文件管理技术方案

> 文件存储在HDFS服务器中。
>
> 应用系统中处理文件流，所有文件操作不落本地盘，生成文件直接写入HDFS。

目录结构设计

目录结构

> 原则上目录结构是文件管理子系统根据自动下载的配置自动建立。
>
> 结构如下：
>
> /{航空公司}/{系统名}/{ENCRYTION|NORMAL}/{INPUT|OUTPUT|DELETED|ARCHIVED}/{模块}/{类别}/{年月}/{年月天}/{时间戳}/具体文件
>
> 具体样例：
>
> /CA/OPRA/NORMAL/INPUT/MAS/OAG/201911/20191118/20191118174530/SSIM-201911.dat

- **压缩文件，会保留原始的压缩文件，并在同级生成一个新的目录，放置解压之后的文件。**

文件操作

1. 通过文件管理子系统配置下载作业，自动建立对应的文件类型的目录。
2. 用户上传文件，通过文件管理子系统的上传api，自动建立相应的文件类型目录（如已存在，则放在对应的已存在的文件类型目录中）
3. 无需开发人员干预。
4. 文件管理子系统提供一个统一的浏览并下载文件的目录，便于开发以及运维人员查看文件，**但不提供新建目录，删除，移动文件等操作**。（待讨论）

用户权限设计

> HDFS中， 分为敏感数据与非敏感数据。对这两种文件，分别创建有读写与只读用户。
>
> 各模块应用系统采用最小权限原则。各业务模块应用不需要写文件则只配置只读用户，如果需要写文件则配置读写权限的用户。
>
> 敏感区的文件只有文件管理子系统配置读写权限的用户，各业务子模块只配置只读用户。

加密策略

ACCA目录中放置行业级的数据，行业级的数据无敏感数据，不加密。

航空公司级的目录全都加密，采用hdfs系统加密，对应用透明。

归档

所有文件按上述中的目录结构保存，不区分冷热文件区。

保留所有原始文件。原始文件压缩则保留压缩文件，原始文件未压缩，直接保留原始文件。

归档的目的：

- 降低文件存储占用的空间
- 降低大量文件，包括海量的小文件，导致的namenode内存不够用的问题。例如，每一个文件名在namenode中占用大约500字节，那么64G的namenode最多可存储1.2亿文件。实际比这个值要少。

归档的操作：

将系统中超过1年（时间需要业务确认）的数据进行归档。

归档前会删除所有临时目录的文件，包括解压之后的文件。保留原始文件。

采用hdfs的archive命令，将目录打包为har文件。例如：/OPRA/CA/CA/INPUT/MAS/OAG/201911目录下的所有文件会变成一个文件/OPRA/CA/CA/INPUT/MAS/OAG/201911.har，有效降低的文件总数。

归档文件的读取：

```
hdfs dfs -cp har:///OPRA/CA/CA/INPUT/MAS/OAG/201911.har /OPRA/CA/CA/INPUT/MAS/OAG/201911
```

需要先将整个har文件解开。

API

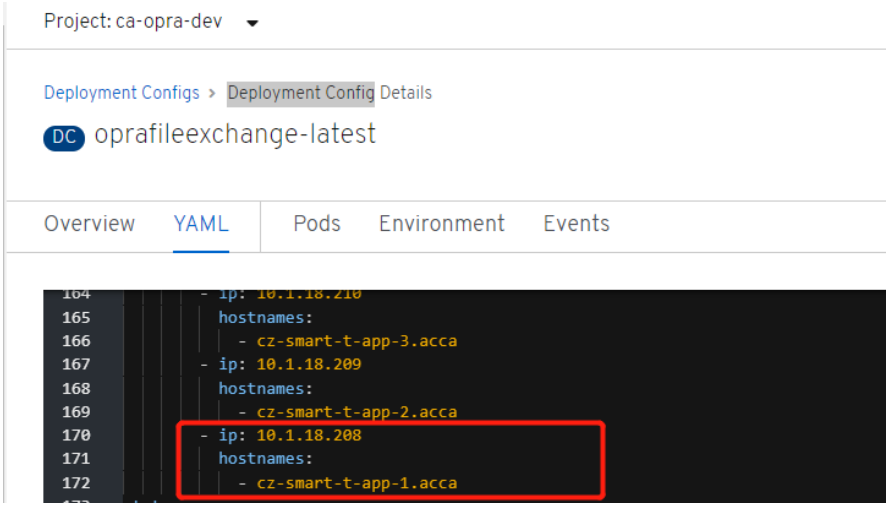
修改hosts文件

本地添加hdfs集群ip与name的对应，之后添加到acca的dns中，可以不用再本地配置

```
10.1.21.11 PAXC-P-HDFS-1
10.1.21.12 PAXC-P-HDFS-2
10.1.21.13 PAXC-P-HDFS-3
10.1.21.14 PAXC-P-HDFS-4
10.1.21.15 PAXC-P-HDFS-5
10.1.21.16 PAXC-P-HDFS-6
10.1.21.17 PAXC-P-HDFS-7
10.1.21.18 PAXC-P-HDFS-8
10.1.21.19 PAXC-P-HDFS-9
10.1.21.20 PAXC-P-HDFS-10
10.1.21.21 PAXC-P-HDFS-11
10.1.21.22 PAXC-P-HDFS-12
10.1.21.23 PAXC-P-HDFS-13
10.1.21.24 PAXC-P-HDFS-14
10.1.21.25 PAXC-P-HDFS-15
10.1.21.26 PAXC-P-HDFS-16
10.1.21.27 PAXC-P-HDFS-17
10.1.21.28 PAXC-P-HDFS-18
```

Openshift添加hdfs集群ip与name的对应

修改对应工程的Deployment Config ， 如图：



HDFS API使用方式

1.application.yml配置如下：

hdfs

```
#UAT
  hdfs:
    mode: cluster
    defaultFS: hdfs://paxcluster
    username: mas
    nameservices: paxcluster
    namenodes: nn1,nn2
    rpc-addresses: PAXC-P-HDFS-4:8030,PAXC-P-HDFS-5:8030


#
hdfs:
 mode: cluster
 defaultFS: hdfs://cz-smart-t-app-1.acca
 username: cz_normal_rw
 nameservices: cz-smart-t-app-1.acca
 namenodes: cz-smart-t-app-1.acca
 rpc-addresses: 10.1.18.208:80200


#hdfsha
  hdfs:
    mode: single
    defaultFS: hdfs://10.1.21.14:8030
    username: mas


#,
  hdfs:
    mode: local
    defaultFS: hdfs://127.0.0.1:0
    username: default
```

2. 启动类需继承ApplicationServer

```
@SpringBootApplication
public class AppServer extends ApplicationServer
```

3. 自动注入

```
@Autowired
private HdfsService hdfsService;
```

4. 获取文件流，与读取本地File文件流的api对比，代码样例：

```
//
                File file = new File(filePath);
        try (BufferedInputStream fis = new BufferedInputStream(new
FileInputStream(file));
                BufferedReader reader = new BufferedReader(new
InputStreamReader(fis, "utf-8"),
                        10 * 1024 * 1024);// 10M
        ) {
......


//hdfsService
                try (BufferedInputStream fis = new BufferedInputStream
(hdfsService.getInpuStream(filePath));
                BufferedReader reader = new BufferedReader(new
InputStreamReader(fis, "utf-8"),
                        10 * 1024 * 1024);// 10M
        ) {
```

hdfsService 中提供了获取输入流，输出流等方法。

上传文件API

> 所有上传操作通过文件子系统完成，并可以触发相应业务系统的导入功能。

具体api由文件管理系统补充。

下载文件API

> 所有下载操作通过文件管理子系统完成。

由文件管理系统补充

图片与缩略图

> 图片也保存到HDFS中。
>
> 文件管理的微服务会提供rest服务，用于实时获取图片的缩略图。类似下面的http url：http://ip:port/filesys/image/thumb/xs?filePath={图片文件的hdfs路径}

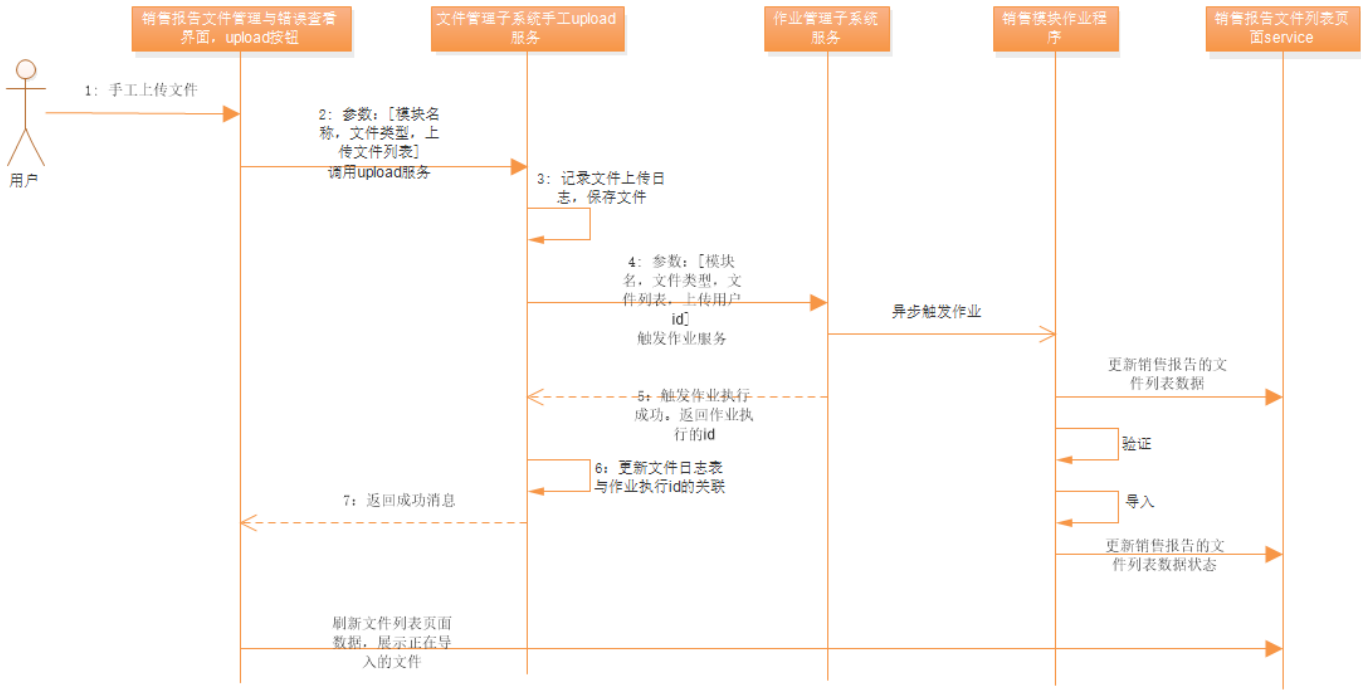
审计

1. hdfs系统本身auditlog日志记录对文件操作的审计

2. 文件管理子系统记录自动抓取并保存的文件操作日志。
3. 文件管理子系统提供api处理用户的文件上传下载操作，并对操作过程记录日志。
4. 文件管理子系统记录归档的操作，并对文件操作记录日志。

与各模块的交互

手工上传文件的统一设计



1、各模块自行评估是否需要有一个文件列表的页面，提供给用户查看当前模块下详细的文件导入列表页面。模块没的文件管理，文件错误查看等功能建议放在一个页面即可。

2、为了统一上传的接口，文件子系统的设计是采用异步触发作业的方式。当用户上传文件后，文件管理子系统保存文件，并异步调用作业管理子系统触发作业执行，然后返回，提示用户成功。前台页面会刷新列表数据，展示作业执行（文件导入）状态。

3、文件管理子系统中提供一个文件日志查询的页面，包含了自动下载的文件，以及手工上传的文件，并记录了对应的导入作业id，可以显示作业执行中的详细信息，但不能保证满足更加明细的业务数据。如果各模块不需要设计自己的文件列表页面，可以使用此页面。

4、上图中省略了文件导入过程中，需要移动文件位置，或者删除文件操作的服务。需要这些服务的请查询文件子系统api。

问题回复：

- 销售已创建ARC/BSP/TCN三个作业，根据文件交互子系统最新设计，这里销售应提供一个作业，还是分别提供ARC/BSP/TCN/手工导入多个作业?请进行确认。

回复：如果能够在导入页面确定用户上传的文件类型，比如有一个专门的BSP文件列表页面，用户上传的文件默认必须是BSP类型，那么upload按钮调用文件子系统服务的时候传模块名（SAL）与文件类型（BSP）即可。如果不能确定文件类型，我的想法是，需要设计一个专门的job，job关联的文件类型为MAN，那么在这个页面固定传模块名（SAL）与文件类型（MAN），MAN的job先解析文件类型，然后再调用响应的解析之后文件类型的job即可。注：job可以调用其他job，直接注入或者通过spring获取另外的job即可；job需要维护在作业管理子系统中，job维护时，需要选择关联一种文件类型。

- 1. 获取作业管理器当前作业ID的接口?

通过JobContext的context.getJobInstanceId();方法获取。

在作业执行的方法中任意位置通过AbstractJob.getJobInstanceId()获取。

在作业执行的方法中任意位置通过JobContext.getCurrentJobContext();获取当前作业的JobContext

- 2 根据作业作业ID，提交作业执行的接口?

作业id这里指的是维护在作业配置页面上的作业。可以通过OprJobApi.runAdhoc方法直接出发作业。具体查看一下api文档或代码。

- 3 开发与测试环境，销售相关作业管理器是否已部署完成，如果还没有，计划什么时候完成?

作业触发调用的服务已部署好。作业维护页面，文件子系统文件列表页面还在开发中

- 4 如何获取手工上传的用户的ID?

通过JobContext类的context.getTaskMessage().getSubmitter();方法获取