

Thursday, March 21. 2013

## Less known Solaris Features - Data Link Multipathing

I used this feature in the HA-loadbalancing tutorial already. However the feature is too useful to stay just a "by-word" in a different article. It is DLMP. Or by its full name: "Data Link Multipathing".

Differences DLMP is a somewhat distant relative of IPMP, however it works on a different layer of the network stack. Whereas IPMP is something totally IP related and thus works on IP interfaces, the DLMP feature works on the data links. Out of this reason you can do a lot of more things with it ... for example setting up VRRP on it (as you manipulate MAC addresses with VRRP, it doesn't work on IPMP interfaces). DLMP is a closer relative of a trunk aggregation also known as IEEE 802.1ax respectively 802.3ad (Etherchannel, LACP are other related names for the same or similar things). I will call this type of aggregation "trunk aggregation" in this article.

You configure the DLMP aggregation with the `dladm` command and the configuration is quite similar to the trunk aggregation. You can even transform a trunk aggregation to DLMP aggregation. However both methods have different capabilities and constraints. With a trunk aggregation, all cables have to terminate in the same switch, with DLMP you can connect each interface to a different switch. Trunk aggregation can spread the load over all member interfaces of an aggregation. DLMP isn't capable to do so like a trunk aggregation. With DLMP you are just doing HA when you have just a single vnic. However when you configure multiple VNICs on an aggregation, it will distribute the VNICs on all the member interfaces. As DLMP doesn't support load spreading, you don't configure a load spreading policy for it. With trunks the switch has to support trunks and you have to configure the switch to accept such aggregations or you have to use LACP, a DLMP aggregation needs no support and no configuration on the switch.

How to use it DLMP really easy to use. Compared to the setup of a trunk aggregation, it is an additional `-m dlmp`:

```
root@solaris:~# dladm create-aggr -m dlmp -l net0 -l net1 aggr0
```

With a `show-link` you will see that you have an additional data link, which is an aggregation consisting out of `net0` and `net1`.

```
Just as we configured a second ago.root@solaris:~# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net0	phys	1500	up	--
net1	phys	1500	up	--
aggr0	aggr	1500	up	net0 net1

With `show-aggr` you can lookup more details: `root@solaris:~# dladm`

```
show-aggr -x
```

LINK	PORT	SPEED	DUPLEX	STATE	ADDRESS	PORTSTATE
aggr0	--	1000Mb	full	up	8:0:27:29:a2:89	--
	net0	1000Mb	full	up	8:0:27:29:a2:89	attached
	net2	1000Mb	full	up	8:0:27:ec:8a:7e	attached

Now unplug one of the cables `root@solaris:~# dladm`

```
show-aggr -x
```

LINK	PORT	SPEED	DUPLEX	STATE	ADDRESS	PORTSTATE
aggr0	--	1000Mb	full	up	8:0:27:ec:8a:7e	--
	net0	0Mb	unknown	down	8:0:27:29:a2:89	standby
	net2	1000Mb	full	up	8:0:27:ec:8a:7e	attached

Conclusion One use case for this feature is to solve a constraint introduced by using VRRP to make the integrated loadbalancing feature highly available as described in blog entry. Or you could use it to give all the VNICs you are using for zones a redundant connection to the network without working with IPMP and even introduce for example network resource manager much simpler than with IPMP working with resource controlled VNICs. Such configurations get complex very fast and DLMP can reduce this complexity.

Do you want to learn more [docs.oracle.com](http://docs.oracle.com) - Managing Oracle Solaris 11.1 Network Performance

Data Link Multipathing Aggregations

Differences of trunk and DLMP aggregations

Posted by Joerg Moellenkamp in English, Solaris at 21:44

Do you have any hints on how to debug LACP-related issues on Sol10? I've repeatedly encountered situations where a 2-line-channel from a T2 to a Cisco Catalyst 6500 failed to register as "up" as long as LACP was enabled. Neither the `dladm` output nor the syslogs are really useful.

Anonymous on Apr 1 2013, 13:38