12th ANNUAL WORKSHOP 2016

CREATING A COMMON SOFTWARE VERBS IMPLEMENTATION

Dennis Dalessandro, Network Software Engineer

Intel

April 6th, 2016

AGENDA

- **Overview**
  - What is rdmavt and why bother?
  - Technical details
  - How did we get it into the kernel?

WHAT AND WHY?

PROBLEM

- **Quite simple: Code Duplication**

PROBLEM

- **ipath, qib, and hfi1 are all based on the same sw verbs**
  - Three drivers for very different hardware
  - Share the same basic software implementation **(for verbs)**
    - QP, PD, CQ, MR, etc is a software construct not involved in putting packets on the wire
    - putting packets on the wire is the job of the HW
- **Code maintenance**
  - Bug in one driver is a bug in all of them
    - has happened, will happen again
  - Improvement for one needs to be made in all
    - has happened, will happen again
    - perhaps not always ported to all drivers
- **There could be other drivers in the future**
  - other HW, other vendors (soft roce?)

- ***note: ipath driver has now been removed from the kernel***

SOLUTION

rdmavt

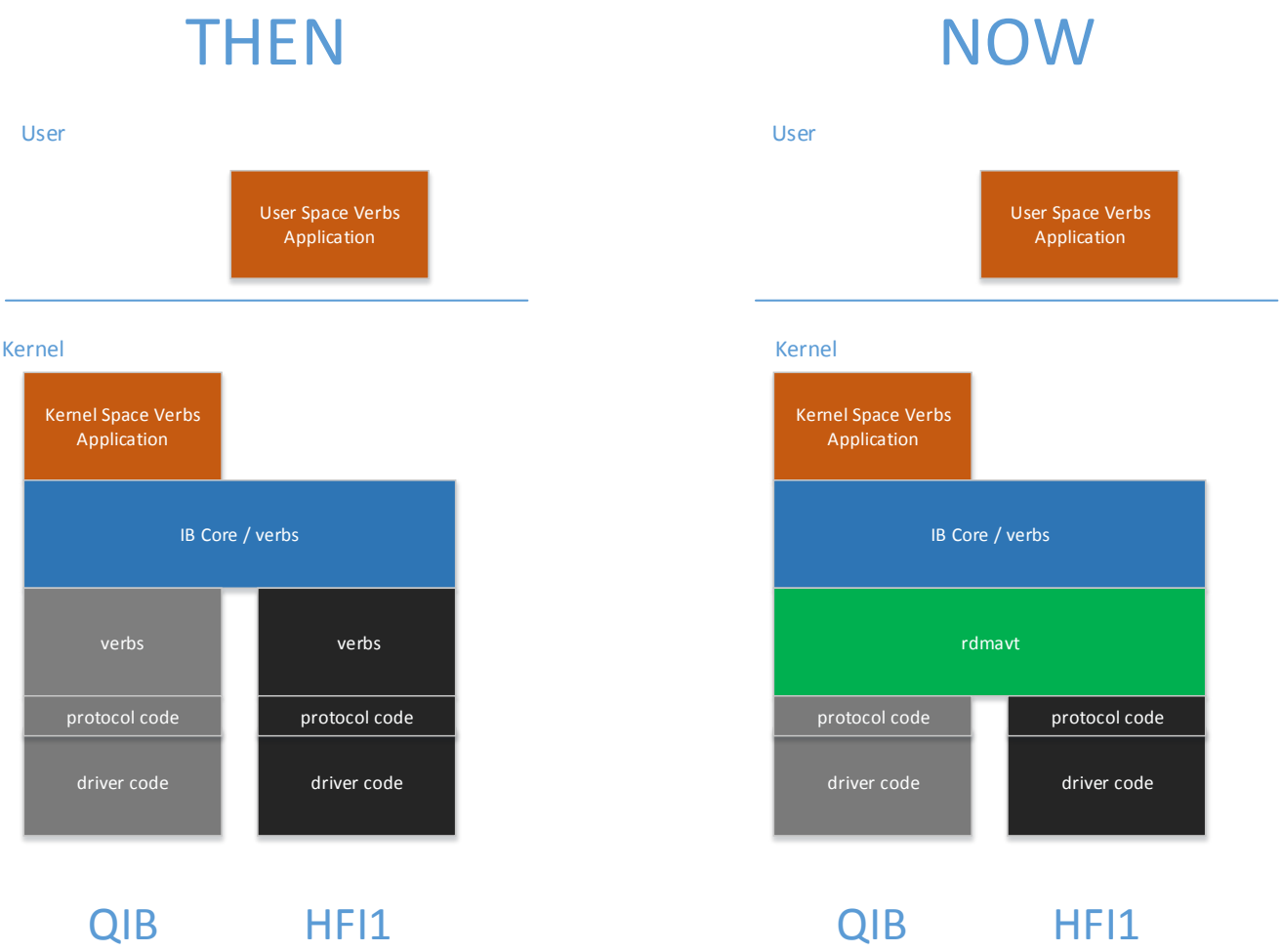
- **rdmavt**
  - Spawned out of discussion of hfi1 submission to linux-rdma
    - Requirement to move hfi1 out of staging
- **RDMA Verbs Transport Library**
  - A new kernel module which implements software verbs
  - Handles things like QP, PD, CQ, MR, etc.
  - Uses drivers like qib and hfi1 to put packets on the wire

GOALS

- **Remove verbs code duplication <u>where possible</u>**
  - These are high performance devices
    - **<u>Performance comes first!</u>**
    - At times we may need to settle for less than perfect (from sw engineering POV)
- **Can not cause regressions**
  - qib code has been stable for a number of years
- **Be able to support multiple hardware devices**
  - Do nothing that prevents other HW from working with rdmavt
  - Not do the actual work for other HW though
- **Incremental development**
  - Just get it done, make it work, improve it as we go
  - Not fire and forget
- **Develop in the open**
  - Code posted very early to GitHub, announced on linux-rdma
  - Intent is for interested parties to review and weigh-in on design

TECHNICAL OVERVIEW

RDMAVT
overview



RDMAVT
overview

- **rdmavt provides**
  - Device registration
    - Drivers register with rdmavt and not with the IB core
  - Data structures and functionality
    - QP, PD, MR, CQ, etc
  - Almost all verbs functions present in qib and hfi1 (more on this shortly)
  - Driver overrides
    - Drivers can choose to implement any of the verbs functions they choose
      - Performance
      - Incompatible hardware
  - Driver private data structures
    - Opaque to rdmavt, exp: qp_priv
  - Calls into drivers for HW specific things
  - API for drivers to call into rdmavt when needed

WHAT DOES RMDAVT NOT DO

- **rdmavt does not**
  - Move/incorporate code that would take the same number of LOC to make generic
    - exp: modify_device() would take too many call backs into drivers, not worth it
  - Handle MAD processing (yet?)
    - Very different between qib and hfi1
    - Probably some opportunities to incorporate, requires more investigation
  - Address protocol code duplication (yet?)
    - The code is very similar between qib and hfi1 in terms of high level functionality
    - Gotchas and HW specific differences
      - Very risky for the initial version
      - Performance is top priority
  - Replace IB core
    - Drivers still use some of the data structures
      - Want to limit of course
      - Why re-invent the wheel?
        - » registration

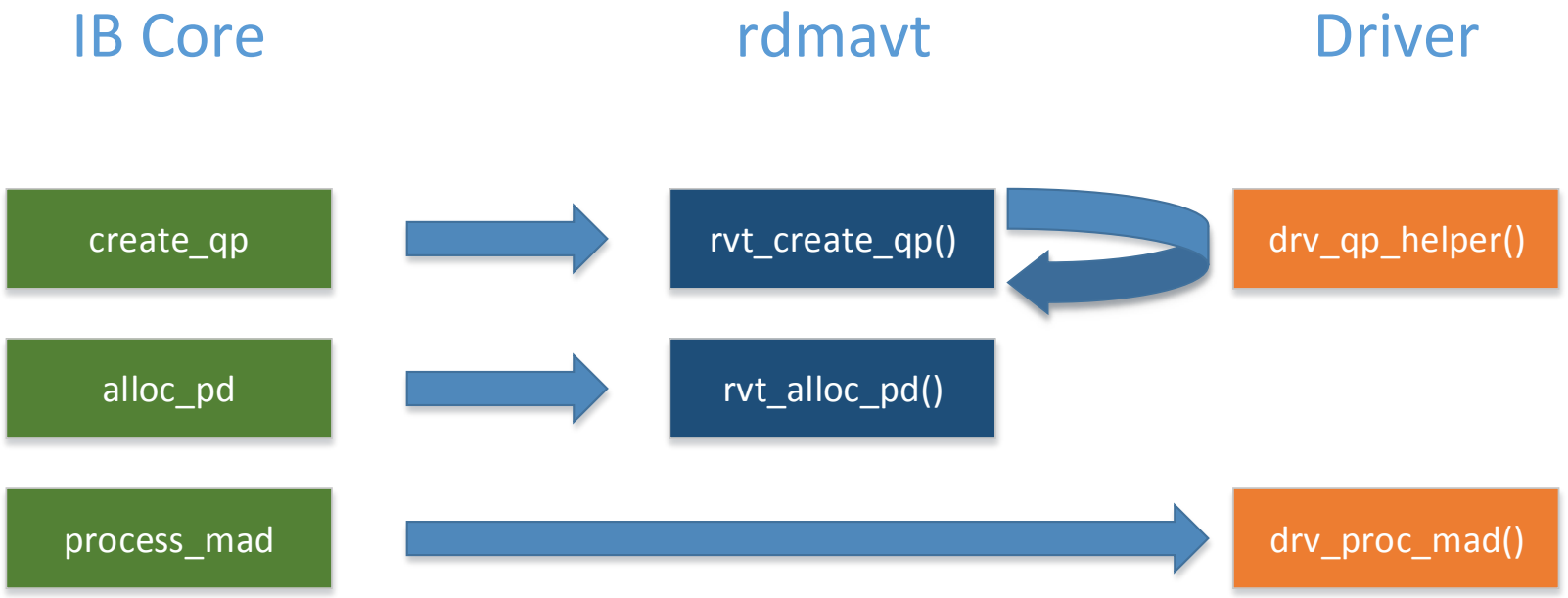
RDMAVT
technical details

- **Device registration**
  - Driver fills in a `struct ib_device` and passes to rdmavt
    - rdmavt needs this info, why have another struct for the same thing?
    - Function pointers for the verbs API
      - if NULL then will use rdmavt version of function
        - » rdmavt checks for required driver provided function(s)
      - if not NULL then the driver's version of that function will be used
        - » no dependency checks needed
  - rdmavt initializes resources
    - locks
    - worker threads
  - rdmavt eventually calls `ib_register_device()` on behalf of the driver

RDMAVT

technical details in a picture

- Too many functions to go into each one
- These are just examples of the different ways a verb is implemented
- May vary between drivers
- Reminder: rdmavt checks for any “helper” functions it might need at reg



RDMAVT
data path

- **Outgoing Data**
  - `rvt_post_send()`
    - Takes a list of work requests
      - Does some checks
      - Queues them for the driver to actually do the send (may kick the driver)
        - » Driver handles calling rdmavt to add CQ entries
        - » Driver handles protocol work
  - Would most likely be a driver specific function if other drivers come along
    - For now it was dead solid duplication so moved
- **Incoming Data**
  - Completely hardware specific (other than post_recv)
  - Driver handles calling rdmavt to add CQ entries
  - Driver handles protocol work

ROAD TO ACCEPTANCE

OPEN DEVELOPMENT PROCESS

- **rdmavt was developed totally in the open**
  - The high level overview was submitted to linux-rdma
  - Went right into writing code... just get it done and make it work
  - Another vendor even submitted patches early on
- **GitHub**
  - Used GitHub repo to make code available early
  - Served as a merged tree
  - Due to the volume of patches provided a point of reference to ensure ordering
  - Set up with 0-day builds
    - Found a few issues
- **Git log shows the development history**
  - We did not submit a “final” version. Rather a continuous work in progress.

CHALLENGES

- **rdmavt is a new driver**
  - hfi1 is a new driver in drivers/staging
  - qib is a stable driver in drivers/infiniband
- **Problems:**
  - qib can not depend on a driver in staging
  - Different maintainers for staging and linux-rdma
  - **Two trees are not the same!**
    - One reason we used GitHub to have a public tree
- **Solution:**
  - Agreement that linux-rdma maintainer would take over drivers/staging/rdma for 4.5
    - Mostly happened, but there are still patches sent to staging list
  - Have rdmavt ready to go by the start of the 4.6 merge window
    - Not just rdmavt, but hfi1 and qib as well

RDMAVT PATH TO THE KERNEL

- **For 4.6**
  - Over 300 patches submitted
    - rdmavt driver
    - qib changes for rdmavt
    - hfi1 changes for rdmavt
    - hfi1 style and bug fixes, some new content
    - hfi1 fixes that were sent to staging but were left in limbo during cut over
  - A ton of work by a lot of people
    - Too many to list, credit is in the git log
- **Number of contentious threads on linux-rdma**
  - Not going to go into further details, it is in the archives though
- **Accepted and appears in Linus' tree for 4.6!**

NEXT STEPS

- **What's next?**
  - Protocol work
    - Is it feasible given performance requirements?
  - Other duplicated, non-verbs code
    - Does it go in rdmavt?
    - Do we need something else?
  - Continue to evolve as needed

LEGAL DISCLAIMERS

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel, Intel Xeon, Intel Xeon Phi™ are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

OPTIMIZATION NOTICE

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

12th ANNUAL WORKSHOP 2016

THANK YOU

Dennis Dalessandro, Network Software Engineer

Intel