下图是Mycat 2.0的多中心集群方案，分布于不同地域（Zone）内的一些Mycat Cluster组成双中心多3中心方案，前提是这些不通中心中的Mycat Cluster 可以以某种方式完成数据库端的数据同步机制。每个中心都有一组Mycat负载均衡器LB，这些LB与同一中心内的Cluster组成一对多关系，即一个LB可以服务一个中心内的所有Cluster的负载均衡请求，也可以是多个LB，每个负担不同的Cluster的流量。此外建议是每一个LB都有一个Backup，平时并不连接Cluster，但监测到Master下线以后，就立即开始连接Cluster并开始工作。

Zone1

MySQL

MySQL

Zone2

Replication

LB2

(back)

LB1

(master)

下图是一个 Mycat Cluster的组成部分，它是位于某个特定中心（Zone）的一个处理单元，包括，一个Mycat cluster包括如下信息：

* 所属的中心（Zone），固定不可变的标示
* 具有地理位置标示，比如北京联通机房1
* 1个或多个采用相同配置（**引用同一个MyCat Schema配置**）的Mycat Server，用作负载均衡

Mycat server 1

Mycat server 2

Mycat server 3

MySQL数据库服务器，是属于一个Zone内部的共享资源，不属于Cluster级别的，主要拥有以下关键信息：

* IP地址、端口和名称
* 所在主机Host，为了区分数据迁移石时候是否需要复制文件
* MySQL群组，定义一组具备主从关系的MySQL服务器之间的关系
* 管理员权限的用户名密码等，用于自动运维操作

此外，配置管理部分，我们需要记录集群中所用的的主机的信息，包括账号密码等,hostId不可变化，可以理解为内部分配的一个编号（不同于主机名），在所有的Zone中保持唯一。

/mycat-hosts/${hostId}

如

host\_1 {hostname:xxxx,ip:xxxx,root:xxxx,password:xxx }

Zone的值是系统初始配置的，一开始就确定有几个Zone，创建cluster的时候选择下拉。

/mycat-zones/${zoneId}

如

bj {name:北京中心}

Mycat Server节点定义如下，nodeId为每个Mycat Sever的唯一编号，在所有的Zone中保持唯一，Server的参数包括用于负载均衡的信息，以及Mycat Server自身的一些特定配置信息。

/mycat-nodes/${nodeId}

如 mycat\_bj\_01

value:**{**

**name:** mycat\_bj\_01**,**

**hostId:host\_1**

**host:192.178.3.1:8066,**

**zone:bj,**

**cluster:mycat-cluster-1,**

**weigth:1,**

**leader:1,**

**state:red(red/green/yellow)**

**system-params:/params from system.xml**

**}**

}

负载均衡器LB的路径如下，loadbalance-id是全Zone唯一编号。

/mycat-lbs/ ${loadbalance-id}

如

Lb1 value {name:load balance bj 01 , hostId:host1 ,status:master/standby}

负载均衡器LBGroup如下定义

/mycat-lbs/ group{$}

如

LBGoup1 value:

{

Zone:bj,

Lb\_servers:[

Balances: [loadbalance\_01, loadbalance\_02]

{

nodeCluster: **mycat-cluster-bj-1,**

**servicePort:8066,**

**},**

{

nodeCluster: **mycat-cluster-bj-2,**

**servicePort:8166,**

**}**

**]**

}

MySQL节点信息，mysqlId全Zone唯一，不可改变

/mycat-mysqls/${mysqlId}

如：

Mysql\_1 {ip:192.168.8.2,port:3366,user:mysql,password:mysql,hostId:host,zone:bj}

MySQL主从复制关系组，定义一组有主从复制关系的MySQL实例，repgroup-id

/mycat-mysqlgroup/${repgroup-id}

比如

Mysql-rep-1 {repType:0,zone:bj,[{writeHost:mysqlId1,readHosts:mysqlId2, mysqlId3},{writeHost:mysqlId3,readHosts:mysqlId4}]

repType:0表示普通的MySQL主从复制关系，1是Galera Cluster 集群。

ZK-Server记录了集群的信息，Mycat-eye、Mycat-Server等从ZK读取配置并协同工作。

Mycat 安装包中提供一个zk-config.bat/sh工具，该工具从 conf/zk-default.txt中加载zk路径到ZK-Server中去，完成ZK-Server数据的初始化过程。然后Mycat-eye与 LB可以用ZK来管理集群。

Mycat-eye的界面原型参考：

defaultzone

LoadBanceGroup

LB1

LB2

Mycat Cluster1

MyCat 1

MyCat 2

MySQL Groups

Groups1

Groups2

说明：

* defaultZone为zk中的第一个zone，是一个下拉框或者可以切换的组件
* Mycat Cluster1表示对应Zone里第一个cluster，这里需要一个图标来表明是“Cluster”，下面的Mycat 1,Mycat 2是属于这个 Cluster的两个Server，也需要一个猫的图标。有多个Cluster就重复上述显示。
* LoadBanlceGroup则为对应Zone里的负载均衡Group，LB1,与LB2为两组负载均衡Group，每个Group是一主一备两个Mycat LB实例组成，必须位于不同的物理服务器上。
* MySQL Groups则是对应Zone里的所有MSQL服务器实例，但每个MySQL实例都属于一个Group，Group里的这些实例组成 主从关系，即使没有主从关系的，仍然需要顶一个一个Group，为了未来的扩展。

对于Mycat Cluster实例

点击以后，出现下面的界面

Mycat Version ,系统关键参数，如内存的使用参数，签订连接数，后端连接数，内存，队列等关键信息。每个Mycat Server当前使用的Write节点 是否发生过切换，如果切换过需要一个提示的图标，另外，如果几个Mycat实例当前用写节点不同，则表明发生了不一致的切换，以第一个MyCAT实例为准，后面与他不同的所有MyCAT的切换标志显示为ERROR的图标。

对于MySQL Group实例

点击以后，

MySQL 1 Version ,系统关键参数，如内存的使用参数，cache的使用情况等，每行一个MySQL实例

主从复制状态信息，区分普通主从与Galera Cluster  两种不同的信息

展示主从拓扑图，对于Glaster来说则是网状互联的 ，因为实际上只有几种固定的主从模式，所以可以做成静态图片，上面对应位置放MySQL节点名称的做法，简化Top图的展示