

# web 构架进化概述

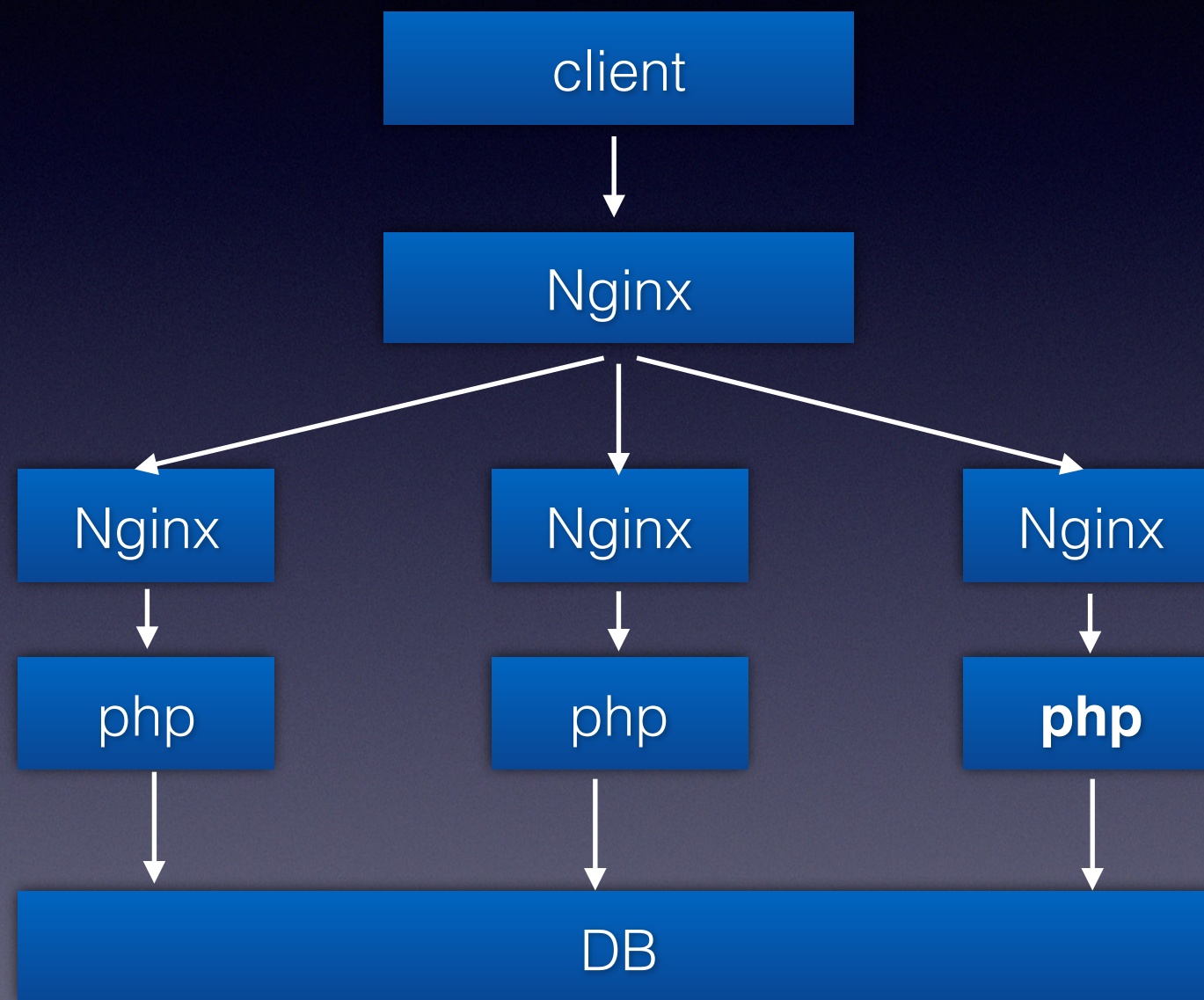
17G



自从数据库面世那一刻  
起，整个软件界就duang  
的一下不一样了！！！！



# php+db构架





# php+db构架的缺点

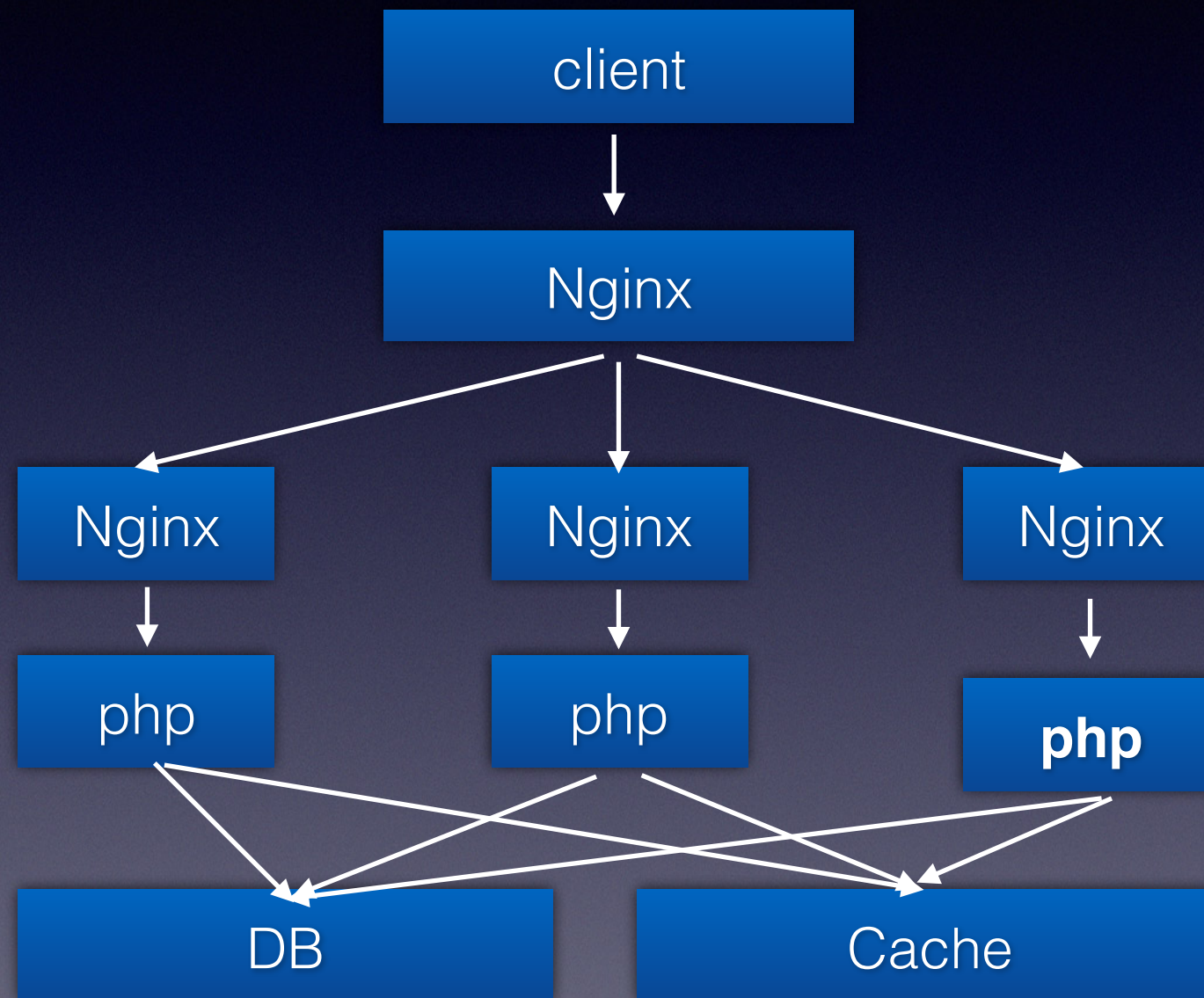
1. 相对大并发下，DB压力会随之增加
2. 进一步导致DB查询性能的降低
3. 进一步增加php进程的处理耗时
4. 进一步减少php进程的可用数量
5. 进一步增加请求的等待时间
6. ...
7. 服务被压死



缓存的出现，拯救了整个世界，  
从此db找到了自己的另一半，  
db&cache过上了幸福的生活



# php + db + cache构架





# 微信开始抢红包了，怎么办？

2015年除夕当日微信红包总量：10.1亿次；

2015年除夕20：00 - 00：48，微信摇一摇总量110亿次；

瞬间并发50Wqps；

1. 目前qb机器32台，每秒可以处理5000请求。按此请求要能达到微信的抗压能力，需要3200台机器。



# UI集群宕机了，怎么办？

php2X集群因为conf配置错误导致机器宕机1小时。

20:00

由于对机器实行各种操作均未见效果，rd经理[REDACTED]口头确认可以重启机器，op经理[REDACTED]电话确认可重启机器。

op [REDACTED]、op [REDACTED]对剩余未恢复服务执行重启操作。

20:07

op [REDACTED]、op [REDACTED]评估压力后，将所有流量切入恢复的三台机器，此时止损完成。

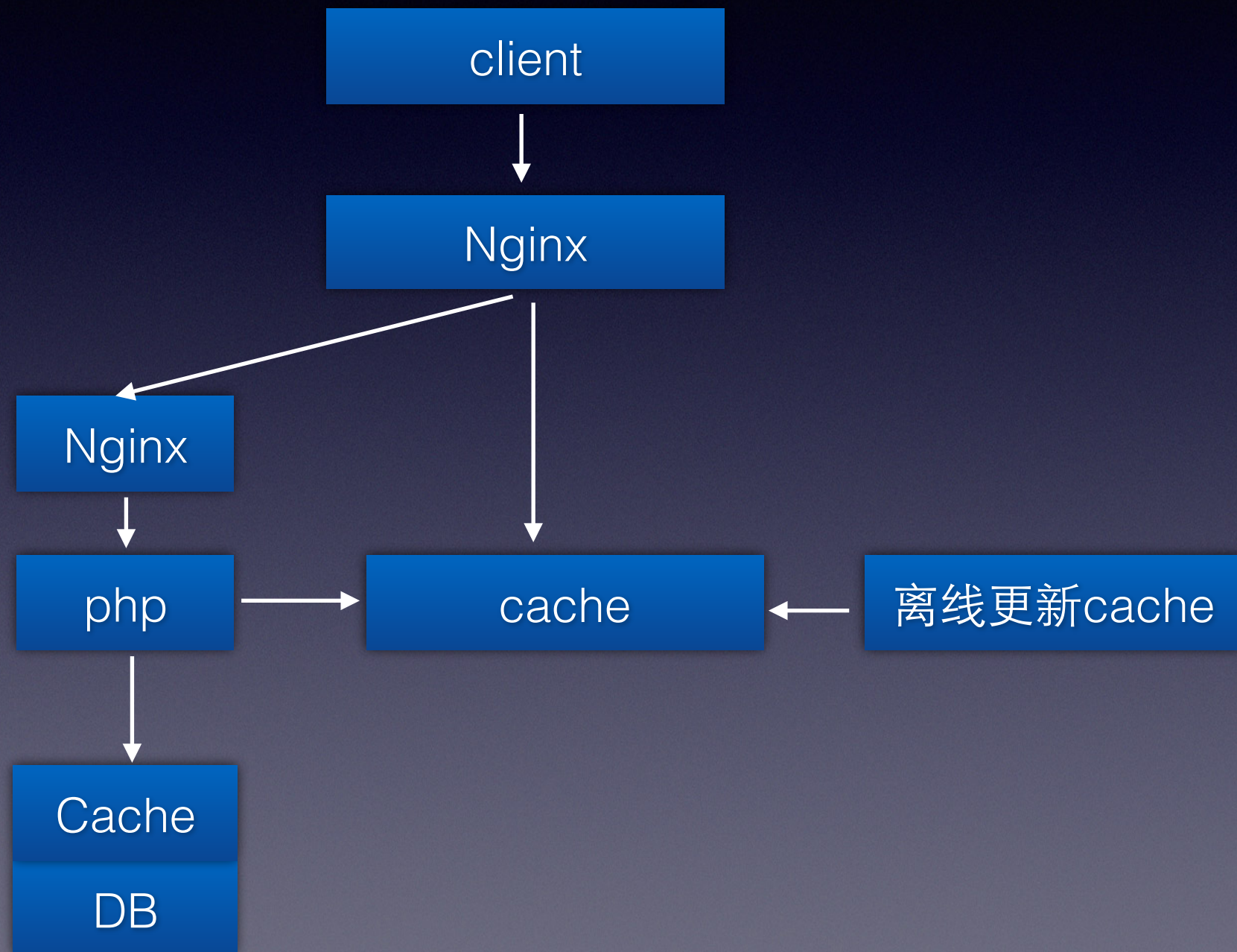
20:20

所有机器恢复服务，流量切回。

op [REDACTED] review所有配置、cms平台、日志切分流量后，确认跟进结束。随后在群中通报跟进结束和服务恢复时间点。



# nginx + cache构架





# cache觅新欢，梅开二度

1. 飞一般的感觉。并发几万qps不再是梦。
2. 就算是UI机器宕机了，还能提供服务，就是这么牛。



# 新构架需要考虑的事情

1. 海量并发下如何做好自我保护;
2. 如何降低构架成本以及平行扩展;
3. 如何做好灾备;
4. 如何使用cache;