

RDBS-HBASE 同步工具 概要设计

目 录

第 1 章	概述	3
1.1	目的	3
1.2	文档历史	3
1.3	术语	3
第 2 章	系统简介	3
2.1	系统功能	3
2.1.1	关系数据库管理.....	3
2.1.2	HBASE 集群连接.....	4
2.1.3	RDBS 变化捕捉并同步到 HBASE.....	4
2.1.4	配置文件管理.....	4
2.1.5	处理日志管理.....	4
2.1.6	其他	4
2.2	系统环境	5
第 3 章	系统用例	5
3.1	用例参与者	5
3.1.1	RDBS 和 HBASE.....	5
3.1.2	工具使用人员.....	5
3.1.3	维护人员	5
3.2	用例简介	5
3.3	数据库连接管理.....	6
3.3.1	通过命令行参数连接数据库并选定同步方式.....	6
3.3.2	用户通过配置文件连接数据库并选定同步表等信息	6
3.3.3	数据库访问受限或者密码错误.....	6
3.4	集群管理	7
3.4.1	集群的连接	7
3.4.2	集群的增加	7
3.4.3	集群的删除	7
3.5	日志和中间文件管理	7
3.5.1	日志写入	7
3.5.2	中间文件的写入.....	8
3.5.3	中间文件的读取.....	8
第 4 章	关键问题和解决方案	8
4.1	变化捕捉策略选择和分析	8
4.1.1	问题描述	8

4.1.2	选择方案：杨鹏的论文：异构数据库同步中间件的设计和实现[4]	8
4.1.3	变化捕捉方法总结	11
4.2	异构处理方法	12
4.2.1	数据库平台异构问题处理	12
4.2.2	数据类型异构处理	12
4.3	RDBS 数据依赖解决（表模式转换）	13
4.3.1	问题描述	13
4.3.2	解决方案	13
4.4	RDBS 数据库的连接维护、中断续传处理、中间文件处理	15
4.4.1	问题描述	15
4.4.2	解决方案	15
第 5 章	系统结构	15
5.1	静态结构	15
5.1.1	概述	15
5.1.2	读表模块	16
5.1.3	表处理模块	17
5.1.4	存表模块	17
5.1.5	内部接口	17
5.2	动态结构	17
第 6 章	数据结构	18
6.1	数据模型	18
6.2	数据库表	18
6.3	配置文件	错误! 未定义书签。
第 7 章	系统性能	19
7.1	性能需求	19
7.2	差错控制	19
第 8 章	参考资料	20

第1章 概述

1.1 目的

在 java 环境下，制作一个从关系数据库到 HBASE 的实时同步的工具，支持 MS-sqlserver、oracle、mysql 三种关系数据库类型。使数据维护人员进行数据库维护和大数
据运算的时候可以本别选用关系数据库和 HBASE，充分发挥两种数据库模式的优点并避开
其弊端。本文档供产品经理，开发人员、测试人员参考。

而在市面上已经有了 HBASE 和 RDBS 的导出工具，例如 sqoop 等，所以此处我们只
做 RDBS 到 HBASES 的同步工具。

1.2 文档历史

文档历史			
日期	参与者	版本	动作
2016-3-10	胡官涛	1.0	初始版本
2016-3-20	胡官涛	2.0	增加详细需求分析
2016-3-28	胡官涛	3.0	增加总体设计和模块设计
2016-4-4	胡官涛	4.0	增加数据字典、细化模块等设计

1.3 术语

术语	说明
RDB	关系数据库，包含 MS-sqlserver、oracle、mysql
JDBC	Java Data Base Connectivity, java 数据库连接
ODBC	Open Database Connectivity, 开放数据库互连
Sqoop	Oracle 开发的关系数据库与 HBASE 双向导出工具
MapReduce	一种编程模型，用于大规模数据集（大于 1TB）的并行运算

第2章 系统简介

2.1 系统功能

2.1.1 关系数据库管理

采用标准 ODBC 数据库接口，可连接的关系数据库为：MS-sqlserver、oracle、mysql。在 RDBS
的连接时均采用远程连接方式，从而更加便于管理，也可以同时连接多个不同数据库。

2.1.2 HBASE 集群连接

在 java 环境下，利用 Hadoop 连接并管理 HBASE 集群。使用 MapReduce 进行大数据量的高速读写。

2.1.3 RDBS 变化捕捉并同步到 HBASE

应用触发器-日志表的变化捕捉策略捕捉 RDBS 的变化，然后将信息同步到 HBASE 并写入日志。可以选择对特定的数据表进行监控同步，也可以设置在某个时间段内进行监控同步。

2.1.4 配置文件管理

每次操作的输入方式为配置文件输入。可以在配置文件中自定义要输入的参数（在已参数类型的子集中），然后通过读取参数直接开始处理，便于管理也不易于出错。

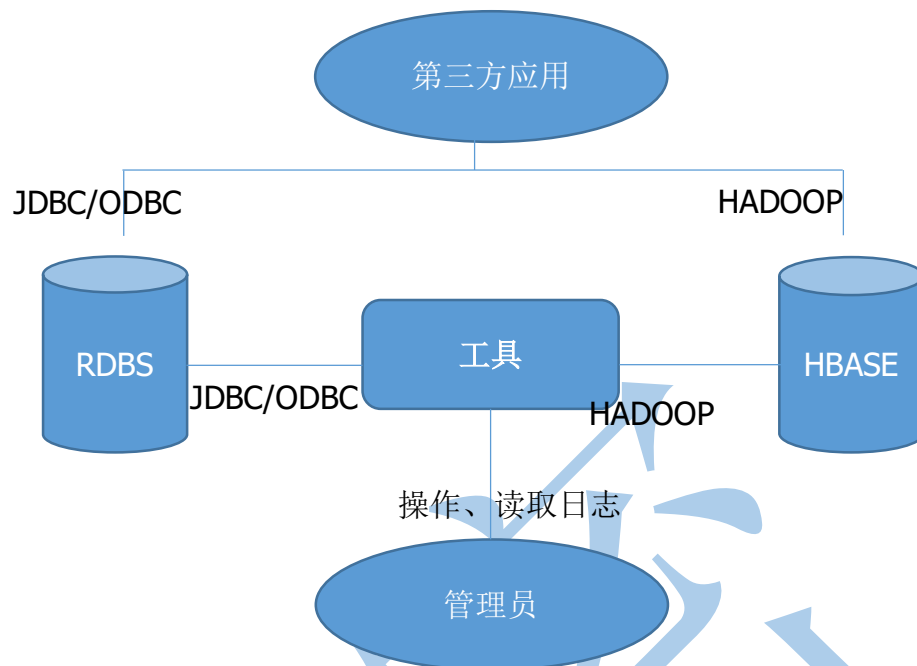
2.1.5 处理日志管理

每次操作都会存入日志文件，并定期备份日志，做好每次监控。

2.1.6 其他

管理的集群大小为小于等于四个；工具响应时间应该小于五秒，总体处理时间视数据量而定；系统稳定，当出现故障能自动尝试重连并记录日志；移植性好，多平台通用。

2.2 系统环境



第3章 系统用例

3.1 用例参与者

3.1.1 RDBS 和 HBASE

支持以 JDBC、ODBC 或者 sql 方式连接的数据库，可连接数据库集群

3.1.2 工具使用人员

支持配置文件配置、命令行操作，实时过程反馈和日志反馈

3.1.3 维护人员

工具日常维护，日志维护

3.2 用例简介

被连接的数据库指定 ip 地址，数据库用户需要密码连接

- ◆ 事件流
 - 1: 工具读取配置文件中的错误密码
 - 2: 工具使用错误密码无法连接到数据库
- ◆ 后置条件

控制台输出错误信息、提示用户重新配置，并写入操作日志。

3.4 集群管理

3.4.1 集群的连接

- ◆ 前置条件

工具读取配置文件中的集群信息
- ◆ 事件流
 - 1: 工具读取配置文件中的集群信息参数
 - 2: 工具根据参数进行集群连接
- ◆ 后置条件

工具根据其他参数对集群中的数据表进行同步操作

3.4.2 集群的增加

- ◆ 前置条件

工具读取配置文件中的新增集群信息
- ◆ 事件流
 - 1: 工具读取用户输入的新增的集群信息参数
 - 2: 工具根据参数进行新增集群连接
- ◆ 后置条件

反馈连接状态，工具根据其他参数对集群中的数据表进行导出或者同步操作

3.4.3 集群的删除

- ◆ 前置条件

工具读取配置文件中的要删除的集群信息参数
- ◆ 事件流
 - 1: 工具读取配置文件中的需要删除的集群信息删除集群信息参数
 - 2: 工具根据参数进行集群删除
- ◆ 后置条件

反馈删除状态，工具根据其他参数对集群中的数据表进行导出或者同步操作

3.5 日志和中间文件管理

3.5.1 日志写入

- ◆ 前置条件

工具读取配置文件中的集群信息

- ◆ 事件流

- 1: 工具读取配置文件中的集群信息参数

- 2: 工具根据参数进行集群连接

- ◆ 后置条件

工具根据其他参数对集群中的数据表进行导出或者同步操作

3.5.2 中间文件的写入

- ◆ 前置条件

工具读取需要导出的关系数据表完成，或者捕捉关系数据表的变化成功

- ◆ 事件流

- 1: 工具获取需要导出的表的数据，或者获取变化的数据表的数据和相关信息

- 2: 工具将信息通过处理模块处理之后得到数据

- 3: 工具将数据存储到 XML 中间文件中

- ◆ 后置条件

中间文件被继续传输到中间文件读取模块

3.5.3 中间文件的读取

- ◆ 前置条件

工具的存表模块接收到 XML 中间文件

- ◆ 事件流

- 1: 工具读取 XML 中间文件中的信息和数据

- ◆ 后置条件

工具将读取的数据根据相关信息存入 HBASE

第4章 关键问题和解决方案

4.1 变化捕捉策略选择和分析

4.1.1 问题描述

关系数据库变化捕捉和同步策略之间联系紧密，因此一个准确高效的变化捕捉策略是必不可少的，而在市面上有各种解决方案供我们选择。通过比较，我们选择以下解决方案。

4.1.2 选择方案：杨鹏的论文：异构数据库同步中间件的设计和实现[4]

- ◆ 概述

该论文中提出了一种基于触发器和专用数据表（自定义的日志表）的数据变化捕捉方法：日志表法。通过触发器将同步所需的变更控制信息记录在统一的表中，这张存有变更控制信息的表被称日志表，在同步时，根据日志表提供的同步控制信息，对更改的数据进行同步。

- ◆ 详细描述（捕捉方法设计）

4.1.5.1 日志表设计

日志表需要记录变更数据项的主键值、变更数据项所在表名称、变更类型、数据变更时间并标志出同步表中哪些字段发生更新。

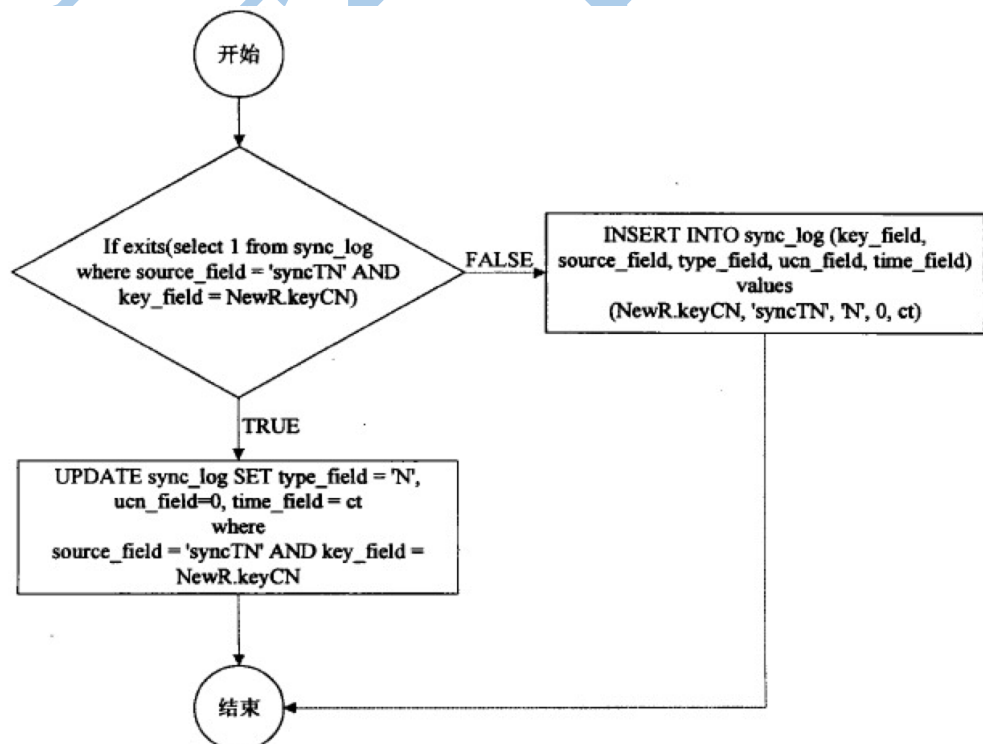
字段	数据类型	记录内容	备注
key_field	varchar(32)	变更数据项主键	
source_field	varchar(32)	数据项所在表名称	
time_field	varchar(1)	变更类型	U/N/D
type_field	datetime	变更时间	系统时钟
ucn_field	int	列更新信息	默认为 0

4.1.5.2 触发器设计

在同步表上建立触发器,针对 INSERT、UPDATE 和 DELE 住操作,触发器完成不同工作,将必要的控制信息记录在日志表中。由于不同数据库中触发器语法不尽相同,流程图中使用伪代码,其中 synceslog 为日志表名称;syncTN 为同步表名称;keycN 为同步表主键字段名称;oldR 表示变更前的记录;NewR 表示变更后的记录;ct 为当前时间。

4.1.5.2.1 新建记录触发器

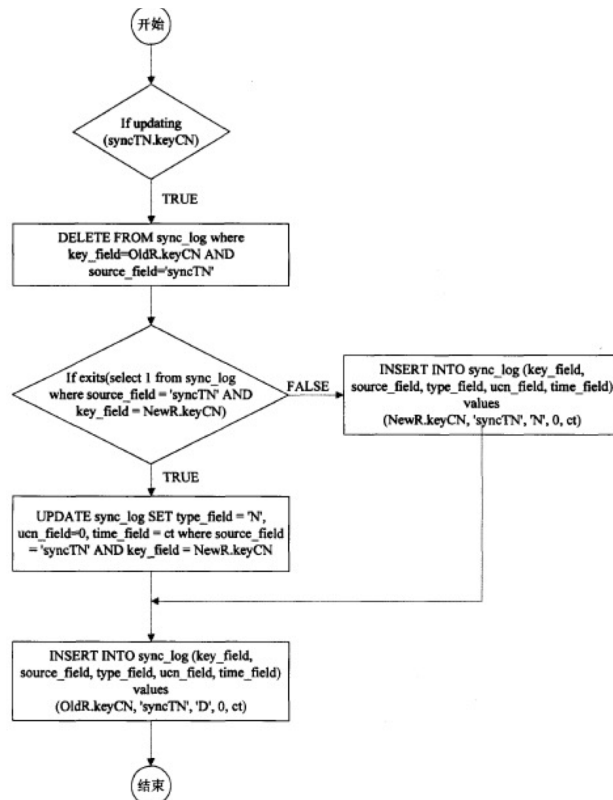
在同步表上执行 INSERT 操作时,触发器的工作流程如下图所示。DML 操作新建一条记录时,针对 INSERT,操作的触发器将新建记录的主键值记入日志表的 key_field 字段;将表名称记入 source_field 字段;将变更状态 “N” 记入 tyPe_field 字段;将操作时间记入 time_field 字段。如果日志表内已经存在该条记录的控制信息(即 key_field 字段内的值等于该记录的主键且 source_field 字段记录的值等于表名称),则使用本次 DML 操作的控制信息更新日志表中的记录。工作流图如下:



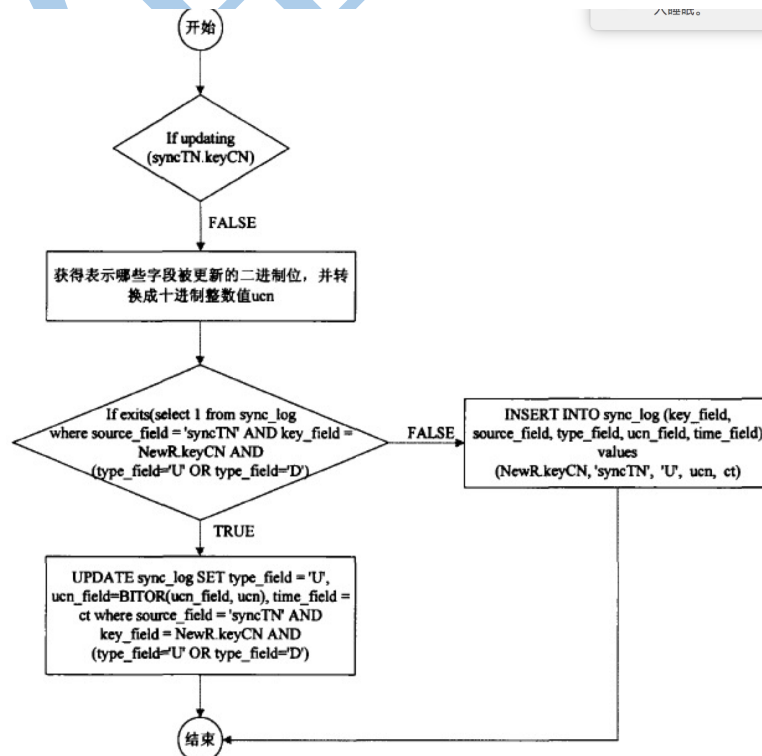
4.1.5.2.2 更新记录触发器

在同步表上执行 uPDATE 操作时,触发器首先判断发生更新的字段是否为主

键,若主键发生更新,则先删除日志表中所有与旧主键值对应的日志记录,然后记录下两条新的控制信息,一条标志出旧主键值对应的记录被删除,另一条标志出新主键值对应的记录被插入, workflows 如下:



若非主键字段发生更新,触发器 workflows 如下图所示。其中 BITOR()函数表示对参数进行位或运算



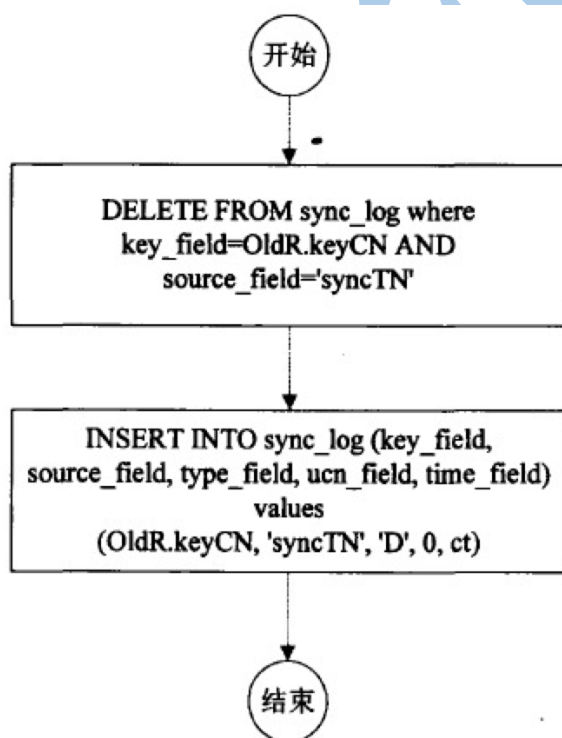
上图的判断分支中增加了 $type_field=u$ 的条件,这使同步表中的一条记录可能对应两条日志记录,这两条日志记录的状态分别是 N 和 U。同一条记录产生的 U 状态控制信息不会覆盖之前的 N 状态控制信息。一条记录的多次更新操作只在日志表内对应一条状态为 u 的控制信息,每次更新操作被累积记录于 ucn_field 字段,更新的字段以二进制位的形式标记,转换为十进制数值后记录于 ucn_field 字段。将 ucn 记入 $ucnfield$ 字段时,要将 ucn 与该字段内的旧值做位或运算。位或运算将多次更新操作所更改的字段累积记录下来,而不是为每次更新操作都保存一条日志记录,减少了对数据库空间的占用。

用来标识字段变更的 ucn 变量可以通过下述步骤获得:

- (1) 令 ucn 等于 0。
- (2) 逐个判断记录中各个字段的值是否发生更新,若发生更新,则从数据库系统表内获得该更新字段对应的 ID 值 id ,令 $ucn=BITOR(ucn,z'(id-1,))$

4.1.5.2.3 删除记录触发器

在同步表上执行 DE 几 LE 操作时,触发器的工作流程如图 3.4 所示。DML 语句删除表内一条记录时,此前日志表内该记录对应的所有控制信息都已经没有意义,触发器首先删除日志表内该记录的所有旧控制信息,然后新建状态为 D 的对应控制信息。



4.1.3 变化捕捉方法总结

综上: 我们采用 4.1.5 中提出的变化捕捉方法。借助日志表中记录的控制信息,可以方便的找到变更数据项并得到最新的数据值。

日志表法变化捕捉方法与一般日志信息记录方式不同,并不是对每一条 DML 操作都记录下一条日志信息,日志表内日志信息的最大数量为同步表内记录数的 2 倍,不会随用户操作数目的增加而无限制的增加。且日志表变化捕捉方法只在日志表中记录变更控制信息,没有记录具体的变更数据值,减少了对数据库空间的占用。

由于绝大多数的数据库都支持触发器,使基于触发器的变化捕捉方法有很强的通用性,在同步表上建立触发器不会改变用户数据结构与字段数据类型,日志表变化捕捉方法与应用无关,使异构数据库同步中间件系统与用户应用之间为松耦合关系,充分保证了用户应用的自治性,使同步系统具有广泛的适用性。

4.2 异构处理方法

4.2.1 数据库平台异构问题处理

4.2.1.1 问题描述

解决异构数据库 RDBS 和 HBASE 导出和同步时数据库间的异构问题

4.2.1.2 解决方案

为解决平台异构,必须为异构数据库同步中间件系统寻找一种具有跨平台性的实现语言。Java 通过其虚拟机机制(JVM)获得优良的跨平台性,在网络服务、应用、平台无关的桌面应用以及嵌入式设备等应用领域中被广泛使用。因此作为一种面向对象、平台无关且支持多线程的编程语言,Java 成为异构数据库同步中间件系统实现的必然选择

4.2.2 数据类型异构处理

4.2.2.1 问题描述

数据库厂商及标准化组织联合颁布过一系列的 SQL 标准(例如 SQL_86、SQL_89、SQL_92 等),现在大多数数据库开发商支持 SQL_92 标准,但并不是完全符合这个标准。每一个数据库开发商都试图扩展这个标准,以增强产品的功能,提高产品的吸引力,例如 Oracle 的 PL/SQL,MS-SQLServer 的 Transact_SQL。HBASE 是基于列存储的,数据结构也和上述数据库不相同,这些导致了在导出和同步的时候必须要考虑数据库数据类型的异构。

4.2.2.2 解决方案分析

不同数据库厂商的异构数据库产品支持不同的数据类型,为转换异构数据类型以实现异构数据库数据之间的数据共享及同步,蒋敏提出将异构数据类型统一转换为 SQL_92 标准数据类型方法,者敬提出将异构数据类型统一转化为已有的 C 语言数据类型。他们都有缺陷:

1、考虑的异构数据类型不够全面。各种方法所考虑的数据类型都为数据库中常见的数据类型,没有考虑异构数据库中各种真正需要转换的特有数据类型。

2、转换目标数据类型有局限性不能完全满足待转换的各个数据类型的需要。在上述方法中无论是 SQL_92 标准中的数据类型、已有的 C 语言数据类型还是自定义的中间数据库类型都只能完成异构数据库中部分数据类型转换的需要。

3、转换过程不完全,没有考虑转换后的数据在网络中传输和在异构平台上使用的问题。由于网络传输和异构软硬件平台带来的编码解码问题,同类数据在转换为统一数据类型后还需进一步转化成能够在网络中正确传输和还原的数据

4.2.2.2 关系数据库数据类型到 java 数据类型 (待完善)

JDBC 数据类型	整数值	Oracle 9i	MS-Sqlserver	Mysql	HBASE	Java 数据类型

4.2.2.3 java 数据类型转换（待完善）

4.2.2.4 数据类型还原到 HBASES（待完善）

4.3 RDBS 数据依赖解决（表模式转换）

4.3.1 问题描述

在处理关系数据库数据的时候，由于关系数据库数据存在主键外键等依赖，必须将其进行必要的处理，做到表模式的转换，才能合理妥当的存入 HBASE。

4.3.2 解决方案

杨寒冰[4]提出的四种表模式变换方法能解决上述问题

4.3.2.1 基本变换

（1）适用场景

以传统关系数据库中表设计模式定义的表变换成 HBase 中的表。

（2）变换方法

Goods 表为表 1，把表 1 的表名作为 HBase 中对应的表名，表 1 也添加到这个表的列族中。之后把表 1 的主键作为对应表的行键。最后把表 1 中定义的所有属性添加到对应表的列族 1 中。



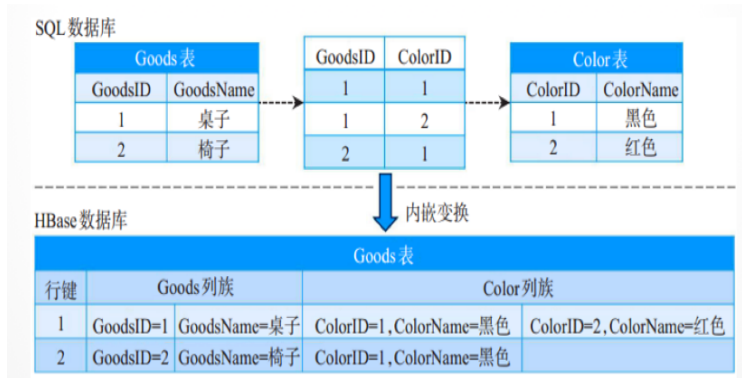
4.3.2.2 内嵌变换

（1）适用场景

查询表 1 中的某一行时就能够得到同这一行所关联的所有表 2 中的数据，这样就可以避免表 1 和表 2 的连接查询。

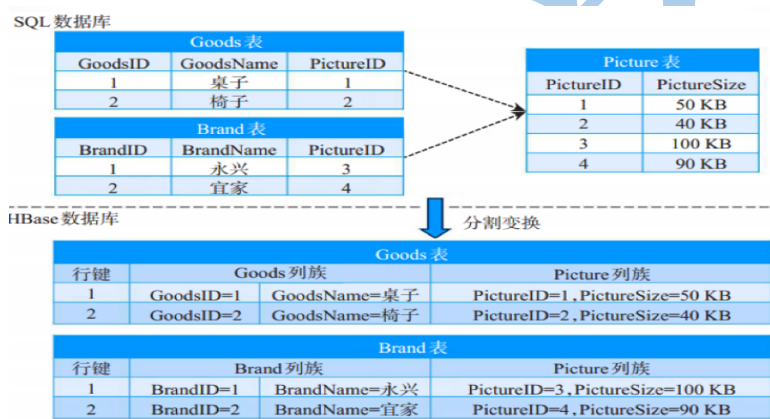
（2）方法

Goods 表为表 1、Color 表为表 2，表 1 的表名作为 HBase 中对应的表名；把表 1 和表 2 添加到列族中。表 1 的主键作为对应表的行键，随后把表 1 中定义的所有属性添加到对应表的列族 1 中，把表 2 中所有同表 1 这一行有关的所有行属性都放入到列族 2 中。根据情况删除表 2。



4.3.2.3 分割变换

当从表 3 与其余表都存在 1 对 1、1 对多或多对多的关系时，可分割表，按照映射关系把表 3 中的每一行内嵌到对应的表中。当查询表 1 或 2 中的某行时就能得到同这一行所关联的所有表 3 的数据，避免表 3 同其余与之有联系的所有表之间的连接查询。



4.3.2.4 内联变换

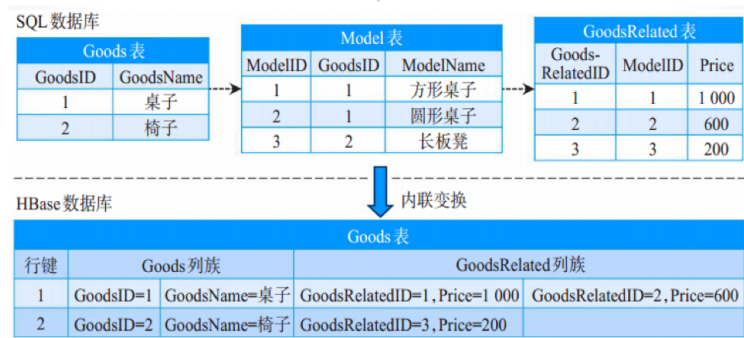
(1) 适用场景

从表 1 到表 2 存在 1 对 1、1 对多或多对多关系，且表 2 到表 3 也存在这些关系时，把表 3 作为表 1 的内联表。当查询表 1 中的某一行时，得到同这一行通过表 2 所间接关联的所有表 3 中的数据，这样就可以避免表 1、表 2 和表 3 这三张表之间的连接查询。

(2) 方法

表 1:Goods(主表)、表 2:Model、表 3:GoodsRelated。

把表 3 中所有同表 1 这一行通过表 2 有间接关联的所有行属性都放入到列族 2 中。



4.4 RDBS 数据库的连接维护、中断续传处理、中间文件处理

4.4.1 问题描述

- 1: 数据库连接中断或者连接不上，网络故障导致的无法连接。
- 2: 同步遇到故障导致同步进行到一半的处理方法
- 3: 中间文件日志表的备份和删除处理

4.4.2 解决方案

- 1: 当遇到数据库连接故障，记录进系统日志并立即重试。并在控制台进行故障和原因输出，要求用户检查网络或者数据库配置。如果一直未成功，终止程序并记录日志。
- 2: 由于工具采用的是中间文件同步方式，每个中间文件有大小限制（100 万条数据为界限），如果同步中断，则保存下最后一条同步处的信息，在下次的时候，再次读取中间文件到上次记录的位置进行同步。
- 3: 每个中间文件同步完成之后，备份到其他目录并删除这个中间文件。

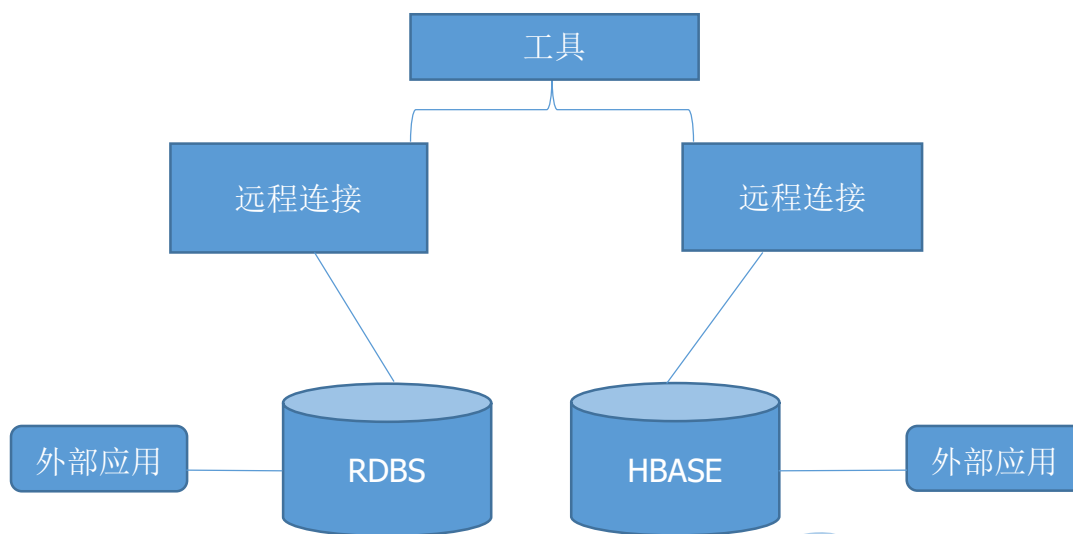
第5章 系统结构

5.1 静态结构

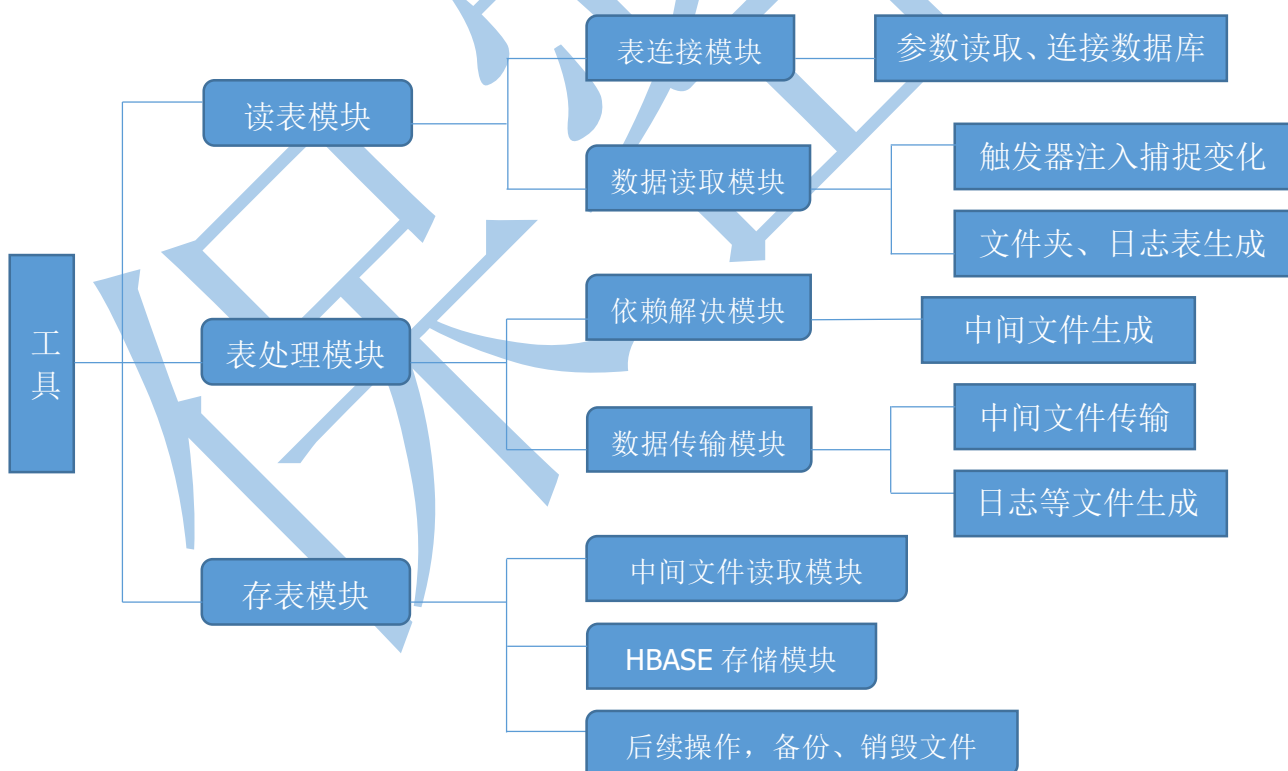
5.1.1 概述

该工具在设计实现过程中遵循如下设计思想:

- (1) Internet 环境中,实现可配置的手动导出与自动同步机制,保证可靠性前提下的成本有效实现。
- (2) 充分保证用户应用的自治性,在不影响原用户应用系统的前提下,于后台以中间件方式完成数据同步。
- (3) 基于 Java 虚拟机,可跨操作系统平台使用,具有精简同步功能集且便于扩展的异构数据库同步中间件平台软件。



工具的模块分解：



5.1.2 读表模块

本模块分层如上图所示，完成功能：

- (1) 接收配置文件中的参数，根据参数进行 RDBS 连接和集群管理，并将 HBASE 参数信息根据需要存入要传输的数据交给表处理模块。

- (2) 读取需要导出的 RDBS 数据表数据交给表处理模块
- (3) 监控需要同步的 RDBS 表，将同步信息交给表处理模块

5.1.3 表处理模块

接收从读表模块传输过来的相关数据信息和 HBASE 信息，解决表数据依赖，生成中间文件传输给存表模块。

5.1.4 存表模块

本模块分层如上图所示，功能如下

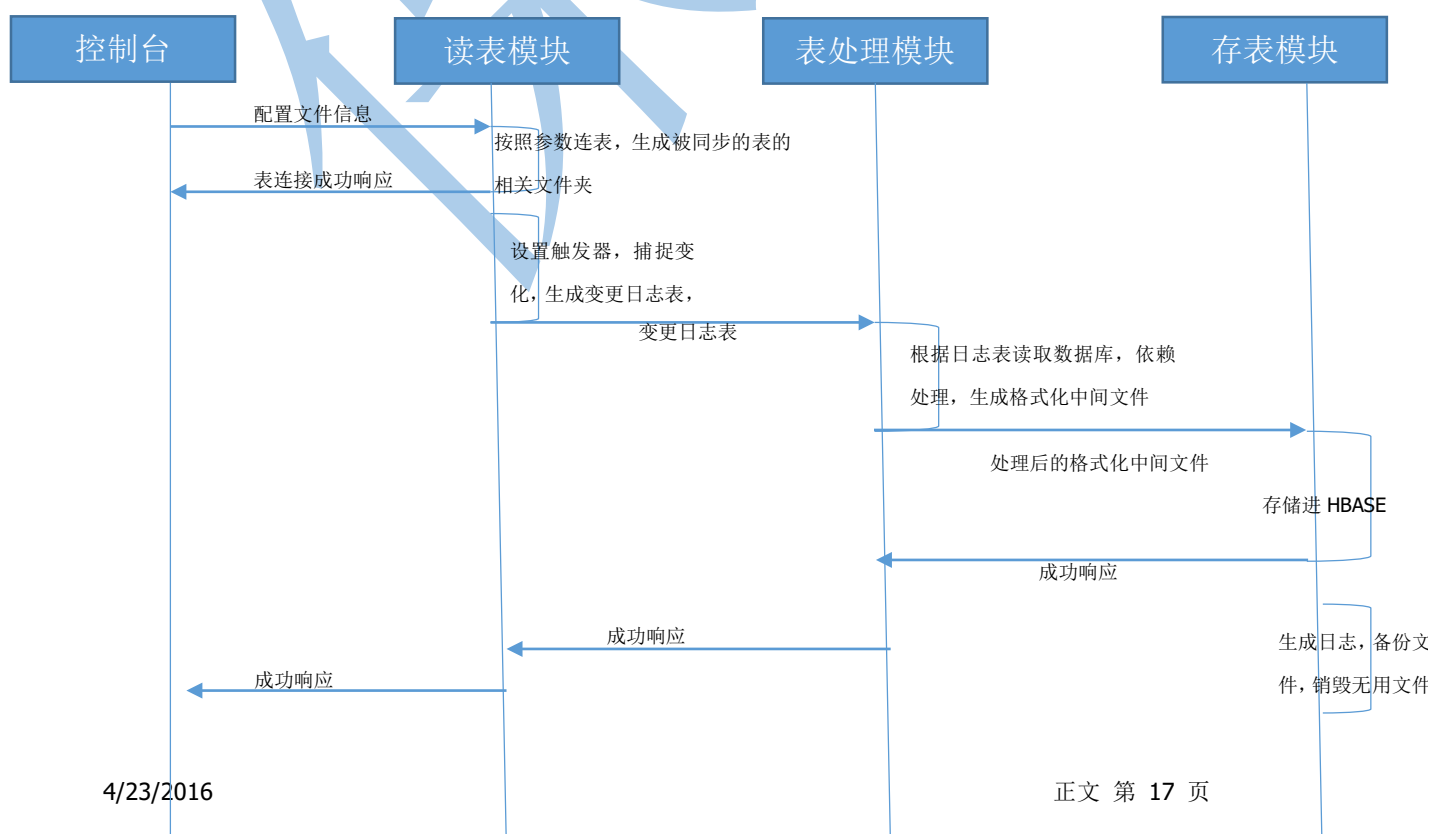
- (1) 接收从表处理模块传输的文件，进行 HBASE 处理
- (2) 根据从表连接模块传输的 HBASE 信息，进行 HBASE 集群管理。

5.1.5 内部接口

- 1: 表连接模块和 sql 数据库，读表模块，: JDBC/ODBC
- 2: 表连接模块和 HBASE，表存储模块: HADOOP
- 3: 读表模块和表处理模块: 表处理模块调用读表模块的 API，得到数据表信息和需要同步的数据。
- 4: 表处理模块和存表模块: 存表模块调用表处理模块的 API，获得需要存往 HBASE 的数据等信息。

5.2 动态结构

5.2.1 一次同步操作过程



第6章 数据结构

描述系统的数据模型、数据库表定义、配置文件定义。

6.1 数据模型

配置文件采用格式化的类 xml 格式文件，每个元数据项代表特定的含义。

中间传输文件采用 xml 文件，格式化的数据易于管理和传输。

日志文件采用普通的一行一行的信息写入，便于节省空间。

6.2 数据库表

6.2.1 配置文件参数设置(xml 文件)

```
<config>
  <RDBS>//关系数据库表
    <database>//表 1
      <type>mysql</type>//数据库类型
      <address>127.0.0.1:3306</address>//地址
      <username>admin</username>//用户名
      <password>000000</password>//密码
      <table>userlist</table>//要同步的表
      <timespace>2016/3/11 12:31:30 -- 2016/3/21 12:31:20</timespace>//同步的时间
    </database>
    <database>
      <type>sqlserver</type>
      <address>127.0.0.1:3306</address>
      <username>admin</username>
      <password>000000</password>
      <table>userlist</table>
      <timespace>2016/3/11 12:31:30 -- 2016/3/21 12:31:20</timespace>
    ...
  </database>
</RDBS>
<HBASE>
  <database>
    <address>127.0.0.1:3306</address>
    <username>admin</username>
```

```

    <password>000000</password>
  </database>
  ...
</HBASE>
</config>

```

6.2.2 log 记录的每次导出/同步信息字典

参数名	time	Username	dbaddress	tablename	affected
解释	操作时间	操作用户	被操作数据库地址	备操作的表	操作详情
示例	2016/3/11 12:31:30	Admin	127. 0. 0. 1	Table01	1002rows affected

6.2.3 每次故障的记录日志

参数名	time	user	dbaddress	tablename	reason
解释	错误发生过时间	造成错误的用户	错误存在的表的地址	错误存在的表名	错误原因
示例	2016/3/11 12:31:30	Admin	127. 0. 0. 1	Table01	Timeout

6.2.4 中间文件（日志表）

字段	数据类型	记录内容	备注
key_field	varchar(32)	变更数据项主键	
source_field	varchar(32)	数据项所在表名称	
time_field	varchar(1)	变更类型	U/N/D
type_field	datetime	变更时间	系统时钟
ucn_field	int	列更新信息	默认为 0

第7章 系统性能

7.1 性能需求

系统的数据量级：HBASE 数据量在 TB 级别，RDBS 低于千万级别

响应时间：接收并开始处理时间：小于 3 秒；总处理时间：视数据量而定

同步性能：每秒达到至少万条信息的同步，总共处理时间不能超过 30 分钟，超时会我询问用户是否继续。

7.2 差错控制

错误类型	原因	解决办法
数据库连接错误	数据库设置不正确或者	取消本次操作，报告错

	数据库本身出错	误，并让用户修正错误后重试
用户参数输入错误	输入参数不规范	取消本次操作并提醒用户重试
其他操作错误	用户不正当操作造成的错误	取消本次操作并提醒用户正确的操作
未知错误	未知异常	记录日志，并提醒用户重启或者联系维护人员

第8章 参考资料

【1】者敬,开放式异构数据库复制框架的研究与实现,中国科学院软件研究所博士学位论文,2004

【2】蒋敏,基于网络隔离的异构数据库同步技术的研究与实现,浙江大学硕士学位论文,200

【3】沈敏,许华虎,季永华等,基于 XML 的分布式异构数据库数据同步系统的研究,计算机工程与应用,2005,41(5):184 — 186

【4】杨鹏，异构数据库同步中间件技术的研究与实现，国防科大研究生论文，2007

— 完毕 —