fineDoc Number: EDCS-<XXXXX>
Last Revision Date: December 17, 2010
Created by: Louis Watta
Template Ver. Number: EDCS-XXXX Rev X

CAE

Nexus 1000V 1.4
QOS Fair Queuing White Paper

TABLE OF CONTENTS

1 Introduction

The Cisco Nexus 1000V, is a Cisco developed server virtualization switching architecture for VMware ESX environments. The Nexus 1000V enables policy based virtual machine (VM) connectivity, mobility of security and network properties, and a non-disruptive operational model for both Server and Network administrators.

Offering a set of network features, management tools and diagnostic capabilities consistent with the customer's existing physical Cisco network infrastructure and enhanced for the virtual world, the Nexus 1000V allows customers to accelerate their adoption of VMs through the unification & simplification of the physical and virtual networks. The n1000v also secures & simplifies the deployment & movement of VM's to increase service velocity while maintaining and enforcing security policy.

1.1 White Paper

The purpose of this white paper is to walk the user through configuring, testing, and verifying QOS Weighted Fair Queuing

1.2 Assumptions

The assumptions of this white paper are that the reader has

- Installed VMware VC 4.1
- Installed Cisco Nexus 1000V 1.4 on an ESX VM in HA mode
- At least 1 ESX/ESXi 4.1 box with VEM module already loaded
- Created a Nexus 1000V Distributed Virtual Switch (DVS) under vCenter
- Added the ESX boxes to the Nexus 1000V DVS

1.3 Quality of Service Weighted Fair Queuing

QOS Weighted Fair Queuing is new in this version of Nexus 1000V. The goal of this feature is to address the following...

- Offer a higher vMotion performance, both for single vMotion and multiple concurrent vMotions, during congestion
- vMotion should not starve other high priority traffic (management, storage, or N1K control), considering the changes to vMotion performance in 4.1.
- Bandwidth guarantees for each traffic class must be respected

We provide pre-defined protocol matching for the following traffic types

- vMotion
- iSCSI
- NFS
- VMware FT
- VMware Service Console
- Cisco Nexus 1000V control, mgmt, packet

We provide support for 64 queues/resource pools per host, only supported on egress uplink ports. **QOS Weighted fair queuing only works with ESX 4.1 because it make use of the new Network I/O Control API VMware introduced in 4.1**

2 Configure the QOS policy

2.1 Configure Class-maps

Requirements:
ESX hosts and vCenter MUST be running 4.1.

We will create a policy for vMotion in this test. Thus we will need to assign the vMotion vmknic to a port-profile on the Nexus 1000V. Below is a sample port-profile for a vmknic. We are using vlan 10 to carry vMotion traffic.

```
port-profile type vethernet VMK
  vmware port-group
  switchport mode access
  switchport access vlan 10
  no shutdown
  system vlan 10
  state enabled
```

```
n1kv-bl# show int virtual

--------------------------------------------------------------------------------
Port        Adapter        Owner                     Mod Host
--------------------------------------------------------------------------------
Veth1       Net Adapter 1  vm1-2                     5   172.18.217.186
Veth2       Net Adapter 1  vm1-1                     5   172.18.217.186
Veth3       Net Adapter 1  windows_vm1               4   172.18.217.180
Veth4       Net Adapter 1  vm1-3                     5   172.18.217.186
Veth5       vmk0           VMware VMkernel           4   172.18.217.180
Veth6       vmk0           VMware VMkernel           5   172.18.217.186
Veth7       vmk1           VMware VMkernel           5   172.18.217.186
Veth8       vmk1           VMware VMkernel           4   172.18.217.180
Veth9       Net Adapter 1  vm1-4                     5   172.18.217.186
```

Above you can see we have two vmknics from hosts .180 and .186 assigned.

Create the QOS Policy:

The steps for creating a QOS queuing policy are as follows

- Create a class-map defining the type of traffic
- Create a policy-map that contains the class-maps you want to queue traffic for
- Assign the policy-map to an uplink port-profile

Create the class-map for vmotion

```
n1kv-bl(config)# class-map type queuing match-all vmotion-class
n1kv-bl(config-cmap-que)# match protocol ?
  n1k_control  N1K control traffic
  n1k_mgmt     N1K management traffic
```

```
  n1k_packet   N1K inband traffic
  vmw_ft       VmWare fault tolerance traffic
  vmw_iscsi    VmWare iSCSI traffic
  vmw_mgmt     VmWare management traffic
  vmw_nfs      VmWare NFS traffic
  vmw_vmotion  VmWare vmotion traffic

n1kv-bl(config-cmap-que)# match protocol vmw_vmotion
```

Above you can see all the predefined protocols that we can match for QOS. Since we are creating a vMotion policy we need to match on vmw_vmotion.

Next create a policy map containing the class-map from above

```
n1kv-bl(config)# policy-map type queuing vmotion-policy
n1kv-bl(config-pmap-que)# class type queuing vmotion-class
n1kv-bl(config-pmap-c-que)# bandwidth percent 50
```

This policy map will allow vMotion traffic to have 50 percent of the bandwidth. All other traffic will fall into the default class which will get the other 50 percent.

Now we have to assign the policy-map to the uplink port-profile. We already have an existing uplink port-profile called “uplink-vpc”

```
n1kv-bl(config)# port-profile uplink-vpc
n1kv-bl(config-port-prof)# service-policy type queuing output vmotion-policy

n1kv-bl# show port-profile name uplink-vpc

port-profile uplink-vpc
 type: Ethernet
 description:
 status: enabled
 max-ports: 32
 inherit:
 config attributes:
  switchport mode trunk
  service-policy type queuing output vmotion-policy
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  service-policy type queuing output vmotion-policy
  no shutdown
 assigned interfaces:
  Ethernet4/2
  Ethernet5/2
 port-group: uplink-vpc
 system vlans: 2,10
 capability l3control: no
 port-profile role: none
```

2.2 Verify the QOS policy-map is working

There are several command on the ESX server that can be run against the vem module of a host to get QOS statistics.

The vemcmd commands for qos are

- vemcmd show qos node
- vemcmd show qos policy
- vemcmd show qos pinst
- vemcmd show qos tables
- vemcmd show qos queue-stats
- vemcmd show qos queue-rate

Once a policy-map is applied to a port-profile these commands will return data

Run vemcmd show port to get the LTL of the vmnic of your uplink port.

```
[root@cae-esx-186 ~]# vemcmd show  port
 LTL   VSM Port  Admin Link  State  PC-LTL  SGID  Vem Port
  18     Eth5/2     UP   UP    FWD       0        vmnic1
  50      Veth7     UP   UP    FWD       0        vmk1
  51      Veth6     UP   UP    FWD       0        vmk0
  53      Veth2     UP   UP    FWD       0        vm1-1.eth0
```

Vmnic1 is our uplink nic and is LTL 18. Use this information for the next command.
Run "vemcmd show queue-stats 18" to get see the classes and pkts matched to the policy

```
[root@cae-esx-186 ~]# vemcmd show qos queue-rate 18
================================================================================
classid   pktsDrop      bytesDrop      pktsIn      bytesIn      pktsOut      bytesOut
           /sec         Kbytes/sec      /sec       Kbytes/sec     /sec
Kbytes/sec
================================================================================
   0           0           0           0           0           0           0
```

You can also run vemcmd show pinst to see a running total

```
[root@cae-esx-186 ~]# vemcmd show qos pinst

id       type
-------- --------
      18 Egress_q
         class        bytes matched         pkts matched
         -------- -------------------- --------------------
                0                  264                    4
```

Now kick off a bunch of vmotions and run "vemcmd show queue-stats 18" to see that the VEM is actively matching vmotion traffic for QOS.

```
[root@cae-esx-186 ~]# vemcmd show qos queue-stats 18
================================================================================
classid    pkts          bytes       pktsIn      bytesIn      pktsOut      bytesOut
          taildrop       taildrop
================================================================================
   0           0           0        39235   1349286506        39235   1349286506
```

Here is output from vemcmd show qos pinst as well.

```
[root@cae-esx-186 ~]# vemcmd show qos pinst

id       type
-------- --------
```

```
18 Egress_q
     class        bytes matched         pkts matched
   -------- -------------------- --------------------
         0           1349286506                39235
```

Here you can see we are actively matching packets for vmotion traffic for QOS.

Alternatively if you have a device that can measure network bandwidth you can use that to determine that the Nexus 1000V is allocating bandwidth correctly. You can use the ERSPAN function of the Nexus 1000V to SPAN the VEM traffic to a traffic collector.

2.3 Add another class-map to the policy

You are not limited to one class-map per policy-map. You can have multiple class-maps. This example will add an NFS class-map to already existing policy.

On the VSM create a new NFS class-map

```
n1kv-bl# config t
n1kv-bl(config)# class-map type queuing match-all nfs-class
n1kv-bl(config-cmap-que)# match protocol vmw_nfs
n1kv-bl(config-cmap-que)# show class-map


  Type queuing class-maps
  ========================


    class-map type queuing match-all nfs-class
      match protocol vmw_nfs

    class-map type queuing match-all vmotion-class
      match protocol vmw_vmotion
```

Now add the new class-map to the policy-map

```
n1kv-bl(config)# policy-map type queuing vmotion-policy
n1kv-bl(config-pmap-que)# class type queuing nfs-class
n1kv-bl(config-pmap-c-que)# bandwidth percent 40
n1kv-bl(config-pmap-c-que)# exit
n1kv-bl(config-pmap-que)# show policy-map

  Type qos policy-maps
  ====================

  Type queuing policy-maps
  ========================
  policy-map type queuing vmotion-policy
    class type queuing vmotion-class
      bandwidth percent 50
    class type queuing nfs-class
      bandwidth percent 40
```

On the ESX hosts verify that the VEM module now sees the nfs-class

```
n1kv-bl# module vem 4 execute vemcmd show qos policy
policyid classid policerid set_type value
-------- -------- -------- -------- --------
       0        0       -1       bw       50
                1       -1       bw       40
```

Here you can see that classid 0 is the vmotion class-map and classid 1 is the nfs classmap.
We can run vemcmd show qos pinst to see bytes matched since the policy was updated.

```
n1kv-bl# module vem 4 execute vemcmd show qos pinst

id       type
-------- --------
      18 Egress_q
            class        bytes matched         pkts matched
            -------- -------------------- --------------------
                   0                    0                    0

                   1              1871030                 1415
```

Here you can see that we have already matched some NFS packets since the policy was updated.

Again if you have device to measure bandwidth you can kick off several vmotions and some NFS traffic and verify that bandwidth correctly gets allocated.