# Dao proxy设计

## 目标

高可用

1.       全网自动检测网络错误并自动切换到可用服务进程

2.       自动检测存储损坏，或存储处理能力不组，并自动切换到其他存储，接口级别或者进程级别

3.       全部分业务的后台进程响应时间（从我们的用户发起，到用户收到）比十月一的系统高

吞吐量

1.       能限制 某个 应用类型（例如session,im,web）在某个应用机房某个接口的吞吐上限

安全性

1.       dao\_proxy合法性，应用合法性，DBD合法性

2.       针对某个应用类型（例如session,im,web），限制其能访问的接口

3.       对每个应用类型进行验证

4.       对每个登陆用户uid进行验证

调试测试统计需求

## Proxy在分层体系中定位及依赖

各业务dao提供的接口各不相同。

业务层提供的接口必然是完整。

业务层提供必要的处理量，负载信息。

Proxy是前端应用接入数据访问层的唯一入口，为使得应用获取最大的可用性，proxy建议和应用部署在同一机房，同时做到单一机房内多点部署

Dao失效或者网络故障的诊断过程中，允许proxy处于行为不确定状态。

Proxy在故障确定的状态下，需要明确告知应用api请求是否达成。

客户端api通过daemon体系识别并接入的proxy，并采用server-lib3基础库做到机房内及跨机房自适应切换。

在proxy进入不确定状态时，由客户端api自行通过超时告知应用请求失败。

在proxy明确进入完全不可用状态，将proxy错误告知上层。

## 设计达成指标

自动处理DB故障，故障发生切换时间不超过1分钟；

正常情况每个请求2秒钟内返回；

负载均衡要求：整体处理能力要达到理论值的80%

访问量超过DB处理能力时，要提高DB处理能力80%以上的处理能力；

在proxy到某个DAO故障时，自动切换到访问正常的DAO，并保持服务稳定；

单个proxy的透传性能要求：1万/秒（智杰提出1万，待讨论）

## 接口发现与管理

协议thrift打包。

业务dao提供本服务提供的所有接口名。

客户端请求使用 接口名.函数调用名 进行调用。

Proxy解出调用包头，采用接口名进行请求分发。

## 负载均衡及故障转移处理框架

静态路由配置：

A：本机房优先（若集合中存在本机房的，则把所有非本机房的删除掉）

B：主库优先（若集合中存在主库的，则把所有非主库的删除掉）

C: 从库优先（若集合中存在从库的，则把所有非从库的删除掉）

E: 一致性优先（参数H）（把H从hash映射查找出groupid，用groupid来过滤，若没有用下一个groupid）

F: 固定某机房（参数Gs）（从集合中选取某些机房）

动态可用性监控：

D: 高响应策略（从集合中选出一个最可能获得最高响应速度的一个）

U: 可用性优先（参数X）（从集合中按可用程度标准X，把所有判定为不可用的删除掉）（client可改）

**DB故障模型**，是描述DAO所在的机房内单个数据点的故障，例如：数据库损坏，数据库性能突发下降，DAO机器性能问题，DBclient堵塞问题，DAO进程处理线程池问题

**传输响应模型**，是描述数据包在往返proxy和dao之间的响应时间和丢包可能。例如网络丢包率升高，发送进程的发送buffer满，接收进程的接收队列小堵，接收进程网络线程处理慢。

Dao处理平均响应时间估计？

路由决策算法：

上述策略按过滤器的形式实现。

使用者根据实际情况自行组合，一次路由请求依次流经过滤器，过滤不符合策略的路由，注意：所有的路由规则到达最后都有一个默认的高响应策略规则，该策略是通过动态的可用性监控来发现。



## 故障检测的基本矛盾

在故障检测中，响应时间和超调量是基本矛盾。

也就是说对故障诊断的时间越长，越有可能精确定位故障。如果允许比较过激的反应，则可以获得较快的故障响应速度。

在我们整个体系中，基本上应用层还是允许丢包的情况出现。而需要防止整个系统的共振及雪崩效应。我更偏向较保守的故障诊断时间。

## DB故障详细实现方案

这个是智杰在迅雷gfs系统中采取的负载均衡方案。

每个DAO进程包含若干个dao接口

每个dao接口的状况用两个值来描述

int free; //当前可用资源数，当前可调用次数，或者有相关性的值

int load; //当前负载，当前处理量除处理性能，或者有相关性的值

基本要求

free值要求大于等于零，当接口完全不可用时，free=0

load值对同一个dao接口，在不同DAO进程有可比性

----------------------------------

参照此设计，目前一个具体实现：

根据目前网络队列+线程池的进程模型，队列的当前大小，剩余大小是一个关键的反映进程处理情况的指标。

所以定义计算方法如下：

free = 队列剩余大小\*20

load = 队列当前大小\*1000 + 上十秒进程处理包量

此方法计算得到的值，是与可用资源，当前负载有相关性的一个值，并且不同Dao进程之间有可比性。

优点：简单。

此计算方法未来可继续完善的点

1.调用数据库或其他存储的时候，若错误太多，则直接设置free=0

2.load值可以考虑不同DAO进程处理性能的差异，对于处理性能强的DAO（例如采用纯内存存储），可以对load值 除一个 与处理性能成正比例的值

3.增加DAO的主动推送故障协议，以达到更短故障信息同步时间

## 传输响应模型详细实现方案

这个是server-lib3网络自适应切换的演进版。

定义词汇：ipport，ip与port两者定义了一个网络目的地。

每个DAO进程，可以有若干个ipport。

传输响应模型，是对每个ipport使用 两个值来描述：

int bad\_level; //传输质量等级，数值越小，质量越高

float resp\_time; //短期平均响应时间，RTT，

两个ipport之间比较传输响应质量的时候，优先比较bad\_level，其次比较resp\_time

----------------------------------

参照此设计，目前一个具体实现是

初始值bad\_level=1; resp\_time=5秒

bad\_level取值范围0,1,2,3,4,5,6

定义一个ping协议包，DAO进程对此包的处理是马上回复pingresp包。

proxy进程定时向所有DAO进程发送ping包

若ping包超时间，则bad\_level++;

若ping包回复，rtt为 X 则进行下述两步：

A若bad\_level>2，则bad\_level=2; 否则bad\_level--;

B resp\_time+= ((X-resp\_time)\*0.5)，（0.5衰减的指数窗平均值更新）

若发生网络断线，则bad\_level++;

优点：简单。

此计算方法未来可继续完善的点

1. 发包后对响应时间影响的预测

### 智杰的基于排名的可用性计算方法

DB故障模型基础计算方法

--DAO进程检测 + Proxy同步DAO的检测结果

传输响应模型基础计算方法

--DAO响应ping包 + Proxy通过ping DAO进行检测

模型特殊更新方法

--通过在特定时机，特定事件处理过程中，修正模型，以补充模型基本计算的不足。

///////////////////////////////////

可用性排名计算

指定一个Dao接口，对提供此Dao接口的所有DAO进程的此功能的可用性进行从高到低的排名。换句话，对预测响应时间的从小到大的排名。排名第一位者，则被选择为预测最高可用性的目的地。

流程：

1. 按DB故障模型进行排名，每个候选DAO进程得到一个排名A （对模型评估相接近的进程，排名允许重复），第一名的最不可能故障
2. 按传输故障模型进行排名，每个候选DAO进程得到一个排名B（对模型评估相接近的进程，排名允许重复），第一名的传输响应时间预测最短

然后对每个候选DAO进程的排名相加A+B结果的值，进行排名。

得到最终的可用性排名

### 王征的故障处理模型

该模型是两部分组成。

1. 静态策略：全网随机处理。
2. 动态故障策略：引入请求平均时间监控，在平均处理时间达到一定阀值认定出现故障，节点不可用。
3. 对不可用节点移入维护队列，定时发送模拟业务请求，

### 风险及应对措施

性能问题

每次请求都要经过一个较为复杂，并且可能更为复杂的路由计算。

缓存，一次路由查询获得一定量请求配额。

故障摇摆

在某一机房请求出现故障后，一旦压力迁移走，请求负载将降低。造成业务总是处于故障与非故障的临界状态摇摆

无法彻底消除，拉长摇摆的周期

加权平均