Recommendation for the Real Application Cluster Interconnect and Jumbo Frames (Doc ID 341788.1)

In this Document

APPLIES TO:

Oracle Database Cloud Schema Service - Version N/A and later
Oracle Database Cloud Service - Version N/A and later
Oracle Database - Enterprise Edition - Version 10.2.0.4 and later
Gen 1 Exadata Cloud at Customer (Oracle Exadata Database Cloud Machine) - Version N/A and later
Oracle Cloud Infrastructure - Database Service - Version N/A and later
Information in this document applies to any platform.
Oracle Server Enterprise Edition - Version: 9.2.0.1 to 11.2

PURPOSE

This note covers the current recommendation for the Real Application Cluster Interconnect and Jumbo Frames

SCOPE

This article points out the issues surrounding Ethernet Jumbo Frame usage for the Oracle Real Application Cluster (RAC) Interconnect. In Oracle Real Application Clusters, the Cluster Interconnect is designed to run on a dedicated, or stand-alone network. The Interconnect is designed to carry the communication between the nodes in the Cluster needed to check for the Clusters condition and to synchronize the various memory caches used by the database.

Ethernet is a widely used networking technology for Cluster Interconnects. Ethernet's variable frame size of 46-1500 bytes is the transfer unit between the all Ethernet participants, such as the hosts and switches. The upper bound, in this case 1500, is called MTU (Maximum Transmission Unit). When an application sends a message greater than 1500 bytes (MTU), it is fragmented into 1500 byte, or smaller, frames from one end-point to another. In Oracle RAC, the setting of DB_BLOCK_SIZE multiplied by the MULTI_BLOCK_READ_COUNT determines the maximum size of a message for the Global Cache and the PARALLEL_EXECUTION_MESSAGE_SIZE determines the maximum size of a message used in Parallel Query. These message sizes can range from 2K to 64K or more, and hence will get fragmented more so with a lower/default MTU.

Jumbo Frames introduces the ability for an Ethernet frame to exceed its IEEE 802 specified Maximum Transfer Unit of 1500 bytes up to a maximum of 9000 bytes. Even though Jumbo Frames is widely available in most NICs and data-center class managed switches it is not an IEEE approved standard. While the benefits are clear, Jumbo Frames interoperability is not guaranteed with some existing networking devices. Though Jumbo Frames can be implemented for private Cluster Interconnects, it requires very careful configuration and testing to realize its benefits. In many cases, failures or inconsistencies can occur due to incorrect setup, bugs in the driver or switch software, which can result in sub-optimal performance and network errors.

DETAILS

Configuration

In order to make Jumbo Frames work properly for a Cluster Interconnect network, careful configuration in the host, its Network Interface Card and switch level is required:

- The host's network adapter must be configured with a persistent MTU size of 9000 (which will survive reboots).

  For example, ifconfig -mtu 9000 followed by ifconfig -a to show the setting completed.

- Certain NIC's require additional hardware configuration.

  For example, some Intel NIC's require special descriptors and buffers to be configured for Jumbo Frames to work properly.

  - The LAN switches must also be properly configured to increase the MTU for Jumbo Frame support. Ensure the changes made are permanent (survives a power cycle) and that both "Jumbo" refer to same size, recommended 9000 (some switches do not support this size).

- Because of the lack of standards with Jumbo Frames the interoperability between switches can be problematic and requires advanced networking skills to troubleshoot.

- Remember that the smallest MTU used by any device in a given network path determines the maximum MTU (the MTU ceiling) for all traffic travelling along that path.

Failing to properly set these parameters in all nodes of the Cluster and Switches can result in unpredictable errors as well as a degradation in performance.

Testing

Request your network and system administrator along with vendors to fully test the configuration using standard tools such as SPRAY or NETCAT and show that there is an improvement not degradation when using Jumbo Frames. Other basic ways to check it's configured correctly on Linux/Unix are using:

Traceroute: Notice the 9000 packet goes through with no error, while the 9001 fails, this is a correct configuration that supports a message of up to 9000 bytes with no fragmentation:

```
[node01] $ traceroute -F node02-priv 9000
traceroute to node02-priv (10.x.x.2), 30 hops max, 9000 byte packets
1 node02-priv (10.x.x.2) 0.232 ms 0.176 ms 0.160 ms

[node01] $ traceroute -F node02-priv 9001
traceroute to node02-priv (10.x.x.2), 30 hops max, 9001 byte packets
traceroute: sendto: Message too long
1 traceroute: wrote node02-priv 9001 chars, ret=-1
```

 * Note: Due to Oracle Bugzilla 7182 (must have logon privileges) -- also known as RedHat Bugzilla 464044 -- older than EL4.7 traceroute may not work correctly for this purpose.
 * Note: Some versions of tracroute, e.g. traceroute 2.0.1 shipped with EL5, **add** the header size on top of what is specified when using the -F flag (same as ping behavior below). Newer versions of traceroute, like 2.0.14 (shipped with OL6) have the old behavior of traceroute version 1 (size of packet is exactly as what is specified with the -F flag).

Ping: With ping we have to take into account an overhead of about 28 bytes per packet, so 8972 bytes go through with no errors, while 8973 fail, this is a correct configuration that supports a message of up to 9000 bytes with no fragmentation:

```
[node01]$ ping -c 2 -M do -s 8972 node02-priv
PING node02-priv (10.x.x.2) 1472(1500) bytes of data.
1480 bytes from node02-priv (10.x.x.2): icmp_seq=0 ttl=64 time=0.220 ms
1480 bytes from node02-priv (10.x.x.2): icmp_seq=1 ttl=64 time=0.197 ms

[node01]$ ping -c 2 -M do -s 8973 node02-priv
From node02-priv (10.x.x.1) icmp_seq=0 Frag needed and DF set (mtu = 9000)
From node02-priv (10.x.x.1) icmp_seq=0 Frag needed and DF set (mtu = 9000)
```

```
--- node02-priv ping statistics ---
0 packets transmitted, 0 received, +2 errors
```

```
For Solaris platform, the similar ping command is:
$ ping -c 2 -s  node02-priv 8972
```

* Note: Ping reports fragmentation errors, due to exceeding the MTU size.

Performance

For RAC Interconnect traffic, devices correctly configured for Jumbo Frame improves performance by reducing the TCP, UDP, and Ethernet overhead that occurs when large messages have to be broken up into the smaller frames of standard Ethernet. Because one larger packet can be sent, inter-packet latency between various smaller packets is eliminated. The increase in performance is most noticeable in scenarios requiring high throughput and bandwidth and when systems are CPU bound.

When using Jumbo Frames, fewer buffer transfers are required which is part of the reduction for fragmentation and reassembly in the IP stack, and thus has an impact in reducing the latency of a an Oracle block transfer.

As illustrated in the configuration section, any incorrect setup may prevent instances from starting up or can have a very negative effect on the performance.

Known Bugs

In some versions of Linux there are specific bugs in Intel's Ethernet drivers and the UDP code path in conjunction with Jumbo Frames that could affect the performance. Check for and use the latest version of these drivers to be sure you are not running into these older bugs.

The following bugzilla bugs 162197, 125122 are limited to RHEL3.

Recommendation

There is some complexity involved in configuring Jumbo Frames, which is highly hardware and OS specific. The lack of a specific standard may present OS and hardware bugs. Even with these considerations, Oracle recommends using Jumbo Frames for private Cluster Interconnects.

Since there is no official standard for Jumbo Frames, this configuration should be properly load tested by Customers. Any indication of packet loss, socket buffer or DMA overflows, TX and RX error in adapters should be noted and checked with the hardware and operating system vendors.

The recommendation in this Note is strictly for Oracle private interconnect only, it does not apply to other NAS or iSCSI vendor tested and validated Jumbo Frames configured networks.

Oracle VM didn't support Jumbo in all ovm2 versions and all ovm3.0, however starting with ovm3.1.1 it's supported, refer to:

http://www.oracle.com/us/technologies/virtualization/ovm-3-1-whats-new-1634275.pdf

To procedure to change MTU, refer to note 283684.1 - "Changing private network MTU only"

REFERENCES

BUG:13332363 - BUG 9795321 HAS REAPPEARED IN 11.2.0.3
NOTE:283684.1 - How to Modify Private Network Information in Oracle Clusterware
NOTE:1085885.1 - CRS root.sh Script Failing on Second Node When MTU Larger than 1500
NOTE:1166925.1 - Oracle VM: Jumbo Frame on Oracle VM
NOTE:300388.1 - Instances Unable To Start If MTU Size Is Different for Cluster_interconnect
NOTE:1290585.1 - Solaris: Wrong MTU Size for VIP or HAIP

Didn't find what you are looking for?