

## Superior NFS Performance using Solarflare Server Adapters

This technical note summarizes performance benchmarking of Linux NFS over 10Gb Ethernet (10GbE) using Solarflare Server Adapters. This note outlines the benefits of 10GbE over Gigabit Ethernet (GbE) for NFS thereby providing a case for upgrading GbE infrastructure to 10 GbE, and to demonstrate that Solarflare Server Adapters offer a cost-effective, high- performance solution whose performance exceeds that of other 10GbE adapters on the market.

NFS is a shared file system technology used both in work groups and in high-performance clusters. While alternative solutions exist, NFS still remains a popular choice given its maturity, ease of configuration, and ease of deployment. Since NFS runs over Ethernet, NFS benefits from performance improvements in the underlying Ethernet network. Specifically, results documented in this technical note provide a rationale for building out new NFS deployments with 10GbE technology as well as a rationale for upgrading existing Ethernet network infrastructure from GbE to 10GbE. Since NFS is a good proxy for other file system storage protocols, Solarflare expects that its 10GbE server adapters will also have superior performance when used with parallel file system solutions such as Lustre.

### METHODOLOGY

In order to evaluate NFS performance a suitable benchmark was chosen, a server and client configuration was devised, and a set of benchmarks were conducted. To simplify benchmarking and to reflect how most customers utilize server adapters, drivers were loaded with default settings. NFSv3 was benchmarked with default settings.

iozone was chosen as the benchmark because it is a commonly used benchmark for NFS and storage benchmarking, it supports multi-client cluster testing, and is freely available for download allowing other parties to replicate the results.

To avoid cache effects and any influence of the server's file I/O RAID subsystem on the benchmark results, the server was configured to export a RAM disk as the shared file system. Although such a configuration would not exist in practice (i.e. the shared file system would be a RAID volume), the configuration ensured that observed results demonstrated the maximum potential NFS performance. Such performance would be achieved in real world deployments, if the file being accessed was in the server's cache or if the server's File I/O could sustain the respective file access rate.

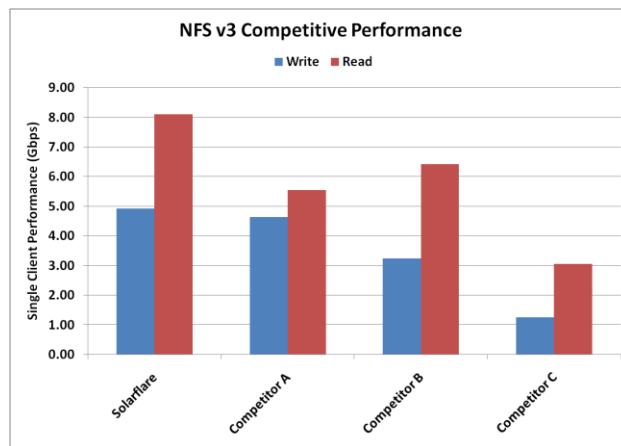
To avoid cache effects on the client side, it is important to boot the client with a restricted system memory size. Specifically system memory was set to half of the iozone benchmark's file size.

The following NFS benchmark experiments were conducted:

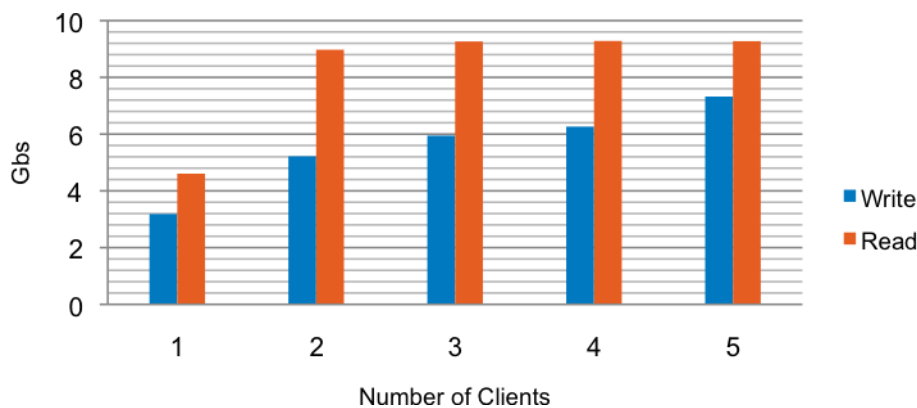
- Back to back (single client) Solarflare server adapter
- Back to back (single client) Competitor benchmarks
- Solarflare server adapter cluster testing (up to 5 clients)

## RESULTS

All tests were conducted using Redhat Enterprise 5.5. Back to back test results, as summarized in the first chart below show Solarflare performance leading all other leading competitors. Of note, 10GbE performance improvement over 1GbE is almost 10x.



Cluster testing demonstrates 10GbE NFS scaling as each client is added to the benchmark, until all the available 10GbE bandwidth is used:



*Note: Single client read/write performance in the cluster test is less than back to back test due to the additional switch latency.*

## SUMMARY

The results show clear benefits of 10GbE for NFS. The results have some limitations, as I/Ozone is a benchmark that serves as an estimate of many real world workloads. An alternative to I/Ozone, *Filebench* (<http://www.fsl.cs.sunysb.edu/~vass/filebench/>) is designed to model differing workloads such as web-server, video streaming etc.

In addition, higher performance could be achieved by benchmarking NAS servers deployed with redundancy in the server's NFS network connectivity (i.e. utilize both ports in a dual port Solarflare adapter in a team). Such benchmarking may require tuning to optimize the interaction between network adapter driver, network stack, NFS server and I/O RAID subsystem.

## BENEFITS

As expected 10GbE NFS provides a significant improvement over GbE NFS, but the improvement is limited for a single client to approximately 4x for writes and 7x for reads. Multi-client benchmarks demonstrate the full benefit of 10GbE, in that aggregate read performance of >9Gbps can be achieved and write performance scales with the number of clients.

Single client performance is limited by NFS's network caches which are designed to fill the network, but which cannot completely fill the available 10GbE bandwidth. Newer Linux kernels do allow for NFS read-write cache tuning, but for some workloads large cache sizes may be detrimental. As such actual single client NFS performance is in part dictated by the overall network latency between the client and the NFS server. Although not tested, a common upgrade path will be to at first upgrade the NFS server with 10GbE, leaving clients with 1GbE NFS interfaces. Such an upgrade should have an immediate positive impact on aggregate NFS performance.

## BENCHMARKING DETAILS

### General Server Configuration:

Kernel option "ramdisk\_size=5000000", then as root:

```
# mkdir -p /srv/ram
# mke2fs -O dir_index -m0 /dev/ram0
# mount -o noatime /dev/ram0 /srv/ram
# chmod 1777 /srv/ram
```

### Add to /etc/exports:

```
/srv/ram *(rw)
```

**Then start NFS:**

```
# service nfs start  
# exportfs -a
```

**General Client Configuration:**

```
# mkdir /mnt/ram  
# mount <server adapter IP address>:/srv/ram /mnt/ram
```

**Back to Back Test Setup:**

Intel® DX58S0 (chipset= Intel-x58; CPU=1 x core i7 920 (2.67GHz))

**On client:**

```
$ cd /mnt/ram  
$ iofzone -a -n 1400m -s 1400m
```

**Back to Back Table of Results:**

**Cluster Setup:**

Dell™-R410 (chipset=Intel-5500; CPU=2 x quadcore E5620 2.40GHz)

Configure password-less ssh on all clients; on one of the clients setup "clients" config file as documented by IOZONE:

```
$ export RSH=ssh  
$ cd /mnt/ram  
$ iofzone -t <num clients 1,2,3,4,5> -r4 -i 0 -i 1 -s 4m -+m clients
```