OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

RDMA Core Community Collaboration

Jason Gunthorpe, CTO

Obsidian Research Corp

[March, 2017]

Introduction

- rdma-core is the new way ‘upstream’ is distributing the user space portion of the Linux kernel stack

- The same team is maintaining the user side and the kernel side, for greater consistency

- Doug Ledford is the lead maintainer, Leon Romanovsky is the 2nd maintainer, and Jason got the ball rolling

- https://github.com/linux-rdma/rdma-core

Scope

- **Purpose:**

Maintain the user space components for Linux’s ‘drivers/infiniband’

Scope

Components

- **User space libraries and tools:**

ibacm	libibcm
libibumad	libibverbs
librdmacm	srp_daemon
rdma-ndd	iwpmd

Scope

Providers

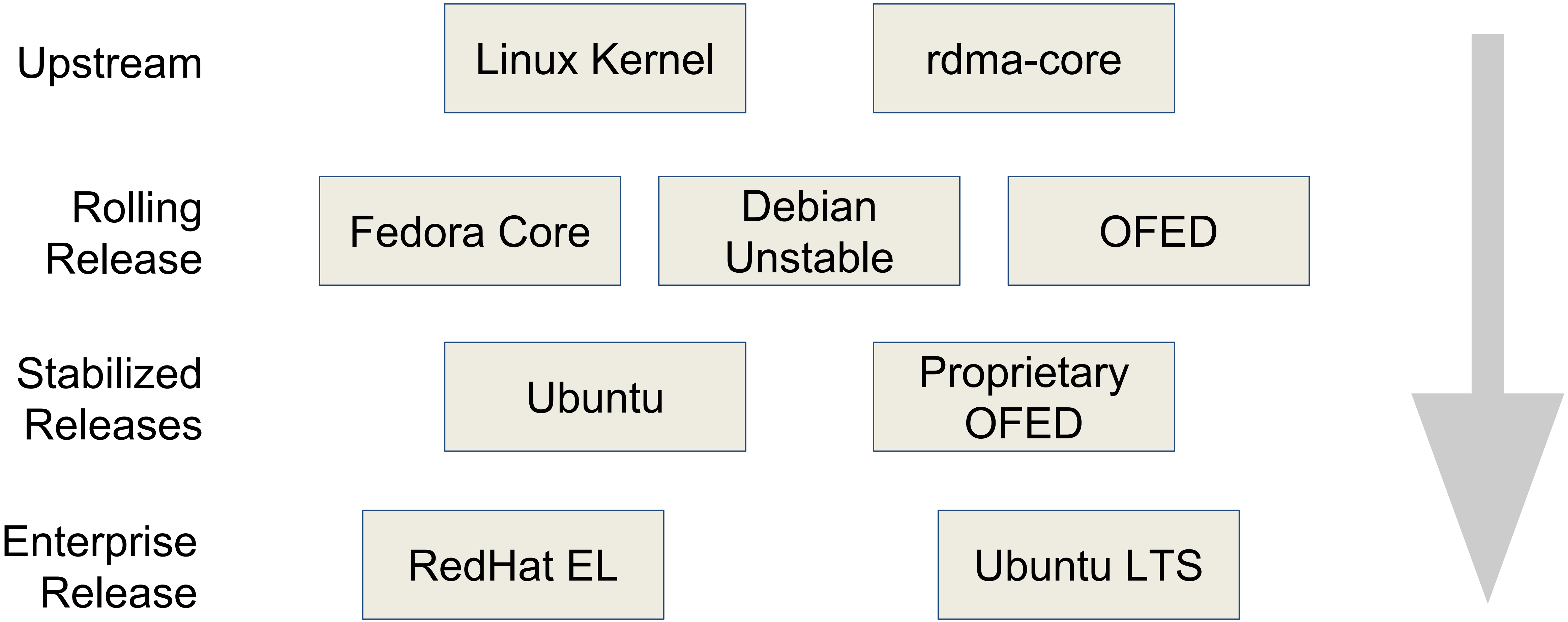
- **Verbs Providers:**

cxgb3	cxgb4
hfi1verbs	hns
i40iw	ipathverbs
mlx4	mlx5
mthca	nes
ocrdma	qedr
rxe	vmw_pvrdma

Scope

- **Not included today**
  - Single vendor libraries: usnic, psm, mxm, etc
  - Providers for obsolete drivers deleted in Linux v4.8
  - Management layer stuff: infiniband-diags, libibmad, python-rdma, opensm, perftools
- **Things must be in upstream Linux before being in rdma-core - no proprietary stuff**
- **Size**
  - 114 C files, 551 files in total
  - 117kloc

Code Flow To Users



Upstream: Linux Kernel, rdma-core

Rolling Release: Fedora Core, Debian Unstable, OFED

Stabilized Releases: Ubuntu, Proprietary OFED

Enterprise Release: RedHat EL, Ubuntu LTS

Immediate Goals

- Increase co-development of user/kernel
- Prepare for the RDMA uAPI change
- Increase and build community participation
- Simplify use, distribution, and testing
- Greater consistency across distributions
- Enable ‘All Provider’ changes
- Code quality/modernization

Progress So Far

- **Launched around August**
- **576 commits merged so far:**
  - 836 files changed,
    139,085 insertions(+), 35,659 deletions(-)
- **Release 12 and Release 13**
- **Available in Fedora Core Devel/26**
- **Included in OFED 4.8**
- **In progress for Debian/Ubuntu**

Major Changes

- **Other workshop presentations cover new APIs**
  - Verbs Direct, Timestamp, Packet Pacing
  - New Providers (hns, qedr, rxe, vmw_pvrdma)

- **Verbs provider interface is now private**
  - No support for out-of-tree libibverbs providers

Clean Up

- **Bring code to a level where distros are comfortable with it**
  - No dangerous gcc warnings, compile correctly on a wide range of architectures, remove cruft, ‘make install’ does what distros want, more rigorous rules for symbol versions, upstream patches from distros, solve ‘distro linter’ issues and copyright audit
- **Run static analysis tools on the code base, solve issues**
  - A limited sparse now runs automatically from travis
  - High warning level on gcc 6.2/clang 3.9 provide static analysis
  - coverity (run by others)
- **Provide C utility libraries (The C Code Archive Network, util/)**
  - Boring C stuff like lists, min/max, container_of. Similar to the kernel
  - Userspace DMA helpers, compiler tools
- **Single build/configure performed using cmake**

How To Participate

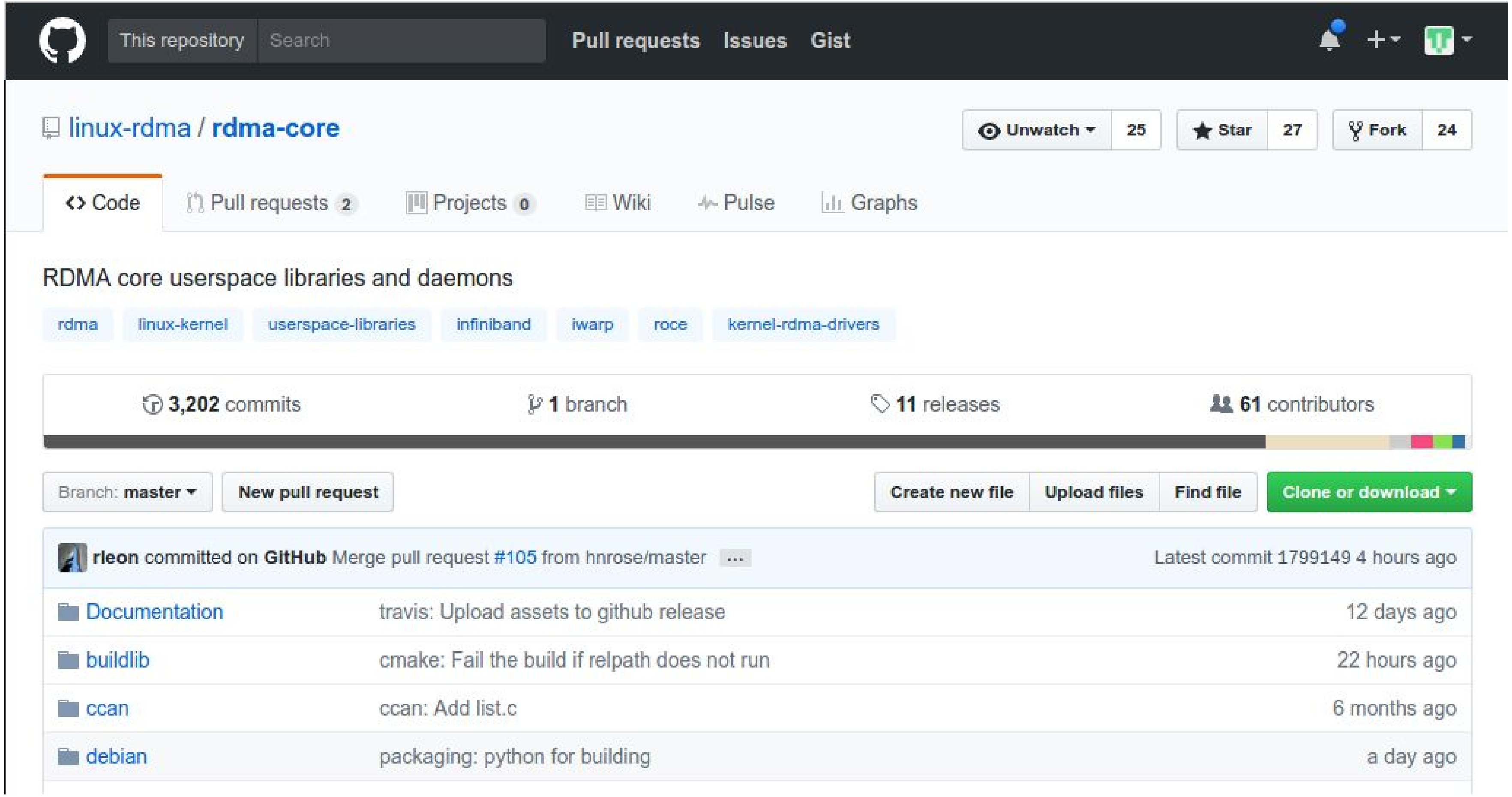
rdma-core uses GitHub for tracking patches.

It does not use the issue tracking system or the Wiki

Extensive discussion should occur on the mailing list: linux-rdma@vger.kernel.org

For significant patches git send-email to the mailing list.

https://github.com/linux-rdma/rdma-core



linux-rdma / rdma-core

RDMA core userspace libraries and daemons

rdma · linux-kernel · userspace-libraries · infiniband · iwarp · roce · kernel-rdma-drivers

3,202 commits · 1 branch · 11 releases · 61 contributors

rleon committed on GitHub Merge pull request #105 from hnrose/master … Latest commit 1799149 4 hours ago

Documentation	travis: Upload assets to github release	12 days ago
buildlib	cmake: Fail the build if relpath does not run	22 hours ago
ccan	ccan: Add list.c	6 months ago
debian	packaging: python for building	a day ago

How To Participate

Try it at home

Test rdma-core locally

Follow instructions in README.md for required packages to build

No need to wait for OFED or your distro

Can be run without disturbing your existing system installation

Use of ‘make install’ difficult and not recommended

Setup:

```
 $ git clone https://github.com/linux-rdma/rdma-core.git
 $ cd rdma-core/
 $ ./build.sh
 [175/175] Linking C shared module lib/libibacmp.so
```

Run in place:

```
 $ build/bin/ibv_devinfo
```

Run other apps:

```
 $ export LD_LIBRARY_PATH=`pwd`/build/lib/
 $ /usr/bin/….
```

How To Participate

Try it at home

Script to build packages using Docker

Fully automatic, fast, reproducible and easy to use once Docker is installed

Script will ‘cross build’ to any distro

Resulting packages can be installed for testing

One time setup, for each image type:

```
$ buildlib/cbuild build-images centos7
```

Produce RPMs for centos7

```
$ buildlib/cbuild pkg centos7
```

Or Ubuntu Xenial

```
$ buildlib/cbuild pkg xenial
```

Templates for centos6/7/7_epel, Debian Jessie/Experimental, FC25, OpenSuSE 13.2,42.1, tumbleweed, Ubuntu Trusty, Xenial

How To Participate

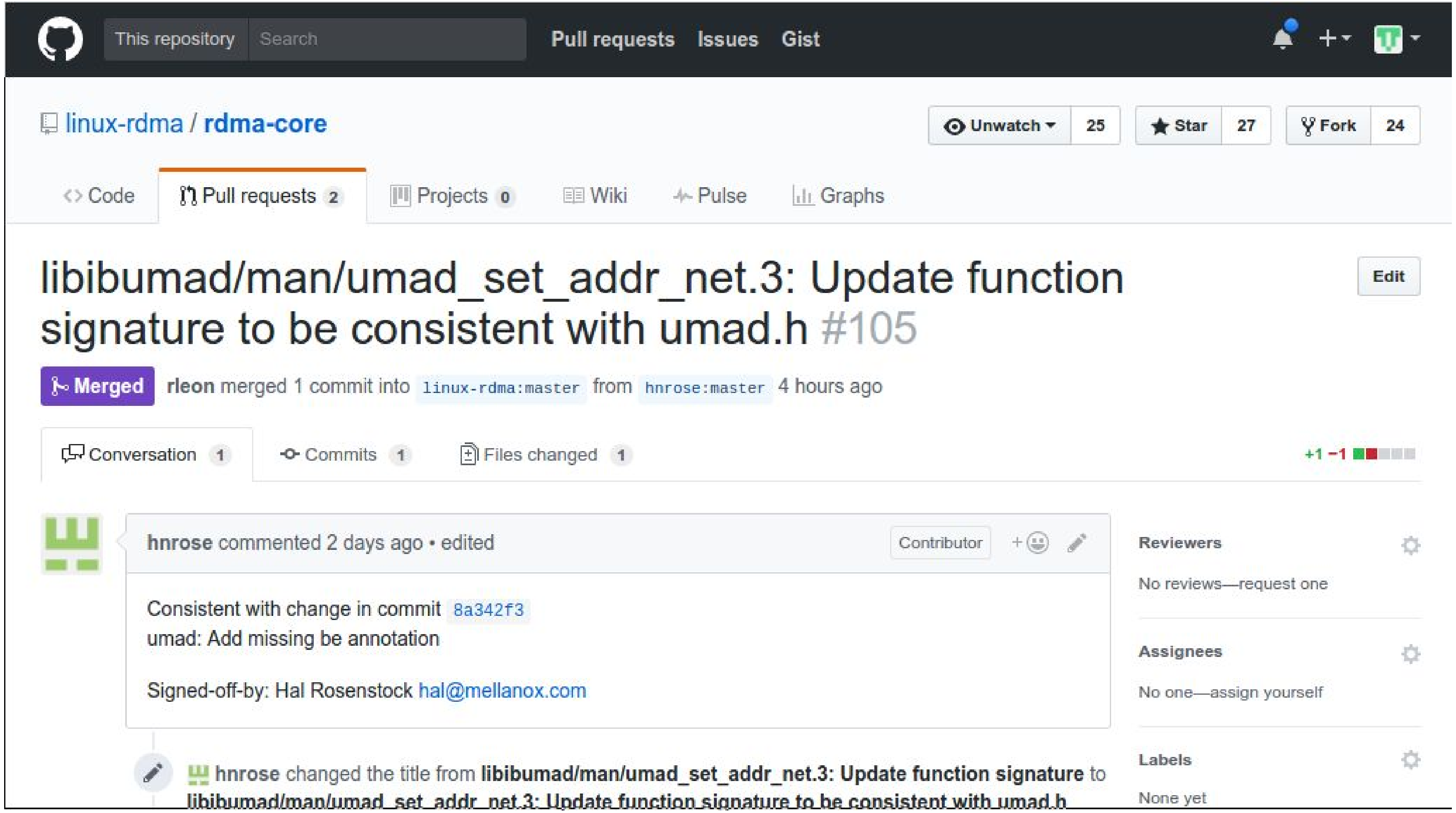
Make a change

Make a Change

Send a Pull Request

Follow GitHub instructions, fork on GitHub, make and test your change, push it to your branch, then send a PR

Force-push your branch with any feedback until the PR is merged



libibumad/man/umad_set_addr_net.3: Update function signature to be consistent with umad.h #105

Merged rleon merged 1 commit into `linux-rdma:master` from `hnrose:master` 4 hours ago

hnrose commented 2 days ago • edited

Consistent with change in commit `8a342f3`
umad: Add missing be annotation

Signed-off-by: Hal Rosenstock hal@mellanox.com

hnrose changed the title from **libibumad/man/umad_set_addr_net.3: Update function signature** to **libibumad/man/umad_set_addr_net.3: Update function signature to be consistent with umad.h**

How To Participate

Make a change

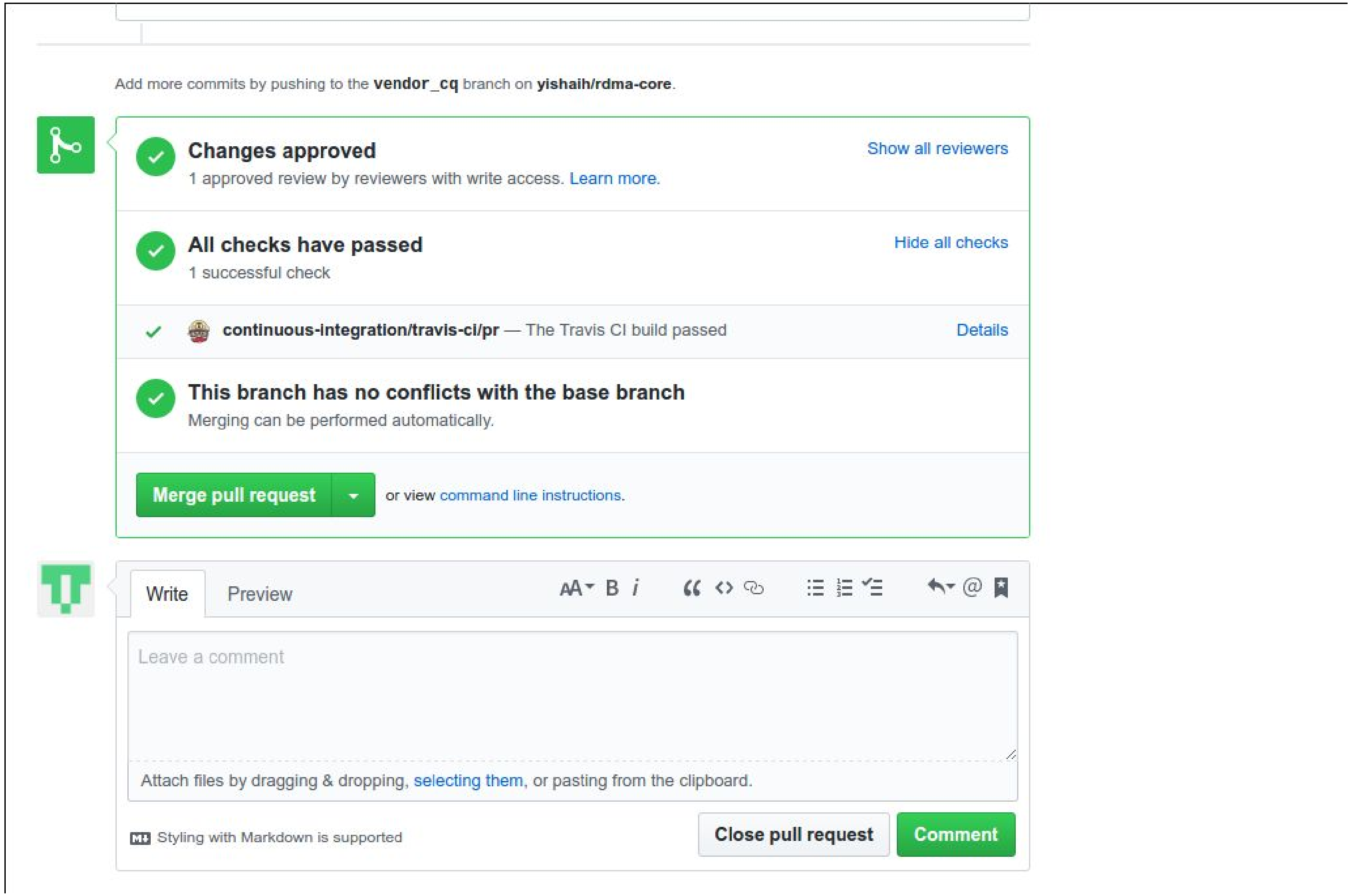
We use Travis CI

Check your PR passes automated build testing

Fix any mistakes and force-push your branch

Run Travis locally using docker via

```
buildlib/cbuild pkg travis
```



Add more commits by pushing to the **vendor_cq** branch on **yishaih/rdma-core**.

✓ **Changes approved**
1 approved review by reviewers with write access. Learn more.

✓ **All checks have passed**
1 successful check

✓ **continuous-integration/travis-ci/pr** — The Travis CI build passed

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

or view command line instructions.

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

How To Participate

Make a change

Travis is setup to run multiple build-tests:

x86-64 using gcc 6.2, clang 3.9

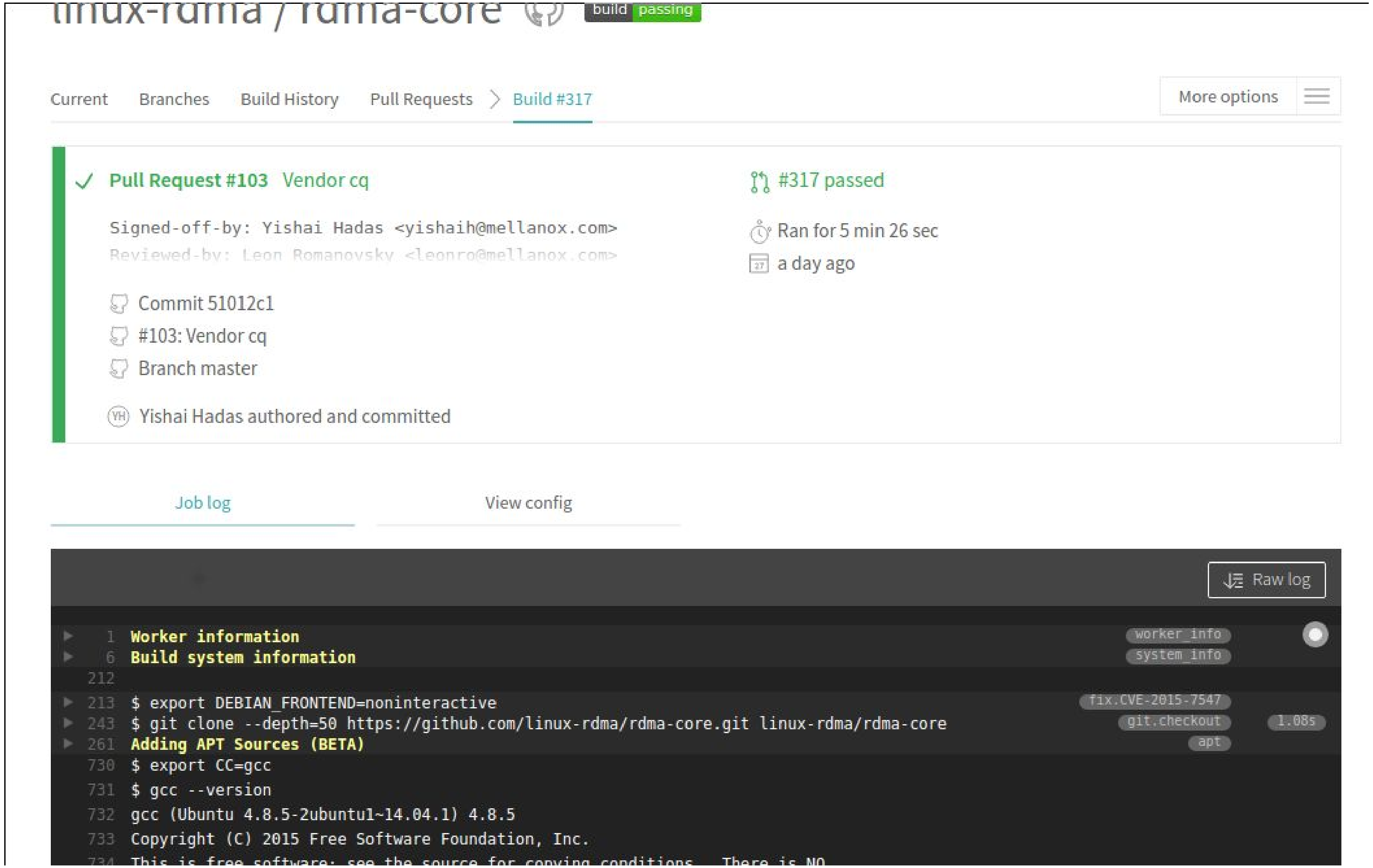
x86-32

Simulation of non-DMA platform

sparse checker

Header file checker

Debian packaging build



✓ Pull Request #103 Vendor cq

Signed-off-by: Yishai Hadas <yishaih@mellanox.com>
Reviewed-by: Leon Romanovsky <leonro@mellanox.com>

Commit 51012c1
#103: Vendor cq
Branch master

Yishai Hadas authored and committed

#317 passed
Ran for 5 min 26 sec
a day ago

```
▶ Worker information                                                    worker_info
▶ Build system information                                              system_info

▶ $ export DEBIAN_FRONTEND=noninteractive                               fix.CVE-2015-7547
▶ $ git clone --depth=50 https://github.com/linux-rdma/rdma-core.git linux-rdma/rdma-core   git.checkout   1.08s
▶ Adding APT Sources (BETA)                                             apt
  $ export CC=gcc
  $ gcc --version
  gcc (Ubuntu 4.8.5-2ubuntu1~14.04.1) 4.8.5
  Copyright (C) 2015 Free Software Foundation, Inc.
  This is free software; see the source for copying conditions.  There is NO
```

How To Participate

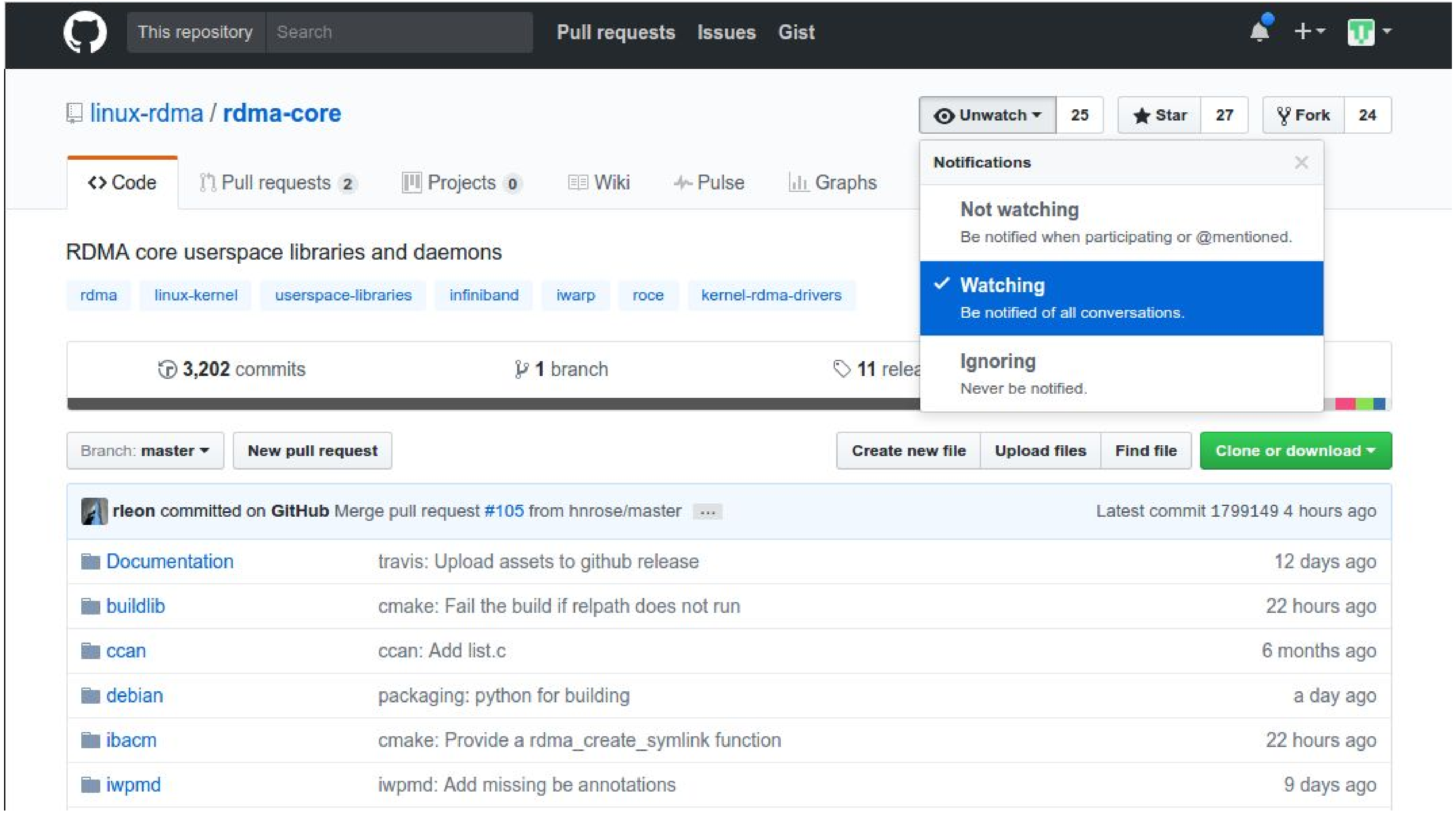
Add your voice

Subscribe to the project on Git Hub and to the mailing list

Review Pull Requests

Comment on development

Tackle an outstanding job



linux-rdma / **rdma-core**

Notifications

- Not watching — Be notified when participating or @mentioned.
- ✓ **Watching** — Be notified of all conversations.
- Ignoring — Never be notified.

RDMA core userspace libraries and daemons

rdma · linux-kernel · userspace-libraries · infiniband · iwarp · roce · kernel-rdma-drivers

3,202 commits · **1** branch · **11** relea

rleon committed on **GitHub** Merge pull request #105 from hnrose/master … Latest commit 1799149 4 hours ago

Documentation	travis: Upload assets to github release	12 days ago
buildlib	cmake: Fail the build if relpath does not run	22 hours ago
ccan	ccan: Add list.c	6 months ago
debian	packaging: python for building	a day ago
ibacm	cmake: Provide a rdma_create_symlink function	22 hours ago
iwpmd	iwpmd: Add missing be annotations	9 days ago

Future Work

systemd boot

- **Review & Move RedHat ideas to upstream**
- **Common systemd .service files for all distros**
- **socket activation for ibacmd**
- **srp_daemon: systemd integration and hotplug**
- **Eliminate ‘opensm.service’ as a dependency:**
  - Fix daemons to handle INIT->ACTIVE changes internally
  - Fix daemons to handle RDMA device hotplug internally
- **rdma-ndd is a good example of this direction**
- **GOAL: Uniform & Correct boot on all distros**

Future Work

Module autoloading

- **Make kernel module loading saner and more like other subsystems:**
  - Autoload uapi modules (ib_uverbs, rmd_ucm, etc) when a RDMA device is installed
  - Autoload the RDMA part of NET drivers (eg mlx5)

- **Currently RedHat does this via custom systemd & modprobe scripts, but it is frail and doesn’t handle hot plug well**
- **Rework modules and auto loading directly in the kernel?**

Future Work

Community Packagers

- OpenSuSE?
- Arch, Gentoo, CoreOS
- Run pre-release builds through something like the OpenSuSE build service to detect problems
- GOAL: Have all distros include rdma-core

Future Work

Kernel uAPI Headers

- **Directly use the kernel ‘include/uapi/’ headers instead of mangled copies**
- **Harder problem for verbs + providers:**

```
struct mlx4_alloc_ucontext_resp_v3 {
        struct ibv_get_context_resp     ibv_resp;
        __u32                           qp_tab_size;
        __u16                           bf_reg_size;
        __u16                           bf_regs_per_page;
};
```

```
struct mlx4_ib_alloc_ucontext_resp_v3 {
     __u32 qp_tab_size;
     __u16 bf_reg_size;
     __u16 bf_regs_per_page;
};
```

- **Make it easier to understand what our uAPI actually is**
- **Pave the way for the new uAPI**

Future Work

MMIO Accessor Macros

- Think like readl/writel in the kernel
- Common API to access a mmap'd PCI bar in user space. Uniformly use the correct methodology for each architecture
- Many bugs in existing providers in this area, lack of barriers, endian swapping, limited arch support
- GOAL: Extend the portability we see kernel side to the user components.

Future Work

Provider Detection and Loading

- All providers load all the time
- Providers duplicate much of the detection code
- .driver files do not really make much sense anymore
- Saner approach to allow vendors to ‘upgrade’ providers

Call To Action

- Participate Upstream!
- Focus testing and development on rdma-core and mainline Linux
- Validate your solutions on new upstream
- Do not expect any new releases of pre-rdma-core stuff

OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

THANK YOU

Jason Gunthorpe, CTO

Obsidian Research Corp

SAMPLE TITLE HERE

- **Sample first bullet point**
  - Sample second bullet point
    - Sample third bullet point

SECTION SLIDE SAMPLE

SAMPLE TITLE SLIDE HERE

Sample subtitle here

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut vel dignissim nisl. Donec egestas, urna a gravida varius, magna velit interdum lacus, eget vehicula enim leo et turpisLorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere.

- **Sample first bullet point**
  - Sample second bullet point
    - Sample third bullet point

SAMPLE TITLE HERE

Sample subtitle here

- **First bullet**
  - Second bullet
    - Third bullet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut vel dignissim nisl. Donec egestas, urna a gravida varius, magna velit interdum lacus, eget vehicula enim leo et turpisLorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere.