

# 架构设计方案草稿

## Impala、cdh研究

### MAVEN-nexus搭建说明

```
wget http://mirrors.hust.edu.cn/apache/maven/maven-3/3.5.2/binaries/  
apache-maven-3.5.2-bin.tar.gz
```

```
tar -zxvf apache-maven-3.5.2-bin.tar.gz -C /usr/local/
```

```
echo ' ' >> ~/.bashrc  
echo '# Set maven enviroment' >> ~/.bashrc  
echo 'export MAVEN_HOME=/usr/local/apache-maven-3.5.2' >> ~/.bashrc  
echo 'export PATH=$MAVEN_HOME/bin:$PATH' >> ~/.bashrc  
source ~/.bashrc
```

```
scp setting.xml root@10.12.40.12:/root
```

```
scp root@10.12.40.12:/usr/local/dubbo/alibaba-dubbo-3639efa/dubbo/target/  
dubbo-2.5.8.jar /Users/scott/workspaces/dev_spaces/
```

```
scp root@10.12.40.12:/usr/local/dubbo/alibaba-dubbo-3639efa/dubbo/target/  
dubbo-2.5.8-sources.jar /Users/scott/workspaces/dev_spaces/
```

```
scp root@10.12.40.12:/usr/local/dubbo/alibaba-dubbo-3639efa/dubbo-  
monitor/dubbo-monitor-default/target/dubbo-monitor-default-2.5.8.jar ~/  
workspaces/dev_spaces/
```

```
scp root@10.12.40.12:/usr/local/dubbo/alibaba-dubbo-3639efa/dubbo-monitor/dubbo-monitor-api/target/dubbo-monitor-api-2.5.8.jar ~/workspaces/dev_spaces/
```

## 框架结构

scrm

-scrm_sps	平台支撑系统
-scrm_ops	平台运营系统
-scrm_app_employee	app店员版
-scrm_app_cashier	app收银版
-scrm_h5_mas	营销活动系统
-scrm_biz_services	后台服务
-scrm_common_services	公共服务
-scrm_job_center	作业中心
-scrm_server_gate	服务网管（未来建设）

包规范：

```
cn.cecocook.scrm.common.xx.service
cn.cecocook.scrm.common.xx.service.bean
cn.cecocook.scrm.common.xx.service.impl
cn.cecocook.scrm.common.xx.service.dubbo
```

```
cn.cecocook.scrm.biz.xx.xx.service
cn.cecocook.scrm.biz.xx.xx.service.bean
cn.cecocook.scrm.biz.xx.xx.service.impl
cn.cecocook.scrm.biz.xx.xx.service.dubbo
```

## 多租户、跨客户设计 done

## Dubbox研究、设计、部署说明

```
wget https://github.com/alibaba/dubbo/tarball/master
```

准备：安装nc、tomcat、zookeeper、java opts/debug opts【回收】等

```
yum install nc -y
```

## 1、部署dubbo-admin

## 2、部署scrm\_common\_services

```
scp scrm_common_services-1.0.0-SNAPSHOT-assembly.tar.gz  
root@10.12.40.12:/usr/local/scrm_framework_beta
```

```
scp /usr/local/dubbo/scrm_common_services-1.0.0-SNAPSHOT-  
assembly.tar.gz root@node-40-57:/usr/local/scrm_framework_beta  
scp /usr/local/dubbo/scrm_common_services-1.0.0-SNAPSHOT-  
assembly.tar.gz root@node-40-58:/usr/local/scrm_framework_beta
```

```
tar -zxvf scrm_common_services-1.0.0-SNAPSHOT-assembly.tar.gz -C ./
```

Mvn 下载

### 2.1、配置OPTS

```
echo '' >> .bashrc  
echo '# Set java enviroment' >> .bashrc  
echo 'JAVA_HOME=/usr/local/java/jdk1.7.0_79' >> .bashrc  
echo 'PATH=$JAVA_HOME/bin:$PATH' >> .bashrc  
echo 'CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar'  
>> .bashrc  
echo 'export JAVA_HOME PATH CLASSPATH' >> .bashrc  
  
echo '# Set dubbo mem OPTS' >> .bashrc  
echo 'DUBBO_MEM_OPTS="-server -Xmx5g -Xms5g -Xmn512m -  
XX:PermSize=128m -Xss256k -XX:+DisableExplicitGC -XX:  
+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:  
+UseCMSCompactAtFullCollection -XX:LargePageSizeInBytes=128m -XX:  
+UseFastAccessorMethods -XX:+UseCMSInitiatingOccupancyOnly -  
XX:CMSInitiatingOccupancyFraction=70"' >> .bashrc  
  
source .bashrc
```

## 3、部署scrm\_sps

## 4、配置：

#### 4.1、多注册中心：

registry： 向指定注册中心注册，在多个注册中心时使用，值为<dubbo:registry>的id属性，多个注册中心ID用逗号分隔，如果不想将该服务注册到任何registry，可将值设为N/A

```
<dubbo:registry id="hzRegistry" address="10.20.153.10:9090" />
<dubbo:registry id="qdRegistry" address="10.20.141.150:9090?
subscribe=false" />
```

```
<dubbo:service registry="hzRegistry,qdRegistry" />
```

#### 4.2、请求并发数：

#### 4.3、集群容错：

cluster： 集群方式，可选：failover/failfast/failsafe/failback/forking

failover：

```
<dubbo:service loadbalance="roundrobin" cluster="failover" retries="2" />
<dubbo:reference loadbalance="roundrobin" cluster="failover" retries="2" />
```

注意：retries 不能设置太大，2次就好了

failfast： 写。

#### 4.4、服务降级：临时屏蔽某个出错的非关键服务，并定义降级后的返回策略。

```
RegistryFactory registryFactory =
ExtensionLoader.getExtensionLoader(RegistryFactory.class).getAdaptiveExtension();
Registry registry = registryFactory.getRegistry(URL.valueOf("zookeeper://10.20.153.10:2181"));
registry.register(URL.valueOf("override://0.0.0.0/com.foo.BarService?category=configurators&dynamic=false&application=foo&mock=force:return+null"));
```

#### 4.5、过滤、监听：

filter： 服务提供方远程调用过程拦截器名称，多个名称用逗号分隔

listener： 服务提供方导出服务监听器名称，多个名称用逗号分隔

#### 4.6、多协议：

使用指定的协议暴露服务，在多协议时使用，值为<dubbo:protocol>的id属性，多个协议ID用逗号分隔

layer：服务提供者所在的分层。如：biz、dao、intl:web、china:acton。

#### 4.7、分组聚合

#### 4.8、事件通知

#### 4.9、并发控制。提供者、消费者（暂时不可生产使用）

executes：服务提供者每服务每方法最大可并行执行请求数

```
<dubbo:service interface="com.foo.BarService" executes="10" />
```

如果<dubbo:service>和<dubbo:reference>都配了actives，<dubbo:reference>优先

Load Balance均衡：

配置服务的客户端的loadbalance属性为leastactive，此Loadbalance会调用并发数最小的Provider

#### 4.10、令牌验证(试用)

#### 4.11、本地存根、本地伪装（服务降级配合）

Mock：设为true，表示使用缺省Mock类名，即：接口名 + Mock后缀，服务接口调用失败Mock实现类，该Mock类必须有一个无参构造函数，与Local的区别在于，Local总是被执行，而Mock只在出现非业务异常(比如超时，网络异常等)时执行，Local在远程调用之前执行，Mock在远程调用后执行

与服务降级的区别，服务降级可以人为直接干预

Dubbo:service 属性：

Token: 令牌验证，为空表示不开启，如果为true，表示随机生成动态令牌，否则使用静态令牌，令牌的作用是防止消费者绕过注册中心直接访问，保证注册中心的授权功能有效，如果使用点对点调用，需关闭令牌功能

Dubbo:reference 属性：

cache：以调用参数为key，缓存返回结果，可选：lru, threadlocal, jcache等

client：客户端传输类型设置，如Dubbo协议的netty或mina。

## 5、规范

5.1、出现服务循环依赖，尽量避免，如果出现可以关闭检查。

5.2、开发和连接测试环境服务时，消费方和服务方可以直连提供者；服务方可以订阅提供方。

5.3、参考推荐用法，例：在 **Provider** 上尽量多配置 **Consumer** 端属性，比如服务使用方更清楚服务性能参数，如调用的超时时间，合理的重试次数

5.4、隐式传参，由消费方设置，提供方获取。用户框架集成，不建议常规业务上使用。

```
RpcContext.getContext().setAttachment("index", "1");
```

5.5、参考服务化最佳实践

熔断策略：hystrix

**熔断机制（Circuit Breaker）**，也叫**自动停盘机制**，是指当股指波幅达到规定的熔断点时，交易所为控制风险采取的暂停交易措施。

kryo序列化

多协议：

设计：

开发规范：

部署说明：

Istio 谷歌的服务治理框架

**Redis部署、客户端utils**

**缓存控制设计**

## zookeeper安装【文档】 - 冯利光

下载jdk1.8和zookeeper3.4.11

```
wget --no-cookies --no-check-certificate --header "Cookie:
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-
securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u151-
b12/e758a0de34e24606bca991d704f6dcbf/jdk-8u151-linux-x64.tar.gz"
```

```
wget https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.4.11/
zookeeper-3.4.11.tar.gz
```

远程传输jdk和zookeeper

```
scp jdk-8u151-linux-x64.tar.gz root@node-40-57:/root/
scp jdk-8u151-linux-x64.tar.gz root@node-40-58:/root/
scp jdk-8u151-linux-x64.tar.gz root@node-40-69:/root/
```

```
scp zookeeper-3.4.11.tar.gz root@node-40-57:/root/
scp zookeeper-3.4.11.tar.gz root@node-40-58:/root/
scp zookeeper-3.4.11.tar.gz root@node-40-69:/root/
```

1> 解压安装。

```
$ tar -zxvf ~/jdk-8u151-linux-x64.tar.gz -C /usr/local/
$ tar -xzvf zookeeper-3.4.11.tar.gz -C /usr/local/
```

2> 配置相关环境变量

```
echo '' >> /root/.bashrc
echo '# Set java enviroment' >> /root/.bashrc
echo 'JAVA_HOME=/usr/local/jdk1.8.0_151' >> /root/.bashrc
echo 'PATH=$JAVA_HOME/bin:$PATH' >> /root/.bashrc
echo 'CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar'
>> /root/.bashrc
echo 'export JAVA_HOME PATH CLASSPATH' >> /root/.bashrc
source ~/.bashrc
```

日志配置：

```
mkdir -p /usr/local/scrm_framework_beta/log
mkdir -p /usr/local/scrm_framework_beta/snapshot
```

```
zookeeper.log.dir=/usr/local/scrm_framework_beta/log
```

zookeeper.tracelog.dir=/usr/local/scrm\_framework\_beta/log

关闭firewall 和 策略检测:

```
systemctl stop iptables.service  
systemctl disable iptables.service
```

```
systemctl stop firewalld.service #停止firewall  
systemctl disable firewalld.service #禁止firewall开机启动
```

```
selinux 永久关闭  
vi /etc/selinux/config  
将SELINUX=enforcing改为SELINUX=disabled  
设置后需要重启才能生效
```

3> 修改conf/zoo.cfg

```
tickTime=2000  
dataDir=/usr/local/scrm_framework_beta/snapshot  
clientPort=2181  
initLimit=5  
syncLimit=2  
server.1=master1-40-12:2888:3888  
server.2=node-40-57:2888:3888  
server.3=node-40-58:2888:3888
```

4> 设置myid

```
dataDir指定的目录下面，创建一个myid文件， 设置数字 1。  
$ mkdir snapshot  
$ vim snapshot/myid
```

5> 其他zk服务器循环执行1~4步骤。

或打包上述配置好的zookeeper复制到各个zookeeper服务器

```
$ scp -P 21968 ./zookeeper.tar.gz vjfk@zk2.vjf.com:/usr/local/zookeeper/  
zookeeper-3.4.11/conf/  
$ scp -P 21968 ./zookeeper.tar.gz vjfk@zk3.vjf.com:/usr/local/zookeeper/  
zookeeper-3.4.11/conf/
```

```
password: 83815c29acfe550ffabf27d7f5414d70
```

6> 启动ZooKeeper集群

```
zookeeper有权限写输出文件。  
$ chmod 745 zookeeper-3.4.11/zookeeper.out
```



各个节点启动：

```
$ bin/zkServer.sh start
```

各个节点验证

```
$ bin/zkServer.sh status
```

client连接

```
$ bin/zkCli.sh -server node-40-57:2181
```

消息队列，高可用安装

**elastic job** 作业调度中心

数据库访问组件设计-**SHARDING-JDBC or Mycat. Done**

数据字典设计-跨客户设计

公共组件

公共服务

- 多租户
- 跨客户开户-分库、分表
- 数据字典服务
- 缓存控制服务
- 用户及权限

第三方服务

- 巨牛

平台**PAAS**最终落地及合作流程

- 1、首先给到我们的部署方案，确定是否能支持
- 2、对确定合作的部分，进行邮件流程确认，并最终大佬邮件签字。
- 3、确认双方配合流程，尤其是7\*24小时的配合和监控