Simplified discovery of service via DNS mechanisms

# 1.简介

应用使用YARN service Registry将endpoints注册到zk中，便于客户端进行查询并访问这些应用。但是，YARN service Registry仅能通过JAVA API或者REST接口获取信息。这使已有客户端和服务访问这些应用比较复杂。由于应用的endpoints动态变化，因此已有服务很难在不改写代码的情况下通过现在的注册读机制来实现访问，例如已有服务必须修改代码，通过REST或者JAVA API来访问YARN Service registry来连接HBase instances。

一个较好的方式是通过DNS将注册信息提供给用户或者已有服务。下面介绍一种新的服务来解决这个问题：YARN DNS service(简称yDNS或者DNS for YARN，DARN)，作用是为YARN Registry提供DNS的标准接口，其实现以下功能：

1. 通过DNS向外提供已有的service-discovery信息，YARN Service registry's recors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. Containers和IP的映射关系，通过container的名字访问，

例如curl http://myContainer.myDomain.com/endpoint，但是reserve查询会有问题，一个ip会对应多个containers

# 2.YARN Service Registry

在Hadoop Cluster中YARN Service Registry使应用可以注册到zk中，然后提供给第三方软件连接方式，当前：

1. Applications可以通过JAVA Registry lib将信息提交到registry
2. Client可以通过Service Registry提供的JAVA和REST APIS，定位服务，并通过network-accessible endpoinsts连接服务，例如REST,IPC,WEB UI和zk quorum path等

# 3.Key Requirements for a DNS-based Service Registry

已有的YARN Service Registry可以作为DNS Service的数据源，因此DNS-based registry需要实现以下功能特性：

1. 支持为YARN Applications的end-points创建DNS records
2. 重启containers或者applications，Record name不变
3. 支持reverse lookups
4. 支持DNSSEC(安全协议)
5. HA
6. Scalable，低延迟

部署特性：

1. 支持和Existing DNS融合，共同作为Hadoop DNS Server。该Server不作为primaray dns server，而且不需要接受其他servers的请求
2. 该DNS Service可以接受DNS标准的TCP和UDP请求。端口在部署时指定

# 4.Discovery vial DNS Records

介绍DNS支持的记录类型，然后YARN ZK-backed registry中的记录如何通过yDNS解析成DNS记录。

## 4.1 Applicable Record Types

yDNS将application registry转换成以下记录，这些记录提供了application和container信息：

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生产多个name
3. SRV Record，A Service Record，服务的hostname和port
4. TXT Record，host或者其他名字相关的任意格式数据，例如关于server,network,datacenter或者其他可读信息
5. PTR Records，pointer records，用于ip和name的映射

## 4.2 DNS相关规则

### 4.2.1 yDNS elements

下面是yDNS中会涉及到的labels，这些elements必须和DNS规则相兼容

1. domain，DNS Domain，在配置项中配置，定义yDNS zone，可用于父dns服务器将该zone传到yDNS，例如yarncluster.com
2. user-name，应用部署者，为了兼容dns name，需要将names转换特殊字符串例如hyphens及spaces
3. application-name，部署的yarn应用的名称，从yarn registry path中推断。将application name作为applicatio-id，便于用户访问
4. container id，yarn给container分配的id
5. component-name，分配给部署component的名称，例如master component
6. api，用于exposed的enpoint，例如publisher

在大多数的情况下，通过以上多种elements/labels的组合来区分服务。Cluster的domain name常作为最后的element。从右向左解析，第一个element为application-user，在某些不好区分的地方，使用前缀来区分，例如container- / api等。例如一个management endpoint可以写为：management-api.griduser.yarncluster.name

目前yarn中不保证application name的唯一性，但是Apache slider可以保证。yDNS将zk path的最后一个element作为apoliction name。同一用户下的同一applicatio name会注册失败。

### 4.2.2 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

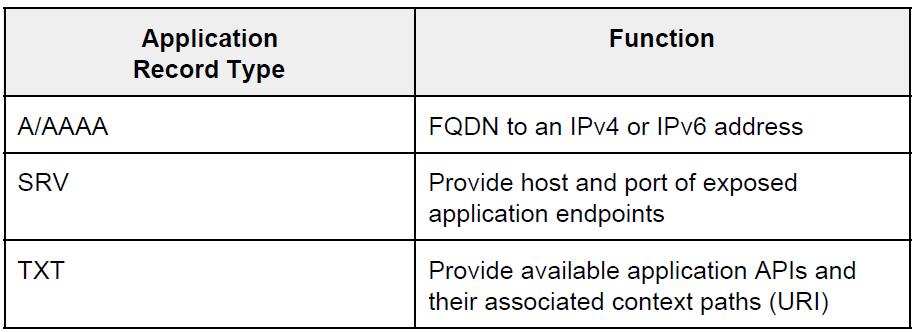
Record typs:

A and AAA，提供API URIs的主机

CNAME，API record name

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

TXT Records，API typ



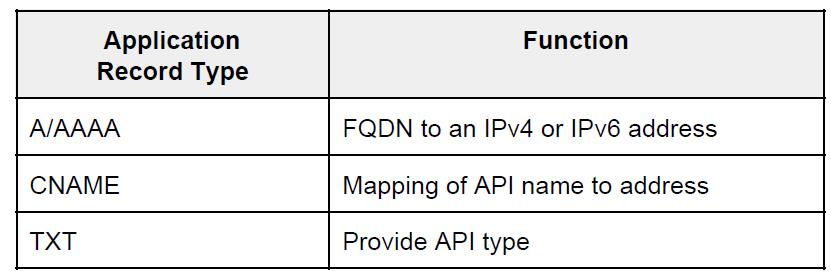
1. Application-level API相关的records

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



### 4.2.3 Per-Container records

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

**Names map to a component**

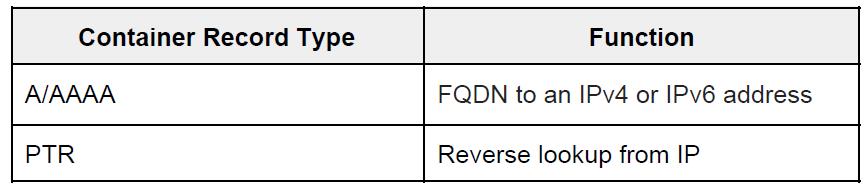
格式如下：<container id>.<domain>

例如：containere37411454001598828013101000004.yarncluster.com

Record types:

A和AAA记录，container ID => container IP

reserve lookup（PTR）,ip => container ID



**Names map to a Component**

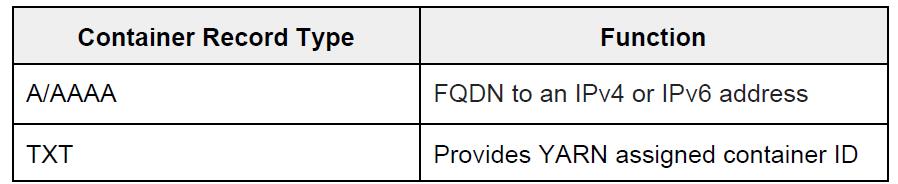
格式: <component>.<application-name>.<user-name>.<domain>

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)



一个应用会有多个component实例，例如hbase regsion servers，开发者有以下选项：

相同的component部署多个containers，yDNS会产生多个DNS记录

component有唯一名称，相应单条DNS记录

### 4.2.4 Registry Service Record Parsing Guidelines

下面是yDNS如何根据yarn registry service生成dns记录

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

应用的部署者也可以选择其他机制获取给定的components中的dns registry信息，例如slider，在部署过程中，应用可为某个component设置参数：

"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，Jira号为：

<https://issues.apache.org/jira/browse/SLIDER1086>)

{

"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读取registerd record，并生成DNS记录，如下所示：

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

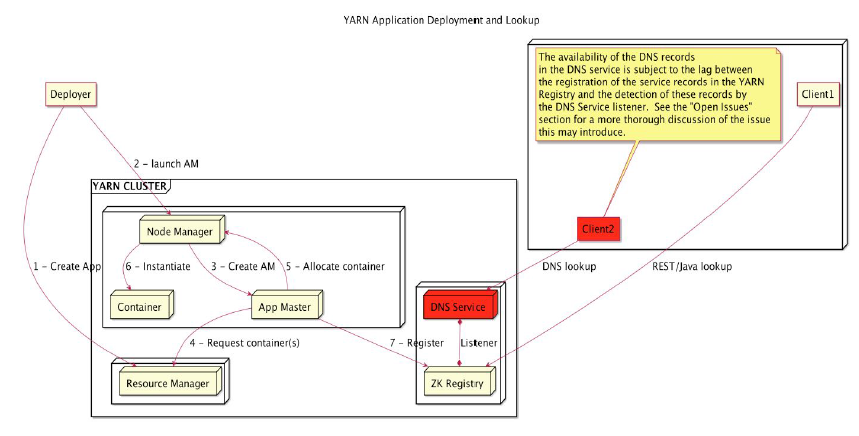
注意事项：

1. 为了保证明确性，不能使用container作为name的前缀
2. Records中有user name，防止不同用户创建相同应用。

# 5.系统架构

## 5.1 架构简介

如下所示：



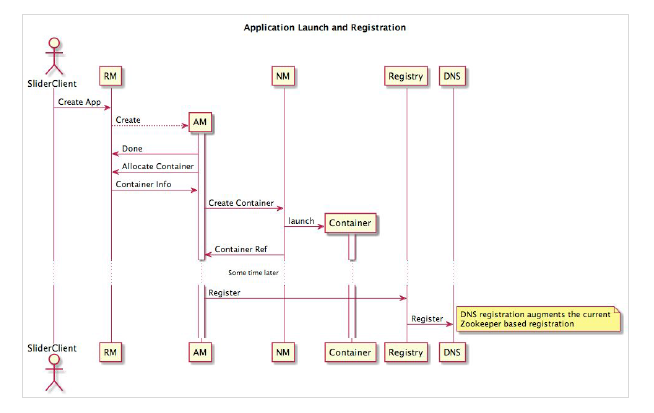
1. A deployer与RM交互，初始化应用，从AM接收container allocation
2. 与NM交互，并在分配的主机上初始化AM container
3. NM启动AM Container
4. AM初始化请求，并从RM中获取container allocations
5. AM与NM交互，并初始化containers
6. NM初始化containers，并根据am提供的信息产生launch context
7. AM与ZK-based yarn registry交互，注册application-level和container-level的service records
8. DNS-based Service从ZK-based yarn registry中获取application和container info

## 5.2 序列图

DNS Servvices需要执行以下核心功能：

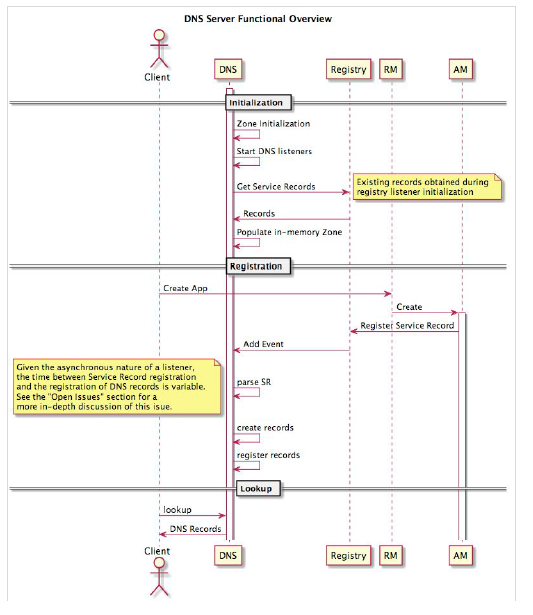
1. 当前application launch和registration sequences需要添加DNS（与application和containers）相关信息的注册

序列图如下：

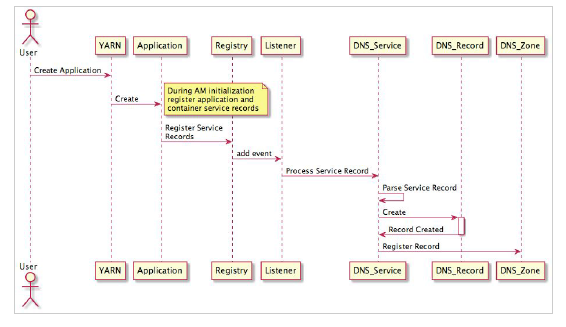


1. DNS Service通过监控zk-based yarn registry执行CRUD操作。添加A listener，例如Curator Tree Cached Lisntener，监控ZK node的变化并处理，同时更新DNS记录。

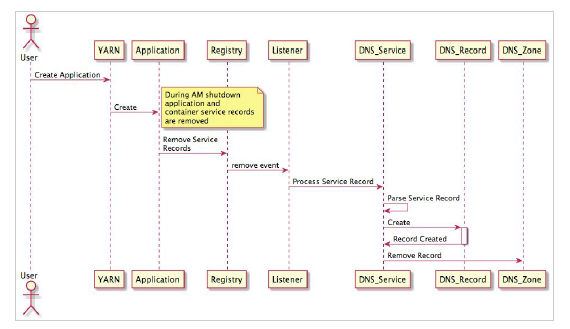
DNS Service的作用如下图所示：



下表描述了DNS record的创建和注册的序列，不同的事件类型，不同的处理步骤，但是大致相同：



Record remove序列如下：

DNS Service的部署：

1. DNS Service 对外提供TCP和UDP Lisnters，默认端口53(管理员端口)
2. DNS service监听内部DNS Requests，而且是标准DNS requests
3. DNS的进程执行环境，在RM中hosted,zk中hosted或者单独的进程，但是作为单独的进程有以下好处：可以分析yDNS的memory,cpu和线程，方便搭建DNS HA，不影响RM和ZK

但是不足点是单独的进程需要额外的管理，DNS service和zk及yarn registry的同步。

## 5.3 High Availabilty

yDNS的高可用性可以通过DNS本身机制来实现， yDNS instances仅是ZK-backed YARN Registry的显示，是stateless的服务，所以yDNS的HA不需要通过其他额外的机制来保证。

其高可用性可以通过启动多个DNS-Servers来实现，为了查询所需要的信息，客户端迭代查询这些DNS Server直到获取响应，机制的实现可以是：

1. resolve.conf，在配置文件中配置迭代的dns
2. zone forwarders，当前的zone可以定义多个forwarders。

这些DNS Service instances为yarn registry 设置listener，接收相同的服务注册事件，并创建相同的DNS records。这些DNS Services配置一个parent DNS Server，当某个DNS Service无效时，通过parent DNS Server来实现请求转发。例如，parent server中可以定义如下：

zone "hadoop.site" IN {

type forward;

forward first;

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

};

根据该配置，parent DNS Server将hadoop.site Domain相关的请求先发送到第一个server中，如果无响应，则依次查询列表中的Servers。

## 5.4 On-premise deployment

YARN DNS Service作为deployed yarn applications的一个zone，也就是说不能作为像返回nslookup google.com响应的primary server，因此不能配置到resolv.conf文件中，原因如下：

1. 功能完善的DNS Server具有认证及DOS等安全特性，能够缓存数据，因此需要一个DNS Server，专门和existing DNS Servier一致。
2. 典型的DNS服务由一个parent server，代理多个servers，响应特定的DNS Domains/zones。该配置由多个相关的DNS组成

支持reverse-lookup，forward lookups很重要，但是reverse looups辅助DNS Client使用的场景，yDNS需要支持：

1. 支持reverse looup zone的创建
2. 应用的部署和销毁触发PTR Records的注册和移除，Pointer Records用于实现network interface和hostname的映射
3. DNSSEC支持reverse zone（forword zone support的mirror，使用相同的public和private keys）

其他的特性，如当前DNS Server查询不到记录，需要转发到primary resolver forwarding。支持zone file的读取，包括提供给clients的静态records，例如和cluster hosts相关的address或者records，这些配置会预先加载到server中。

## 5.5 Security/DNSSEC Support

DNSSEC支持DNS Data的认证和完整性验证，包括对已有DNS数据的访问拒绝机制。YARN DNS Service不提供DNSSEC相关交互。如果yDNS支持DNSSEC，需要以下：

1. 读取配置好的DNSSEC pubkey
2. 注册DNSKEY public key记录
3. 读取标准的DNS private key file
4. 为需要的zone records(SON,NS)签名
5. 为query response创建签名
6. 返回标准的拒绝响应格式

Hadoop组件的安装认证机制和yDNS无关，这些组件不和users直接交互，这些组件和Yarn registry相互独立。

# 6.Implementation details

yDNS的实现有多种方案，如下所示：

## 6.1 Bind Server

BIND是实现DNS协议的开源软件，yDNS需要为BIND增加yarn registry的listeners，解析service records，通过update protocol messages发送DNS records请求

优点：

DNS兼容

FULL DNSSEC支持

最小代码量

缺点：

配置与维护复杂

动态zone管理

Statefule, BIND Server需要从yarn registry中获取状态，并写入zone文件中

为了支持ha，需要启动多个BIND Server

## 6.2 Other Service Discovery Solutions

还有其他的解决方案，如Consul和SkyDNS，这些方式也是通过DNS protocols来查询这些服务

优点：

已有技术

提供其他的Enterprise feature

API接口简单，JAVA/REST

缺点：

管理复杂

缺少DNSSEC

接口不完全

## 6.3 配置

YARN DNS server从yarn-site.xml中读取配置，相关的配置如下：

|  |  |
| --- | --- |
| 配置 | 描述 |
| hadoop.registry.dns.enabled | Cluster是否启用dns，默认是false |
| hadoop.registry.dns.domain-name | Hadoop cluster records相关domain name |
| hadoop.registry.dns.bind-address | Dns listenter需要绑定的网络address |
| hadoop.registry.dns.bindport | Dns listener的端口，默认53 |
| hadoop.registry.dns.dnssec.enabled | 是否开启dnssec |
| hadoop.registry.dns.publickey | Server的pubkey |
| hadoop.registry.dns.privatekeyfile | 私有密钥key file |
| hadoop.registry.dnsttl | DNS records的TTL值，默认1 |
| hadoop.registry.dns.zonesubnet | 和cluster containers相关的subnet indicator |
| hadoop.registry.dns.zonesdir | 放置zone 配置文件的目录 |

# 7.相关Jar

dnsJava

<dependency>

<groupId>dnsjava</groupId>

<artifactId>dnsjava</artifactId>

<version>2.1.7</version>

</dependency>

附录：相关Jira:

https://issues.apache.org/jira/browse/YARN-913

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

[https://issues.apache.org/jira/browse/SLIDER-1086](https://issues.apache.org/jira/browse/SLIDER1086)