Slider App开发流程

Apache Slider自带HBase、Storm和Accumulo的服务实现，如果其他应用也采用Slider在Yarn中部署，需要安装Slider规范进行编写一系列的配置和脚本，然后定义打包文件。应用直接运行在Yarn中，每个应用都包括打包、部署及生命周期管理。下面对如何将App进行Sliderize化，进行介绍。

# 一、App安装

下面介绍jmemcached的Slider app安装包的定义，一个App package的目录格式应该如下所示：

unzip -l "$@" jmemcached-1.0.0.zip

Archive: /jmemcached-1.0.0.zip

Length Date Time Name

637 07-15-14 19:17 appConfig-default.json

1673 07-15-14 17:58 metainfo.xml

0 07-15-14 17:54 package/

0 07-15-14 18:03 package/files/

122880 07-15-14 18:03 package/files/jmemcached-1.0.0.tar

0 07-15-14 19:31 package/scripts/

1530 07-15-14 19:31 package/scripts/memcached.py

1287 07-15-14 18:46 package/scripts/params.py

1581 07-15-14 19:16 README.txt

252 07-15-14 17:58 resources-default.json

在安装包中，必须的文件为metainfo.xml，package目录（包括软件安装包，安装执行脚本），及默认的appconfig、resource定义文件。下面进行介绍：

## 1.1 metainfo.xml

该文件中至少要定义application，包括名字、描述及版本信息，在application中要至少要有一个components类型。下面是MEMCACHED的例子的定义文件，其中也包括了tarball的描述：

<metainfo>

<schemaVersion>2.0</schemaVersion>

<application>

<name>MEMCACHED</name>

<comment>Memcache is a network accessible key/value storage system, often used as a distributed cache.</comment>

<version>1.0.0</version>

<exportedConfigs>None</exportedConfigs>

<exportGroups>

<exportGroup>

<name>Servers</name>

<exports>

<export>

<name>host\_port</name>

<value>${MEMCACHED\_HOST}:${site.global.listen\_port}</value>

</export>

</exports>

</exportGroup>

</exportGroups>

<components>

<component>

<name>MEMCACHED</name>

<category>MASTER</category>

<compExports>Servers-host\_port</compExports>

<commandScript>

<script>scripts/memcached.py</script>

<scriptType>PYTHON</scriptType>

</commandScript>

</component>

</components>

<osSpecifics>

<osSpecific>

<osType>any</osType>

<packages>

<package>

<type>tarball</type>

<name>files/jmemcached-1.0.0.tar</name>

</package>

</packages>

</osSpecific>

</osSpecifics>

</application>

</metainfo>

在这个定义文件中，首先定义了Memcached的名称及版本号，<exportGroups>定义了使用的端口。<components>中定义了一个类型为MEMCACHED MASTER，同时指定了执行脚本memcached.py和执行环境。在osSpeccifics中定义了Memcached所使用的tar包，指定了其位置。如果没有安装tar包，需要自己构建一个。

## 1.2 resource-default.json

resource.json文件，定义了服务中有哪些components。每个应用都必须有一个slider-appmaster，用于container的申请和管理。增加MEMCACHED组件，可以指定优先级、实例的数目及yarn.memory。下面是一个实例：

{

"schema" : "http://example.org/specification/v2.0.0",

"metadata" : {

},

"global" : {

},

"components": {

"slider-appmaster": {

},

"MEMCACHED": {

"yarn.role.priority": "1",

"yarn.component.instances": "1",

"yarn.memory": "256"

}

}

}

## 1.3 appConfig.json

该文件用于定义应用的配置信息，其中有几项是必须要定义：

1）application.def，应用安装包所在的位置，默认是HDFS中

2）java\_home，目标主机的jdk所放路径

3）package\_list，app tarball在app package中的位置，以files开始的相对路径（该项在0.60中没有，可能使用了默认的位置）

4）config\_types:一般是空值

5）site.global.app\_user，运行app的用户

6）site.global.app\_root，app root的位置，Slider将tarball解压缩到${AGENT\_WORK\_ROO

T}/app/install目录，因此根据application tarball的结构附件一个安装包的相对路径，例如jmemcached-1.0.0,（这个一定是解压后的目录名）

可以为应用添加其他的参数，这个根据需要进行定义，格式一般为：site.global.xxx

下面是一个例子：

{

"schema": "http://example.org/specification/v2.0.0",

"metadata": {

},

"global": {

"application.def": ".slider/package/MEMCACHED/jmemcached-1.0.0.zip",

"java\_home": "/usr/jdk64/jdk1.7.0\_67",

"site.global.additional\_cp": "/usr/lib/hadoop/lib/\*",

"site.global.xmx\_val": "256m",

"site.global.xms\_val": "128m",

"site.global.memory\_val": "200M",

"site.global.listen\_port": "${MEMCACHED.ALLOCATED\_PORT}{PER\_CONTAINER}"

},

"components": {

"slider-appmaster": {

"jvm.heapsize": "256M"

}

}}

## 1.4 基本命令（脚本）

所有的Appliation都需要实现以下几个命令INSTALL/CONFIGURE/START/STOP/STA

-TUS，一些命令可以使空的。这些命令都是通过py脚本来实现。

在slider agent/resource\_management中，已经实现了公用的执行脚本，如script.py,confi

g-dictionary.py等。在MEMCACHED的例子中，实现了两个脚本：parameter.py和Memcached.py：

1）parameter.py，该脚本用于从配置文件中获取应用参数，该参数可以定义在appConfig.json或者自定义xml文件中。

from resource\_management import \*

config = Script.get\_config()

app\_root = config['configurations']['global']['app\_root']

java64\_home = config['hostLevelParams']['java\_home']

pid\_file = config['configurations']['global']['pid\_file']

additional\_cp = config['configurations']['global']['additional\_cp']

xmx\_val = config['configurations']['global']['xmx\_val']

xms\_val = config['configurations']['global']['xms\_val']

memory\_val = config['configurations']['global']['memory\_val']

port = config['configurations']['global']['listen\_port']

2）memcached.py，该脚本用于实现应用的启动等各个操作，脚本要实现的功能基本上要有展开给定的tarball，读取配置及生成命令字符串，执行启动命令，将PID写入文件中（用于检测daemon的状态信息）。下面是Jmemcached的例子：

import sys

from resource\_management import \*

class Memcached(Script):

def install(self, env):

self.install\_packages(env)

def configure(self, env):

import params

env.set\_params(params)

def start(self, env):

import params

env.set\_params(params)

self.configure(env)

process\_cmd = format("{java64\_home}/bin/java -Xmx{xmx\_val} -Xms{xms\_val} -classpath {app\_root}/\*:{additional\_cp} com.thimbleware.jmemcached.Main --memory={memory\_val} --port={port}")

Execute(process\_cmd,

user=params.app\_user,

logoutput=False,

wait\_for\_finish=False,

pid\_file=params.pid\_file

)

def stop(self, env):

import params

env.set\_params(params)

def status(self, env):

import params

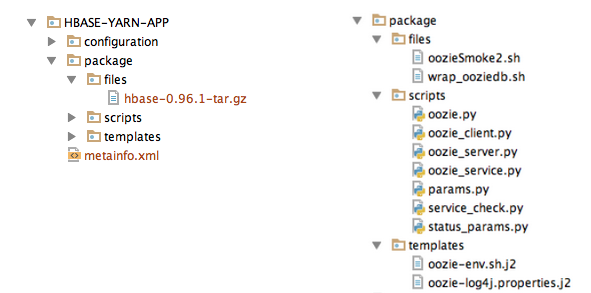
env.set\_params(params)

check\_process\_status(params.pid\_file)

# 二、App Package的文件结构

所有应用包、定义、配置和脚本都被打包到一个压缩文件，可以通过Yarn应用部署公里，包括slider进行包的转换。

App安装包是一个Zip package，包括所有的应用相关文件，如下图所示：



1. app definition file，应用的结构、内容、定义、支持的平台和版本等
2. default configurations目录，和应用相关的配置，及文件
3. cmd\_impl folder，应用/组件的管理操作
4. scripts folder，实现管理操作的不同脚本
5. templates folder，应用部署的不同模板
6. files folder，其中脚本,txt文件及tarballs等

## 1.1 应用定义文件:metainfo.xml

# 附录：maven文件打包