GPFS Multi-Cluster Routing HOWTO *v1*

A Visual HOWTO

Prepared by:
Brian Finley, IBM

GPFS Multi-Cluster

Well, I'm making the presumption that you already know what multi-cluster is, and that you're just trying to assess how to make it work in a certain scenario.

This visual HOWTO is intended to help you to do just that.



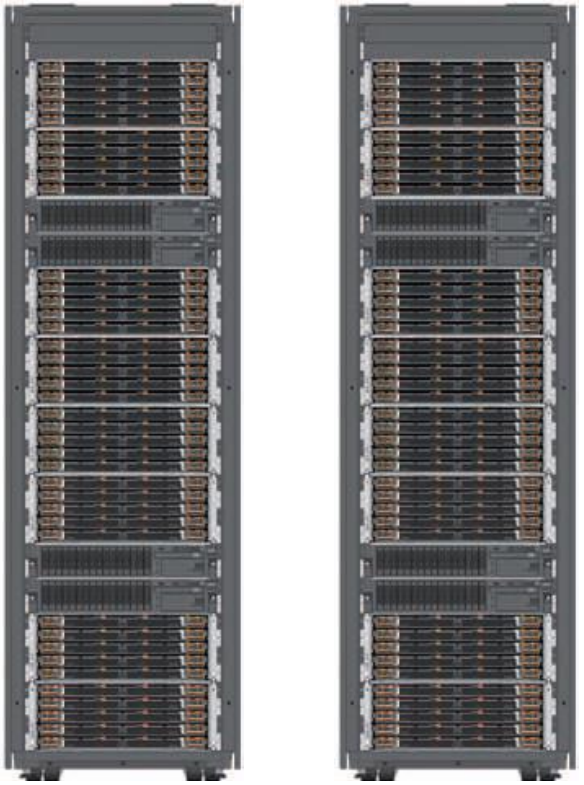
Fundamentals for Multi-Cluster

Required

- Two (or more) GPFS clusters
- Within each pair of connected clusters, you must have consistent name resolution for the GPFS daemon-interfaces for all nodes.
- Within each pair of connected clusters, all nodes must be able to route to all other nodes' GPFS daemon-interfaces via IP.
- Each GPFS cluster must maintain it's own quorum.

Optional

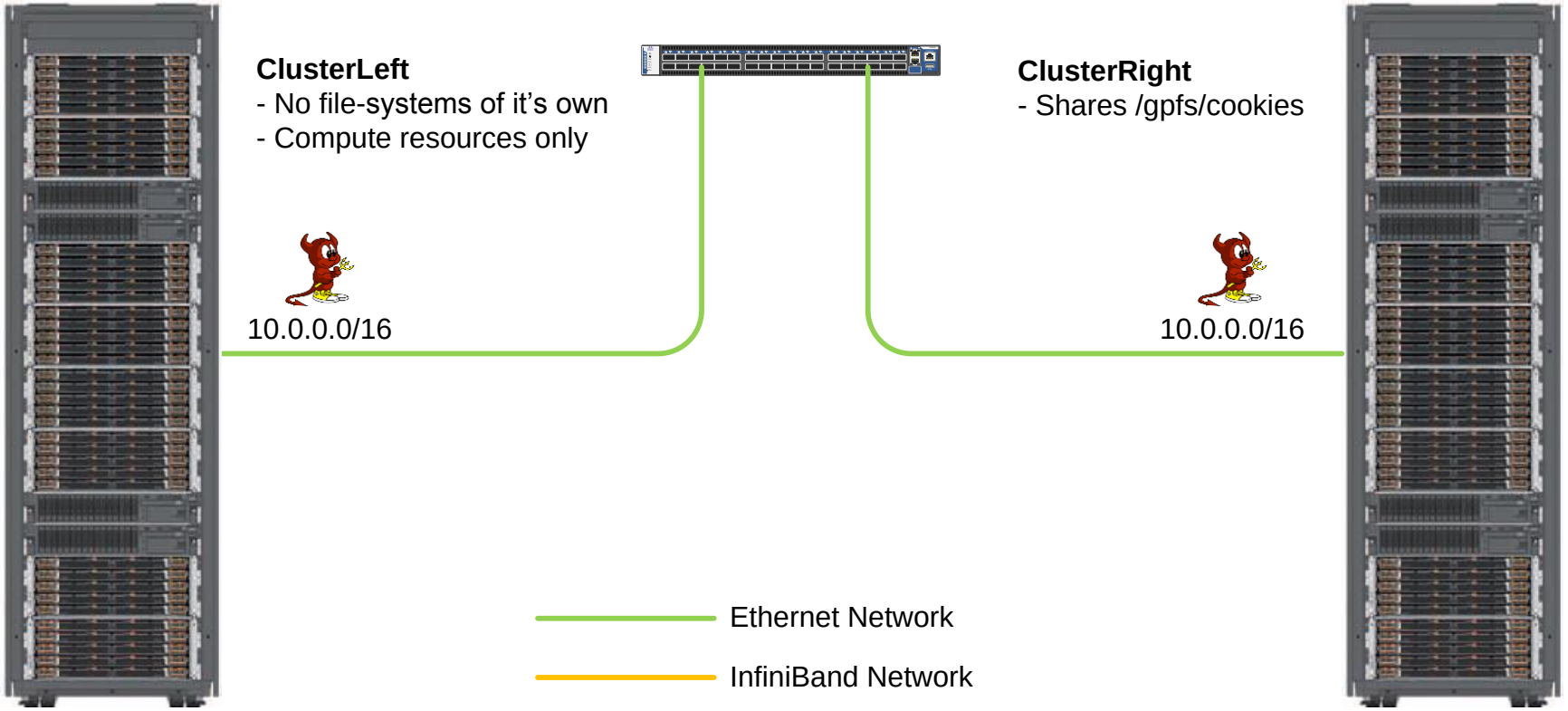
- Pairs of GPFS clusters may be on the same InfiniBand fabric, in which case RDMA can be used for *data* communications, even from one cluster to the other.
- It's OK if one or more of the GPFS clusters is a client only cluster, with no storage of it's own.
- IP traffic for either administrative or data communication between clusters may be passed over *any* physical layer technology, although faster is certainly better.



A Very Simple Topology

Two GPFS clusters with One IP Network

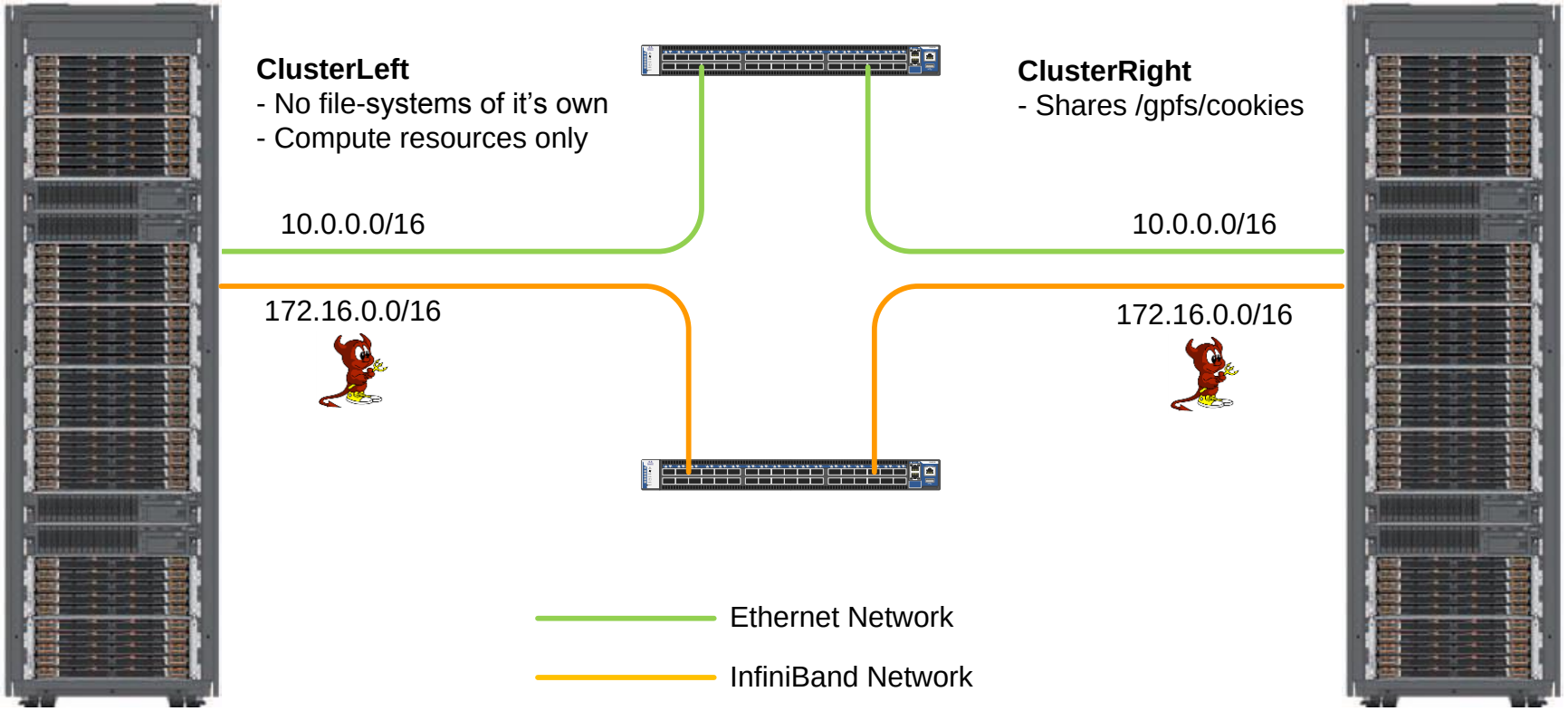
- Both clusters have their main OS and GPFS daemon-interfaces on the 10.0.0.0/16 network.
- Both clusters use IP for administrative *and* data traffic.
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



A Slightly Less Simple Topology

Two GPFS clusters with **Two** IP Networks (one for GPFS)

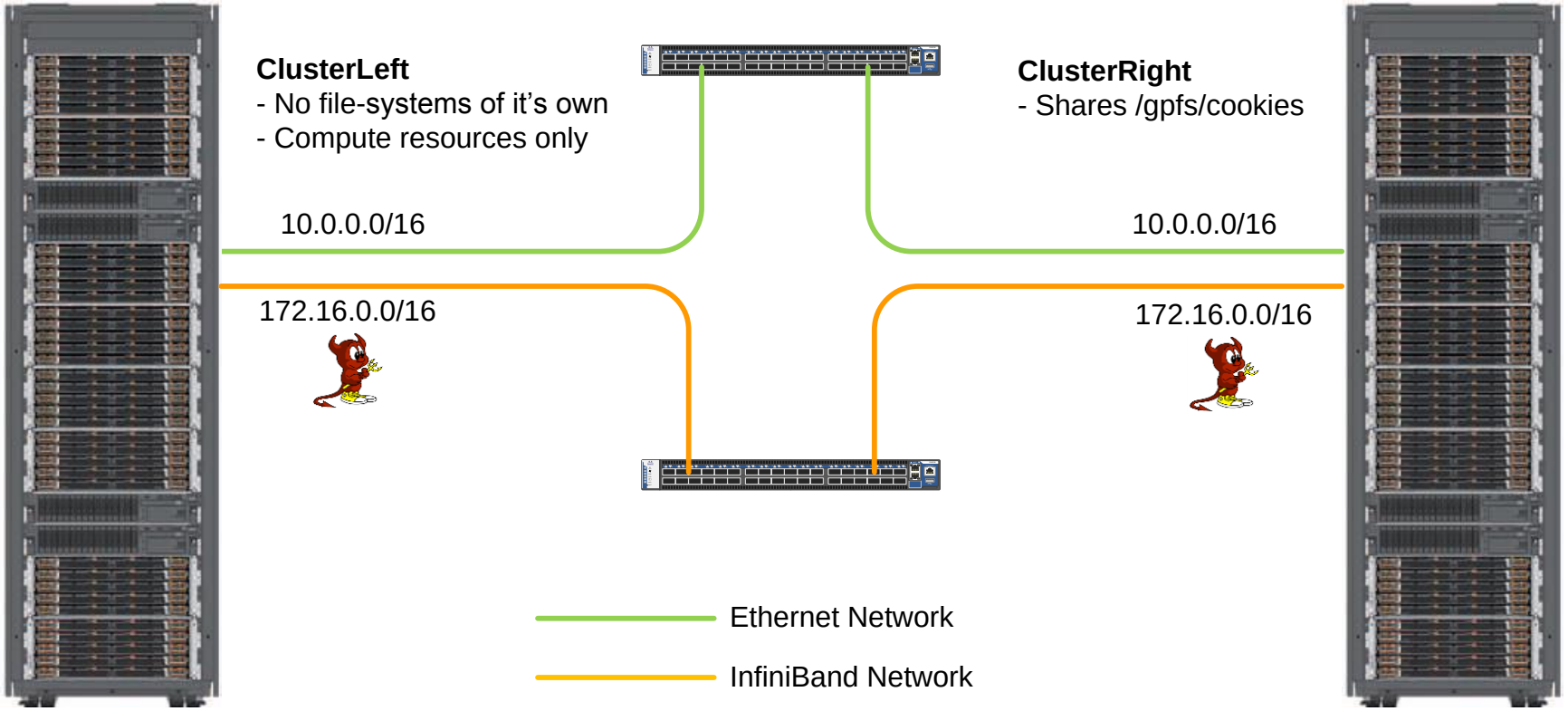
- Both clusters use 10.0.0.0/16 as their main OS network.
- **Both clusters have their GPFS daemon-interfaces on the 172.16.0.0/16 network.**
- **Both clusters use IP over 172.16.0.0/16 for administrative and data traffic.**
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



Still Simple with Better Performance

Two GPFS clusters with Two IP Networks (one for GPFS)

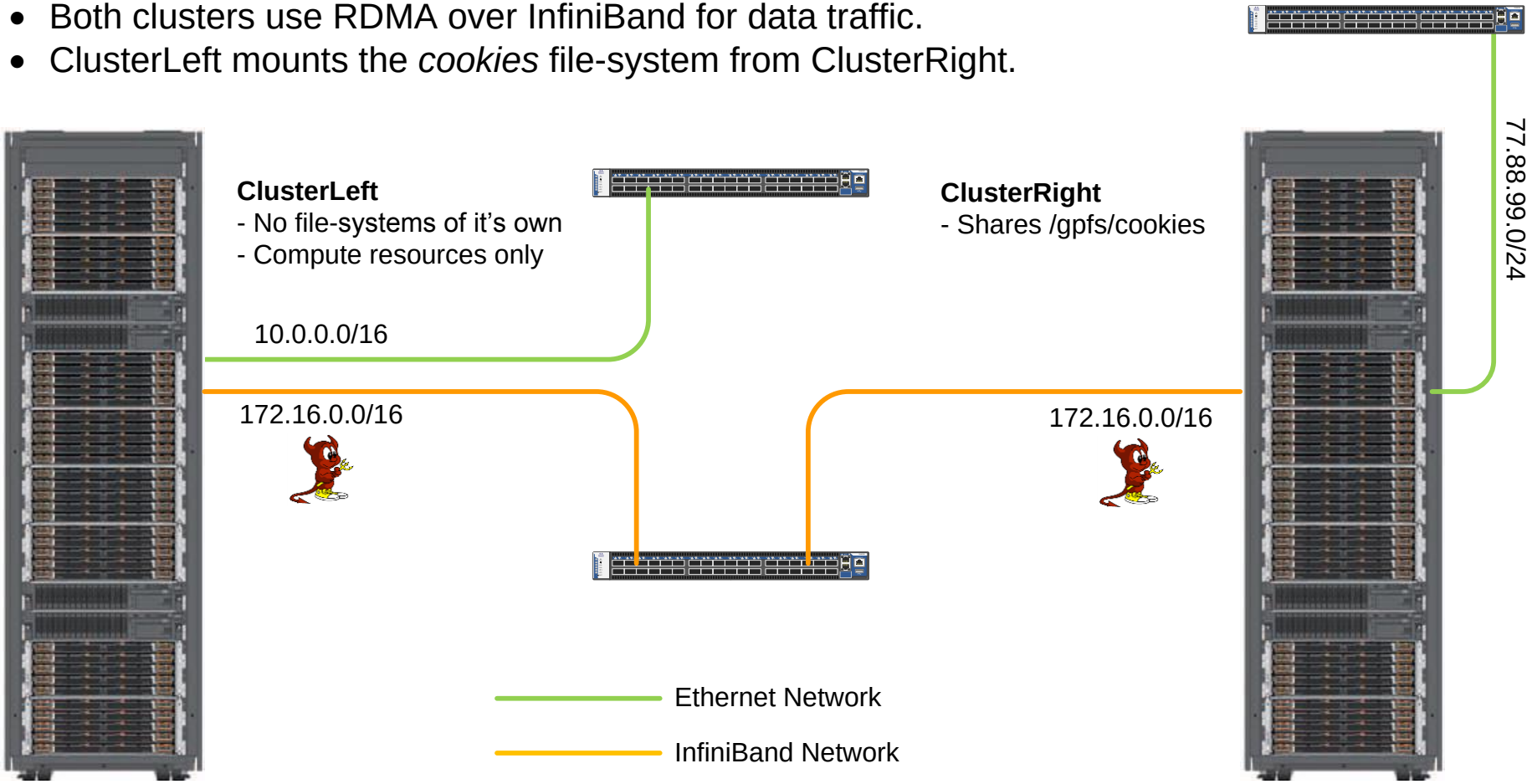
- Both clusters use 10.0.0.0/16 as their main OS network.
- Both clusters have their GPFS daemon-interfaces on the 172.16.0.0/16 network.
- Both clusters use IP over 172.16.0.0/16 for administrative traffic.
- **Both clusters use RDMA over InfiniBand for data traffic.**
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



Still Simple but Three Nets

Two GPFS clusters with **Three** IP Networks (one for GPFS)

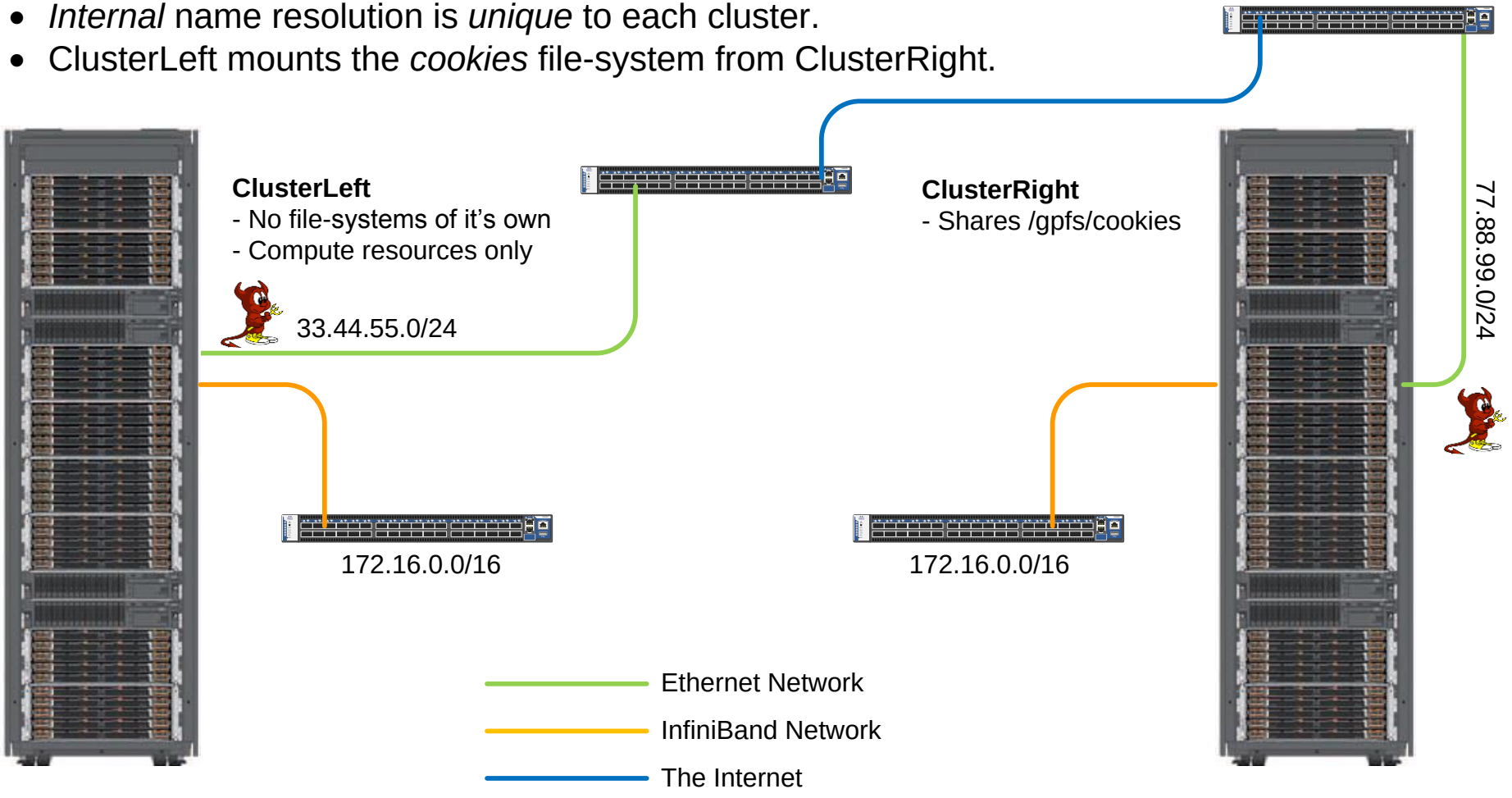
- **Still simple as both clusters have their daemon-interfaces on the 172.16.0.0/16 net.**
- **ClusterLeft uses 10.0.0.0/16 as it's main OS network.**
- **ClusterRight uses 77.88.99.0/24 as it's main OS network.**
- Both clusters use IP over 172.16.0.0/16 for administrative traffic.
- Both clusters use RDMA over InfiniBand for data traffic.
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



Still Fairly Simple but Appears Complex

Two GPFS clusters with **Four** IP Networks (**two** for GPFS)

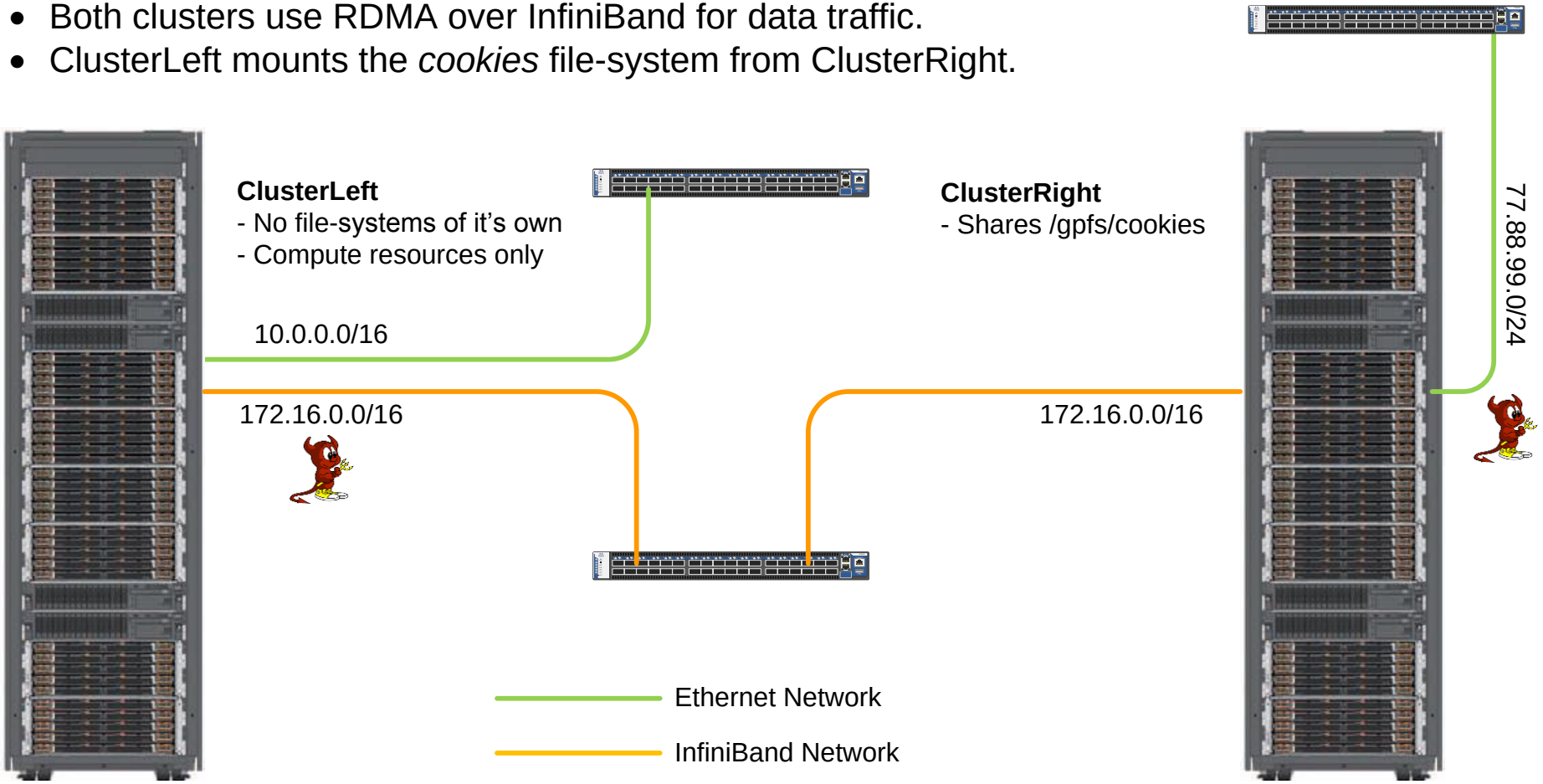
- Each cluster has it's own internal net, which both happen to use 172.16.0.0/16. This is OK because they are non-routed nets and thus do not present a conflict.
- The daemon-interfaces for each cluster are on their external nets (**green**), which are routed.
- *External* name resolution is *common and consistent* across both clusters.
- *Internal* name resolution is *unique* to each cluster.
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



Now it's Gettin' Funky *(page 1)*

Two GPFS clusters with **Three** IP Networks (**two** for GPFS)

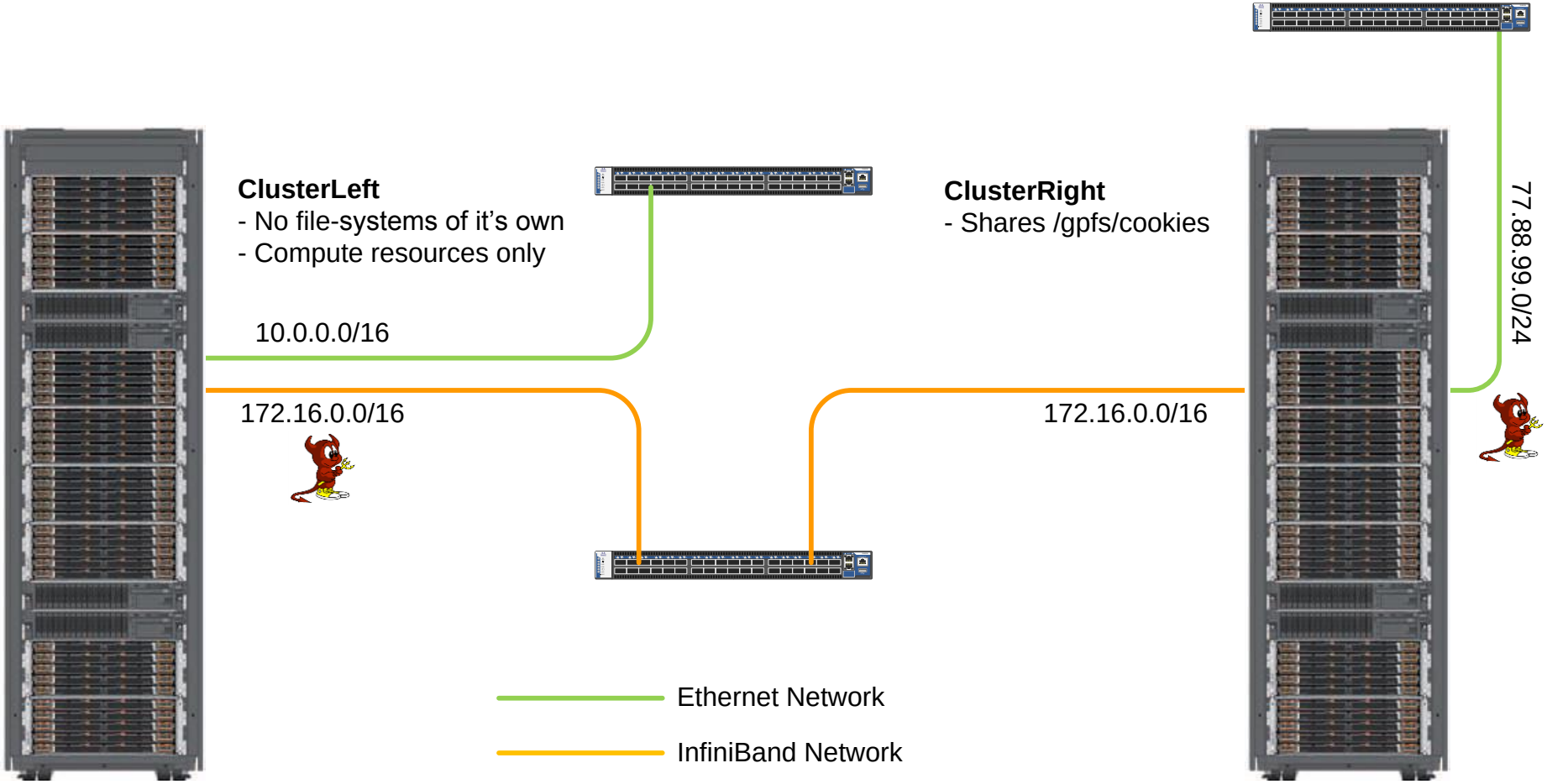
- ClusterLeft has it's daemon-interface on the 172.16.0.0/16 network.
- ClusterRight has it's daemon-interface on the 45.67.89.0/24 network.
- ClusterLeft requires static route entries to reach ClusterRight's daemon-interfaces.
- ClusterRight can already access ClusterLeft's daemon-interfaces directly.
- Both clusters use RDMA over InfiniBand for data traffic.
- ClusterLeft mounts the *cookies* file-system from ClusterRight.



Now it's Gettin' Funky *(page 2)*

Before Routing Changes

- ClusterLeft has no default gateway.
- ClusterLeft can *not* ping the daemon-interfaces of ClusterRight (77.88.99.0/24)
- ClusterLeft can ping the *non*-daemon-interfaces of ClusterRight (172.16.0.0/16)
- ClusterRight can ping the the daemon-interfaces of ClusterLeft (172.16.0.0/16)



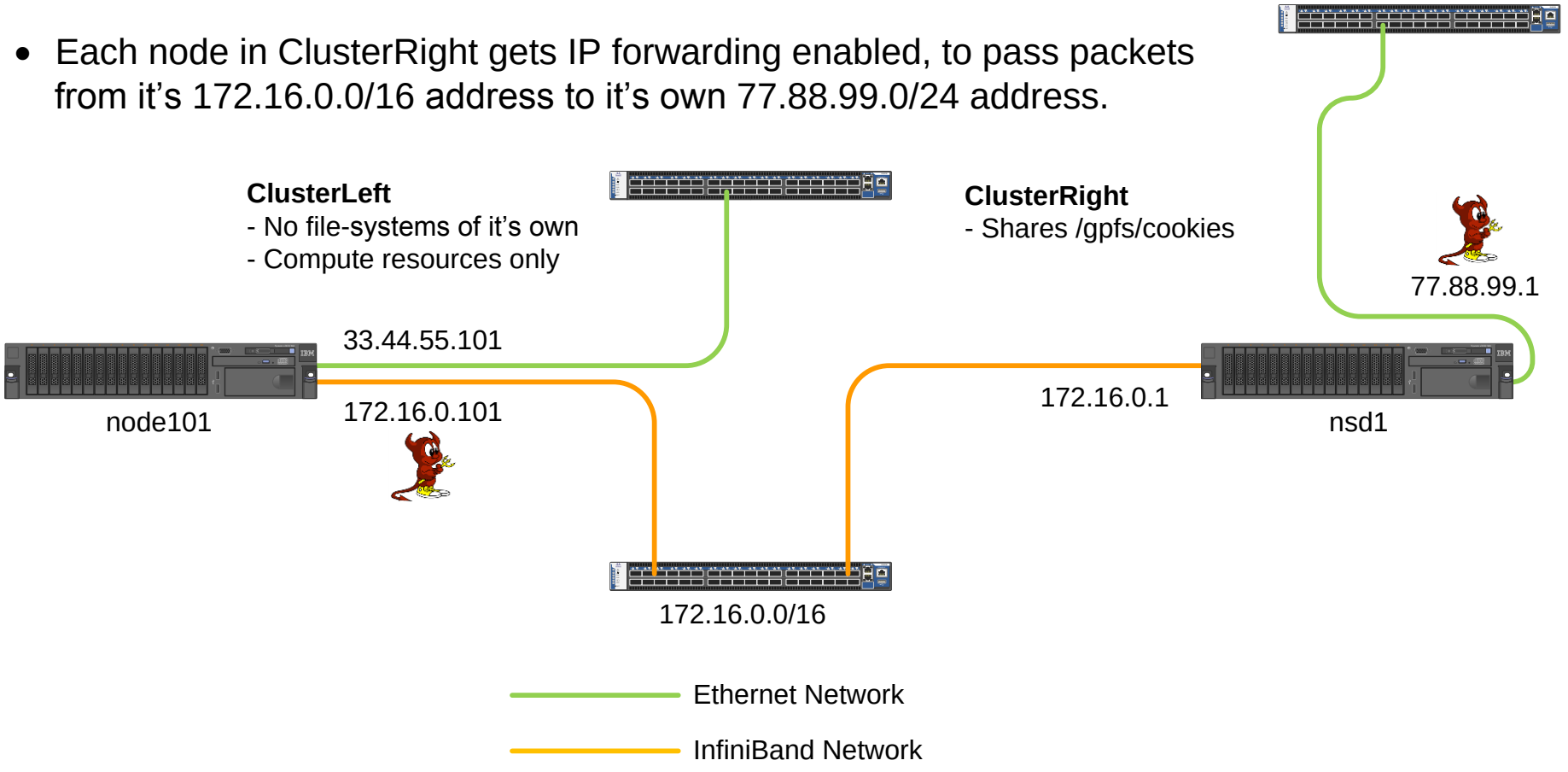
Now it's Gettin' Funky *(page 3)*

What are Those Routing Changes?

- Each node in ClusterLeft gets a shiny new static route entry for each node in ClusterRight, that makes the 172.16.0.0/16 address on each ClusterRight node the gateway to that same node's own 77.88.99.0/24 address.

```
ip route add 77.88.99.1 dev ib0 via 172.16.0.1
```

- Each node in ClusterRight gets IP forwarding enabled, to pass packets from it's 172.16.0.0/16 address to it's own 77.88.99.0/24 address.



Now it's Gettin' Funky *(page 4)*

Implementing the routing, step 1 of 2

```
#!/bin/sh
#
# Install this script as /sbin/ifup-local on all machines in ClusterLeft.
#
# It will be executed after each time an interface is brought up, such as on
# boot or via "ifup eth0" or "ifup ib0".  It will add static route entries on
# the node where it is run, allowing that ClusterLeft node to communicate with
# the daemon-interfaces on nodes in ClusterRight.  This example is appropriate
# for RHEL, CentOS, and friends.  Other distros have similar methods available.
#
# Test with "/sbin/ifup-local ib0" or other interface name.
#
# 2012.10.25 Brian Elliott Finley <bfinley@us.ibm.com>
# - Created
#

IFACE=$1
if [ -z $IFACE ]; then
    echo "Please try:"
    echo
    echo "  $0 NETWORK_INTERFACE"
    echo
    exit 1
fi

if [ "$IFACE" == "ib0" ]; then

    # nsd1.ClusterRight
    ip route add 77.88.99.1 dev ib0 via 172.16.0.1

    # nsd2.ClusterRight
    ip route add 77.88.99.2 dev ib0 via 172.16.0.2

    # nsd3.ClusterRight
    ip route add 77.88.99.3 dev ib0 via 172.16.0.3

fi
```

Now it's Gettin' Funky *(page 5)*

Implementing the routing, step 2 of 2

- On each of the nodes in ClusterRight, add this entry to /etc/sysctl.conf:

```
#
# Allow packets received on this NSD server's 172.16.0.0/24 interface to
# be routed to it's own 78.88.99.0/24 interface.
#
net.ipv4.ip.forward = 1
```

- This setting will take effect automatically on each boot. Let's also go ahead and tell it to take effect now:

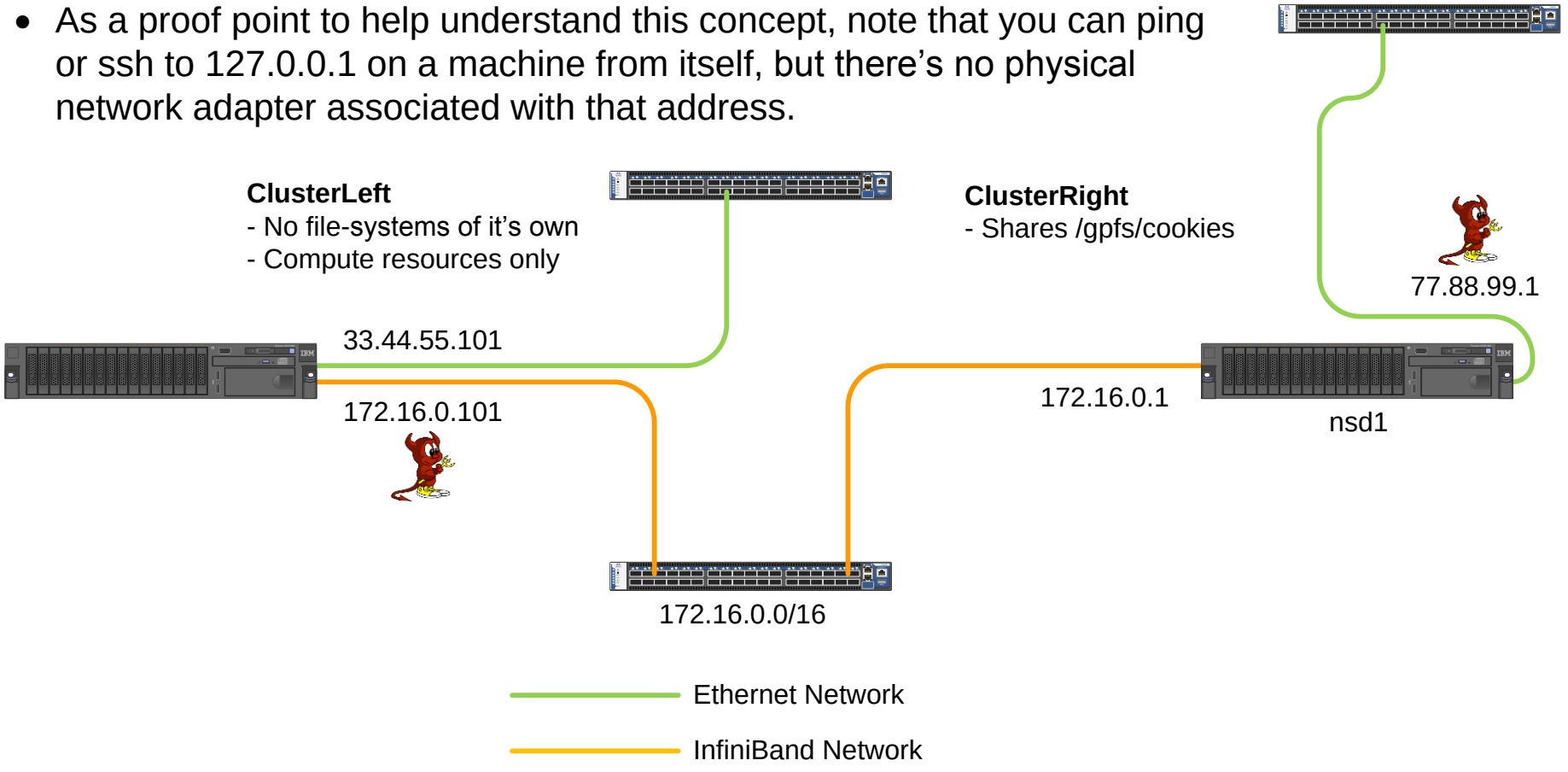
```
sysctl -p
```

- Don't we need to add static routes on the ClusterRight nodes? Nope -- the nodes in ClusterRight have an interface on the same network as the ClusterLeft nodes' daemon-interfaces and thus need no additional route entries.

Now it's Gettin' Funky *(page 6)*

After the Routing Changes *(Left to Right)*

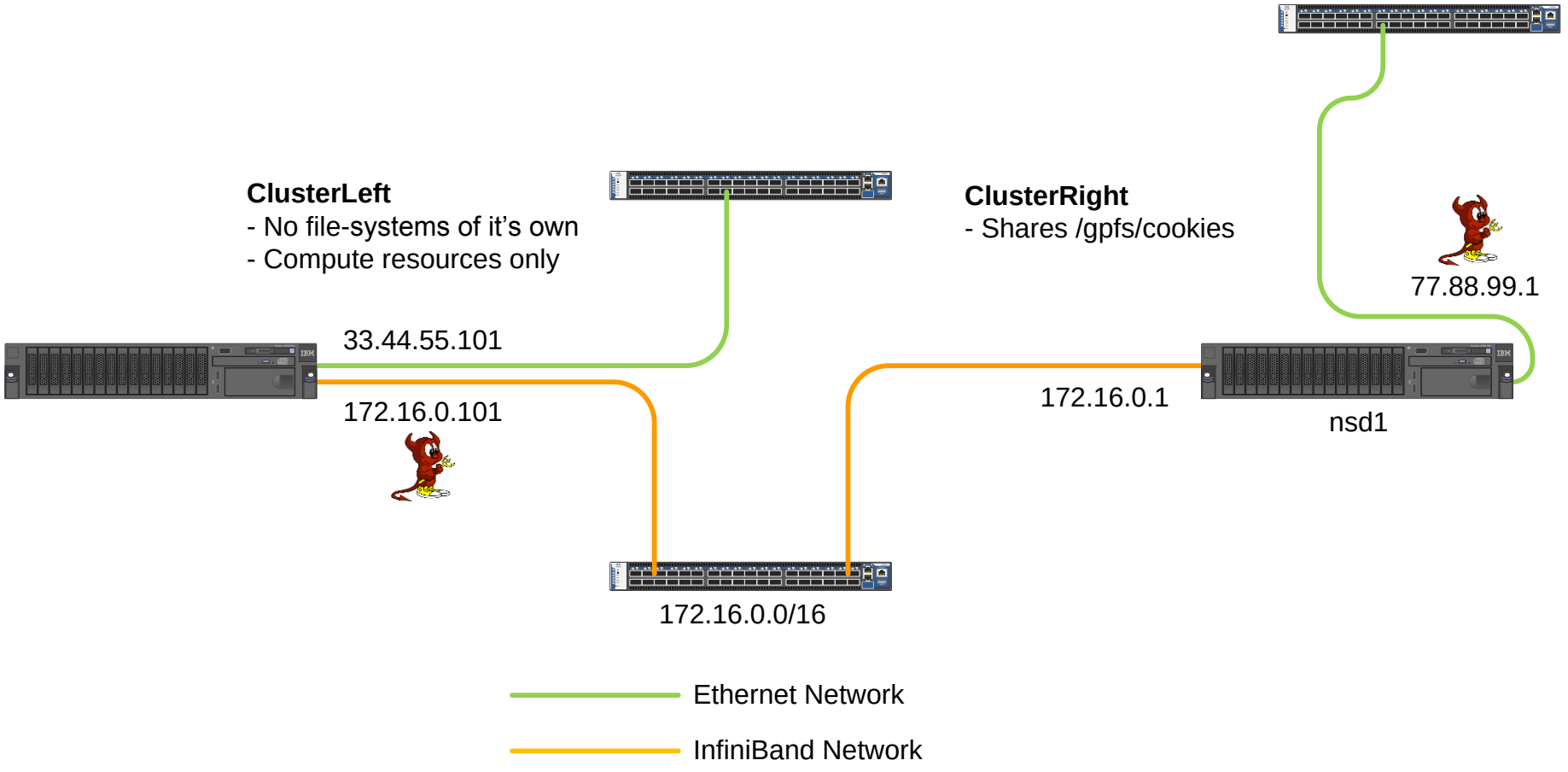
- When the GPFS daemon on node101 needs to talk to the GPFS daemon on nsd1, it sends the traffic to 172.16.0.1, where it gets routed up to 77.88.99.1.
- Because 172.16.0.1 and 77.88.99.1 are on the same IP stack on the same host (nsd1), this traffic never actually touches the physical adapter with the 77.88.99.1 address, and takes no performance hit, even if 77.88.99.1 is assigned to a slower technology, such as 1GbE.
- As a proof point to help understand this concept, note that you can ping or ssh to 127.0.0.1 on a machine from itself, but there's no physical network adapter associated with that address.



Now it's Gettin' Funky *(page 6)*

After the Routing Changes *(Right to Left)*

- When the GPFS daemon on nsd1 goes to respond to node101, no IP forwarding is involved.
- It simply hands the IP stack some data, and says, "Deliver to 172.16.0.101".
- The IP stack looks up the route to deliver to that address, and sees that it has an interface on that very network, so it just drops the packet on the wire and it goes straight to node101.



Conceptual Wrap-up

It’s all about the daemon-interfaces

- In order to have a properly functioning GPFS multi-cluster implementation, all daemon-interfaces need to speak to all daemon-interfaces for paired clusters.
- TCP/IP is very flexible and provides many ways of achieving connectivity between nodes, including the use of static routes.
- While it’s possible to specify a default gateway and “be done with it”, you could be passing traffic out across the Internet, or introducing a single point of failure (your gateway). And I hope this HOWTO makes it easy enough to understand what you need to know and do to even if you have to incorporate a wee bit of routing, so that you’ll steer away from the default gateway option.
- Oh, and of course, it’s important to know that you can still use RDMA for data transfers, even in a multi-cluster environment.

Happy computing!

-Brian Finley

For more information, contact:

Brian Finley
IBM, Executive IT Specialist
Scale Out Cloud and Technical Computing
Mobile: +1 469.667.2110
bfinley@us.ibm.com

Scott Denham
IBM, Consulting IT Specialist
IT Architect, Upstream Petroleum HPC
Phone: +1 713-940-1178
sdenham@us.ibm.com

Ray Paden
IBM, IT Architect - Cross Segment
STG, Software Defined Systems
Phone: +1 512-286-7055
raypaden@us.ibm.com