

dns -- Balancing mail connections using the DNS

Carlo Contavalli

ccontavalli at masobit.net

This document describes how to use your own DNS as a load balancer for the whole PigeonAir (<http://www.pigeonair.net/>) project.

The proposed setup works well using Bind 8.0.0 and above, but is believed to work with most Bind versions.

1. Before starting

This document was written as part of the documentation of the PigeonAir Project to provide help and support to users, system administrators or developers.

While every effort has been made to ensure that the information is accurate at the time of publication, this document may contain errors, omissions, incongruences or wrong technical details. No liability for damages is accepted by the Author/Authors, the publishers or any other organization or person providing the information, arising from any errors or omissions that may appear, however caused.

In case you find an error, you would like to propose better solutions than those discussed in this document or you would like to discuss an idea regarding this document or its content, we would be glad to hear from you and please feel free to contact us by writing to the <pigeon-dev at ml.pigeonair.net> mailing list or by directly contacting one of the authors.

1.1. Intended Audience

This document is meant to explain DNS Administrators how to setup their own Bind DNS to balance connections in a PigeonAir Cluster. You should be worried about using the DNS as balancer if your PigeonAir cluster is made of more than one node and if your installation does not use any other kind of balancer.

It is strongly discouraged to use DNS balancing in any cluster with more than 3/4 nodes, since it is *always* better to use a dedicated appliance or system to balance connections.

1.2. Copyright Notice

This document was written by Carlo Contavalli <ccontavalli at masobit.net> and is thus Copyright (C) Carlo Contavalli 2003, 2004 and the PigeonAir Project.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts.

Any example of program code available in this document should be considered protected by the terms of the GNU General Public License.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by their respective owners.

2. DNS Records for standard mail services

This section will discuss DNS configuration needed to balance services like IMAP, POP3 and SMTP.

2.1. POP3/IMAP Protocol

Main purpose of the DNS records regarding the POP3/IMAP services is to allow users to use a symbolic name to connect to one of the POP3/IMAP servers.

Which POP3/IMAP node of the cluster the user will actually end up to connect to is not really important: using a shared storage cluster, each node of the cluster is able to access user data, while using a distributed storage the server receiving the connection will take care to forward it to correct node. If you don't know what I mean, please make sure to read %TODO%.

To load balance connections, it is thus sufficient to add multiple A records, pointing to all the POP3/IMAP Proxies/Servers, depending on the setup being used.

This needs to be done for all the domain handled by the cluster, either adding all A records for each domain or by adding a CNAME which points to the list of A records.

As an example, if we need to provide an IMAP/POP3 server with address "imap.mydomain.org" and we have a PigeonAir cluster made of 3 nodes having with ip address 1.2.3.4, 1.2.3.5 and 1.2.3.6, we need to add the following records in the "mydomain.org" zone:

```
imap.mydomain.org. A 1.2.3.4  
A 1.2.3.5
```

A 1.2.3.6

As said in the previous section, those records should be added to all the zones of every virtual domain handled by the cluster. If you don't like this, either use another balancer, use *CNAMEs*, or try to find a better solution.

2.2. SMTP Protocol

The SMTP protocol handles both the reception of new emails from remote servers and the sending of emails for dialup customers (relaying).

The two setups require two different sets of DNS records to be added, that will be discussed in the specific sections.

Note that there are many possible solutions, here we will discuss just one of them, probably not even the best one.

2.2.1. SMTP emails reception

As in the IMAP/POP3 case, for the PigeonAir cluster to work correctly it is enough for connections to get to anyone of the cluster nodes handling SMTP. The servers will take care internally either to deliver the mail to the disks, or to forward them to the correct node of the cluster.

One of the easiest solution to use is just to provide multiple equal-priority MX records in each zone being handled by the mail cluster.

As an example, in the case we have 3 cluster nodes named `node00.mail.org`, `node01.mail.org` and `node02.mail.org` that need to handle the emails directed to `mydomain.org`, we would could add the following records to the `mydomain.org` DNS zone:

```
MX 10 node00.mail.org.  
MX 10 node01.mail.org.  
MX 10 node02.mail.org.
```

In case your customers do not want your servers to appear in their zones, you can always insert the necessary A records in the `mydomain.org` zone, as shown below:

```
MX      10      node00.mydomain.org.  
      MX      10      node01.mydomain.org.  
      MX      10      node02.mydomain.org.  
  
node00.mydomain.org. A      1.2.3.4  
node01.mydomain.org. A      1.2.3.5  
node02.mydomain.org. A      1.2.3.6
```

This setup is a bit more inconvenient in the case we will ever need to change the IP address of the SMTP servers.

2.2.2. SMTP emails sending (relay)

As in all the above cases, it is enough for a given client to connect to any one of the PigeonAir cluster nodes to send a mail, provided it is authorized to do so.

Providers are usually required to provide the name of the SMTP server for the customers to use, for example smtp.mydomain.org.

In this case, you can simply add as many A records for smtp.mydomain.org as many nodes of the cluster you want your customers to use.

As an example, if you have 3 cluster nodes behaving as relay hosts for you dialup customers, with the ip addresses used in the previous examples, and you want your customers to use “smtp.mydomain.org” to send emails, you can add the following records in the “mydomain.org” DNS zone:

```
smtp.mydomain.org. A 1.2.3.4  
A 1.2.3.5  
A 1.2.3.6
```

You can either setup a single SMTP server record for outgoing emails, or you can add one for every one of your virtual domains.

2.2.3. Notes

In all the previous examples, all the nodes of the cluster where used for every virtual domain. By configuring properly the DNS, you can also choose to use certain cluster nodes for certain domains and thus reserve some resources for particular services.

2.3. DNS Records for web interfaces

Web interfaces are kind of trickier to setup correctly. The main problem here is that web interfaces need to track users, they need to store session in order to remember what they were doing and their own authentication data.

PigeonAir web interfaces all relay on PHP4, so you can use any PHP4 method for sharing sessions. In case you use one of the methods, you don't have to worry about DNS records anymore, you can simply add as many A records as nodes in the cluster handling web requests, allowing customers to connect to any of them on any request.

Those methods either relay on a shared file system, like NFS, keeping all sessions shared among server, or on an external database that takes care of sharing sessions.

Since we don't like NFS (introduces a SPF), and seems a waste to use a whole DB just to handle PHP sessions (depends on the environment, but in most cases it is), we describe here a simple method you can use to allow users to connect to any host and then bind them to one of the nodes until the session expires (or the user logs out).

As all the interfaces are very lightweight, the method has proved to be very reliable and at doing a very good average load balancing among servers.

If you still want to share a file system or use a shared db, please refer to %TODO%.

2.3.1. How it works

In the described setup, sessions are not shared among servers. This means that once a user logs in, it must be bound to a single server until it logs out.

The mechanism provided by all PigeonAir interfaces relies on a "magic name", which is the name being used by users to access the web interface.

This name will resolve to any of the nodes involved in the mail cluster. The node receiving the connection, after seeing the "magic name" has being used, will redirect the client to a mangled version of the mangled name, corresponding to the name of the server itself in the DNS.

As an example, users may access the webmail using "webmail.mydomain.org", which resolves to any of the cluster nodes. Once one of the nodes receives a connection for "webmail.mydomain.org", the magic name, the connection will be redirected to (for example) "webmail00.mydomain.org", the name of the node, thus binding the user to a given node until the end of the session.

A mechanism to detect users who decide to bookmark the mangled name and to redirect users anyway is being implemented right now in all the web interfaces, in order to keep the load "well balanced".

The mangling rules allow for virtual domains to make use of "magic names", exactly as shown above.

2.3.2. Setting up the DNS

Ok, the "magic" name is configurable from PigeonAdmin and PigeonReader configuration files. For the sake of simplicity, let's say it has been chosen to be "webmail", and let's say we have the same 3 cluster nodes as in the previous examples.

In order to correctly setup the zone for "mydomain.org", we would need to add something like:

```
; To balance the connection to the webmail
webmail.mydomain.org. A 1.2.3.4
A 1.2.3.5
A 1.2.3.6

; To allow the web mail to "bind"
; the connection to a particular host
```

```
webmail00.mydomain.org. A 1.2.3.4  
webmail01.mydomain.org. A 1.2.3.5  
webmail02.mydomain.org. A 1.2.3.6
```

If you want to, keep also in mind you are free:

- to force users to use just one of the cluster nodes, by simply using an A record with a different name than the magic one.
- to force the connections to follow on given cluster nodes, by properly changing the A records for the mangled names, or by changing the configuration of the PigeonReader or PigeonAdmin web interface.