Best Practices for Oracle Solaris Network Performance with Oracle VM Server for SPARC

by Jon Anderson, Pradhap Devarajan, Darrin Johnson, Narayana Janga, Raghuram Kothakota, Justin Hatch, Ravi Nallan, and Jeff Savit

How to ensure you get the best virtual network performance from Oracle Solaris with Oracle VM Server for SPARC.

Published July 2014

Introduction

This article presents a set of best practices which can be used to improve virtual networking performance on Oracle VM Server for SPARC.

Please note that the configuration and setup of Logical Domains (LDOMs) is beyond the scope of this document except in the specific context of virtual network performance. The information included here is intended as a companion, not a replacement, for the official Oracle VM Server for SPARC documentation and hardware-specific documentation.

About Oracle VM Server for SPARC

Oracle VM Server for SPARC (previously called Sun Logical Domains) provides highly efficient, enterprise-class virtualization capabilities for Oracle's SPARC T-Series servers and supported M-Series servers from Oracle.

Oracle VM Server for SPARC allows you to create multiple virtual servers on one system to take advantage of the massive thread scale offered by supported SPARC servers and the Oracle Solaris operating system (OS). All the virtualization capabilities described in this document are a standard part of Oracle Solaris that are provided at no additional cost.

Oracle VM Networking and Performance

CPU and memory performance are minimally impacted in control and logical (guest) domains. However, I/O and networking performance can potentially be impacted by the overhead of an I/O domain. You are encouraged to assess your Oracle VM I/O and networking performance with your load requirements.

Note: All information provided here relates to Oracle's SPARC T4 and SPARC T5 platform and the 10 GbE IXGBE interface.

Additionally, Oracle recommends balancing network load across multiple guest domains, hardware I/O, and compute resources.

As a rule of thumb, for Oracle VM Server for SPARC 3.1 or above, it is recommended that **at least** one full core (SPARC T4 and SPARC T5 platforms) be used for each 10 Gb/sec Ethernet device. However, for maximum performance, two full cores are recommended. This number was determined during internal empirical testing. Ideally, domains should be assigned their own CPU cores so that there is no intra-domain competition for CPU resources.

Oracle VM offers flexibility in the assignment of I/O resources to domains, particularly in the realm of networking. With this flexibility comes a wide range of performance possibilities. This article strives to provide guidance regarding the most suitable configuration for certain workloads.

Oracle VM supports networking features that allow you to do the following:

- Create I/O domains—separate from the control domain that can be directly connected to a range of PCIe physical devices—from an individual PCIe port (I/O domain) to a PCIe root complex (root domain). Other domains can use virtual network devices to connect to an I/O domain. This gives granularity at the PICe root complex level or endpoint device level. For more information, refer to "I/O Domain Overview."

- Create as many I/O domains as there are PCIe endpoint devices in a system, because Oracle VM supports Direct I/O (DIO). Within a domain, you create a virtual PCIe block device and switch that allows direct connections to individual PCIe devices. PCIe I/O virtualization must be enabled for this function; see `ldm(1M) set-io iov=on`. For more information, "refer to Creating an I/O Domain by Assigning PCIe Endpoint Devices."

- Parcel out a PCIe device separately to individual domains, without the intervention of the CPU or hypervisor. This is because multiple domains can share a single PCIe device through the PCIe Single Root I/O Virtualization (SR-IOV) feature. For more information, refer to "Creating an I/O Domain by Assigning PCIe SR-IOV Virtual Functions."

Oracle VM networking provides the following benefits:

- Direct access to hardware that can be shared widely across domains

- Potentially, the need for fewer adapters, fewer switch ports, less cabling, and lower power costs

- Reduced or eliminated dependency on the primary domain for I/O

Domains with these features configured can be migrated while running, without having to interrupt their activities.

Note: Live Migration when SR-IOV Virtual Functions (VFs) are in use is available only with the Dynamic SR-IOV feature introduced by Oracle VM Server for SPARC 3.1. Earlier versions of Oracle VM Server for SPARC support the Hybrid I/O feature. This allows up to three virtual network devices per NIU (nxge

driver) to have directly assigned Direct Memory Access (DMA) resources. In the latest Oracle VM software, this feature is deprecated in favor of SR-IOV and support is limited to only UltraSPARC T2 CPU–based platforms.

Generally, it is recommended that you run the latest version of Oracle VM Server for SPARC available and keep systems software *and* firmware updated. Oracle highly recommends that control and I/O domains run the latest version of Oracle Solaris 11.1 (SRU9+), rather than Oracle Solaris 10, to maximize functionality and performance.

Although both Oracle Solaris 10 and Oracle Solaris 11 can be used as a guest OS, for best performance and to use additional features, it is recommended to deploy Oracle Solaris 11.1 (SRU9 or later revisions).

Virtual Networking in Oracle VM

There are two basic components involved in virtual networking in Oracle VM:

- Virtual switch (vsw): A virtual switch is similar to an Ethernet switch. It runs in an I/O or service domain and switches Ethernet packets over Logical Domain Channels (LDC).
- Virtual network device (vnet): A virtual network device is plumbed into the guest domain. It is analogous to a physical network device in a non-virtualized environment, where you plug one end of the cable into the network port and the other end of that cable into a switch. In the virtual world, you do the same; you create a virtual network device for your logical domain and connect it to a virtual switch in an I/O or service domain.

The performance of the virtual network depends on both domains involved: I/O or service and guest. The transfer of packets involves copying of data, necessarily requiring CPU cycles. In the absence of data movement, no cycles are consumed. In budgeting CPU resources, you should consider the maximum amount of bandwidth you expect to support on a given domain. Resource consumption should be regularly monitored to ensure optimum performance.

Virtual Networking Resource Management

Oracle recommends the following:

- For control domains (if they are supporting high I/O) and for I/O or service domains, it is recommended that you assign 8 GB of memory and at least two full CPU cores (assuming a SPARC T4 or SPARC T5 platform) for each 10 Gb/sec of required bandwidth. This is in addition to the resource requirements of other virtual services that the domain might be provisioning.
- For a guest domain, 4 GB of memory and two full CPU cores per 10GbE link is recommended. This ensures adequate fanout capability for the vnet device. For Oracle Solaris 11–based domains, you can check the effective fanout by using the `dladm show-linkprop <linkname>` command to list the CPU fanout resources associated with the vnet data link.

  ```
  # dladm show-linkprop net1
  ...
  ```

```
net1      cpus-effective       r-    0-15
net1      rxfanout-effective   r-    8
net1      rxrings-effective    r-    8
net1      txrings-effective    r-    8
...
```

Note that the default `rxfanout-effective` value is `8` for achieving the best performance on a 10 GbE link.

- For configurations that require several LDOMs with higher bandwidth, Oracle recommends allocating domain Virtual Central Processing Units (vCPUs) on core boundaries. See the `ldm(1M)` `add-core`, `set-core`, `add-domain`, and `set-domain` subcommands and the whole-core constraint. The number of LDOMs supported with the whole-core constraint is dependent on the underlying platform. Oracle recommends using Oracle Solaris Zones within optimally configured LDOMs if finer-grained resourcing is required.

- For performance-sensitive configurations, Oracle recommends whole-core allocations to minimize the "cache pollution" effects that can be present when execution context is shared across disparate virtual memory regions. This means different execution contexts with different data access patterns can cause premature invalidation of cache lines. Depending on the application, cache pollutioncan significantly impact performance.

Inter-domain communication within Oracle VM Server for SPARC is carried over point-to-point interfaces called Logical Domain Channels (LDC). Each connection of a virtual resource uses an LDC. With virtual networking, each vnet is linked to the vsw by an LDC and is also, by default, linked directly to every other vnet attached to the vsw. The number of LDCs required increases exponentially with each vnet device attached. Although the number of LDCs available is finite, this is normally not a problem unless the LDOMs configuration is extremely complex. In such a situation, it is possible to reduce the vnet/vsw LDC consumption by disabling the `inter-vnet-link` property on the switch. See the `ldm(1M)` manual page for more details.

Disabling the `inter-vnet-link` property on the switch causes the traffic from a domain to be routed through the service domain (hosting vsw), which has a significant performance impact on inter-domain communication. Hence, for the best overall performance, it is highly recommended that the `inter-vnet-link` property be enabled (which is the default setting).

You can look at LDC usage on the system from the control domain using the `ldm list-bindings -e` command. You can also get an overview using the `kstat -p|grep ldc` command.

Virtual Networking Software

For the best virtual networking performance, Oracle recommends running the latest Oracle VM Server for SPARC software (currently version 3.1). Please refer to the release notes to determine the minimum hardware and software requirements. As mentioned previously, to maximize functionality and performance, Oracle strongly recommends running Oracle Solaris 11.1 SRU9+ in any non-guest (control,

IO/service) domain. To benefit from recent significant improvements in vnet and vsw performance, run at least Oracle Solaris 11.1 SRU9 in both the I/O or service and guest domains. For Oracle Solaris 10, install 150031-07 or a later patch on guest domains.

Apart from running the recommended software, it is also necessary to ensure that `extended-mapin-space` is set to `on` in both the guest and I/O or service domains that are hosting the virtual switch. Oracle VM Server for SPARC software version 3.1 or later and associated firmware set this property to `on` by default. To check it, run the following command:

```
# ldm ls -l <domain-name> |grep extended-mapin
```

If `extended-mapin-space` is not `on`, such as in LDOMs that predate the Oracle VM Server for SPARC 3.1 software upgrade, you can turn it `on` by using the following command:

```
# ldm set-domain extended-mapin-space=on <domain-name>
```

The changes to the `extended-mapin-space` property trigger a delayed reconfiguration in the primary domain and require a reboot. LDOMs need to be stopped and restarted after a reboot.

You can check the mode configured on the LDCs by using the `kstat` command, for example:

```
# kstat -p|grep dring_mode
vnet:0:vnetldc0x0:dring_mode    4
vnet:0:vnetldc0x3:dring_mode    4
vnet:1:vnetldc0x1:dring_mode    4
vnet:1:vnetldc0x6:dring_mode    4
```

The following are the available modes:

```
#define VIO_TX_DRING            0x1
#define VIO_RX_DRING            0x2
#define VIO_RX_DRING_DATA       0x4
```

With current software and `extended-mapin-space` set to `on`, `dring_mode` should be `4`.

Virtual Networking Performance

The key performance benefits delivered by the latest Oracle VM and Oracle Solaris software are realized through improved code efficiency and the large send offload (LSO) feature. This feature allows the TCP protocol stack to write large packets to the data link that handles the framing, which amortizes per-packet costs in the stack similar to what Jumbo Frames do at the data link layer. LSO can be observed through the following `kstat` statistics.

```
# kstat -p|grep lso
tcp:0:tcpstat:tcp_lso_disabled  0
tcp:0:tcpstat:tcp_lso_enabled   168
tcp:0:tcpstat:tcp_lso_pkt_out   170173983
tcp:0:tcpstat:tcp_lso_times     32649030
```

```
vnet:0:vnetldc0x1:lso_enabled   1
vnet:0:vnetldc0x1:lso_ipackets  0
vnet:0:vnetldc0x1:lso_max_len   8192
vnet:0:vnetldc0x1:lso_opackets  20915690
```

The graphs shown in Figure 1 through Figure 3 are meant to illustrate possible performance with the latest software and a balanced LDOM configuration with sufficient hardware resources assigned. Actual performance might vary, subject to the workload, system, and network configuration.

Note: These graphs are provided for illustration purposes only; they are not a guarantee of present or future performance.

The following system configuration was used to obtain the performance measurements shown in Figure 1, Figure 2, and Figure 3.

- SPARC T4-1 server at 2.85 GHz
- 32 VCPU Control Domain (CDOM), 32 GB memory, IXGBE device
- 16 VCPU LDOM, 16 GB memory
- Oracle Solaris 11.1 SRU11.1 (in all LDOMs)
- `uperf` (www.uperf.org) using the `iperf` profile (TCP)
- Thirty-second test; 8 K I/O size (read and write)
- 64 K and 1 M TCP window sizes (WND)

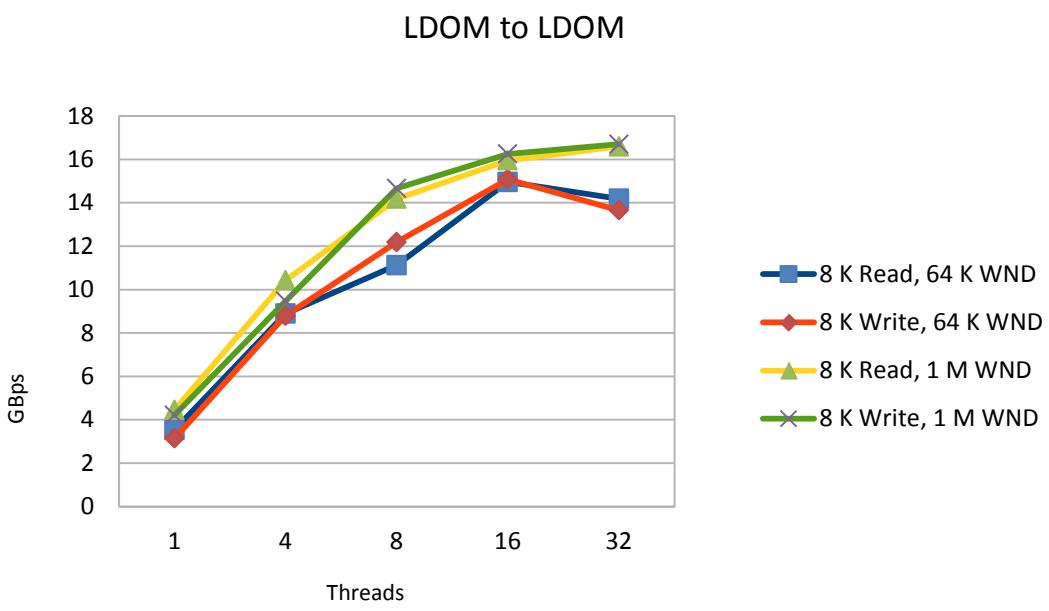

Figure 1. LDOM-to-LDOM Performance Measurements

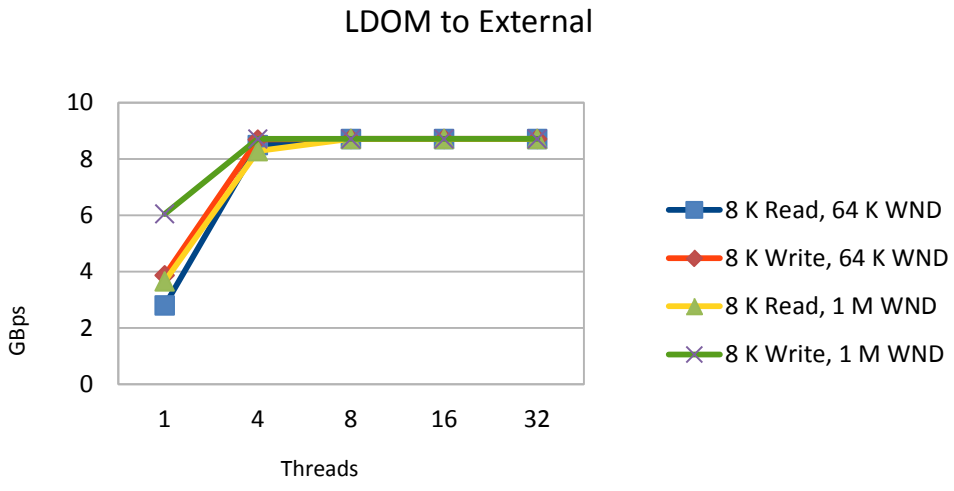

Figure 2. LDOM-to-External (vnet->vsw->IXGBE 10 GbE)* Performance Measurements

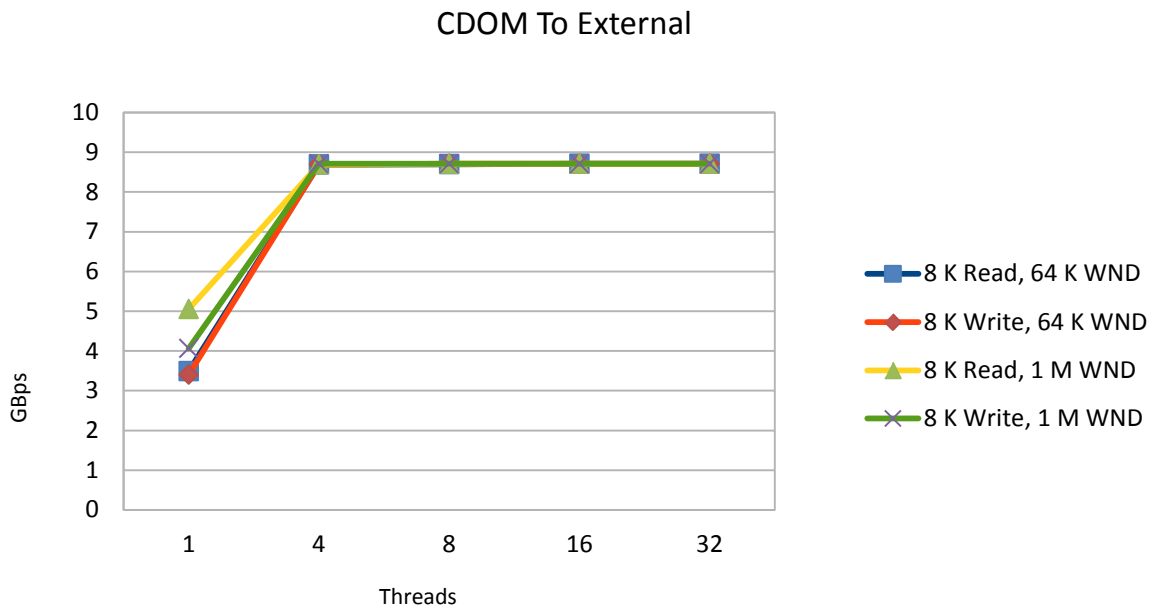

Figure 3. CDOM-to-External (via IXGBE 10 GbE) Performance Measurements

In this benchmark, network interface card (NIC) saturation occurs at a reported ~8.7 Gb/sec using the default MTU of 1500 bytes.

If you are running Oracle Solaris 11+, you can use the `dlstat(1M)` command, as shown in Listing 1, to view how traffic is being split across the pseudo media access control (MAC) rings associated with the vnet device, for example:

```
dlstat show-link [-r|-t] [-i <interval>] <vnet device>
```

```
# dlstat show-link -r -i 1 net1
LINK  TYPE      ID  INDEX    IPKTS   RBYTES    INTRS    POLLS   IDROPS
net1    rx   local     --        0        0        0        0        0
net1    rx   other     --        0        0        0        0        0
net1    rx      hw      0   38.98K  237.63M   38.98K        0        0
net1    rx      hw      1   65.54K  398.06M   65.54K        0        0
net1    rx      hw      2   13.14K   91.20M   13.14K        0        0
net1    rx      hw      3   23.59K  150.55M   23.59K        0        0
net1    rx      hw      4   31.25K  220.17M   31.25K        0        0
net1    rx      hw      5   57.22K  385.30M   57.22K        0        0
net1    rx      hw      6   67.02K  444.18M   67.02K        0        0
net1    rx      hw      7   57.83K  389.23M   57.83K        0        0
```

Listing 1

I/O Domain Networking Performance

For a contrast, Figure 4 shows some results using SR-IOV rather than vnet/vsw. A single SR-IOV virtual function (see `ldm(1M) create-vf`) was created by using the IXGBE device and then assigned to a four-core guest domain. The same I/O test was then repeated using `uperf` and the system configuration described earlier.

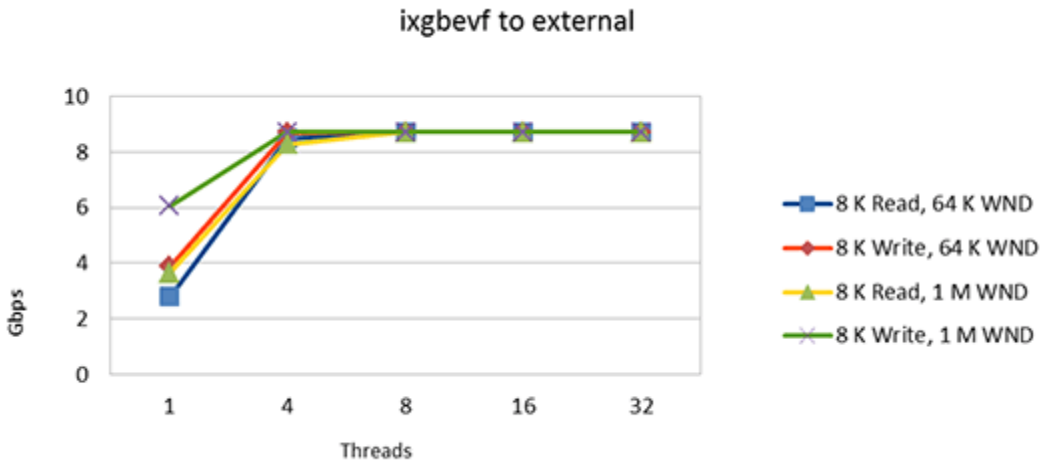

Figure 4. LDOM (IODOM)-to-External (via ixgbevf) Performance Measurements

Observe that performance is virtually equivalent to the physical device within the scope of these tests. The main differences appear to be fewer MAC rings and fewer interrupts (IXGBE has an interrupt per ring). This difference might be more significant on platforms that have lower single-thread performance. Note that this is still a shared resource; performance is expected to deteriorate as utilization of this resource increases. There is still only a single 10 GbE link underlying any virtual functions.

System Tunables

Generally, a guest domain behaves in much the same way as a regular system in that any specific tuning should be performed in response to specific workload requirements, for instance the TCP window sizes. This is entirely dependent on the requirements of the running applications. Applications drive system behavior.

The `ip_soft_rings_cnt` default is 2. The recommended value is the number of virtual CPUs.

This parameter applies to Oracle Solaris 10 only. Oracle Solaris 11+ implements a completely new MAC layer with built-in fanout capability dependent upon the platform architecture.

The `ip_soft_rings_cnt` variable determines the number of worker threads to be used to fan out the incoming TCP/IP connections. `ip_soft_rings_cnt` should be tuned based on system type and whether link aggregation is being used. Setting the value to the vCPU count/2 is a good initial setting. This value is multiplied internally by 2 on sun4v systems, which is why you need to divide what you want by 2.

Set the value in `/etc/system`:

```
set ip:ip_soft_rings_cnt=8
```

A reboot is required.

A setting of 8 should result in 16 soft rings being created per device.

Jumbo Frames

In Oracle VM Server for SPARC versions prior to 3.1, Jumbo Frames are beneficial in increasing network throughput because per-packet transfer costs are amortized. Oracle VM Server 3.1 (and dependent OS versions) provides the LSO feature, which improves the performance for the standard Ethernet MTU (1500 bytes) when using the TCP protocol. This obviates the requirement for Jumbo Frames in most cases. In fact, because of the difficulty in diagnosing path MTU problems that can arise, it is recommended that you avoid configuring the Jumbo Frames feature unless there is a specific need for it.

For Oracle VM versions prior to 3.1 (and requisite Oracle Solaris support) Jumbo Frames are invariably required to achieve the best network throughput.

Setting Jumbo Frames

The `default_mtu` variable should be set to a value equal to or less than the highest MTU supported by the networking infrastructure, including any switches and routers. A maximum MTU size of 9216 (1024*9) is common, but actual MTU size is dependent on the specifications of individual devices.

Whatever value you determine, it is critical that the MTU be set to this value on all virtual switches that are associated with the device. To change the MTU in your vnets, it is usually sufficient to change the MTU for the vsw device only, for example:

```
# ldm set-vswitch mtu=9000 vsw-10g-priv-primary
```

LDOMs that have a vnet through this vsw will need to be rebooted.

On Oracle Solaris 11, to change the MTU for a physical interface, set the `mtu` property of the physical data link using the `dladm set-linkprop` command. On Oracle Solaris 10, to change the MTU for a physical interface, please refer to the specific documentation for your hardware and for the Oracle Solaris release that you are running.

Caveats for Jumbo Frames

The primary caveat for Jumbo Frames is the requirement that the frames be enabled from end to end throughout the network infrastructure and within the virtual machine. Verification can often be done by running a bandwidth test (for example, sending 1 MB messages) and watching for hangs.

With the vnet/vsw changes introduced by Oracle VM Server for SPARC 3.1 (and at least Oracle Solaris 11.1 SRU9 or Oracle Solaris 10 version with 150031-07 or a later patch), virtual network performance when using TCP is actually better without Jumbo Frames due to LSO.

In Oracle Solaris 11 you can check supported MTUs by running the `dladm show-linkprop -p mtu` command. The output will list supported MTUs in the `POSSIBLE` column. Non-standard MTUs are not supported on virtual functions (VFs) of the Intel 82599 chipset, for example, Niantic-based cards, such as X1109a-z. Refer to `/kernel/drv/ixgbevf.conf`.

Link Aggregation

Link aggregation (IEEE 802.3ad) provides a mechanism for one or more network links to be aggregated to form a link aggregation trunk (. The aggregated link appears to clients as a single data link. Network configuration done with the vsw created on the aggregated link enables the following:

- Increased near-linear bandwidth, as long as there are sufficient multiple, simultaneous network streams to saturate the bandwidth
- High availability for the vnet created on this vsw

The efficiency of an aggregated link depends on the hashing policy chosen. See the `dladm(1M)` manual page. The effectiveness of an aggregated link can be monitored by using the `dlstat(1M)` command if you are running Oracle Solaris 11+, for example:

```
# dlstat show-aggr -r -i 1 aggr1
```

Note that to realize the full potential, link aggregation requires workloads that can saturate aggregated bandwidth.

See Also

- *Oracle VM Server for SPARC 3.1 Administration Guide*, Chapter 8: "Using Virtual Networks"
- *Oracle Solaris 11 Tunable Parameters Reference Manual*
- *Oracle Solaris 10 Tunable Parameters Reference Manual*
- "Virtual network performance greatly improved!"

About the Authors

Jon Anderson and Pradhap Devarajan are software developers in Oracle's Systems RPE group, Darrin Johnson is Software Director for the group, and Narayana Janga is a senior principal engineer in the group.

Raghuram Kothakota is a software developer for the Oracle VM group, and Jeff Savit is the Principal Technology Product Manager for the group.

Revision 1.0, 07/24/2014