Simplified discovery of services via DNS mechanisms

——YARN-4757

# 1.简介

在YARN集群中，Service启动的主机和端口都是不确定的，YARN-913:Add a way to register long-lived services in a YARN Cluster在服务启动时提供了注册机制，将服务的binding info注册到ZK中，客户端可以通过查询ZK获取服务的绑定信息，然后进行访问。

但是，客户端需要通过JAVA 或者REST API来获取信息，对于已有的客户端程序，不修改代码很难访问到YARN分配的binding信息，而且在运行过程中，Service可能会因为重启或者异常导致bind信息的动态变化。

使用DNS域名解析机制可以将registry info提供给客户端，可以使客户端和Registry Service解耦，而对binding info的变化不会对客户端有影响。下面介绍yDNS(YARN DNS service)的实现，该方案使用标准DNS 接口将部署的应用在YARN Registry中的信息导入到DNS服务器中：

1. 通过DNS向外提供已有的service-discovery信息，将YARN Service registry's records转换为DNS记录，从而使用户可以通过标准的DNS客户端机制（例如DNS SRV记录，描述host:port）查询YARN Applciation信息
2. 实现Containers和IP的映射，通过DNS lookup获取containers的ip地址。

通过该机制，可以实现container name-based communication，例如：curl

http://mycontainer.myDomain.com/ednpoint即可访问myContainer。

注：相关JIRA:https://issues.apache.org/jira/browse/YARN-4757

# 2.功能及系统架构介绍

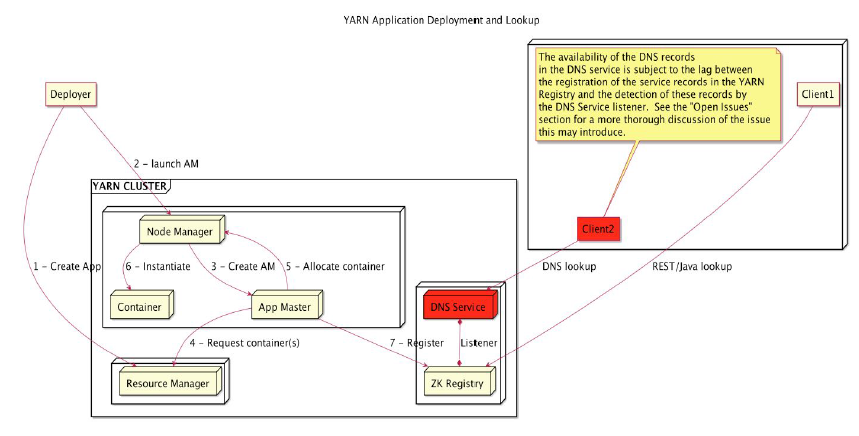
yDNS要实现如下功能：

1. 根据yarn applications注册到registry service中的endpoint信息创建DNS Records
2. containers或者applications的重启不会引起DNS Record names的变化
3. 支持reverse lookups（根据IP查询name）
4. 通过DNSSEC(Domain Name System Security Extensions)实现安全机制
5. 高可用性
6. 可扩展，查询的低延时及高吞吐量

而且，yDNS不能作为primaray dns server（不接受其他DNS Server的查询请求），yDNS Server可以和已有的DNS（例如hadoop zone/domain server）配合使用。yDNS需要接受TCP和UDP的接口，DNS protocols的默认接口是管理接口53，所以在部署时需要通过admin用户进行配置。

## 2.1 系统架构

如下所示：



yDNS系统架构与执行流程

yDNS的工作流程如下：

1. deployer向RM提交app creation请求并进行前期工作，RM向deployer返回AM allocation
2. deployer根据allocation分配结果，与NM交互，并在NM上启动AM container
3. AM完成初始化工作以后，与RM交互，获取Container Allocations，然后根据Allocations结果与NM交互并启动container
4. AM与ZK Registry交互，将app的binding info，写到zk中，其中包括application-level和container-level的service records
5. yDNS为zk registry增加listener，监听到application事件后，获取service records，并生成DNS Records
6. 客户端有两种方式访问服务，一种是直接通过JAVA/REST API从zk registry中获取binding info，第二种是通过dns lookup获取application的binding info

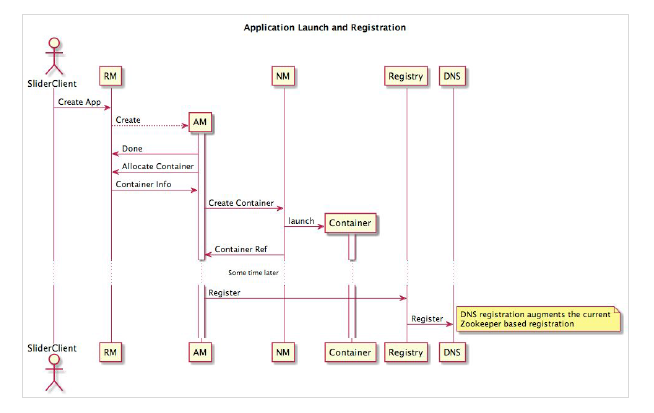
以上是Application启动及yDNS的使用流程

## 2.2 执行时序图

yDNS Service有多个关键执行序列，下面进行简单的分析

### 2.2.1 Application launch and registration

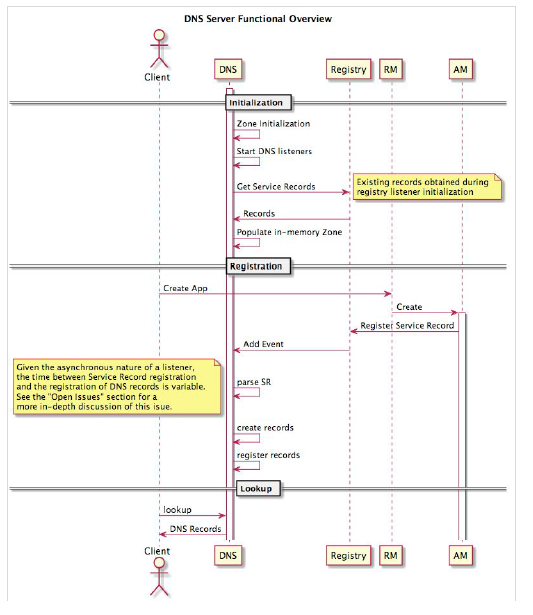
application常规launch和registration的基础上，增加了将application和containers相关信息注册到DNS服务中，过程如下：



DNS Service通过监控zk-based YARN registry来执行CRUD 操作将application binding信息写入DNS Server。监控是通过为zk添加Curator Tree Cache Listener来实现，需要处理的事件包括zk node的增加，删除及更新等，node相关的Service records会根据规则转换为DNS Records。

### 2.2.2 DNS Service internals

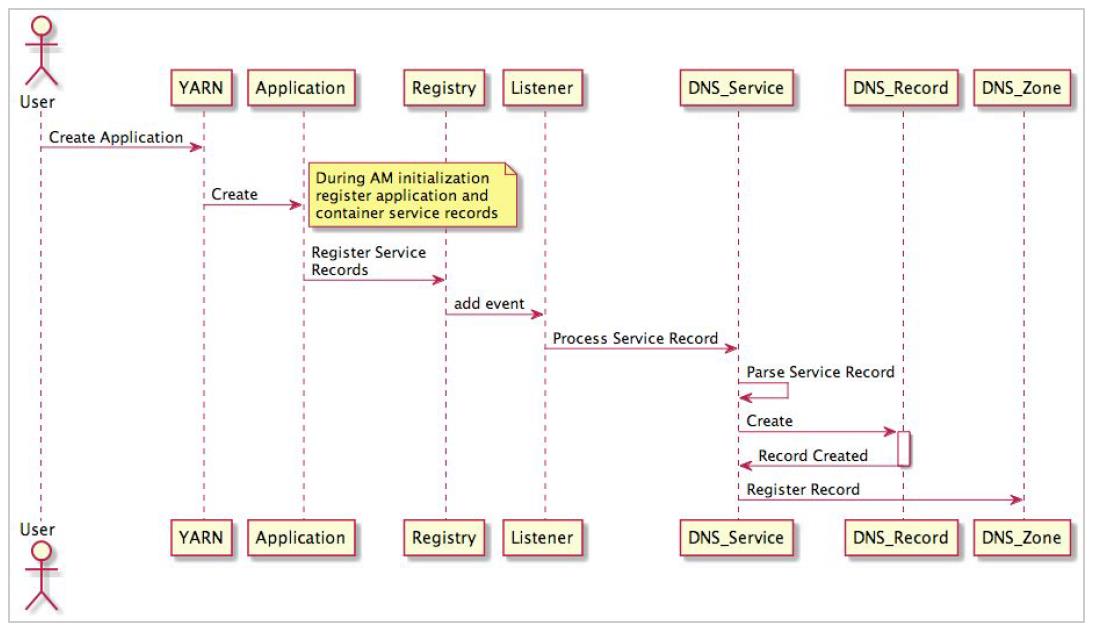
DNS Service内部实现的主要功能如下表所示：



DNS Service的启动初始化过程包括zone initialization，启动DNS listners，并从YARN Registry中获取已启动应用的信息，并生成记录。在客户端创建应用后，会触发add事件，根据事件的信息，得到相应的Service record并添加仅DNS Server。DNS service接收客户端的lookup请求，并返回DNS Record。

### 2.2.3 DNS Record的创建

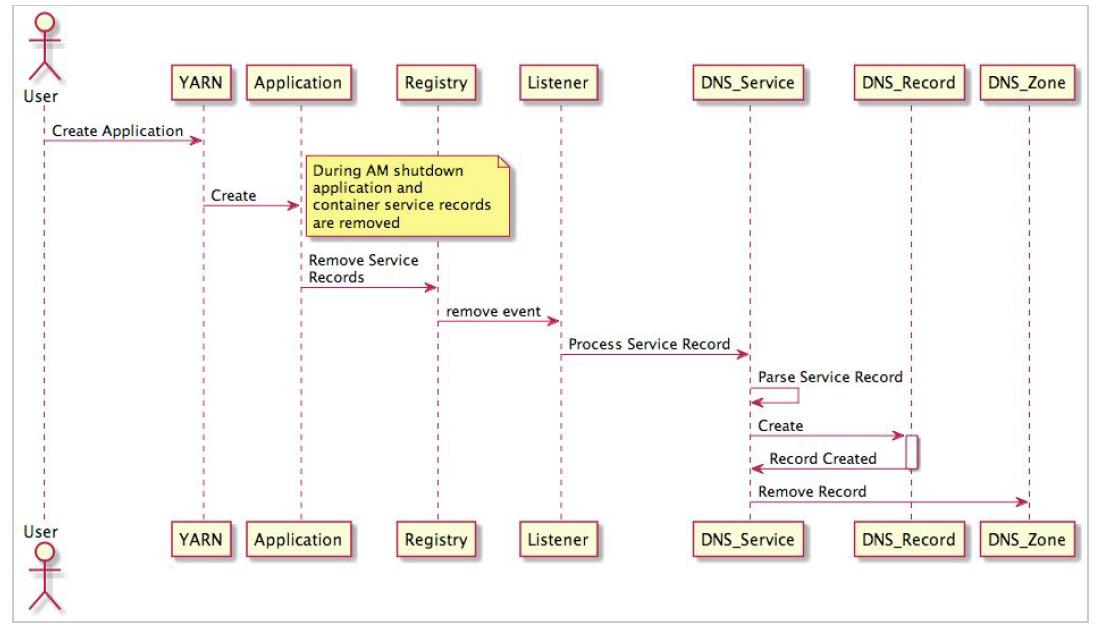
下表描述了DNS Record的创建和注册序列，service record的更新过程类似（仅触发事件不同）



用户创建应用后，将service records写入YARN Registry中会触发add event，DNS Service监听到add事件后，读取service record，并生成DNS Record，并写到DNS Zone中

### 2.2.4 DNS Record removal

移除过程如下所示：



DNS Service监听到YARN Registry remove event后，根据事件中的Service record，生成对应DNS Record，并从DNS Zone中移除。

## 2.3 DNS Service deployment

DNS Service会在配置的端口中初始化UDP和TCP listeners，默认53端口仅可以通过管理权限来进行访问。DNS Service接收到DNS Requests（standard dns requests），这些requests来自Users或者其他的DNS Servers（以YARN DNS Service作为forwarder）。

DNS进程的执行环境可以有多种选择，例如运行在RM或者ZK主机上，也可以运行在单独的进程中。单独运行有以下好处：

1）可以单独监控yDNS的内存、cpu及线程的性能

2）可以部署多个yDNS实例，实现高可用性

但是将yDNS嵌入到RM或者ZK中也是可行的方案，而且单独部署需要一定管理的开销，同时DNS Service和YARN Registry Service之间会有一定的异步性。

## 2.4 High Availability

可以通过Standard DNS mechanisms来实现yDNS的高可用性，yDNS的instances是zk-based YARN Registry的一个数据映射，本身不存储自身的数据，是stateless的服务。

为了实现yDNS的高可用性，可以通过启动多个DNS Service instances，并进行相应配置即可实现，每个DNS Server都注册为YARN Registry的listener，接收相同的注册事件，并创建相同的DNS Records集合。配置一个Parent DNS Server，将yDNS Service作为该DNS的forwarders。例如，一个Parent DNS Server有以下配置：

zone "hadoop.site" IN {

type forward;

forward first;

forwarders { host1 port 9900; host2 port 9001; };

};

Parent DNS Server将”hadoop.site”的请求首先发送到列表中的第一个Server，然后没有建立连接，则第二个Server则响应该请求，依次类推。

YARN DNS Service被设计响应YARN APP相关的zone的请求，并不适合做primary server，这样可以不用考虑安全性，DOS attacks等威胁，也不用设置缓存策略增强功能，而且primary Server相关的配置及管理都比较复杂。

# APP Service Record && DNS Record

前面介绍了yDNS 的系统架构和工作流程，下面介绍App Service Record转换为DNS Record。

## 3.1 yDNS Element

下面是yDNS中会涉及到的elements(labels)，这些elements和DNS 规则（RFC 1035）是相兼容的：

1. domain，在配置项中配置，作为cluster DNS Domain的名称，可用于父dns服务器定义yDNS Zone的名称，例如yarncluster.com
2. user-name，应用部署者，作为DNS names的一部分
3. application-name，部署的yarn app的名称，从yarn registry path中获取。以application-name而不是application-id作为dns names的element，主要是考虑可读性
4. container id，yarn给container分配的id
5. component-name，component的名称，例如master component。虽然component name并不是yarn的核心概念，但是作为yDNS entries的一部分很有用
6. api，用于提供给客户端或用户的enpoint、

通过以上多种elements/labels的组合成DNS Record的name，便于区分创建的服务。Cluster的domain name常作为最后的element。

下面是一个name的例子：management-api.griduser.yarncluster.name，从右向左解析（不包括yarncluster.name ，domain name），第一个element为application-user。为了便于区分，常使用前缀例如container-或者api。

在yarn中并不能保证application name的唯一性（slider中不存在这个问题）,yDNS Service将zk path中的最后一个element作为application name，与user name组合可保DNS record name的唯一性。

## 3.2 DNS Record

yDNS将application registry中的application和container信息转换成以下记录：

1. A and AAA records，FQDN和IPV4/IPV6的映射，最常用的记录
2. CName Record，别名记录，该记录允许yDNS为特定的container或者services生成多个name
3. SRV Record，将DNS Record name映射service的<hostname|port>
4. TXT Record，根据DNS Record name获取任意格式数据，

例如关于server,network,datacenter或者其他account信息

1. PTR Records，pointer records，根据ip获取host name

## 3.3 yDNS中的DNS Records

### 3.3.1 Per-Application records

每个application的record有两种类型

1. Names that maps to an application

格式如下：\*.<application-name>.<user-name>.<domain>

Example，Application:zkapp1，user:gridUser，domain:yarncluster.com，

names为zkapp1.griduser.yarncluster.com

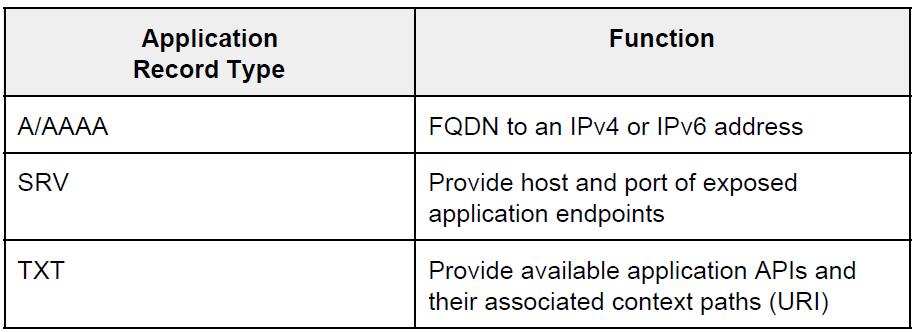
Record typs:

A and AAA，提供API URIs的主机名

CNAME， address records

SRV Records， yarn registry 定义中的host/port格式的apis

TXT Records，API types及相关的context paths



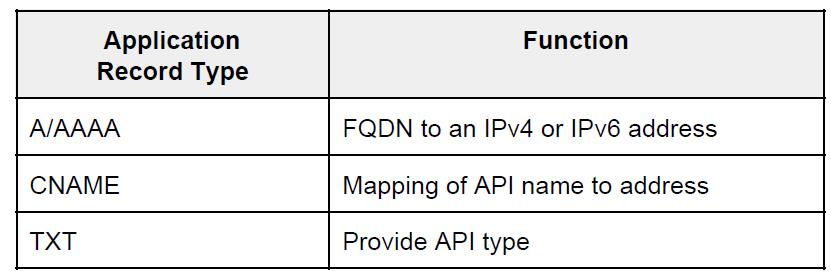
1. Application-level API相关的records

API Records(CNAME和TXT records)，AM hosted的URIS

FORMAT :<api>-api.<application-name>.<user-name>.<domain>

例如：publisher-exports-api.zkapp1.griduser.yarncluster.com

api的前缀用于区分record names



### 3.3.2 Per-Container Records

Per-Container的records有两种types， A/AAA records

1. Names map to a component

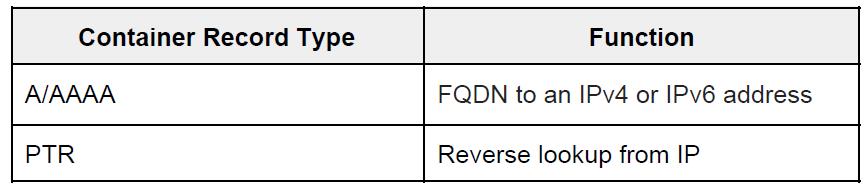
格式如下：<container id>.<domain>

例如：containere37411454001598828013101000004.yarncluster.com

Record types:

A和AAA记录，container ID => container IP

reserve lookup（PTR）,ip => container ID



1. Names map to a Component

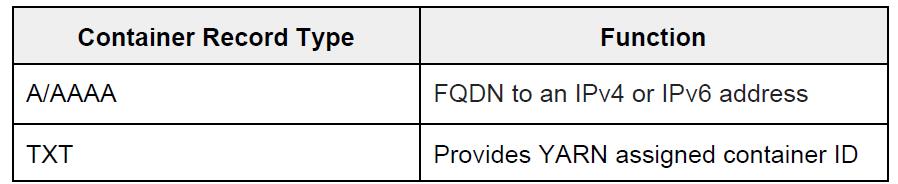
格式: <component>.<application-name>.<user-name>.<domain>

例如：zk1.zkapp1.griduser.yarncluster.com

Record types:

A和AAA记录，component name => container IP

reserve lookup（PTR）,ip => component ID(container ID的列表)



应用会有多个component实例，例如hbase regsion servers，开发者可以采用将相同的component部署多个containers，yDNS会产生多个DNS记录或者component使用唯一名称，对应单条DNS记录

## 3.4实例

下面是一个根据yarn registry application registrations生成dns records的一个实例，仅是yDNS的一个使用简单举例：

1. Registry path的最后一个element，被称为application-name，registry path如下:/registry/users/aUser/services/org-apache-yarn/myApplicatioName，生成的appName为myApplicationName
2. 另外，也会从application node中获取user-name，例如上面的aUser

3）custom Aliases

应用的部署者可以在程序中设置component的DNS registry信息，例如在部署过程中，一个componet的配置如下：

"components": {

"sliderappmaster":{

"jvm.heapsize": "512M"

},

“MY\_COMP": {

“service.record.attribute.name”: “mycomponent"

}

}

通过slider可以获取一个record的name attribute：

{

"type" : "JSONServiceRecord",

"description" : "MASTER”,

**“name” : “mycomponent”,**

"external" : [ ],

"internal" : [ ],

"yarn:id" : "container\_e03\_1449766058071\_0003\_01\_000002",

"yarn:persistence" : "container",

"yarn:ip" : "172.17.0.20",

"yarn:hostname" : "myContainer"

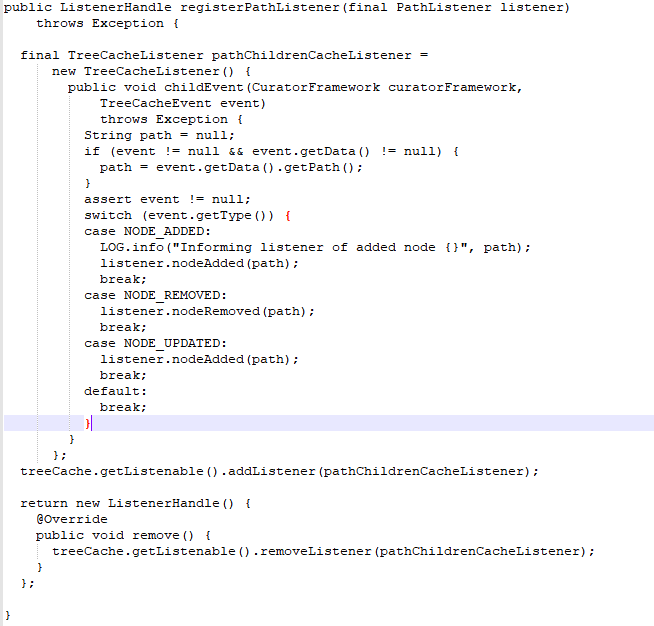
}

yDNS service根据获取到的registered record，并生成DNS记录，如下所示：

<alias>.<username>.<domain> , e.g. mycomponent.griduser.yarncluster.com.

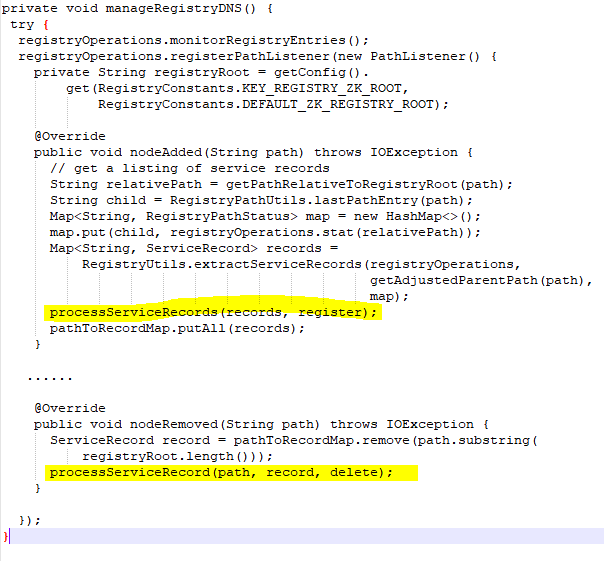
## 3.5核心实现

根据以上流程的分析，yDNS的核心功能是通过对ZK中registry信息进行watch，如下所示：



当启动的服务注册到zk时，将其解析出来添加DNS 记录，

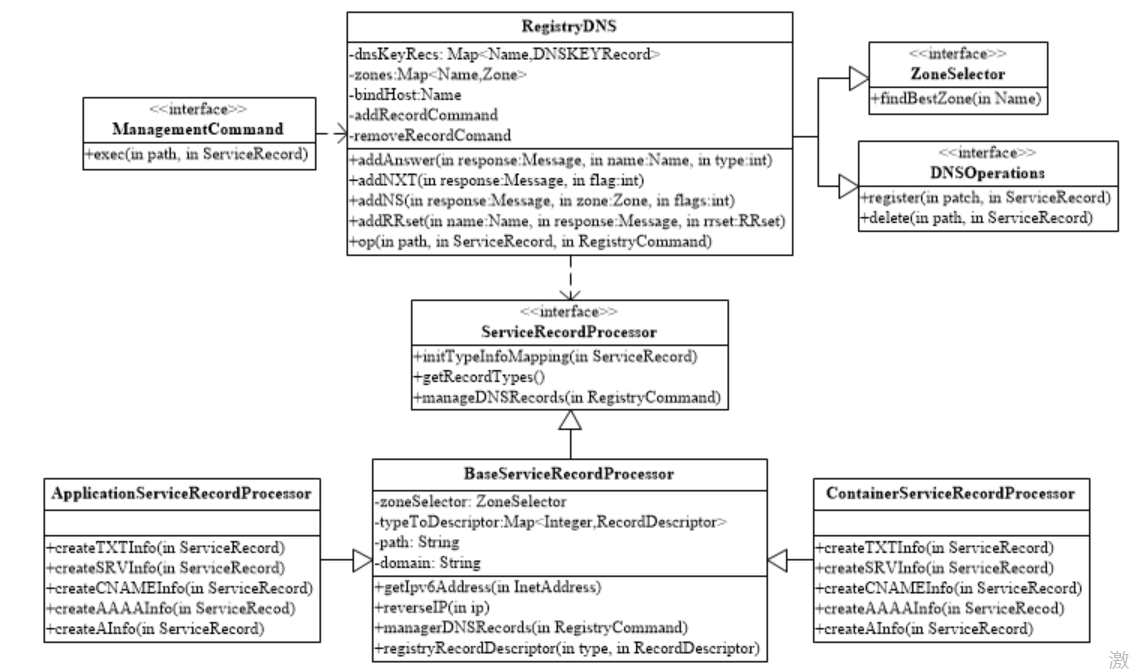
RegistryDNSServer#managerRegistryDNS



对应用创建A record的执行如下：

*protected void createAInfo(ServiceRecord record) throws Exception {  
 AApplicationRecordDescriptor recordInfo = new AApplicationRecordDescriptor(  
 getPath(), record);  
 registerRecordDescriptor(Type.A, recordInfo);  
}*

为Application创建A记录。相关类图如下：



# **YARN DNS的启动和使用**

1）配置参数

core-site.xml中增加配置如下：

*<property>*

*<name>hadoop.registry.dns.domain-name</name>*

*<value>ycluster</value>*

*</property>*

*<property>*

*<name>hadoop.registry.dns.bind-port</name>*

*<value>5335</value>*

*</property>*

*<property>*

*<name>hadoop.registry.dns.enabled</name>*

*<value>true</value>*

*</property>*

*<property>*

*<name>hadoop.registry.dns.bind-address</name>*

*<value>0.0.0.0</value>*

*</property>*

*<property>*

*<name>hadoop.registry.zk.quorum</name>*

*<value>fys1.cmss.com:2181,fys3.cmss.com:2181,fys2.cmss.com:2181</value>*

*</property>*

配置Registry DNS反向查询，增加参数：

*<property>*

*<name>hadoop.registry.dns.zone-mask</name>*

*<value>255.255.255.0</value>*

*</property>*

*<property>*

*<name>hadoop.registry.dns.zone-subnet</name>*

*<value>172.17.0.0</value>*

*</property*

1. 启动DNS Server，命令如下：

*yarn --daemon start registrydns*

启动进程如下：

*$ /usr/jdk64/jdk1.8.0\_77/bin/jps*

*18864 RegistryDNSServer*

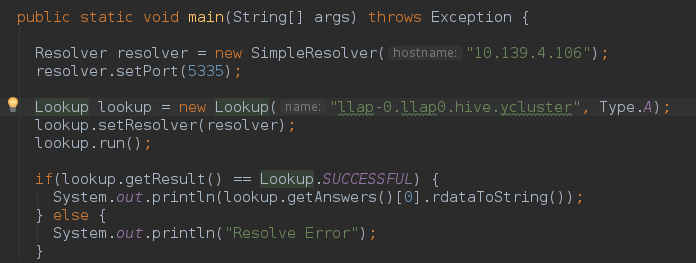
*$ netstat -anp|grep 18864*

*tcp 0 0 0.0.0.0:5335 0.0.0.0:\* LISTEN 18864/java*

1. 配置主机DNS，配置/etc/resolve.conf

*nameserver 10.139.4.106#5335*

1. 启动Hive LLAP后，通过JAVA Api查看其注册的主机信息如下：



输出主机ip地址为：10.139.4.106，其在DNS中的记录如下：

