# 5. Insight HD 4.0 版本修改列表

## 5.1 预研

1. 编译 ambari/ambari-metrics/ambari-infra&logsearch/ambari-views

【操作方法】

后续补充

2. 屏蔽HDP2.4

屏蔽除redhat6/redhat7之外的所有资源库选项

修改本地源配置，删除通过连接网络获取最新版本的方式，改成直接读本地自定义xml展示版本信息。

【操作方法】

3. 删除ACCUMULO/ATLAS/DRUID/FALCON/KNOX/ZEPPELIN服务，其中Slider服务不能直接配置deleted，需要手动删除目录。

【操作方法】

1. HDP2.6目录下添加删除属性
2. HDP2.5目录下添加删除属性
3. HDP2.3目录下添加删除属性
4. Common-servies下添加版本集成模板
5. HDP2.6/2.5/2.4依次添加自研服务

注意：Slider服务比较特殊。后续补充

4. 设置最小集合

修改文件：（步骤后续补充）



5. 给ambari-server与ambari-agent设置buildNumber

【操作方法】

后续补充

6. 替换汉化文件 ambari-admin.js与ambari-web.js

【操作方法】使用工具查询替换js文件。

## 5.2 第一阶段汉化

### 5.2.1 汉化文件

1. 汉化两个重要js文件：messages.js和i18n.config.js

【操作方法】

1. 汉化messages.js文件：目录：ambari-web/app
   1. 需要exchange\_messages.xlsx，messages.js（英文），ViewReplaceTool.py文件

操作方法：将Excel文件以及python脚本上传到xshell里面的messages.js同一目录里面

【说明】messages.js是一个源码汉化的字符串库，所需汉化文件的内容与Excel表格可能不是一一对应的

* 1. 将三者放置在同一个目录下面，Python环境下执行ViewReplaceTool.py文件
  2. Py程序结束之后的messages.js即为改文件

1. 汉化i18n.config.js----文件改动较少

目录：ambari-admin/src/main/resources/ui/admin-web/app/scripts/i18n.config.js

* 1. 使用比对文件的方式比对i18n.config.js的3.6及4.0版本的差别
  2. 根据日志的结果，在i18n.config.js的3.6汉化版本下进行改进，生成4.0版本

2. ambari-admin其他需要汉化文件：

【操作方法】

1. 找到需要修改的汉化文件的如下地址：
   1. ambari-admin/src/main/resources/ui/admin-web/app/index.html
   2. ambari-admin/src/main/resources/ui/admin-web/app/views/modals/AddRepositoryModal.html
   3. ambari-admin/src/main/resources/ui/admin-web/app/views/modals/AddVersionModal.html
   4. ambari-admin/src/main/resources/ui/admin-web/app/views/remoteClusters/list.html
   5. ambari-web/public-static/index.html
2. 查找admin-ch.xlsx中的内容，同时进一步修改相关内容并检查补充
3. 将admin-ch.xlsx以及python脚本，put到hd4.0源的根目录下
4. admin-ch.xlsx是部分汉化文件

3. messages.js新增内容及其他修改：

【操作方法】

（1）修改message.js文件：

'app.aboutAmbari.version': '版本 4.0'

在其下新增两行：

'alert': '个警告',

'alerts': '个警告',

（2）修改模板文件 ambari-web/app/templates/application.hbs

{{pluralize App.router.mainAlertDefinitionsController.unhealthyAlertInstancesCount singular="alert" plural="alerts"}}

修改为：

{{pluralize App.router.mainAlertDefinitionsController.unhealthyAlertInstancesCount singular="t:alert" plural="t:alerts"}}

（3）修改如下文件第57行

ambari-web/app/templates/main/service/info/summary.hbs

{{view.alertsCount}} {{pluralize view.alertsCount singular="alert" plural="alerts"}}</span>

修改为：

{{view.alertsCount}} {{pluralize view.alertsCount singular="t:alert" plural="t:alerts"}}</span>

（4）修改如下文件第36行

ambari-web/app/templates/main/service/info/summary/master\_components.hbs

{{comp.alertsCount}} {{pluralize comp.alertsCount singular="alert" plural="alerts"}}</span>

修改为：

{{comp.alertsCount}} {{pluralize comp.alertsCount singular="t:alert" plural="t:alerts"}}</span>

4. 页面顶部“0s个操作中”删除“s”：

修改文件：ambari-web/app/utils/host\_progress\_popup.js

this.set("popupHeaderName", numRunning + Em.I18n.t('hostPopup.header.postFix').format(numRunning === 1 ? "" : "s"));

修改为：

this.set("popupHeaderName", numRunning + Em.I18n.t('hostPopup.header.postFix').format(numRunning === 1 ? "" : ""));

5. 其他需要汉化文件：

（1）修改：ambari-web/app/views/main/dashboard/widgets.js

查找修改：

Region In Transition

改为

迁移中的Region

（2）修改文件 ambari-web/app/models/host\_component.js，

其中getTextStatus: function (value)函数汉化：

getTextStatus: function (value) {

switch (value) {

case this.installing:

return '安装中...';

case this.install\_failed:

return '安装失败';

case this.stopped:

return '已停止';

case this.started:

return '已启动';

case this.starting:

return '启动中...';

case this.stopping:

return '停止中...';

case this.unknown:

return '心跳丢失';

case this.upgrade\_failed:

return '升级失败';

case this.disabled:

return '禁用';

case this.init:

return '正在安装...';

}

return '未知';

},

（3）修改文件ambari-web/app/utils/string\_utils.js

解决：“HDFS 和 YARN 服务s在安装过程中会重新启动。”的显示问题

Line204：plural = plural || singular + 's';

修改为：

Line204： plural = plural || singular ;

（4）屏蔽注销版本按钮：修改模板文件

ambari-admin/src/main/resources/ui/admin-web/app/views/stackVersions/

### 5.2.2 版权修改

1. 左下角版权修改：

修改ambari-web/app/assets/index.html文件

<title>Ambari</title>

改为：

<title>Insight HD</title>

<a href="http://www.apache.org/licenses/LICENSE-2.0" target="\_blank">Licensed under the Apache License, Version 2.0</a>.<br>

<a href="/licenses/NOTICE.txt" target="\_blank">See third-party tools/resources that Ambari uses and their respective authors</a>

改为：

<a href="#" target="\_blank">浪潮大数据平台</a>.<br>

<a href="#" target="\_blank">© 2016 Inspur. All Rights Reserved.</a>

2. 版本号修改

修改ambari-web/app/templates/common/about.hbs文件

【操作方法】

1. 修改

<div class="project">Apache Ambari</div>

修改为：

<div class="project"> Incloud Insight HD Manager </div>

1. 修改

<a href="http://ambari.apache.org/" target="\_blank">{{t app.aboutAmbari.getInvolved}}</a>

修改为：

<a>{{t app.aboutAmbari.getInvolved}}</a>

1. 修改

<a href="http://www.apache.org/licenses/LICENSE-2.0" target="\_blank">{{t app.aboutAmbari.licensed}}</a>

修改为：

<a>{{t app.aboutAmbari.licensed}}</a>

1. 删除

{{view.ambariVersion}}

修改ambari-admin/src/main/resources/ui/admin-web/app/views/modals/AboutModal.html

【操作方法】

1. 删除

<span ng-bind="ambariVersion"></span>

1. 修改

<a href="http://ambari.apache.org/" target="\_blank"><span id="i18n-34">{{'common.getInvolved' | translate}}</span></a>

<br>

<a href="http://www.apache.org/licenses/LICENSE-2.0" target="\_blank"><span id="i18n-35">{{'common.license' | translate}}</span></a>

为：

<a ><span id="i18n-34">{{'common.getInvolved' | translate}}</span></a>

<br>

<a><span id="i18n-35">{{'common.license' | translate}}</span></a>

### 5.2.3 Logo图片修改

1. 修改主要左上角产品Logo—云海Insight HD

替换文件：ambari-web/app/assets/img/logo-white.png

替换文件：ambari-web/app/assets/img/ambary-logo.png

2. 修改关于弹框内Inspur图片：

替换文件：ambari-web/app/assets/img/logo.png

3. 修改页面title小图：

【操作方法】

1. 添加文件：logo-title.png至ambari-web/app/assets/img/
2. 修改文件ambari-web/app/assets/index.html：

<link rel="shortcut icon" href="/img/logo.png" type="image/x-icon">

修改为：

<link rel="shortcut icon" href="/img/logo-title.png" type="image/x-icon">

1. 修改文件ambari-admin/src/main/resources/ui/admin-web/app/index.html：

<link rel="shortcut icon" href="/img/logo.png" type="image/x-icon">

修改为：

<link rel="shortcut icon" href="/img/logo-title.png" type="image/x-icon">

4. 修改页面宽度：

1）ambari-admin/src/main/resources/ui/admin-web/app/styles/main.css

修改width从1130px到1169px

line 600：@media (min-width: 1200px) {

line 602：width: 1130px;

为：

line 600：@media (min-width: 1200px) {

line 602：width: 1169px;

2）修改文件：ambari-web/app/styles/application.less

line 5096：.container,

line 5099：.navbar-fixed-bottom .container {

line 5100：width: 1130px;

为：

line 5096：.container,

line 5099：.navbar-fixed-bottom .container {

line 5100：width: 1169px;

### 5.2.4 色值修改

（1）页面上端颜色条修改为蓝色，注意admin视图需要同步修改，以及字体颜色修改

修改文件ambari-web/app/styles/common.less

line 155,156,160,166 色号从#555555, #33333, #33333, #33333统一修改为#338bb8

line 162,164 色号从#c3c3c3修改为#dadada

line 155：@top-nav-bg-color-from: #338bb8;

line 156：@top-nav-bg-color-to: #338bb8;

line 160：@top-nav-menu-active-text-color: #338bb8;

line 162：@top-nav-menu-text-color: #dadada;

line 164：@top-nav-menu-dropdown-border-color: #dadada;

line 166：@top-nav-menu-dropdown-text-color: #338bb8;

line 30,31 色号从#ff0000, #5AB400;修改为#c80404; #1784D2;

line 30：@health-status-red: #c80404;

line 31：@health-status-green: #1784D2;

line 125,131,138从#ff0000修改为##c80404

line 125/131/138：50% { background-color: ##c80404; }

（2）“管理Insigh HD”页上端的色值，版本参数栏的背景色值

修改文件ambari-admin/src/main/resources/ui/admin-web/app/styles/main.css

line459-464 从(#555555, #333333) 修改为#338bb8, #338bb8

line 458：#top-nav .navbar.navbar-static-top .navbar-inner {

line 459：background-image: -moz-linear-gradient(top, #338bb8, #338bb8);

line 460：background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#338bb8), to(#338bb8));

line 461：background-image: -webkit-linear-gradient(top, #338bb8, #338bb8);

line 462：background-image: -o-linear-gradient(top, #338bb8, #338bb8);

line 463：background-image: linear-gradient(to bottom, #338bb8, #338bb8);

line 464：filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#338bb8, endColorstr=#338bb8);

（3）操作成功的页面的色值

修改文件：ambari-web/vendor/styles/bootstrap.css

line 3294从#5bb75b，修改为#338bb8

line 3297从#62c462, #51a351 修改为#338bb8, #338bb8

line 3291：.btn-success {

line 3294：background-color: #338bb8;

line 3297：background-image: -webkit-linear-gradient(top, #338bb8, #338bb8);

（4）用于修复部分色值BUG

将所有色值#468847修改为#0088cc

ambari-web/public/stylesheets/vendor.css 1处

ambari-web/app/styles/enhanced\_service\_dashboard.less 1处

ambari-web/vendor/styles/bootstrap.css 8处

ambari-web/vendor/styles/bootstrap-combobox.css 3处

### 5.2.5 服务描述汉化

1. ambari-server/src/main/resources/common-services/\* 所有metainfo.xml文件描述进行汉化

AMBARI\_INFRA/0.1.0/metainfo.xml

Core shared service used by Ambari managed components 修改为：

---被HD管理组件共享的核心服务

AMBARI\_METRICS/0.1.0/metainfo.xml

***HD Metrics***

A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster 修改为：

---进行集群度量指标的收集、存储，提供集群度量指标的检索服务

FLUME/1.4.0.2.0/metainfo.xml

A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS 修改为：

----一种分布式服务，可以进行大规模流式数据的收集、聚合，并持久化到HDFS。

HAWQ/2.0.0/metainfo.xml

Apache HAWQ - Hadoop Native SQL for High-Performance Advanced Analytics 修改为：

---基于Hadoop的高性能SQL分析引擎。

HBASE/0.96.0.2.0/metainfo.xml

----分布式非关系型数据库，提供统一的配置管理和数据同步。

----Hbase ThriftServer 替换为：HBase ThriftServer

----Hbase RestServer 替换为：HBase RestServer

HDFS/2.1.0.2.0/metainfo.xml

----Hadoop分布式文件系统。

HIVE/0.12.0.2.0/metainfo.xml

----数据仓库系统，支持大数据集或大表的数据存储、即时查询和分析。

HUE/3.11.0/metainfo.xml

----Hadoop UI系统,实现在Web控制台与Hadoop集群进行交互来分析和处理数据。

KAFKA/0.8.1/metainfo.xml

---高吞吐的分布式消息系统。

KERBEROS/1.10.3-10/metainfo.xml

----基于Ticket的网络认证协议，支持在非安全网络中提供安全的身份认证。

MAHOUT/1.0.0.2.3/metainfo.xml

----提供协同过滤、聚类、分类等机器学习算法的分布式算法库。

OOZIE/4.0.0.2.0/metainfo.xml

OOZIE/4.2.0.2.3/metainfo.xml

----负责Hadoop任务的流程协调、调度和执行。

PIG/0.12.0.2.0/metainfo.xml

----分布大数据集的脚本平台。

PXF/3.0.0/metainfo.xml

----支持HPP查询外部数据源,包括Hive、HBase、HDFS和关系型数据库。

RANGER/0.4.0/metainfo.xml

RANGER/0.5.0/metainfo.xml

RANGER/0.7.0/metainfo.xml

----Hadoop的综合安全管理组件。

RANGER\_KMS/0.5.0.2.3/metainfo.xml

----Ranger密钥管理组件。

SPARK/1.2.1/metainfo.xml

SPARK/1.3.1/metainfo.xml

SPARK/1.4.1/metainfo.xml

----基于内存计算的大规模数据快速处理的通用引擎。

SQOOP/1.4.4.2.0/metainfo.xml

----Hadoop和结构化数据存储（如关系数据库）之间的数据交换工具。

STORM/0.9.1/metainfo.xml

----Hadoop流计算框架。

TEZ/0.4.0.2.1/metainfo.xml

----基于Yarn的新一代Hadoop查询处理框架。

YARN/2.1.0.2.0/metainfo.xml（初始处）

----负责资源调度、资源隔离、资源监控的资源管理框架。

YARN/2.1.0.2.0/metainfo.xml（2/3处）

----Hadoop新一代MapReduce计算框架。

ZOOKEEPER/3.4.5/metainfo.xml

----提供高可靠分布式协调的统一服务。

SPARK2/2.0.0/metainfo.xml

----提供Spark2大规模数据处理的快速通用引擎。

LOGSEARCH/0.5.0/metainfo.xml

----提供日志聚合、分析和可视化服务。

2. 查找ambari-server/src/main/resources/stacks/HDP/2.6/services/\*

SPARK2/metainfo.xml

----提供Spark2大规模数据处理的快速通用引擎。

3. 查找ambari-server/src/main/resources/stacks/HDP/2.3/services/\*

HBASE/metainfo.xml

----分布式非关系型数据库，提供统一的配置管理和数据同步。

## 5.3 第二阶段汉化

### 5.3.1 Metric监控

1） 在汉化文件中找到widgets文件夹：该文件夹包含如下汉化需要文件

1. cp.sh：将项目中涉及到的英文文件拷贝到临时文件中

2. mkdir.sh：建立汉化临时目录文件

3. cpback.sh：将汉化好的中文文件覆盖到项目中

4. readme.txt：相关操作手册

5. widgets-ch.xlsx：翻译字典

6. native2ascci-242.sh：将翻译好的文件转换成系统识别的汉化文件（转码编译）

7. WidgetsReplaceTool.py：翻译脚本

2） 请注意.sh文件中的如下变量的地址修改为符合所处环境的相应地址：

1. ambari\_en\_stack\_dir

2. ambari\_en\_common\_dir

3. ambari\_ch\_stack\_dir

4. ambari\_ch\_common\_dir

3） 具体执行步骤请参照：readme.txt文件的内容

1. alert的更改涉及到了json文件和python脚本

查找是请使用find -name \*.json和find -name alert\*.py查找

2. 首先执行mkdir.sh文件

3. 之后执行cp.sh文件

注意要备份拷贝过来的文件夹，以保证以后能还原

具体方法为复制一份文件夹到backup下面

4. 执行翻译py脚本 AlertsReplaceTool.py

5. 执行native.sh将json文件翻译为可识别的ascii编码形式，在py脚本第二行中添加一行utf8命令

6. 转码完成之后执行cpback.sh文件

### 5.3.2 警告监控

1） 在汉化文件中找到alerts文件夹：该文件夹包含如下汉化需要文件

1. cp.sh：将项目中涉及到的英文文件拷贝到临时文件中

2. mkdir.sh：建立汉化临时目录文件

3. cpback.sh：将汉化好的中文文件覆盖到项目中

4. readme.txt：相关操作手册

5. alerts-ch.xlsx：翻译字典

6. native.sh：将翻译好的文件转换成系统识别的汉化文件（转码编译）

7. AlertsReplaceTool.py：翻译脚本

2） 请注意.sh文件中的如下变量的地址修改为符合所处环境的相应地址：

1. ambari\_en\_host\_dir

2. ambari\_en\_common\_dir

3. ambari\_ch\_host\_dir

4. ambari\_ch\_common\_dir

3） 具体执行步骤请参照：readme.txt文件的内容

### 5.3.3 视图

1） 在汉化文件中找到views文件夹：该文件夹包含如下汉化需要文件

1. cp.sh：将项目中涉及到的英文文件拷贝到临时文件中

2. mkdir.sh：建立汉化临时目录文件

3. cpback.sh：将汉化好的中文文件覆盖到项目中

4. readme.txt：相关操作手册

5. views-ch.xlsx：翻译字典

6. ViewsReplaceTool.py：翻译脚本

2）具体执行步骤请参照：readme.txt文件的内容

### 5.3.4 监控源码

1. 修改ambari-server/src/main/java/org/apache/ambari/server/alerts下文件

1）AgentHeartbeatAlertRunnable.java

line 43：private static final String INIT\_MSG = "{0} is initializing";

line 48：private static final String HEALTHY\_MSG = "{0} is healthy";

line 53：private static final String STATUS\_UPDATE\_MSG = "{0} is waiting for status updates";

line 58：private static final String HEARTBEAT\_LOST\_MSG = "{0} is not sending heartbeats";

line 63：private static final String UNHEALTHY\_MSG = "{0} is not healthy";

line 68：private static final String UNKNOWN\_MSG = "{0} has an unknown state of {1}";

修改为

line 43：private static final String INIT\_MSG = "{0}正在初始化";

line 48：private static final String HEALTHY\_MSG = "{0}处于健康状态";

line 53：private static final String STATUS\_UPDATE\_MSG = "{0}在等待状态更新";

line 58：private static final String HEARTBEAT\_LOST\_MSG = "{0}没有发送心跳";

line 63：private static final String UNHEALTHY\_MSG = "{0}处于非健康状态";

line 68：private static final String UNKNOWN\_MSG = "{0}处于一个未知状态：{1}";

2）AmbariPerformanceRunnable.java

line 81：private static final String PERFORMANCE\_OVERVIEW\_TEMPLATE = "Performance Overview:"

line 93：private static final String PERFORMANCE\_AREA\_FAILURE\_TEMPLATE = " Unable to execute performance alert area {0}: ({1})";

line 111：REQUEST\_BY\_STATUS("Database Access (Request By Status)",

line 126：HRC\_SUMMARY\_STATUS("Database Access (Task Status Aggregation)",

line 146：REST\_API\_GET\_CLUSTER("REST API (Cluster)",

修改为

line 81：private static final String PERFORMANCE\_OVERVIEW\_TEMPLATE = "性能总览："

line 93：private static final String PERFORMANCE\_AREA\_FAILURE\_TEMPLATE = " 无法运行性能告警区域 {0}： ({1})";

line 111：REQUEST\_BY\_STATUS("访问数据库（状态请求）",

line 126：HRC\_SUMMARY\_STATUS("访问数据库（任务状态聚合）",

line 146：REST\_API\_GET\_CLUSTER("REST API （集群）",

3）ComponentVersionAlertRunnable.java

line 61：private static final String ALL\_COMPONENTS\_CORRECT\_MSG = "All components are reporting their expected versions.";

line 66：private static final String UPGRADE\_IN\_PROGRESS\_MSG = "This alert will be suspended while the upgrade to {0} is in progress.";

line 71：private static final String UNKNOWN\_COMPONENT\_MSG\_TEMPLATE = "Unable to retrieve component information for {0}/{1}";

line 76：private static final String MISMATCHED\_VERSIONS\_MSG = "The following components are reporting unexpected versions: ";

line 82：private static final String CLUSTER\_PROVISIONING\_MSG = "The cluster is currently being provisioned. This alert will be skipped.";

line 87：private static final String CLUSTER\_OUT\_OF\_SYNC\_MSG = "The cluster's CURRENT version could not be determined.";

修改为：

line 61：private static final String ALL\_COMPONENTS\_CORRECT\_MSG = "所有组件均为所需版本。";

line 66：private static final String UPGRADE\_IN\_PROGRESS\_MSG = "当在升级{0}时，这项告警条目将被暂时停止工作。";

line 71：private static final String UNKNOWN\_COMPONENT\_MSG\_TEMPLATE = "暂时无法获取如下组件的信息：{0}/{1} ";

line 76：private static final String MISMATCHED\_VERSIONS\_MSG = "如下组件的版本和所需版本不一致： ";

line 82：private static final String CLUSTER\_PROVISIONING\_MSG = "集群正在准备中，此警告将被跳过。";

line 87：private static final String CLUSTER\_OUT\_OF\_SYNC\_MSG = "无法确认集群当前版本。";

4）StaleAlertRunnable.java

line 67：private static final String ALL\_ALERTS\_CURRENT\_MSG = "All alerts have run within their time intervals.";

line 72：private static final String STALE\_ALERTS\_MSG = "There are {0} stale alerts from {1} host(s):\n{2}";

修改为

line 67：private static final String ALL\_ALERTS\_CURRENT\_MSG = "所有监控都在其时间间隔内运行正常";

line 72：private static final String STALE\_ALERTS\_MSG = "在主机{1}中存在有{0} 个过期警告：\n{2}";

2. 修改ambari-server/src/main/java/org/apache/ambari/server/events/listeners/alerts/下文件

1）AgentHeartbeatAlertRunnable.java

line 198：aggregateAlert.setText("There are no instances of the aggregated alert.");

line 200：aggregateAlert.setText("There are alerts with a state of UNKNOWN.");

修改为

line 198：aggregateAlert.setText("没有聚合警告的实例。");

line 200：aggregateAlert.setText("存在UNKNOWN状态的实例。");

### 5.3.5 汉化文件

1）汉化ambari-web/app/utils/ajax/ajax.js文件

line 576："context": data.displayName + " Service Check",

line 620：'context': 'Cancel operation',

line 1848："context": "Remove IOP select",

line 2101："context": "Check hosts",

line 2386："context": 'Restart all services',

line 2408："context": "Restart all required services",

line 2883："context": "Check host",

修改为：

line 576："context": data.displayName + "服务检测",

line 620：'context': '操作取消',

line 1848："context": "移除IOP选择",

line 2101："context": "检测主机",

line 2386："context": '重启所有服务',

line 2408："context": "重启所需服务",

line 2883："context": "检测主机",

### 5.3.6 汉化警告状态

1）汉化 ambari-web/app/models/alerts/alert\_definition.js

line 259：'CRITICAL': 'CRIT',

line 260：'WARNING': 'WARN',

line 261：'OK': 'OK',

line 262：'UNKNOWN': 'UNKWN',

line 263：'PENDING': 'NONE'

修改为：

line 259：'CRITICAL': '严重',

line 260：'WARNING': '警告',

line 261：'OK': '正常',

line 262：'UNKNOWN': '未知',

line 263：'PENDING': '无'

2）汉化 ambari-web/app/models/alerts/alert\_instance.js

位置为line166-170，汉化内容参考1）

3）汉化 ambari-web/app/views/main/alert\_definitions\_view.js

line 194：label: 'OK'

line 199：label: 'WARNING'

line 203：label: 'CRITICAL'

line 207：label: 'UNKNOWN'

line 211：label: 'NONE'

line 293：label: 'Past 1 hour'

line 297：label: 'Past 1 Day'

line 301：label: 'Past 2 Days'

line 305：label: 'Past 7 Days'

line 309：label: 'Past 14 Days'

line 313：label: 'Past 30 Days'

修改为：

line 194：label: '正常'

line 199：label: '警告'

line 203：label: '严重'

line 207：label: '未知'

line 211：label: '无'

line 293：label: '最近1小时'

line 297：label: '最近1天'

line 301：label: '最近2天'

line 305：label: '最近7天'

line 309：label: '最近14天'

line 313：label: '最近30天'

4）汉化 ambari-web/app/views/main/host/host\_alerts\_view.js

位置为line155-167，参考汉化内容见3）

5）汉化 ambari-web/app/views/main/alerts/definition\_details\_view.js

位置为line186-203，参考汉化内容见3）

6）汉化时间：ambari-web/app/utils/date/date.js

Line 60：format = format || 'ddd, MMM DD, YYYY HH:mm';

Line 76：var format = 'ddd MMM DD YYYY';

修改为

Line 60：format = format || 'YYYY-MM-DD HH:mm:ss';

Line 76：var format = 'YYYY-MM-DD';

## 5.4 集成服务

注意：HDP2.3、2.5、2.6 stacks与common-services集成关系，同步修改

### 5.4.1 Hawq/PXF

（步骤后续补充）

### ；5.4.2 Solr

（步骤后续补充）

### 5.4.3 Hue

（步骤后续补充）

### 5.4.4 DataSpace

（步骤后续补充）

### 5.4.5 Redis

该组件需要依赖于Zookeeper，需要在metainfo中添加如下内容：

如下内容添加在标签service下面，与configuration-dir为兄弟节点

<requiredServices>

<service>ZOOKEEPER</service>

</requiredServices>

### 5.4.6 MPP

该组件已经在Insight HD4.0.1中被屏蔽

### 5.4.7 ML

该组件需要依赖于Spark1和Spark2，需要在metainfo中添加如下内容：

如下内容添加在标签service下面，与configuration-dir为兄弟节点

<requiredServices>

<service>ZOOKEEPER</service>

<service>SPARK</service>

<service>SPARK2</service>

</requiredServices>

## 5.5 集成组件

### 5.5.1 COMMON-SERVICES/HDFS

#### 5.5.1.1 metainfo.xml

在line150添加内容：

<components>

<component>

<name>HDFS\_HTTPFS</name>

<displayName>HDFS HttpFS Server</displayName>

<category>SLAVE</category>

<cardinality>0-1</cardinality>

<versionAdvertised>true</versionAdvertised>

<commandScript>

<script>scripts/hdfs\_httpfs.py</script>

<scriptType>PYTHON</scriptType>

<timeout>1200</timeout>

</commandScript>

<logs>

<log>

<logId>hdfs\_httpfs</logId>

<primary>true</primary>

</log>

</logs>

<dependencies>

<dependency>

<name>HDFS/HDFS\_CLIENT</name>

<scope>host</scope>

<auto-deploy>

<enabled>true</enabled>

</auto-deploy>

</dependency>

</dependencies>

</component>

#### 5.5.1.2 configuration

（1）hdfs-log4j.xml

【备注】关于日志大小的配置，暂不修改。

157,158c115,116

< hadoop.security.log.maxfilesize={{hadoop\_security\_log\_max\_backup\_size}}MB

< hadoop.security.log.maxbackupindex={{hadoop\_security\_log\_number\_of\_backup\_files}}

---

**> hadoop.security.log.maxfilesize=256MB**

**> hadoop.security.log.maxbackupindex=20**

177a136,137

**> hdfs.audit.log.maxfilesize=1024MB**

**> hdfs.audit.log.maxbackupindex=200**

206,207c166,167

< log4j.appender.RFA.MaxFileSize={{hadoop\_log\_max\_backup\_size}}MB

< log4j.appender.RFA.MaxBackupIndex={{hadoop\_log\_number\_of\_backup\_files}}

---

**> log4j.appender.RFA.MaxFileSize=256MB**

**> log4j.appender.RFA.MaxBackupIndex=10**

（2）hdfs-logsearch-conf.xml

新版本新增配置文件，无需修改。

（3）hdfs-site.xml

新增本有改进，可直接使用，除此之外，新增参数有（暂不修改）：

**510a505,534**

**> <property>**

**> <name>dfs.namenode.http-bind-host</name>**

**> <value>0.0.0.0</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

**> <property>**

**> <name>dfs.namenode.rpc-bind-host</name>**

**> <value>0.0.0.0</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

**> <property>**

**> <name>dfs.namenode.servicerpc-bind-host</name>**

**> <value>0.0.0.0</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

**> <property>**

**> <name>dfs.client.use.datanode.hostname</name>**

**> <value>true</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

**> <property>**

**> <name>dfs.datanode.use.datanode.hostname</name>**

**> <value>true</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

**> <property>**

**> <name>dfs.allow.truncate</name>**

**> <value>true</value>**

**> <on-ambari-upgrade add="true"/>**

**> </property>**

（4）httpfs-env.xml：

直接拷贝可用。

（5）httpfs-log4j.xml：

直接拷贝可用。

（6）httpfs-site.xml：

直接拷贝可用。新拷贝的xml文件赋权限a+x

#### 5.5.1.3 package/scripts

（1）hdfs\_httpfs.py：

Line105添加如下内容：

def get\_component\_name(self):

return "hdfs-httpfs"

拷贝，同时赋权限a+x

start和stop方法中确保参数为：

(self, env, upgrade\_type=None)

（2）params\_linux.py：

在line113行添加内容：

#httpfs default parameters

httpfs\_user = config['configurations']['httpfs-env']['httpfs\_user']

httpfs\_group = config['configurations']['httpfs-env']['httpfs\_group']

httpfs\_port = config['configurations']['httpfs-env']['httpfs\_port']

httpfs\_admin\_port = config['configurations']['httpfs-env']['httpfs\_admin\_port']

httpfs\_conf\_dir = config['configurations']['httpfs-env']['httpfs\_conf\_dir']

httpfs\_log\_dir = config['configurations']['httpfs-env']['httpfs\_log\_dir']

httpfs\_env\_content = config['configurations']['httpfs-env']['content']

httpfs\_log4j\_content = config['configurations']['httpfs-log4j']['content']

### 5.5.2 COMMON-SERVICES/HBase

#### 5.5.2.1 metainfo.xml

在line134添加内容如下：

<component>

<name>HBASE\_THRIFTSERVER</name>

<displayName>Hbase ThriftServer</displayName>

<category>SLAVE</category>

<cardinality>0+</cardinality>

<versionAdvertised>true</versionAdvertised>

<dependencies>

<dependency>

<name>HBASE/HBASE\_CLIENT</name>

<scope>host</scope>

<auto-deploy>

<enabled>true</enabled>

</auto-deploy>

</dependency>

</dependencies>

<commandScript>

<script>scripts/hbase\_thriftserver.py</script>

<scriptType>PYTHON</scriptType>

</commandScript>

</component>

<component>

<name>HBASE\_RESTSERVER</name>

<displayName>Hbase RestServer</displayName>

<category>SLAVE</category>

<cardinality>0+</cardinality>

<versionAdvertised>true</versionAdvertised>

<dependencies>

<dependency>

<name>HBASE/HBASE\_CLIENT</name>

<scope>host</scope>

<auto-deploy>

<enabled>true</enabled>

</auto-deploy>

</dependency>

</dependencies>

<commandScript>

<script>scripts/hbase\_restserver.py</script>

<scriptType>PYTHON</scriptType>

</commandScript>

</component>

#### 5.5.2.2 package/scripts

（1）hbase\_restserver.py

Line32新增如下内容：

from resource\_management.libraries.functions.check\_process\_status import check\_process\_status

Line64新增如下内容：

def get\_component\_name(self):

return "hbase-thriftserver"

拷贝，同时赋权限a+x。

start和stop方法中确保参数为：

(self, env, upgrade\_type=None)

（2）hbase\_thriftserver.py

Line32新增如下内容：

from resource\_management.libraries.functions.check\_process\_status import check\_process\_status

Line64新增如下内容：

def get\_component\_name(self):

return "hbase-thriftserver"

拷贝，同时赋权限a+x。

start和stop方法中确保参数为：

(self, env, upgrade\_type=None)

### 5.5.3 COMMON-SERVICES/Zookeeper

#### 5.5.3.1 3.4.5/metainfo.xml

在line71 添加内容如下：

<component>

<name>ZOOKEEPER\_REST</name>

<displayName>Zookeeper Rest Server</displayName>

<category>SLAVE</category>

<cardinality>0+</cardinality>

<versionAdvertised>true</versionAdvertised>

<commandScript>

<script>scripts/zookeeper\_rest.py</script>

<scriptType>PYTHON</scriptType>

</commandScript>

<dependencies>

<dependency>

<name>ZOOKEEPER/ZOOKEEPER\_CLIENT</name>

<scope>host</scope>

<auto-deploy>

<enabled>true</enabled>

</auto-deploy>

</dependency>

</dependencies>

</component>

在line101添加内容如下：

<package>

<name>wget</name>

</package>

#### 5.5.3.2 3.4.5/package/scripts

（1）zookeeper\_rest.py

Line34添加如下内容：

from resource\_management.libraries.functions.check\_process\_status import check\_process\_status

Line82添加如下内容：

def get\_component\_name(self):

return "zookeeper-rest"

拷贝，同时赋权限a+x。

start和stop方法中确保参数为：

(self, env, upgrade\_type=None)

（2）repoin.py

直接拷贝可用，同时赋权限a+x。

（3）status\_params.py

Line45添加内容如下：

zk\_rest\_pid\_file = format("{zk\_pid\_dir}/zookeeper\_rest.pid")

（4）params\_linux.py

Line32添加内容如下：

import repoin

Line41添加内容如下：

baseurl = repoin.baseurl

download\_url = format('{baseurl}zookeeper/zookeeper\_src.tgz')

Line75添加内容如下：

zk\_rest\_pid\_file = status\_params.zk\_rest\_pid\_file

#### 5.5.3.3 3.4.5/package/templates

添加文件rest\_script.j2，直接拷贝可用，同时赋权限a+x。

### 5.5.4 STACKS/HDFS

全部集成至common-services，参考5.5.1

### 5.5.5 STACKS/HBase

全部集成至common-services，参考5.5.2

## 5.6 集成补丁

### 5.6.0 准备工作

（1）hook.py

Line24 添加内容如下：

from patch import \*

patch\_provider()

（2）patch.py

拷贝可用。

（3）patch-site.xml

拷贝可用。

（4）repoin.py

拷贝可用。

### 5.6.1 Sqoop

<service>

<service\_name>SQOOP</service\_name>

<role>SQOOP</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>replace</patch\_type>

<source\_dir>/hd-patch/sqoop/sqoop-1.4.6.2.6.1.0-129.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/sqoop/sqoop-1.4.6.2.6.1.0-129.jar</target\_dir>

</service>

<service>

<service\_name>SQOOP</service\_name>

<role>SQOOP</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<source\_dir>/hd-patch/sqoop/lib/kafka-clients-0.10.1.2.6.1.0-129.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/sqoop/lib/</target\_dir>

</service>

<service>

<service\_name>SQOOP</service\_name>

<role>SQOOP</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<source\_dir>/hd-patch/sqoop/lib/metrics-core-2.2.0.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/sqoop/lib/</target\_dir>

</service>

<service>

<service\_name>SQOOP</service\_name>

<role>SQOOP</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<source\_dir>/hd-patch/sqoop/lib/scala-library-2.10.6.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/sqoop/lib/</target\_dir>

</service>

<service>

<service\_name>SQOOP</service\_name>

<role>SQOOP</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<source\_dir>/hd-patch/sqoop/lib/scala-library-2.11.8.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/sqoop/lib/</target\_dir>

</service>

### 5.6.2 Hive

<service>

<service\_name>HIVE</service\_name>

<role>HIVE\_CLIENT</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<sour

ce\_dir>/hd-patch/hive/lib/com.inspur.hive.serde2.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/hive/lib/</target\_dir>

</service>

<service>

<service\_name>HIVE</service\_name>

<role>HIVE\_SERVER</role>

<stack\_current\_version>2.6.1.0-129</stack\_current\_version>

<patch\_type>add</patch\_type>

<source\_dir>/hd-patch/hive/lib/com.inspur.hive.serde2.jar</source\_dir>

<target\_dir>/usr/hdp/2.6.1.0-129/hive/lib/</target\_dir>

</service>

### 5.6.3 PXF

### 5.6.4 Hawq

### 5.6.4 Ranger

### 5.6.6 HBase Bulkload

## 5.7 logsearch模块汉化

1. 获取logsearch的汉化压缩文件：“logsearch汉化.zip”；

2. 解压该文件并；

3. 汉化logsearch的webapp：

3.1 进入替换文件夹；

3.2 找到webapp文件夹，根据“替换说明.txt”，替换掉logsearch下面的webapp文件：

webapp的具体目录为：ambari-logsearch/ambari-logsearch-portal/src/main/webapp；

4. 相关logsearch配置修改：

4.1 根据“logsearch配置”这个文件夹的txt文件，替换掉相应目录下的.j2文件；

4.2 根据“修改metainfo及增加conf”，修改如下组件的metainfo和logsearch-conf：

Dataspace，hpp，hue，ml，pxf，solr

## 5.8 组件屏蔽

### 5.8.1 屏蔽组件SLIDER

（1）删除如下SLIDER目录结构：

1. ambari-server/src/main/resources/stack/HDP/2.6/services/SLIDER

2. ambari-server/src/main/resources/stack/HDP/2.5/services/SLIDER

3. ambari-server/src/main/resources/stack/HDP/2.4/services/SLIDER

4. ambari-server/src/main/resources/stack/HDP/2.3GlusterFS/services/SLIDER

5. ambari-server/src/main/resources/stack/HDP/2.3ECS/services/SLIDER

6. ambari-server/src/main/resources/stack/HDP/2.3/services/SLIDER

7. ambari-server/src/main/resources/stack/HDP/2.2/services/SLIDER

8. ambari-server/src/main/resources/common-services/SLIDER

（2）去除掉其他组件对SLIDER的依赖：

1. 删除YARN的metainfo中如下代码：

ambari-server/src/main/resources/common-services/YARN/2.1.0.2.0/metainfo.xml

Line 220 – Line 225：

<dependency>

<name>SLIDER/SLIDER</name>

<scope>host</scope>

<auto-deploy>

<enabled>true</enabled>

</auto-deploy>

</dependency>

### 5.8.2 停用组件INSIGHT MPP

（1）ambari-server/src/main/resources/stacks/HDP/2.6/services/GREENPLUM/目录下：

将metainfo.xml重命名为metainfo.xml.bak

（2）ambari-server/src/main/resources/stacks/HDP/2.5/services/GREENPLUM/目录下：

将metainfo.xml重命名为metainfo.xml.bak

### 5.8.3 屏蔽组件删除和安装功能

（1） 屏蔽主界面操作中“添加服务”

ambari-web/app/templates/main/service/all\_services\_actions.hbs目录下：

删除掉line28-33，内容如下：

<li {{bindAttr class="view.serviceController.isAllServicesInstalled:disabled :add-service-button"}}>

<a href="#"

{{bindAttr class="view.serviceController.isAllServicesInstalled:disabled"}}

{{action gotoAddService target="view.serviceController"}}>

<i class="icon-plus icon-white"></i> {{t services.service.add}}</a>

</li>

（2） 屏蔽组件服务操作中“删除组件”

ambari-web/app/models/host\_component.js目录下：

找到DELETE\_SERVICE处(411-416)，在下方加入红字内容，具体如下：

DELETE\_SERVICE: {

action: 'deleteService',

context: ctx.get('serviceName'),

label: Em.I18n.t('services.service.actions.deleteService'),

cssClass: 'icon-remove',

isHidden: true

},

（3） 屏蔽主机“添加服务”

ambari-web/app/templates/main/host/summary.hbs目录下：

找到line29-44，删除如下红字内容：

{{#isAuthorized "HOST.ADD\_DELETE\_COMPONENTS"}}

<div class="btn-group pull-right">

<button id="add\_component" data-toggle="dropdown" {{bindAttr class="view.addComponentDisabled:disabled :btn :dropdown-toggle"}}>

<span class="icon-plus"></span>&nbsp;{{t add}}

</button>

<ul class="dropdown-menu">

{{#each component in view.addableComponents}}

<li>

<a href="javascript:void(null)" data-toggle="modal" {{action addComponentWithCheck component target="controller"}}>

{{component.displayName}}

</a>

</li>

{{/each}}

</ul>

</div>

{{/isAuthorized}}

（4） 屏蔽主机“删除主机”

ambari-web/app/views/main/host/details.js目录下：

找到line84-89，删除如下红字内容：

if (App.isAuthorized("HOST.ADD\_DELETE\_HOSTS ")) {

result.push({

action: 'deleteHost',

liClass: '',

cssClass: 'icon-remove',

label: this.t('hosts.host.details.deleteHost')

});

}

（5） 屏蔽主机组件“删除”

ambari-web/app/templates/main/host/details/host\_component.hbs目录下：

找到line137-141，删除如下内容：

<li {{bindAttr class="view.isDeleteComponentDisabled:disabled"}}>

<a href="javascript:void(null)" data-toggle="modal" {{action "deleteComponent" view.content target="controller"}}>

{{t common.delete}}

</a>

</li>