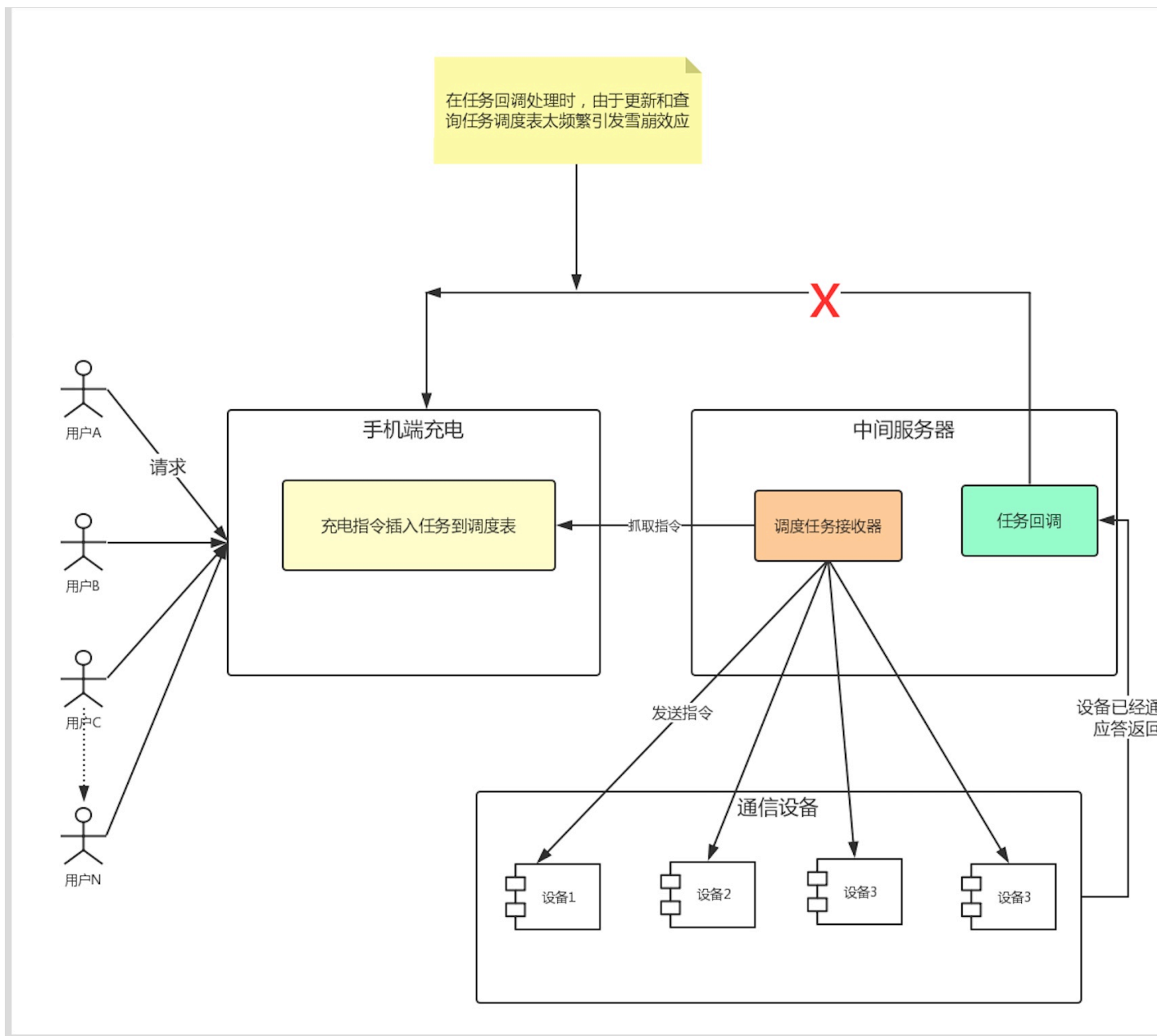


事故的发生

2018年7月6日，晚8点左右，学校充电保出现大面积的服务宕掉，影响面很大，影响的充电插座有1300个左右，此事故持续了两天。给学校的同学老师，公司的领导带了很大的不便，研发部在此表示深感歉意。

事故原因

由于公司业务日趋壮大，以前隐藏的问题也逐步暴露，而且预估这个过程还会延续。本次充电保充电失败的原因是由于，充电任务调度临时表没有一个垃圾回收机制，导致了任务数据越积越多。同时这个任务表的数据读写很频繁，在并发来的时候，这个数据表出现了锁表。锁表会引起请求超时，同时有没有给用户相应提示。用户在使用过程又进行点击，随着点击量的增大，对后端数据库造成极大的冲击，甚至导致“雪崩”现象(请求像滚雪球一样，越滚越大)，最后是服务瘫痪。

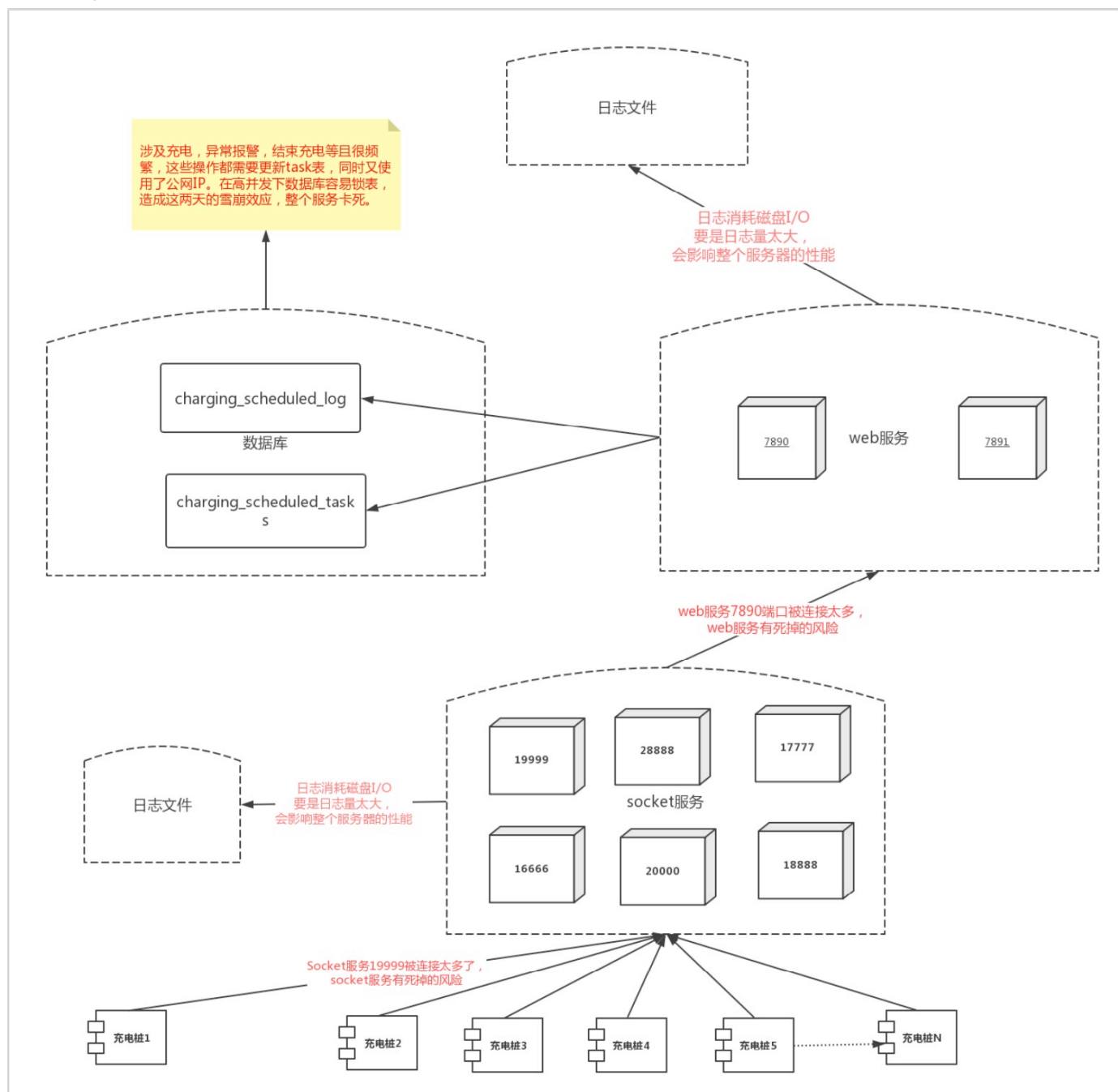


总结分析

通过对充电宝的整体架构进行排查分析，发现除了数据库之外还存在的一些列的隐患，隐患如下：

1. 任务调度接收器的web服务端口7890，连接连接了所有学校的充电保、洗衣，如果有高并发会引起服务不可用。
2. 任务回调的socket服务端口19999，连接了所有充电保所有设备，如果socket服务端口阻塞也会引起服务不可用。这个在电控中已经存在了这样的问题，据统计先在学校充电桩连接这个端口有个好几百个个，也已经到了封顶。
3. 任务调度表的数据没有做定期清理，建议3天做一次清理。

4. 在程序中记录日志比较频繁，也是对磁盘的一个大开销，也是建议3天做一次清理。
5. 原先的架构充电宝是部署在10服务器，这是一台综合业务服务器，包含了公司的所有的GC手机端，后台的业务。如果充电宝其中有一个问题就影响10服务器的使用。



系统优化方案

1. 以后新增的充电宝、洗衣业务不能部署在10正式服了，而是单独部署到新的的服务器上去。
2. 对于10服务器，对所有热门业务订餐、充电宝、智能充电进行性能分析，对大表进行水平分表。对临时中间表进行定时清理。具体的数据表已经由研发各小组去进行梳理。
3. socket服务和web服务的端口，采取一个学校一组端口，一套代码的方式进行处理。这样子

出问题的影响面也比较小。