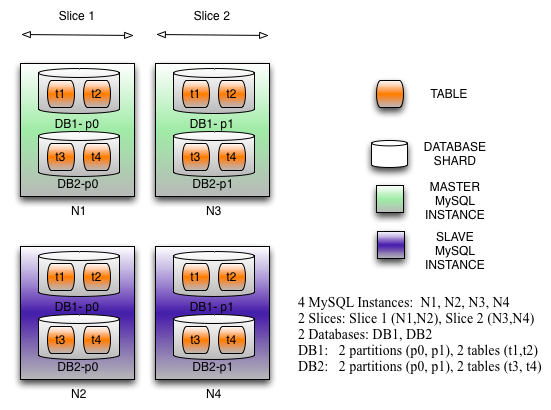
使用Apache Helix构建Mysql-cluster

Mysql-cluster增加了sharding,automatic replication和failure detection。Mysql在单节点上执行，但是再进行扩展很难实现，Fullmatix可以使Mysql以分布式的方式运行。在多实例中分发shards，对于MySQL的一些特性不能够很好的支持，例如不能在多个shards实现Joins和transaction，但是在同一个shard内部还是支持joins和transactions。

# 系统架构

架构图如下所示：



1. MySQL Instances

使用Mysql 5.6安装的Server实例，用户可以启动任意多的Mysql实例，而且在单物理机上也可以启动一个或者多个MySQL实例。

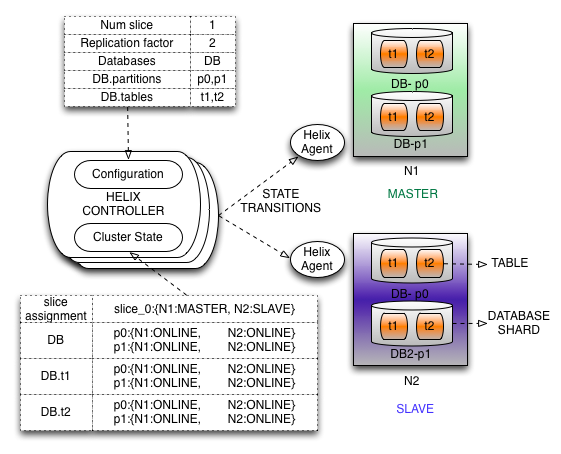
1. Slice及replication factor（数据副本）

Slice为包含相同数据的数据分片子集，可以分布在所有的实例中。Slice 实例代表着replication，以#slices\*代表着副本的名称。

1. Sharding/Partitioning（数据片段）

Sharding将数据分成多个部分，MySQL的数据分成数据库和表，在本处sharding以Database为粒度，在一个数据库中所有的表都被sharded。

Mysql-cluster的运行图如下所示：



# Quick Demo

Quick Demo 的Cluster中有两个Mysql实例，并启动两个Helix Agent来监控MySQL instances，这两个agent以Participants的角色运行。一个Agent作为Master，另外一个作为Slave，Slave自动周期实现与Master的数据同步。客户端以Spectator的角色，获取cluster的external view，感知master/slave及database shards的位置。初始化以后，程序模拟失败及自动回复：

1. 客户端发送100条数据写操作，然后验证两个节点有相同的数据，副本是否正确建立
2. Disable当前Master，模拟失败，Helix自动将Slave切换到Master
3. 客户端自动检测新的Master，并发送200条写操作
4. Enable老的master，然后将cluster作为slave，slave识别Current Master，并创建副本
5. 停止操作，验证两个节点的数据是否相同。因为Helix会自动实现错误监测，处理和恢复

## 2.1下载

1) 下载mysql tarball

*wget http://ftp.jaist.ac.jp/pub/mysql/Downloads/MySQL-5.6/mysql-5.6.35-linux-glibc2.5-i686.tar.gz*

2)下载mysql-cluster源码

*git clone* [*https://github.com/kishoreg/fullmatix.git*](https://github.com/kishoreg/fullmatix.git)

1. 启动如下

*cd mysql-cluster*

*./quick-demo-setup-mysql.sh ~/mysql-5.6.35-linux-glibc2.5-i686.tar.gz*

*输出如下所示：*

*./quick-demo-helix-setup.sh*

*./quick-demo-run.sh*

# 执行分析

3．详细架构分析

https://github.com/kishoreg/fullmatix/wiki/Installation-and-setup