Linux Kernel RDMA API

Tom Talpey
Network Appliance
April 26, 2005

Problem

- There can’t be two RDMA frameworks in Linux
- At some layer, all Linux RDMA stacks must tie together
- So, where is the “cleavage point”?
- The lower this cut is made, the better

Proposal

- The obvious place to make this cut is at, or just below, the Verbs layer
  - This is where the differences lie
  - This is where HCA/RNIC vendors innovate
  - There are plenty of examples ☺
- It would be a disaster to expose the cut to applications
  - DAPL can be used to smooth this however

A solved problem

- Verbs abstractions are not new, and not hard
  - DAPL has been ported to several of them, for example
- Several vendors have iWARP verbs implemented
- And RNIC-PI spec is out for comment

Time is of the Essence

- OpenIB is already part of 2.6.11
  - And on-by-default in several distros
- OpenIB defines Verbs, RDMA kernel infrastructure
- OpenIB is the default RDMA template

Don’t Duplicate the Effort!

- RDMA is needed by numerous efforts
  - iSER
  - NFS/RDMA
  - MPI
  - User DAPL
  - Xen
  - Windows
- OpenIB is already addressing these

Don’t Duplicate the Forum!

- Any separate effort would be seen as competing by users and potential users
- This would confuse customers/users and delay the acceptance of iWARP in Linux
- It would also be difficult to do. Twice.

An Upper Layer perspective

- Consider the NFS/RDMA client and server in Linux (two separate implementations, actually), currently using kDAPL
- Without kDAPL, if OpenIB and OpenRDMA are not compatible, I end up with two copies of the upper layer!
  - Users must install two new, separate NFS/RDMA kmods???
  - Or, NFS/RDMA must duplicate code just to support functionally equivalent xprts?

Proposal(s)

- Immediately begin working with OpenIB to shape Linux kernel RDMA APIs
- Identify necessary extensions to OpenIB infrastructure to include iWARP support
  - Connection model, node naming likely issues
- Consider RNIC-PI as a candidate for unification
  - But consider alternatives too