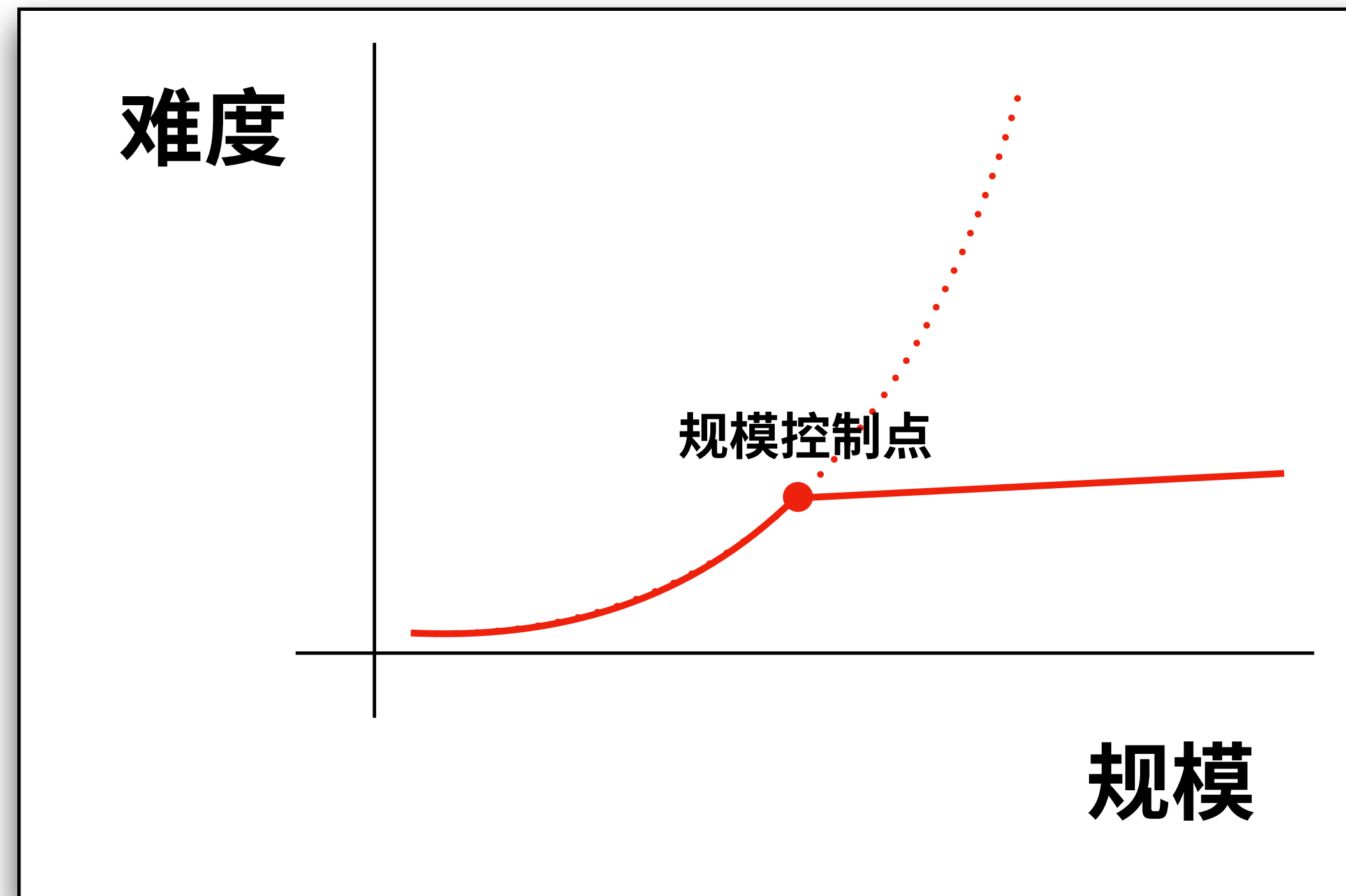
- 关注 setData 的传入对象的大小
- 分多次调用 setData



长度未知的遍历

- 限制最大值
- 分段遍历





祝大家都能收获爆款小程序

Thanks & FAQ

