Development on an Apple Silicon Mac and RHEL/Rocky Linux 8.x

Intro

Lustre development on ARM machines is possible but requires some more care on an Apple Silicon Mac due to some kernel restrictions. Most out of the box EL8 aarch64 versions assume a 64K page size which is not supported on Apple Silicon. 4k and 16k are supported. As result a install ISO file loops in Grub since the kernel cannot boot. This is fixed for EL9+ aarch64 versions and do not require this process.

This document walks through the steps required to run EL8 virtualized on an Apple Silicon Mac.

Process

Virtualization is not possible. Therefore, we must emulate a VM to build a kernel with a compatible page size. Due to emulation, this is very slow. Here, we use UTM with QEMU. This walk-through uses Rocky 8.7 aarch64 as an example.

Setup

1. Install UTM and create a new VM with the Rocky iso. This loops in Grub due to the above restriction.
2. Stop the VM. Right-click on VM `Edit->QEMU` and untick `Use Hypervisor` to enable emulation. In the same settings window, change `Display ->Emulated Display Card` from `virtio-gpu-pci` to `virtio-ramfb` otherwise there is not display output.
3. Boot and install OS as usual.

New kernel

1. First install some build dependencies as root:

```
dnf -y groupinstall 'C Development Tools and Libraries'
dnf -y groupinstall 'Development Tools'
dnf -y install ncurses-devel openssl-devel elfutils-libelf-devel python3
dnf config-manager --set-enabled powertools
dnf -y install dwarves wget
```

2. Grab a kernel and unpack:

```
wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.18.tar.gz
tar xf linux-4.18.tar.gz
```

3. Config kernel and use booted kernel config as a base

```
cd linux-4.18/
make  O=~/build/kernel mrproper
cp /boot/config-`uname -r` ~/build/kernel/.config
```

4. Change kernel page size. With, e.g., `make menuconfig` search `CONFIG_ARM64_64K_PAGES` and change it to `4K` .

```
make O=~/build/kernel menuconfig
sed -ri '/CONFIG_SYSTEM_TRUSTED_KEYS/s/=.+/=""/g' ~/build/kernel/.config
grep PAGES ~/build/kernel/.config # sanity check
```

5. (optional) set target name

```
vim Makefile # to change `EXTRAVERSION` kernel name suffix
sed  -i 's/^EXTRAVERSION.*/EXTRAVERSION = -4Kpages/'  Makefile # alternative
make O=~/build/kernel kernelversion # check kernel version target
```

6. Build the kernel (this takes a while due to emulation...)

```
make -j 4 O=~/build/kernel
make O=~/build/kernel modules_install
```

7. Install kernel

```
cp ~/build/kernel/arch/arm64/boot/Image /boot/vmlinuz-4.18.0-4Kpages
cp -v ~/build/kernel/System.map /boot/System.map-4.18.0-4Kpages
kernel-install add 4.18.0-4Kpages /boot/vmlinuz-4.18.0-4Kpages
```

8. Reboot into the new kernel. If successful, shutdown the VM.
9. Re-enable the `Use Hypervisor` option and change the `Emulated Display Card` back to `virtio-gpu-pci` to enable virtualization.
10. Refer to this guide for building Lustre MASTER with Rocky 8.7. Double check there is enough space available in the VM.

(Optional) Migrate VM from UTM to VMware Fusion Pro

VMware Fusion Pro does not support emulation and therefore the used Rocky image cannot boot without building a kernel with a compatible page size first. Moreover, even when migrating to VMware, the stock kernel is incompatible. So, it is important to first build the Lustre kernel.

Note, Oracle Linux 8.7 is the only flavor that can boot out of the box with VMware Fusion. Rocky/Alma/CentOS do not boot without a compatible kernel.

The following documents the steps for a successful migration:

1. Make sure to have build a Lustre kernel before continuing. (Other kernel configurations probably work too except the stock kernel).
2. In UTM first run `dracut --force --no-hostonly` which rebuilds the initramfs with all drivers available. This is important to support the storage bus type in VMware. Later (when in VMware), `dracut --force` rebuilds initramfs only with the needed drivers.
3. Shutdown the VM and convert the qcow2 VM file to vmdk via `qemu-img` . If command not available, install qemu (e.g., via brew: `brew install qemu`). This command creates a new vmdk file, leaving qcow2 untouched.:

```
# First, navigate to the UTM VM directory
qemu-img convert -p -f qcow2 -O vmdk <VM_file_name>.utm/Data/*.qcow2  ~/Downloads/rocky8-7.vmdk
```

4. Add VM in VMware Fusion and use `Create a custom virtual machine` , choose `Use an existing virtual disk` .
5. Boot into the existing Lustre kernel.
6. Final steps:
    a. Fix swap: `swapon /dev/mapper/rl-swap --fixpgsz`
    b. The NIC has changed, so the old config file no longer applies: Navigate to `/etc/sysconfig/network-scripts/` , change the file name and the config, i.e., the `NAME` and `DEVICE` fields.
    c. Set a new hostname
    d. Regenerate hostkeys `cd /etc/ssh && rm ssh_host_* && ssh-keygen -A`
    e. (optional) nvme drivers on EL8 seem to be buggy regarding low power states. This results in an unresponsive machine for 30 seconds. syslog reports `nvme nvme0: I/O 123 QID 2 timeout, aborting; nvme nvme0: Abort status: 0x0`. This can be avoided by changing the disk interface to SATA in the VM disk settings.
7. Reboot and done.