Tag: TDDL

[补充实习笔记] TDDL入门

January 31, 2015

TDDL(Taobao Distributed Data Layer) 是平台架构&开放平台-分布式产品组提供的一套分布式数据访问引擎。

为什么需要TDDL？

核心的目标是为了解决以下三个问题：

一、数据访问路由

将针对数据的读写请求发送到最合适的地方。

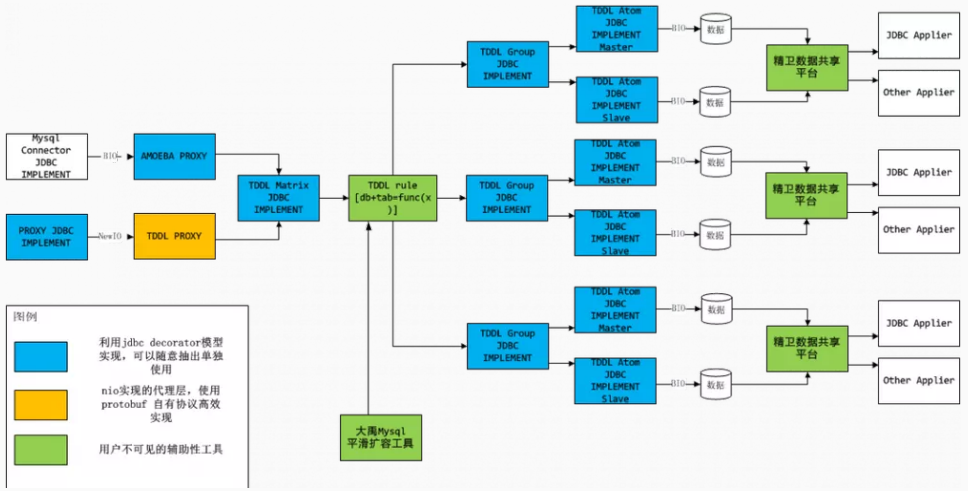
二、数据的多向非对称复制

一次写入，多点读取

三、数据存储的自由扩展

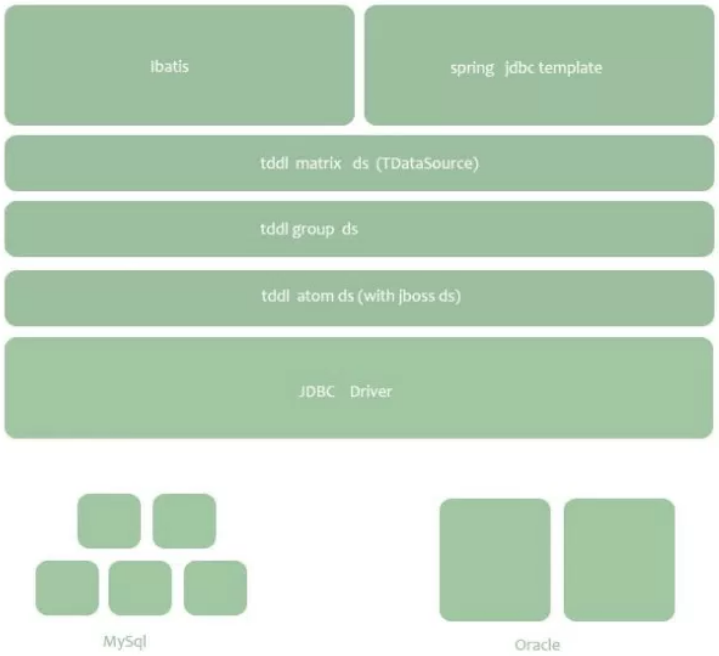
不再受限制于单台机器的容量瓶颈与速度瓶颈，平滑迁移。

TDDL流程图



所有蓝色的部分（TDDL三层架构），都是实现了jdbc接口的具体实现，因此可以按照需要随意选用。之所以为此，原因很简单，因为每层都有损耗，不管是易用性还是性能，都会有，所以要看用户的选择，需要用什么就上什么，我们不干涉。具体的三层架构见下：

TDDL整体三层架构



TDDL主要部署在ibatis或者其他ORM框架之下，JDBC Driver之上，三层数据源每层都按JDBC规范实现， 所以可以将其当作与普通数据源实例化并且注入到各种ORM框架中使用。
TDDL架构分为了3层，最上层的TDataSource负责**分库分表路由**和**结果合并**，持有多个GroupDS实例；中间层TGroupDataSource 负责**主备切换**和**读写分离**，即根据读写特性与权重，选择一个具体的DB Server进行操作，持有多个AtomDB实例；最下层AtomDataSource持有原子的数据源（剥离的JBOSS数据源）并且可以动态改IP ,连接信息等，能够动态响应数据源创建与销毁。包括IP、用户名、密码、最大最小连接池、连接参数，从Diamond上拉取并交给JBOSS DB。JBOSS DB完成数据库连接管理与池化操作。3层数据源都可以单独使用，应对不同的应用场景。
先来看一条SQL的具体执行流程。而后分别分析三层。

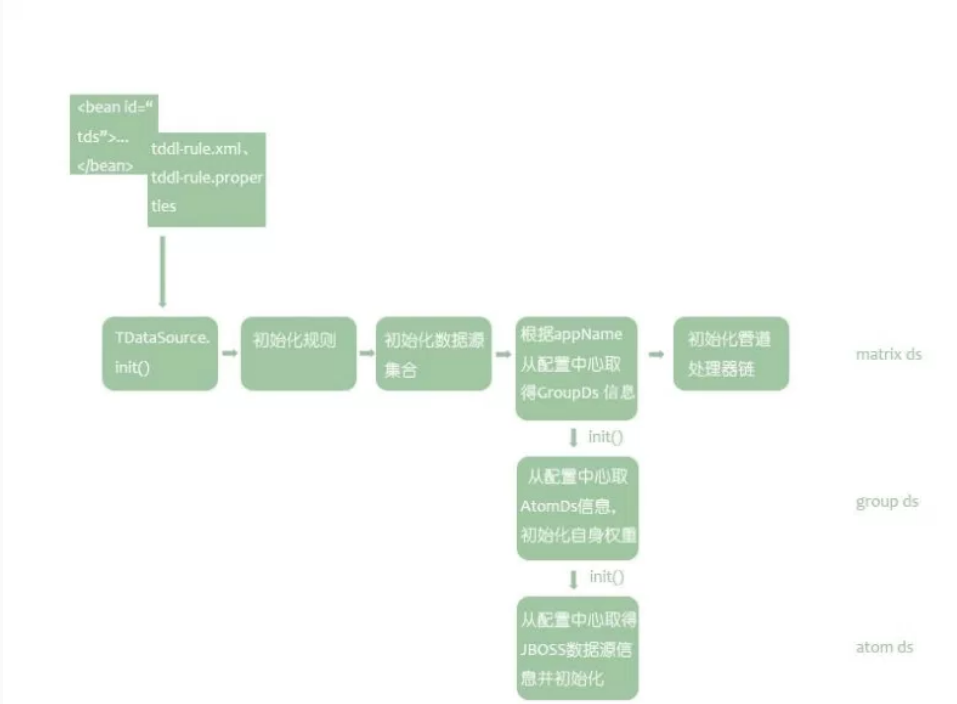
TDDL执行SQL过程

其中sql解析/ 规则计算/表名替换/选择GroupDs执行sql/合并处理多个结果集 5个步骤在Matrix层执行， 根据权重选AtomDs/具有重试策略地在AtomDs上执行SQL 2个步骤在Group层执行， 读写数控制、线程并发数控制/执行sql返回结果集 2个步骤在Atom层执行。

为何要做SQL解析？

在这一步，TDDL将SQL中分库分表的条件进行解析提取，包括order by, group by, limit m,n, join等信息、sum, max, min等聚合函数信息、distinct信息。（此外，TDDL行复制需要重新拼写SQL，带上sync_version字段。次要理由。）

TDDL初始化



TDDL初始化主要包括规则信息的初始化和各层数据源拓扑结构的初始化，其中前者包括 规则中的分库分表字段、规则字符串的处理和库表对应关系初始化，后者包括Matrix层 的dsMap初始化，Group层每个GroupDs实例中多个(或单个)AtomDs的可用不用，权重关系 等初始化，Atom层根据本地配置或者持久配置中心的数据源信息初始化剥离的JBOSS数据源， 并且将自身实例和这些数据源——对应映射起来。这一初始化的过程是从上到下的，也就是 说只有Matrix层配置的GroupDs才会初始化，只有这类GroupDs中配置的AtomDs才会初始化。

TDDL Matrix Data Source

Matrix DS 用于管理数据片sharding，核心功能是根据给定的参数从规则引擎中计算出目标库的路由信息，然后进行路由。
基本功能：
1) 数据的水平切分
数据库的水平切分，从本质来说就是根据关键的key，将数据切分为多个小片。主要的切片方法是取模和按日期，还有些会将这两种方式综合起来使用。而TDDL的目标也就在于对这些情况进行更好的支持。因为mysql的表有数据量的限制，因此在切库的基础上，还要进行适度的切表，以保证数据库的硬盘的使用率。
2) 事务管理
TDDL提供了最基本的事务管理功能，首要的目标是让业务现有的事务模型更加简化。
3) 数据库选择
与事务管理结合，允许业务指定查询的具体目标，用于彻底的替换dbRoute.提供更简单的配置方式和配置模型。与事务管理结合后，可以提供更为清晰的事务判断和事务管理。

4) 规则的集中管理
规则的集中管理有利于进行数据库的平滑扩展时的状态切换和统一管控。在进行平滑扩展时，分库分表信息必须持有两套不同的规则，在准备完成的时候一并进行切换。
5) 不同数据散列结构读写分离
主要针对以下场景的支持，假设有一张表，是auction_auctions ,在主库内是分布在两个oracle中的， 从库则分布在16个mysql中，每个mysql有4个表。
那么在这种情况下的读写分离就是不同数据散列结构的读写分离了。TDDL对这种场景也可以予以支持。
TDDL Group Data Source
Group datasource 是构建于atom datasource之上的数据源，他的目标是管理多组数据完全相同的数据库。比如通过mysql或通过oracle进行数据复制后的主备数据库。目标是在rjdbc的基础上，对主备，备库可读的场景进行更好的支持。
基本功能：
1) 主备数据库动态相互容灾切换
支持进行主备的对调切换，状态对调后备库变为主库，主库变为备库，类似rjdbc的作用。
2) 相同数据片内读写分离
针对mysql replication机制进行的数据主备复制，可以直接使用group datasource来支持读写分离。读写分离支持权重设置，允许对不同库使用不同的权重。
3) 读重试
一台数据库挂掉后，如果是个fatal exception ，那么会进入读重试，以确保尽可能多的数据访问可以在正常数据库中访问。
4) 数据库挂掉后的线程保护，不会因为一个数据库挂掉导致所有线程卡死。
使用try – lock机制来进行线程保护，在第一次捕捉到fatal exception以后，只允许一个线程进入数据库进行数据访问，直到数据库可以正常的工作为止。
5) 流量控制，数据库保护

TDDL Atom Data Source

Atom Datasource 如其名，就是一个原子的datasource。从本质来说，就是一个对jboss的浅包装。允许业务通过非jndi的方式获取jboss的数据源，同时也可以协助dba进行统一的数据库管理。依托于diamond server进行数据配置的统一管理。从而实现一个数据库，唯一的对应一套ip+port+username+password+schema的目的。减少配置中的环节。让数据库使用更简便。
基本功能：
1) Dba管控
2) 定期密码变更
3) Jboss数据源管理的相关功能
功能介绍：
a) 线程数统计
线程统计的作用是，在线程进入数据库访问的getConnection()阶段，线程数自增1，在退出数据库访问的close()阶段，线程数自减1。使用这种方式统计真正与数据库交互的线程数。
这样如果数据库挂掉，那么会有更多线程等待已经挂掉数据库的连接，如果到达伐值（可在app规则中设置，默认关闭）,那么会拒绝超出伐值后的线程访问数据库的请求。
b) Try-lock机制
如果数据库发生了fatal exception(也就是断连接，数据库硬盘满)等异常的时候，TDDL会抓取到这种异常。一旦发现这种异常，立刻对数据库访问进行保护，只允许单个线程访问数据库，其他线程访问则全部抛出错误。
使用这种方式可以比较有效的保护业务机器的处理能力。
c) 状态切换
DBA发现问题时，可以迅速反应，将数据库状态置为NA,那么所有访问都会直接抛出异常，从而保护业务服务器。
d) 执行次数统计
允许通过设置指定单位时间片内执行的修改类型的sql和查询类型的sql的个数。
Atom使用的参数全部在Diamond上，包括三部分：
1、global:数据库ip, port, dbname信息
2、app:appName和数据源连接池、连接参数、用户的映射
3、passwd:数据库用户名和密码的映射
（2、3可以有多个）

Diamond取数据过程

以IC最新的Mysql动态数据源为例：
因为做了分库分表，所以最上层是Matrix数据源（TDataSource），appName是IC_MYSQL
那么拼成Matrix的dataId：com.taobao.tddl.v1_IC_MYSQL_dbgroups
组名是DEFAULT_GROUP（后续所有dataId的组名都是DEFAULT_GROUP）
以dataId和组名通过http GET访问diamond：(以日常10.232.10.23为例)
http://10.232.10.23:8080/diamond-server/config.co?group=DEFAULT_GROUP&dataId=com.taobao.tddl.v1_IC_MYSQL_dbgroups
得到：
MYSQL_ICDB_00,MYSQL_ICDB_01,MYSQL_ICDB_02,MYSQL_ICDB_03,MYSQL_ICDB_04,MYSQL_ICDB_05,MYSQL_ICDB_06,MYSQL_ICDB_07,MYSQL_ICDB_08,MYSQL_ICDB_09,MYSQL_ICDB_10,MYSQL_ICDB_11,MYS
这个内容是appName为IC_MYSQL的Matrix的group列表（对应TGroupDataSource）
对group列表中的每个group，例如MYSQL_ICDB_00
拼成Group的dataId：com.taobao.tddl.jdbc.group_V2.4.1_MYSQL_ICDB_00 这个dataId 用来获取一个group内部的主备结构，读写权重配置
组名是DEFAULT_GROUP
以dataId和组名通过http GET访问diamond：(以日常10.232.10.23为例)
http://10.232.10.23:8080/diamond-server/config.co?group=DEFAULT_GROUP&dataId=com.taobao.tddl.jdbc.group_V2.4.1_MYSQL_ICDB_00
得到
MYSQL_67_ICDB0:r10w10p0,MYSQL_69_ICDB0:r0w0p0
这个内容是MYSQL_ICDB_00的具体主备结构配置；逗号分隔开每个库；每个库的冒号前面是Atom数据源(对应TAtomDataSource))的key，冒号后面是这个库对应的读写权重。
对每个Atom数据源的key，例如MYSQL_67_ICDB0
拼成Atom相关的2个dataId：
com.taobao.tddl.atom.global.MYSQL_67_ICDB0 这个用来获取数据库本身的信息
com.taobao.tddl.atom.app.IC_MYSQL.MYSQL_67_ICDB0 这个用来获取appName对物理库如何使用的信息（其中IC_MYSQL就是最开始Matrix数据源的appName）
http://10.232.10.23:8080/diamond-server/config.co?group=DEFAULT_GROUP&dataId=com.taobao.tddl.atom.global.MYSQL_67_ICDB0
得到：ip=10.232.31.67 port=3306 dbName=icdb0 dbType=mysql dbStatus=RW
http://10.232.10.23:8080/diamond-server/config.co?group=DEFAULT_GROUP&dataId=com.taobao.tddl.atom.app.IC_MYSQL.MYSQL_67_ICDB0
得到：userName=ic minPoolSize=10 maxPoolSize=20 idleTimeout=60 blockingTimeout=500 connectionProperties=connectTimeout=1000;characterEncoding=gbk;socketTimeout=1000
根据上面两个结果中的dbName和userName拼成数据库密码的dataId：
com.taobao.tddl.atom.passwd.icdb0.mysql.ic
http://10.232.10.23:8080/diamond-server/config.co?group=DEFAULT_GROUP&dataId=com.taobao.tddl.atom.passwd.icdb0.mysql.ic
encPasswd=6eaf1b*********a8
这个是加密后的密码
至此，数据库所有的配置信息获取完毕。
在异常流程下，目前主要通过两种方式来保证业务可以正常启动
1. 如果这个机器曾经启动过，那么如果diamond死掉，他会使用本地cache启动机器
2. 如果这个机器从来没有启动过，diamond死掉，那么可以让sa scp数据文件到本地机器中，启动服务

Posted in 生活, 笔记

Tagged TDDL, 天猫, 实习

Leave a comment