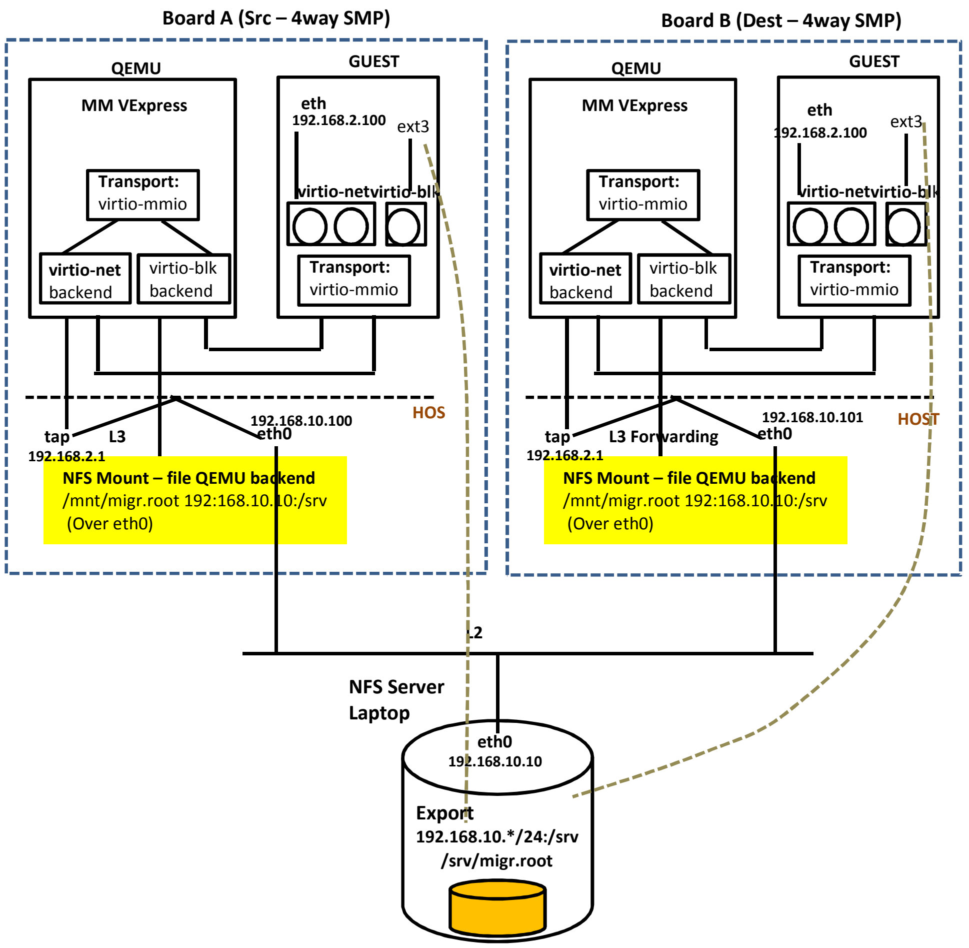
8/7/2014
Mario Smarduch
Samsung OSG
Revised: 10/10/2014 for ARMv8

ARMv7 Dirty Page Log Test – validation through VM migration



1) Routing

Routing Tables on both Guests:
default 192.168.2.1
192.168.2.0/24 eth0

Routing Tables on Gauss A	Routing Tables on Gauss B	Routing Tables on NFS Server
default 192.168.10.10	default 192.168.10.10	host route: 192.168.2.100 via 192.168.10.100
192.168.2.0/24 tap	192.168.2.0/24 tap	192.168.10.10/24 eth0
192.168.10.0/24 eth0	192.168.10.0/24 eth0	

2) Additional Software

NFS Client – rpcbind, statd to mount NFS folder: mount.nfs –w 192.168.10.10:/srv /mnt

3) Disk Image preparation

Normal file created with 'dd', mkfs.ext3 on empty file, mount via loop, and copy contents of File System to mount. Used Filesystem from Linaro

4) Command on Target, Source

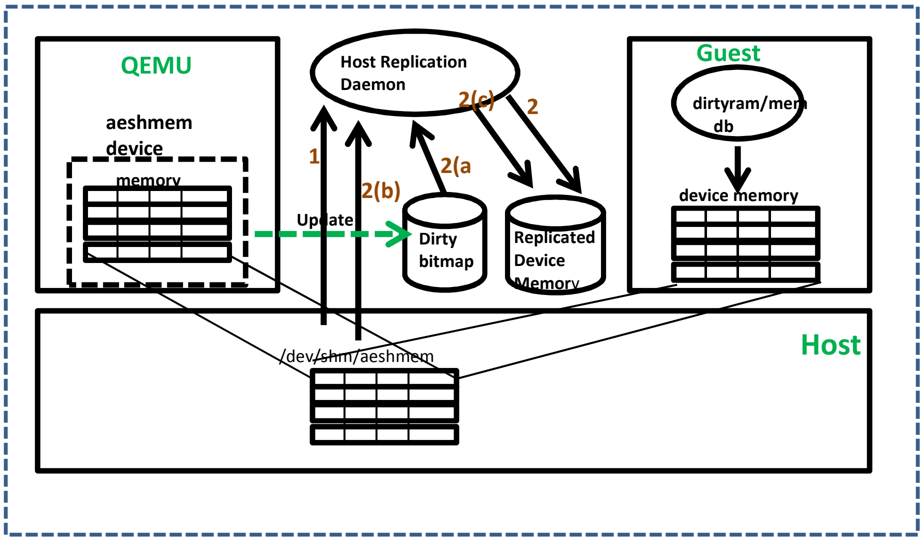
<u>Target Board</u>

# Allocates 409600 * 4k pages, dirty 2048 pages every 50 ms in steps of 409600/2048, fastest dirty rate achieved 20mS over 1Gbps link
- run ./dirtyram 409600 2048 50 &
# Print echo to console every one second interval
- run ./writeout.sh
- Break into monitor
- migrate –d tcp:192.168.10.101:4321
- info migrate – wait for completion

At this point Guest on Board B – will have full access to disk image, it's network connectivity limited to 192.168.2.0/24 subnet and 192.168.10.101 host address. Can't reach NFS Server or anything outside provided NFS Server is connected to outside. Switch host routes on NFS Server to 192.168.2.100 via 192.168.10.101.

The configuration is limited, due to one Ethernet interface.

ARMv8 Dirty Page Log Test – validation through emulated device ram replication



Operation

- QEMU implements a device with memory backed by shared memory segment accessible from host
- Guest discovers/maps device memory (similar to ivshmem)
- Host replication process continuously replicates guest device memory to regular file or could be over socket to remote host.

Referencing Above

1, 2: Replication daemon 'pre copies' guest device memory to file

2(a): Replication daemon requests dirty bitmap from QEMU, it writes it to host file (green arrow)

2(b)(c): scans dirty bitmap copies dirty pages from device memory to host file, sleeps, returns to 2(a)

Testing:

1. Launch Guest – reference command any other way to bring guest up is fine. As long as you don't map anything into range 0x20000000 – 0x20100000
   qemu-system-aarch64 –enable-kvm –nographic \
   –smp 2 –kernel Image.guest –m 512 \
   -initrd. Initrd.ext3 –append "rdinit=/sbin/init console=ttyAMA0 mem=512M root=/dev/ram"

2. Create few files: > /home/root/dest_aeshmem; > /home/root/dirty_bitmap
3. On guest run 'tstaeshm dirty 0x20000000' – test randomly updates pages in 1MB range. File '/dev/shm/aeshmem' size should be 1MB.
4. On host run 'replicatev8' – reads dirty bitmap, copies associated pages from guest device memory to host file. As the test runs you can checksum both files to see their contents differ
5. After couple minutes stop 'tstaeshm', 'replicatv8' should exit.
6. sum –r /dev/shm/aeshmem; sum –r /home/root/dest_aeshmem – should be identical