Datacenter Fabric Workshop

kDAPL

Progress Report On Standardization of RDMA APIs

Arkady Kanevsky, Ph.D
Chair of DAT Collaborative

August 22, 2005

What happened since last workshop?

- GPL v2 has been added
  - allowed submission of DAPL SF RI to OpenIB
- Successful DAPL Plugfest #2
  - both IB and iWARP members participated
  - kDAPL and uDAPL tests
  - PR released
- ATS v1 spec on DAT reflector
  - ready for ratification
- DAPL 1.3 work in progress

DAPL spec next version - I

- addition of iWARP and IBTA v1.2 functionality
- What has been approved and is in spec draft?
  - socket based connection model
  - addition of DTO type in completion event
  - RMR context for RDMA Read local data
  - RMR protection scope PZ and EP and new RMR_create
  - LMR triplet format
  - RMR bind RMR_handle argument added
  - connection request private data truncation exposure
  - requested data transfer length clarification & error behavior
  - RDMA Read to RMR
  - kDAPL physical pages of one size registration

DAPL spec next version - II

- What is being addressed now?
  - FMR
    - Memory Region allocation and binding
    - remote and local invalidation
  - 0-based virtual addresses
  - errata
  - binary and source backwards compatibility
    - DAPL 1.3 or DAPL 2.0

Datacenter Fabric Workshop

kDAPL

kernel Direct Access Programming Library (kDAPL)

James Lentini

Network Appliance

jlentini@netapp.com

August 22, 2005

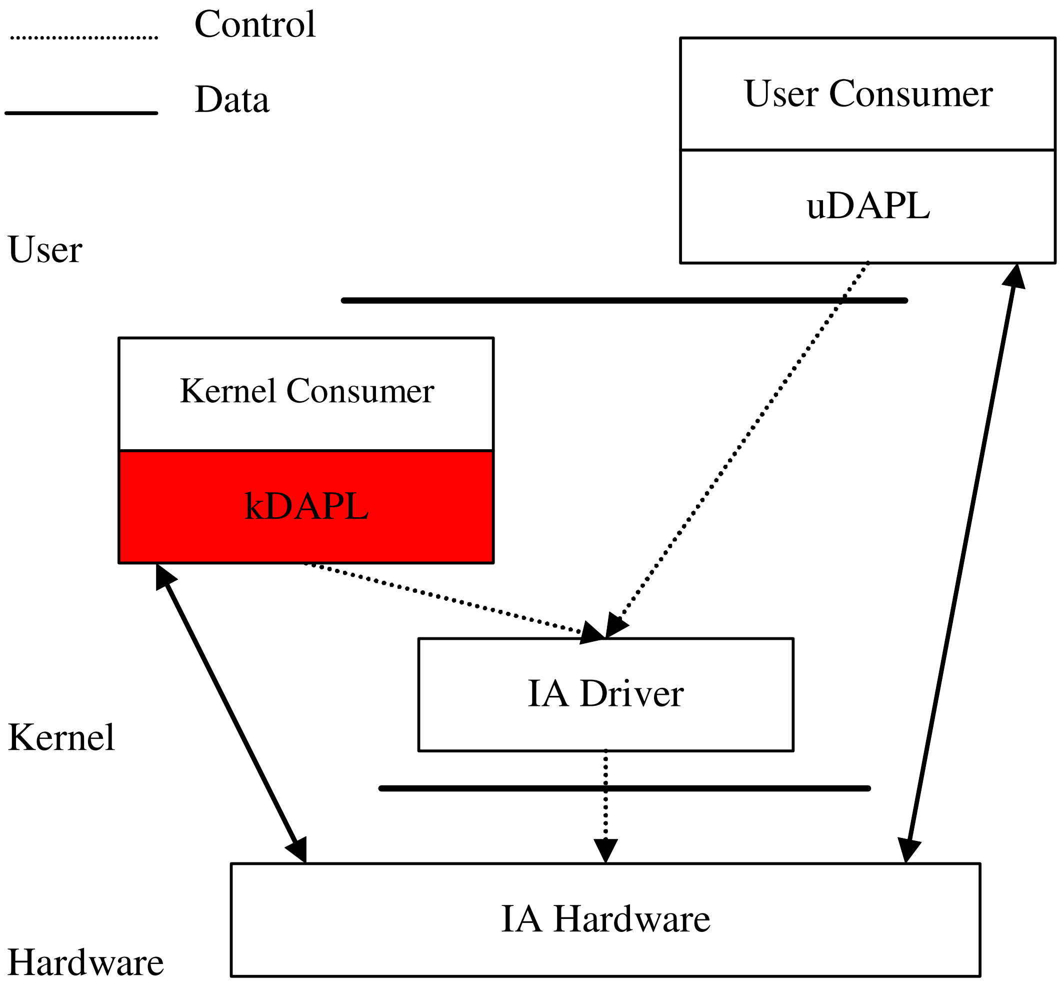
Overview

- Kernel interface for RDMA networks
  - generic interfaces for establishing connections, event processing, memory registration, and data transfer operations
- Based on DAT Collaborative kDAPL Specification, Version 1.2
- Modifications for Linux kernel design and coding standards

Features

- Ability to support both InfiniBand and iWARP
- Simple connection API
  - BSD Sockets-like
  - Internet Protocol (IP) addressing
- Unified event model
  - connection request
  - connection establish/disconnect
  - data transfer operations (DTOs) and memory binds
  - software events
  - asynchronous errors

Architecture



Testing

- kdapltest test tool used to test implementation. Two primary modes:
  - – Transaction Testing: simulates a transaction based protocol
  - – Performance Testing: pipelined RDMA read or write performance test

kDAPL Consumers

- NFS-RDMA
  - client: http://sourceforge.net/projects/nfs-rdma
  - server: http://www.citi.umich.edu/projects/rdma/
- iSER (iSCSI Extensions for RDMA)
  - initiator: https://openib.org/svn/gen2/trunk/src/linux-kernel/infiniband/ulp/iser/

Source Code

- kDAPL located at

  https://openib.org/svn/gen2/trunk/src/linux-kernel/infiniband/ulp/kdapl/

  – README contains configuration instructions

- kdapltest located at

  https://openib.org/svn/gen2/utils/src/linux-kernel/kdapl/dapltest/

  – README contains command usage

Thanks!

- Thanks to Tom Duffy, Bernhard Fischer, Sean Hefty, Christoph Hellwig, Itamar Rabenstein, and Hal Rosenstock for their help porting the code.

Datacenter Fabric Workshop

kDAPL

Progress Report On OpenIB uDAPL

Arlin Davis – Intel Corporation
arlin.r.davis@intel.com

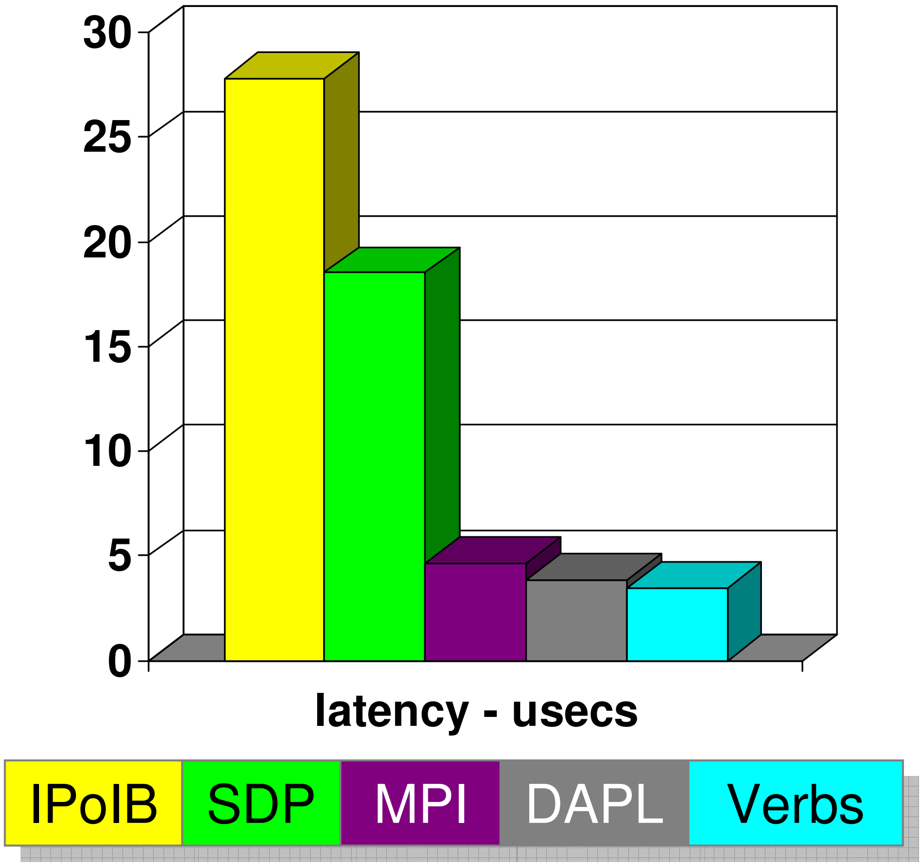
August 22, 2005

What happened since last workshop?

- New 1.2 provider for OpenIB
- Developed in stages
  - uVerbs and socket-based CM
  - uVerbs/uCM using hard coded path records
  - uVerbs/uCM/uAT
- Code recently moved to trunk
- Features
  - Standard features minus RMR's, SRQ
  - inline sends

uDAPL developer testing

- Simple
  - dtest
  - netpipe
- Scale up
  - dapltest
    - regress.sh (1500+)
- Scale up and out
  - Intel-MPI test suite
    - 2000+ tests
    - 32 nodes – 128 processes



uDAPL todo list

- uCM fix, ib_cm_init_qp_attr
- use ibv calls for GID and attributes
- add async event processing
- consolidate async, uAT, uCM work threads
- shared receive queues
- DAT 1.3 modifications

Still under discussion

- direct CQ wait objects
- memory windows
- merged EVD support (connect/dto)
- shared memory support
- build tree