

Load-Sharing with iptables and Linux-HA

Michael Schwartzkopff*, MultiNET Services GmbH, Munich, Germany

June 24, 2009

Copyright: GFDL, invariant section: Author.

Version: 0.20

Basics

In this article I will describe how to configure a load-sharing cluster with Linux and `iptables`. If a member of the cluster sees traffic it will calculate a hash from some characteristics from the IP traffic (like source address, source port, or ...) and autonomously decide if it is responsible for that traffic. For more about that mechanism see `man iptables`.

Linux-HA is used to create the fail-over between the nodes of the cluster. The cluster IP address is a resource and the resource agent controls the responsibilities for specific hash values on a specific node. Feel free to read the `IPaddr2` script to get the idea of the load-sharing. Please also mail me (misch@multinet.de) if you encounter problems with the resource agent. Thanks.

Prerequisites

First of all, you have to be sure, if the `CLUSTERIP` target of `iptables` is compiled into your distribution. Nearly any recent distributions work. Just look if the file

`/lib/modules/<kernel-no>/kernel/net/ipv4/netfilter/ipt_CLUSTERIP.ko` exists. If not, bad. Use an other distro or compile your kernel.

Get the Linux-HA software from the project website www.clusterlabs.org. In version 2.99 of the cluster-software `heartbeat` the load-sharing is included into the `IPaddr2` resource agent, so need the download my script any more.

To forward the traffic to all nodes in the clusters the `CLUSTERIP` target uses a multicast MAC addresses. Some routers, esp. Cisco, do not learn these MAC addresses. You have to configure it by hand in the static ARP table.

For tests and demonstration I use Xen based virtual machines. You only need one hardware to simulate everything. Of course having only one hardware does not make sense in a production environment.

*misch@multinet.de

Compiling, Installation

Just do an ordinary installation on Linux-HA as described in Learning HB <http://www.clusterlabs.org/wiki/Install>. I used a complete new installation of version 1.0.4 of `pacemaker` and compiled the debian packages from the scratch (lenny repository: www.multinet.de/debian_experimental_main).

You can choose either the heartbeat or the OpenAIS clusterstack. Be sure to configure it correctly and start it.

Please use at least version 1.0.4 of `pacemaker` for two reasons. There are a awful lot of bugs in previous versions beeing fixed the latest version and you will hardly find anybody willing to support you installation, if you have a pre-2.99 `heartbeat` clusterstack.

Configuration with the CRM subshell

In this version of the document I will show the configuration of the clustered IP addresses with the CRM subshell command since it is so easy to use. The configuration of the cloned IP address in CRM notation is:

```
primitive resIP ocf:heartbeat:IPaddr2 \  
  operations $id="resIP-operations" \  
  op monitor interval="10s" timeout="20s" start-delay="0" \  
  params ip="1.2.3.4" nic="eth0" cidr_netmask="24" \  
  clusterip_hash="sourceip-sourceport" \  
  meta resource-stickiness="0" \  
clone cloneIP resIP \  
  meta clone-max="2" globally-unique="true" clone-node-max="2"
```

Some remarks about the specific finetuning of the resource.

- Please enter the IP address and nic you need insted of this samples.
- Please take care of the specific `resource-stickiness="0"` of the primitive resource that overrides all default stickiness of the cluster. Otherwise the resource would not fail back to the other node if it becomes available after a problem. Instead all instances of the clone would stay on the same node which a kind of contradicts or load-sharing.
- Please also note the `clone-max="2"` and `clone-node-max="2"`. This means that this resource appears twice in the custer and if can run TWO times on one node. Normally one instance of the resource would run on each node of my two node cluster, but in case of a failure the instance of the failed node has to be taken over by the second node. Of course on a cluster with more nodes you can adjust these values to your needs.
- The `globally-unique` attribute has to be set to true, so the cluster can distinguish between both instances of the resource. This definitely is the case since every instance is responsible for one hash value. Otherwise the cluster would refuse to start two instances of the resource on one node.

Have fun

You can check cluster now by pinging the cluster IP address. In my case it answers on ping 1.2.3.4 if your routing is correct.

It is also possible to log in to the cluster with SSH. But you never know, what node will answer. Sometimes it is the first node, sometimes it is the second node. So if you want to log in to one specific node, always use the dedicated IP address, not the cluster address.

For a test of the cluster you can switch one of the nodes to “standby” with a simple right-click on the node while pinging the cluster. In my case I loose perhaps 1 ping, sometimes I get 1 duplicate. But within 2 seconds the other node took over.

Now you can use the cluster as a base for the installation of all (!) other services like Web servers, Databases, ... Of course the synchronization of the data between the applications is not part of Linux-HA. This has to be controlled inside the application. But where no data synchronization is needed this is a very simple way to build a load-sharing cluster to handle large amounts of traffic.

This concept is also described on the web site <http://www.linux-ha.org/ClusterIP>.

If you want to learn more about Linux high availability and clusters: I wrote a book “Clusterbau mit Linux-HA Version 2”. It was published by O’Reilly. At the moment it is only available in German, but when you ask O’Reilly, perhaps they will translate it.

Commercial support on Linux clusters is also available. Just mail me: misch@schwartzkopff.org

Have fun clustering!

Michael Schwartzkopff