BI系统用户隔离方案

(仅供内部使用）

For internal use only

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拟制:  Prepared by | 王光明 |  | 日期：  Date | 2012-6-20 |
| 审核:  Reviewed by |  |  | 日期：  Date | yyyy-mm-dd |
| 批准:  Granted by |  |  | 日期：  Date | yyyy-mm-dd |



华为技术有限公司

Huawei Technologies Co., Ltd.

版权所有 侵权必究

All rights reserved

修订记录Revision record·

|  |  |  |  |
| --- | --- | --- | --- |
| 日期  Date | 修订版本Revision version | 修改描述  change Description | 作者  Author |
| 2012-06-20 | V0.8 | 初稿 | 王光明 |
| 2012-06-25 | V0.9 | 评审意见：1.缺失网关机权限  2：缺失整个服务流程的说明  3：hiveJDBC服务的权限问题 | 王永强 |
| 2012-7-4 | V1.0 | 根据评审意见更新文档 | 王光明 |

[1 背景介绍 4](#_Toc329261505)

[2 BI系统架构简介 4](#_Toc329261506)

[2.1 BI系统上下文 4](#_Toc329261507)

[2.2 BI部署架构 6](#_Toc329261508)

[3 方案设计 7](#_Toc329261509)

[3.1 概述 7](#_Toc329261510)

[3.2 数据通道资源隔离方案 8](#_Toc329261511)

[3.2.1 OS本地资源分配方案 8](#_Toc329261512)

[3.2.2 HIVE资源分配方案 9](#_Toc329261513)

[3.2.3 开发/测试访问生产数据方案 11](#_Toc329261514)

[3.3 Hadoop集群资源隔离方案 12](#_Toc329261515)

[3.3.1 MapReduce资源隔离方案 12](#_Toc329261516)

[3.3.2 HDFS资源分配方案 13](#_Toc329261517)

[3.4 MySQL集群资源隔离方案 14](#_Toc329261518)

[3.5 Hadoop/MySQL网关资源隔离方案 14](#_Toc329261519)

[3.6 应用资源隔离方案 14](#_Toc329261520)

[3.6.1 报表服务器资源隔离方案 14](#_Toc329261521)

[3.7 BI审计系统 15](#_Toc329261522)

[3.7.1 审计日志获取 15](#_Toc329261523)

[**1.** 命令行方式 15](#_Toc329261524)

[**2.** 应用程序方式 15](#_Toc329261525)

[3.7.2 审计工具设计 16](#_Toc329261526)

[4 BI系统资源规划及分配 16](#_Toc329261527)

[4.1 资源命名及原则 16](#_Toc329261528)

[4.2 业务组及用户隶属关系设定 17](#_Toc329261529)

[4.3 资源规划和分配 18](#_Toc329261530)

[5 业务应用场景及约束 19](#_Toc329261531)

[5.1 测试人员典型应用流程 19](#_Toc329261532)

[5.2 生产人员典型应用流程 20](#_Toc329261533)

[5.3 业务约束 21](#_Toc329261534)

[6 附录 HIVE元数据重建方案 22](#_Toc329261535)

# 背景介绍

现有的BI系统中，所有业务共享一套BI Hadoop资源。在同一个业务内部，开发/测试/生产系统也没有实现隔离。在实际运行过程中，因为操作人员的各种误操作，导致大量数据处理任务重新计算，造成了BI资源的大量浪费，严重拖延了业务开发进度。提供确保用户之间互相隔离的BI环境是当前急迫的任务。

用户隔离包括两部分的含义：

* 业务隔离：即隶属不同业务的用户之间资源隔离，防止业务之间的干扰和资源争用。对于业务之间需要共享的数据，规划专门的BI资源，并分配给专门的用户组负责维护。
* 业务内部隔离：为了提高业务开发效率，业务内部开发/测试/生产系统之间应隔离，确保开发/测试人员的操作不影响生产系统的正常运行。

为了实现BI系统用户隔离，在现有硬件条件下，需要重新规划BI资源，并设置权限控制系统。具体来说，需要

* 规划并分配BI资源。
* 规划并分配公共BI资源。定义所有业务共享的BI资源，例如业务共享数据（如核心数据模型等）需要的存储和计算资源等。
* 规划并分配业务BI资源。在每个业务内部，需要
  + - 实现开发、测试和生产系统的资源隔离。
    - 提供相应的工具，支持开发/测试系统方便的读取生产系统的数据（但不允许修改数据）。
* 提供权限控制系统。公共资源允许所有业务读取；业务专有资源只允许业务内部人员访问。业务内部研发/测试、生产环境实现隔离。提供审计系统，支持对每个操作员的操作审计。
* 提供应用规范，指导各BI业务人员遵循规范完成业务应用开发/测试和运营。

# BI系统架构简介

## BI系统上下文

BI系统为业务提供数据分析平台，支撑各类数据分析应用，为业务生产经营提供决策依据。BI系统上下文如下图所示：

1. BI 系统上下文

如图示，BI系统的涉众主要包括四类：

* 业务数据来源实体：待分析的原始数据来源，包括图中的终端设备、业务服务器，网站等。终端主要从客户端收集用户行为信息，业务服务器和网站主要提供服务端收集的用户和业务信息。为了支持业务和用户数据收集，需要在终端、业务服务器和网站上部署相应数据采集模块。BI系统通过HTTP/FTP等标准协议从数据来源实体获得数据。
* 应用/业务开发者：应用/业务开发者从BI系统下载数据采集SDK等，嵌入到应用中，从而实现从终端收集用户行为信息等。
* BI系统管理员：完成BI系统构建和维护，完成业务BI资源分配和业务用户管理。是本文的主要读者，本文主要描述如何完成BI资源的分配和业务用户管理。
* 业务用户：包括图中的业务开发人员、业务测试人员和业务生产/运营人员。他们是BI系统的最终用户。业务开发人员利用BI系统完成数据业务开发和验证，业务测试人员利用BI完成数据业务验证，生成/运营人员利用BI系统完成业务生产系统的数据分析，从而获得运营决策支持。

从BI内部，层次架构上可分为数据收集层、ETL层（包括数据导入/导出）、数据存储和处理、业务数据逻辑、数据应用等5层。

* 数据采集层：包括SDK信息采集服务器、Piwik服务器。数据采集层根据BI系统与外部数据源之间的标准协议完成数据采集和缓存。数据采集层的业务逻辑相对而言较为稳定，BI系统只提供生产系统资源，业务开发/测试人员需要自己搭建环境验证数据采集逻辑的正确性。数据采集层的资源由BI系统管理员统一管理和维护，业务用户不感知。本文后续不描述这部分的资源规划。
* ETL层：对应BI系统中的数据通道，完成数据导入导出处理。数据通道上需要根据业务要求部署ETL业务逻辑。BI系统中的数据通道资源由所有业务的研发/测试/生产人员共享，BI系统管理员需完成数据通道资源的规划和分配。
* 数据存储和处理层：对应图中的Hadoop集群和MySQL集群，为BI系统提供计算和存储资源，是BI系统的主体。BI系统中的Hadoop集群和MySQL集群由所有业务的研发/测试/生产人员共享，BI系统管理员需完成存储和计算资源的规划和分配。
* 业务数据逻辑层：对应图中的Hadoop/MySQL 研发/测试/生产网关，是业务数据逻辑的主要载体。为了实现研发/测试和生产系统的隔离，BI系统分别为研发/测试和生产建立独立的Hadoop/MySQL网关。在每个网关内部，BI系统管理员需要完成网关资源在业务用户间的规划和分配。
* 数据应用层：包括图中的报表应用和OLAP应用。根据资源消耗要求的不同，BI系统对不同的应用资源采用不同的共享策略。对报表应用，由于每个业务对计算和存储的资源要求不高，所有业务的研发/测试/生产系统共享一套报表服务器资源。而对于OLAP应用，由于对存储和计算的资源要求很高，BI系统需要分别为测试、生产系统提供独立的资源。BI系统管理员需要完成数据应用资源在业务用户间的规划和分配。

除此以外，为了方便各种业务任务的调度，需要提供TCC (Task Control Center), BI系统分别为测试和生产系统提供独立的TCC调度系统，以方便完成各类数据处理任务的调度。业务用户在TCC系统上的开户由TCC完成，具体的开户方式请参考TCC设计文档，本文不再赘述。

## BI部署架构

BI系统上下文中每个方块为一个逻辑节点，在该逻辑节点上需要部署的系统软件和应用软件如下图所示：

1. BI 系统软件部署示意

在现有硬件条件下，BI系统的物理部署架构如下图所示：

****

1. BI 系统物理部署架构

图中，深灰色的节点尚在规划中，即MySQL集群、OLAP服务器（测试）、文件服务器尚未正式部署。现有的BI系统只具备Hadoop集群，提供基于HIVE的海量数据存储和处理能力。依赖MySQL集群的业务需要待MySQL集群部署后方可移植到BI系统中。

为未来的解决方案中，数据通道节点需要组建集群，以确保业务数据批量导入/导出的性能。另外，需要考虑为TCC设置单独的物理节点，以实现调度系统与业务数据逻辑的物理隔离，提高数据网关的处理能力。

# 方案设计

本章描述BI资源用户隔离总体方案。

## 概述

参考BI系统上下文章节的描述，对业务用户感知的子系统，BI系统管理员需要维护各子系统的系统/平台软件，并制定业务资源规划，为各业务用户组/用户分配资源，并确保各业务之间、业务内部研发/测试和生产系统之间BI资源隔离。此类资源在BI系统中有四类：

* 测试专用逻辑节点，包括TCC（测试）、Hadoop/Mysql网关（测试）、OLAP应用（测试)。对这类节点，BI管理员需要为各业务测试组开户并分配相应的资源。
* 研发/测试/生产共享逻辑节点，包括Hadoop集群、Mysql集群、数据网关（数据导入/导出）、报表应用（报表应用暂不为研发人员提供环境）等。对此类节点，BI管理需要为每个业务的研发/测试/生产组分别开户并分配资源。对属于用一业务的用户组，为了方便研发/测试人员直接利用生产系统的数据完成数据业务开发和测试，需要允许研发/测试组成员读取生产系统的数据（但不能写入）。
* 生产专用逻辑节点， 包括TCC（生产）、Hadoop/Mysql网关（生产）、OLAP应用（生产) 三个节点。对这类节点，BI管理员需要为各业务生产组开户并分配相应的资源。
* 研发专用逻辑节点，目前只有Hadoop/Mysql网关（研发）。对这类节点，BI管理员需要为各业务研发组开户并分配相应的资源。

需要特别指出的是，为了管理和用户使用的方便，同一用户组/用户在所有逻辑节点上的名称应尽可能保持一致。例如，对智汇云生产组，如果其在Hadoop/MySQL网关（生产）节点上的组名为HiCloudProdGrp， 则其在数据通道节点、Hadoop集群、报表应用服务器、OLAP应用服务器（生产）节点上的用户组名称也为HiCloudProdGrp。

下面以数据流向顺序描述各种BI资源的用户隔离策略。按此顺序，分别描述数据通道、Hadoop集群/MySQL集群、数据网关、数据应用资源的隔离方案。在此基础上，最后描述BI审计系统的设计方案。

## 数据通道资源隔离方案

为了确保hadoop网关上HIVE引擎的处理性能，将业务数据传输任务配置到独立的数据传输通道节点上执行。数据通道主要为BI系统提供ETL功能，支持

* 数据导入：将数据从数据采集节点（SDK采集服务器等）导入到Hadoop集群/MySQL集群
* 数据导出：将数据从Hadoop集群/MySQL集群导出到数据应用服务器，如报表服务器等。需要指出的是，未来的第三方数据应用也需要经由数据通道导出或访问数据，数据通道需要支持数据访问行为审计。本版本暂不考虑第三方数据应用访问BI系统及数据访问审计。

参考“BI系统软件部署示意图”，数据通道上主要部署Hadoop客户端、HIVE、MySQL Proxy、FTP等系统软件和业务ETL逻辑。数据通道供所有业务研发/测试和生产用户共享，待规划和分配的资源包括

* OS本地资源：在操作系统上建立用户组和用户，分配用户的本地文件系统Home目录及访问权限。FTP权限跟OS用户对应，业务ETL逻辑部署到用户HOME目录/子目录下。Hadoop中HDFS的访问权限也由其OS用户决定。
* HIVE资源：为所有业务研发/测试/生产人员建立HIVE用户，建立HIVE DB，并实现业务用户之间的隔离。
* MySQL Proxy:用于从MySQL集群中导入导出数据。由于目前暂未提供MySQL集群能力，本版本暂不考虑MySQL Proxy资源规划和分配问题。

下面分别描述OS本地资源和HIVE资源在数据通道上的隔离方案。

### OS本地资源分配方案

操作系统本地资源分配主要指用户开户，包括设立用户组、建立用户、维护用户和用户组之间的关系、为每个用户设立HOME目录并设置访问权限等。

数据通道中，业务OS用户的Home目录一般形式如下：

/home /XXX（组名）/YYY（用户名） drwxr-x---

需要注意的是，每个业务用户Home目录下可能需要存放大量数据，直接将“/home /XXX（组名）/YYY（用户名）”建立到系统盘中很可能存在磁盘空间不足问题。因此，必须将每个用户组对应的目录“/home /XXX（组名）”通过ln命令重定向到磁盘空间容量足够的数据盘上。通过ln重定向的另一个好处是易于支持硬件扩容。

每个用户的Home目录访问权限是“drwxr-x---”,即自己拥有完全权限，同组用户拥有读和执行权限，其他用户无任何权限。对于用户自己创建的文件或目录，缺省拥有同样的权限控制，具体通过umask设置实现。

按照上面的规则，对每个业务，在数据通道内需分别创建开发、测试和生产组，并根据需要在各个组中增加成员。需要指出的是，由于同一业务中开发、测试人员可能需要访问生产系统的数据，以支持开发或测试活动，因此，对于同一业务中开发/测试组的用户，必须同时加入到生产组中。图示如下：

1. 同一业务内部用户和用户组之间的关系

另外，考虑到所有用户都可能有临时数据存放需求，在OS中创建所有用户共享的大容量临时存放目录是必要的。因此在数据通道每个物理节点上创建一个临时目录，供所有业务用户共享。临时目录所有者可以是任意用户成员，也可以是root用户。临时目录位置和权限控制如下：

/home /temp drwxrwxrwx

同样，“/home/tmp”目录的真实物理存放地址需要通过ln命令指向某数据盘。任何业务用户均可以在此目录下创建临时文件或子目录，并修改自己创建的临时目录/文件的权限属性。

### HIVE资源分配方案

参考BI部署架构，为了实现研发/测试和生产的隔离，需要在每个（研发、测试、生产）Hadoop/MySQL网关安装独立的HIVE引擎，同时，在数据通道上也需要安装HIVE客户端，支持访问Hadoop集群上的数据，以实现HIVE数据提取或导入。显然，数据通道上的研发/测试和生产用户只能分别访问研发/测试或生产网关上HIVE引擎中的数据。数据通道上HIVE客户端和Hadoop网关上HIVE引擎的对应关系如下图所示：

1. 数据通道HIVE客户端与Hadoop网关HIVE引擎的关系

如上图所示，为了支持数据通道上的用户访问HIVE资源，需要在数据通道上安装三套HIVE客户端，以分别访问研发/测试或生产网关上的HIVE资源。对数据通道上的业务用户组，研发/测试或生产用户组中的用户分别只能访问其对应的HIVE客户端。

在理想情况下，在每个Hive引擎内部，应对不同业务分别设定不同的DB，以实现业务数据库之间的隔离。业务内部的权限控制基于Hive 的内部机制实现，由业务指定专人负责权限分配。同时，对所有业务共享的Hive资源，应建立公共的HIVE DB，并指定专人负责公共HIVE资源的权限管理。理想规划中的HIVE资源分配示意如下：

1. HIVE资源规划示意

然而，由于前期的智汇云业务未考虑在HIVE中建立业务DB（而是采用了default），如果立即修改为上述方案，将要求修改现有业务中与DB操作相关的全部脚本。因此，本规范只要求新加入平台的业务参考执行，已有业务采用平滑切换的方式逐步切换到新的HIVE DB中。

为了确保HIVE元数据信息的安全，需要对所有HIVE元数据定期备份。考虑到HIVE元数据并不多，为了简化实现，采用全量备份的方式，通过import工具将Mysql全库导出并备份到与Mysql数据不共享的磁盘硬件上。

通过JDBC访问HIVE时，如果HIVE中设置权限控制，将要求JDBC调用程序中输入“用户名”和“密码”（分别对应hive客户端所在主机的OS用户，OS用户密码）。由于当前环境中HIVE未设置权限控制，HIVE Server不会对JDBC的调用者鉴权，导致非授权用户也可以非法访问/修改HIVE中的数据。为此，现环境中需临时通过停止ThriftServer服务等禁止JDBC调用HIVE。

### 开发/测试访问生产数据方案

在HIVE资源分配方案中，为了确保生产系统的安全，对研发、测试和生产各自配置独立的HIVE引擎。基于同样的考虑，在规划HDFS资源时，也将同一业务中研发、测试和生产所能访问的HDFS资源分离。然而，在实际使用中，为了提高开发、测试效率，开发/测试系统往往需要直接读取生产系统中数据。 例如，BI系统中待测试的典型应用格式如下：

FROM from\_statement

INSERT OVERWRITE TABLE tablename1 [PARTITION (partcol1=val1, partcol2=val2 ...)

[IF NOT EXISTS]] select\_statement1

[INSERT OVERWRITE TABLE tablename2 [PARTITION ...

[IF NOT EXISTS]] select\_statement2]

[INSERT INTO TABLE tablename2 [PARTITION ...] select\_statement2] ...;

上述语句中，红色部分需要在生产系统中执行，而蓝色部分需要在研发/测试环境中执行。可是，由于生产、研发、测试系统部署不同的HIVE引擎中，为了执行上述HIVE脚本，需要将生产系统中与红色部分处理相关的HIVE元数据修改后复制到研发/测试环境的HIVE元数据库中。

为方便业务开发/测试人员的调用，应该提供简便的工具，以实现上述红色部分元数据的（修改后）复制。结合BI实际应用场景，我们提供脚本工具getDDLInfo 和 RebuildPartInfo, 研发/测试人员调用这些脚本工具可方便的完成HIVE元数据的获取和重建。

getDDLInfo命令在屏幕上直接显示执行结果，具备以下的命令格式：

getDDLInfo [-p|-t] $table\_name

-p : 获得表的分区信息。当携带该参数时， 以列表方式显示该表的全部分区信息，分析信息中

包括分区键名称、分区键值、存储位置等。每行数据格式如下：

pKey1 = xxx ,… pKeyN = yyy, location = zzz

其中，pKey1, pKeyN需要替换为该表真实的分区键名，XXX, YYY需替换为真实

的分区键值， location为固定名称，zzz为该分区所在的绝对位置。

-t: 获取恢复建表语句的DDL信息，即显示create table语句。

$table\_name: 待获取建表DDL或分区信息的表名称，表名中包括其隶属的数据库信息，即具

备$db.table格式默认default库

RebuildPartInfo具备以下的命令格式：

RebuildPartInfo [-a|-ds|-dr|-de|-l] [-f|-i|-n] $src $table\_name $PartInfo1 $partInfo2， …

-a : 复制全表的信息。在这种情况下 $PartInfo1 $partInfo2参数被忽略

-ds: 复制由$PartInfo1 指定的时间以后（包括$PartInfo1）的所有分区的信息， $PartInfo1的格

式后续说明。

-dr: 复制由[$PartInfo1, $partInfo2 ]指定的时间范围内的所有分区信息

-de: 复制由$PartInfo1 指定的时间以前（包括$PartInfo1）的所有分区的信息。

-l : 复制由$PartInfo1 $partInfo2， …等参数指定的所有分区信息

-f: 当存在表/分区信息重复时，利用新信息强制覆盖旧信息

-i: 当存在表/分区信息重复时，提醒操作员。根据操作员选择覆盖或不覆盖。

-n: 当存在表/分区信息重复时，直接返回失败。

$src: 存放复制信息的原元数据库。 1- 生产（默认）Hive， 2-测试Hive 3-研发

$table\_name: 待复制的表名称，表名中包括其隶属的数据库信息，即具备 $db.table格式，DB

默认default

$PartInfo1： 分区列表信息，具备“xxx;yyy;…”格式。其中“;”是多个分区级别之间的分隔符 ,

上述例，”xxx”是一级分区键值，”yyy”是“xxx”一级分区下的二级分区键值，当

需要复制某时间范围内的分区信息时，最后一级分区必须是时间类型的分区。

$PartInfo2：只在-l、-dr选项时有效，格式 与$PartInfo1一致。当为 –dr选项时，要求最后一级

分区是时间类型的分区

当开发/测试人员需要利用生产系统的数据执行测试代码时，首先需要手工调用getDDLInfo、RebuildPartInfo等工具完成HIVE元数据信息的获取和重建。getDDLInfo、RebuildPartInfo工具的设计和实现参看后面附录中的详述。

显然，基于上述工具，可以实现研发/测试环境跟生产系统的自动同步，只需要定时调用上述脚本，将某时间区间内生成的分区信息自动同步到研发/测试环境即可。当然，在自动同步时，需要处理好分区重复的问题，对于研发/测试中已生成的分区信息，要避免被覆盖。考虑到进度的关系，本版本暂不支持自动同步。

需要特别指出的是，HIVE本身的数据存放在HDFS文件系统中，为了支持研发/测试环境访问生产数据，还需要将业务生产系统拥有的HDFS文件资源向业务研发/测试人员开放“读”权限，即将同一业务的研发/测试组成员加入到生产组中。

## Hadoop集群资源隔离方案

Hadoop集群不直接对业务用户开放，无需为业务用户在Hadoop集群中分配本地OS资源。Hadoop集群中的资源主要是MapReduce资源和HDFS资源，下面分别描述两种资源的隔离方案。

### MapReduce资源隔离方案

MR资源分配方法与Hadoop集群采用的资源调度器相关。现有Hadoop集群主要有三种调度方式：FIFO、公平调度和容量调度，简介如下

* FIFO：
  + 特点：先进先出，单队列调度。优点是简单，缺点是资源利用率低。
* 容量调度（Capacity Scheduler ）
  + 特点：多队列，每个队列分配一定系统容量（Guaranteed Capacity）。空闲资源可以被动态分配给负载重的队列。支持作业优先级
  + 优点：支持多作业并行执行，提高资源利用率。动态调整资源分配，提高作业执行效率
  + 缺点：队列设置和队列选择无法自动进行，用户需要了解大量系统信息
* 公平调度（Fair Scheduler）
  + 特点：将作业分组——形成作业池（based on a configurable attribute , such as user name, unix group,…），给每个作业池分配最小共享资源（Minimum map slots, Minimum reduce slots ）。将多余的资源平均分配给每个作业。
  + 优点：支持作业分类调度，使不同类型的作业获得不同的资源分配，提高服务质量。动态调整并行作业数量，充分利用资源
  + 缺点：不考虑节点的实际负载状态，导致节点负载实际不均衡

结合BI业务实际，我们采用容量调度方式，修改mapred-site.xml文件，指定hadoop的调度器为capacity调度器。在容量调度器的配置文件capacity-scheduler.xml中，预先设定每个操作系统（Hadoop客户端）用户组所使用的调度队列，并配置每个队列可使用的MR资源。后续在操作系统上增加操作员时，将隶属同一业务的操作员分配到该业务对应的用户组。通过这种间接方式，实现MR资源在业务间的分配，图示如下：

1. MR资源分配示意

### HDFS资源分配方案

利用Hadoop命令建立HDFS文件/文件目录时，可以指定每个文件/文件目录的可用空间大小及访问权限。利用此特点，可以预先建立一些文件目录，并设置每个文件目录的可使用的空间大小和访问权限。然后将这些文件目录授权给各具体的业务用户组。通过这种方式，可以实现HDFS资源在业务/业务操作员之间的分配。图示如下：

1. HDFS资源分配示意

需要特别指出的是，基于上述资源规划和配置策略时，所有应用的HIVE建表DDL必须采用外部表（EXTENAL）的形式，且指定的HDFS文件存取位置必须与该用户组拥有的HDFS文件目录相匹配。否则，可能因HDFS文件权限控制机制导致HIVE操作失败。

## MySQL集群资源隔离方案

MySQL集群目前尚未构建，其资源隔离方案后续再补充。

## Hadoop/MySQL网关资源隔离方案

网关是BI系统中主要业务逻辑的载体。参考BI系统上下文，BI系统分别为研发、测试和生产组设立单独的网关。

参考“BI系统软件部署示意图”，网关上主要部署Hadoop客户端、HIVE、MySQL Proxy等系统软件和业务数据处理逻辑。其中Hadoop客户端主要供HIVE调用，不对最终业务用户暴露。对每个Hadoop/MySQL网关，其待规划和分配的资源包括

* OS本地资源：在操作系统上建立用户组和用户，分配用户的本地文件系统Home目录及访问权限。业务数据处理逻辑部署到用户HOME目录/子目录下。
* HIVE资源：对各个网关，为所有业务对应用户组建立HIVE用户，并实现业务用户之间的隔离。
* MySQL Proxy:用于从MySQL集群中导入导出数据，由于目前暂未提供MySQL集群能力，本版本暂不考虑MySQL Proxy资源规划和分配问题。

Hadoop/MySQL网关上OS本地资源的分配和HIVE资源的分配与数据通道上同类资源分配类似，不同的是网关本身已实现研发、测试和生产的隔离。具体的资源隔离方案参考前面数据通道的描述。

## 应用资源隔离方案

参考BI系统上下文，应用资源指基于BI平台开发的各类数据应用依赖的系统资源。数据应用包括报表、OLAP工具、推荐引擎、营销引擎等。目前的数据应用平台只有报表服务器比较成熟，本文只描述报表服务器的资源隔离方案，其他数据应用的资源隔离方案待后续补充。

### 报表服务器资源隔离方案

参考“BI系统软件部署示意图”，在现有架构下，报表服务器基于FineReport平台建设，报表数据存放到MySql数据库中。报表应用服务器资源供测试和生产人员共享（开发人员要求的报表环境由自己搭建）。

报表服务器上待分配的资源包括：

* OS本地资源：在操作系统上建立用户组和用户，分配用户的本地文件系统Home目录及访问权限。OS本地资源的分配策略参考数据通道的相关方法。
* MySQL资源：为每个业务测试/生产组分别建立独立的数据库、数据库连接和数据库用户名。
* FineReport资源: 为每个业务测试/生产组建立单独的用户组和模板上传目录。

报表服务器上具体的资源规划参考“BI系统资源规划和分配” 章节。

## BI审计系统

为了规范每个操作员的行为，减少因误操作而造成的损失，BI系统中需要提供审计系统，支持系统管理员回溯各种意外事件。

为了支持审计，BI系统应记录审计日志，审计日志中至少应包括以下内容：

* 操作员（必选）：动作执行者，这里指登录BI系统的操作系统用户名
* 操作时间（必选）：动作执行时间，时间信息中包括年、月、日、时、分、秒信息。
* 操作类型（必选）：即事件类型，如创建目录/文件等。
* 操作参数（可选）：例如，对于创建目录操作，创建的具体目录信息等。

### 审计日志获取

BI上的用户可通过两种方式访问BI资源：命令行方式（通过CLI界面执行各种操作）和应用程序方式（通过开放接口访问BI资源），下面分别介绍这两种方式下如何生成审计日志以支持审计。

### 命令行方式

对CLI方式，由于操作员在BI系统的各种操作都是OS命令或Hadoop命令，审计日志的生成显然依赖操作系统支持。我们需要打开操作系统的相关配置，使其能自动记录操作员的各种操作，并确保日志记录的安全，防止相关人员删除审计日志信息。

为此，需要在所有直接对业务用户开放的BI系统物理节点上，通过linux syslog记录用户操作日志。具体而言，在每个需要审计的物理节点上，利用root权限

* 针对使用c-shell的用户，在/etc/csh.cshrc中增加：

alias precmd 'logger -p debug "`tty` `whoami` `history 1`"'

* 针对使用bash / ksh的用户, 在/etc/profile中增加：

export PROMPT\_COMMAND='logger -p debug "`tty` `whoami` `history 1`"'

同时，将syslog的服务端配置到相对而言负载较轻的节点上。在现有部署架构上，建议配置到报表节点“10.120.5.41”上。

执行上述配置后，每个物理节点将在本机“/var/log/cmdhistory”目录下生成用户的CLI操作日志，同时会将全部操作员CLI日志同步到syslog服务器所在的节点“/data1/cmdlogs”目录下。

为了防止审计日志丢失，在syslog服务器上，应提供审计日志的备份功能。审计日志采用增量备份方式，每天凌晨2:00启动备份程序完成审计日志的增量备份。

### 应用程序方式

访问BI系统的应用程序包括两类：

* 直接在物理节点上Linux环境下执行的应用程序。对这类应用程序，操作日志依赖应用程序自身记录。现有的TCC属于此类应用，其需要记录审计日志以方便BI系统的审计。
* 通过JDBC或SSH接口访问BI资源。对这类应用，其操作日志可以在应用程序自身和开放接口两个地方记录。对BI系统而言，意味着需要记录JDBC调用日志及SSH操作日志。（正在开发中的HIVE客户端工具通过JDBC访问BI系统，后续版本的TCC将通过SSH访问BI资源）

本文只讨论BI系统本身为支持应用程序访问需要记录的审计日志，应用程序自身的审计日志不属于设计范围。从上面的描述可见，BI系统中需要记录JDBC访问日志和SSH访问日志。

### 审计工具设计

完整的BI审计系统应该支持命令行操作审计和应用程序操作审计。从前面的描述可见，命令行操作审计通过syslog工具实现，而应用程序操作审计依赖应用程序自身和BI系统的JDBC/SSH访问日志。考虑到时间的关系，本版本只考虑CLI操作审计，应用程序的操作审计暂不实现。

直接通过查看操作日志执行审计并不方便，我们应该提供简便的审计工具。在本版本中暂时提供简单的审计工具Audit，方便审计员根据需要完成审计信息的过滤。

Audit命令格式如下：

Audit -o –t $operator $startTime $end\_time

-o : 审计某指定操作员的操作记录， $operator参数设定需要审计的操作员名称。不携带”-o”参

数时，列出所有操作员的记录

-t: 列出由参数[$startTime , $endTime] 指定的时间区间内的所有操作记录。 不携带$endTime

参数时，列出从$startTime以后的所有操作。

Audit采用linux自带的Grep工具实现，具体实现过程略。

# BI系统资源规划及分配

从上章的描述可见，BI资源隔离主要基于Linux用户管理系统实现。通过设立不同的操作系统用户组，让每个用户组拥有独立的OS本地资源、HIVE资源、MR资源、HDFS资源和应用资源（目前只有报表服务器），每个业务的开发/测试/生产部门隶属不同的用户组。本章描述具体的BI资源规划及分配方案。

## 资源命名及原则

为了便于管理，对用户组，采用以下命名规则：

用户组名称： xxxYYYGrp, 其中，

* “xxx”为业务名称缩写。 例如，智汇云业务取名为HiCloud，业务公共部分取名为 Common.
* “YYY”只能取以下三个取值

Dev :  表示开发团队

Test: 表示测试团队

Prod: 表示生产团队

* Grp: 固定名称，表示”组”

以智汇云开发组为例，其用户组名为HiCloudDevGrp.

对OS本地资源（HOME目录），以zzz用户为例，采用以下的规则

zzz用户的Home目录及权限控制：

/home/ xxxYYY/zzz drwxr-x--- 其中，

* xxxYYY 为zzz操作员隶属的组名（去除了Grp），含义与上述的用户组名称一致。
* /home/ xxxYYY为指向某数据盘的连接，该目录必须通过ln 命令连接到某数据盘
* zzz为操作员的名字
* drwxr-x--- 为访问权限。对操作员自己创建的子目录或文件，通过umask缺省设置相同的权限控制属性。

需要指出的是，同一业务中，开发/测试组的所有成员必须同时隶属于生产组，以支持开发/测试人员读取生产系统的数据。

对HIVE资源，考虑现有HIVE脚本均直接利用Default数据库，修改为业务特有数据库将要求修改所有的脚本，本期暂不要求已有业务更改数据库。但对新加入的业务，需要创建独立的HIVE DB。HIVE DB命名规则如下

xxxYYYDB 其中，

* “xxx”为业务名称缩写。 例如，智汇云业务取名为HiCloud，业务公共部分取名为 Common.
* “YYY”只能取以下三个取值

Dev :  表示开发团队

Test: 表示测试团队

Prod: 表示生产团队

* DB: 固定名称，表示”数据库”

MySQL中每个业务组的DB名字与HIVE中完全一致。

对于MapReduce资源，前面说过， MR资源通过调度队列实现用户隔离。调度队列命名规则与用户组命名规则类似，采用下面的方式：

mapred.capacity-scheduler.xxxYYY.queue ，格式说明如下

* 蓝色部分为固定内容
* “xxx”表示业务名称缩写，与用户组中的业务名称缩写保持一致
* “YYY”的含义与用户组的含义相同，也只能取Dev 、Test、Prod三个值

对于HDFS资源，其规则与OS本地资源规则类似。本方案中只规划每个用户组所拥有的HDFS资源根目录名称及访问权限模型，用户组的成员只能在该根目录下创建子目录或文件。HDFS文件命名规则如下：

/AppData/xxxYYY drwxr-x--- 其中,

* 蓝色部分为固定内容
* “xxx”表示业务名称缩写，与用户组中的业务名称缩写保持一致
* “YYY”的含义与用户组的含义相同，也只能取Dev 、Test、Prod三个值
* drwxr-x--- 为访问权限

## 业务组及用户隶属关系设定

根据目前的规划，共有智汇云、应用云、广告、Push、UP等业务将需要共享BI资源。在本规划中，每个业务的缩写及其说明见下表：

|  |  |  |
| --- | --- | --- |
| **业务名称** | **缩写** | **描述** |
| 智汇云 | HiCloud | 对应终端云平台类的业务，如智汇云AppStore等 |
| 应用云 | AppCloud | 对应终端云中各具体应用，如网盘、天天聊等 |
| 广告 | AD | 广告业务 |
| 杂项 | Misc | 对应终端云中各种公共能力，如UP、PUSH等，因为每个公共能力对BI资源的要求都比较小，归类到虚拟的杂项业务中，其它难以归类的各种小应用也归入此虚拟业务类。 |
| 公共业务 | Common | 虚拟业务组，提供所有业务共享模型的处理，如BI系统核心模型及其处理等。 |
| BI支持业务 | BI | 主要用于BI人员协助各业务生产组完成数据应用开发， BI组成员可以无障碍（无需切换用户）读取所有业务生产组的数据。 |

上述业务中，对除BI支持业务外，其余业务均设立研发、测试和生产用户组，BI支持业务只设立生产用户组。业务用户组中的成员与用户组的关系如下图所示：

1. BI系统用户组成员和用户组的关系

上述关系说明如下：

* 同一业务中，所有研发/测试组的成员均同时隶属该业务的生产组，以支持研发/测试人员访问业务生产系统的数据；
* 所有业务生产组的用户成员均同时隶属于Common业务生产组，以支持访问Common业务生产系统中的数据
* BI业务生产组成员均同时隶属于其他所有业务的生产组，以支持BI成员读取其它业务生产系统中的数据。

## 资源规划和分配

在目前的部署架构下，BI系统资源规划如附件所示：



# 业务应用场景及约束

本章描述BI用户资源隔离后业务用户使用BI系统的过程，供业务用户参考。

所有业务用户必须预先开户才能登录并访问BI系统，用户开户时必须指定其隶属的一个或多个预先分配的用户组。对生产用户，一般只会隶属一个用户组；对研发/测试用户，一般还隶属于同一业务的生产组。

用户在BI系统上的开户是指其在BI系统中端到端的开户，包括在数据通道、Hadoop集群/MySQL集群、Hadoop/MySQL网关、数据应用服务器、TCC上的开户。其中TCC的开户由TCC完成，其它系统的开户由BI管理员完成。

用户开户完成后，根据需要在BI系统上开发或部署应用软件，应用软件的部署位置参考部署架构，软件部署成功后，用户再登录TCC，调度应用程序完成数据处理。

## 测试人员典型应用流程

由于E2E的数据逻辑往往非常复杂，数据业务测试一般需要分阶段进行。以数据导入ETL逻辑、数据处理主逻辑、数据导出ETL逻辑、应用逻辑分段测试为例，业务测试人员E2E测试流程如下图所示（蓝虚线指向业务测试人员操作的逻辑实体）：

1. 业务E2E测试典型流程

## 生产人员典型应用流程

业务E2E测试完成后，生产人员将全部数据逻辑部署到BI系统上，执行E2E的业务数据分析处理，并根据数据分析结果优化运营等。生产人员E2E应用BI系统的流程如下图所示（蓝虚线指向业务生产人员操作的逻辑实体）

1. 业务运行典型流程

## 业务约束

参考“BI资源规划和分配”章节描述，所有BI上的业务/用户应遵循以下约束：

* 新增用户时，必须预先选定用户隶属的组，且其本地文件系统中的home目录应与其隶属的组的目录匹配，该用户创建的新文件/新目录权限应与其隶属组的Home目录文件权限一致。

例如，在智汇云开发组中增加操作员XXX时，其本地文件系统home目录为 “/home/HiCloudDev/XXX”，该用户新建的文件/文件目录权限为“drwx r-x ---”。

* 同一业务中，开发、测试、生产系统应使用统一的DDL。建表/建立分区的DDL中通过环境变量或普通变量屏蔽不同用户组的差别，数据存放位置信息应该与该DDL执行者拥有的HDFS文件系统权限一致。

例如，

CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db\_name.]table\_name

[(col\_name data\_type [COMMENT col\_comment], ...)]

[COMMENT table\_comment]

[PARTITIONED BY (col\_name data\_type [COMMENT col\_comment], ...)]

[CLUSTERED BY (col\_name, col\_name, ...) [SORTED BY (col\_name [ASC|DESC], ...)] INTO num\_buckets BUCKETS]

[

[ROW FORMAT row\_format] [STORED AS file\_format]

| STORED BY 'storage.handler.class.name' [WITH SERDEPROPERTIES (...)]

]

[LOCATION hdfs\_path]

[TBLPROPERTIES (property\_name=property\_value, ...)] (Note: only available starting with 0.6.0)

[AS select\_statement]

上述建表语句中，两个红色部分应该用变量或环境变量值代替，以使该脚本无需修改即可被研发/测试/生产系统使用。hdfs\_path位置指向的地址必须与该DDL执行者所拥有的HDFS文件系统目录相匹配。 再如，

ALTER TABLE table\_name ADD PARTITION (partCol = 'value1') location 'loc1'

上述增加分区的DDL中，与存储位置相关的红色部分应该用变量或环境变量值代替，locl位置指向的地址必须与该DDL执行者所拥有的HDFS文件系统目录相匹配。

# 附录 HIVE元数据重建方案

为了支持研发/测试环境访问生产系统数据，BI系统提供getDDLInfo、RebuildPartInfo脚本工具，方便研发/测试人员完成HIVE元数据的获取和重建，从而快速支持开发和测试。本章描述getDDLInfo、RebuildPartInfo脚本的实现方案。

为了实现getDDLInfo、RebuildPartInfo等脚本，首先需要分析HIVE的元数据结构。现网环境中，HIVE元数据采用MySQL存放，其核心数据模型如下图所示：

1. HIVE元数据核心模型

上述模型中HIVE元数据表说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **表名** | **说明** | **主键** | **说明** |
| TBLS | 所有hive表的基本信息 | TBL\_ID | 通过SD\_ID关联存储信息 |
| TABLE\_PARAM | 表级属性，如是否外部表，表的文件数等 | TBL\_ID |  |
| COLUMNS | Hive表字段信息(字段注释，字段名，字段类型，字段序号) | SD\_ID,Column\_name | 通过SD\_ID关联存储信息，通过SD\_ID间接关联到表信息。 |
| SDS | 数据存储信息。所有hive表、表分区所对应的hdfs数据目录和数据格式 | SD\_ID | 通过serde\_id关联其序列化/凡序列化信息。 |
| SERDE\_PARAM | 序列化反序列化信息，如行分隔符、列分隔符、NULL的表示字符等 | SERDE\_ID | 对同一SD\_ID对应的存储记录，SERDE\_ID与SD\_ID值一致。 |
| PARTITIONS | Hive表分区信息 | PART\_ID | 通过tbl\_id关联到表，通过SD\_ID关联到存储信息。 |
| PARTITION\_KEYS | Hive分区表分区键信息 | TBL\_ID,PKey\_Name | 表中的integer\_idx表示分区级别 |
| PARTITION\_KEY\_VALS | Hive表分区名(键值) | PART\_ID, Integer\_idx | integer\_idx表示分区级别，0表示最高级别分区 |
| Partition\_params | 分区属性信息，如分区大小、文件数等 | part\_id, Param\_key | 暂时未用。 |
| Sequence\_table | ID分配器，记录HIVE中下一个可用的tbl\_id, sd\_id, part\_id, serde\_id, db\_id等表示。 | sequence\_name |  |

由上可见，各种ID（如tbl\_id, sd\_id等）等业务无关的字段主要用于实现表信息的关联，由Sequece\_table维护各ID的分配信息。为了在测试/开发环境中重建生产系统中某个表/表分区的信息，有两个方法：

* 直接复制生产系统中与这些表/分区相关的记录，然后将所有“ID”类的字段根据sequece\_table维护的信息替换成研发/测试环境中的ID值。该方法非常简单，缺点是与HIVE的元数据模型高度绑定。
* 利用获得的元数据信息重建DDL，然后调用HIVE引擎在研发/测试环境中执行这些DDL。淘宝采用了该方案。

为了减少风险，降低getDDLInfo、RebuildPartInfo与HIVE元数据表结构的依赖关系，本文采用第二种方法实现，下面描述具体的实现。

为了实现在研发/测试环境中重建生产系统表/分区的结构，首先需要获得建表的DDL信息。根据对HIVE元数据模型的分析，获得建表DDL的过程实现如下：

1. 利用HIVE元数据获得建表DDL过程

与获得建表DDL相关的SQL伪代码如下：

1. 获取建表DDL伪SQL

利用HIVE引擎重建某分区的DDL一般如下：

ALTER TABLE table\_name ADD PARTITION (partCol = 'value1') location 'loc1';

ALTER TABLE table\_name ADD PARTITION (partCol = 'value2') location 'loc2';

为了获得ALTER语句依赖的分区信息，其实现过程设计如下：

1. 利用HIVE元数据获取分区信息过程

与获得ALTER语句相关信息的SQL伪代码如下：

1. 获取分区信息伪SQL代码