Datacenter Fabric Workshop

iWARP support In OpenIB

Clem Cole

Ammasso

August 22, 2005

Goals for this Talks

- Report on who, what and why we are doing this.
- Give some background behind it
- Explain what has been done and where we are heading
- Start/Continue the dialogue to ensure we get to where the community wants to be
- Solicit feedback on direction and suggestions for requirements

Who – The Players

- OpenIB group invited RNIC vendors to add iWARP support to OpenIB code base
- Not yet to the point of every voice singing praises
- But at least three RNIC vendors accepted invitation:
  - Ammasso
  - NetEffect
  - Siliquent
- Other participants are welcomed and encouraged

Why End Users/Interconnect Customers Find This Important

- Minimize Time to Market for
  - RDMA, IB, iWARP
- Better “Out of Box Experience”
  - HW support in the distro
  - common ULP’s, user mode API’s & kernel interfaces
  - choice in Transport Providers
- Increase adoption of RDMA Transports by ISVs
  - Simpler qualification for ISVs
- Increase code stability, maturity and market acceptance of RDMA technology

The Idea

OPEN IB ALLIANCE

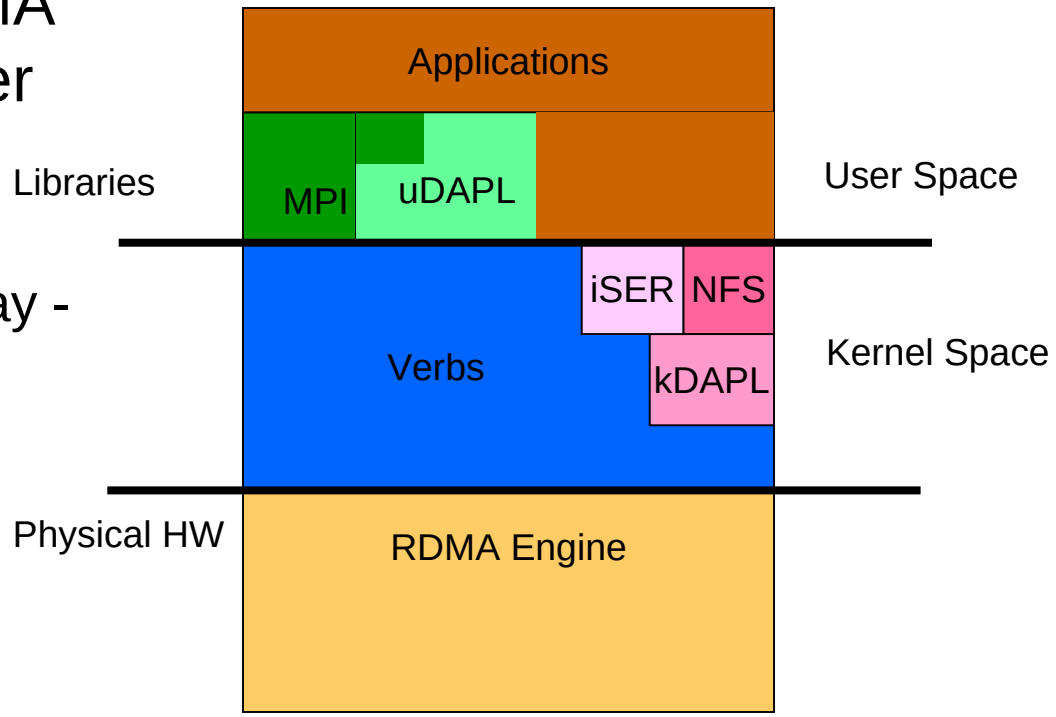
- Create a mechanism for multiple HW vendors to:
  - Plug their RDMA transport provider into the Linux OpenIB RDMA stack
  - By creating single OpenIB verbs interface to support both iWARP and IB

OpenIB Project Goals

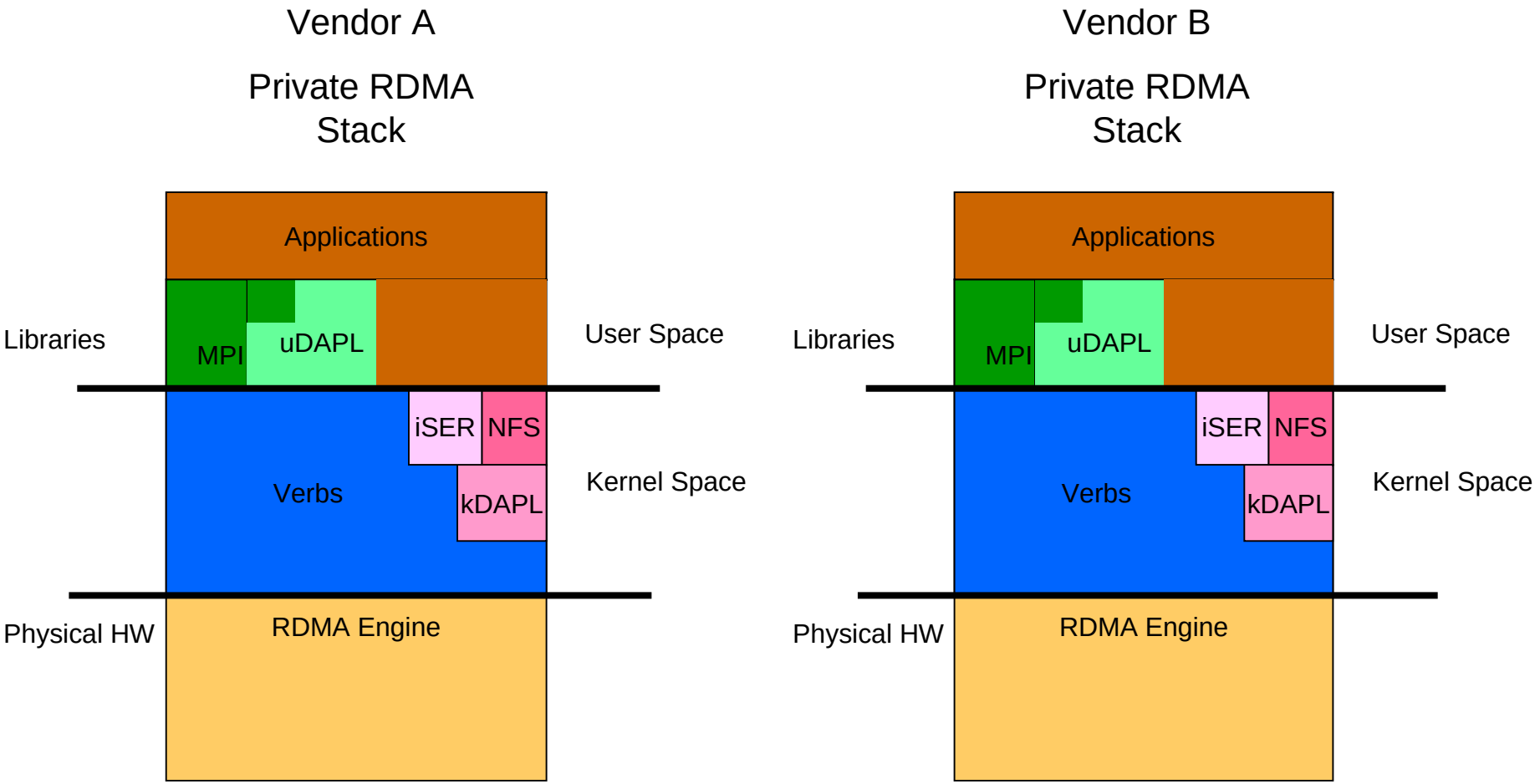
- Multi-vendor support
- Vendor interoperability (within transport families)
- Ease of portability for existing OpenIB apps
- Low application overhead for data transfer
- Supports multiple ULP's, User mode APIs and Kernel Applications including but not limited too:
  - socket extensions
  - async-io
  - DAT – kDAPL/uDAPL
  - IT API
- Speed up acceptance of RMDA into Linux
  - If it's not in kernel.org – tain't going to be in the Linux distros

A Linux RDMA Protocol Stack

- Problem Statement
  - At some layer, all RDMA stacks must tie together
    - Library layer [today - greens]
    - Kernel ULP [sort of today - pinks]
    - Common verbs [blue]



Same Application – Two Different Stack Implementations

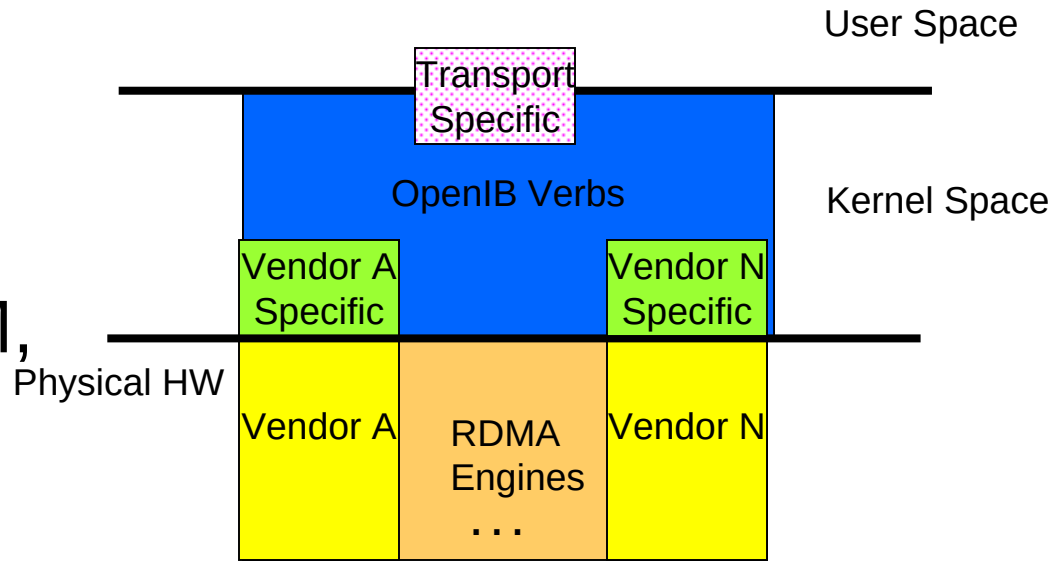


The Issue

- Linux Community finds multiple RDMA frameworks unacceptable
  - Too confusing
  - To much to absorb by too many people
- So, where is the “connection point”?
  - Today, we connect at the application
  - This is too high in the stack, forcing replication
- The lower the connection point, more can be shared between stacks and simplify testing *etc*.

Core RDMA Framework

- Supports multiple vendor's HW
- Keep the HW specific driver layer thin
- *Allow* for Transport specific things like (OpenSM, iWARP CM, *etc*.) as needed



Don’t Duplicate the Effort!

- RDMA is needed/exploited by multiple efforts (including but limited too):
  - iSER/iSCSI
  - NFS/RDMA & NFSv4
  - MPI
  - DAT – uDAPL / kDAPL
  - Other OS [Windows, proprietary UNIX, embedded *etc*.]
- Today, these are all replicated per vendor

Don’t Duplicate the Forum!

OPEN IB ALLIANCE

- Separate Linux efforts is being seen as competing by users and potential users and ISVs
  - OpenIB
  - OpenRDMA
  - Vendor specific
- This confuses customers/end users and delays the acceptance of RDMA in Linux
- Getting a stack in Linux is difficult to do once, much less twice, thrice, *etc*...

Solution

OPEN IB ALLIANCE

- The obvious place to make this cut is at the verbs layer
  - This is where both the commonalities and the primary differences lie
  - Below this layer, the HCA / RNIC vendors innovate

Guiding Directives

- OpenIB & iWARP support into the Linux core as soon as possible
- Any solution is not to be viewed as an end-all-be-all
- Any solution is alive and is expected to mature in the future
  - Things change…
- Be focused – start with pieces that are known to work
  - Time to market is key, people are tired of waiting for RDMA
  - Start with current OpenIB gen2 and Ammasso's 1100 code bases
- Make a working iWARP patch that is minimally intrusive to OpenIB & kernel.org
  - Minimize learning curve of Linux community
  - Add the minimum to IB data structures/enumerations to get the job done
- Maximize # of common verbs that support both IB and iWARP

Non-Goals

- Create a CM that works for both IB and iWARP at time t0
  - *i.e.* don't try to be be "perfect" – "good enough is ok"
- Come up with a solution to work on any OS other than the current Linux 2.6 development stream
  - In fact, we **are** trying to make the code conform to the Linux and OpenIB norms
- Try to develop a solution for every known RDMA fabric
  - But don't toss the baby out with the bath water (*i.e.* feel free listen, learn and exploit solutions from current and past lessons of VIA, DAPL, RNIC PI, *etc*. as appropriate)
  - However, you don't have to be bound by past history

Phase One

- Add Ammasso 1100 RNIC code in a OpenIB branch so everyone can see the work [completed and active commentary underway]
- Make iWARP code conform to “Linux” and OpenIB norms [in process]
  - Code rework (typedefs, data structs, enums)
  - Build using OpenIB methods
  - Error and return code normalization
- Extend the OpenIB device registration data structures to include connection establishment verbs.
  - This preserves compatibility for existing applications and avoids addressing the Linux host stack integration issues at the start

Phase One – cont.

- Modification of the existing DTO data types and enumerations to include support required by iWARP, but minimize the impact to existing OpenIB applications
- Modification to kDAPL to include support for iWARP transport providers
- Ability to use Linux network management tools to query, configure and control iWARP interfaces.
  - Plan is to achieve initially with a NETDEV pseudo-device for RDMA ports

Proposed Items for Later Phases

- Ability to migrate host established connections to an iWARP QP
- Handle differences in IB and iWARP CM models

- Other??

Status

- Caitlin Bestler of Siliquent and Tom Tucker of Ammasso have agreed to be joint iWARP for OpenIB Maintainers
- Using a phased approach
  - preserves the time to market goal
  - improves the likelihood of initial acceptance by the OpenIB group
- OpenIB trunk was branched 8/8/2005
- Ammasso 1100 code is being added to that branch as we speak

Datacenter Fabric Workshop

OPEN IB ALLIANCE

Open Discussion