System Boot and RDMA

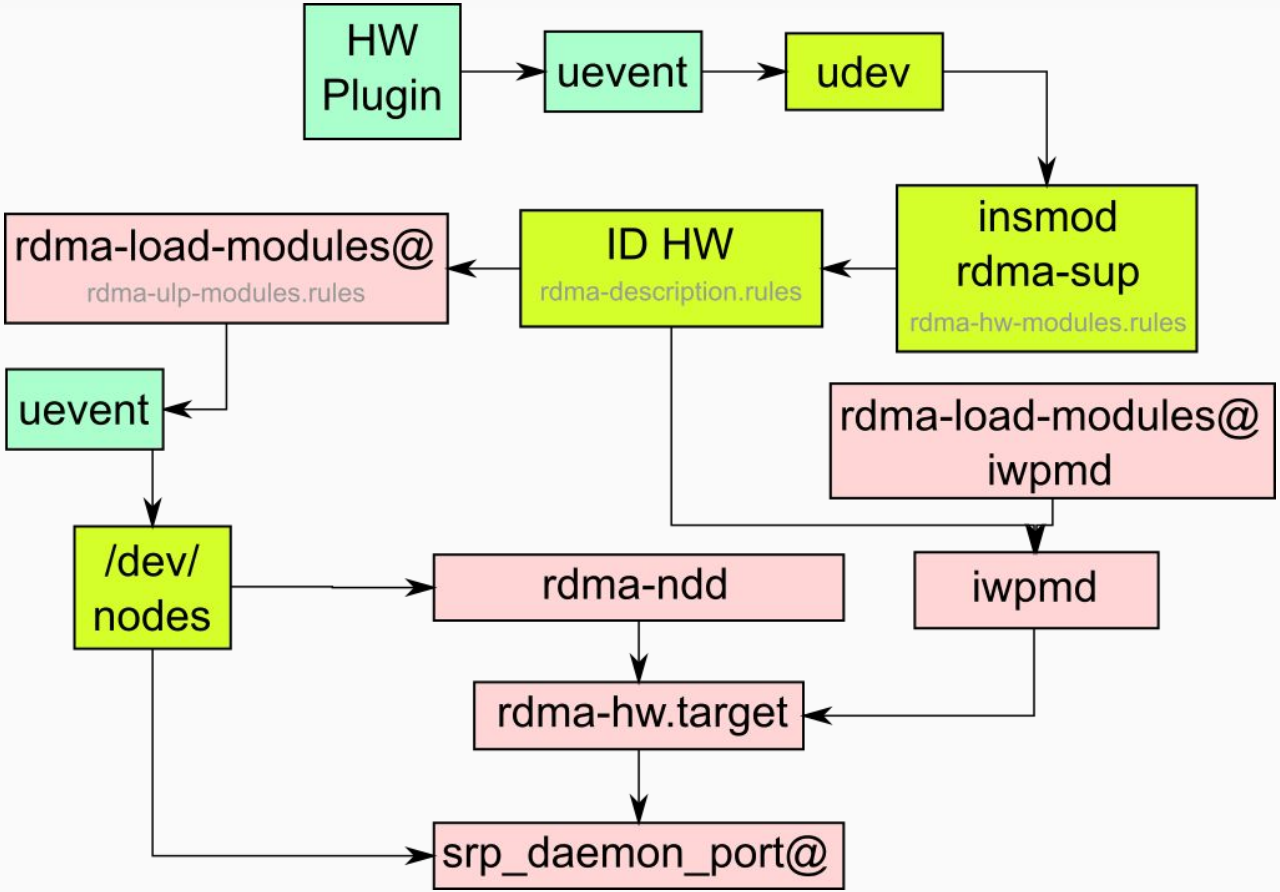
Jason Gunthorpe

New Scheme

rdma-core 15 includes a new udev & systemd based approach:

- Completely hot plug safe
- Hot unplug stops excess daemons
- Socket activation in ibacm
- ‘scriptless’
- Systemd Device Binding
- Fine grained module loading

Boot Flow Chart, rdma-core v15



Changes outside rdma-core

- RedHat specific *rdma.service* is now mostly *rdma-hw.target*, but ordering is a little bit looser
- '*udev settle*' is no longer part of the boot. It can be added by the admin if necessary via *systemd-udev-settle.service*
- To guarantee hot plug ordering daemons must **Requires** or **BindsTo** a RDMA device name, eg
  - /dev/infiniband/umad0
  - sys-subsystem-rdma-devices-mlx4:0-umad.device
- Distros need hot-plug compatible network scripts for IPoIB devices eg Network Manager or systemd-networkd

More to do

- Allow **Requires**/**BindsTo** to specify a port GUID eg */dev/infiniband/by-port-guid/.../umad, uverbs, issm*
- Have the kernel tell user space what the link technology is, instead of guessing in the udev rules
- Report RDMA driver from the kernel
- Kernel Autoload more modules
- Autoload rxe modules, and sane configuration approach for enabling *systemd-networkd*?

Next Steps

udev **IMPORT{program}** helper:

- Use RDMA netlink to get more information from kernel
- Tell udev to create */dev/../by-port-guid/..* names
- Spread the link technology to umad files eg:
  don’t start *srp_daemon* except on IB
- Make the rest of the Red Hat contributed scripts common

RDMA device load latency

Some drivers (eg mlx4) take up to 7 seconds to load their RDMA component

- If loading is triggered by udev (eg no-initrd) then it is totally async to bootup
- RDMA device appears *after* the systemd presents a login prompt
- If loaded in the initrd then boot delays 7 seconds, but the main system sees the RDMA device immediately
- New approach handles both cases, but can be surprising to users

Kernel Module Autoloading

RDMA has always lacked autoloading, need to fix it.

rdma-netlink now autoloads properly.

Suggestion: Use rdma-netlink to autoload uAPI modules

eg libibverbs does a RDMA netlink query to get the uverbs name and major:minor - the query triggers loading the module using netlink autoloading.

Kernel Server Module Autoloading

This are: iSER, SRP Target, NFS Server, nvm-fabrics server?

How?

Can we autoload SCSI target transports somehow?

What about NFS?

Device Renaming

aka predictable or persistent device names

- Provide kernel facility to rename RDMA devices
- Have udev invoke kernel facility on hot add event to set RDMA device name

Basically copy netdev approach, naming policy lives in user space.

Reliance on ‘name’

- RDMA relies on the name for many operations, eg *libibverbs* and *libumad* heavily uses */sys/class/infiniband/XXX/*
- Thus ordering during bootup is more complex and racy. Device can rename during/after *ibv_get_device_list*(). Still want everything to work right.
- RDMA netlink introduces a ‘RDMA device index’ similar to netdev’s if_index

What Name To Use?

- Historically meaningless names like ‘mlx4_0’ for the first Mellanox CX4 class device. Like the bad days of WiFi
- Many devices are now one port/one PCI function, so naming is very important to identify the physical port on the card
  Driver reports physical port label on the card?
- RoCE and iWarp devices are linked to a physical ethernet device.
  If single port, *re-use the netdev name for the port in RDMA ?*
- Otherwise copy approach now used by ethernet? *‘rdmas0’* ‘rdmap2s3’?

How To Implement

- Add device renaming RDMA netlink operation to kernel
- Update *libibverbs*/*libumad* to use RDMA netlink and RDMA device index - never sysfs
- Use a **RUN{program}** udev helper to issue the rename at hot add time
- Consider merging udev stuff into systemd as a udev builtin

IPoIB/VNIC Naming

Automatically created netdevs, but associated with single RDMA devices.

- Today *ipoib0*
- Perhaps *ipoibo0_p1*, port 1 of device *rdmao0 ?*
- Can be done in userspace by udev if we can decide on a format

‘Link-Down’ bootup

RDMA has a mixture:

- IB and OPA devices boot with the link up on driver load
- RoCE and iWarp devices follow their master ethernet link and boot with the link down

Link down is prefered. It gives userspace a chance to setup the device, start listening sockets, enable ibacm, etc. Necessary to eliminate boot races.

How can we change the IB devices over? What brings the link up?

Thank You!