杭州银行
分布式批量平台(HZBAT)

厉华

版本修订

版本	日期	修订人	内容
0.1.0	2015-07-15	厉华	创建文档
0.2.0	2015-08-27	厉华	新增集群管理相关内容

目录索引

1 概述

杭州银行分布式批量平台(以下简称 HZBAT)是基于开源分布式计算框架 (以下简称 DC4C)的分布式集群化、高可靠性高伸缩性、自主研发的批量平台。

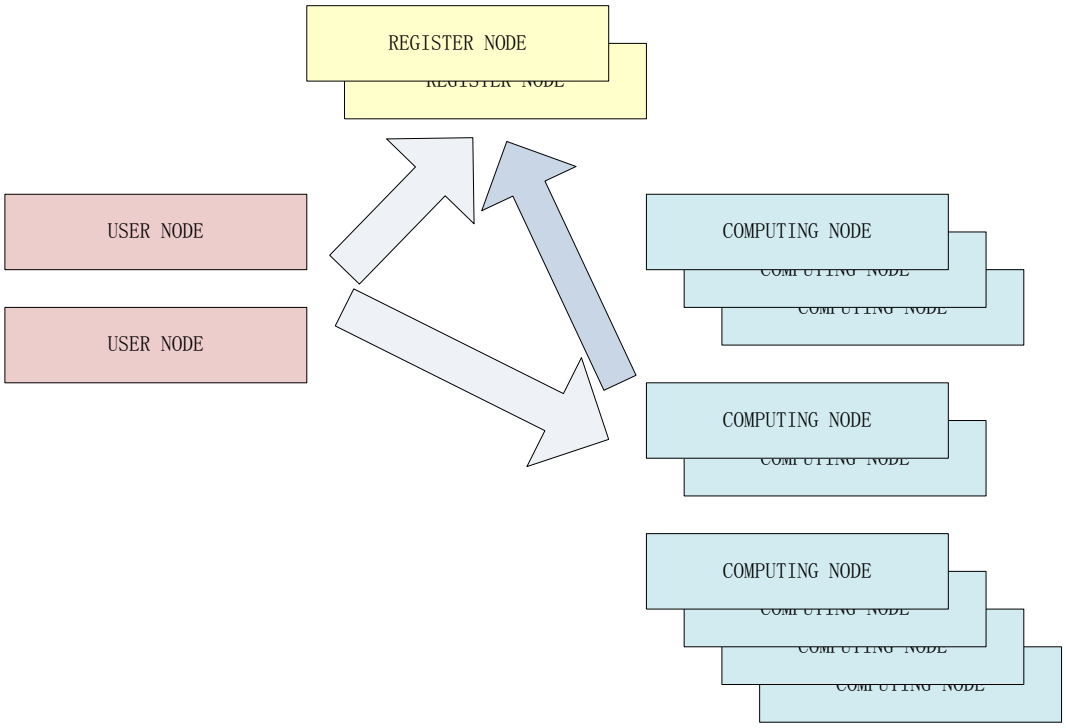
DC4C 实现了分布式集群部署、管理和任务分发，其中 DAG 任务调度引擎提供了有向无环图批量树的通用调度。

HZBAT 根据 DAG 任务调度引擎提供的接口，重载了装载配置、更新任务、更新批量、卸载配置更新计划等回调接口，实现了从我行环境的数据库中装载配置、更新状态回数据库等功能，并编译成独立可执行程序，便于直接执行或脚本调用。

1.1 DC4C 简介

DC4C 体系结构包含**基础平台架构**、**用户 API 包**和**有向无环图（下面简称 DAG）的任务调度引擎**。

基础平台架构包含三类节点：**注册节点**、**计算节点**和**用户节点**。



注册节点（守护进程）：负责接受计算节点注册、状态变更、注销；接受用户节点查询空闲计算节点；接受 telnet 连接在线查询和管理。进程框架为父子进程监控进程异常，可以同时起多对保持计算节点注册信息冗余，提高可靠性。

计算节点（守护进程）：负责向注册节点注册；接受用户节点分派任务并反馈执行结果；随时向注册节点报告状态。进程框架为父进程+子进程组（计算节点组）监控进程异常。

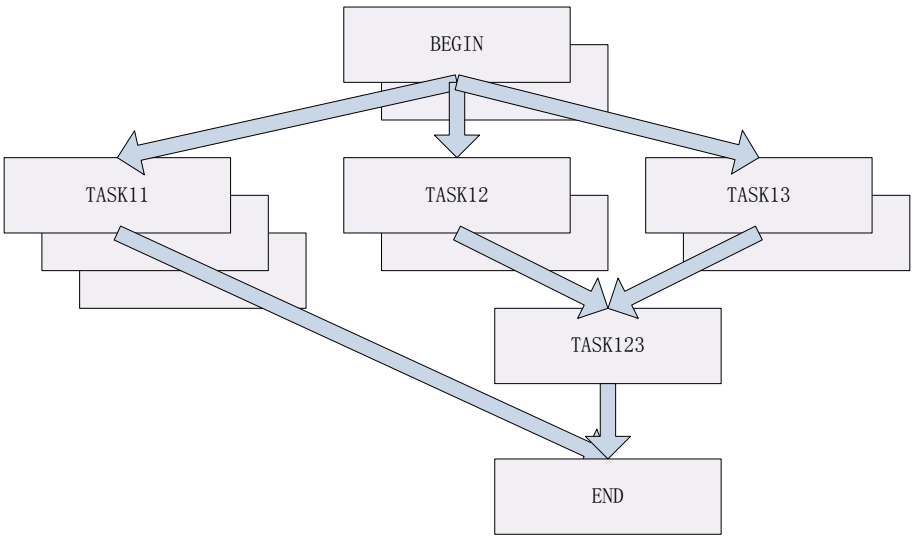
用户节点（可以是守护进程、命令行进程或其它任何类型的用户自己控制的进程）：用户程序调用用户 API，向注册节点查询当前空闲的计算节点，分派任务给计算节点并监督执行。

计算节点启动后向**注册节点**注册后保持长连接，并互相心跳。

用户节点用户应用通过分派任务 API 向**注册节点**查询当前空闲的**计算节点**集合，然后发送执行命令行和执行程序 MD5 给**计算节点**，**计算节点**校验执行程序 MD5，如果不匹配，联动请求**用户节点**分发新版程序，然后再执行命令。

用户 API 包供用户节点和计算节点应用调用，实现**用户节点**的任务分派、计算节点应用控制等功能。

有向无环图(DAG)任务调度引擎封装了基础平台架构提供的基本任务分派功能，实现了有向无环图数据结构的任务流的执行控制，便于用户直接搭建复杂任务依赖关系调度平台。



（DC4C 更多资料请参阅其产品文档）

2 数据库表结构

2.1 计划表

TABLE NAME		dag_schedule		
TYPE	LENGTH	NAME	DESC	REMARK
INT	4	order_index	顺序索引	NOT NULL
STRING	64	schedule_name	计划名称	NOT NULL
STRING	256	schedule_desc	计划描述	
STRING	19	begin_datetime	开始执行日期时间	
STRING	19	end_datetime	结束执行日期时间	
INT	4	progress	进度	DC4C_DAGSCHEDULE_PROGRESS_INIT 0 DC4C_DAGSCHEDULE_PROGRESS_EXECUTING 1 DC4C_DAGSCHEDULE_PROGRESS_FINISHED 2 DC4C_DAGSCHEDULE_PROGRESS_FINISHED_WITH_ERROR 4
UNIQUE INDEX1 : order_index UNIQUE INDEX2 : schedule_name				

2.2 批量表

TABLE NAME		dag_batches_info		
TYPE	LENGTH	NAME	DESC	REMARK
STRING	64	schedule_name	计划名称	NOT NULL
STRING	64	batch_name	批量名称	NOT NULL
STRING	256	batch_desc	批量描述	
INT	4	view_pos_x	图形编辑坐标 X	
INT	4	view_pos_y	图形编辑坐标 Y	
INT	4	interrupt_by_app	是否允许应用中断	1:interrupt 0:not interrupt
STRING	19	begin_datetime	开始执行日期时间	
STRING	19	end_datetime	结束执行日期时间	
INT	4	progress	进度	DC4C_DAGBATCH_PROGRESS_INIT 0 DC4C_DAGBATCH_PROGRESS_EXECUTING 1 DC4C_DAGBATCH_PROGRESS_FINISHED 2 DC4C_DAGBATCH_PROGRESS_FINISHED_WITH_ERROR 4

STRING	256	pretask_program_and_params	准备任务 执行命令行	
INT	4	pretask_timeout	准备任务 超时时间（秒）	
INT	4	pretask_progress	准备任务 进度	DC4C_DAGTASK_PROGRESS_INIT 0 DC4C_DAGTASK_PROGRESS_EXECUTING 1 DC4C_DAGTASK_PROGRESS_FINISHED 2 DC4C_DAGTASK_PROGRESS_FINISHED_WITH_ERROR 4
INT	4	pretask_error	准备任务 执行系统响应码	
INT	4	pretask_status	准备任务 执行应用响应码	
UNIQUE INDEX1 : schedule_name , batch_name				

2.3 批量依赖关系表

TABLE NAME		dag_batches_direction		
TYPE	LENGTH	NAME	DESC	REMARK
STRING	64	schedule_name	计划名称	NOT NULL
STRING	64	from_batch	前依赖批量名称	当为""时表示无前依赖
STRING	64	to_batch	后依赖批量名称	当为""时表示无后依赖 如： 批量 A -> 批量 B -> 批量 C 配置如下： 前依赖批量：批量 A 后依赖批量：批量 B 前依赖批量：批量 B 后依赖批量：批量 C
INDEX1 : schedule_name , from_batch				

2.4 批量任务表

TABLE NAME		dag_batches_tasks		
TYPE	LENGTH	NAME	DESC	REMARK
STRING	64	schedule_name	计划名称	NOT NULL
STRING	64	batch_name	批量名称	NOT NULL
INT	4	order_index	顺序索引	NOT NULL
STRING	256	program_and_params	执行命令行	NOT NULL
INT	4	timeout	执行超时	
STRING	19	begin_datetime	开始执行日期时间	

STRING	19	end_datetime	结束执行日期时间	
INT	4	progress	进度	DC4C_DAGTASK_PROGRESS_INIT 0 DC4C_DAGTASK_PROGRESS_EXECUTING 1 DC4C_DAGTASK_PROGRESS_FINISHED 2 DC4C_DAGTASK_PROGRESS_FINISHED_WITH_ERROR 4
INT	4	error	执行系统响应码	
INT	4	status	执行应用响应码	
UNIQUE INDEX1 : schedule_name , batch_name , order_index				

2.5 批量过滤表

TABLE NAME		dag_batches_filter		
TYPE	LENGTH	NAME	DESC	REMARK
STRING	64	schedule_name	计划名称	NOT NULL
STRING	64	batch_name	批量名称	NOT NULL
STRING	16	filter_type	过滤类型	NOT NULL "DD" : 按每月指定日执行 "MM-DD" : 按每年指定月日执行 "WDAY" : 按每周星期几执行
STRING	1024	filter_param	过滤参数	"DD" : 01-31 或月初"MB"或月末"ME"，多个参数用','分隔，如 "18"或"10,20,30"或"MB" "WDAY" : 1-7
UNIQUE INDEX1 : schedule_name , batch_name				

3 系统设计

DAG 基于 DC4C 基础平台架构和任务分发、HZBAT 基于 DAG 调度引擎框架重载了从数据库装载配置和实时更新状态回数据库的回调接口。

3.1 三层概念

批量平台里分三层概念：**计划**、**批量**和**任务**。

任务是最小执行单元，一个任务即一个执行命令行，即一个执行进程。在

DAG 调度引擎中，任务被分发到 DC4C 分布式集群（计算节点组群）中空闲的计算节点上，由该计算节点负责创建进程、执行该命令行、监控执行过程、反馈执行结果（进程返回值）回用户节点（任务分发方）。

批量由一组任务构成，任务之间平等没有优先级，可以达到完全并发。

计划即有向无环图批量集合，每个批量是该图上的一个节点，批量与批量之间有先后依赖关系，只有所有前置批量都执行完成后才能开始其依赖后续批量。

3.2 用户节点主控 hzbat

3.2.1 命令行语法

```
$ hzbat
hzbat v1.5.3 build Jul 14 2015 16:18:20
USAGE : hzbat business_date [ (SCHEDULE_NAME) [ (BATCH_NAME) ] ]
                                [ ALL ]
```

不带参数执行运行 hzbat 可以看到版本号、编译时间以及命令行参数说明。

第一个参数是业务日期，用于根据批量过滤表做日期过滤。

第二个参数是计划名，用于查询计划表以及其它表装载配置用。当写成”ALL”时则依次执行所有计划 ID 大于等于 0 的所有计划，知道成功完成或失败中断。

第三个参数是批量名，可选参数，当存在时，仅执行指定计划中的指定批量。

3.2.2 调度流程

DAG 任务调度引擎装载计划配置，按照前置依赖、后续依赖关系依次并行执行批量，批量中的任务也是并发执行，任务分发至 DC4C 基础平台架构部署的最大并发量（计算节点数量）。

如果所有任务都成功结束，DAG 图全部执行完成。

如果某个任务返回失败（error 系统错误或 status 应用错误），且所属批量设置了不忽略错误标志，DAG 调度引擎等待正在执行的所有批量任务执行结束后中断计划执行，让人工介入处理。人工处理完后继续执行前面未成功任务和之前未

开始任务。

3.2.3 代码框架伪代码

```
main()
    根据命令行参数分支
        按序执行计划 ID 大于等于 0 的所有计划
        执行单个计划
        执行单个批量

按序执行计划 ID 大于等于 0 的所有计划()
    查询计划表中计划 ID 大于等于 0 的所有计划
        执行单个计划

执行单个计划()
    执行批量表中的准备任务集合，更新准备任务状态回数据库
        读取计划表中该计划所有准备任务，构造任务集合，同时根据批量过滤表筛选
        调用 DC4C 开始批量任务函数
        调用 DC4C 处理批量任务函数，更新准备任务状态回数据库
    从数据库装载任务集合
        读取计划表、批量表、批量依赖关系表、批量任务表、批量过滤表填充 DAG 大配置结构
        调用 DAG 解析大配置结构构造有向无环图运行时配置
    调用 DAG 开始计划函数
    调用 DAG 处理计划函数，回调接口更新任务状态回数据库
    卸载配置更新状态回数据库

执行单个批量()
    查询批量表该批量准备任务
    调用 DC4C 执行任务函数
    读取批量任务表中的任务集合，同时根据批量过滤表筛选
    调用 DC4C 开始批量任务函数
    调用 DC4C 处理批量任务函数，更新准备任务状态回数据库
```

3.3 批量任务管理界面

批量任务管理界面用于查看、编辑和监控批量平台中的所有计划、批量和任务配置。

（因该界面未开发，目前只能手工维护 sql 文件导入数据库，以下仅为技术需求）

界面主要分左(约 20%)、右(约 80%)两块，左边是计划列表，右边是计划监控、编辑区域。

左边的计划列表（读取数据库表“计划表”）最后是新增计划小表单：行输入框“计划名”、按钮“新增计划”。左边最下面是删除计划小表单：行输入框“计划名”、按钮“删除计划”，删除计划前需手工输入计划名，防止误删。

点击左边计划名后右边显示该计划的批量有向图环图（读取数据库表“批量表、批量依赖关系表”）。"INSERT"键添加批量节点，左键单击选定批量节点，左键拖动已有批量节点调整坐标，右键拖动新增依赖关系箭头， "DELETE"键删除已选定批量节点。左键双击批量节点弹出批量信息配置小窗口。右键双击切换监控（默认）、编辑模式。监控模式以颜色表达该批量节点状态（对应“批量表”progress 字段），浅蓝色为初始状态，黄色为正在执行中，绿色为成功结束，红色为失败结束。批量信息配置小窗口分为上下两区域(约 90%,10%)，上区域维护准备任务（命令行、超时时间、开始执行时间、结束执行时间、状态、系统响应码、应用响应码），查看任务信息（命令行、超时时间、开始执行时间、结束执行时间、状态、系统响应码、应用响应码），下区域维护批量过滤信息（过滤类型、过滤参数）。建议计划监控中可每十秒钟读取数据库刷新。

4 部署方案

4.1 综合积分系统日终批量（生产环境）

4.1.1 集群部署

节点类型	数量	网络地址和服务端口
注册节点	2*1 对	pointbat@168.68.3.121:16001 pointbat@168.68.3.122:16001

计算节点	2*10 个	pointbat@168.68.3.121:17001~17010 pointbat@168.68.3.122:17001~17010
用户节点	-	pointbat@168.68.3.121 pointbat@168.68.3.122

复制 DC4C 自带管理脚本 dc4c.do 到 168.68.3.121@pointbat，修改（红色部分）如下：

```
    start)
        call "dc4c_rserver -r 168.68.3.121:16001 $*"
        call "dc4c_wserver -r 168.68.3.121:16001,168.68.3.122:16001 -w
168.68.3.121:17001 -c 10 $*"
        ;;
```

启停 dc4c：

```
$ dc4c.do start
$ dc4c.do stop
```

在 168.68.3.122@pointbat 里的 dc4c.do 修改如下：

```
    start)
        call "dc4c_rserver -r 168.68.3.122:16001 $*"
        call "dc4c_wserver -r 168.68.3.121:16001,168.68.3.122:16001 -w
168.68.3.122:17001 -c 10 $*"
        ;;
```

启停 dc4c：

```
$ dc4c.do start
$ dc4c.do stop
```

设置 dc4c 集群 rsh 配置：

```
$ cat .rhosts
168.68.3.121        dc4c
$ ls -la .rhosts
-rw------- 1 dc4c dc4c 18 8 月   27 10:14 .rhosts
```

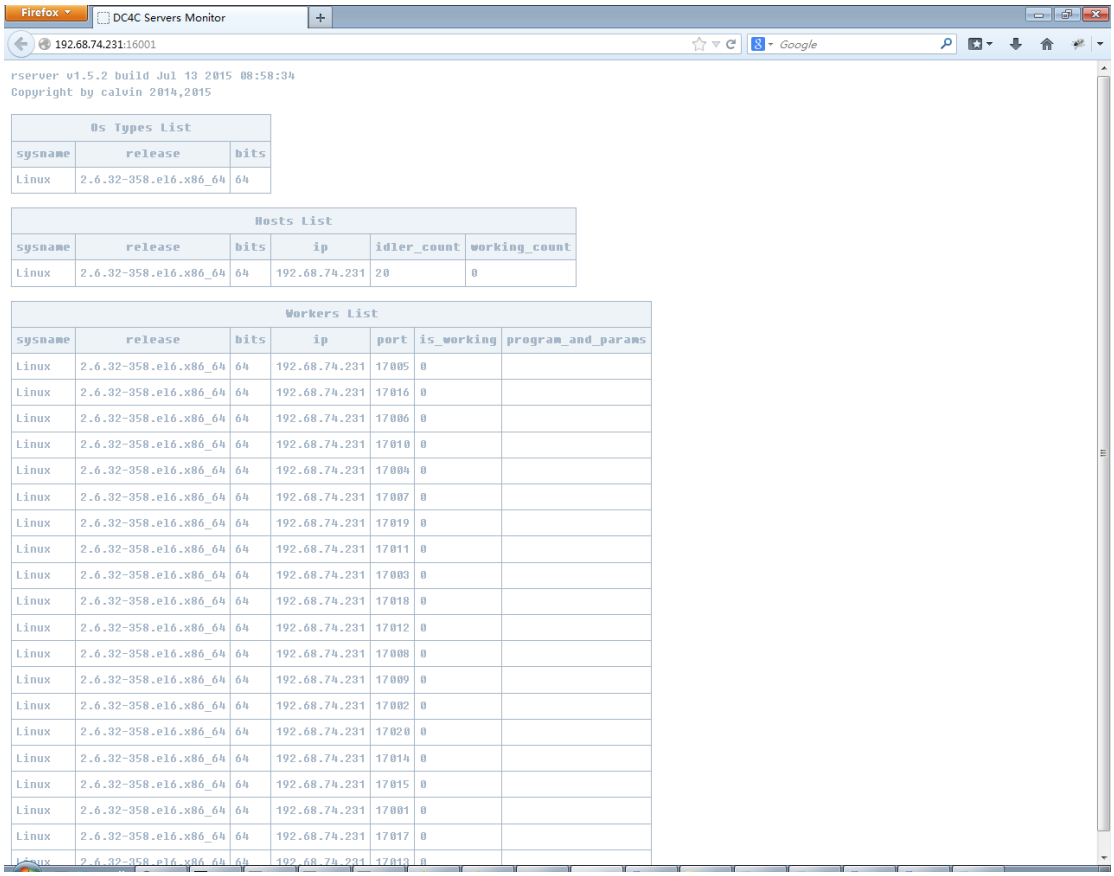
配置 dc4c 集群：

```
$ cat $HOME/dc4c_cluster.conf
168.68.3.121        dc4c
168.68.3.122        dc4c
```

用集群方式启动：

```
$ dc4c_cluster.do start
```

直接用 telnet 连接注册节点或者用浏览器访问注册节点可以看到集群当前状态



（注：上图为开发环境，仅供演示）

用自带测试程序向整个集群发送任务" dc4c_test_worker_hello world"，每个计算节点都将回射字符串"hello world"

```
$ dc4c_test_batch_master "" -2 -2 " dc4c_test_worker_hello world"
```

4.1.2 批量计划配置

```
set names gb2312 ;

-- ----------------------------------------------------------------------
-- 清理所有数据
-- ----------------------------------------------------------------------

truncate table pm_hzbat_schedule ;
truncate table pm_hzbat_batches_info ;
truncate table pm_hzbat_batches_filter ;
truncate table pm_hzbat_batches_tasks ;
truncate table pm_hzbat_batches_direction ;

-- ----------------------------------------------------------------------
```

```
-- 插入计划表数据
-- ----------------------------------------------------------------------------

INSERT INTO pm_hzbat_schedule VALUES (10,'POIT_STAD','数据标准化','','',0);
INSERT INTO pm_hzbat_schedule VALUES (20,'POIT_CALC','积分计算','','',0);
INSERT INTO pm_hzbat_schedule VALUES (25,'POIT_DAYE','日切','','',0);
INSERT INTO pm_hzbat_schedule VALUES (30,'POIT_RPT','报表','','',0);

-- ----------------------------------------------------------------------------
-- 插入批量信息表数据
-- ----------------------------------------------------------------------------

-- 数据标准化计划

INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHA901','客户信息同步
',0,0,1,'','',0,'POTHA901 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB001','存款类到标准数据
',0,0,1,'','',0,'POTHB001 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB002','信贷类到标准数据
',0,0,1,'','',0,'POTHB002 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB003','基金数据到标准数
据',0,0,1,'','',0,'POTHB003 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB004','黄金 T+D 数据到标
准数据',0,0,1,'','',0,'POTHB004 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB005','电子国债到标准数
据',0,0,1,'','',0,'POTHB005 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB006','网银数据到标准数
据',0,0,1,'','',0,'POTHB006 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB007','借记卡数据到标准
数据',0,0,1,'','',0,'POTHB007 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB008','渠道类数据到标准
数据',0,0,1,'','',0,'POTHB008 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB009','微信银行数据到标
准数据',0,0,1,'','',0,'POTHB009 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB010','借记卡 pos 消费数
据到标准数据',0,0,1,'','',0,'POTHB010 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB011','凭证式国债到标准
数据',0,0,1,'','',0,'POTHB011 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB012','电子式国债到标准
数据',0,0,1,'','',0,'POTHB012 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB013','理财签约到标准数
据',0,0,1,'','',0,'POTHB013 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB014','信贷类到标准数据
',0,0,1,'','',0,'POTHB014 -s',60,'',0,0,0,0);
```

```
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB015','理财和基金到标准数据',0,0,1,'','',0,'POTHB015 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB016','信用卡 pos 消费到标准数据',0,0,1,'','',0,'POTHB016 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB017','代理类到标准数据',0,0,1,'','',0,'POTHB017 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB018','代发工资签约数据标准化',0,0,1,'','',0,'POTHB018 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB019','结构性存款',0,0,1,'','',0,'POTHB019 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB020','个人自动转存',0,0,1,'','',0,'POTHB020 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB021','资金归集签约',0,0,1,'','',0,'POTHB021 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB022','微贷卡网银签约',0,0,1,'','',0,'POTHB022 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB023','代扣业务签约',0,0,1,'','',0,'POTHB023 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB024','我行 POS 机具交易量',0,0,1,'','',0,'POTHB024 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB025','基金按购买金额送积分到标准',0,0,1,'','',0,'POTHB025 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB026','在线缴费按笔数送积分',0,0,1,'','',0,'POTHB026 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB027','信用卡还款',0,0,1,'','',0,'POTHB027 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB028','自考报名',0,0,1,'','',0,'POTHB028 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB029','学费缴费按笔数送积分',0,0,1,'','',0,'POTHB029 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB030','理财购买按金额送积分',0,0,1,'','',0,'POTHB030 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_STAD','POTHB031','资金归集成功送积分',0,0,1,'','',0,'POTHB031 -s',60,'',0,0,0,0);

-- 积分计算计划

INSERT INTO pm_hzbat_batches_info VALUES ('POIT_CALC','POTHC001','积分计算',0,0,1,'','',0,'POTHC001 -m',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_CALC','POTHC101','积分汇总',0,0,1,'','',0,'POTHC101 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_CALC','POTHC201','积分汇总',0,0,1,'','',0,'POTHC201 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_CALC','POTHC301','积分入账
```

```
',0,0,1,'','',0,'POTHC301 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_CALC','POTHC401','积分失效
',0,0,1,'','',0,'POTHC401 -s',60,'',0,0,0,0);

-- 积分计算计划

INSERT INTO pm_hzbat_batches_info VALUES ('POIT_DAYE','POTHD001','业务日期切换
',0,0,1,'','',0,'POTHD001 -s',60,'',0,0,0,0);

-- 报表生成计划

INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR001','积分账号月汇总
',0,0,1,'','',0,'POTHR001 -m',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR101','客户积分产生日报表
',0,0,1,'','',0,'POTHR101 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR201','客户积分产生月报表
',0,0,1,'','',0,'POTHR201 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR301','客户积分产生季报表
',0,0,1,'','',0,'POTHR301 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR401','客户积分产生年报表
',0,0,1,'','',0,'POTHR401 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR102','条线积分产生日报表
',0,0,1,'','',0,'POTHR102 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR202','条线积分产生月报表
',0,0,1,'','',0,'POTHR202 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR302','条线积分产生季报表
',0,0,1,'','',0,'POTHR302 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR402','条线积分产生年报表
',0,0,1,'','',0,'POTHR402 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR103','条线客户积分产生日
报表',0,0,1,'','',0,'POTHR103 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR203','条线客户积分产生月
报表',0,0,1,'','',0,'POTHR203 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR303','条线客户积分产生季
报表',0,0,1,'','',0,'POTHR303 -s',60,'',0,0,0,0);
INSERT INTO pm_hzbat_batches_info VALUES ('POIT_RPT','POTHR403','条线客户积分产生年
报表',0,0,1,'','',0,'POTHR403 -s',60,'',0,0,0,0);

-- ----------------------------------------------------------------------------
-- 插入批量过滤表数据
-- ----------------------------------------------------------------------------

INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR201','DD','ME');
INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR202','DD','ME');
```

```
INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR203','DD','ME');
INSERT INTO pm_hzbat_batches_filter VALUES
('POIT_RPT','POTHR301','MM-DD','03-31,06-30,09-30,12-31');
INSERT INTO pm_hzbat_batches_filter VALUES
('POIT_RPT','POTHR302','MM-DD','03-31,06-30,09-30,12-31');
INSERT INTO pm_hzbat_batches_filter VALUES
('POIT_RPT','POTHR303','MM-DD','03-31,06-30,09-30,12-31');
INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR401','MM-DD','12-31');
INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR402','MM-DD','12-31');
INSERT INTO pm_hzbat_batches_filter VALUES ('POIT_RPT','POTHR403','MM-DD','12-31');

-- ----------------------------------------------------------------------------
-- 插入批量依赖表数据
-- ----------------------------------------------------------------------------

-- 数据标准化计划

INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHA901');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB001');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB002');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB003');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB004');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB005');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB006');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB007');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB008');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB009');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB010');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB011');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB012');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB013');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB014');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB015');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB016');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB017');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB018');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB019');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB020');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB021');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB022');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB023');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB024');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB025');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB026');
```

```
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB027');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB028');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB029');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB030');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','','POTHB031');

INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHA901','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB001','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB002','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB003','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB004','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB005','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB006','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB007','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB008','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB009','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB010','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB011','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB012','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB013','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB014','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB015','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB016','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB017','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB018','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB019','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB020','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB021','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB022','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB023','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB024','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB025','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB026','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB027','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB028','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB029','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB030','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_STAD','POTHB031','');

-- 积分计算计划

INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_CALC','','POTHC001');
INSERT INTO pm_hzbat_batches_direction VALUES
('POIT_CALC','POTHC001','POTHC101');
```

```
INSERT INTO pm_hzbat_batches_direction VALUES
('POIT_CALC','POTHC101','POTHC201');
INSERT INTO pm_hzbat_batches_direction VALUES
('POIT_CALC','POTHC201','POTHC301');
INSERT INTO pm_hzbat_batches_direction VALUES
('POIT_CALC','POTHC301','POTHC401');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_CALC','POTHC401','');

-- 积分计算计划

INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_DAYE','','POTHD001');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_DAYE','POTHD001','');

-- 报表生成计划

INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','','POTHR001');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR001','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','','POTHR101');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR101','POTHR201');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR201','POTHR301');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR301','POTHR401');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR401','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','','POTHR102');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR102','POTHR202');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR202','POTHR302');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR302','POTHR402');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR402','');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','','POTHR103');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR103','POTHR203');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR203','POTHR303');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR303','POTHR403');
INSERT INTO pm_hzbat_batches_direction VALUES ('POIT_RPT','POTHR403','');

-- ----------------------------------------------------------------------------
-- 插入任务表数据
-- ----------------------------------------------------------------------------

-- ----------------------------------------------------------------------------
-- 测试用小计划
-- ----------------------------------------------------------------------------

-- INSERT INTO pm_hzbat_schedule VALUES (-5,'TEST','测试','','',0);

-- INSERT INTO pm_hzbat_batches_info VALUES ('TEST','TEST1','测试
```

```
1',0,0,1,'','',0,'dc4c_test_worker_sleep 1',60,'',0,0,0,0);
-- INSERT INTO pm_hzbat_batches_info VALUES ('TEST','TEST2','测试
2',0,0,1,'','',0,'dc4c_test_worker_sleep 2',60,'',0,0,0,0);

-- INSERT INTO pm_hzbat_batches_direction VALUES ('TEST','','TEST1');
-- INSERT INTO pm_hzbat_batches_direction VALUES ('TEST','TEST1','TEST2');
-- INSERT INTO pm_hzbat_batches_direction VALUES ('TEST','TEST2','');
```

4.1.3 批量执行脚本

```
if [ $# -eq 1 ] ; then
	SCHEUDLE_NAME=$1
elif [ $# -eq 2 ] ; then
	SCHEUDLE_NAME=$1
	BATCH_NAME=$2
else
	echo "USAGE : hzbat.sh (SCHDULE_NAME) [ (BATCH_NAME) ]"
	exit 7
fi

DATE=`sql "SELECT DATE_FORMAT(business_date,'%Y-%m-%d') FROM pmp_spma_para
WHERE system_id='hzbank'"|head -4|tail -1|awk '{print $2}'`
HZBAT_STDOUT_DATE_LOG=$HOME/log/hzbat_stdout_${DATE}.log
> $HZBAT_STDOUT_DATE_LOG

date >>$HZBAT_STDOUT_DATE_LOG

TIME=`date +%H`
while [ $TIME -lt 6 ] ; do
	FLAG1=`sql "select count(*) from ftp_rar_para where run_flag<>'3' and
busi_date=( SELECT business_date FROM pmp_spma_para WHERE
system_id='hzbank')"|head -4|tail -1|awk '{print $2}'`
	FLAG2=`sql "select count(*) from ftp_rar_para where run_flag='3' and
busi_date=( SELECT business_date FROM pmp_spma_para WHERE
system_id='hzbank')"|head -4|tail -1|awk '{print $2}'`
	echo "$FLAG1 $FLAG2" >>$HZBAT_STDOUT_DATE_LOG
	if [ $FLAG1 -eq 0 ] && [ $FLAG2 -gt 0 ] ; then
		break;
	fi
	sleep 600
	TIME=`date +%H`
done
```

```
date >>$HZBAT_STDOUT_DATE_LOG

function gogogo
{
    echo "BUSIDATE[$DATE]"
    if [ x"$BATCH_NAME" = x"" ] ; then
        time hzbat "$DATE" $SCHEUDLE_NAME
    else
        time hzbat "$DATE" $SCHEUDLE_NAME $BATCH_NAME
    fi
    echo $?
}

gogogo | tee -a $HZBAT_STDOUT_DATE_LOG

date >>$HZBAT_STDOUT_DATE_LOG
```

该脚本配置进 crontab 将在凌晨 4 点执行，效果为每个十分钟检查数据是否就绪，如果就绪或者到达 6 点，依次执行计划 ID 大于等于 0 的所有计划，hzbat 程序生成日志$HOME/log/hzbat.log，脚本标准输出也 tee 一份到 $HOME/log/hzbat_stdout_(业务日期).log 以备查。

4.2 ****系统日终批量（生产环境）

4.2.1 集群部署

节点类型	数量	网络地址和服务端口
注册节点	2*1 对	168.68.2.161:16001@xxxbat 168.68.2.162:16001@xxxbat
计算节点	10*10 个	168.68.2.171:17001~17010@xxxbat 168.68.2.172:17001~17010@xxxbat 168.68.2.173:17001~17010@xxxbat 168.68.2.174:17001~17010@xxxbat 168.68.2.175:17001~17010@xxxbat 168.68.2.176:17001~17010@xxxbat

		168.68.2.177:17001~17010@xxxbat 168.68.2.178:17001~17010@xxxbat 168.68.2.179:17001~17010@xxxbat 168.68.2.180:17001~17010@xxxbat
用户节点	-	168.68.2.151@xxxbat

...

4.2.2 批量计划配置

4.2.3 批量执行脚本