

Building Lustre - in Docker Experiences on the Path @ HU Berlin

Detlev Conrad Mielczarek

CMS Humboldt Universität zu Berlin

30 Sept - 1 Oct 2026 Lustre Admin & Developer Workshop, Paris

The Initial Position:

- ▶ Return to centralised HPC services in 2024
- ▶ HPC Team formed after hardware acquired
- ▶ Intentions of internal compute cloud support (Didn't happen)
- ▶ Dell sold a lustre system:
With setup - but self administration

Hardware:

- ▶ 1× Management server
- ▶ 2× Metadata servers
- ▶ 4× Storage servers
- ▶ 2× ME5084 PowerVault with 68× 16 TB HDDs
- ▶ ≈ 1.6 PB of storage space
- ▶ 100 GBit/s Ethernet (no special adjustments)



We want Features & Updates

Issues:

- ▶ Contractor provisioned Lustre 2.15.x, Rocky 8.10
- ▶ Dynamic Nodemaps available in 2.16.0
- ▶ Rocky 8: old kernel, new hardware, performance?

Solution:

- ▶ Build our own test bed (virtualised)
- ▶ Build lustre (& document the process)
- ▶ Update
- ▶ Share it with the world

Virtual Machines and Lustre

For administering lustre, a test bed would be good.

- ▶ At the time, Whamcloud Wiki supported CentOS 7 and older
wiki.whamcloud.com/display/PUB/Building+Lustre+from+Source
(Since updated)
- ▶ How to build for Rocky 8 ? - Rocky 9 ?
- ▶ The wiki mentioned VMs, so start in a VM
build, analyse errors, tweak, until it works

Virtual Machines and Lustre

For administering lustre, a test bed would be good.

- ▶ At the time, Whamcloud Wiki supported CentOS 7 and older
wiki.whamcloud.com/display/PUB/Building+Lustre+from+Source
(Since updated)
- ▶ How to build for Rocky 8 ? - Rocky 9 ?
- ▶ The wiki mentioned VMs, so start in a VM
build, analyse errors, tweak, until it works

Result:

- ▶ It works including the patched kernel - lets share this:
github.com/DetlevCM/lustre-build-instructions/tree/main
- ▶ Can we integrate this into Lustre ? → DWG
jira.whamcloud.com/browse/LUDOC-553

Virtual Machines and Lustre

Building, testing Lustre in VMs is very easy & straightforward.

- ▶ standard test kvm-VM:
 - ▶ 8 threads
 - ▶ 100GB (shared home/root)
 - ▶ 16GB RAM
- ▶ Lustre includes a local test `lustre/tests/llmount.sh`
- ▶ Can also use virtual disks for distinct MDTs/OSTs

Virtual Machines and Lustre

Building, testing Lustre in VMs is very easy & straightforward.

- ▶ standard test kvm-VM:
 - ▶ 8 threads
 - ▶ 100GB (shared home/root)
 - ▶ 16GB RAM
- ▶ Lustre includes a local test `lustre/tests/llmount.sh`
- ▶ Can also use virtual disks for distinct MDTs/OSTs

We want to test with multiple machines:

- ▶ Proxmox !
 - ▶ Build a virtual Lustre architecture
 - ▶ Works just as well as a single VM
 - ▶ Permits larger scale testing

Virtual Machines and Lustre

Building, testing Lustre in VMs is very easy & straightforward.

- ▶ standard test kvm-VM:
 - ▶ 8 threads
 - ▶ 100GB (shared home/root)
 - ▶ 16GB RAM
- ▶ Lustre includes a local test `lustre/tests/llmount.sh`
- ▶ Can also use virtual disks for distinct MDTs/OSTs

We want to test with multiple machines:

- ▶ Proxmox !
 - ▶ Build a virtual Lustre architecture
 - ▶ Works just as well as a single VM
 - ▶ Permits larger scale testing

Following a DWG-call, what about patchless Lustre?

Virtual Machines and Lustre

Patchless Lustre:

- ▶ Mentioned by the lustre devs in a DWG-call.
- ▶ Builds successfully, runs
- ▶ In operational use with Rocky 9.x at the HU
- ▶ Made it part of the wiki:
[wiki.lustre.org/Compiling_Lustre#Rocky_9.4/9.6_\(RHEL_compatible\)](https://wiki.lustre.org/Compiling_Lustre#Rocky_9.4/9.6_(RHEL_compatible))

Building in Docker

If it builds patchless, can we build in a container?

- ▶ Yes we can !
- ▶ Scripted build process
- ▶ Platform independent
- ▶ Easily accessible environment for testing, development
- ▶ Employed to build updated binaries for our system

Building in Docker

If it builds patchless, can we build in a container?

- ▶ Yes we can !
- ▶ Scripted build process
- ▶ Platform independent
- ▶ Easily accessible environment for testing, development
- ▶ Employed to build updated binaries for our system

Difference to docker files in Lustre Source Tree?

- ▶ Focus on automatic rpm generation
Client, on rocky server too
- ▶ Use of ldiskfs, no zfs

Building in Docker

Design Considerations:

- ▶ Automate as much as possible, but allow options & adaptability
- ▶ Try to keep the complexity low
 - ▶ Two files, two commands
 - ▶ But optional parameters
- ▶ Docker provides the underlay
 - < 50 lines of active docker definitions
- ▶ The build itself runs in bash
 - < 200 lines of active bash code

```

101 fl
102
103 # also get as set the version from the container
104 linux=$(s /usr/src/kernels/) # get distro version from container
105 fl
106
107 # build and install a2fagsrc if we want to build the server
108 if [ "$BuildServer" == "true" ];
109 then
110     Buildpath=/build/a2fagsrc-$version2fack-$versionLinux
111     echo $Buildpath
112     cp -r $a2fagsrcSourceRepo /build/a2fagsrc-$version2fack-$versionLinux
113     cd $Buildpath
114     # build a2fagsrc
115     git checkout $version2fack
116     ./configure --enable-a2facklib
117     # these two tests fail
118     cd $Buildpath/tests
119     # d_print_acl_d_print_acl
120     # m_repair_acl_m_repair_acl
121     cd $Buildpath
122     make rpm
123
124 # now install them, we're lazy, install all!
125 and install -p /root/rpmbuild/RPMS/x86_64/*.rpm
126 mkdir -p $Buildpath-rpm
127 # /root/rpmbuild/RPMS/x86_64/*.rpm $Buildpath-rpm/
128 fl
129
130 Buildpath=/build/lustre-release-$versionLustre-$versionLinux
131 echo $Buildpath
132 cp -r $LustreSourceRepo /build/lustre-release-$versionLustre-$versionLinux
133 cd $Buildpath
134 git checkout $versionLustre
135 # disable the kernel check - we build in a container
136 sed -i "s/BuildRequires: kernel >= 3.08/BuildRequires: kernel >= 3.08/g"
137 $Buildpath/lustre.spec.in
138
139 ./configure.sh
140 if [ "$BuildServer" == "true" ];
141 then
142     # for the server support build we need to disable another check
143     sed -i "s/././g" -e "s/define(\\(\\)space\\(\\)\\)HAVE_SERVER_SUPPORT config.h 2- /"
144     ./configure --with-lustre=$Buildpath/lustre.spec.in
145     ./configure --enable-server --with-linux=/usr/src/kernels/$versionLinux
146 else
147     ./configure --with-linux=/usr/src/kernels/$versionLinux
148 fl
149
150 # make regular rpms
151 make rpms
152 mkdir -p $Buildpath-rpm
153 mv *.rpm $Buildpath-rpm/
154
155 # make dms rpms
156 make dms-rpm
157 mkdir -p $Buildpath-dms-rpm
158 mv *.rpm $Buildpath-dms-rpm/
159
160 # make the folder read-and-writable to all
161 chmod -R a+rx $Buildpath
162 chmod -R a+rw $Buildpath-rpm
163 chmod -R a+rw $Buildpath-dms-rpm
164

```

Building in Docker

Two commands to build lustre for current Rocky 9:

- ▶ `docker build . \`
`-f lustre_builder_for_rocky9 \`
`-t lustre_builder_for_rocky9`
- ▶ `docker run --volume ./build:/build \`
`-e BuilderOptions=" \`
`--versionLustre=2.17.0 server \`
`--versionLinux=5.14.0-570.32.1.el9_6.x86_64" \`
`lustre_builder_for_rocky9`

Builds packages, provides rpms, retains kernel rpms.

```

lustre_builder_rocky_9: bash — Konsole
+ cd /tmp/rpmbuild-lustre--oaoGRn4w/BUILD
+ rm -rf lustre-2.17.0.dirty
+ RPM_EC=0
++ jobs -p
+ exit 0
detlevcm@detlevcm-Tuxedo:~/tmp/lustre_builder_rocky_9> ls b
uild/
build-rocky_bash
e2fsprogs-vi.47.3-wci-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-abi-stablelists-5.14.0-570.32.1.el9_6.noarch.rpm
kernel-core-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-debuginfo-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-debuginfo-common-x86_64-5.14.0-570.32.1.el9_6.x86_64
.rpm
kernel-daveel-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-modules-5.14.0-570.32.1.el9_6.x86_64.rpm
kernel-modules-core-5.14.0-570.32.1.el9_6.x86_64.rpm
lustre-release-2.17.0-5.14.0-570.32.1.el9_6.x86_64
lustre-release-2.17.0-5.14.0-570.32.1.el9_6.x86_64-debuginfo
lustre-release-2.17.0-5.14.0-570.32.1.el9_6.x86_64.rpm
lustre-release.src
detlevcm@detlevcm-Tuxedo:~/tmp/lustre_builder_rocky_9> ls b
uild/lustre-release-2.17.0-5.14.0-570.32.1.el9_6.x86_64-rpm
kmod-lustre-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-debuginfo-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-devel-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-osd-ldisksfs-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-osd-ldisksfs-debuginfo-2.17.0.dirty-1.el9.x86_64
.rpm
kmod-lustre-osd-wbcfs-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-osd-wbcfs-debuginfo-2.17.0.dirty-1.el9.x86_64.r
pm
kmod-lustre-tests-2.17.0.dirty-1.el9.x86_64.rpm
kmod-lustre-tests-debuginfo-2.17.0.dirty-1.el9.x86_64.rpm
lustre-2.17.0.dirty-1.el9.x86_64.rpm
lustre-2.17.0.dirty-1.src.rpm
lustre-debuginfo-2.17.0.dirty-1.el9.x86_64.rpm
lustre-debugsources-2.17.0.dirty-1.el9.x86_64.rpm
lustre-devel-2.17.0.dirty-1.el9.x86_64.rpm
lustre-lokit-2.17.0.dirty-1.el9.x86_64.rpm
lustre-osd-ldisksfs-mount-2.17.0.dirty-1.el9.x86_64.rpm
lustre-osd-ldisksfs-mount-debuginfo-2.17.0.dirty-1.el9.x86_6
4.rpm
lustre-osd-wbcfs-mount-2.17.0.dirty-1.el9.x86_64.rpm
lustre-osd-wbcfs-mount-debuginfo-2.17.0.dirty-1.el9.x86_64.
rpm
lustre-resource-agents-2.17.0.dirty-1.el9.x86_64.rpm
lustre-tests-2.17.0.dirty-1.el9.x86_64.rpm
lustre-tests-debuginfo-2.17.0.dirty-1.el9.x86_64.rpm
detlevcm@detlevcm-Tuxedo:~/tmp/lustre_builder_rocky_9>

```

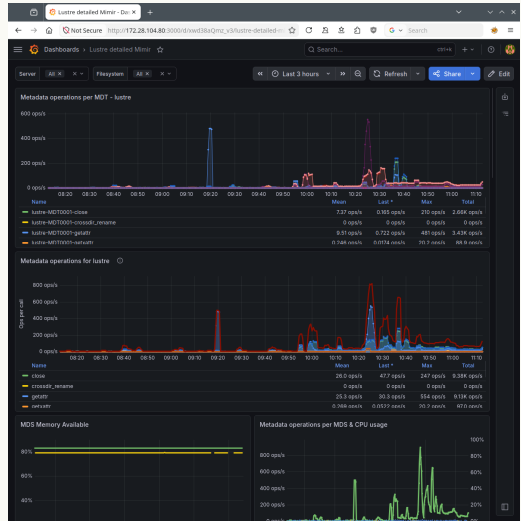
- ▶ Codes, Scripts are publicly available
 - ▶ github.com/DetlevCM/lustre-build-instructions/tree/main/build_in_docker
- ▶ Optimization potential with respect to
 - ▶ build structure
 - ▶ permissions
 - ▶ error handling
- ▶ Maybe integrate it with lustre ?
 - ▶ jira.whamcloud.com/browse/LU-20181

Lustre Experiences

- ▶ Lustre stability itself has been very good
(Including accidental reboots, server crashes)
- ▶ Updates were pretty painless
Update/Reinstall operating system, remount, works
- ▶ Reconfiguration of the OST and MDT volumes successful,
including data migration
- ▶ Lustre documentation is overall very good, but dispersed
- ▶ Some issues/difficulties with broken file links
(Removable with `lfs rm_entry`)
- ▶ We ran into a HybridIO bug (on 2.17.0)
solution: patch client, or turn HybridIO off

Lustre Experiences

- ▶ Monitoring with lustrefs_exporter
 - ▶ Prometheus
 - ▶ Grafana
 - ▶ Testing Mimir
 - ▶ Easy to set up dashboards required updating
- ▶ runs on the Management Server
 - ▶ Uses otherwise near-idle resource
 - ▶ Also serves software (nfs)



Thank you for listening !

Any questions?