FINE-TUNING SCALEIO PERFORMANCE

EMC® ScaleIO®

v.1.3x

Technical Note

P/N 302-001-168
REV 07

December 16, 2015

This document describes best practices for maximizing performance in high-performance (more than 60,000 IOPS) ScaleIO v1.3x environments. Topics include:

ScaleIO is a software-only solution that uses existing servers' local disks and LAN to create a virtual SAN that has all the benefits of external storage—but at a fraction of the cost and complexity. ScaleIO utilizes the existing local internal storage and turns it into internal shared block storage. For many workloads, ScaleIO storage is comparable to, or better than external shared block storage.

Revision history

The following table presents the revision history of this document:

Revision	Date	Description
01	June 09, 2014	First release of this document
02	September 15, 2014	Updates for ScaleIO v1.3x
03	December 1, 2014	Updates for ScaleIO v1.3x
04	January 26, 2105	Revised the Windows MTU procedure
05	April 30, 2015	Revised to expand scope to all high-performance environments
06	October, 2015	Removed recommendation to change the parameter *txqueuelen* to 10000, added jumbo frame recommendations, added disable ACK instructions
07	December, 2015	Added workaround procedures for scenario where large block I/O performance is lower than expected.

Tuning ScaleIO performance

You can improve ScaleIO system performance in terms of IOPS, latency, and bandwidth, by making environment-specific fine-tunings on the operating system, network, and ScaleIO components. This document describes these performance-related best practices.

NOTICE

To avoid the need to restart the system after making these configuration changes, it is strongly recommended to perform the changes before running any applications, or, minimally, before running applications with production data.

Performance tuning is very case-specific; to prevent undesirable effects, it is highly recommended to thoroughly test all changes. For further assistance, contact EMC Support.

We can separate performance tuning into during installation and post-installation activities.

Performance tuning during installation

This section describes steps to take during installation of ScaleIO to enhance performance.

The following table describes how to enhance initial performance optimization during installation. For more information, see the installation instructions in the documentation.

Installation method	Do
Installation Manager	In the CSV file, set **Optimize IOPs** = **Yes**
VMware deployment wizard	In the Add SDSs screen, select **Optimize for Flash** for each SDS.
Manual installation	Use the **CONF=IOPS** flag when installing the SDS

Performance tuning post-installation

This section describes steps to take after installation to enhance performance, in the following areas:

- "Tuning SDS nodes" on page 3
- "Tuning SDC nodes" on page 10
- "Additional network adjustments" on page 15

Tuning SDS nodes

You can achieve optimum performance by making the following adjustments on every SDS node. The first section is applicable to SDS nodes in all operating systems.

For all ScaleIO systems

Allocate more network memory buffers that the SDS can use for I/O. The more buffers added, the more I/Os the SDS can handle in parallel.

On every SDS node, configure the number of buffers by using the **set_num_of_io_buffers** command and setting its value for higher performance (default: 1, max: 10).

For higher performance, set the value to 3.

Note: When using the replication splitter for RecoverPoint, set the value to 5.

Syntax

```
scli --set_num_of_io_buffers (--sds_id <ID> |
--sds_name <NAME> | --sds_ip <IP>) --num_of_io_buffers
<VALUE>
```

Example

```
scli --set_num_of_io_buffers --sds_ip 10.25.137.41
--num_of_io_buffers 5
```

It is recommended to disable the caching on the SSD (flash device) storage pool. Use the following syntax:

```
scli --set_rmcache_usage
--protection_domain_name <protection_domain_name>
--storage_pool_name <pool name> --dont_use_rmcache
```

Linux

Perform the following steps, for all NICs in the ScaleIO system.

Note: Prior to activating MTU settings on the logical level, you must set Jumbo frames = MTU 9000\9126 on the physical switch ports that are connected to the server. Failure to do so may lead to network disconnects and packet drops.

Refer to your relevant vendor guidelines on how to configure jumbo frame support.

When enabling jumbo frames, one can expect approximately 10% improvement in performance if all the network components fully support jumbo frames. If some network components do not fully support jumbo frames, it is recommended to use an MTU of 1,500.

1. Perform one of the following:

   - For non-reboot persistent configurations, type the following command: **`Ifconf ethx MTU 9000`**
   - For persistent configurations, add the following line to the file `/etc/sysconfig/network-scripts/ifcfg-eth[n]`:

     ```
     MTU=9000
     ```

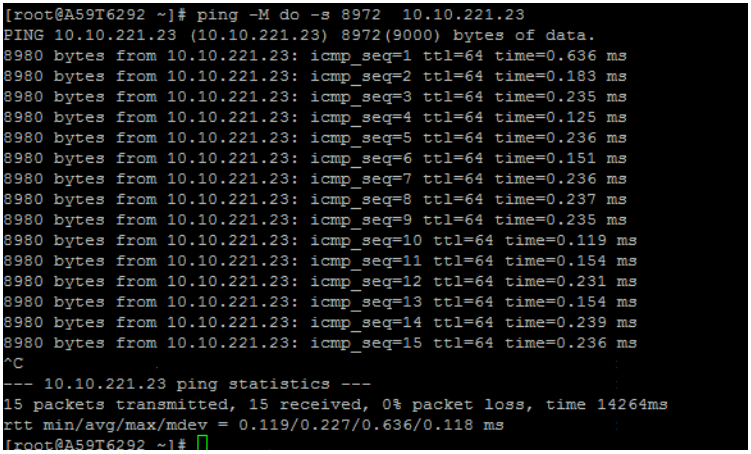
Type the command: **`Linux:ifconfig <NIC_NAME> MTU 9000`**

To make it reboot persistent, type the command: **`echo 'MTU=9000' >> /etc/sysconfig/network-scripts/ifcfg-<NIC_NAME>`**

To apply the changes, type the command: **`service network restart`**

To test the command, type the command: **`ping -M do -s 8972 <DESTINATION_IP_ADDRESS>`**

Output similar to the following should be displayed:



```
[root@A59T6292 ~]# ping -M do -s 8972  10.10.221.23
PING 10.10.221.23 (10.10.221.23) 8972(9000) bytes of data.
8980 bytes from 10.10.221.23: icmp_seq=1 ttl=64 time=0.636 ms
8980 bytes from 10.10.221.23: icmp_seq=2 ttl=64 time=0.183 ms
8980 bytes from 10.10.221.23: icmp_seq=3 ttl=64 time=0.235 ms
8980 bytes from 10.10.221.23: icmp_seq=4 ttl=64 time=0.125 ms
8980 bytes from 10.10.221.23: icmp_seq=5 ttl=64 time=0.236 ms
8980 bytes from 10.10.221.23: icmp_seq=6 ttl=64 time=0.151 ms
8980 bytes from 10.10.221.23: icmp_seq=7 ttl=64 time=0.236 ms
8980 bytes from 10.10.221.23: icmp_seq=8 ttl=64 time=0.237 ms
8980 bytes from 10.10.221.23: icmp_seq=9 ttl=64 time=0.235 ms
8980 bytes from 10.10.221.23: icmp_seq=10 ttl=64 time=0.119 ms
8980 bytes from 10.10.221.23: icmp_seq=11 ttl=64 time=0.154 ms
8980 bytes from 10.10.221.23: icmp_seq=12 ttl=64 time=0.231 ms
8980 bytes from 10.10.221.23: icmp_seq=13 ttl=64 time=0.154 ms
8980 bytes from 10.10.221.23: icmp_seq=14 ttl=64 time=0.239 ms
8980 bytes from 10.10.221.23: icmp_seq=15 ttl=64 time=0.236 ms
^C
--- 10.10.221.23 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14264ms
rtt min/avg/max/mdev = 0.119/0.227/0.636/0.118 ms
[root@A59T6292 ~]# 
```

2. To modify the I/O scheduler of the devices, type the following on each server, for each SDS device:

   `echo noop > /sys/block/<device_name>/queue/scheduler`

   Example

   `echo noop > /sys/block/sds/queue/scheduler`

Hint: To make these changes effective immediately, run **sysctl -p** after making the changes.

3. It is recommended to change the kernel tunables by copying the content of `/opt/emc/scaleio/sds/cfg/emc.conf` into `/etc/sysctl.conf` (while leaving `/opt/emc/scaleio/sds/cfg/scaleio.conf` as is).

   Note: You need to determine whether these parameters are beneficial for your environment.

Windows

Perform the following steps, for all NICs in the ScaleIO system.

Note: Prior to activating MTU settings on the logical level, you must set Jumbo frames = MTU 9000\9126 on the physical switch ports that are connected to the server. Failure to do so may lead to network disconnects and packet drops.

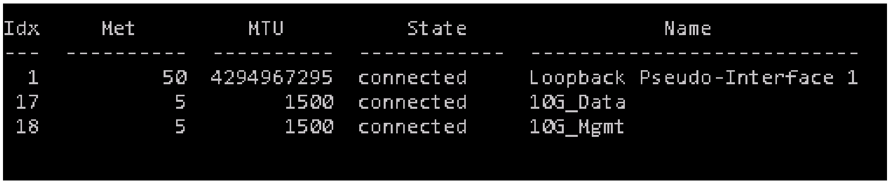
Refer to your relevant vendor guidelines on how to configure jumbo frame support.

When enabling jumbo frames, one can expect approximately 10% improvement in performance if all the network components fully support jumbo frames. If some network components do not fully support jumbo frames, it is recommended to use an MTU of 1,500.

1. Change the Maximum Transmission Unit (MTU) setting to 9,000, or the highest value that is supported by the switch and the connected nodes. First, determine the appropriate NIC name by typing the command:

   ```
   Netsh interface ipv4 show interface
   ```

   Output similar to the following should appear:

   ```
   Idx     Met         MTU          State                Name
   ---  ----------  ----------  ------------  ---------------------------
     1          50  4294967295  connected     Loopback Pseudo-Interface 1
    17           5        1500  connected     10G_Data
    18           5        1500  connected     10G_Mgmt
   ```

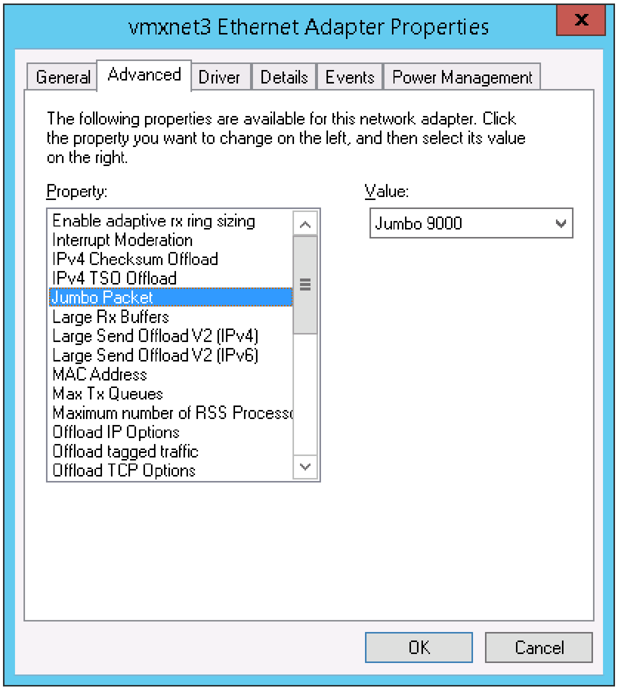
   In this example, index 17 is the appropriate network.

2. Type the command:

   ```
   netsh interface ipv4 set subinterface <network_ID>
   mtu=9000 store=persistent
   ```

   where *network_ID* is the ID from the output in the previous step, in this case `17`

3. In the **Advanced** tab of the **Adapter Properties** dialog for your vendor and driver, change the value of **Jumbo Packet** to **9000**, as illustrated in the following figure:



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

Enable adaptive rx ring sizing
Interrupt Moderation
IPv4 Checksum Offload
IPv4 TSO Offload
Jumbo Packet
Large Rx Buffers
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
MAC Address
Max Tx Queues
Maximum number of RSS Process
Offload IP Options
Offload tagged traffic
Offload TCP Options

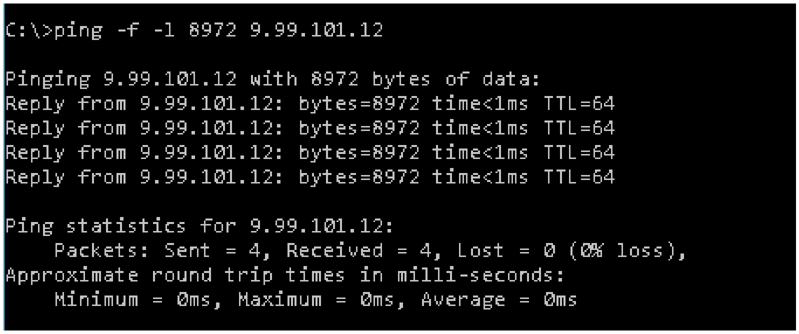
Value: Jumbo 9000

4. Click **OK**. The network connection may disconnect briefly during this phase.

5. Test that the configuration is working, by typing the command:

```
ping -f -l 8972 <Destination_IP_Address>
```

Output similar to the following should be displayed:



```
C:\>ping -f -l 8972 9.99.101.12

Pinging 9.99.101.12 with 8972 bytes of data:
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64

Ping statistics for 9.99.101.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Note: Ensure that the switch supports 10 GB ethernet.

ESX

Perform the following steps, for all NICs in the ScaleIO system.

Note: Prior to activating MTU settings on the logical level, you must set Jumbo frames = MTU 9000\9126 on the physical switch ports that are connected to the server. Failure to do so may lead to network disconnects and packet drops.

Refer to your relevant vendor guidelines on how to configure jumbo frame support.

When enabling jumbo frames, one can expect approximately 10% improvement in performance if all the network components fully support jumbo frames. If some network components do not fully support jumbo frames, it is recommended to use an MTU of 1,500.

1. Change the Maximum Transmission Unit (MTU) setting to 9,000 on the vSwitches and on the SVM (be sure to make the change in `/etc/sysconfig/network/ifcfg-ethX`):

   a. Type the command:

   ```
   esxcfg-vswitch -m 9000 <vSwitch>
   ```

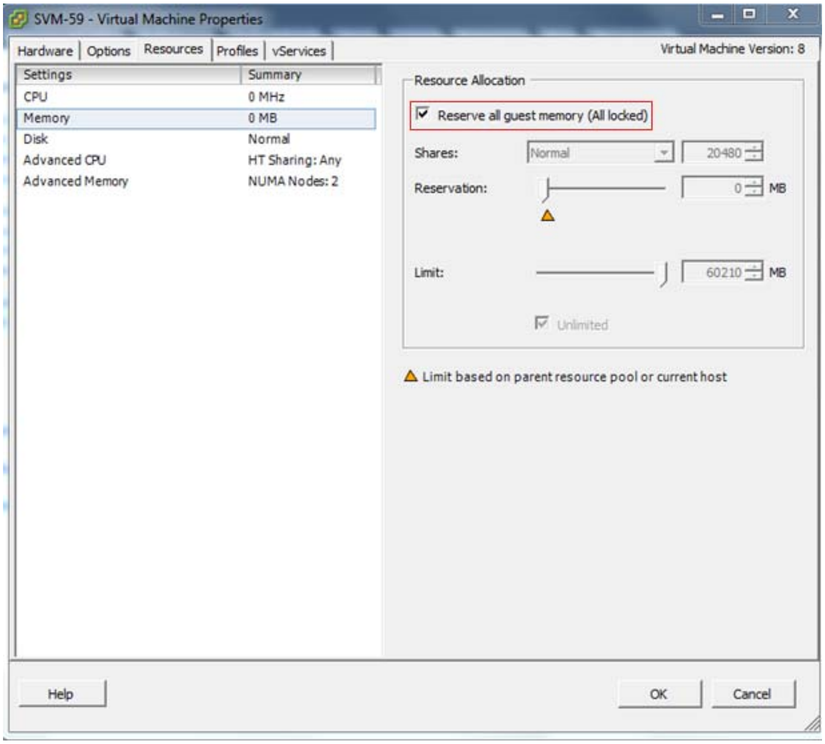
   b. Create VMKernel with jumbo frames support by typing the following commands:

   ```
   1. esxcfg-vswitch -d
   2. esxcfg-vswitch -A vmkernel# vSwitch#
   3. esxcfg-vmknic -a -i <ip address> -n <netmask> -m 9000
      <portgroup name>
   ```

   Note: If you change JumboFrames on an existing vSwitch, the packet size for VMKernel does not change, and therefore, the existing VMKernel must deleted and a new one must be created.

2. Double the CPU and memory assigned to the SVM. In general, 4 CPUs and 4GB of memory is sufficient, but this may vary in your environment.

3. From the **Resources** tab of the **Virtual Machine Properties** window, select **Reserve all guest memory (All locked)**.



Settings	Summary
CPU	0 MHz
Memory	0 MB
Disk	Normal
Advanced CPU	HT Sharing: Any
Advanced Memory	NUMA Nodes: 2

☑ Reserve all guest memory (All locked)

Shares: Normal 20480

Reservation: 0 MB

Limit: 60210 MB

☑ Unlimited

⚠ Limit based on parent resource pool or current host

Note: In addition to the above settings, every SVM is also a Linux machine, and should *also* be tuned according to the SDS and SDC settings described for Linux.

Tuning SDC nodes

You can achieve optimum performance by making the following tunings on every SDC node, in both physical and virtual environments.

Linux

1. Edit the file `/etc/init.d/scini` on each SDC node, by adding the following parameters in the `/sbin/insmod $DRV_BIN` line. Adjust the parameter values to the needs of your workload.

   - **`netConSchedThrd=8 netSockSndBufSize=4194304`**
   - **`netSockRcvBufSize=4194304 mapTgtSockets=4`**

Parameter	Description
netConSchedThrd	Amount of SDC kernel threads involved in performing networking-related tasks
netSockSndBufSize	The default TCP socket send buffer size used by the SDC when opening connections
netSockRcvBufSize	The default TCP socket receive buffer size used by the SDC when opening connections
mapTgtSockets	Amount of TCP sockets that the SDC will open to connect to one SDS (not greater than 8)

   After the editing, the line should look similar to this:

```
/sbin/insmod $DRV_BIN netConSchedThrd=8
netSockSndBufSize=4194304 netSockRcvBufSize=4194304
mapTgtSockets=4
```

2. Restart the service by typing the following command:

```
service scini restart
```

NOTICE

If this command does not succeed in restarting the service, it could be because an application or file system is using the storage. Unmount the potential user and try again. If you are still unsuccessful, you will need to restart the machine, as directed at the end of this procedure.

Hint: To make these changes effective immediately, run **sysctl -p** after making the changes.

3. It is recommended to change the kernel tunables by copying the content of `/opt/emc/scaleio/sdc/cfg/emc.conf` into `/etc/sysctl.conf` (while leaving `/opt/emc/scaleio/sdc/cfg/scaleio.conf` as is).

   Note: You need to determine whether these parameters are beneficial for your environment.

   Note: Prior to activating MTU settings on the logical level, you must set Jumbo frames = MTU 9000\9126 on the physical switch ports that are connected to the server. Failure to do so may lead to network disconnects and packet drops.

   Refer to your relevant vendor guidelines on how to configure jumbo frame support.

   When enabling jumbo frames, one can expect approximately 10% improvement in performance if all the network components fully support jumbo frames. If some network components do not fully support jumbo frames, it is recommended to use an MTU of 1,500.

4. For all NICs in the ScaleIO system, perform one of the following (if the SDC resides on the same host as an SDS where you have already configured these parameters, proceed to the next step):

   - For non-reboot persistent configurations, type the following command: **`Ifconf ethx MTU 9000`**
   - For persistent configurations, add the following line to the file `/etc/sysconfig/network-scripts/ifcfg-eth[n]`:

     ```
     MTU=9000
     ```

     Type the command: **`Linux: ifconfig <NIC_NAME> MTU 9000`**

     To make it reboot persistent, type the command: **`echo 'MTU=9000' >> /etc/sysconfig/network-scripts/ifcfg-<NIC_NAME>`**

     To apply the changes, type the command: **`service network restart`**

     To test the command, type the command: **`ping -M do -s 8972 <DESTINATION_IP_ADDRESS>`**

Note: The previous changes will take effect on the next restart. To make the changes effective immediately, also run the commands from the Linux shell.

5. Restart the SDC node.

   Restarting is only necessary if the service did not restart in step 2 on page 10 .

VMware

From ScaleIO v1.31, the SDC can be installed either directly on ESX (preferred method), or on an SVM (ScaleIO Virtual Machine). If you install the SDC directly on ESX, after the SDC is installed, type the following esxcli command:

```
esxcli system module parameters set -m scini -p
"netConSchedThrd=4  mapTgtSockets=4
netSockRcvBufSize=4194304  netSockSndBufSize=4194304"
```

Furthermore, if you issue this command, ESX will delete other existing parameters. Therefore, the SDC GUID and MDM IP address should be provided as part of the same command.

For example:

```
esxcli system module parameters set -m scini -p
"netConSchedThrd=4 mapTgtSockets=4
netSockRcvBufSize=4194304 netSockSndBufSize=4194304
IoctlIniGuidStr=12345678-90AB-CDEF-1234-567890ABCDEF
IoctlMdmIPStr=192.168.144.128"
```

To increase per device queue length (which can be lowered by default by ESX to 32), type the following esxcli command:

```
esxcli storage core device set -d <DEVICE_ID> -O
<QUEUE_LENGTH>
```

where ‹QUEUE_LENGTH› can be number in the range 32-256 (default=32).

For example:

```
esxcli storage core device set -d
eui.16bb852c56d3b93e3888003b00000000 -O 256
```

Windows

Perform the following steps, for all NICs in the ScaleIO system.

Note: Prior to activating MTU settings on the logical level, you must set Jumbo frames = MTU 9000\9126 on the physical switch ports that are connected to the server. Failure to do so may lead to network disconnects and packet drops.

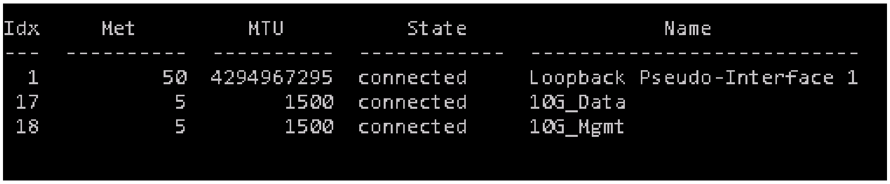
Refer to your relevant vendor guidelines on how to configure jumbo frame support.

When enabling jumbo frames, one can expect approximately 10% improvement in performance if all the network components fully support jumbo frames. If some network components do not fully support jumbo frames, it is recommended to use an MTU of 1,500.

1. Change the Maximum Transmission Unit (MTU) setting to 9,000, or the highest value that is supported by the switch and the connected nodes (if the SDC resides on the same host as an SDS where you have already configured these parameters, proceed to step 6 on page 15). First, determine the appropriate NIC name by typing the command:

   `Netsh interface ipv4 show interface`

   Output similar to the following should appear:

```
Idx     Met         MTU          State                Name
---  ----------  ----------  ------------  ---------------------------
  1          50  4294967295  connected     Loopback Pseudo-Interface 1
 17           5        1500  connected     10G_Data
 18           5        1500  connected     10G_Mgmt
```

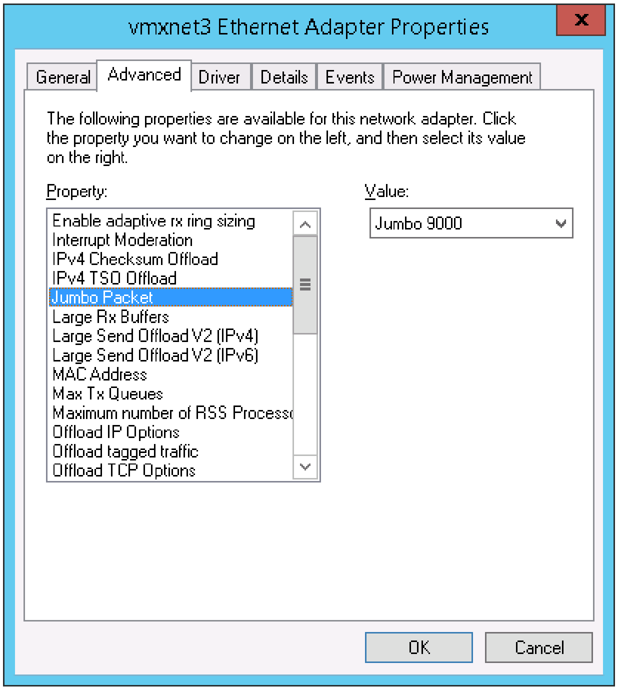
   In this example, index 17 is the appropriate network.

2. Type the command:

   `netsh interface ipv4 set subinterface <network_ID>`
   `mtu=9000 store=persistent`

   where *network_ID* is the ID from the output in the previous step, in this case `17`

3. In the **Advanced** tab of the **Adapter Properties** dialog for your vendor and driver, change the value of **Jumbo Packet** to **9000**, as illustrated in the following figure:



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

Enable adaptive rx ring sizing
Interrupt Moderation
IPv4 Checksum Offload
IPv4 TSO Offload
Jumbo Packet
Large Rx Buffers
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
MAC Address
Max Tx Queues
Maximum number of RSS Process
Offload IP Options
Offload tagged traffic
Offload TCP Options

Value:

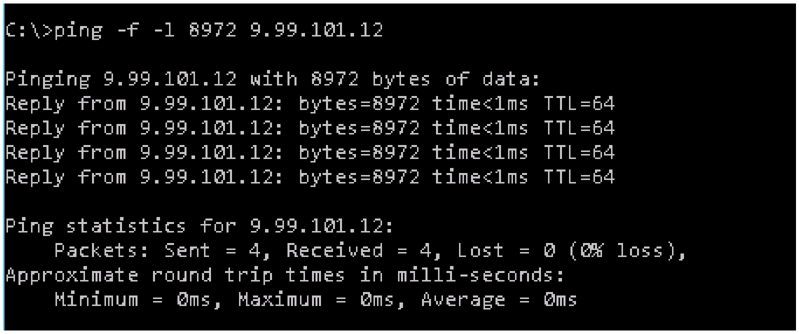
Jumbo 9000

4. Click **OK**. The network connection may disconnect briefly during this phase.

5. Test that the configuration is working, by typing the command:

```
ping -f -l 8972 <Destination_IP_Address>
```

Output similar to the following should be displayed:



```
C:\>ping -f -l 8972 9.99.101.12

Pinging 9.99.101.12 with 8972 bytes of data:
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64
Reply from 9.99.101.12: bytes=8972 time<1ms TTL=64

Ping statistics for 9.99.101.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Note: Ensure that the switch supports 10 GB ethernet.

6. Edit the SDC registry `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\scini\Parameters` as follows:
   - `"mapTgtSockets"=dword:00000004`
   - `"netSockRcvBufSize"=dword:04194304`
   - `"netConSchedThrd"=dword:00000008`
7. Restart the host machine.

   For a description of the parameters, see "Linux" on page 10.
8. In the **Controllers** section of the **Windows Device Manager,** disable and then enable the ScaleIO HBA device.

   NOTICE

   If the device does not restart, it could be because an application or file system is using the storage. Unmount the potential user and try again. If this does not work, restart the machine.

Additional network adjustments

The following network adjustments can also be performed, to increase network performance.

Windows

Disable delayed ack by modifying the following **REG_DWORD** to **1** on all ScaleIO network interfaces:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\<Interface GUID>\ TcpAckFrequency

ESXi

Disable delayed ack by typing the following command: `vsish -e set /net/tcpip/instances/defaultTcpipStack/sysctl/_net_inet_tcp_delayed_ack 0`

Linux (different on each Linux distribution):

Please refer to the Linux distribution documentation.

Troubleshooting and getting help

This section provides steps that you can take to troubleshoot performance-specific issues.

Large block I/O performance is lower than expected

If performance degradation on large block read/write I/O has been observed, perform the following steps to remedy the issue:

MDM

1. Login to the MDM ScaleIO VM.
2. Increasing the number of tgt (SDS) sockets per IP can improve write I/O performance. On each SVM containing an MDM, add the following line to `/opt/emc/scaleio/mdm/cfg/conf.txt`:

   ```
   Mdm_tgt_sockets_per_ip = 4
   ```

   Note: Very large clusters may not benefit from changing this setting.

3. Reset MDM by deleting the service in `/opt/emc/scaleio/mdm/bin/delete_service.sh`, and recreating the service with `/opt/emc/scaleio/mdm/bin/create_service.sh`.
4. Wait for the cluster to stabilize and return to operational, then repeat the above steps on the second MDM.

SDS

1. Login to one SDS ScaleIO VM (the first one can also be an MDM).
2. Reset each SDS with the following commands:

   a. `/opt/emc/scaleio/sds/bin/delete_service.sh`

   b. `/opt/emc/scaleio/sds/bin/create_service.sh`

3. Wait for rebuilds and rebalances to finish, and then repeat the flow on each of the SDSs on the other nodes.

SDC

1. Log in to the first ESX.
2. Execute the command:

   ```
   esxcli system module parameters set -m ixgbe -p "LRO=1"
   ```

3. From the vSphere client, migrate all the VMs to another ESX sharing the same volumes.
4. From the vSphere client, initiate an ESX restart cycle.
5. Wait for the rebuild and rebalance to finish.
6. Repeat these procedures on the next ESX in the system.

EMC support, product, and licensing information can be obtained as follows:

Product information — For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at:

https://support.emc.com

Technical support — Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

Published December 16, 2015

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com).