Multihome / Virtualhost support

Multihome / Virtualhost support is a way in which a single machine can present two or more views of its documents based on the IP address that is accessed.

- What is Multihome / Virtualhost support?[1]
- Configuring NCSA HTTPd for Virtualhost Support[2]
- Configuring Solaris for Virtualhost Support[3]
- The VirtualHost Library[4]

What is Multihome / Virtualhost support?

Every machine on the internet has an IP address. Most of them have at least one name which maps to that IP address via DNS. Some even have more than one name. Usually, this allows a provider to change which machine was providing a service by changing what this CNAME points to. Common CNAMES were mailhost, news, and gopher, until this little thing called the WWW came along. Now, www is probably one of the most common CNAMES. There's only one problem: CNAMES and DNS names in general, are meant to be transparent as far as the machine is concerned. So, although a machine could be both www.foo.org and www.bar.org, the machine has no way of knowing which name it was called by, since they map to the same IP address.

But, machines are also able to have more than one IP address. This was originally for gateways, machines which sit on two different subnets, and have a different IP address on each. Most machines now adays can have a different IP address per interface, and most machines have two IP addresses, 127.0.0.1 on the loopback interface, and another on the ethernet interface. Some platforms allow you to have more than one address on a single interface. They usually do this by providing Virtual Interfaces[5].

Multihome / Virtualhost / Virtual Interface support is the ability for NCSA HTTPd to recognize which interface (and thereby which IP address) the server was called from. You can then have www.foo.org and www.bar.org map to 2 different IP addresses, and the server can tell which one it is responding as.

Basically, the usual reason for this functionality is so that a company can provide a Web Presence (*Be* On the Web[6]) without most of the expense, by hiding behind a different Internet Service Provider. With it (as shown by the included configuration files) you can also provide a different service to users on the local machine (via the name localhost).

Configuring NCSA HTTPd for Virtualhost Support

In the server configuration file (httpd.conf), after specifying the normal parameters

[1] #What
[2] #Config
[3] #Solaris
[4] #Library
[5] http://www.apache.org/docs/vif.info
[6] http://sdg.ncsa.uiuc.edu/~mag/work/specs/RFCs/rfc1775.txt

for your server, you start VirtualHost[7] section using the HTML–like syntax similar to the access configuration file (access.conf).
The syntax is:

```
<VirtualHost name>
Directive
Directive
</VirtualHost>
```

Where *name* is an IP address or hostname which corresponds to a single IP address. The directives that you can use are just about any that come from the other part of the server configuration file, including logfile names, the resource configuration file name, ServerAdmin, ServerName, and AnnotationServer. An example might be:

```
<VirtualHost 127.0.0.1>
ServerName localhost.foo.bar.com
ResourceConfig conf/localhost_srm.conf
TransferLog logs/localhost_access_log
</VirtualHost>
```

With this example, any accesses to the server by the name localhost will have the configuration options given override the other options. The server will parse the conf/localhost_srm.conf file and set its defaults accordingly if the server is called by the name localhost. All access to localhost will be logged in logs/localhost_access_log. All errors, referer, and user agent information will be logged to the default log files.

The resource configuration files for the virtual host are virtually identical to the regular configuration file, with some exceptions. Most of the fancy index'ing directives will work, but have unpredictable results. DocumentRoot, DirectoryIndex, AccessName, UserDir, DefaultType and DefaultIcon will override the default in the regular resource configuration file (srm.conf). In addition, ScriptAlias, Alias, Redirect, RedirectTemp and RedirectPermanent will only apply if you have them in the virtual host resource configuration file.

It should be able to handle any number of VirtualHosts up to the limits of the kernel and machine memory. Please place the virtual information after all the other information. If you don't, it might work, but the results can be unpredictably. Also be aware that most operating systems have a rather low number of open file descriptors allowed per process.

Configuring Solaris for Virtualhost Support

From Solaris 2 FAQ[8]

4.11) How can I have multiple addresses per interface?

Solaris 2.x provides an undocumented feature in ifconfig that allows having more than one IP address per interfaces.

Syntax:
```
ifconfig IF:N ip-address up
```

where "IF" is an interface (e.g., le0) and N is a number between 1 and 255. Removing

[7] ../setup/httpd/VirtualHost.html
[8] ftp://ftp.fwi.uva.nl/pub/solaris/solaris2.html

the pseudo interface and associated address is done with "`ifconfig IF:N 0.0.0.0 down`".

The Library

- RFC 1775 – Defining Being on the Web[9]
- Virtual Interface Information[10]
- Linux Virtual Howto[11]
- Solaris 2 FAQ[12]

Return to the tutorial index[13]

NCSA HTTPd Development Team / httpd@ncsa.uiuc.edu[14] / 10−20−95

9 http://sdg.ncsa.uiuc.edu/~mag/work/specs/RFCs/rfc1775.txt
10 http://www.apache.org/docs/vif.info
11 http://www.qosina.com/apache/virtual.html
12 ftp://ftp.fwi.uva.nl/pub/solaris/solaris2.html
13 index.html
14 ../FeedBack.html