Name resolution and network selection for PBSPro network messages on dual-network machines

0. Introduction

PBSPro makes certain assumptions about name resolution that do not always hold in modern clusters when there are machines with more than one interface (and certainly in clusters with nodes on different networks).

In PBSPro 13.1 variables were introduced to cope with this; this document will explore how to use them in typical situations (since the Admin Guide tends to merely document the presence of the variables), toegether with OS facilities to influence how TPC/IP packets are routed.

We do make assumptions in this document about the sanity of the name resolution framework in this document:
- every name resolves to **only one** IP address (so there is no ambiguity), i.e. no name appears on two lines in /etc/hosts;
- All different resolvers (hosts files, NIS, DNS, LDAP) agree (unless all relevant IP addresses are in /etc/hosts on <u>all</u> the hosts);
- ***Name resolution should be consistent across the entire cluster, i.e. that a given IP address resolves to the same canonical name across all hosts that run PBSPro daemons[1]***;
- The canonical name for each address (first on the line in /etc/hosts) is *usually* **fully qualified** (with domain name); the output of "hostname" is *usually* the short name.

External machines that are just used to submit jobs or run client commands (e.g. qstat, pbs_rsub) can resolve IP addresses to diffrerent names.

TPP and pbs_comm

In PBSPro 13.1, when TPP is enabled (which is the default) daemons communicate with each other through a pbs_comm daemon that does basic buffering but can also route between networks (or through a set of pbs_comm daemons, to be specified using PBS_LEAF_ROUTERS; pbs_comm network graphs can become quite complex).

By default, if PBS_LEAF_ROUTERS is unspecified, either PBS_SERVER or PBS_PRIMARY plus PBS_SECONDARY are the addresses at which daemons expect to find a pbs_comm daemon (and will register itself to all of them); in other words, if nothing special is specified, there is an active/active failover pair of pbs_comm daemons running on the primary and secondary servers, and if there is only one server, it also runs a single pbs_comm daemon.

A host will query all leaf routers it knows to find out if a target addresss has been registered to it and will use the first pbs_comm that tells it that it can reach a target. If the connection to the

---

1MoMs will often interpret names in exec_host, exec_host2 and exec_vnode attributes that were filled in by the server according to their local name resolution framework, so discrepancies can often cause strange failures (sometimes only for multinode jobs).

target is lost, then it will renegotiate anew connection with all leaf routers again (which means the network is resilient against failures of a single pbs_comm if all MoMs are e.g. connected to two pbs_comm daemons in a failover pair).

1. Nominal Situation
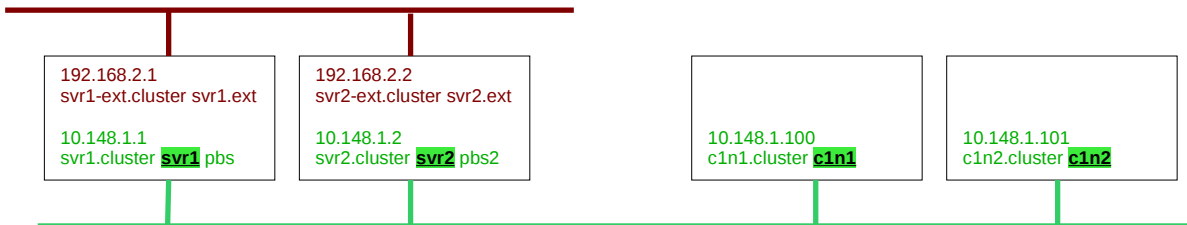
Here all hostnames (marked in colour boxes) resolve to an IP address that is on the network that PBSPro is meant to use.

The /etc/pbs.conf is the same on all nodes:

PBS_SERVER=svr1
PBS_PRIMARY=svr1.cluster
PBS_SECONDARY=svr2.cluster

Nodes are created using

qmgr -c "create node c1n1"

Note that *the name used to create a node must be the hostname of the node (output of "hostname")*. If not, hooks and MoM v2 config files may not work as expected, since MoM will silently assume the natural vnode name to be its hostname when merging information from hooks or v2 config files.

This will result in the following situation:
- pbs_comm on svr1 and svr2 will listen on all network interfaces
- svr1 and svr2 will register to both pbs_comm daemons using their 10.148.1.x address
- execution nodes will register to both pbs_comm daemons using their 10.148.1.10x address
- When trying to contact a MoM at server startup, the server will send a "hello" to the 10.148.1.10x addresses of the execution hosts, which will match the node's registered address
- When MoM boots it will contact PBS_SERVER (i.e. 10.148.1.1). If the server is failed over to the secondary, the node will remain down until the secondary tries to recover it in its "ping down nodes" routines (in less than 5 minutes).
- PBS commands on the execution host will be directed to svr1 (PBS_SERVER) and then to PBS_SECONDARY if it does not respond (svr2.cluster).

2. Server Hostname on Different Interface
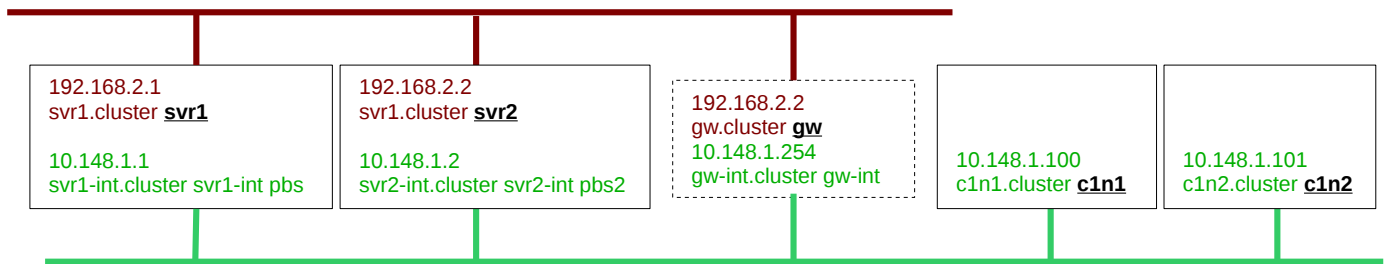
Here the same default /etc/pbs.conf will lead to very different results:

- pbs_comm daemons on svr1 and svr2 will listen to all network interfaces
- svr1 and svr2 will register to pbs_comm using their external network address 192.168.2.x
- Execution nodes will typically resolve PBS_SERVER and PBS_SECONDARY to either:
  - the 10.148.1.x addresses (since they are not on the external network), in which case pbs_comm will deny having "seen this host register itself"; it will simply not work
  - the 192.168.2.x addresses, which are unreachable without passing through a gateway, which becomes a single point of failure (or worse –- if it is a NAT gateway it may drop connections if its connection tracking table is full).

Please note that NAT gateways are not supported by Altair due to their complex tuning requirements to avoid dropping packets and connections, and their effects on PBSPro client command authentication: PBSPro client commands on execution nodes will <u>not</u> work if they have to cross a NAT gateway unless Munge authentication is used.

Obviously if there is no gateway node at all then the 192.168.2.x addresses are simply unreachable by internal nodes.

This can be optimised to remove the dependency on the gateway in more than one way, which is ***always recommended by Altair***:

i. host routes can be installed on the execution nodes so that they send packets destined for svr1 to svr1-int, and for svr2 to svr2-int

ii. /etc/pbs.conf on execution nodes can use the aliases "pbs" and "pbs2" in PBS_SERVER, PBS_PRIMARY and PBS_SECONDARY.
In that case, ***only on the servers***:
  - PBS_SERVER needs to also be set to "pbs": PBS_SERVER needs to be consistent across the cluster;
  - PBS_SERVER_HOST_NAME needs to be set to the hostname of the machine (since it no longer matches PBS_SERVER);
  - PBS_LEAF_NAME needs to be set to the "pbs" or "pbs2" alias so that the servers register to pbs_comm with the internal interface address.

iii. On execution nodes, PBS_LEAF_ROUTERS can be set to "svr1-int,svr2-int" so that MoMs know to approach the pbs_comm from the internal network. Note that in contrast to the two previous solutions client commands on execution nodes will still go through the gateway (and fail if it is down or if it is a NAT gateway)

Note that solution ii cannot be implemented after the fact without also changing the job suffix (the last old job will be called <N>.svr1 and the first new job will be <N+1>.pbs), so client commands (and dependencies) referring to jobs that were already present on the server will need to use the "old PBS_SERVER" as an explicit suffix. That is extremely messy, and can only be fixed by updating the suffix in the data store (which is not that hard) in the job, job_attr tables and in all dependency attributes of jobs.

3. Multinetwork Cluster

Sometimes a situation as described in the previous section cannot be avoided, if e.g. there are execution nodes on more than one network connected to the server:
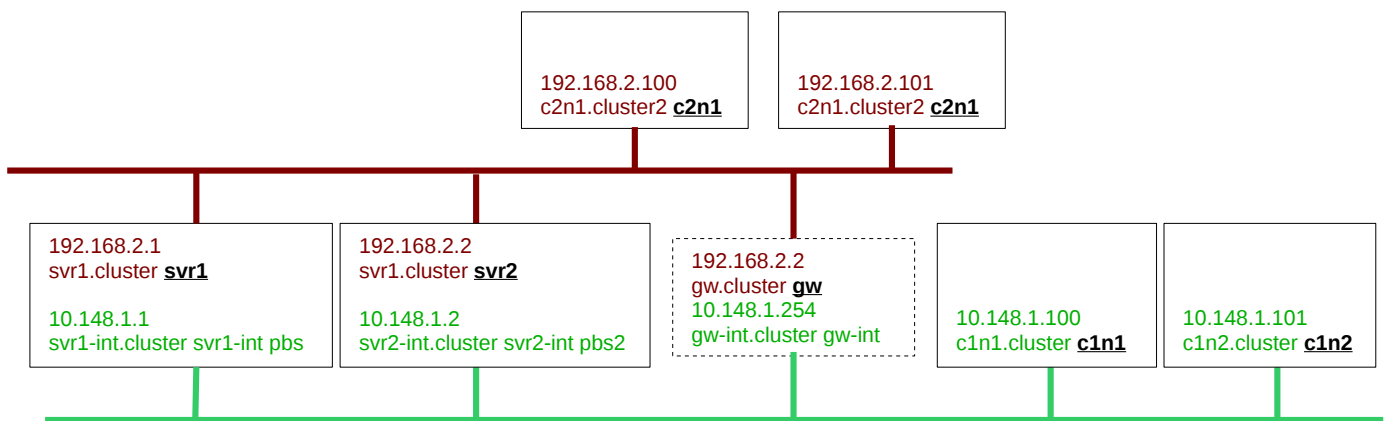
The same solutions apply here, except that the second solution cannot be used on its own (since the PBS leaf address for the server and PBS_SERVER cannot be on two networks at the same time).

4. Multihomed Execution Nodes with Hostname bound to "wrong" interface
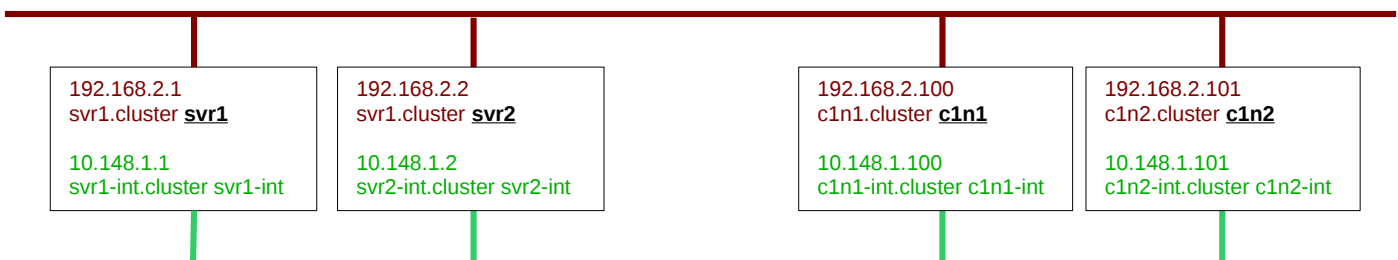
In this situation the execution nodes also have a hostname bound to the "wrong" interface, on the network where you do not want messages between pbs_comm and MoM daemons to pass.

It should be possible to use PBS_LEAF_NAME to tell the node to register with pbs_comm using its internal IP address 10.148.1.100.

However, until PBS-13894 is fixed, it will introduce problems for multihost jobs (a mismatch between the manes used in the exec_host2 job attribute and the execution node's hostname).

In this situation, it is thus better to actually still use the official hostnames when creating nodes, but to install host routes on the two server nodes so that they know to use the internal network to contact the nodes.