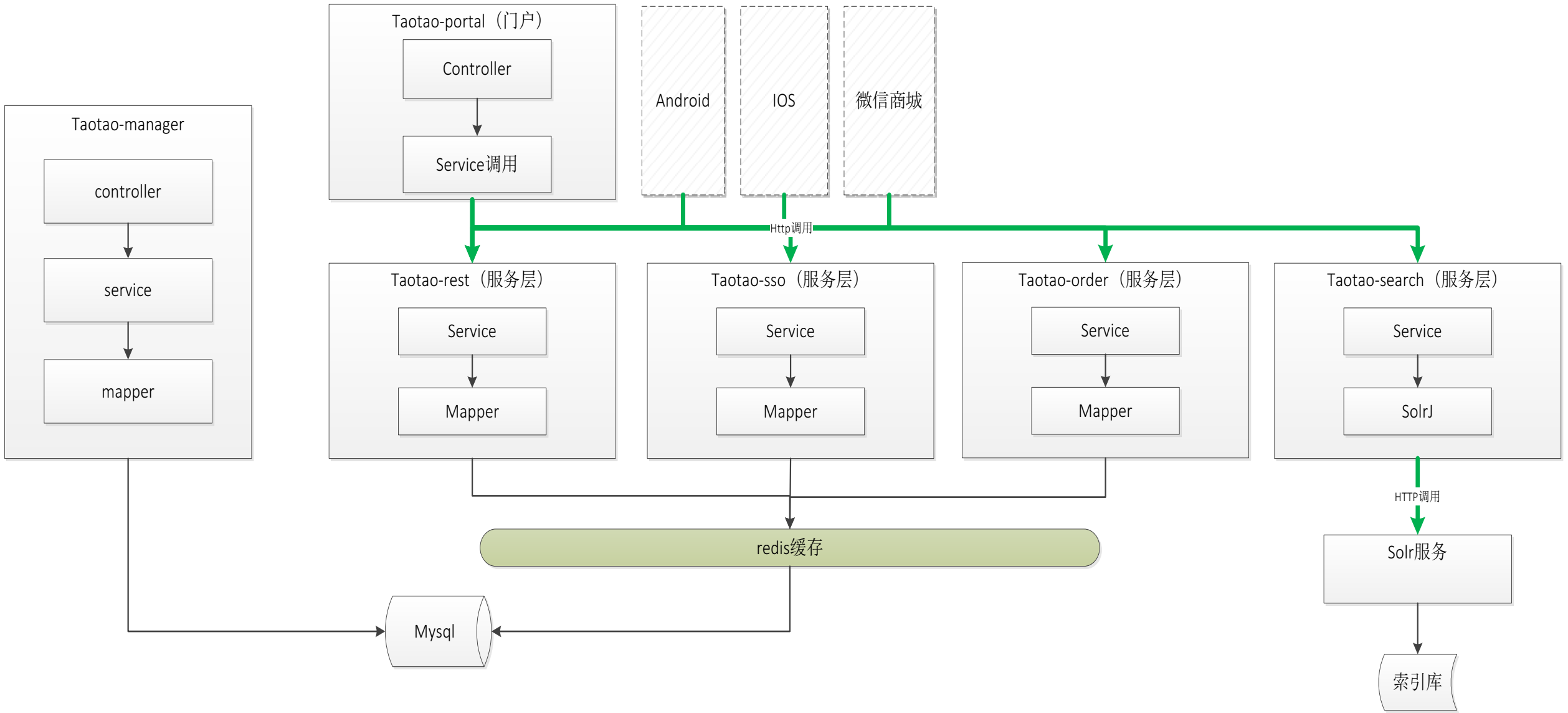
Web1.0

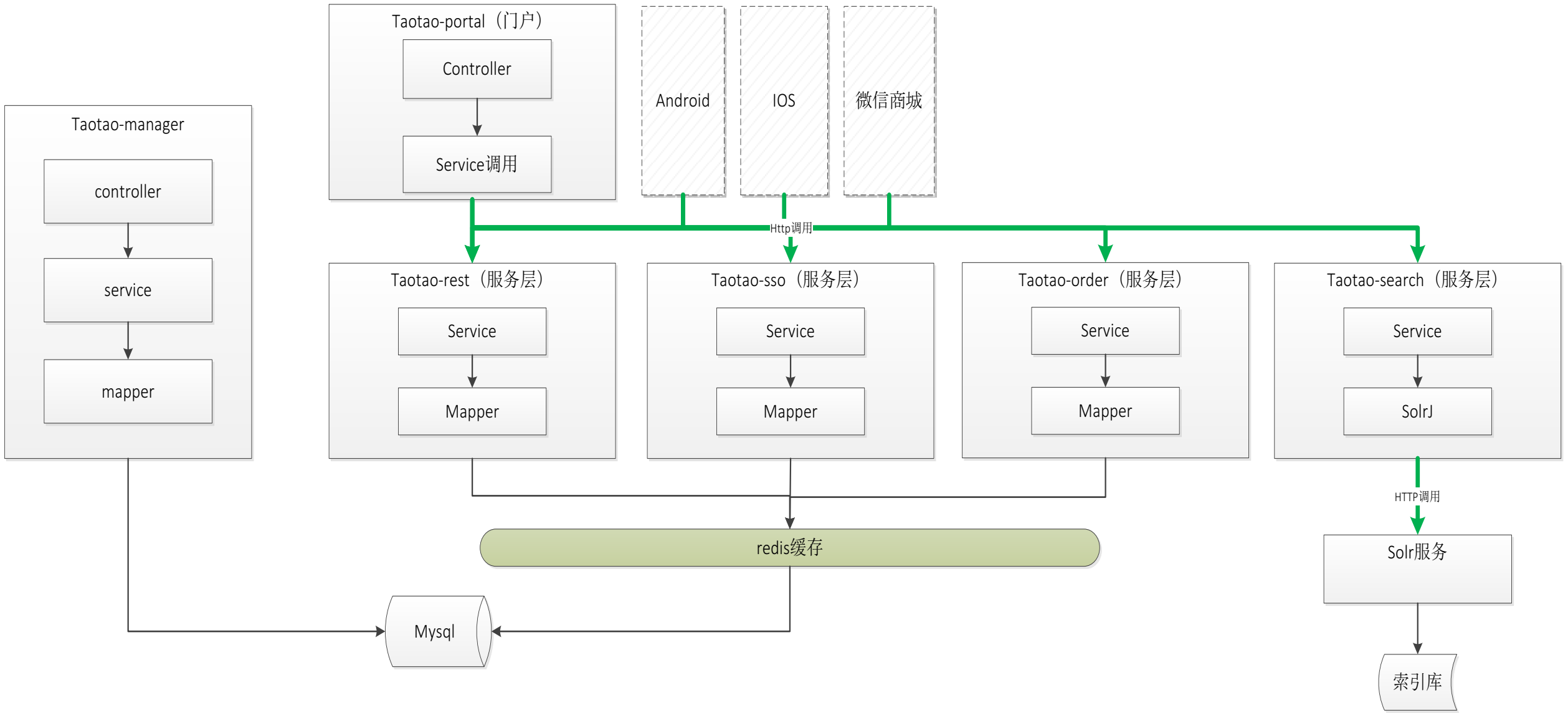
购物商城（中型）

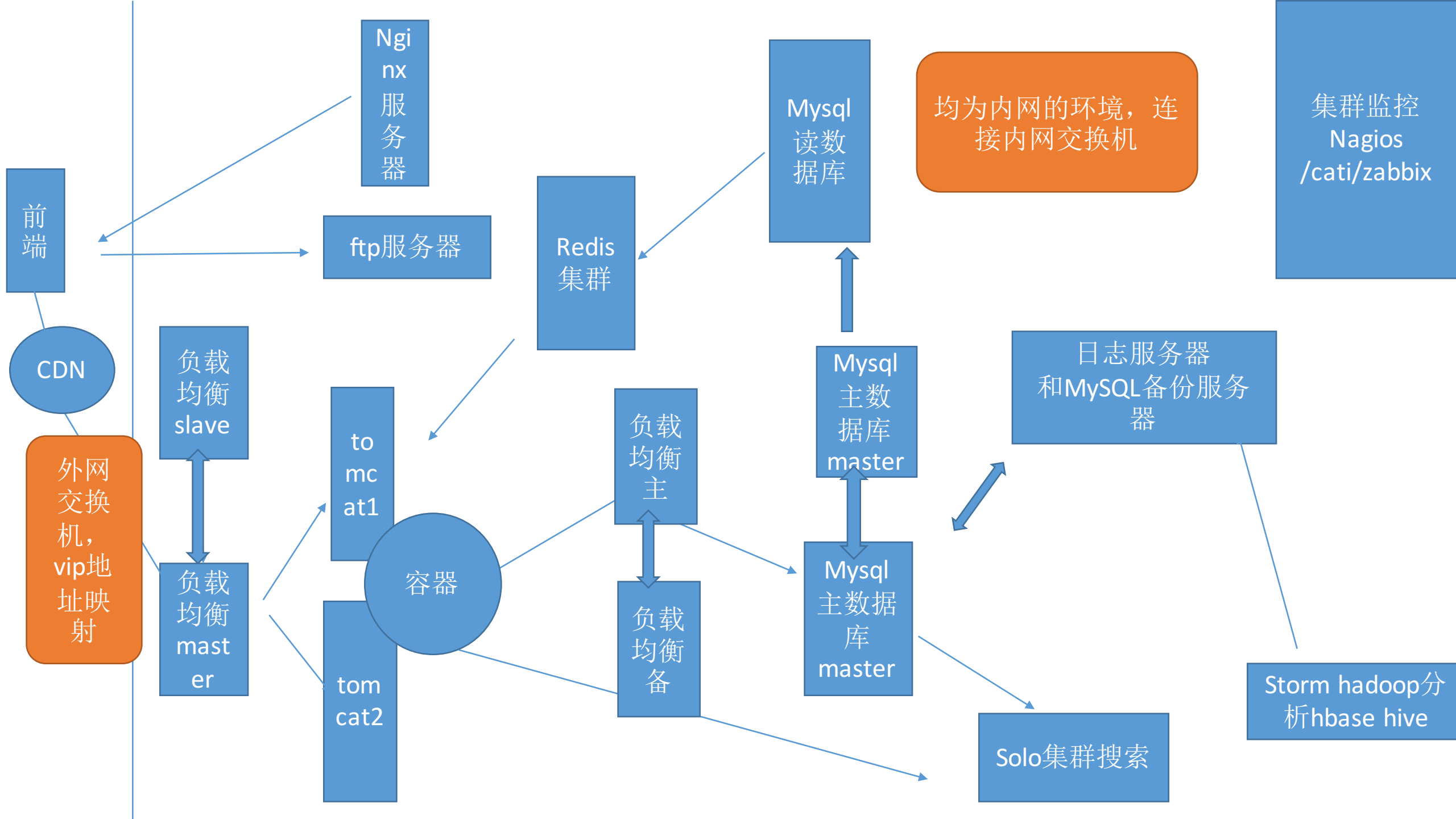
架构初设计

Log4j日志记录


Taotao-portal（门户）
Controller
Service调用
Android
IOS
微信商城
Taotao-manager
controller
service
mapper
Http调用
Taotao-rest（服务层）
Service
Mapper
Taotao-sso（服务层）
Service
Mapper
Taotao-order（服务层）
Service
Mapper
Taotao-search（服务层）
Service
SolrJ
HTTP调用
Solr服务
索引库
redis缓存
Mysql

Log4j日志记录


Taotao-portal（门户）
Controller
Service调用
Android
IOS
微信商城
Taotao-manager
controller
service
mapper
Http调用
Taotao-rest（服务层）
Service
Mapper
Taotao-sso（服务层）
Service
Mapper
Taotao-order（服务层）
Service
Mapper
Taotao-search（服务层）
Service
SolrJ
HTTP调用
Solr服务
索引库
redis缓存
Mysql


Nginx服务器
前端
ftp服务器
Redis集群
Mysql读数据库
均为内网的环境，连接内网交换机
集群监控Nagios/cati/zabbix
CDN
负载均衡slave
外网交换机，vip地址映射
负载均衡master
tomcat1
tomcat2
容器
负载均衡主
负载均衡备
Mysql主数据库master
Mysql主数据库master
日志服务器和MySQL备份服务器
Solo集群搜索
Storm hadoop分析hbase hive

域名分配

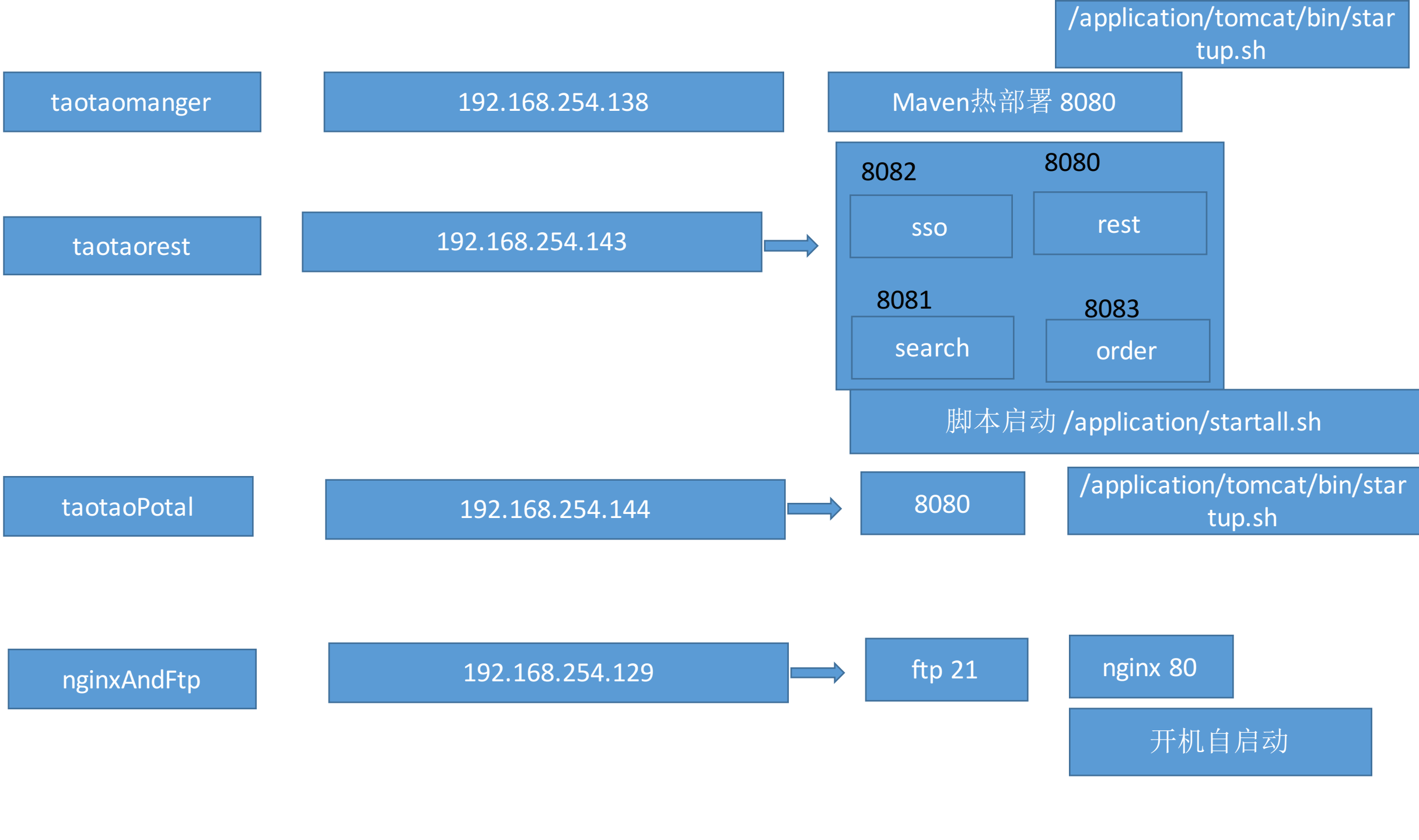
系统	域名
Taotao-portal	www.taotao.com
Taotao-rest	rest.taotao.com
Taotao-search	search.taotao.com
Taotao-sso	sso.taotao.com
Taotao-order	order.taotao.com
Taotao-manager	manager.taotao.com

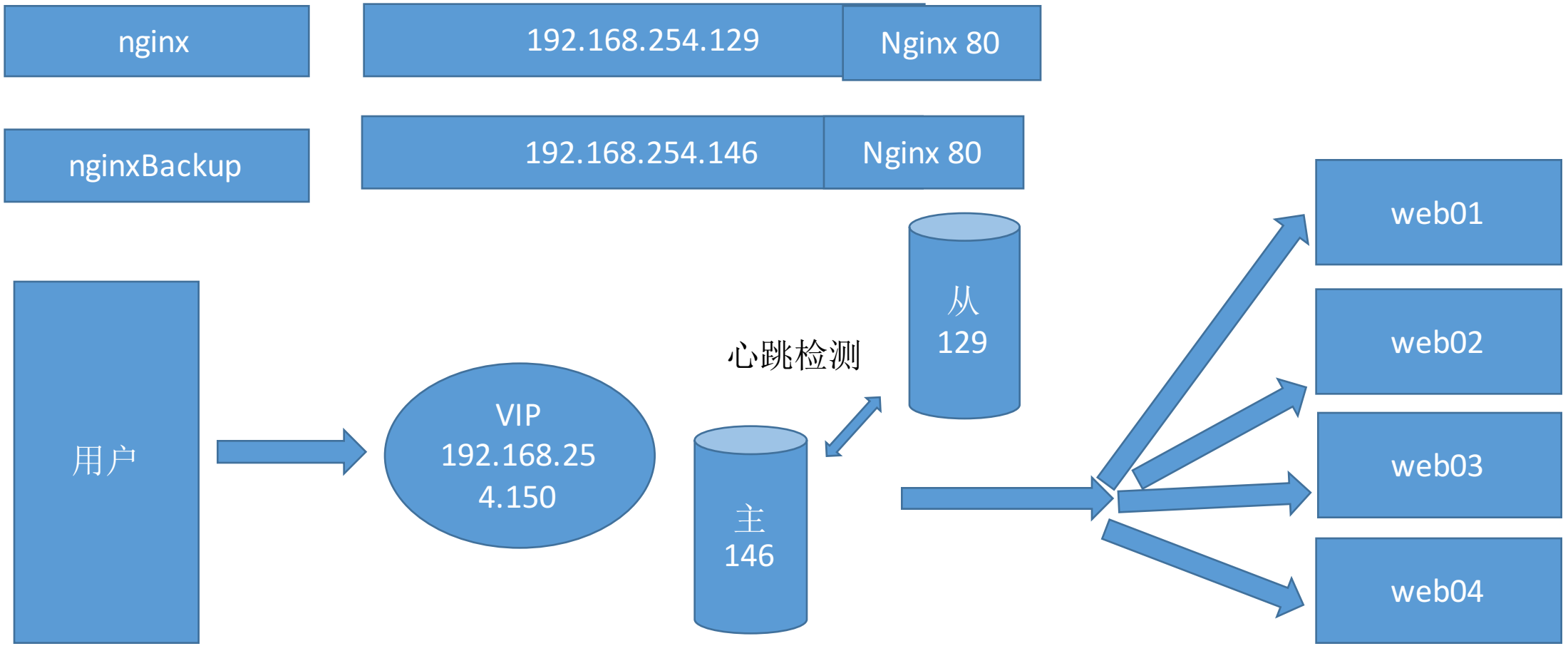
192.168.254.150 potal.taotao.com
192.168.254.150 order.taotao.com
192.168.254.150 search.taotao.com
192.168.254.150 sso.taotao.com
192.168.254.150 order.taotao.com
192.168.254.150 manager.taotao.com
192.168.254.150 rest.taotao.com
192.168.254.150 www.taotao.com

服务器规划：
图片服务器：1台虚拟机Nginx+ftp
Redis集群：1台 （6个实例）
Solr集群：1台 （6 个实例）
Zookeeper集群 1 台（3 个实例）
Mysql：2台 （主主备份）
Mysql主主负载均衡 2 台 lvs+keepalived
Mysql从数据库: 1台
Taotao-manager：1台
Taotao-porta：1台
Taotao-rest、taotao-search、taotao-sso、taotao-order：1台（4个tomcat实例）
Nginx：2 台 主 备(负载均衡) nginx+keepalived
Nfs (1主1备） 主rsync+nfs 备 rsync+inotify（脚本）实时同步
共需要11台虚拟机。

项目实施

同局域网内，一个机房


/application/tomcat/bin/startup.sh
taotaomanger
192.168.254.138
Maven热部署 8080
8082
8080
sso
rest
taotaorest
192.168.254.143
8081
8083
search
order
脚本启动 /application/startall.sh
taotaoPotal
192.168.254.144
8080
/application/tomcat/bin/startup.sh
nginxAndFtp
192.168.254.129
ftp 21
nginx 80
开机自启动



反向代理，负载均衡， 高可用Nginx+keepalived

启动
Nginx /application/nginx/sbin/nginx
Keepalived
/sbin/keepalived –f ***

MySQL主数据库 192.168.254.142 3306

同步账号backup2 zzyhappy 主主备份

MySQL主数据库2 192.168.254.147 3306

同步账号backup zzyhappy 主从备份

MySQ从数据库 192.168.254.145 3306

MySQL高可用
高可扩展性

1 MySQL自带方案

LVS+Keepalived
高可用 负载均衡

VIP
192.168.254.1
52

jdbc

心跳检测

从145

主142

主147

mysql开机自启动启动
lvs脚本，然后开keepalived
Lvs /etc/rc.local/realserver
Keepalived /sbin/keepalived start –f ****

redisCluster

192.168.254.128

伪分布

缓存高可用，
高性能

主库

Reids01 7001

Reids02 7002

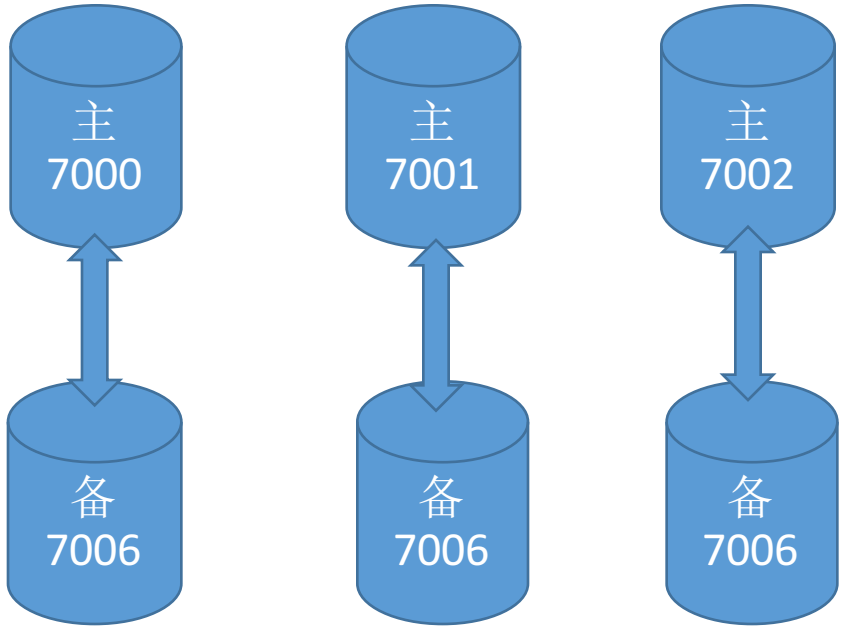
Reids02 7002

从库

Reids05 7005

Reids04 7004

Reids6 7006



Vip
192.168.254.12
8

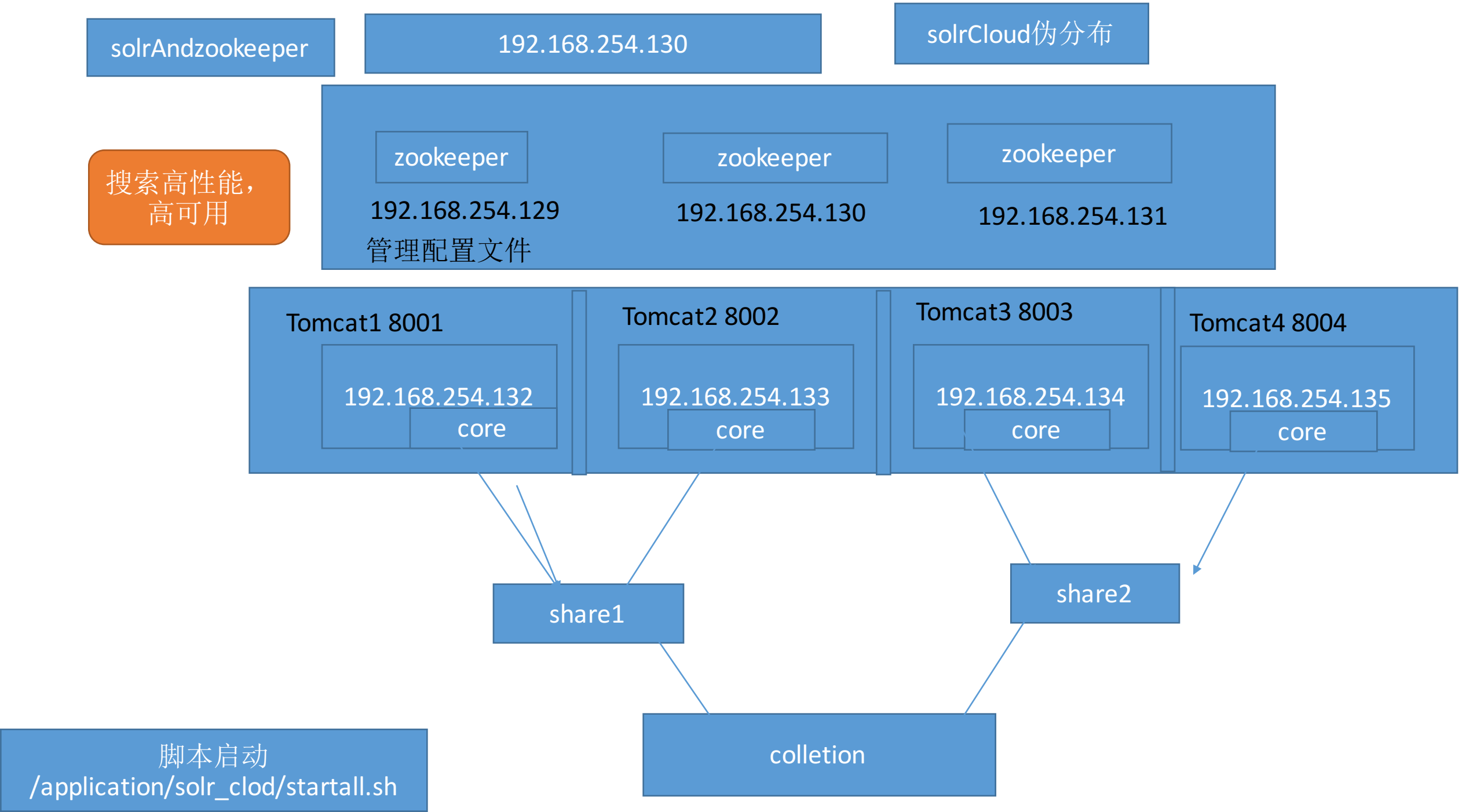
Web工程

Session共享

脚本启动
/usr/local/redis-Cluster/starall.sh

方案2 Nginx

2 Ip_hash Nginx


solrAndzookeeper
192.168.254.130
solrCloud伪分布
搜索高性能，
高可用
zookeeper
192.168.254.129
管理配置文件
zookeeper
192.168.254.130
zookeeper
192.168.254.131
Tomcat1 8001
192.168.254.132
core
Tomcat2 8002
192.168.254.133
core
Tomcat3 8003
192.168.254.134
core
Tomcat4 8004
192.168.254.135
core
share1
share2
colletion
脚本启动
/application/solr_clod/startall.sh

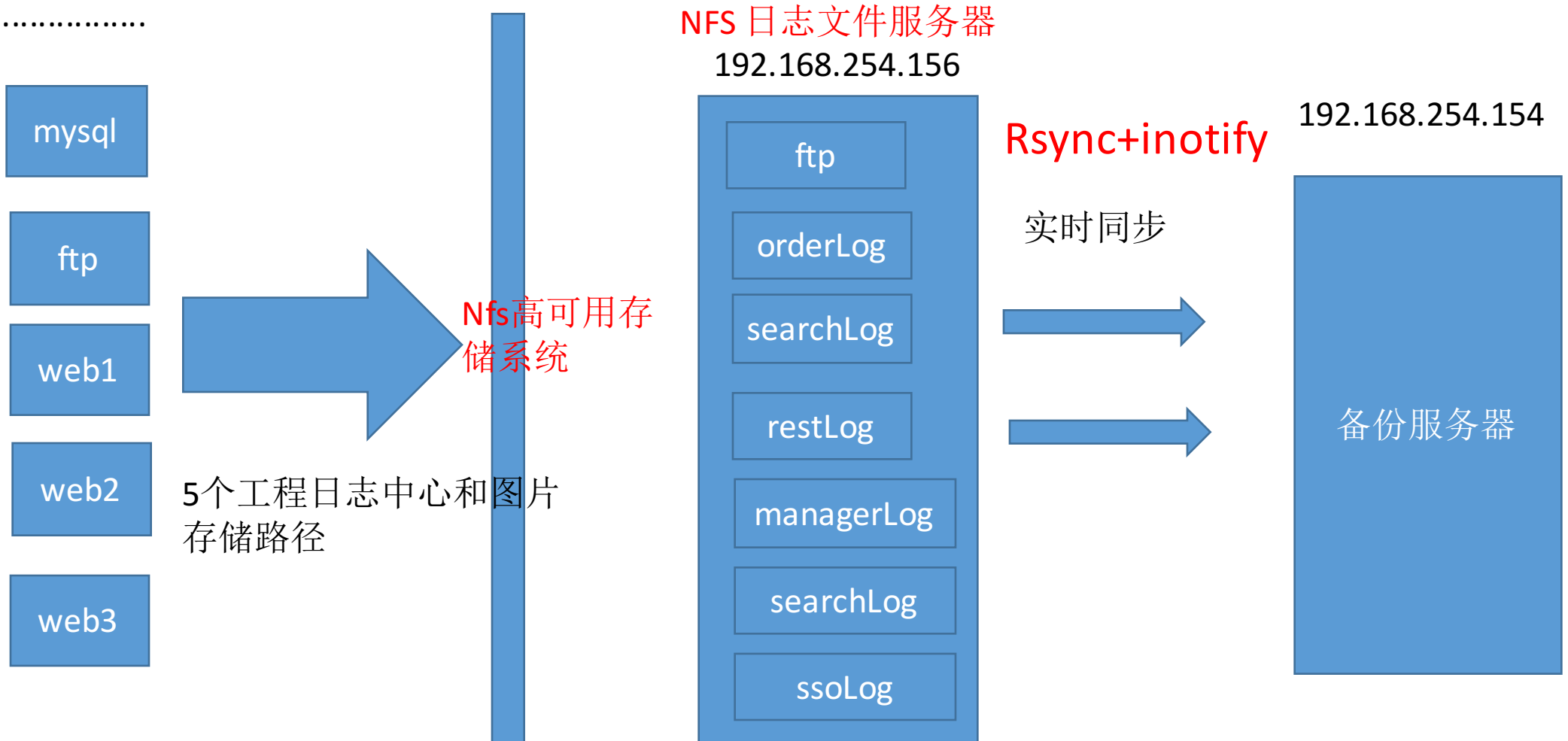
存储方案1

NFS1 192.168.254.156

备份192.168.254.154

```
mount -t nfs 192.168.254.156:/application/tomcats/ssoLog /application/serviceLog/ssoLog
mount -t nfs 192.168.254.156:/application/tomcats/orderLog /application/serviceLog/orderLog
mount -t nfs 192.168.254.156:/application/tomcats/searchLog /application/serviceLog/searchLog
mount -t nfs 192.168.254.156:/application/tomcats/restLog /application/serviceLog/restLog
.....
```

全部开机自启动



NFS1 192.168.254.156主

NFS1 192.168.254.154备

最好开机启动一下/application/software/rsync.sh脚本

Rsync客户端 +inotifywait 开机自启动
/usr/local/bin/inotifywait

Rsync服务端 同步密码 zzy:12345

注释问题很烦

软件

1 jdk1.8.0_40
2 pagehelper分页插件
3 springMVC+Spring4.1.3 mybatis3.2.8+各种客户端jar
5 阿里连接池 druid1.0.9
6 centos自带ftp
7 nginx 1.8
8 keepalived**
9 redis-2.72
10 zookeeper-3.46
12 host修改器switchHosts
13 mysql5.56
14 solr-4.10.3
15 apache-tomcat-7.0.47
16 IK Analyzer2012分词器
17 centos自带LVS 、
18 nfs-utils,rpcbind inotify-tools3.14

硬件

1 centos7-3.10.0***.x86_64
2 8g内存
3 windows+vmware

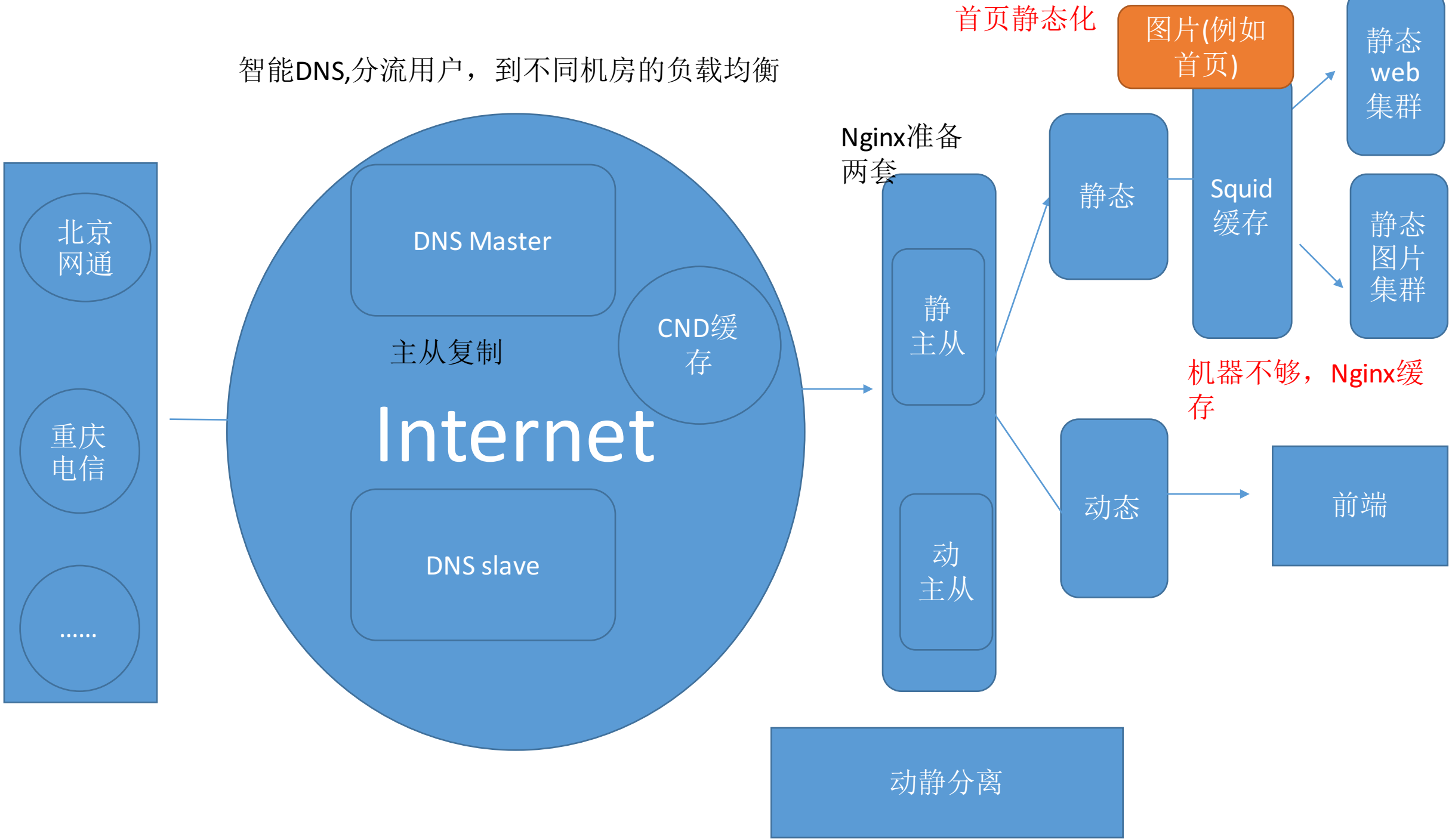
项目视频

优化

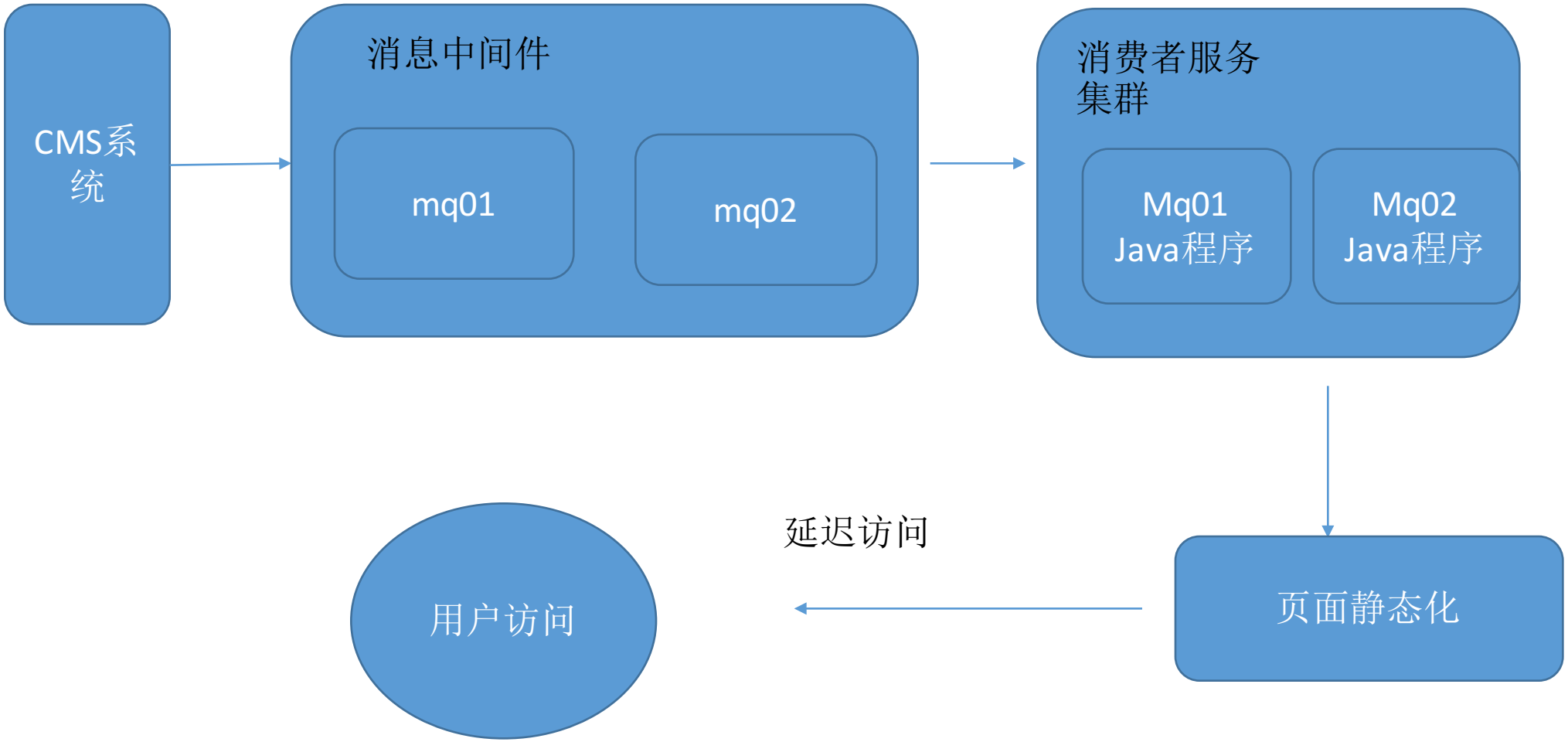
- 2 MySQL从数据库集群，同时做负载均衡
- 2 MySQL代理，不用在程序中确定从数据？？？？？
- Nginx +keepalived手动控制,脚本监控
- LVS +keepalived手动控制脚本监控，增加监控ke
- 数据库使用DRBD来保证数据的可靠性
- Tomcat集群处理高并发,根据服务器性能和使用情况设置不同的JVM参数，
- 安装监控nagios监控cpu,带宽,javaVM,内存，磁盘，端口+shell脚本双监控
- 搭建Nfs+heartbeat+DRBD高可用日志管理和图片服务器文件系统，对其进行读写lvs
- 配置文件统一管理zookeeper
- Squid缓存，Web缓存
- Web静态数据和动态数据分离，静态数据类图片服务器

拓展方案

去IOE化方案


首页静态化
图片(例如首页)
静态web集群
智能DNS,分流用户，到不同机房的负载均衡
Nginx准备两套
静态
Squid缓存
静态图片集群
北京网通
DNS Master
CND缓存
主从复制
静主从
重庆电信
Internet
机器不够，Nginx缓存
DNS slave
动态
前端
.....
动主从
动静分离

静态化


CMS系统
消息中间件
mq01
mq02
消费者服务集群
Mq01 Java程序
Mq02 Java程序
页面静态化
延迟访问
用户访问

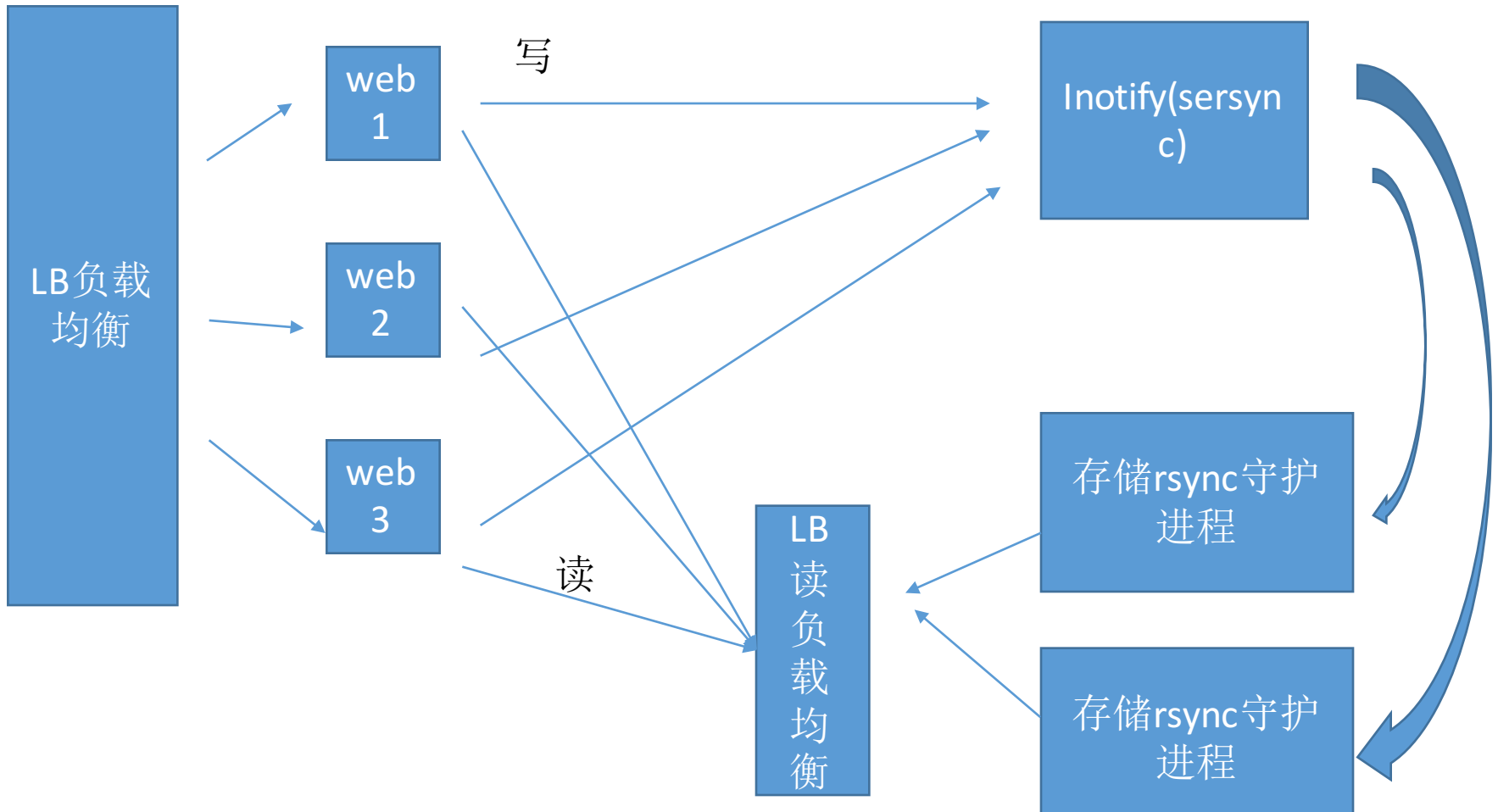
非分布式文件系统

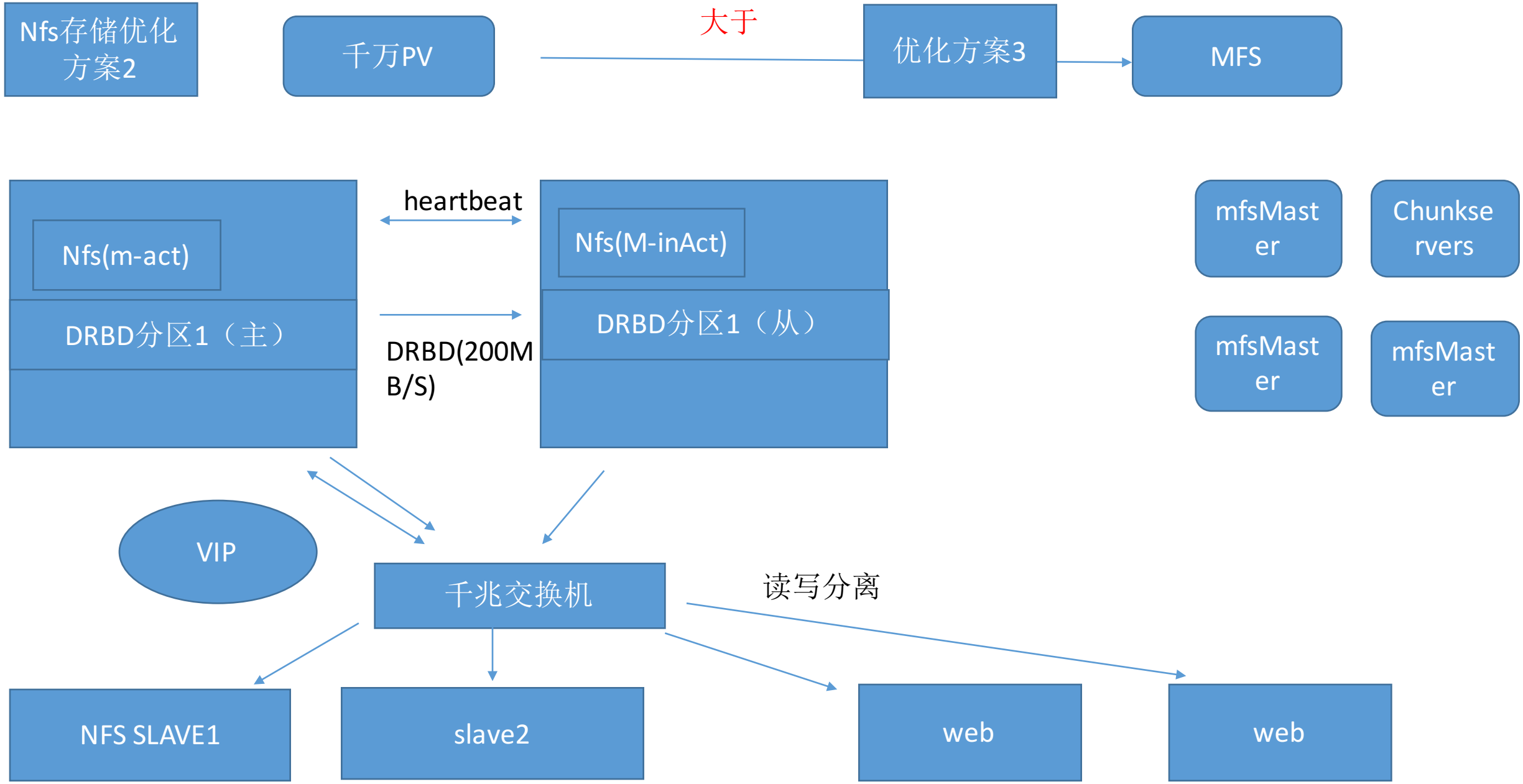
Nfs存储优化方案1

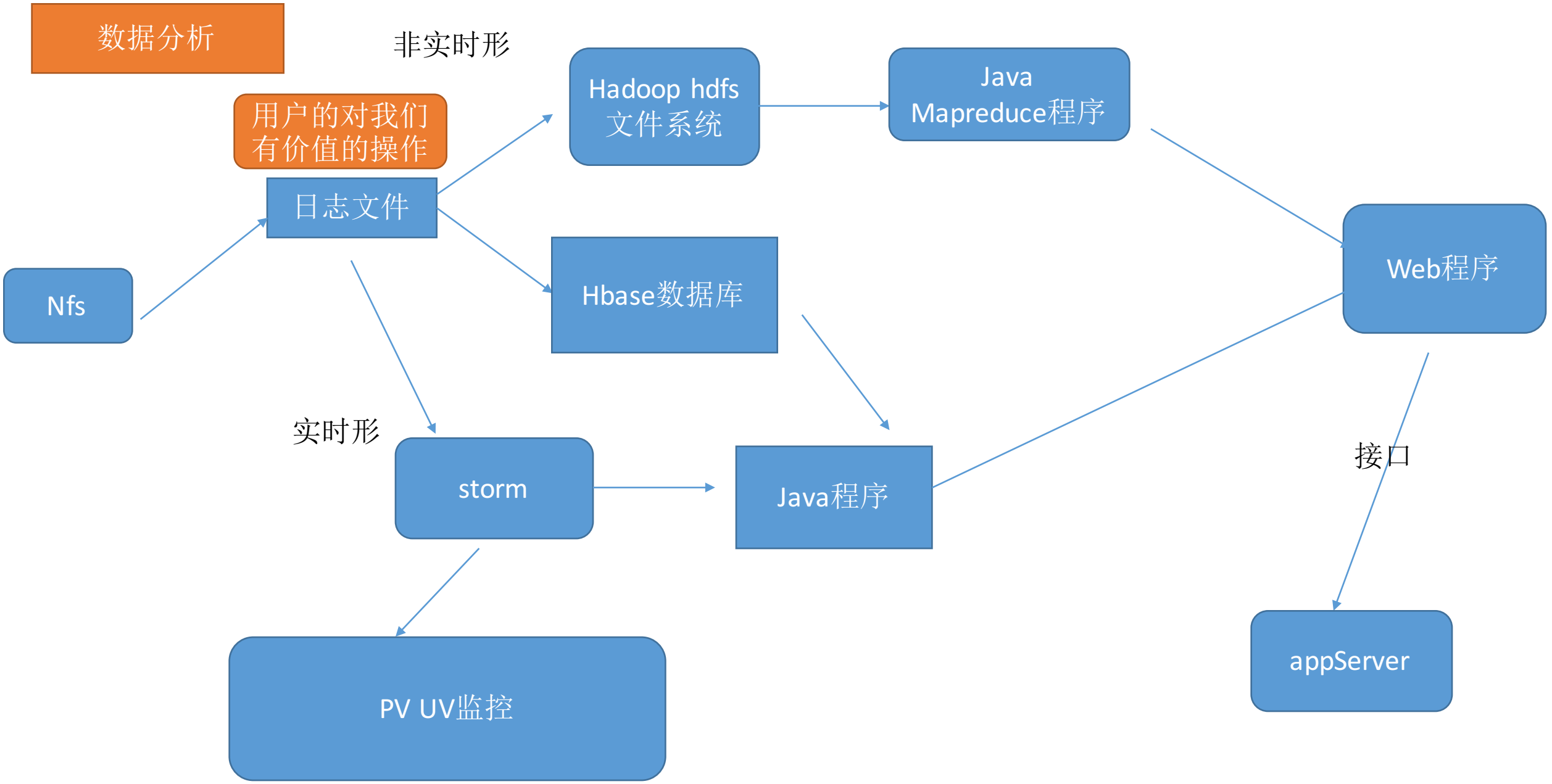
应用层，效率低

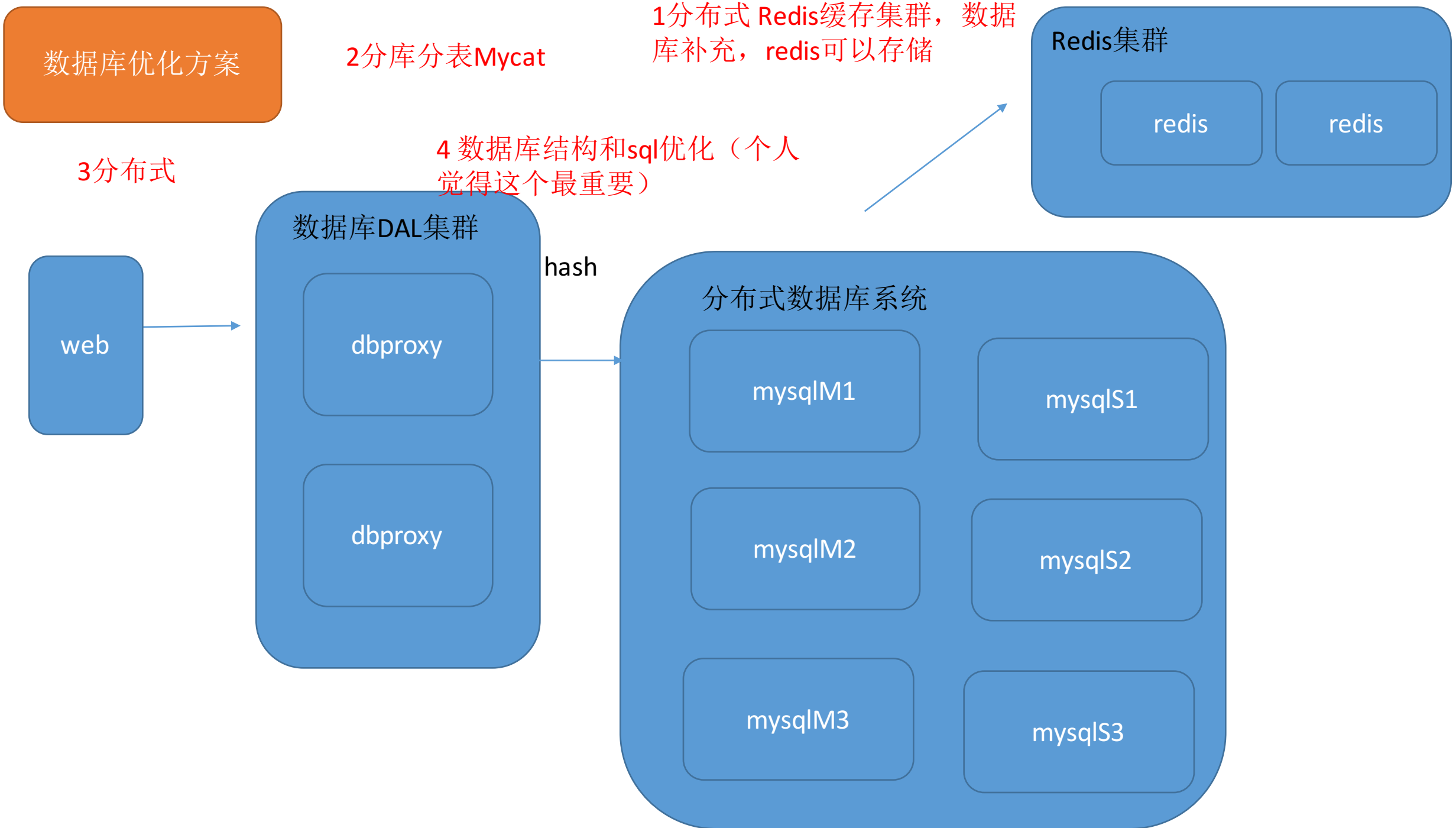
效率1S 200kb内无延迟数量小

类MySQL主从复制读写分离




Nfs存储优化方案2
千万PV
大于
优化方案3
MFS
Nfs(m-act)
DRBD分区1（主）
heartbeat
DRBD(200MB/S)
Nfs(M-inAct)
DRBD分区1（从）
mfsMaster
Chunkservers
mfsMaster
mfsMaster
VIP
千兆交换机
读写分离
NFS SLAVE1
slave2
web
web


数据分析
非实时形
用户的对我们有价值的操作
Hadoop hdfs 文件系统
Java Mapreduce程序
日志文件
Nfs
Hbase数据库
Web程序
实时形
storm
Java程序
接口
PV UV监控
appServer


数据库优化方案
2分库分表Mycat
1分布式 Redis缓存集群，数据库补充，redis可以存储
Redis集群
redis
redis
3分布式
4 数据库结构和sql优化（个人觉得这个最重要）
数据库DAL集群
web
dbproxy
dbproxy
hash
分布式数据库系统
mysqlM1
mysqlS1
mysqlM2
mysqlS2
mysqlM3
mysqlS3

Mysql 写出现瓶颈

方案1

双向复制

主主复制（非高可用）

mysqlMaster

mysqlMaster

Id 写入 1,3,5，……奇数的数据

Id 写入 2,3,6，……奇数的数据

方案2

1 4 7

mysqlMaster

Mysql Slave1

3 6 9

2 5 8

mysqlMaster

mysqlMaster

Mysql Slave2

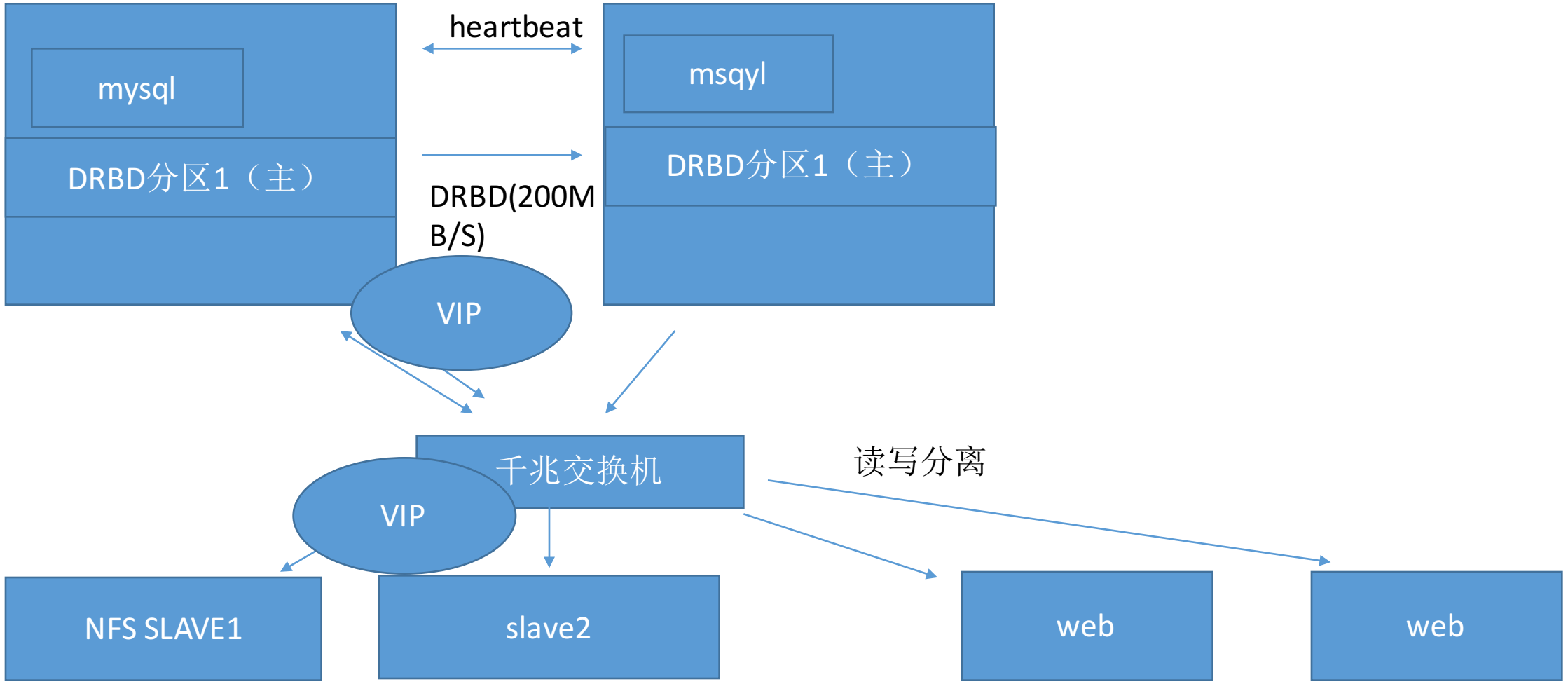
单向环状主主复制

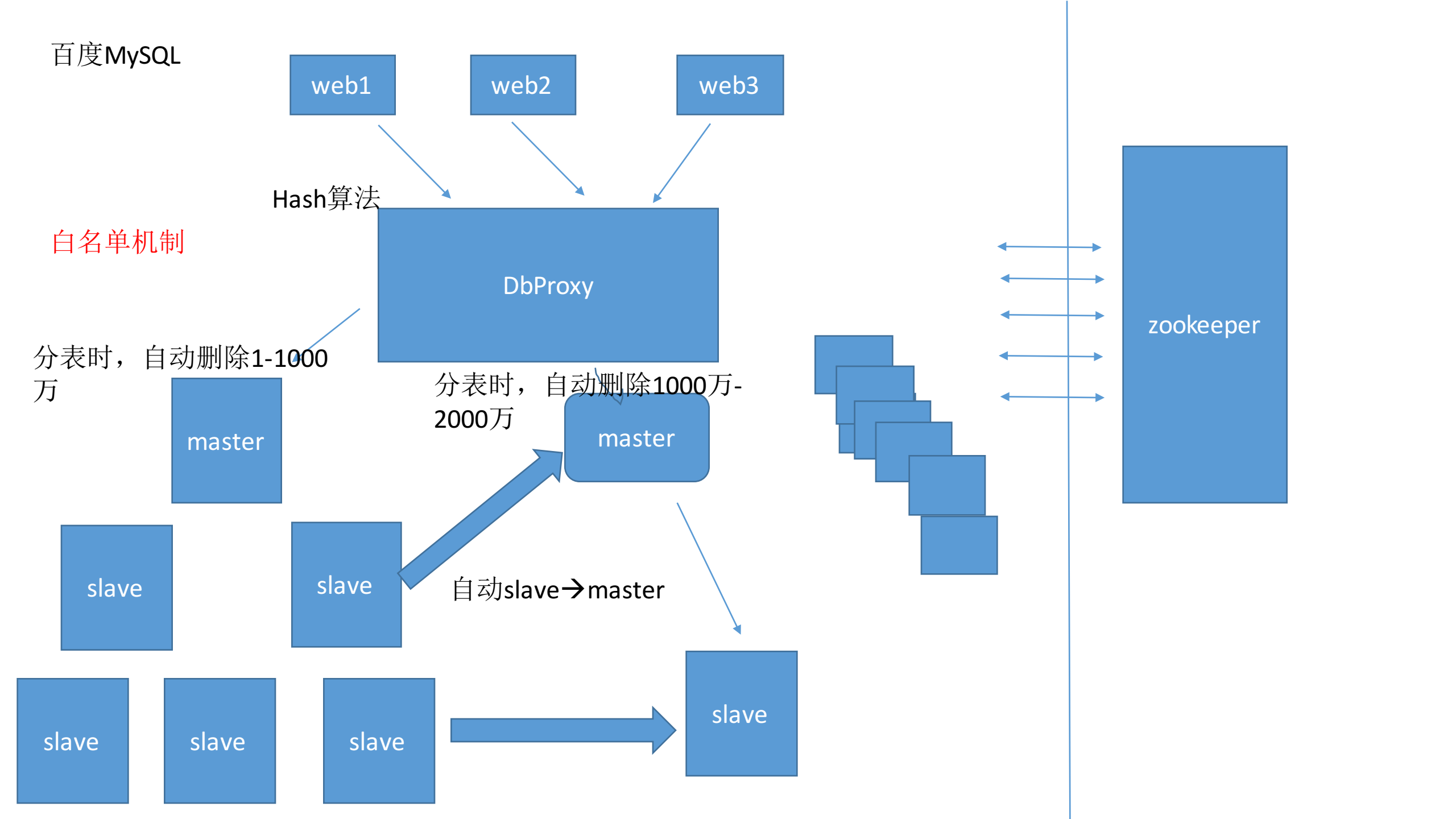
都可在主库下增加高可用，增加不同的从库

Mysql备份方案

千万PV

非mysql自带方案,mysql自带方案局限




百度MySQL
web1
web2
web3
Hash算法
白名单机制
DbProxy
zookeeper
分表时，自动删除1-1000万
master
分表时，自动删除1000万-2000万
master
slave
slave
自动slave→master
slave
slave
slave
slave

Keepalived和heartbeat是服务器级别，只有服务宕机，或者软件服务端机，或者心跳检测失效

1 hearbeat擅长存储同步，nfs,mysql，底层的数据同步
2 web服务器还是lvs keepalived
3 lvs+hearbeat不好，nginx+heartbeat可以

END