

A Distributed Virtual Host Management System for VMware GSX.

Edward Balas

ebalas@iu.edu

Advanced Network Management Lab

<http://www.anml.iu.edu/>

4/22/02

Overview:

Based on our need to run a large number of virtual hosts and our inherent laziness, we have developed a Distributed Virtual Host Management System(DVHMS) for use with VMware GSX (LINUX ONLY). With this system we are able to effectively manage a large number virtual machines that are spread across many Servers. The system is based on VMware's Perl API and standard technologies such as XML and rdist we are able to provide central configuration, execution control and monitoring from a lightweight command-line interface.

Introduction:

The DVMS is tailored to environments where there are a multiple identical copies of a number of master host images, such as a networking laboratory. It provides central configuration and execution control across multiple servers with minimal effort the administrator. Focus was placed keeping the system lightweight and provided the greatest amount of automation.

Virtual Hosts are broken into 2 types: Masters and Run Copies. Masters are never actually run on the network but are used to generate the Run Copies. Run Copies are copies of the Masters, with things like MAC address being modified, and are the virtual hosts to be run on the network. This way we can ensure that Masters are not corrupted in the execution process.

There are two basic types of servers that need to exist in this system: Management and Host Servers. The Management Server(MS) operates as a repository for configuration file and Masters. It also acts as a DHCP server. The Host Servers(HS) execute the Run Copies, it main feature is the ability to run VMware GSX.

The configuration file stores all data related to the system: this includes a definition of the available Masters, the available HSs and the Run Copies each should run, and the IP and MAC address that each Run Copy should use.

To change the network-wide profile for all the virtual hosts, an administrator only needs to edit the master configuration and execute one command. The systems is highly automated.

There are 3 basic tasks associated with using the DVHMS: Master Image generation, configuration and execution control.

Master Image generation:

Generating Masters is fairly straight forward, and is the only manual task required for this system.

1. Create a virtual machine as normal with GSX on one of the HS.
2. Once the Operating System has been installed and the system is fully operational, suspend the host.
3. Rename the virtual host's configuration file to *vhms.cfg*.
4. Next, all paths virtual hosts configuration file need to be changed from absolute paths to relative paths.
5. After this, the directory should be tarred, but not compressed and located on Master Server wherever desired.

Configuration:

The configuration process involves editing an XML document.

The basic configuration Elements are:

IMG:

Defines a Master Image

SRV:

Defines a HS

VHI:

Defines the list of Run Copies to be made for a given master on a specific HS.

VHD:

Defines the actual Run Copy Instance.

Example config with only 1 HS and 5 Run Copies, 2 of 1 Master and 3 of another:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE NIFTICONF SYSTEM "masterconf.dtd" [
<!ENTITY br "
">
]>
<VHMS DHCP="10.0.0.251">
  <IMGLIST>
    <IMG ID="rh6.2">
      <DESCR>A RedHat 6.2 image</DESCR>
      <LOC URL="file://var/vhms/mir/redhat62_master.tar"></LOC>
    </IMG>
    <IMG ID="man8.1">
      <DESCR>Mandrake8.1 image</DESCR>
      <LOC URL="file://var/vhms/mir/mandrake81_master.tar"></LOC>
    </IMG>
  </IMGLIST>
  <SRVLIST>
    <SRV ID="hs1" IP="10.0.0.1" MIR="/var/vhms/mir" RUN="/var/vhms/run">
      <VHI ID="rh6.2">
        <VHD RID="one" MAC="00:50:56:00:00:01" IP="10.0.0.1"></VHD>
        <VHD RID="two" MAC="00:50:56:00:00:02" IP="10.0.0.2"></VHD>
      </VHI>
      <VHI ID="man8.1">
        <VHD RID="three" MAC="00:50:56:00:00:03" IP="10.0.0.3"></VHD>
        <VHD RID="four" MAC="00:50:56:00:00:04" IP="10.0.0.4"></VHD>
        <VHD RID="five" MAC="00:50:56:00:00:05" IP="10.0.0.5"></VHD>
      </VHI>
    </SRV>
  </SRVLIST>
</VHMS>

```

Once the Configuration file has been updated, the user simply needs to run *mconfig* to update the entire system. When this application runs, it will dist new Masters to where they are needed and then update each HS by calling a script called *lconfig*. As each Run Copy contains a full file system, the slowest part of the process is generating these Run Copies as it involves a significant amount of disk activity.

Execution Control:

Execution in normal operation is handled transparently through *mconfig*, when a configuration change is made *mconfig* automatically stops old vhosts and starts new vhosts. However situations do arise where manual intervention and monitoring are needed. For this there is an application called *vmctl*. *Vmctl* allows users to monitor all hosts on the network, and control their execution. With one command a user can stop or suspend all virtual hosts in the system.

Example *vmctl* status output:

```
[foobar@hs1 bin]$ vmctl status
```

(V)Host			State	Uptime (hrs)	CPU%	MEM
hs1:			on	123.98	1.50	2137
one	156.56.105.1	rh6.2	on	0.09	1.40	129
two	156.56.105.2	rh6.2	on	0.09	0.00	129
three	156.56.105.3	man8.1	on	0.09	0.00	132
four	156.56.105.4	man8.1	on	0.09	0.00	132
five	156.56.105.5	man8.1	on	0.09	0.00	132

Security Considerations:

The VMware API does not do transport encryption, thus it is advisable to either use IPSec to secure all traffic between the MS and the HS or to use a private network that is not routed.

For More information:

Online Documentation for VMware GSX:

http://www.vmware.com/support/gsx/doc/index_gsx_linux.html

Online Documentation for VMware GSX perl API:

http://www.vmware.com/support/gsx/doc/perl_api_gsx_linux.html